Document Title	Diagnostic Extract Template
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	673

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

Document Change History

Date	Release	Changed by	Description
2017-12-08	4.3.1	AUTOSAR Release Management	• minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2016-11-30	4.3.0	AUTOSAR Release Management	• Support for OBD • Support for J1939 • Support for Fim configuration • Support for environmental conditions • Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2015-07-31	4.2.2	AUTOSAR Release Management	• Minor corrections / clarifications / editorial changes; For details please refer to the ChangeDocumentation
2014-10-31	4.2.1	AUTOSAR Release Management	• Initial Release

Disclaimer

Table of Contents

Bibliography

[1] Specification of RTE Software
AUTOSAR_SWS_RTE

[2] Layered Software Architecture
AUTOSAR_EXP_LayeredSoftwareArchitecture

[3] Specification of ECU Configuration
AUTOSAR_TPS_ECUConfiguration

[4] ASAM MCD 2D ODX
http://www.asam.net
ASAM MCD-2D ODX v2.2.0.pdf

[5] XML Schema Production Rules
AUTOSAR_TPS_XMLSchemaProductionRules

[6] System Template
AUTOSAR_TPS_SystemTemplate

[7] Specification of ECU Configuration Parameters (XML)
AUTOSAR_MOD_ECUConfigurationParameters

[8] Software Component Template
AUTOSAR_TPS_SoftwareComponentTemplate

[9] Basic Software Module Description Template
AUTOSAR_TPS_BSWModuleDescriptionTemplate

[10] Specification of Diagnostic Communication Manager
AUTOSAR_SWS_DiagnosticCommunicationManager

[11] Specification of Diagnostic Event Manager
AUTOSAR_SWS_DiagnosticEventManager

[12] Standardization Template
AUTOSAR_TPS_StandardizationTemplate

[13] Requirements on Diagnostic Extract Template
AUTOSAR_RS_DiagnosticExtractTemplate

[14] Specification of Function Inhibition Manager
AUTOSAR_SWS_FunctionInhibitionManager

[15] SAE J1939 Top Level Document

[16] Unified diagnostic services (UDS) – Part 1: Specification and requirements (Release 2006-12)
http://www.iso.org

[17] Road vehicles – End-of-life activation of on-board pyrotechnic devices – Part 2: Communication requirements

http://www.iso.org

[18] Road vehicles – Communication between vehicle and external equipment for emission-related diagnostic – Part 5: Emission-related diagnostic services.
http://www.iso.org

[19] Road vehicles – Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements – Part 3: Common message dictionary
http://www.iso.org

[20] SAE J1939-73 Application Layer – Diagnostics

[21] Software Process Engineering Meta-Model Specification
http://www.omg.org/spec/SPEM/2.0/

[22] Generic Structure Template
AUTOSAR_TPS_GenericStructureTemplate

1 Introduction

1.1 Overview

The distributed nature of an AUTOSAR ECU development requires an optimized capturing of information. In particular, diagnostic information (i.e. DEM and DCM configuration) shall be captured only once by the person with the best knowledge and therefore being able to take responsibility better than one centralized individual.

In the configuration approach before the advent of the `DiagnosticExtract`, the Basic Software Modules DCM and DEM are entirely configured centrally. During integration, all SW-Cs above the RTE [1] (Application Software) introduce ports to be connected to the BSW modules [2]. Additionally, SW-Cs express needs which shall be fulfilled by the BSW.

The market shows a high demand for transferring diagnostic demands of the OEM-specific configuration process to their tier-1 suppliers.

In the past, due to the absence of integral options, many different file formats like ODX or EcuC [3] are often used. But neither ODX nor EcuC is well suited to transfer this information.

For example, ODX [4] lacks in fault memory details and EcuC (which was never designed for becoming the vehicle for data exchange between different organizations) has a very generic nature that renders the enforcement of a strict model formalization very difficult.

On top of that, the integration of EcuC definitions into an existing configuration (especially the PDUs) cannot be fully automated.

Therefore, the obvious solution approach has been to define a new standardized AUTOSAR exchange format on diagnostic functionality that can be used similar to a System Description, formalized as an ARXML [5] file.

In this spirit, the configuration of diagnostic functionality becomes similar to the configuration of the communication part within the System Description [6].

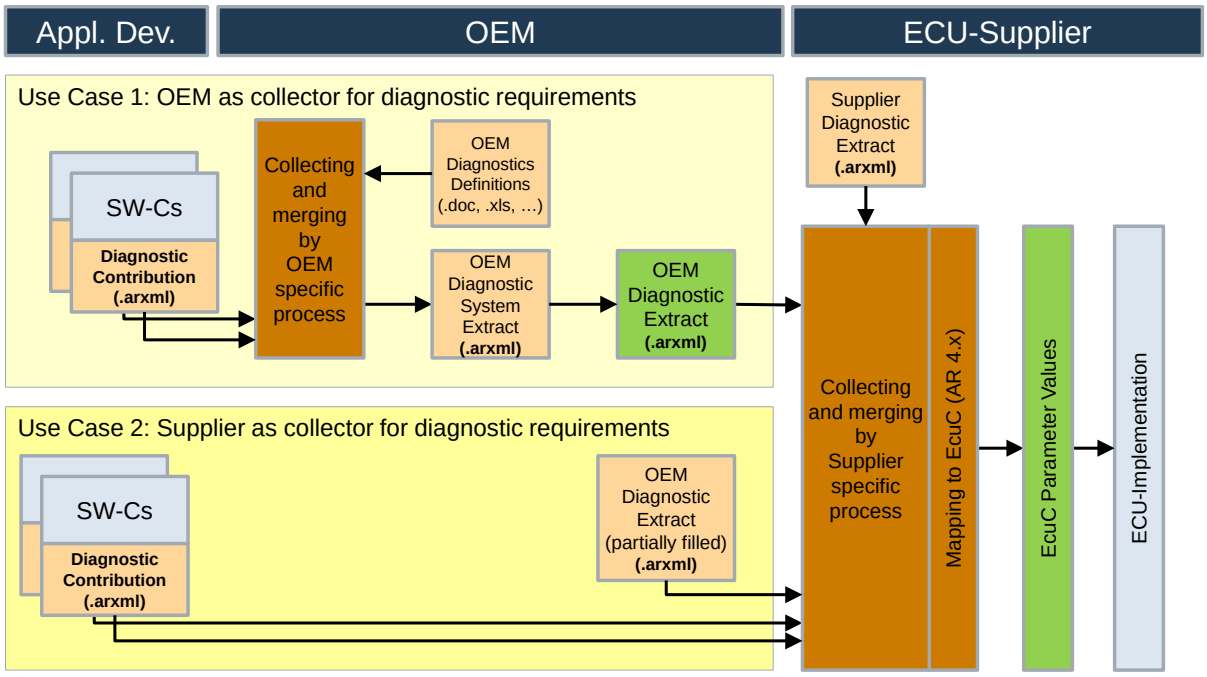

Figure 1.1: Scope of this document in the ECU Development work-flow

Figure 1.1 shows the configuration process of diagnostics for two generalized use cases. This process involves three parties:

- **OEM** or diagnostic requester
- **Application Developer** or Application Developer
- **ECU-Supplier** or integrator

The specific role of these contributors to the diagnostic extract is in detail explained in the following sub-chapters.

1.1.1 OEM

The **OEM** or requester of diagnostic data uses the `DiagnosticExtract` to define the diagnostic interfaces of one or multiple ECUs. It may also define some `InternalBehavior`s as requirements for the **ECU-Supplier** or **Application Developer**

- Defines the values of the DTCs
- Defines the UDS services and sub-services supported by the ECUs
- Defines the required events needed by a specific composition implemented by an **Application Developer**

NOTE: This list represents an example; this document does not define a specific ownership of each element.

In the first use case, the DiagnosticExtract is used to exchange information which is transformed into the EcuC configuration (M2 to M1 mapping, see also [3] and [7]).

Second, the **OEM** uses the DiagnosticExtract to document requirements to be implemented by a supplier. These requirements are expressed in textual language and can not be mapped directly to any EcuC configuration parameters (no M2 to M1 mapping possible).

1.1.2 Application Developer

The Application Developers implement their **software-components** with the corresponding **software-component description**. The role "Application Developer" can be assumed by both an OEM and a supplier. In other words, both OEM and supplier may contribute application software to a given ECU.

With the introduction of this concept, the Application Developer has the possibility to provide diagnostic information relevant to the software-components as part of the DiagnosticExtract.

The Application Developer may also receive some input as requirement from the **OEM** within the DiagnosticExtract in textual form as for example:

- Definition of the content of a specific ReadDataByIdentifier implemented by this software-component
- Definition of the events needed for this software-component

NOTE: Only as example, this document does not define a specific ownership of each element.

In the first use case, the Application Developer defines the parameters of a specific ReadDataByIdentifier, i.e. the content of the diagnostic request but not the DID itself. The DID of this command will usually be defined by the **OEM**.

Secondly, the software-component events including information like Debouncing and OperationCycle may be defined by the **Application Developer**. The **Application Developer**may also define events and diagnostic jobs which are not needed by a specific **OEM** but for another one.

Suppliers may use the same software for multiple **OEMs** and need to reuse it. This implies that some DiagnosticExtract information coming from a software-component may be ignored during the integration if not needed for a specific project.

1.1.3 ECU-Supplier

The **ECU-Supplier** or integrator receives one or several DiagnosticExtract files from the **OEM** and from multiple **Application Developers**. The main goal of the **In-**

tegrator is to integrate all delivered DiagnosticExtract and to generate the EcuC configuration from it.

Since this concept does not define a specific ownership for each element like DIDs, parameters of a UDS service, Events, Sessions, etc. the integrator has to ensure that the complete information is still valid after merging it.

- Mapping of DTCs to Events
- Merge of Events
- Mapping of services

Some DTCs may already be mapped to events - especially in cases where both come from the same party. But if the DTCs are defined by the **OEM**and the SW implemented by other supplier acting as an **Application Developer** the integrator has to ensure that both are mapped together.

In some cases, an Event may be defined multiple times. An **OEM** defines the Events which shall be implemented by an **Application Developer**. A Supplier implements a software-component which will be used in multiple projects and which also detects this type of error and also defines this same event.

Both events may have different naming but the same meaning. The integrator has to detect this redundancy during the integration and merge them together.

In another case, the **OEM** requires a specific ReadDataByIdentifier and an **Application Developer** implements it. If the implementation is performed for one specific project only, the **Application Developer** may map the DID from the **OEM** to the already defined job in their software-component.

In other cases in which the **Application Developer** implements a generic diagnostic job, it will be a task of the **ECU-Supplier** to merge this information and to map the jobs to the corresponding DID.

1.1.4 Exchanging of Files

During an ECU development project, the three main roles (OEM, Application Developer, ECU Integrator) exchange DiagnosticExtract files. The timing and frequency of exchanges and the content in each of these exchanged files is highly dependent on the individual project setup and situation.

Therefore, the DiagnosticExtract format has been designed to allow for gradual enrichment of definitions contributed at largely arbitrary points in time by the different roles in order to meet the needs of “Decentralized Configuration”.

For any exchange path between any two roles, the same file format based on the DiagnosticExtract template is used. It is then up to a company specific process and tooling to merge the collected DiagnosticExtract files while resolving conflicts (contradictions, redundancies etc.).

As final result, a consistent and complete DiagnosticExtract file is available which is input to derivation of the configuration for the diagnostic modules of the Basic Software.

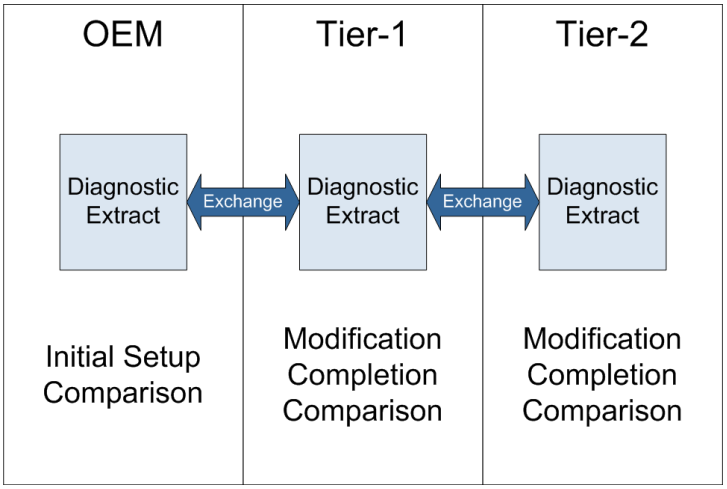

Figure 1.2: Exchange of diagnostic configuration between OEM, Tier-1, and Tier-2

Even after the DiagnosticExtract has been fully integrated and is ready to go for deriving the configuration of the diagnostic stack on EcuC level it is still foreseen to feed it back to e.g. an OEM.

In this case the OEM has the ability to review the configuration of the diagnostic stack on the level of the diagnostic extract.

At some point, this information may also be taken to (directly or via indirection of other formats) create a configuration of a diagnostic client.

1.1.5 Relationship to software-component Service Needs

Before the introduction of the Diagnostic Extract, Service Needs were used to describe diagnostic requirements on software-component level. These configuration requirements are referenced to the related BSW module DCM or DEM in order to provide the corresponding configuration on BSW level.

The usage of Service Needs is only possible on Atomic software-component level whereas the assignment of diagnostic demands must be possible on Composition level.

The software-component Service Needs within the software-component Description are still to be used along with the DiagnosticExtract in order to annotate the software-component ports which are relevant for further mapping and handling as defined by the DiagnosticExtract.

From software-component Developer's perspective, the DiagnosticExtract is therefore used partially as substitution and partially as extension of the software-

component Service Needs. The reasoning for “substitution” is the avoidance of redundant diagnostic definitions.

Since some diagnostic properties potentially definable by software-component Developers are not covered in software-component Service Needs, the DiagnosticExtract can also be viewed as “extension” to the original purpose of software-component Service Needs.

1.1.6 Recommendation and Hints

Multiple parties may have different understanding of which parts shall be provided by each one. There is no defined rule to indicate who is responsible for each part. At the end, it is the **ECU-Supplier** in his role as integrator who has to ensure that all mappings are done and that the ECU runs as expected by the **OEM**.

In case the **OEM** does not have his own diagnostic requirements, the **ECU-Supplier** has to provide the complete DiagnosticExtract. In this case, the **OEM** may only receive the DiagnosticExtract as part of the delivery. The process itself how the parties work with this format is not defined within this specification.

Figure 1.1 shows a recommended way how to handle the DiagnosticExtract between the different parties. In use case 1, some software-components are implemented by the **OEM** (or by a Supplier of the OEM) and the first merging of DiagnosticExtract data occurs at the **OEM**.

In use case 2, the **OEM** provides the diagnostic requirements via DiagnosticExtract and multiple **Application Developer**provide information related to their implementation, the merging is performed completely by the **ECU-Supplier**.

Also a combination of use cases 1 and 2 is allowed. Also the **ECU-Supplier** may implement some part of the SW inclusive their corresponding DiagnosticExtract.

1.1.7 Limitations

This first release of the DiagnosticExtract template focuses on defining diagnostic requirements a single ECU only. That means that currently, distributed diagnostic functionality for a system or partial system consisting of multiple ECUs cannot be defined using the DiagnosticExtract template.

In future releases, the DiagnosticExtract template is expected to be extended to also cover configuration of distributed diagnostic functionality. Similar to the description of communication dependencies in the System Description, it shall be possible to describe diagnostic demands on system level to derive the diagnostic demands for a specific ECU from this description.

In general, the DiagnosticExtract does not support process-related parts to document the maturity of diagnostic configuration data. This means, that a data object

cannot be marked as “draft” or “released”. This issue needs to be solved by AUTOSAR within a general concept in a future release. Therefore, it does not make sense to introduce a solution for diagnostics only.

1.2 Scope

This document describes the formal description of contributions to the diagnostic configuration.

On the level of meta-modeling, the content described in this document conceptually relates to the definition of `SwcServiceDependency` resp. `BswServiceDependency`, as defined by the `Software Component Template` [8] resp. `Basic Software ModuleDescription Template` [9].

Further relations exist to the specification of communication in AUTOSAR systems as described by the `System Template` [6].

Further relations exist to the specification of the `Diagnostic Communication Manager` [10] as well as to the `Diagnostic Event Manager` [11].

The relation of the `DiagnosticExtract` to the rest of the AUTOSAR meta-model is sketched in Figure 1.3.

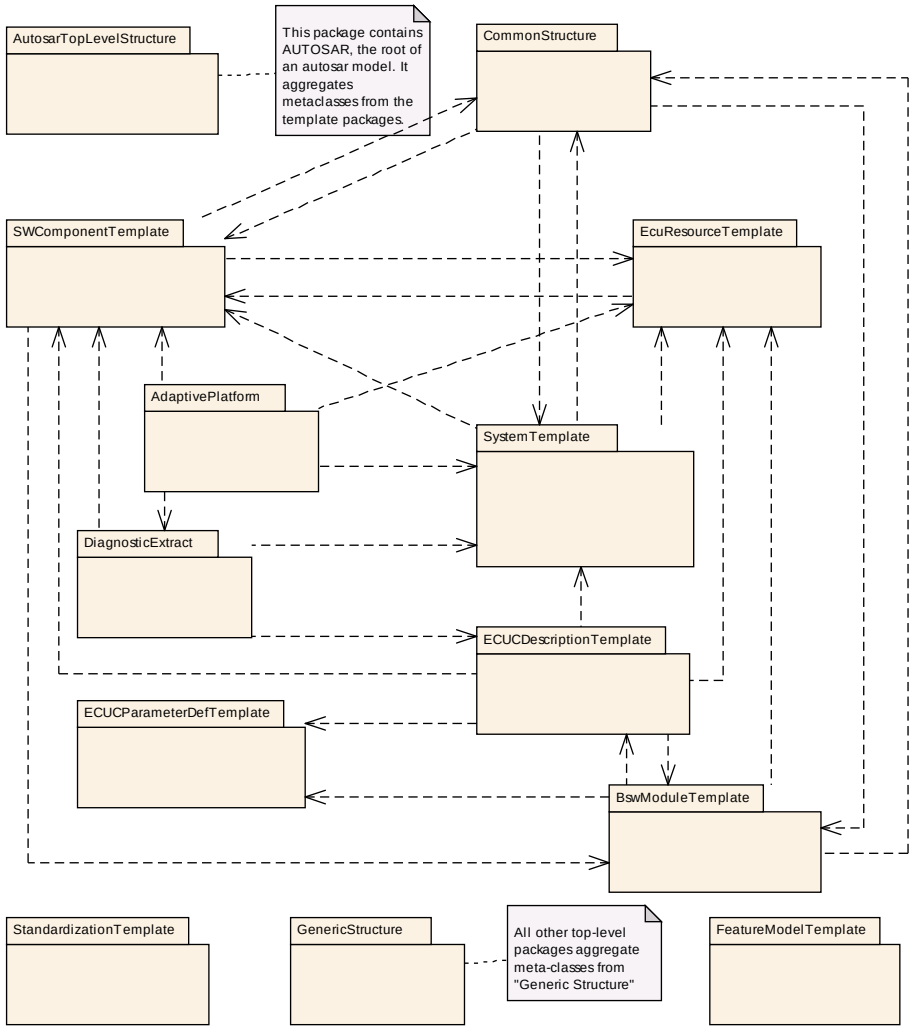

Figure 1.3: The relation of the `DiagnosticExtract` to the rest of the AUTOSAR meta-model

1.3 Abbreviations

The following table contains a list of abbreviations used in the scope of this document along with the spelled-out meaning of each of the abbreviations.

Abbreviation	*meaning*
API	Application Programming Interface
BSW	Basic Software
BswM	Basic Software Manager
CAN	Controller Area Network
CSE	Codes for Scaling Units
DEM	Diagnostics Communication Manager
DCY	Driving Cycle
DEM	Diagnostics Event Manager

Abbreviation	*meaning*
DID	Diagnostic Identifier
DTC	Diagnostic Trouble Code
DTR	Diagnostic Test Result
DoIP	Diagnostics over IP
ECU	Electrical Control Unit
ECUC	ECU Configuration
FID	Function Identifier
FIM	Function Inhibition Manager
GID	Group Identifier
ID	Identifier
IO	Input/Output
IP	Internet Protocol
IUMPR	In-Use Monitor Performance Ratio
ISO	International Organization for Standardization
LIN	Local Interconnect Network
NRC	Negative Response Code
OBD	On-Board Diagnostic
ODX	Open Diagnostic Data Exchange
OEM	Original Equipment Manufacturer
PDU	Protocol Data Unit
PID	Parameter Identifier
PTO	Power Take Off
RA	Routing Activation
RAM	Random Access Memory
RID	Routine Identifier
ROE	Response on Event
ROM	Read-Only Memory
RTE	Run-TIme Environment
RS	Requirements Specification
RX	Receive
SPN	Suspect Parameter Number
SW	Software
SWC	Software Component
SWCD	Software Component Description
TID	Test Identifier
TPS	Template Specification
TX	Transmit
SWS	Software Specification
UDS	Unified Diagnostic Services
UML	Unified Modeling Language
VFB	Virtual Functional Bus
VIN	Vehicle Identification Number
WWH-OBD	World-Wide Harmonized On-Board Diagnostics

Abbreviation	*meaning*
XML	Extensible Markup Language
XSD	XML Schema Definition

Table 1.1: Abbreviations used in the scope of this Document

1.4 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. `PortPrototype`s. 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:

Class	**AUTOSAR**			
Package	M2::AUTOSARTemplates::AutosarTopLevelStructure			
Note	Root element of an AUTOSAR description, also the root element in corresponding XML documents. **Tags:** xml.globalElement=true			
Base	ARObject			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
adminData	AdminData	0..1	aggr	This represents the administrative data of an Autosar file. **Tags:** xml.sequenceOffset=10
arPackage	ARPackage	*	aggr	This is the top level package in an AUTOSAR model. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variation Point.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30

Attribute	*Type*	*Mul.*	*Kind*	*Note*
fileInfoComment	FileInfoComment	0..1	aggr	This represents a possibility to provide a structured comment in an AUTOSAR file. **Tags:** xml.roleElement=true; xml.sequenceOffset=-10; xml.typeElement=false
introduction	Documentation Block	0..1	aggr	This represents an introduction on the Autosar file. It is intended for example to rpresent disclaimers and legal notes. **Tags:** xml.sequenceOffset=20

Table 1.2: AUTOSAR

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.

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see Standardization Template, chapter Support for Traceability ([12]).

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see Standardization Template, chapter Support for Traceability ([12]).

1.5 Requirements Tracing

The following table references the requirements specified in [13] and links to the fulfillment of these.

Requirement	Description	Satisfied by
[RS_DEXT_00001]	Diagnostic data exchange	[TPS_DEXT_01003] [TPS_DEXT_01004] [TPS_DEXT_01005] [TPS_DEXT_01007] [TPS_DEXT_01008] [TPS_DEXT_01014] [TPS_DEXT_01019] [TPS_DEXT_01020] [TPS_DEXT_01022] [TPS_DEXT_01023] [TPS_DEXT_01024] [TPS_DEXT_01025] [TPS_DEXT_01026] [TPS_DEXT_01027] [TPS_DEXT_01028] [TPS_DEXT_01029] [TPS_DEXT_01046] [TPS_DEXT_01055] [TPS_DEXT_01056] [TPS_DEXT_01057] [TPS_DEXT_01060] [TPS_DEXT_01062] [TPS_DEXT_01063] [TPS_DEXT_01066] [TPS_DEXT_01069] [TPS_DEXT_01075]
[RS_DEXT_00002]	No description	[TPS_DEXT_01003] [TPS_DEXT_01004] [TPS_DEXT_01005] [TPS_DEXT_01038] [TPS_DEXT_01047] [TPS_DEXT_01055] [TPS_DEXT_01062] [TPS_DEXT_01063]
[RS_DEXT_00003]	SessionControl	[TPS_DEXT_01039] [TPS_DEXT_01045] [TPS_DEXT_01081] [TPS_DEXT_01082]
[RS_DEXT_00004]	ECUReset	[TPS_DEXT_01019] [TPS_DEXT_01020] [TPS_DEXT_01021] [TPS_DEXT_01045] [TPS_DEXT_01056]
[RS_DEXT_00005]	ClearDiagnosticInformation	[TPS_DEXT_01022] [TPS_DEXT_01045]
[RS_DEXT_00006]	ReadDTCInformation	[TPS_DEXT_01034] [TPS_DEXT_01045] [TPS_DEXT_01060]
[RS_DEXT_00007]	ReadDataByIdentifier	[TPS_DEXT_01045] [TPS_DEXT_01050] [TPS_DEXT_01054]
[RS_DEXT_00008]	ReadMemoryByAddress	[TPS_DEXT_01024] [TPS_DEXT_01045]
[RS_DEXT_00009]	SecurityAccess	[TPS_DEXT_01036] [TPS_DEXT_01037] [TPS_DEXT_01038] [TPS_DEXT_01045] [TPS_DEXT_01053]
[RS_DEXT_00010]	CommunicationControl	[TPS_DEXT_01029] [TPS_DEXT_01030] [TPS_DEXT_01031] [TPS_DEXT_01032] [TPS_DEXT_01045] [TPS_DEXT_01057] [TPS_DEXT_01074]
[RS_DEXT_00011]	ReadDataByPeriodicIdentifier	[TPS_DEXT_01045]
[RS_DEXT_00012]	DynamicallyDefineDataIdentifier	[TPS_DEXT_01045]
[RS_DEXT_00013]	WriteDataByIdentifier	[TPS_DEXT_01045] [TPS_DEXT_01050] [TPS_DEXT_01054]
[RS_DEXT_00014]	IOControl	[TPS_DEXT_01015] [TPS_DEXT_01016] [TPS_DEXT_01017] [TPS_DEXT_01018] [TPS_DEXT_01045] [TPS_DEXT_01051]
[RS_DEXT_00015]	RoutineControl	[TPS_DEXT_01035] [TPS_DEXT_01045] [TPS_DEXT_01049] [TPS_DEXT_01077] [TPS_DEXT_01078] [TPS_DEXT_01079] [TPS_DEXT_01080]
[RS_DEXT_00016]	RequestDownload	[TPS_DEXT_01027] [TPS_DEXT_01045]
[RS_DEXT_00017]	RequestUpload	[TPS_DEXT_01028] [TPS_DEXT_01045]
[RS_DEXT_00018]	TransferData	[TPS_DEXT_01026] [TPS_DEXT_01045]

[RS_DEXT_00019]	RequestTransferExit	[TPS_DEXT_01025] [TPS_DEXT_01045]
[RS_DEXT_00020]	WriteMemoryByAddress	[TPS_DEXT_01023] [TPS_DEXT_01045]
[RS_DEXT_00021]	ControlDTCSetting	[TPS_DEXT_01045] [TPS_DEXT_01075] [TPS_DEXT_01076]
[RS_DEXT_00022]	ResponseOnEvent	[TPS_DEXT_01033] [TPS_DEXT_01045]
[RS_DEXT_00023]	Configuration of events	[TPS_DEXT_01048] [TPS_DEXT_01067] [TPS_DEXT_01068] [TPS_DEXT_01069] [TPS_DEXT_01083] [TPS_DEXT_01084] [TPS_DEXT_01085] [TPS_DEXT_03002] [TPS_DEXT_03003] [TPS_DEXT_03004] [TPS_DEXT_03005] [TPS_DEXT_03007] [TPS_DEXT_03011] [TPS_DEXT_03015] [TPS_DEXT_03016]
[RS_DEXT_00024]	Configuration of DTCs	[TPS_DEXT_01064] [TPS_DEXT_01065] [TPS_DEXT_01066] [TPS_DEXT_01086] [TPS_DEXT_03000] [TPS_DEXT_03003] [TPS_DEXT_03012] [TPS_DEXT_03013] [TPS_DEXT_03014]
[RS_DEXT_00025]	Combined Events	[TPS_DEXT_03003]
[RS_DEXT_00026]	Enable Conditions	[TPS_DEXT_03015] [TPS_DEXT_03018]
[RS_DEXT_00027]	Storage Conditions	[TPS_DEXT_03001] [TPS_DEXT_03006] [TPS_DEXT_03010] [TPS_DEXT_03016] [TPS_DEXT_03019]
[RS_DEXT_00028]	Enable Condition Groups	[TPS_DEXT_01084] [TPS_DEXT_03010] [TPS_DEXT_03015]
[RS_DEXT_00029]	Storage Condition Groups	[TPS_DEXT_01084] [TPS_DEXT_03016]
[RS_DEXT_00030]	Assignment of Enable Condition Groups	[TPS_DEXT_03010]
[RS_DEXT_00031]	Assignment of Storage Condition Group	[TPS_DEXT_03010]
[RS_DEXT_00032]	Configuration of Extended Data Records	[TPS_DEXT_03008]
[RS_DEXT_00033]	Configuration of Snapshot Records	[TPS_DEXT_03009]
[RS_DEXT_00034]	Description of Data Identifiers	[TPS_DEXT_01000] [TPS_DEXT_01001] [TPS_DEXT_01002] [TPS_DEXT_01017] [TPS_DEXT_01050] [TPS_DEXT_01054] [TPS_DEXT_01072] [TPS_DEXT_01134] [TPS_DEXT_01135] [TPS_DEXT_01136] [TPS_DEXT_01137] [TPS_DEXT_01138]
[RS_DEXT_00035]	Description of Dynamic Data Identifiers	[TPS_DEXT_01000]
[RS_DEXT_00036]	Description of Routine Identifiers	[TPS_DEXT_01088]
[RS_DEXT_00037]	Description of I/O Identifiers	[TPS_DEXT_01089]
[RS_DEXT_00038]	Description of array data types	[TPS_DEXT_01001] [TPS_DEXT_01002]
[RS_DEXT_00039]	Diagnostic Service Table	[TPS_DEXT_01006]
[RS_DEXT_00040]	Diagnostic Sessions	[TPS_DEXT_01011] [TPS_DEXT_01081] [TPS_DEXT_01082] [TPS_DEXT_01139]
[RS_DEXT_00041]	Access Permissions	[TPS_DEXT_01012] [TPS_DEXT_01052] [TPS_DEXT_01061] [TPS_DEXT_01062] [TPS_DEXT_01063] [TPS_DEXT_01071]
[RS_DEXT_00042]	Security Levels	[TPS_DEXT_01012] [TPS_DEXT_01038] [TPS_DEXT_01053]

[RS_DEXT_00043]	Description of data elements	[TPS_DEXT_03020]
[RS_DEXT_00045]	Textual descriptions	[TPS_DEXT_01064] [TPS_DEXT_01065] [TPS_DEXT_01066] [TPS_DEXT_01067] [TPS_DEXT_01068] [TPS_DEXT_01069] [TPS_DEXT_01071]
[RS_DEXT_00047]	Custom Diagnostic Service	[TPS_DEXT_01009] [TPS_DEXT_01010] [TPS_DEXT_01021] [TPS_DEXT_01030] [TPS_DEXT_01031]
[RS_DEXT_00049]	Properties of individual diagnostic services	[TPS_DEXT_01013] [TPS_DEXT_01052] [TPS_DEXT_01061]
[RS_DEXT_00050]	Properties of all diagnostic services of a given kind	[TPS_DEXT_01061]
[RS_DEXT_00051]	Subfunctions of Diagnostic Services	[TPS_DEXT_01013] [TPS_DEXT_01014] [TPS_DEXT_01018] [TPS_DEXT_01019] [TPS_DEXT_01020] [TPS_DEXT_01021] [TPS_DEXT_01022] [TPS_DEXT_01023] [TPS_DEXT_01024] [TPS_DEXT_01025] [TPS_DEXT_01026] [TPS_DEXT_01027] [TPS_DEXT_01028] [TPS_DEXT_01029] [TPS_DEXT_01030] [TPS_DEXT_01031] [TPS_DEXT_01034] [TPS_DEXT_01039] [TPS_DEXT_01056] [TPS_DEXT_01057] [TPS_DEXT_01060] [TPS_DEXT_01075] [TPS_DEXT_01076] [TPS_DEXT_01078]
[RS_DEXT_00052]	Mapping of diagnostic services to the PortPrototypes of ApplicationSwComponentTypes	[TPS_DEXT_01040] [TPS_DEXT_01041] [TPS_DEXT_01042] [TPS_DEXT_01043] [TPS_DEXT_01044] [TPS_DEXT_01049] [TPS_DEXT_01050] [TPS_DEXT_01051] [TPS_DEXT_03002] [TPS_DEXT_03007] [TPS_DEXT_03017] [TPS_DEXT_03018] [TPS_DEXT_03019] [TPS_DEXT_03020]
[RS_DEXT_00053]	Debouncing of diagnostic events	[TPS_DEXT_01048] [TPS_DEXT_03004] [TPS_DEXT_03005] [TPS_DEXT_03017]
[RS_DEXT_00054]	Operation cycles	[TPS_DEXT_01086] [TPS_DEXT_01087]
[RS_DEXT_00055]	Aging	[TPS_DEXT_03021]
[RS_DEXT_00056]	Indicator	[TPS_DEXT_03022]
[RS_DEXT_00057]	RequestFileTransfer	[TPS_DEXT_01090]
[RS_DEXT_00058]	Indicate that an ECU supports ODB	[TPS_DEXT_01122]
[RS_DEXT_00059]	Support for different protocols	[TPS_DEXT_01124]
[RS_DEXT_00060]	Function	[TPS_DEXT_01096] [TPS_DEXT_01097] [TPS_DEXT_01098] [TPS_DEXT_01099] [TPS_DEXT_01100] [TPS_DEXT_01101] [TPS_DEXT_01121]
[RS_DEXT_00061]	Relationship between functions and diagnostic events	[TPS_DEXT_01095] [TPS_DEXT_01098] [TPS_DEXT_01099] [TPS_DEXT_01100] [TPS_DEXT_01101]
[RS_DEXT_00062]	Pre-configuration of the Fim when the Dem configuration is not yet available	[TPS_DEXT_01095]
[RS_DEXT_00063]	Relation between functions on Fim level and software-components	[TPS_DEXT_01102]
[RS_DEXT_00064]	Definition of an SPN	[TPS_DEXT_01103] [TPS_DEXT_01106]

[RS_DEXT_00065]	Definition of freeze frames on J1939	[TPS_DEXT_01104] [TPS_DEXT_01105]
[RS_DEXT_00066]	Mapping between a J1939 controller application and a software-component	[TPS_DEXT_01108]
[RS_DEXT_00067]	Definition of J1939 DTC	[TPS_DEXT_01107]
[RS_DEXT_00068]	Definition of a Diagnostic Parameter Identifier	[TPS_DEXT_01092]
[RS_DEXT_00069]	Support for OBD Mode 0x01 (RequestCurrentPowertrain DiagnosticData)	[TPS_DEXT_01125]
[RS_DEXT_00070]	Support for OBD Mode 0x02 (RequestPowertrainFreeze FrameData)	[TPS_DEXT_01126]
[RS_DEXT_00071]	Support for OBD ModeModes 0x03 / 0x07 / 0x0A (Request EmissionRelatedDiagnostic TroubleCodes)	[TPS_DEXT_01127]
[RS_DEXT_00072]	Support for OBD Mode 0x04 (ClearResetEmissionRelated DiagnosticInformation)	[TPS_DEXT_01128]
[RS_DEXT_00073]	Support for OBD Mode 0x06 (RequestOnBoardMonitoring TestResults)	[TPS_DEXT_01129]
[RS_DEXT_00074]	Support for OBD Mode 0x08 (RequestControlOfOnBoard Device)	[TPS_DEXT_01130]
[RS_DEXT_00075]	Support for OBD Mode 0x09 (RequestVehicleInformation)	[TPS_DEXT_01131]
[RS_DEXT_00076]	Definition of Diagnostic Test Identifier	[TPS_DEXT_01132]
[RS_DEXT_00077]	Description of the utilization of UDS for supporting WWH-OBD	[TPS_DEXT_01133]
[RS_DEXT_00078]	Support for In Use Monitor Performance Ratio	[TPS_DEXT_01110]
[RS_DEXT_00079]	Support for environment conditions	[TPS_DEXT_01113] [TPS_DEXT_01114] [TPS_DEXT_01115] [TPS_DEXT_01116] [TPS_DEXT_01117] [TPS_DEXT_01118] [TPS_DEXT_01119] [TPS_DEXT_01120]

Table 1.3: RequirementsTracing

2 Use Cases

2.1 Use cases for diagnostic data exchange

The basic usage of the `DiagnosticExtract` is the exchange of diagnostic data between the different parties involved in the diagnostic development process to allow the configuration of the DCM and the DEM and to provide the description of corresponding application interfaces to implement diagnostic services and fault handling.

2.2 Configuration of DCM

The configuration of the DCM includes the setup of diagnostic services and the assignment of data objects provided by one or more software components (e.g. Composition 1, Composition 2).

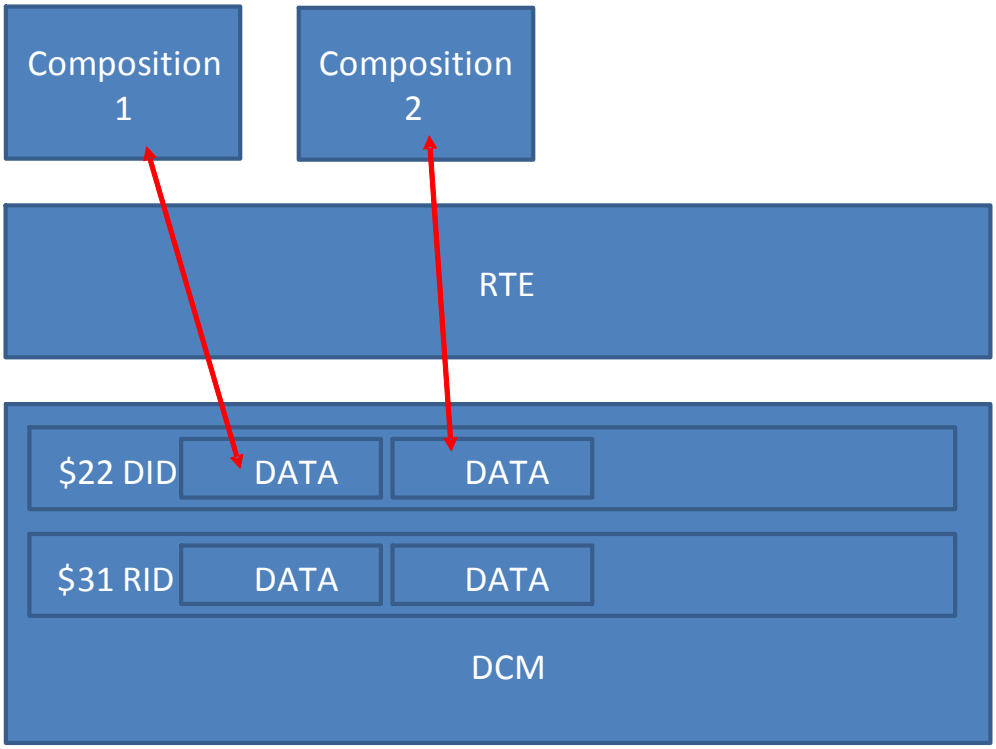

Figure 2.1: Assignment of DCM-related data objects

2.3 Configuration of DEM

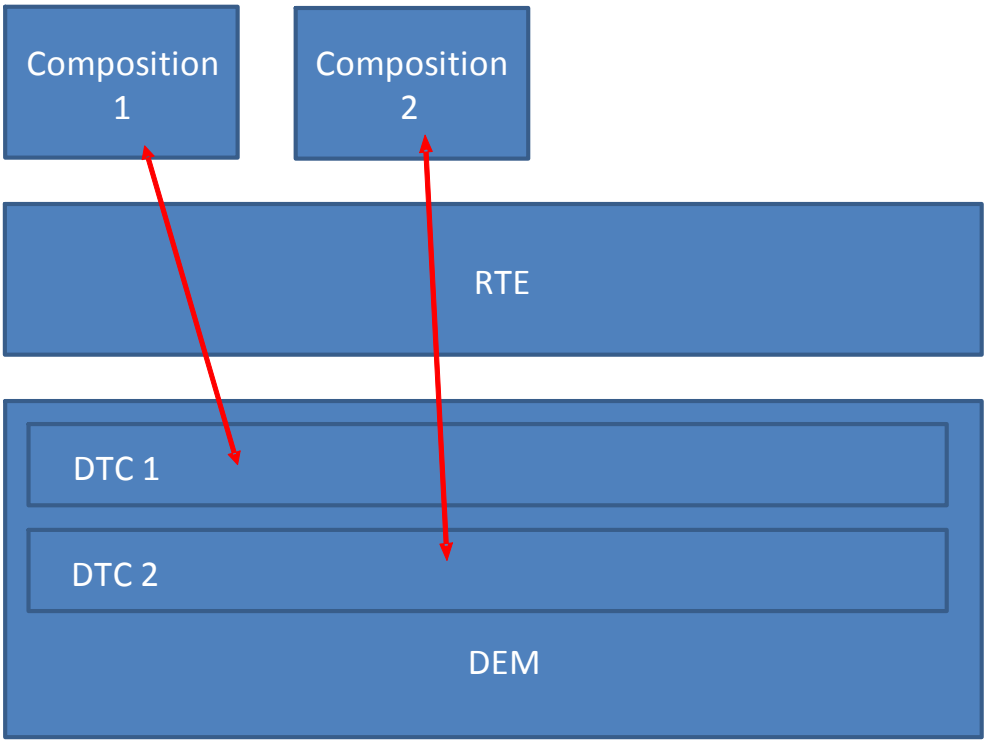

Figure 2.2: Assignment of DEM-related data objects

The configuration of the DEM includes fault memory data (DTCs and environmental data) and the assignment of corresponding data by one or more SwComponentTypes (e.g. Composition 1, Composition 2).

As already explained, the diagnostic development process is distributed among different parties. On the one hand side, the OEM needs to describe the general requirements for a diagnostic system that have to be implemented by an ECU:

1. Depending on the diagnostic system, the OEM can provide a completely or partly-filled DiagnosticExtract including the description of PortInterfaces:
   - Integrator/SWC developer (OEM or Tier 1) is responsible for the completion (detailing of predefined diagnostic content).
   - Integrator/SWC developer (OEM or Tier 1) is responsible for the specific configuration of diagnostic content defined by himself).
   - New integration of updated Diagnostic descriptions by integrator.
2. Return of completely or partly-filled DiagnosticExtract to OEM for:
   - Documentation

- ECU testing
- Integration reviews
- Failure correction

Use case examples:

- Configuration of UDS service 0x22 (`ReadDataByIdentifier`)
- Configuration of UDS service 0x2E (`WriteDataByIdentifier`)
- Configuration of UDS service 0x31 (`RoutineControl`)
- Configuration of UDS service 0x2F (`I/O-Control`)
- Configuration of DEM DTCs
- Configuration of Combined Events
- Mapping of events to DTCs
- Configuration of DTC-related environmental data
- Mapping of DEM Events to their corresponding Enable Conditions and Storage Conditions
- Configure general DCM parameters
- Description of diagnostic demands that are not relevant for code generation but have to be exchanged between OEM and Tier1 (e.g. set and reset condition for a DTC)

Refinement of use cases:

The OEM already provides a System Template for an ECU which describes the ECU Supplier SW parts as a `CompositionSwComponentType` where only inputs and outputs are known.

1. The OEM creates a `DiagnosticExtract` which describes the diagnostic interfaces of an ECU. Supported services are described (e.g. RDBI/WDBI/Routine Control) as well as their corresponding input/output parameters and return values (those which are optional in UDS standard).
2. SWC Developer at OEM or OEM SW Supplier develops SWC and also describes the Diagnostic information using `ServiceNeeds` as diagnostic contribution.
3. The OEM Diagnose Responsible for a Project creates the mappings between the `DiagnosticExtract` and the SWC available on OEM side (from 2.).
4. The OEM Diagnostic Responsible for a Project creates the mappings between the `DiagnosticExtract` and the `CompositionSwComponentType`s which will be implemented by the ECU Supplier or SW Developer.
5. The ECU Suppliers receives the ECU Extract including `DiagnosticExtract` from the OEM and imports it to the project.

6. In the same way is in point 2., the SWC Developer on supplier or Tier 2 side describes the Diagnostic information using `ServiceNeeds` as diagnostic contribution.

In the same way as in point 3: The ECU Supplier Diagnostic Responsible creates the mapping between the `PortInterface`s of the `DiagnosticExtract` (from 5.) and the `SwComponentType`s as provided in 6.

For the usage of indicators, it might happen that indicators defined on BSW level in DEM might not be automatically mapped to the implementation on SWC level. This would then require a manual mapping step by the integrator to resolve the mismatch.

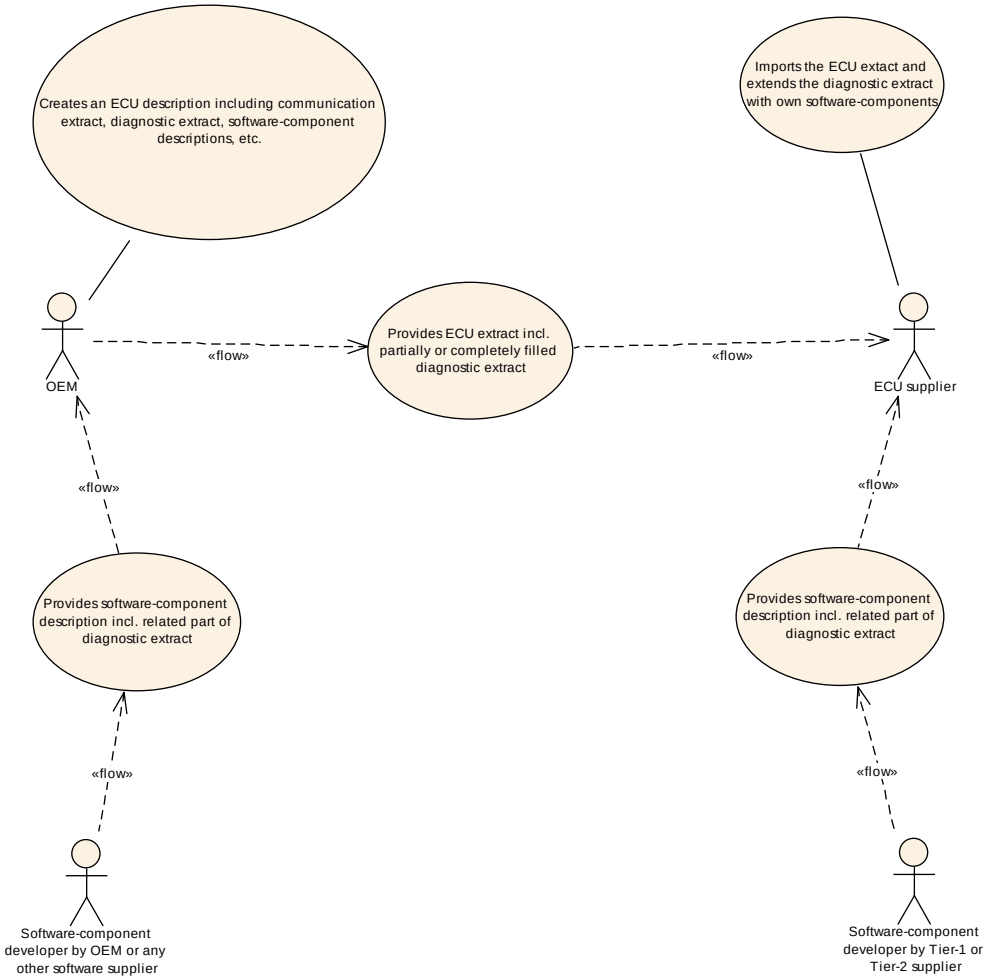

Figure 2.3: Workflow scenario for diagnostic data exchange

2.4 Configuration of the Fim

2.4.1 Model Function Inhibition

A typical use case for the Fim is the definition of “control mechanisms for software components and the functionality therein” [14]. By this means, the Fim can significantly

modify the behavior of the application software at run-time, e.g. in response to a sensor failure.

The conditions for deciding about inhibitions are derived from *diagnostic events*. If a *diagnostic event* that relates to a hardware sensor is reported as "failed" then the Fim can degrade the behavior of the application software to no longer rely on the sensor information that has become unreliable.

Consequently, the DiagnosticExtract needs to provide the basic mechanisms to create the definition of such inhibition rules that relate pieces of the application software (which are visible to the DiagnosticExtract as formally defined SwComponentTypes) to the definition of *diagnostic events*.

2.4.2 Model Fim configuration before Dem exists

The configuration of the Fim module and the Dem module are closely intertwined because the Fim uses the *diagnostic events* defined in the context of the Dem. This means that the Fim can only be configured if the Dem exists as a basis for creating this configuration.

Clearly, this is contradicting the approach of a "decentralized configuration" of the diagnostics stack's behavior. It may happen that the configuration of the Fim and the configuration of the Dem by means of the DiagnosticExtract are distributed over different organizations within a concrete vehicle project.

However, the organizations would not be able to make progress independently from each other because the work product of one is the immediate prerequisite that the other organization can start working.

Therefore one important use case for the DiagnosticExtract is the ability to configure the Fim module in the scope of the DiagnosticExtract **without the immediate need to already have access to the corresponding configuration of the Dem module**.

In other words, a typical work-flow could look like this:

1. Create the Fim configuration on the basis of a collection of alias objects that represent the *diagnostic events* that are defined in the context of the Dem module within the configuration of the Fim and **in parallel** create the Dem configuration and thereby add definitions of the actual *diagnostic events*.
2. Ship the configuration of the Dem to the organization that is responsible for the creation of the configuration of the Fim.
3. Let this organization take care of resolving the alias *diagnostic events* by way of referring from the Fim configuration to the the *diagnostic events* contained in the actual Dem configuration.

2.5 Configuration of J1939 Diagnostics

Within the System Extract, J1939 diagnostic is configured on a service level by the diagnostic messages DM01 .. DM57, which will be handled by J1939 Dcm.

The diagnostic content which is provided by the diagnostic messages is defined within the `DiagnosticExtract` separately for each J1939 function, and contributes to the configuration of the J1939 Dcm as well as the parts of the Dem configuration that are relevant for J1939.

J1939 functions (known as Controller Application or CA within the SAE J1939 Standard [15]) are identified by a NAME and an address (assigned at run-time) and are represented in AUTOSAR by the `J1939NmNode`.

2.5.1 Modeling of J1939 Diagnostic Aspects independent of the Deployment

The main use case for modeling the J1939 in the `DiagnosticExtract` is the possibility to create the diagnostic configuration relevant for a J1939 function, even if the underlying communication architecture is not (fully) defined and if no vehicle model (represented by a `System`) exists.

The J1939 diagnostics modeling is done during functional development before the `System` exists where the functions are deployed. This approach represents a case of decentralized configuration of the diagnostic stack.

2.5.2 J1939 Diagnostic Content modeled in the Diagnostic Extract

The diagnostic content for J1939 consists of the following topics:

SPN The Suspect Parameter Number is used throughout the J1939 specification to identify measured values (physical signals) and commands, the communication signals to which these are mapped, and diagnostic events caused by these.

Signals Communication signals that are relevant for diagnostics are specifically annotated, and reported by the J1939 diagnostics.

DTCs J1939 DTCs consist of an SPN, identifying the source of the diagnostic event, and an FMI (Failure Mode Indicator), identifying the problem with this source, like a boundary exception. In addition, an event counter is contained in the DTC.

Freeze Frames contain measured valued captured at the time a diagnostic event is reported as failed. J1939 supports two different kinds of freeze frames:

- **standardized freeze frames** (reported by DM04), which have a legislated layout
- **expanded freeze frames** (reported by DM24/DM25), which have a configurable layout

3 Conceptual Background

Chapter 1 has already given an overview on the intended way of using the DiagnosticExtract template and files. This chapter gives further background information on the overall concept behind the new format to create a better basis for understanding of the meta-model described in Chapter 4.

3.1 Definition of relevant Diagnostic Elements

[TPS_DEXT_01046] ECU configuration is not suitable to be exchanged between partners in an ECU development project ⌈ The ECU configuration (EcuC) parameters defined by the AUTOSAR Software Specification (SWS) documents for Dem and Dcm are not suitable to be exchanged between partners in an ECU development project.

Besides proprietary ways of using the EcuC format, the main reason for EcuC parameters being inappropriate to be exchanged is their closeness to implementation (e.g. parameter on buffer sizes). ⌋*(RS_DEXT_00001)*

[TPS_DEXT_01047] Differences in the development processes for diagnostics at automotive OEMs and ECU suppliers ⌈ Additionally, there are differences in the development processes for diagnostics at automotive OEMs and ECU suppliers resulting in different views on relevant diagnostic properties to be exchanged and different ways of deriving and defining them as diagnostic requirements. ⌋*(RS_DEXT_00002)*

Therefore, the identification of all diagnostic properties and requirements as superset from the companies' views forms the basis on which the DiagnosticExtract template has been defined.

3.2 Abstraction from EcuC Level

The DiagnosticExtract template does not only focus on relevant diagnostic properties and requirements but also - if required - lift them onto an appropriate abstraction level to make them meaningfully exchangeable (e.g. debouncing requirements that abstract from mapping on a concrete ECU).

However, for many EcuC parameters identified as relevant, an abstraction is not useful or not required and thus, these parameters are mapped 1:1 to equivalent elements of the DiagnosticExtract template.

[TPS_DEXT_01140] Values contained in DiagnosticExtract shall be taken for the derivation of basic software modules ⌈ The values specified in a given DiagnosticExtract shall be taken for the derivation of basic software modules in the diagnostic stack.

This means that parameter values in the Ecuc are created that could be subject to constraints, e.g. a valid interval. ⌋()

Therefore, model elements in the DiagnosticExtract shall be carefully checked against the constraints formulated in EcuC. However, this document does not contain specific constraints on a detailed basis.

3.3 Independence of Definition

With respect to development processes, the DiagnosticExtract format also enables more independence when defining requirements on diagnostic functionality than possible with EcuC parameters. The approach of “decentralized configuration” is met in the DiagnosticExtract template in mainly two ways described in the following sub-chapters.

3.3.1 Use of ≪atpSplitable≫ enabling separation of elements over several physical files

Most elements of the DiagnosticExtract template can be split over several physical files. Therefore, parts of these elements (e.g. certain attributes) can be defined by, for example, an OEM and other parts of these elements by, for example, an ECU supplier.

3.3.2 Use of self-contained mapping elements

Many diagnostic requirements are established by mappings between diagnostic elements (e.g., DTC to DemEvent mapping). However, the “decentralized configuration” approach requires that these mappings can be flexibly defined at almost any time within the ECU development process and by any of the involved companies respectively roles.

Therefore, the DiagnosticExtract template defines self-contained mapping elements that have references to two (or potentially more) diagnostic elements to define a mapping.

The self-contained mapping elements can be created any time after the diagnostic elements to be mapped together have been defined. Alternatively, a mapping element can be created after only one diagnostic element has been defined indicating the need to make the mapping complete by defining the additional diagnostic element(s) to map to.

4 Common Meta Model Elements

4.1 Introduction

This chapter contains a description of the meta-model for the specification of the DiagnosticExtract in AUTOSAR. The goal of the specification of the DiagnosticExtract is to contribute to the description of the configuration of the Diagnostic Communication Manager [10] (Dcm) and the Diagnostic Event Manager [11] (Dem)

The meta-model can be roughly divided into five sections

- A common section that contains meta-classes shared between the description of the diagnostic services (that roughly corresponds to the Dcm) and the diagnostic event handling (that roughly corresponds to the Dem), see section 4.2.
- A section dedicated to the configuration of the diagnostic services, see section 5.
- A section dedicated to the configuration of the diagnostic event handling, see section 6.
- A section dedicated to the configuration of the functional inhibition from the diagnostics point of view, see section 7.
- A section dedicated to the configuration of the diagnostics on J1939, see section 8.

4.2 Data Identifier vs. Routine vs. Data Element

This chapter highlights the formal modeling of some of the central parts of AUTOSAR diagnostics when it comes to configuration. There are some concepts widely used that need to be reflected in the meta-model.

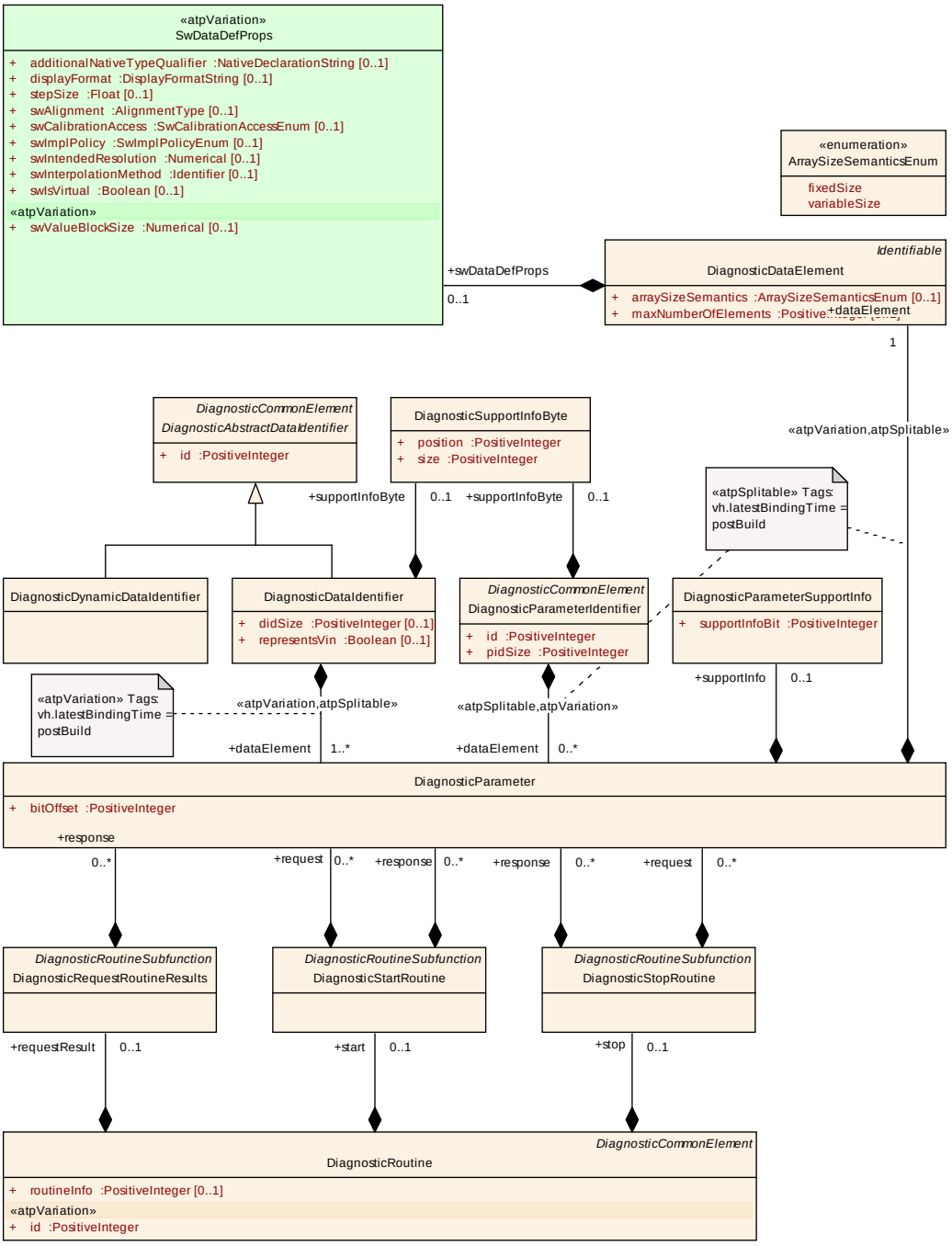

Figure 4.1: Common Diagnostic elements

Class	**DiagnosticCommonElement (abstract)**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents a common base class for all diagnostic elements. It does not contribute any specific functionality other than the ability to become the target of a reference.			
Base	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Attribute	Type	Mul.	Kind	Note

Table 4.1: DiagnosticCommonElement

Class	DiagnosticParameter			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents the ability to describe information relevant for the execution of a specific diagnostic service, i.e. it can be taken to parameterize the service.			
Base	ARObject			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
bitOffset	PositiveInteger	1	attr	This represents the bitOffset of the DiagnosticParameter
dataElement	DiagnosticDataElement	1	aggr	This represents the related dataElement of the DiagnosticParameter **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
supportInfo	DiagnosticParameterSupportInfo	0..1	aggr	This attribute represents the ability to define which bit of the support info byte is representing this part of the PID.

Table 4.2: DiagnosticParameter

The purpose of the `DiagnosticCommonElement` is to provide a common reference target for all kinds of diagnostic elements. This aspect is explained in more detail in section 4.4.

The purpose of a `Data Identifier` (DID) is to associate a unique numerical value to a piece of data related to diagnostics. From the modeling point of view, this means that the modeling of the `Data Identifier` needs to provide an attribute that represents the numeric value as well as a relation to a `Data Element` representing a set of diagnostic piece of data.

[TPS_DEXT_01000] AUTOSAR diagnostics supports two kinds of data identifiers
⌈ In AUTOSAR, two kinds of data identifiers are supported:

- The `DiagnosticDataIdentifier` inherits from `DiagnosticAbstractDataIdentifier` and is used to define data identifiers fully known at configuration time. A `DiagnosticDataIdentifier` shall have at least 1 `dataElement`.
- The `DiagnosticDynamicDataIdentifier` inherits from `DiagnosticAbstractDataIdentifier` and is used to define data identifiers fully known **only** at run time. Consequently, there is **no formal means** to define `dataElement` at configuration time.

⌋*(RS_DEXT_00034, RS_DEXT_00035)*

Class	**DiagnosticDataIdentifier**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents the ability to model a diagnostic data identifier (DID) that is fully specified regarding the payload at configuration-time. **Tags:** atp.recommendedPackage=DiagnosticDataIdentifiers			
Base	ARElement, ARObject, CollectableElement, DiagnosticAbstractDataIdentifier, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataElement	DiagnosticParameter	1..*	aggr	This is the dataElement associated with the DiagnosticDataIdentifier. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dataElement, variationPoint.shortLabel vh.latestBindingTime=postBuild
didSize	PositiveInteger	0..1	attr	This attribute indicates the size of the DiagnosticDataIdentifier.
representsVin	Boolean	0..1	attr	This attributes indicates whether the specific DiagnosticDataIdentifier represents the vehicle identification.
supportInfoByte	DiagnosticSupportInfoByte	0..1	aggr	This attribute represents the supported information associated with the DiagnosticDataIdentifier.

Table 4.3: DiagnosticDataIdentifier

Class	**DiagnosticDynamicDataIdentifier**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents the ability to define a diagnostic data identifier (DID) at run-time. **Tags:** atp.recommendedPackage=DiagnosticDataIdentifiers			
Base	ARElement, ARObject, CollectableElement, DiagnosticAbstractDataIdentifier, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 4.4: DiagnosticDynamicDataIdentifier

Class	**DiagnosticAbstractDataIdentifier (abstract)**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents an abstract base class for the modeling of a diagnostic data identifier (DID).			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

Attribute	Type	Mul.	Kind	Note
id	PositiveInteger	1	attr	This is the numerical identifier used to identify the DiagnosticAbstractDataIdentifier in the scope of diagnostic workflow

Table 4.5: DiagnosticAbstractDataIdentifier

[TPS_DEXT_01072] Purpose of attribute DiagnosticDataIdentifier.representsVin ⌈ There is a use case for identifying a specific DiagnosticDataIdentifier that carries the so-called *vehicle identification number* (VIN).

It is therefore important to be able to formally indicate this characteristic. For this purpose the attribute DiagnosticDataIdentifier.representsVin is available. ⌋*(RS_DEXT_00034)*

[constr_1324] Existence of attribute DiagnosticDataIdentifier.representsVin ⌈ Within the context of a given DiagnosticContributionSet, the attribute DiagnosticDataIdentifier.representsVin shall have the value `true` for only a single DiagnosticDataIdentifier. ⌋*()*

Please note that the VIN is only relevant in the context of *diagnostics over IP* (DoIP). However, there is no constraint that bounds the validity of [constr_1324] to the existence of a DiagnosticConnection that is build on top of an IP stack.

If the attribute exists and there is no IP used then the meaning of the attribute is simply irrelevant. Anyway, this situation should not be attributed to a misconfiguration.

The concept of the Data Element represents a piece of information decomposed from the data identified by a DID and exchanged between the DEM and, for example, a tester.

The nature of such a Data Element could be compared to the nature of an ISignal[1] and therefore the modeling of a Data Element by means of the meta-class DiagnosticDataElement aggregates SwDataDefProps in the role swDataDefProps in order to provide a reference to SwBaseType.

The aggregation of SwDataDefProps can also be used to refer to a DataConstr in order to specify a valid data range for the DiagnosticDataElement.

Class	DiagnosticDataElement			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents the ability to describe a concrete piece of data to be taken into account for diagnostic purposes.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
arraySizeSemantics	ArraySizeSemanticsEnum	0..1	attr	This attribute controls the meaning of the value of the array size.

[1] which represents the payload in “regular” bus communication

Attribute	Type	Mul.	Kind	Note
maxNumberOfElements	PositiveInteger	0..1	attr	The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
swDataDefProps	SwDataDefProps	0..1	aggr	This property allows to specify data definition properties in order to support the definition of e.g. computation formulae and data constraints.

Table 4.6: DiagnosticDataElement

[constr_1394] Value of `DiagnosticDataElement.maxNumberOfElements` depending on its existence ⌈ If the attribute `DiagnosticDataElement.maxNumberOfElements` exists then its value shall be greater than 0. ⌋*()*

[TPS_DEXT_01134] Definition of a `DiagnosticDataElement` used in the context of a DID obtained by sender-receiver communication ⌈ If the `DiagnosticDataElement` is aggregated by a `DiagnosticParameter` that in turn is aggregated by a subclass of `DiagnosticAbstractDataIdentifier` and the `DiagnosticDataElement` is also referenced by a `DiagnosticServiceDataMapping` then the referenced `DiagnosticDataElement` (by way of `SwDataDefProps`) shall refer to a `SwBaseType` with attribute `baseTypeSize` set to **either** 8, 16, or 32 and attribute `baseTypeEncoding` set to **either** `NONE` or `2C`.

In this case it is possible to define the `DiagnosticDataElement` either as a scalar or as an array (see [TPS_DEXT_01001], [TPS_DEXT_01002]). ⌋*(RS_DEXT_00034)*

[TPS_DEXT_01135] Definition of a `DiagnosticDataElement` used in the context of a DID obtained by client/server communication ⌈ If the `DiagnosticDataElement` is aggregated by a `DiagnosticParameter` that in turn is aggregated by a subclass of `DiagnosticAbstractDataIdentifier` and the `DiagnosticDataElement` is also referenced by a `DiagnosticServiceSwMapping` then the referenced `DiagnosticDataElement` (by way of `SwDataDefProps`) shall refer to a `SwBaseType` with attribute `baseTypeSize` set to 8 and attribute `baseTypeEncoding` set to `NONE`.

In this case it is only possible to define the `DiagnosticDataElement` as an array (see [TPS_DEXT_01001], [TPS_DEXT_01002]). The ability to define a Variable-Size Array shall only be used for the last element of the DID. ⌋*(RS_DEXT_00034)*

[TPS_DEXT_01136] Definition of a `DiagnosticDataElement` used in the context of a diagnostic routine ⌈ If the `DiagnosticDataElement` is aggregated by a `DiagnosticParameter` that in turn is aggregated by either a `DiagnosticStartRoutine`, `DiagnosticStopRoutine`, or `DiagnosticRequestRoutineResults` then the `DiagnosticDataElement` (by way of `SwDataDefProps`) shall refer to a `SwBaseType` with attribute `baseTypeSize` to **either** 8, 16, or 32 and attribute `baseTypeEncoding` set to **either** `NONE` or `2C`.

In this case it is possible to define the `DiagnosticDataElement` either as a scalar or as an array (see [TPS_DEXT_01001], [TPS_DEXT_01002]). The ability to define a

Variable-Size Array shall only be used for the last argument to the diagnostic routine. ⌋ *(RS_DEXT_00034)*

[constr_1470] Value of `DiagnosticParameter.bitOffset` ⌈ The value of `DiagnosticParameter.bitOffset` shall only be set to a multiple of 8. ⌋*()*

Enumeration	**ArraySizeSemanticsEnum**
Package	M2::AUTOSARTemplates::CommonStructure::ImplementationDataTypes
Note	This type controls how the information about the number of elements in an ApplicationArrayDataType is to be interpreted.
Literal	***Description***
fixedSize	This means that the ApplicationArrayDataType will always have a fixed number of elements. **Tags:** atp.EnumerationValue=0
variableSize	This implies that the actual number of elements in the ApplicationArrayDataType might vary at run-time. The value of arraySize represents the maximum number of elements in the array. **Tags:** atp.EnumerationValue=1

Table 4.7: ArraySizeSemanticsEnum

[TPS_DEXT_01137] Applicability of `DiagnosticDataIdentifier.didSize` ⌈ The attribute `DiagnosticDataIdentifier.didSize` may exist if the value of `DiagnosticDataIdentifier.id` is **in the range 0xF400-0xF4FF**.

A typical case for the existence of `DiagnosticDataIdentifier.didSize` is the understanding that the `DiagnosticDataIdentifier` is relevant for OBD and the `DiagnosticDataIdentifier.id` is in the designated range. ⌋*(RS_DEXT_00034)*

[constr_1471] Existence of `DiagnosticDataIdentifier.didSize` ⌈ The attribute `DiagnosticDataIdentifier.didSize` **shall not exist** if the value of `DiagnosticDataIdentifier.id` is **outside the range 0xF400-0xF4FF**. ⌋*()*

[TPS_DEXT_01138] Applicability of `DiagnosticDataIdentifier.supportInfoByte` ⌈ The attribute `DiagnosticDataIdentifier.supportInfoByte` may exist if the value of `DiagnosticDataIdentifier.id` is **in the range 0xF400-0xF4FF**.

A typical case for the existence of `DiagnosticDataIdentifier.supportInfoByte` is the understanding that the `DiagnosticDataIdentifier` is relevant for OBD and the `DiagnosticDataIdentifier.id` is in the designated range. ⌋ *(RS_DEXT_00034)*

[constr_1472] Existence of `DiagnosticDataIdentifier.supportInfoByte` ⌈ The attribute `DiagnosticDataIdentifier.supportInfoByte` **shall not exist** if the value of `DiagnosticDataIdentifier.id` is **outside the range 0xF400-0xF4FF**. ⌋*()*

4.2.1 Usage of SwDataDefProps

Please note that the definitions of properties like computation methods[2], limits[3], or units[4] of diagnostic data elements is based on shared resources of the AUTOSAR meta-model, namely by aggregation of meta-class SwDataDefProps.

This meta-class contributes a wealth of possible properties related to the definition of data in general and, in this case, diagnostics in particular.

However, it is important to understand that SwDataDefProps is so expressive and versatile that its applicability needs to be constrained (in this specific case, see [constr_1325]) for specific deployments according to the requirements that originate from the semantics of the piece of data that is decorated by SwDataDefProps.

[constr_1325] Allowed attributes of SwDataDefProps for DiagnosticDataElement.swDataDefProps ⌈ The allowed attributes of SwDataDefProps for the aggregation in the role DiagnosticDataElement.swDataDefProps are defined in table 4.8. ⌋()

Attributes of SwDataDefProps	DiagnosticDataElement.swDataDefProps
additionalNativeTypeQualifier	N/A
annotation	N/A
baseType.baseTypeDefinition.baseTypeEncoding	D
baseType.baseTypeDefinition.baseTypeSize	D
baseType.baseTypeDefinition.byteOrder	D
baseType.baseTypeDefinition.memAlignment	N/A
baseType.baseTypeDefinition.nativeDeclaration	N/A
compuMethod	D
dataConstr	D
displayFormat	D
implementationDataType	N/A
invalidValue	N/A
swAddrMethod	N/A
swAlignment	N/A
swBitRepresentation	N/A
swCalibrationAccess	N/A
swCalprmAxisSet	N/A
swComparisonVariable	N/A
swDataDependency	N/A
swImplPolicy	N/A
swIntendedResolution	N/A
swInterpolationMethod	N/A
swIsVirtual	N/A
swPointerTargetProps	N/A
swRecordLayout	N/A
swRefreshTiming	N/A
swTextProps	N/A
swValueBlockSize	N/A

[2]formalized as CompuMethod in AUTOSAR
[3]formalized as DataConstr in AUTOSAR
[4]formalized as Unit in AUTOSAR

Attributes of SwDataDefProps	DiagnosticDataElement.swDataDefProps
unit	D
valueAxisDataType	N/A

Table 4.8: Allowed attributes of SwDataDefProps for DiagnosticDataElement.swDataDefProps

The following legend applies to table 4.8:

Abbr.	Description
D	**Define** the attribute independent from settings to the left.
I	**Inherit** the definition from the left for usage in the scope of this element.
N/A	Attribute is **not applicable** for usage in the scope of this element.
M	Attribute is **meaningless** in the scope of this element. As it was allowed in previous versions, declaring it as Not Applicable (NA) would break compatibility. Tools shall ignore such an attribute without a warning.

Table 4.9: Legend of table "Allowed attributes of SwDataDefProps for DiagnosticDataElement.swDataDefProps"

Please note that, in comparison to similar tables appearing in other AUTOSAR documents (e.g. [8]), table 4.8 intentionally goes into more detail regarding the applicability of the attributes of SwBaseType. This is in contrast to similar tables contained in, e.g. the specification of the Software-Component Template [8]

The attributes of SwBaseType are considered of **paramount importance for the definition of the semantics** of the enclosing DiagnosticDataElement and thus the emphasis is justified.

Class	≪`atpVariation`≫ **SwDataDefProps**			
Package	M2::MSR::DataDictionary::DataDefProperties			
Note	This class is a collection of properties relevant for data objects under various aspects. One could consider this class as a "pattern of inheritance by aggregation". The properties can be applied to all objects of all classes in which SwDataDefProps is aggregated. Note that not all of the attributes or associated elements are useful all of the time. Hence, the process definition (e.g. expressed with an OCL or a Document Control Instance MSR-DCI) has the task of implementing limitations. SwDataDefProps covers various aspects: • Structure of the data element for calibration use cases: is it a single value, a curve, or a map, but also the recordLayouts which specify how such elements are mapped/converted to the DataTypes in the programming language (or in AUTOSAR). This is mainly expressed by properties like swRecordLayout and swCalprmAxisSet • Implementation aspects, mainly expressed by swImplPolicy, swVariableAccessImplPolicy, swAddrMethod, swPointerTagetProps, baseType, implementationDataType and additionalNativeTypeQualifier • Access policy for the MCD system, mainly expressed by swCalibrationAccess • Semantics of the data element, mainly expressed by compuMethod and/or unit, dataConstr, invalidValue • Code generation policy provided by swRecordLayout **Tags:** vh.latestBindingTime=codeGenerationTime			
Base	ARObject			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
additionalNativeTypeQualifier	NativeDeclarationString	0..1	attr	This attribute is used to declare native qualifiers of the programming language which can neither be deduced from the baseType (e.g. because the data object describes a pointer) nor from other more abstract attributes. Examples are qualifiers like "volatile", "strict" or "enum" of the C-language. All such declarations have to be put into one string. **Tags:** xml.sequenceOffset=235
annotation	Annotation	*	aggr	This aggregation allows to add annotations (yellow pads ...) related to the current data object. **Tags:** xml.roleElement=true; xml.roleWrapperElement=true; xml.sequenceOffset=20; xml.typeElement=false; xml.typeWrapperElement=false
baseType	SwBaseType	0..1	ref	Base type associated with the containing data object. **Tags:** xml.sequenceOffset=50

Attribute	Type	Mul.	Kind	Note
compuMethod	CompuMethod	0..1	ref	Computation method associated with the semantics of this data object. **Tags:** xml.sequenceOffset=180
dataConstr	DataConstr	0..1	ref	Data constraint for this data object. **Tags:** xml.sequenceOffset=190
displayFormat	DisplayFormatString	0..1	attr	This property describes how a number is to be rendered e.g. in documents or in a measurement and calibration system. **Tags:** xml.sequenceOffset=210
implementationDataType	ImplementationDataType	0..1	ref	This association denotes the ImplementationDataType of a data declaration via its aggregated SwDataDefProps. It is used whenever a data declaration is not directly referring to a base type. Especially • redefinition of an ImplementationDataType via a "typedef" to another ImplementationDatatype • the target type of a pointer (see SwPointerTargetProps), if it does not refer to a base type directly • the data type of an array or record element within an ImplementationDataType, if it does not refer to a base type directly • the data type of an SwServiceArg, if it does not refer to a base type directly **Tags:** xml.sequenceOffset=215
invalidValue	ValueSpecification	0..1	aggr	Optional value to express invalidity of the actual data element. **Tags:** xml.sequenceOffset=255
stepSize	Float	0..1	attr	This attribute can be used to define a value which is added to or subtracted from the value of a DataPrototype when using up/down keys while calibrating.
swAddrMethod	SwAddrMethod	0..1	ref	Addressing method related to this data object. Via an association to the same SwAddrMethod it can be specified that several DataPrototypes shall be located in the same memory without already specifying the memory section itself. **Tags:** xml.sequenceOffset=30

Attribute	*Type*	*Mul.*	*Kind*	*Note*
swAlignment	AlignmentType	0..1	attr	The attribute describes the intended alignment of the DataPrototype. If the attribute is not defined the alignment is determined by the swBaseType size and the memoryAllocationKeywordPolicy of the referenced SwAddrMethod. **Tags:** xml.sequenceOffset=33
swBitRepresentation	SwBitRepresentation	0..1	aggr	Description of the binary representation in case of a bit variable. **Tags:** xml.sequenceOffset=60
swCalibrationAccess	SwCalibrationAccessEnum	0..1	attr	Specifies the read or write access by MCD tools for this data object. **Tags:** xml.sequenceOffset=70
swCalprmAxisSet	SwCalprmAxisSet	0..1	aggr	This specifies the properties of the axes in case of a curve or map etc. This is mainly applicable to calibration parameters. **Tags:** xml.sequenceOffset=90
swComparisonVariable	SwVariableRefProxy	*	aggr	Variables used for comparison in an MCD process. **Tags:** xml.sequenceOffset=170; xml.typeElement=false
swDataDependency	SwDataDependency	0..1	aggr	Describes how the value of the data object has to be calculated from the value of another data object (by the MCD system). **Tags:** xml.sequenceOffset=200
swHostVariable	SwVariableRefProxy	0..1	aggr	Contains a reference to a variable which serves as a host-variable for a bit variable. Only applicable to bit objects. **Tags:** xml.sequenceOffset=220; xml.typeElement=false
swImplPolicy	SwImplPolicyEnum	0..1	attr	Implementation policy for this data object. **Tags:** xml.sequenceOffset=230

Attribute	Type	Mul.	Kind	Note
swIntendedResolution	Numerical	0..1	attr	The purpose of this element is to describe the requested quantization of data objects early on in the design process. The resolution ultimately occurs via the conversion formula present (compuMethod), which specifies the transition from the physical world to the standardized world (and vice-versa) (here, "the slope per bit" is present implicitly in the conversion formula). In the case of a development phase without a fixed conversion formula, a pre-specification can occur through swIntendedResolution. The resolution is specified in the physical domain according to the property "unit". **Tags:** xml.sequenceOffset=240
swInterpolationMethod	Identifier	0..1	attr	This is a keyword identifying the mathematical method to be applied for interpolation. The keyword needs to be related to the interpolation routine which needs to be invoked. **Tags:** xml.sequenceOffset=250
swIsVirtual	Boolean	0..1	attr	This element distinguishes virtual objects. Virtual objects do not appear in the memory, their derivation is much more dependent on other objects and hence they shall have a swDataDependency . **Tags:** xml.sequenceOffset=260
swPointerTargetProps	SwPointerTargetProps	0..1	aggr	Specifies that the containing data object is a pointer to another data object. **Tags:** xml.sequenceOffset=280
swRecordLayout	SwRecordLayout	0..1	ref	Record layout for this data object. **Tags:** xml.sequenceOffset=290
swRefreshTiming	MultidimensionalTime	0..1	aggr	This element specifies the frequency in which the object involved shall be or is called or calculated. This timing can be collected from the task in which write access processes to the variable run. But this cannot be done by the MCD system. So this attribute can be used in an early phase to express the desired refresh timing and later on to specify the real refresh timing. **Tags:** xml.sequenceOffset=300

Attribute	Type	Mul.	Kind	Note
swTextProps	SwTextProps	0..1	aggr	the specific properties if the data object is a text object. **Tags:** xml.sequenceOffset=120
swValueBlockSize	Numerical	0..1	attr	This represents the size of a Value Block **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime xml.sequenceOffset=80
unit	Unit	0..1	ref	Physical unit associated with the semantics of this data object. This attribute applies if no compuMethod is specified. If both units (this as well as via compuMethod) are specified the units shall be compatible. **Tags:** xml.sequenceOffset=350
valueAxisDataType	ApplicationPrimitiveDataType	0..1	ref	The referenced ApplicationPrimitiveDataType represents the primitive data type of the value axis within a compound primitive (e.g. curve, map). It supersedes CompuMethod, Unit, and BaseType. **Tags:** xml.sequenceOffset=355

Table 4.10: SwDataDefProps

4.2.2 Definition of Arrays

There are several use cases for a `DiagnosticDataElement` that actually represents an array of information. In some cases the array size is static and will not change at run-time, and in some cases the array size needs to change at run-time to fulfill the intended purpose.

[TPS_DEXT_01001] Definition of a fixed-sized array ⌈ A `DiagnosticDataElement` shall be interpreted as a **fixed-size array** if **all** of the following conditions apply:

1. The attribute `DiagnosticDataElement.maxNumberOfElements` exists.
2. The value of the attribute `DiagnosticDataElement.maxNumberOfElements` is set to a value > 0.
3. The value of `DiagnosticDataElement.arraySizeSemantics` either does not exist or is set to `ArraySizeSemanticsEnum.fixedSize`.

⌋*(RS_DEXT_00034, RS_DEXT_00038)*

[TPS_DEXT_01002] Definition of a variable-sized array ⌈ A `DiagnosticDataElement` shall be interpreted as a **variable-size array** if **all** of the following conditions apply:

1. The attribute `DiagnosticDataElement.maxNumberOfElements` exists.

2. The value of the attribute DiagnosticDataElement.maxNumberOfElements is set to a value > 0.
3. The value of DiagnosticDataElement.arraySizeSemantics is set to ArraySizeSemanticsEnum.variableSize.

The value of DiagnosticDataElement.maxNumberOfElements shall be considered the maximum array size in terms of the number of elements. ⌋(*RS_DEXT_00034, RS_DEXT_00038*)

[constr_1326] Existence of a variable-sized array ⌈ The value of the attribute DiagnosticDataElement.arraySizeSemantics **shall not** be set to ArraySizeSemanticsEnum.variableSize if the respective DiagnosticDataElement is referenced from a DiagnosticServiceDataMapping. ⌋()

4.2.3 Definition of textual Strings

DiagnosticDataElement can be used to model a **textual string** that shall be send to or received from the ECU by the tester.

In this case it will be necessary to define the **intended encoding** such that the part of the software on the ECU that produces or consumes of the string on the ECU can be checked (after the establishment of a DiagnosticMapping) for a matching encoding specification.

The encoding in the scope of the DiagnosticDataElement can be defined using the attribute DiagnosticDataElement.swDataDefProps.baseType.baseTypeDefinition.baseTypeEncoding.

4.3 Textual Documentation

A Data Identifier also usually comes with some textual description that explains the meaning of the Data Identifier in short form. This ability is available via the inheritance from Identifiable, in particular by means of the attributes desc and/or introduction (see Figure 4.2).

This also means that the ability to add some form of textual description is widely usable in the scope of the DiagnosticExtract. Many meta-classes are derived from e.g. DiagnosticCommonElement (which inherits from Identifiable) or directly from Identifiable and therefore qualify for the described form of documentation.

In other words, the technology described in Figures 4.2 and 4.3 is not limited to DiagnosticDataElement but has a much wider applicability in the context of the DiagnosticExtract.

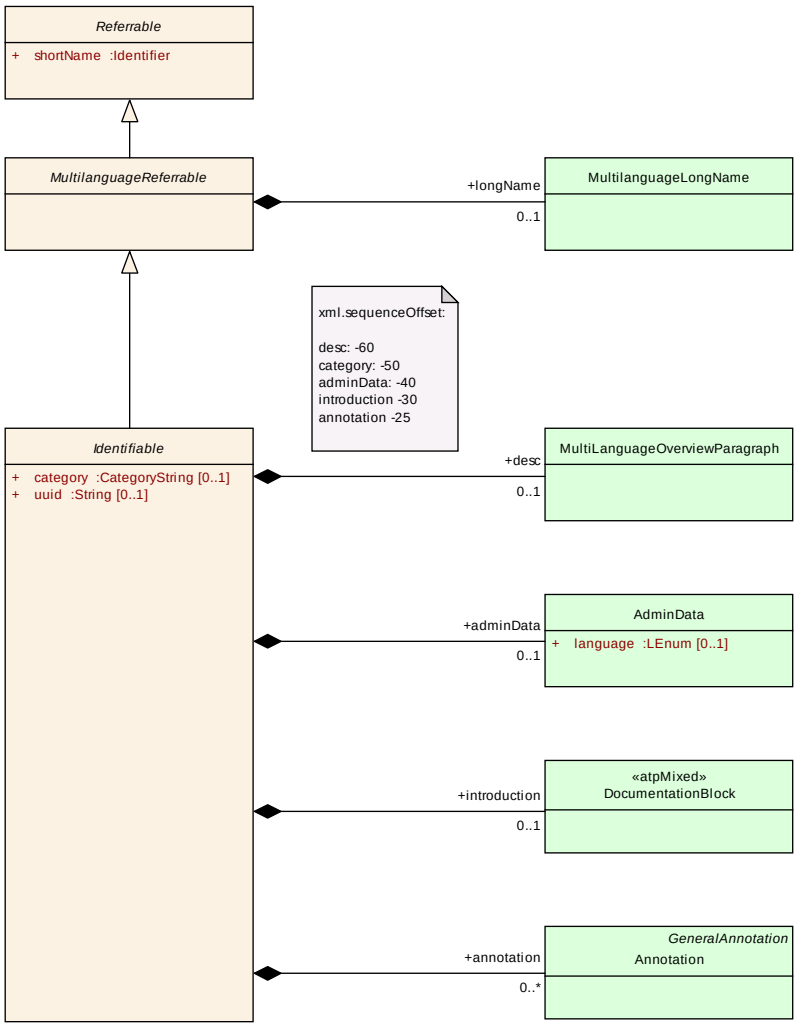

Figure 4.2: Definition of a textual description by means of `desc` and `introduction`

The details regarding the specification of textual content that goes along a given diagnostics element is detailed in Figure 4.3. In fact, `DocumentationBlock` provides a very sophisticated ability to define structured text that may consist e.g. of multiple paragraphs (formalized by meta-class `MultiLanguageOverviewParagraph` aggregated in the role `p`).

In addition to the ability to attach structured text, it is also possible to use the `annotation` (see Figure 4.2) to add short annotations (comparable to the usage of sticky notes) to diagnostic elements.

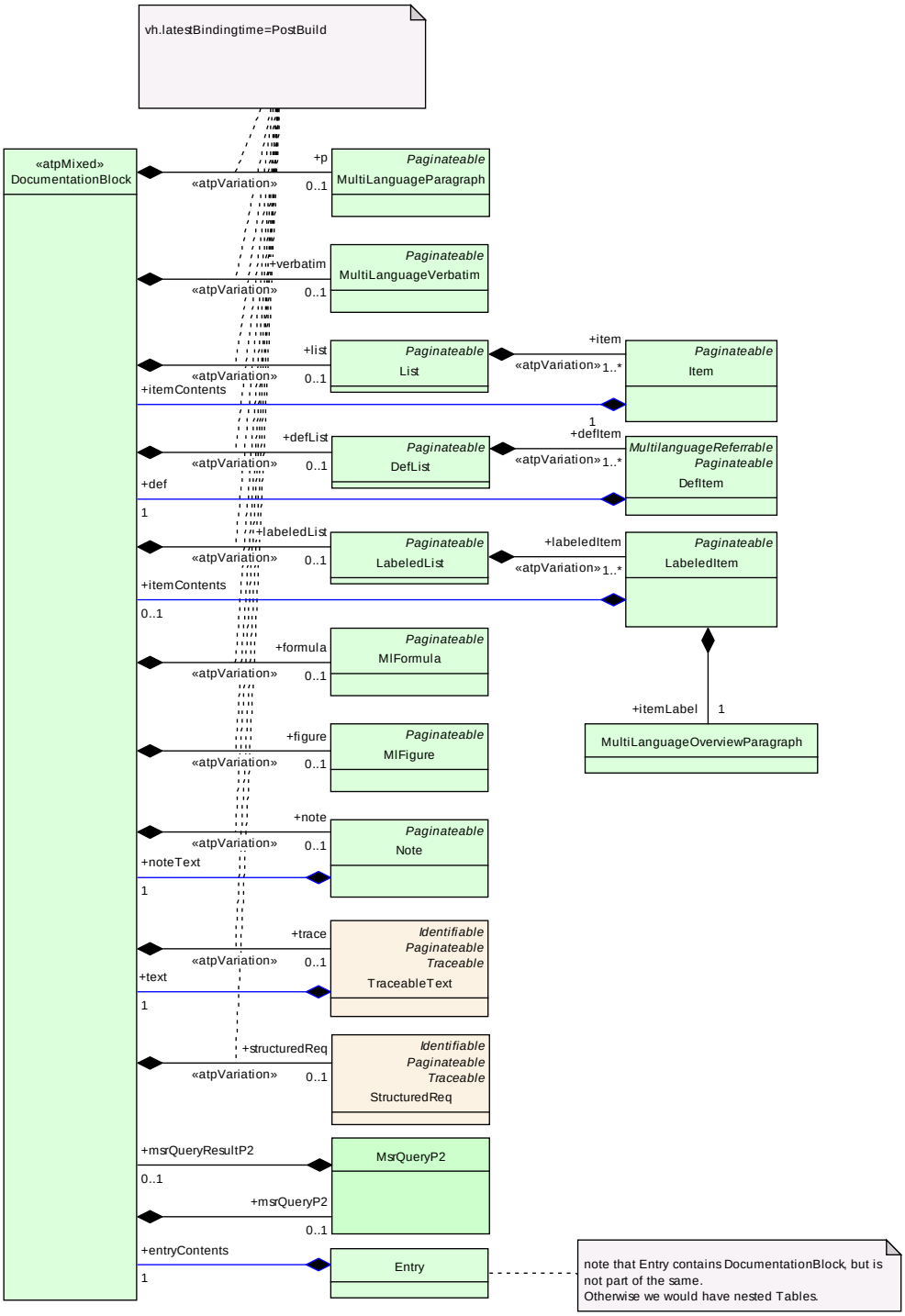

Figure 4.3: Details of the modeling of `DocumentationBlock`

Class	≪`atpMixed`≫ **DocumentationBlock**			
Package	M2::MSR::Documentation::BlockElements			
Note	This class represents a documentation block. It is made of basic text structure elements which can be displayed in a table cell.			
Base	ARObject			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

Attribute	*Type*	*Mul.*	*Kind*	*Note*
defList	DefList	0..1	aggr	This represents a definition list in the documentation block. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=40
figure	MIFigure	0..1	aggr	This represents a figure in the documentation block. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=70
formula	MIFormula	0..1	aggr	This is a formula in the definition block. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=60
labeledList	LabeledList	0..1	aggr	This represents a labeled list. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=50
list	List	0..1	aggr	This represents numbered or unnumbered list. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=30
msrQuery P2	MsrQueryP2	0..1	aggr	This represents automatically contributed contents provided by an msrquery in the context of DocumentationBlock.
note	Note	0..1	aggr	This represents a note in the text flow. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=80
p	MultiLanguageP aragraph	0..1	aggr	This is one particular paragraph. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=10
structured Req	StructuredReq	0..1	aggr	This aggregation supports structured requirements embedded in a documentation block. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=100

Attribute	Type	Mul.	Kind	Note
trace	TraceableText	0..1	aggr	This represents traceable text in the documentation block. This allows to specify requirements/constraints in any documentation block. The kind of the trace is specified in the category. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=90
verbatim	MultiLanguageVerbatim	0..1	aggr	This represents one particular verbatim text. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild xml.sequenceOffset=20

Table 4.11: DocumentationBlock

Class	MultiLanguageOverviewParagraph			
Package	M2::MSR::Documentation::TextModel::MultilanguageData			
Note	This is the content of a multilingual paragraph in an overview item.			
Base	ARObject			
Attribute	Type	Mul.	Kind	Note
l2	LOverviewParagraph	1..*	aggr	This represents the text in one particular language. **Tags:** xml.roleElement=true; xml.roleWrapperElement=false; xml.sequenceOffset=20; xml.typeElement=false; xml.typeWrapperElement=false

Table 4.12: MultiLanguageOverviewParagraph

4.4 Diagnostic Contribution

[TPS_DEXT_01003] `DiagnosticContributionSet` is the central part of the `DiagnosticExtract` ⌈ The central part of formalization of the concept of the `DiagnosticExtract` is the `DiagnosticContributionSet`.

To some extent, it fulfills a similar role as the `System` [6] in the communication domain. ⌋*(RS_DEXT_00001, RS_DEXT_00002)*

The `DiagnosticContributionSet` maintains references to `DiagnosticCommonElement` and by this means the actual definition of the extent of diagnostic contribution takes place.

In other words, the actual extent of a given contribution is created by the collection of `DiagnosticCommonElement`s referenced by the `DiagnosticContributionSet`.

[TPS_DEXT_01004] DiagnosticContributionSet defines the scope of the DiagnosticExtract ⌈ The DiagnosticContributionSet has the ability to define the scope of the given DiagnosticExtract. This means that the DiagnosticContributionSet represents the DiagnosticExtract for the rest of the AUTOSAR model.

The scope may vary between the scope of an entire system down to the contribution of a specific tier-1 supplier to a much bigger context. ⌋*(RS_DEXT_00001, RS_DEXT_00002)*

[TPS_DEXT_01055] Standardized values of DiagnosticContributionSet.category ⌈ The scope of the DiagnosticContributionSet, on the other hand is determined by the value of its category. The following values are predefined by AUTOSAR:

- DIAGNOSTICS_ABSTRACT_SYSTEM_DESCRIPTION: this DiagnosticContributionSet represents a more or less high-level definition that can be taken as a template for creating concrete DiagnosticContributionSets of category DIAGNOSTICS_SYSTEM_EXTRACT or DIAGNOSTICS_ECU_EXTRACT
- DIAGNOSTICS_SYSTEM_EXTRACT: the scope of this DiagnosticContributionSet consists of several EcuInstances.
- DIAGNOSTICS_ECU_EXTRACT: the scope of this DiagnosticContributionSet consists of a **single** EcuInstances.

⌋*(RS_DEXT_00001, RS_DEXT_00002)*

[constr_1327] Multiplicity of DiagnosticEcuInstanceProps.ecuInstance ⌈ The multiplicity of DiagnosticEcuInstanceProps.ecuInstance shall be limited to 1 and the enclosing DiagnosticContributionSet shall only refer to at most one DiagnosticEcuInstanceProps if the enclosing DiagnosticContributionSet is of category DIAGNOSTICS_ECU_EXTRACT. ⌋*()*

[constr_1328] Consistency of DiagnosticEcuInstanceProps.ecuInstance and DiagnosticServiceTable.ecuInstance ⌈ Each DiagnosticServiceTable referenced by any given DiagnosticContributionSet in the role serviceTable shall define a reference in the role DiagnosticServiceTable.ecuInstance to an EcuInstance that is also referenced in the role DiagnosticEcuInstanceProps.ecuInstance by a DiagnosticEcuInstanceProps referenced by the mentioned DiagnosticContributionSet. ⌋*()*

Please note that [constr_1328] resolves an intentional redundancy in the meta-model. Both DiagnosticContributionSet and DiagnosticServiceTable are able refer to EcuInstance with the idea that both DiagnosticContributionSet and DiagnosticServiceTable can be modeled independently from each other.

Of course, once the DiagnosticContributionSet and DiagnosticServiceTable are integrated in the same context (in particular by establishing the reference DiagnosticContributionSet.serviceTable) the individual references to the applicable EcuInstances need to be consistent.

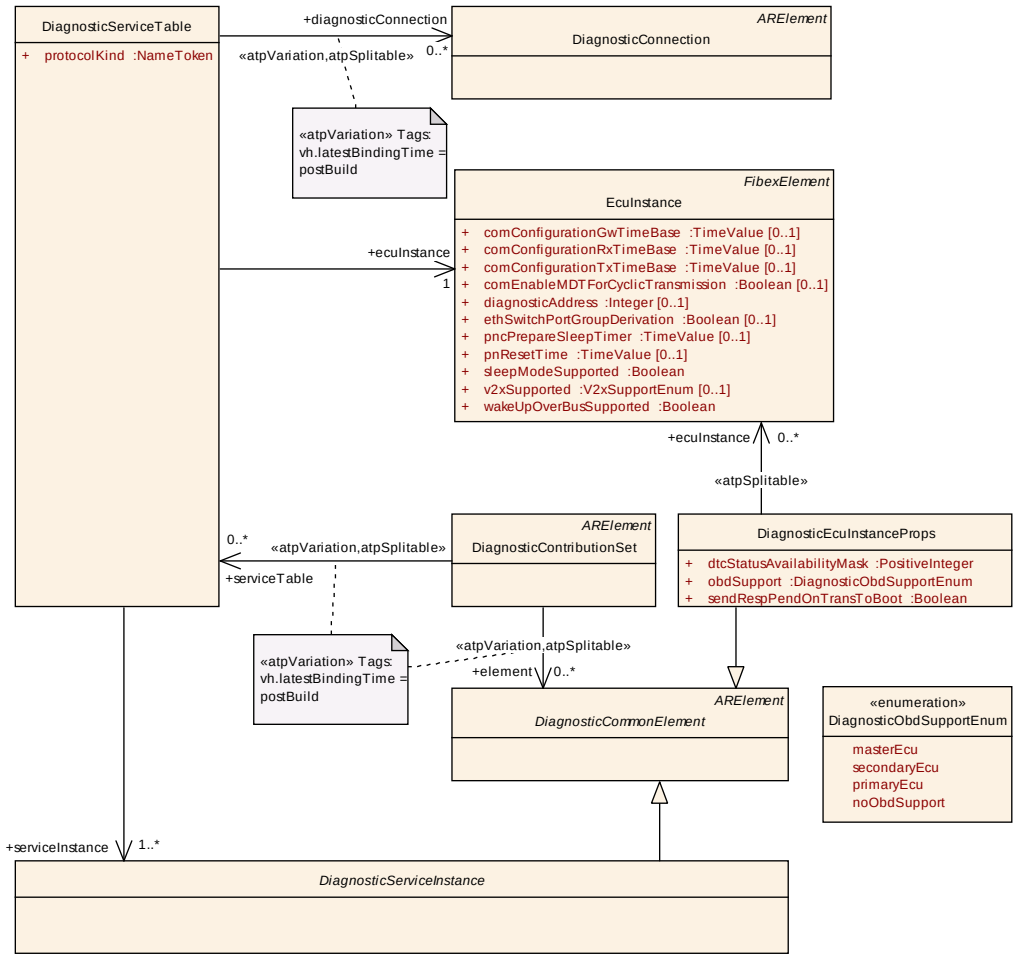

Figure 4.4: Diagnostic Contribution

[TPS_DEXT_01005] `DiagnosticContributionSet` can exist independently ⌈ The `DiagnosticContributionSet` has been modeled as an `ARElement` so that its instances can exist independently from the existence of context-providing model-elements inside a given `ARPackage`. ⌋*(RS_DEXT_00001, RS_DEXT_00002)*

Class	**ARElement (abstract)**			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::ARPackage			
Note	An element that can be defined stand-alone, i.e. without being part of another element (except for packages of course).			
Base	ARObject, CollectableElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 4.13: ARElement

Class	**ARPackage**			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::ARPackage			
Note	AUTOSAR package, allowing to create top level packages to structure the contained AREIements. 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.			
Base	ARObject, AtpBlueprint, AtpBlueprintable, CollectableElement, Identifiable, MultilanguageReferrable, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
arPackage	ARPackage	*	aggr	This represents a sub package within an ARPackage, thus allowing for an unlimited package hierarchy. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variation Point.shortLabel vh.latestBindingTime=blueprintDerivationTime xml.sequenceOffset=30
element	PackageableEle ment	*	aggr	Elements that are part of this package **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variation Point.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=20
referenceB ase	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. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=shortLabel xml.sequenceOffset=10

Table 4.14: ARPackage

[TPS_DEXT_01005] elaborates on an important aspect that makes the `DiagnosticExtract` independent from the existence of a context. For example, it would have been possible to aggregate `DiagnosticContributionSet` somewhere, e.g. at `System`.

This kind of modeling intentionally puts `DiagnosticContributionSet` on the same level as e.g. `System`, as far as model granularity is concerned.

Class	**DiagnosticContributionSet**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticContribution			
Note	This meta-class represents a root node of a diagnostic extract. It bundles a given set of diagnostic model elements. The granularity of the DiagonsticContributionSet is arbitrary in order to support the aspect of decentralized configuration, i.e. different contributors can come up with an own DiagnosticContributionSet. **Tags:** atp.recommendedPackage=DiagnosticContributionSets			
Base	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
commonProperties	DiagnosticCommonProps	0..1	aggr	This attribute represents a collection of diagnostic properties that are shared among the entire DiagnosticContributionSet. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=commonProperties
element	DiagnosticCommonElement	*	ref	This represents a DiagnosticCommonElement considered in the context of the DiagnosticContributionSet **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=element, variationPoint.shortLabel vh.latestBindingTime=postBuild
serviceTable	DiagnosticServiceTable	*	ref	This represents the collection of DiagnosticServiceTables to be considered in the scope of this DiagnosticContributionSet. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=serviceTable, variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 4.15: DiagnosticContributionSet

4.5 Diagnostic Protocol

[TPS_DEXT_01124] Semantics of meta-class `DiagnosticProtocol` ⌈ The meta-class `DiagnosticProtocol` can be used to describe the usage of different diagnostic protocols as well as their priority. ⌋*(RS_DEXT_00059)*

Class	**DiagnosticProtocol**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticContribution			
Note	This meta-class represents the ability to define a diagnostic protocol. **Tags:** atp.recommendedPackage=DiagnosticProtocols			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

Attribute	Type	Mul.	Kind	Note
diagnostic Connectio n	DiagnosticConn ection	*	ref	This represents the collection of applicable DiagnosticConnections for this DiagnosticProtocol. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=diagnosticConnection, variationPoint.shortLabel vh.latestBindingTime=postBuild
priority	PositiveInteger	1	attr	This represents the priority of the diagnostic protocol in comparison to other diagnostic protocols. Lower numeric values represent higher protocol priority: • 0 - Highest protocol priority • 255 - Lowest protocol priority
protocolKin d	NameToken	1	attr	This identifies the applicable protocol.
sendResp PendOnTr ansToBoot	Boolean	0..1	attr	The purpose of this attribute is to define whether or not the ECU should send a NRC 0x78 (response pending) before transitioning to the bootloader (in this case the attribute shall be set to "true") or if the transition shall be initiated without sending NRC 0x78 (in this case the attribute shall be set to "false").
serviceTab le	DiagnosticServi ceTable	0..1	ref	This represents the service table applicable for the given diagnostic protocol. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=serviceTable, variation Point.shortLabel vh.latestBindingTime=postBuild

Table 4.16: DiagnosticProtocol

Each `DiagnosticProtocol` refers to at most one `DiagnosticServiceTable` and to a collection of `DiagnosticConnection`s.

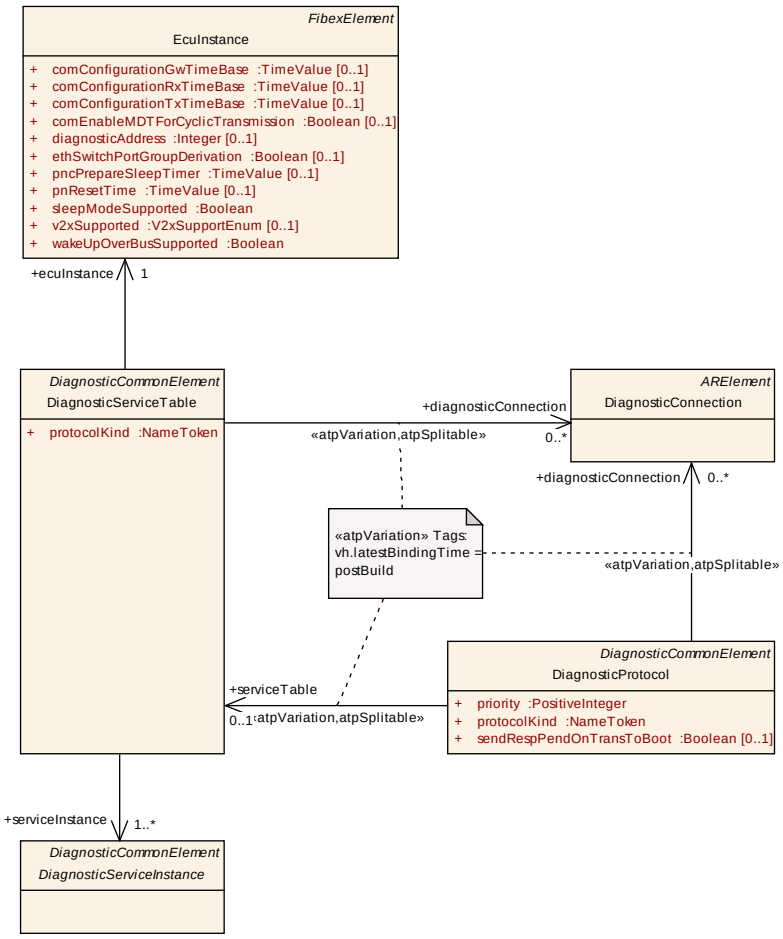

Figure 4.5: Modeling of `DiagnosticProtocol`

Please note that both `DiagnosticServiceTable` and `DiagnosticProtocol` have an attribute named `protocolKind`.

Class	**DiagnosticServiceTable**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticContribution			
Note	This meta-class represents a model of a diagnostic service table, i.e. the UDS services applicable for a given ECU. **Tags:** atp.recommendedPackage=DiagnosticServiceTables			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

Attribute	Type	Mul.	Kind	Note
diagnostic Connection	DiagnosticConnection	*	ref	This represents the DiagnosticConnection that is taken for handling the data transmission for the enclosing DiagnosticServiceTable. It is possible to refer to more than one diagnosticConnections in order to support more than one diagnostic tester. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=diagnosticConnection, variationPoint.shortLabel vh.latestBindingTime=postBuild
ecuInstance	EcuInstance	1	ref	This represents the applicable EcuInstance for this DiagnosticServiceTable.
protocolKind	NameToken	1	attr	This identifies the applicable protocol.
serviceInstance	DiagnosticServiceInstance	1..*	ref	This represents the collection of DiagnosticServiceInstances to be considered in the scope of this DiagnosticServiceTable,

Table 4.17: DiagnosticServiceTable

The attribute DiagnosticServiceTable.protocolKind shall be used to define the applicability of a DiagnosticServiceTable for a given protocol before the formal definition of the protocol even exists.

In other words, the attribute gives the designer of the DiagnosticServiceTable a means to express an intention about the usage of the DiagnosticServiceTable.

The attribute DiagnosticServiceTable.protocolKind, on the other hand, is used to define the actual nature of the DiagnosticProtocol.

By means of DiagnosticProtocol.serviceTable both "ends" of this aspect are put together and therefore it is reasonable to express a constraint about the value of attribute protocolKind

[constr_1405] Value of DiagnosticProtocol.serviceTable vs. DiagnosticServiceTable.protocolKind ⌈ If the reference DiagnosticProtocol.serviceTable exists then the value of DiagnosticProtocol.protocolKind shall be identical to the value of DiagnosticServiceTable.protocolKind. ⌋*()*

[TPS_DEXT_01006] The role of DiagnosticServiceTables ⌈ The existence of a DiagnosticServiceTable creates a formal relation between a collection of DiagnosticServiceInstances and the DiagnosticConnection that formalizes a *conduit* for specific pairs of diagnostic request and response messages taken to transmit the diagnostic service invocations from a tester to the applicable instance of the AUTOSAR diagnostic stack and convey the response of the diagnostic stack back to the tester.

In particular, this means that a DiagnosticServiceTable describes the set of DiagnosticServiceInstances that are available via DiagnosticConnection

which is finally a request message to address a diagnostic service to an ECU and a response message to be used by the ECU to respond to the service. ⌋ *(RS_DEXT_00039)*

Class	**DiagnosticConnection**			
Package	M2::AUTOSARTemplates::SystemTemplate::DiagnosticConnection			
Note	DiagnosticConncection that is used to describe the relationship between several TP connections. **Tags:** atp.recommendedPackage=DiagnosticConnections			
Base	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
functionalRequest	TpConnectionIdent	*	ref	Reference to functional request messages.
periodicResponseUudt	PduTriggering	*	ref	Reference to UUDT responses.
physicalRequest	TpConnectionIdent	0..1	ref	Reference to a physical request message.
response	TpConnectionIdent	0..1	ref	In the vast majority of cases a response is required. However, there are also cases where providing the response is not possible and/or not allowed.
responseOnEvent	TpConnectionIdent	0..1	ref	Reference to a ROE message.

Table 4.18: DiagnosticConnection

Here is an example of a service table for UDS diagnostics:

$14 - GroupOfDTC: 0xFFFFFF

$19 - Subfunction: $02, Subfunction $06

$22 - DataID: 0x1111, DataID: 0x2222

$2E - DataID: 0x1111, DataID: 0x2222

$2F - IO-ID:0x3333

[TPS_DEXT_01091] Relation between a `DiagnosticServiceTable` and one or more `DiagnosticConnection`s ⌈ In principle, the relation between a `DiagnosticServiceTable` and one or more `DiagnosticConnection`s can be established in two possible ways:

- By means of the reference `DiagnosticServiceTable.diagnosticConnection`. This way, the concept of protocols and their priorities relative to each other is **not** considered.

- By means of the references DiagnosticProtocol.diagnosticConnection and DiagnosticProtocol.serviceTable. This way, the existence of several protocols and their priorities **is positively considered**.

⌋()

Please note that alternatives mentioned in [TPS_DEXT_01091] may or may not be handled as alternatives in actual projects. It may be possible that in a first modeling step protocols are ignored entirely and therefore the respective relations are created by means of DiagnosticServiceTable.diagnosticConnection.

Later in time and as the project progresses, protocols may become a thing and are consequently introduced in the model. In response to this change, the relation in question is now created by means of the references DiagnosticProtocol.diagnosticConnection and DiagnosticProtocol.serviceTable.

However, the existing relation created by means of the reference DiagnosticServiceTable.diagnosticConnection is not necessarily required to be removed. It may just as well continue to exist. However, in this case an obvious consistency rule as described in [constr_1406] applies.

[constr_1406] DiagnosticServiceTable.diagnosticConnection vs. DiagnosticProtocol.diagnosticConnection ⌈ If a DiagnosticServiceTable exists that fulfills the following conditions:

- reference DiagnosticServiceTable.diagnosticConnection exists
- the DiagnosticServiceTable is referenced by means of DiagnosticProtocol.serviceTable

then all of the DiagnosticConnections referenced by means of DiagnosticServiceTable.diagnosticConnection shall also be referenced in the role diagnosticConnection from a DiagnosticProtocol that in turn references the respective DiagnosticServiceTable in the role DiagnosticProtocol.serviceTable. ⌋
()

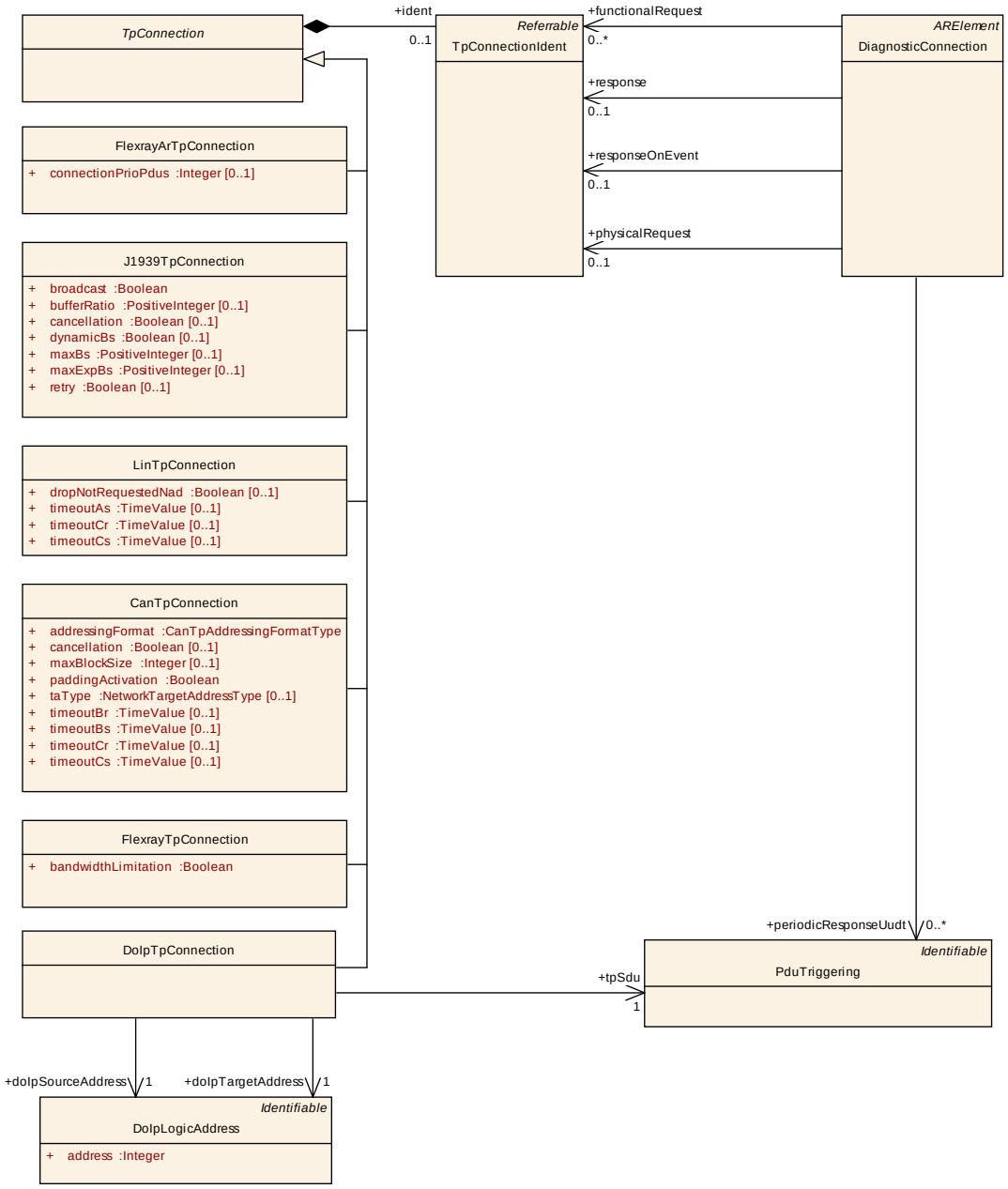

Figure 4.6: Modeling of `DiagnosticConnection`

4.6 Diagnostic Common Properties

[TPS_DEXT_01007] Common properties of a `DiagnosticExtract` ⌈ There are some properties of a `DiagnosticExtract` that are shared among all elements of the `DiagnosticExtract`. These properties are modeled by means of the meta-class `DiagnosticCommonProps`. ⌋*(RS_DEXT_00001)*

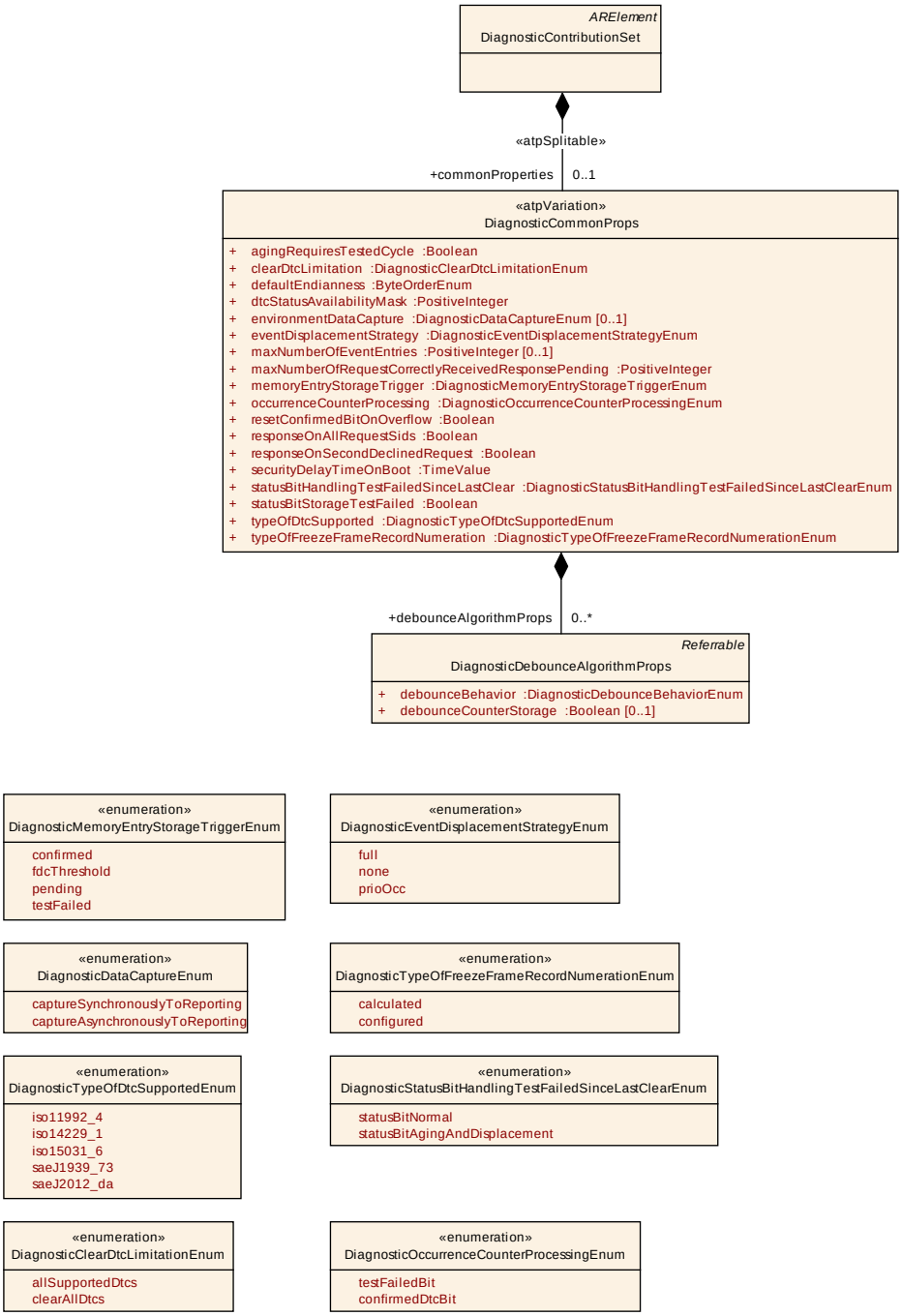

Figure 4.7: Common Diagnostic Properties

[TPS_DEXT_01008] `DiagnosticContributionSet` **defines the scope for the application of the common diagnostic properties** ⌈ `DiagnosticContributionSet` aggregates `DiagnosticCommonProps` and by this means defines the scope for the application of the common diagnostic properties. ⌋*(RS_DEXT_00001)*

Class	≪atpVariation≫ **DiagnosticCommonProps**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps			
Note	This meta-class aggregates a number of common properties that are shared among a diagnostic extract. **Tags:** vh.latestBindingTime=codeGenerationTime			
Base	ARObject			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
agingRequiresTestedCycle	Boolean	1	attr	Defines whether the aging cycle counter is processed every aging cycles or else only tested aging cycle are considered. If the attribute is set to TRUE: only tested aging cycle are considered for aging cycle counter. If the attribute is set to FALSE: aging cycle counter is processed every aging cycle.
clearDtcLimitation	DiagnosticClearDtcLimitationEnum	1	attr	Defines the scope of the DEM_ClearDTC Api.
debounceAlgorithmProps	DiagnosticDebounceAlgorithmProps	*	aggr	Defines the used debounce algorithms relevant in the context of the enclosing DiagnosticCommonProps. Usually, there is a variety of debouncing algorithms to take into account and therefore the multiplicity of this aggregation is set to 0..*.
defaultEndianness	ByteOrderEnum	1	attr	Defines the default endianness of the data belonging to a DID or RID which is applicable if the DiagnosticDataElement does not define the endianness via the swDataDefProps.baseType attribute.
dtcStatusAvailabilityMask	PositiveInteger	1	attr	Mask for the supported DTC status bits by the Dem.
environmentDataCapture	DiagnosticDataCaptureEnum	0..1	attr	This attribute determines whether the capturing of environment data is done synchronously inside the report API function or whether the capturing shall be done asynchronously, i.e. after the report API function already terminated.
eventDisplacementStrategy	DiagnosticEventDisplacementStrategyEnum	1	attr	This attribute defines, whether support for event displacement is enabled or not, and which displacement strategy is followed.
maxNumberOfEventEntries	PositiveInteger	0..1	attr	This attribute fixes the maximum number of event entries in the fault memory.
maxNumberOfRequestCorrectlyReceivedResponsePending	PositiveInteger	1	attr	Maximum number of negative responses with response code 0x78 (requestCorrectlyReceived-ResponsePending) allowed per request. DCM will send a negative response with response code 0x10 (generalReject), in case the limit value gets reached. Value 0xFF means that no limit number of NRC 0x78 response apply.

Attribute	Type	Mul.	Kind	Note
memoryEntryStorageTrigger	DiagnosticMemoryEntryStorageTriggerEnum	1	attr	Describes the primary trigger to allocate an event memory entry.
occurrenceCounterProcessing	DiagnosticOccurrenceCounterProcessingEnum	1	attr	This attribute defines the consideration of the fault confirmation process for the occurrence counter.
resetConfirmedBitOnOverflow	Boolean	1	attr	This attribute defines, whether the confirmed bit is reset or not while an event memory entry will be displaced.
responseOnAllRequestSids	Boolean	1	attr	If set to FALSE the DCM will not respond to diagnostic request that contains a service ID which is in the range from 0x40 to 0x7F or in the range from 0xC0 to 0xFF (Response IDs).
responseOnSecondDeclinedRequest	Boolean	1	attr	Defines the reaction upon a second request (ClientB) that can not be processed (e.g. due to priority assessment). TRUE: when the second request (Client B) can not be processed, it shall be answered with NRC21 BusyRepeatRequest. FALSE: when the second request (Client B) can not be processed, it shall not be responded.
securityDelayTimeOnBoot	TimeValue	1	attr	Start delay timer on power on in seconds. This delay indicates the time at ECU boot power-on time where the Dcm remains in the default session and does not accept a security access.
statusBitHandlingTestFailedSinceLastClear	DiagnosticStatusBitHandlingTestFailedSinceLastClearEnum	1	attr	This attribute defines, whether the aging and displacement mechanism shall be applied to the "TestFailedSinceLastClear" status bits.
statusBitStorageTestFailed	Boolean	1	attr	This parameter is used to activate/deactivate the permanent storage of the "TestFailed" status bits. true: storage activated false: storage deactivated
typeOfDtcSupported	DiagnosticTypeOfDtcSupportedEnum	1	attr	This attribute defines the format returned by Dem_DcmGetTranslationType and does not relate to/influence the supported Dem functionality.
typeOfFreezeFrameRecordNumeration	DiagnosticTypeOfFreezeFrameRecordNumerationEnum	1	attr	This attribute defines the type of assigning freeze frame record numbers for event-specific freeze frame records.

Table 4.19: DiagnosticCommonProps

Enumeration	DiagnosticMemoryEntryStorageTriggerEnum
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps
Note	Trigger types to allocate an event memory entry.
Literal	Description

confirmed	Status information of UDS DTC status bit 3 **Tags:** atp.EnumerationValue=0
fdcThreshold	Threshold to allocate an event memory entry and to capture the Freeze Frame. **Tags:** atp.EnumerationValue=1
pending	Status information of UDS DTC status bit 2. **Tags:** atp.EnumerationValue=2
testFailed	Status information of UDS DTC status bit 0. **Tags:** atp.EnumerationValue=3

Table 4.20: DiagnosticMemoryEntryStorageTriggerEnum

Enumeration	**DiagnosticDataCaptureEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps
Note	Data triggering types
Literal	***Description***
captureAsynchronouslyToReporting	This represents the intention to capture the environment data asynchronously after the actual capture API function terminated. **Tags:** atp.EnumerationValue=0
captureSynchronouslyToReporting	This represents the intention to capture the environment data synchronously within the capture API function. **Tags:** atp.EnumerationValue=1

Table 4.21: DiagnosticDataCaptureEnum

Enumeration	**DiagnosticTypeOfFreezeFrameRecordNumerationEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps
Note	FreezeFrame record numeration type
Literal	***Description***
calculated	Freeze frame records will be numbered consecutive starting by 1 in their chronological order. **Tags:** atp.EnumerationValue=0
configured	Freeze frame records will be numbered based on the given configuration in their chronological order. **Tags:** atp.EnumerationValue=1

Table 4.22: DiagnosticTypeOfFreezeFrameRecordNumerationEnum

Enumeration	**DiagnosticClearDtcLimitationEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps
Note	Scope of the DEM_ClearDTC Api.

Literal	*Description*
allSupported Dtcs	DEM_ClearDtc API accepts all supported DTC values. **Tags:** atp.EnumerationValue=0
clearAllDtcs	DEM_ClearDtc API accepts ClearAllDTCs only. **Tags:** atp.EnumerationValue=1

Table 4.23: DiagnosticClearDtcLimitationEnum

Enumeration	**DiagnosticEventDisplacementStrategyEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps
Note	Defines the displacement strategy.
Literal	*Description*
full	Event memory entry displacement is enabled, by consideration of priority active/passive status, and occurrence. **Tags:** atp.EnumerationValue=0
none	Event memory entry displacement is disabled. **Tags:** atp.EnumerationValue=1
prioOcc	Event memory entry displacement is enabled, by consideration of priority and occurrence (but without active/passive status). **Tags:** atp.EnumerationValue=2

Table 4.24: DiagnosticEventDisplacementStrategyEnum

Enumeration	**DiagnosticOccurrenceCounterProcessingEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps
Note	The occurrence counter triggering types.
Literal	*Description*
confirmedDtc Bit	The occurrence counter is triggered by the TestFailed bit if the fault confirmation was successful (ConfirmedDTC bit is set). **Tags:** atp.EnumerationValue=0
testFailedBit	The occurrence counter is only triggered by the TestFailed bit (and the fault confirmation is not considered). **Tags:** atp.EnumerationValue=1

Table 4.25: DiagnosticOccurrenceCounterProcessingEnum

Enumeration	**DiagnosticTypeOfDtcSupportedEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticCommonProps
Note	Supported Dtc Types
Literal	*Description*

iso11992_4	ISO11992-4 DTC format **Tags:** atp.EnumerationValue=0
iso14229_1	ISO14229-1 DTC format (3 byte format) **Tags:** atp.EnumerationValue=1
iso15031_6	ISO15031-6 DTC format (2 byte format) **Tags:** atp.EnumerationValue=2
saeJ1939_73	SAEJ1939-73 DTC format **Tags:** atp.EnumerationValue=3
saeJ2012_da	SAE_J2012-DA_DTCFormat_00 (3 byte format) **Tags:** atp.EnumerationValue=4

Table 4.26: DiagnosticTypeOfDtcSupportedEnum

5 Diagnostic Services

5.1 Introduction

The meta-model for the diagnostic services according to AUTOSAR, to a large degree, takes over aspects of the description of diagnostic services according to the definition of *Unified Diagnostic Services* (UDS) as of ISO 14229 [16].

5.2 Service Instance vs. Service Class

When it comes to diagnostic services, the meta-model distinguishes between the concept of a diagnostic service *instance* vs. the concept of a diagnostic service *class*.

As the terminology suggests, the diagnostic service *instance* (formalized as `DiagnosticServiceInstance`) implements a concrete use of a diagnostic service in a given context whereas purpose of the diagnostic service *class* (formalized as `DiagnosticServiceClass`) is to provide properties that are shared among all existing diagnostic service *instance*s in the model.

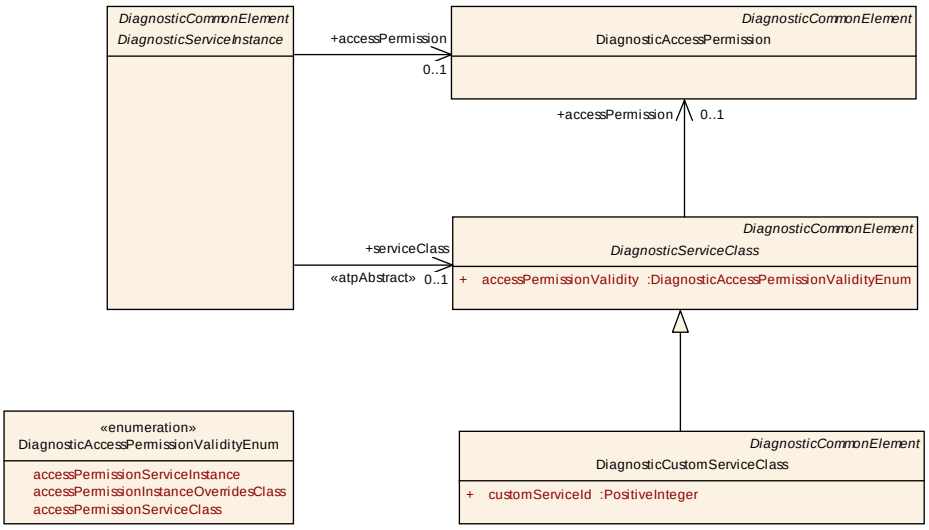

Figure 5.1: Common service elements

[constr_1329] Existence of concrete sub-classes of `DiagnosticServiceClass` in the context created by a `DiagnosticContributionSet` ⌈ One of the following mutually exclusive conditions shall apply for the existence of any concrete sub-class of `DiagnosticServiceClass` in the context created by a `DiagnosticContributionSet`:

- The subclass of `DiagnosticServiceClass` shall only appear once in the context created by a `DiagnosticContributionSet`
- If the subclass of `DiagnosticServiceClass` appears multiple times in the context created by a `DiagnosticContributionSet` then all instances shall have identical values for all of their attributes.

  In case of aggregations the number of aggregated elements shall be identical and the values of primitive attributes of aggregated elements shall again be identical.

⌋()

The background of [constr_1329] is obviously related to the semantics of `DiagnosticServiceClass` which is to define model attributes that are shared among all `DiagnosticServiceInstance`s.

This would not be possible if more that one `DiagnosticServiceClass` with a diverging set of attribute values exists.

Class	**DiagnosticServiceClass (abstract)**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommonService			
Note	This meta-class provides the ability to define common properties that are shared among all instances of sub-classes of DiagnosticServiceInstance.			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

Attribute	Type	Mul.	Kind	Note
accessPermission	DiagnosticAccessPermission	0..1	ref	This represents the collection of DiagnosticAccessPermissions that allow for the execution of the referencing DiagnosticServiceClass.
accessPermissionValidity	DiagnosticAccessPermissionValidityEnum	1	attr	This attribute is responsible for clarifying the validity of the accessPermission reference.

Table 5.1: DiagnosticServiceClass

Class	DiagnosticServiceInstance (abstract)			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommonService			
Note	This represents a concrete instance of a diagnostic service.			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
accessPermission	DiagnosticAccessPermission	0..1	ref	This represents the collection of DiagnosticAccessPermissions that allow for the execution of the referencing DiagnosticServiceInstance..
serviceClass	DiagnosticServiceClass	0..1	ref	This represents the corresponding "class", i.e. this meta-class provides properties that are shared among all instances of applicable sub-classes of DiagnosticServiceInstance. The subclasses that affected by this pattern implement references to the applicable "class"-role that substantiate this abstract reference. **Stereotypes:** atpAbstract

Table 5.2: DiagnosticServiceInstance

[TPS_DEXT_01009] Limited support for the configuration of custom diagnostic services ⌈ Beside the support for explicitly modeled diagnostic services, there is also a limited support for the configuration of custom diagnostic services.

The formalization, however, goes only so far as to define the `DiagnosticCustomServiceClass` and its attribute `customServiceId` that allows for the definition of the custom service identifier used for the purpose. ⌋*(RS_DEXT_00047)*

[constr_1330] Custom service identifier shall not overlap with standardized service identifiers ⌈ The value of the attribute `customServiceId` shall not be set to any of the values reserved for standardized service identifiers as defined by the ISO 14229-1, see [16]. ⌋*()*

[TPS_DEXT_01010] Configuration of custom diagnostic services ⌈ The support for the configuration of custom diagnostic services within the `DiagnosticExtract`

does not extend beyond the ability to define that attribute DiagnosticCustomServiceClass.customServiceId.

There is no corresponding formalization of a diagnostic service instance that supports references to e.g. a DiagnosticDataIdentifier. ⌋(*RS_DEXT_00047*)

Class	**DiagnosticCustomServiceClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommonService			
Note	This represents the ability to define a custom diagnostic service class and assign an ID to it. Further configuration is not foreseen from the point of view of the diagnostic extract and consequently needs to be done on the level of ECUC. **Tags:** atp.recommendedPackage=DiagnosticCustomServiceClasses			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
customServiceId	PositiveInteger	1	attr	This attribute may only be used for the definition of custom services. The values shall not overlap with existing standardized service IDs.

Table 5.3: DiagnosticCustomServiceClass

5.3 Access Permission, Session, Security Level

This chapter discusses a set of meta-classes that have been created to represent the concept of an *access permission* used in the context of the Dcm.

5.3.1 Introduction to Access Permission

The DiagnosticAccessPermission is used to describe the ability (or the lack thereof) to execute a diagnostic service depending on the referenced DiagnosticSecurityLevel and DiagnosticSession (see Figure 5.2).

At runtime, DiagnosticSessions are used to create a context for the execution of diagnostic functionality. Servers usually support a variety of different DiagnosticSessions. It is possible to switch between DiagnosticSessions at runtime.

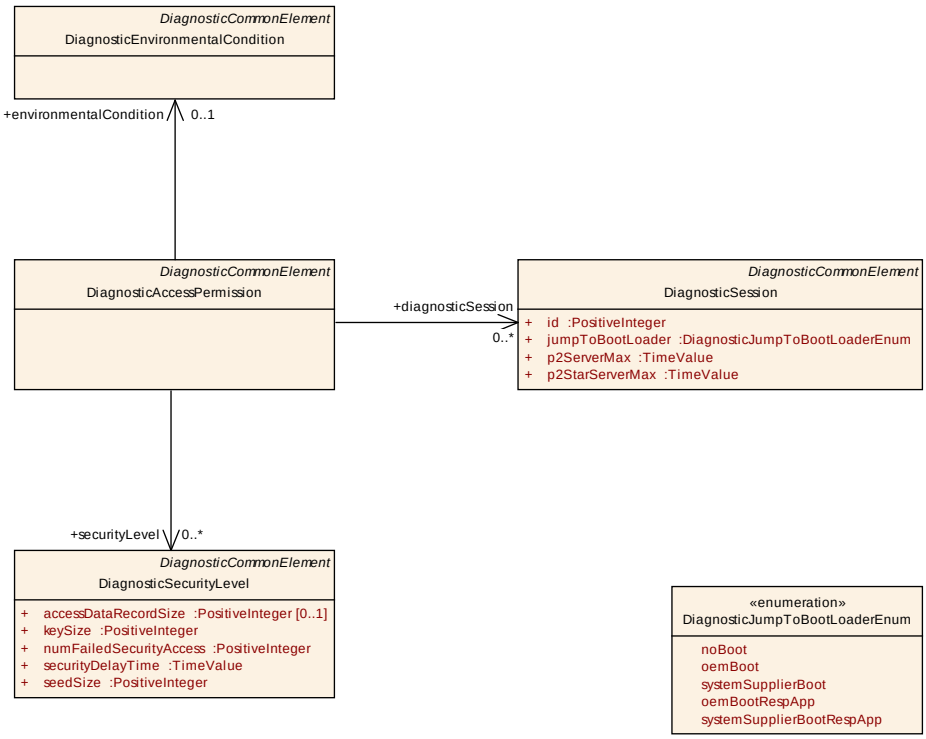

Figure 5.2: Common model elements relevant for the Dcm

[TPS_DEXT_01139] Semantics of the references from `DiagnosticAccessPermission` ⌈ The semantics of the references from `DiagnosticAccessPermission` to

- `DiagnosticSession` in the role `diagnosticSession`
- `DiagnosticSecurityLevel` in the role `securityLevel`
- `DiagnosticEnvironmentalCondition` in the role `environmentalCondition`

in terms of how access permission is granted is subject to the specification of the Dcm [10]. ⌋*(RS_DEXT_00040)*

[TPS_DEXT_01011] Semantics of `DiagnosticSession.id` ⌈ The value of the attribute `DiagnosticSession.id` has a given semantics according to ISO 14229-1 [16]. For the sake of completeness, the dedicated values of `DiagnosticSession.id` are:

0x01 This represents the *default session*. This session has a specific semantics in the context of diagnostics communication such that e.g. any diagnostic service that is supposed to execute in the *default session* cannot require any reference to a `DiagnosticSecurityLevel`.

0x02 This represents the *programming session*.

0x03 This represents the *extended diagnostic session*.

0x04 This represents the *safety system diagnostic session*.

The value range **0x40** .. **0x5F** is reserved for manufacturer-specific use. ⌋ *(RS_DEXT_00040)*

Class	**DiagnosticSecurityLevel**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm			
Note	This meta-class represents the ability to define a security level considered for diagnostic purposes. **Tags:** atp.recommendedPackage=DiagnosticSecurityLevels			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
accessDataRecordSize	PositiveInteger	0..1	attr	This represents the size of the AccessDataRecord used in GetSeed. Unit:byte.
keySize	PositiveInteger	1	attr	This represents the size of the security key. Unit: byte.
numFailedSecurityAccess	PositiveInteger	1	attr	This represents the number of failed security accesses after which the delay time is activated.
securityDelayTime	TimeValue	1	attr	This represents the delay time after a failed security access. Unit: second.
seedSize	PositiveInteger	1	attr	This represents the size of the security seed. Unit: byte.

Table 5.4: DiagnosticSecurityLevel

[TPS_DEXT_01012] Rationale for the modeling of the multiplicity of `DiagnosticAccessPermission.securityLevel` ⌈ The multiplicity of `DiagnosticAccessPermission.securityLevel` has been set to 0..* with the following motivation:

- The `DiagnosticSession` where the attribute `DiagnosticSession.id` is set to 0x01 shall **not** be associated with a `DiagnosticSecurityLevel`.
- There are no associated `DiagnosticSecurityLevel`s required. As a consequence, the execution of the `DiagnosticServiceInstance` that references the given `DiagnosticAccessPermission` is always possible.

⌋*(RS_DEXT_00041, RS_DEXT_00042)*

[TPS_DEXT_01070] Description of textually semi-formal formulated pre- and run-conditions for the validity of `DiagnosticAccessPermission` ⌈ AUTOSAR supports the description of textually formulated semi-formal pre- and run-conditions for the validity of `DiagnosticAccessPermission`.

This can be done by means of the attribute `DiagnosticAccessPermission.introduction.structuredReq`. ⌋*()*

For more details regarding the modeling of the semi-formal text please refer to Figure 4.3. An example of how the definition of pre- and run-conditions may look like in ARXML is sketched in listing 5.1.

To make this approach work it is important to standardize possible values of the attribute category such that the semi-formal semantics of the definition of pre- and run-conditions is protected by regulation of the AUTOSAR standard.

[TPS_DEXT_01071] Standardized values of DiagnosticAccessPermission.introduction.structuredReq ⌈ The following possible values of DiagnosticAccessPermission.introduction.structuredReq are standardized by AUTOSAR:

- **DIAG_ACCESS_PERM_PRE_COND**: this value describes the pre-condition of the corresponding DiagnosticAccessPermission.
- **DIAG_ACCESS_PERM_RUN_COND**: this value describes the run-condition of the corresponding DiagnosticAccessPermission.

⌋(*RS_DEXT_00041*, *RS_DEXT_00045*)

Listing 5.1: Example for the definition of pre- and run-conditions for DiagnosticAccessPermission

```
<DIAGNOSTIC-ACCESS-PERMISSION>
    <SHORT-NAME>exampleAccessPermission</SHORT-NAME>
    <INTRODUCTION>
        <STRUCTURED-REQ>
            <SHORT-NAME>precondition</SHORT-NAME>
            <CATEGORY>DIAG_ACCESS_PERM_PRE_COND</CATEGORY>
            <DESCRIPTION>
                <P>
                    <L-1 L="EN">This is a textual description of a pre-
                        condition</L-1>
                </P>
            </DESCRIPTION>
        </STRUCTURED-REQ>
        <STRUCTURED-REQ>
            <SHORT-NAME>runcondition</SHORT-NAME>
            <CATEGORY>DIAG_ACCESS_PERM_RUN_COND</CATEGORY>
            <DESCRIPTION>
                <P>
                    <L-1 L="EN">This is a textual description of a run-
                        condition</L-1>
                </P>
            </DESCRIPTION>
        </STRUCTURED-REQ>
    </INTRODUCTION>
    <DIAGNOSTIC-SESSION-REFS>
        <DIAGNOSTIC-SESSION-REF DEST="DIAGNOSTIC-SESSION">/AUTOSAR/
            UseCase_230/ExampleSession</DIAGNOSTIC-SESSION-REF>
    </DIAGNOSTIC-SESSION-REFS>
    <SECURITY-LEVEL-REFS>
        <SECURITY-LEVEL-REF DEST="DIAGNOSTIC-SECURITY-LEVEL">/AUTOSAR/
            UseCase_230/ExampleSecurityLevel</SECURITY-LEVEL-REF>
    </SECURITY-LEVEL-REFS>
</DIAGNOSTIC-ACCESS-PERMISSION>
```

[constr_1419] Value of `DiagnosticSecurityLevel.accessDataRecordSize` ⌈
If the attribute `DiagnosticSecurityLevel.accessDataRecordSize` exists then its value shall be greater than zero. ⌋*()*

Class	**DiagnosticAccessPermission**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm			
Note	This represents the specification of whether a given service can be accessed according to the existence of meta-classes referenced by a particular DiagnosticAccessPermission. In other words, this meta-class acts as a mapping element between several (otherwise unrelated) pieces of information that are put into context for the purpose of checking for access rights. **Tags:** atp.recommendedPackage=DiagnosticAccessPermissions			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagnosticSession	DiagnosticSession	*	ref	This represents the associated DiagnosticSessions
environmentalCondition	DiagnosticEnvironmentalCondition	0..1	ref	This represents the environmental conditions associated with the access permission.
securityLevel	DiagnosticSecurityLevel	*	ref	This represents the associated DiagnosticSecurityLevels

Table 5.5: DiagnosticAccessPermission

Class	**DiagnosticSession**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm			
Note	This meta-class represents the ability to define a diagnostic session. **Tags:** atp.recommendedPackage=DiagnosticSessions			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
id	PositiveInteger	1	attr	This is the numerical identifier used to identify the DiagnosticSession in the scope of diagnostic workflow
jumpToBootLoader	DiagnosticJumpToBootLoaderEnum	1	attr	This attribute represents the ability to define whether this diagnostic session allows to jump to Bootloader (OEM Bootloader or System Supplier Bootloader). If this diagnostic session doesn't allow to jump to Bootloader the value JumpToBootLoaderEnum.noBoot shall be chosen.

Attribute	*Type*	*Mul.*	*Kind*	*Note*
p2ServerMax	TimeValue	1	attr	This is the session value for P2ServerMax in seconds (per Session Control). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds.
p2StarServerMax	TimeValue	1	attr	This is the session value for P2*ServerMax in seconds (per Session Control). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds.

Table 5.6: DiagnosticSession

Enumeration	**DiagnosticJumpToBootLoaderEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm
Note	This enumeration contains the options for jumping to a boot loader.
Literal	***Description***
noBoot	This diagnostic session doesn't allow to jump to Bootloader. **Tags:** atp.EnumerationValue=0
oemBoot	This diagnostic session allows to jump to OEM Bootloader. In this case the bootloader send the final response. **Tags:** atp.EnumerationValue=1
oemBootRespApp	This diagnostic session allows to jump to OEM Bootloader and application sends final response. **Tags:** atp.EnumerationValue=3
systemSupplierBoot	This diagnostic session allows to jump to System Supplier Bootloader. In this case the bootloader send the final response. **Tags:** atp.EnumerationValue=2
systemSupplierBootRespApp	This diagnostic session allows to jump to System Supplier Bootloader and application sends final response. **Tags:** atp.EnumerationValue=4

Table 5.7: DiagnosticJumpToBootLoaderEnum

Class	**DiagnosticSecurityLevel**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm			
Note	This meta-class represents the ability to define a security level considered for diagnostic purposes. **Tags:** atp.recommendedPackage=DiagnosticSecurityLevels			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
accessDataRecordSize	PositiveInteger	0..1	attr	This represents the size of the AccessDataRecord used in GetSeed. Unit:byte.
keySize	PositiveInteger	1	attr	This represents the size of the security key. Unit: byte.
numFailedSecurityAccess	PositiveInteger	1	attr	This represents the number of failed security accesses after which the delay time is activated.
securityDelayTime	TimeValue	1	attr	This represents the delay time after a failed security access. Unit: second.
seedSize	PositiveInteger	1	attr	This represents the size of the security seed. Unit: byte.

Table 5.8: DiagnosticSecurityLevel

5.3.2 Prioritization of Access Permission

The definition of *access permission* itself can be done on different levels. It is therefore necessary to define how the existence of *access permission* on these different levels is supposed to be interpreted.

[TPS_DEXT_01061] Supported scenarios for the definition of *access permission*
⌈ The following scenarios are possible for the definition of *access permission*:

- The *access permission* is defined on the level of a `DiagnosticServiceClass`. In this scenario, the **intended semantics** is that this configuration is binding for **all** `DiagnosticServiceInstance`s derived from the `DiagnosticServiceClass`.

  The configuration of a `DiagnosticServiceInstance.accessPermission` is considered as an error and shall be reported accordingly.

  This scenario applies if `DiagnosticServiceClass.accessPermissionValidity` is set to the value `accessPermissionServiceClass`.

- The *access permission* is defined on the level of an individual `DiagnosticServiceInstance`. In this scenario, the **intended semantics** is that the `DiagnosticServiceClass` shall **not make any assumptions about the definition of the applicable *access permission***.

The configuration of a DiagnosticServiceClass.accessPermission is considered as an error and shall be reported accordingly. This scenario applies if DiagnosticServiceClass.accessPermissionValidity is set to the value accessPermissionServiceInstance.

- The definition of both DiagnosticServiceClass.accessPermission and DiagnosticServiceInstance.accessPermission is positively allowed.

  In this scenario, the **intended semantics** is that if DiagnosticServiceClass.accessPermission exists the individual DiagnosticServiceInstances **are not required to define** DiagnosticServiceInstance.accessPermission **but if they do** then the DiagnosticServiceInstance.accessPermission **gets priority** over the definition of DiagnosticServiceClass.accessPermission.

  This basically boils down to the ability to **override** the setting for *access permission* made on the level of a DiagnosticServiceClass by the setting on the level of an DiagnosticServiceInstance.

  At the same time, this scenario saves some file footprint because (given the existence of DiagnosticServiceClass.accessPermission) there is no need to define individual DiagnosticServiceInstance.accessPermission unless there is a dedicated need for them.

  This scenario applies if DiagnosticServiceClass.accessPermissionValidity is set to the value accessPermissionInstanceOverridesClass.

⌋(*RS_DEXT_00041*, *RS_DEXT_00049*, *RS_DEXT_00050*)

The scenarios defined by [TPS_DEXT_01061] need modeling support in order to allow the user to precisely express the intended semantics of a model with respect to *access permission*. For this purpose the attribute DiagnosticServiceClass.accessPermissionValidity is available.

Enumeration	**DiagnosticAccessPermissionValidityEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommonService
Note	This meta-class provides settings for how the accessPermission shall be resolved between DiagnosticServiceInstance and DiagnosticServiceClass.
Literal	***Description***
accessPermissionInstanceOverridesClass	This means that accessPermission set at the DiagnosticServiceInstance will override the accessPermission defined at the DiagnosticServiceClass. **Tags:** atp.EnumerationValue=1
accessPermissionServiceClass	This means that the DiagnosticServiceClass is in charge to define the accessPermission. **Tags:** atp.EnumerationValue=2

accessPermission Service Instance	This means that the DiagnosticServiceInstance is in charge of defining the accessPermission **Tags:** atp.EnumerationValue=3

Table 5.9: DiagnosticAccessPermissionValidityEnum

[TPS_DEXT_01062] Existence of `DiagnosticServiceClass.accessPermissionValidity` in an incomplete model ⌈ If the attribute `DiagnosticServiceClass.accessPermissionValidity` does not exist then it shall be assumed that the configuration is incomplete. ⌋*(RS_DEXT_00001, RS_DEXT_00002, RS_DEXT_00041)*

Please note that the model state described in [TPS_DEXT_01062] is still allowed because it may only be possible to decide about the value of the attribute at later points in the model's life cycle.

[TPS_DEXT_01063] Existence of `DiagnosticServiceClass.accessPermissionValidity` in a complete model ⌈ As the model's life cycle approaches the point where the model is considered complete the attribute `DiagnosticServiceClass.accessPermissionValidity` shall exist in order to be able to properly figure out the intended model semantics. ⌋*(RS_DEXT_00001, RS_DEXT_00002, RS_DEXT_00041)*

[TPS_DEXT_01052] Existence of attribute `DiagnosticServiceInstance.accessPermission` ⌈ Regarding the existence of the attribute `DiagnosticServiceInstance.accessPermission` the following rules apply:

- If neither the attribute `DiagnosticServiceInstance.accessPermission` or `DiagnosticServiceClass.accessPermission` exist it is assumed that the configuration is incomplete as no access permission is defined.
- If either the attribute `DiagnosticServiceInstance.accessPermission` or `DiagnosticServiceClass.accessPermission` exists but does not have further references to `DiagnosticSession`, `DiagnosticEnvironmentalCondition`, or `DiagnosticSecurityLevel` then this means that the affected diagnostic services can be executed in any diagnostic session or security level. In other words, no restriction applies.

⌋*(RS_DEXT_00041, RS_DEXT_00049)*

5.4 Environmental Conditions for the Execution of Diagnostic Services

In some cases, diagnostic functionality can only be executed if the vehicle is in a (safe) state that allows for the respective diagnostics function. For example, one such condition is that the vehicle is not driving, i.e. vehicle speed == 0.

The meta-class DiagnosticEnvironmentalCondition formalizes the idea of a condition which is evaluated during runtime of the ECU by looking at "environmental" states (such as the mentioned vehicle speed).

DiagnosticEnvironmentalConditions are based on the active modes in the application software or basic software or by comparison of DiagnosticDataElements with constant values. Atomic conditions can be combined by logical operations to form more complex conditions.

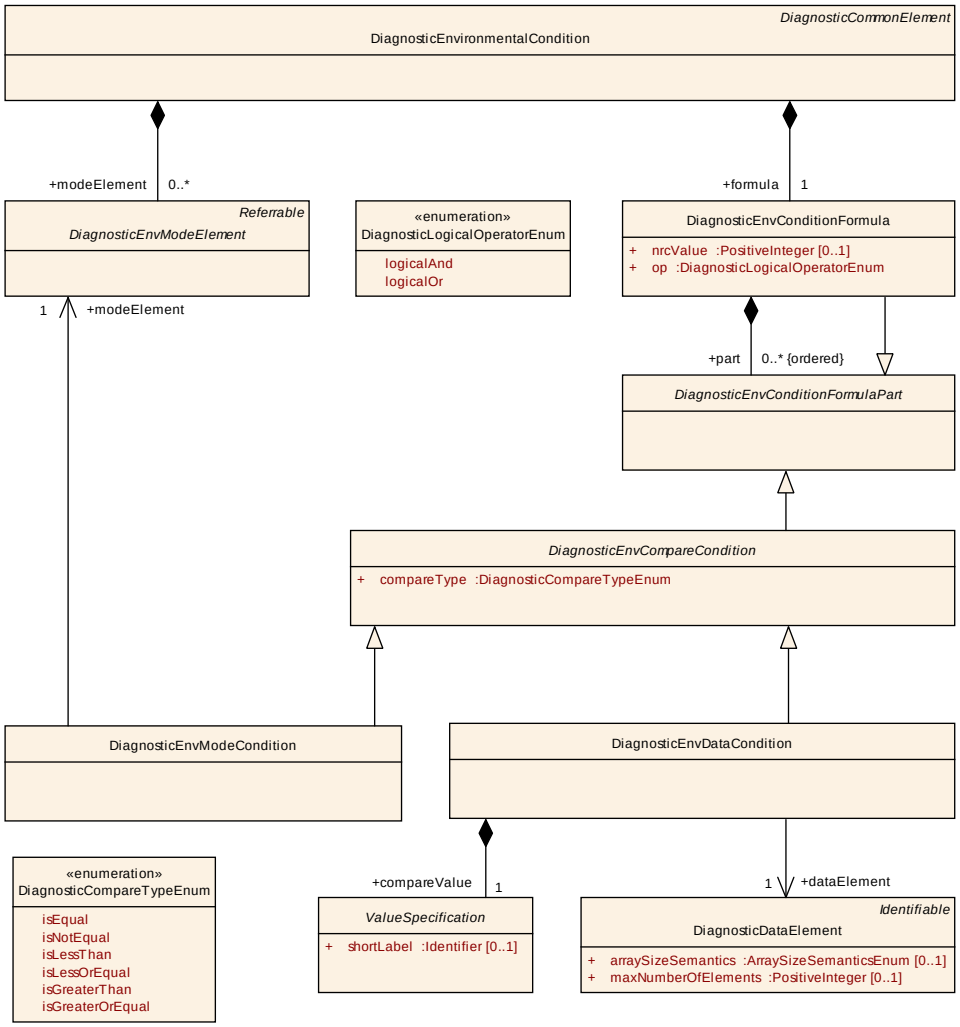

Figure 5.3: Formal modeling of the consideration of environmental conditions

Please note that it is possible to create a nested hierarchy (of arbitrary depth) of DiagnosticEnvConditionFormula. This modeling is supported by the fact that DiagnosticEnvConditionFormula inherits from and, at the same time, aggregates DiagnosticEnvConditionFormulaPart.

Class	**DiagnosticEnvironmentalCondition**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
Note	The meta-class DignosticEnvironmentalCondition formalizes the idea of a condition which is evaluated during runtime of the ECU by looking at "environmental" states (e.g. one such condition is that the vehicle is not driving, i.e. vehicle speed == 0). **Tags:** atp.recommendedPackage=DiagnosticEnvironmentalConditions			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
formula	DiagnosticEnvConditionFormula	1	aggr	This attribute represents the formula part of the DiagnosticEnvironmentalCondition.
modeElement	DiagnosticEnvModeElement	*	aggr	This aggregation contains a representation of ModeDeclarations in the context of a DiagnosticEnvironmentalCondition.

Table 5.10: DiagnosticEnvironmentalCondition

5.4.1 Environmental Condition Formula

The core part of a `DiagnosticEnvironmentalCondition` is the `DiagnosticEnvConditionFormula`.

Class	**DiagnosticEnvConditionFormula**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
Note	A DiagnosticEnvConditionFormula embodies the computation instruction that is to be evaluated at runtime to determine if the DiagnosticEnvironmentalCondition is currently present (i.e. the formula is evaluated to true) or not (otherwise). The formula itself consists of parts which are combined by the logical operations specified by DiagnosticEnvConditionFormula.op. If a diagnostic functionality cannot be executed because an environmental condition fails then the diagnostic stack shall send a negative response code (NRC) back to the client. The value of the NRC is directly related to the specific formula and is therefore formalized in the attribute DiagnosticEnvConditionFormula.nrcValue.			
Base	ARObject, DiagnosticEnvConditionFormulaPart			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
nrcValue	PositiveInteger	0..1	attr	This attribute represents the concrete NRC value that shall be returned if the condition fails.
op	DiagnosticLogicalOperatorEnum	1	attr	This attribute represents the concrete operator (supported operators: and, or) of the condition formula.
part (ordered)	DiagnosticEnvConditionFormulaPart	*	aggr	This aggregation represents the collection of formula parts that can be combined by logical operators.

Table 5.11: DiagnosticEnvConditionFormula

[TPS_DEXT_01113] Evaluation of a DiagnosticEnvConditionFormula ⌈ A DiagnosticEnvConditionFormula embodies the computation instruction that is to be evaluated at runtime to determine if the DiagnosticEnvironmentalCondition is currently present (i.e. the formula is evaluated to true) or not (otherwise).

The DiagnosticEnvConditionFormula itself consists of parts which are combined by the logical operations represented by the attribute op. ⌋*(RS_DEXT_00079)*

Class	**DiagnosticEnvConditionFormulaPart (abstract)**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
Note	A DiagnosticEnvConditionFormulaPart can either be a atomic condition, e.g. a DiagnosticEnvCompareCondition, or a DiagnosticEnvConditionFormula, again, which allows arbitrary nesting.			
Base	ARObject			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.12: DiagnosticEnvConditionFormulaPart

[TPS_DEXT_01114] DiagnosticEnvConditionFormula that has no parts ⌈ A DiagnosticEnvConditionFormula that has no parts shall be evaluated to false. This rule shall apply independently of the value of DiagnosticEnvConditionFormula.op. ⌋*(RS_DEXT_00079)*

[TPS_DEXT_01115] DiagnosticEnvConditionFormula that has one part ⌈ A DiagnosticEnvConditionFormula that has one part shall be evaluated to the evaluation result of this part. This rule shall apply independently of the value of DiagnosticEnvConditionFormula.op. ⌋*(RS_DEXT_00079)*

[TPS_DEXT_01116] DiagnosticEnvConditionFormula that has more than one part ⌈ The evaluation result of a DiagnosticEnvConditionFormula with more than one part shall be calculated by combining the results of the parts with the logical Operation specified by DiagnosticEnvConditionFormula.op.

The evaluation shall be done in a "short-cut" manner, i.e. following the rules for the evaluation of the logical operators in C.

The consequences of this approach are:

- The parts shall be evaluated in the specified order starting at the first element. This is why the aggregation of part is decorated by the ordered qualifier.
- after the evaluation of each part a check shall be executed if it is still possible that the evaluation of the remaining parts (to true or false) could change the overall result.
- As soon as a change of the overall result is no longer possible, i.e. any of the following conditions evaluates to false:
  - there are no parts left

– the current part was evaluated to false and DiagnosticEnvConditionFormula.op == DiagnosticLogicalOperatorEnum.logicalAnd
– the current part was evaluated to true and DiagnosticEnvConditionFormula.op == DiagnosticLogicalOperatorEnum.logicalOr

the evaluation of the parts shall be finalized and the evaluation result of the current part shall be considered the overall evaluation result of the formula.

Regarding the strategy for returning NRC values please refer to the specification of the SWS Dcm [10].

⌋(*RS_DEXT_00079*)

[TPS_DEXT_01117] Semantics of DiagnosticEnvConditionFormula.nrcValue ⌈ If a diagnostic functionality cannot be executed because an environmental condition fails, i.e. the formula is evaluated to false, then the diagnostic stack shall send an optional negative response code (NRC) back to the client (if it is present).

If no value for NRC is defined then a pre-defined NRC value as defined in the SWS Dcm [10] will be sent back. The value of the NRC is directly related to the specific DiagnosticEnvConditionFormula and is therefore formalized in the attribute DiagnosticEnvConditionFormula.nrcValue. ⌋(*RS_DEXT_00079*)

[constr_1464] Allowed value range of DiagnosticEnvConditionFormula.nrcValue ⌈ The value of attribute DiagnosticEnvConditionFormula.nrcValue shall be limited to the interval [1..255]. ⌋()

The rationale for the existence of [constr_1464] is provided by ISO 14229-1 [16].

5.4.2 Atomic Conditions

Atomic conditions in the context of a DiagnosticEnvConditionFormula are described by means of meta-class DiagnosticEnvCompareConditions. The formulation of DiagnosticEnvCompareConditions is based on the idea of a comparison at runtime of some variable data with a constant value.

The actual type of the comparison (==, !=, <, <=, ...) is specified by means of the attribute DiagnosticEnvCompareCondition.compareType.

Class	**DiagnosticEnvCompareCondition (abstract)**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
Note	DiagnosticCompareConditions are atomic conditions. They are based on the idea of a comparison at runtime of some variable data with something constant. The type of the comparison (==, !=, <, <=, ...) is specified in DiagnosticCompareCondition.compareType.			
Base	ARObject, DiagnosticEnvConditionFormulaPart			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

Attribute	Type	Mul.	Kind	Note
compareType	DiagnosticCompareTypeEnum	1	attr	This attributes represents the concrete type of the comparison.

Table 5.13: DiagnosticEnvCompareCondition

DiagnosticEnvCompareCondition is an abstract meta-class that acts as a base class for two concrete meta-classes designed to handle different operand types.

The specific sub-classes (the details are explained in sections 5.4.2.1 and 5.4.2.2) of DiagnosticEnvCompareCondition support a different set of attributes of DiagnosticCompareTypeEnum for setting the value of attribute compareType.

5.4.2.1 Data Condition

[TPS_DEXT_01118] Semantics of DiagnosticEnvDataCondition ⌈ The meta-class DiagnosticEnvDataCondition represents an atomic condition that compares the current value of the referenced DiagnosticDataElement with a constant value defined by the ValueSpecification aggregated in the role compareValue. ⌋*(RS_DEXT_00079)*

[constr_1465] Allowed values of compareType in the context of a DiagnosticEnvDataCondition ⌈ Within the context of a DiagnosticEnvDataCondition **all values** of DiagnosticCompareTypeEnum are supported for the inherited attribute compareType. ⌋*()*

Class	DiagnosticEnvDataCondition			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
Note	A DiagnosticEnvDataCondition is an atomic condition that compares the current value of the referenced DiagnosticDataElement with a constant value defined by the ValueSpecification. All compareTypes are supported.			
Base	ARObject, DiagnosticEnvCompareCondition, DiagnosticEnvConditionFormulaPart			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
compareValue	ValueSpecification	1	aggr	This attribute represents a fixed compare value taken to evaluate the compare condition.
dataElement	DiagnosticDataElement	1	ref	This reference represents the related diagnostic data element.

Table 5.14: DiagnosticEnvDataCondition

5.4.2.2 Mode Condition

[TPS_DEXT_01119] Semantics of DiagnosticEnvModeCondition ⌈ The meta-class DiagnosticEnvModeCondition represents an atomic condition that com-

pares the current value of the referenced ModeDeclarationGroupPrototype with the value of a ModeDeclaration taken as the reference value. ⌋(*RS_DEXT_00079*)

The concrete modeling of this aspect is sketched in Figure D.4.

Please note that the ModeDeclarationGroupPrototype as well as the applicable ModeDeclaration are both referenced in the concrete modeling of the attribute DiagnosticEnvSwcModeElement.mode.

The idea behind this modeling approach (for more information, please refer to Figure D.4) is that the ModeDeclaration can only be a member of the specific ModeDeclarationGroup taken to type the respective ModeDeclarationGroupPrototype anyway.

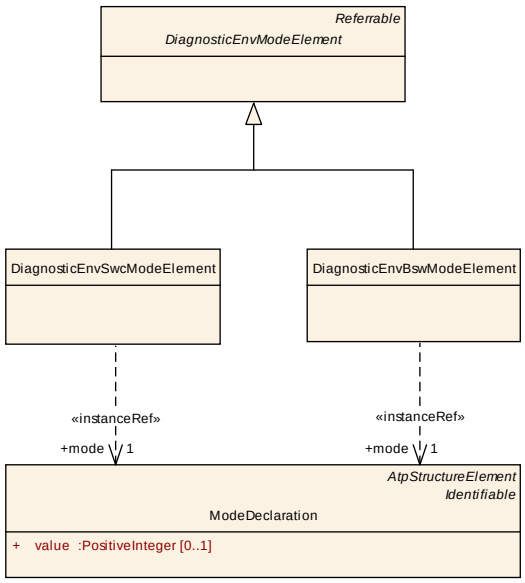

Figure 5.4: Specializations of DiagnosticEnvModeElement

Admittedly, this relation could be expressed by means of two references and a formal constraint **or** it could be expressed by way of a **single** reference that's implying an interpreting of the model in a specific way. AUTOSAR opts for the latter alternative, which is detailed by [TPS_DEXT_01120].

[TPS_DEXT_01120] Comparison of the value of a ModeDeclarationGroupPrototype with a ModeDeclaration ⌈ For the comparison of the value of a ModeDeclarationGroupPrototype with a ModeDeclaration, two alternatives (see Figure 5.4) apply, depending on whether mode condition is executed in application software (swc) or basic software (bsw):

- The ModeDeclarationGroupPrototype referenced in the role DiagnosticEnvSwcModeElement.mode.contextModeDeclarationGroup shall be compared to the ModeDeclaration referenced in the role DiagnosticEnvSwcModeElement.mode.targetMode.
- The ModeDeclarationGroupPrototype referenced in the role DiagnosticEnvBswModeElement.mode.contextModeDeclarationGroup shall be

compared to the ModeDeclaration referenced in the role DiagnosticEnvBswModeElement.mode.targetMode.

⌋(*RS_DEXT_00079*)

[constr_1466] Allowed values of compareType in the context of a DiagnosticEnvModeCondition ⌈ Within the context of a DiagnosticEnvDataCondition **only a subset** of the values of DiagnosticCompareTypeEnum is supported for the inherited attribute compareType, namely:

- DiagnosticCompareTypeEnum.isEqual
- DiagnosticCompareTypeEnum.isNotEqual

⌋()

[constr_1467] References in DiagnosticEnvModeCondition ⌈ In a DiagnosticEnvModeCondition the reference modeElement shall only point to a DiagnosticEnvModeElement that is aggregated inside the same DiagnosticEnvironmentalCondition as the DiagnosticEnvModeCondition itself. ⌋()

Please note that the main benefit of the existence of the dedicated meta-class DiagnosticEnvModeElement is to keep the (serialized) model clean. Given the fulfillment of [constr_1467], the potentially lengthy InstanceRef for identifying the operands of a mode comparison does not (if applicable) have to be repeated but can be reused multiple times in the context of the enclosing DiagnosticEnvironmentalCondition.

Class	**DiagnosticEnvModeCondition**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::Environmental Condition			
Note	DiagnosticEnvModeCondition are atomic condition based on the comparison of the active ModeDeclaration in a ModeDeclarationGroupProtoype with the constant value of a ModeDeclaration. The formulation of this condition uses only one DiagnosticEnvElement, which contains enough information to deduce the variable part (i.e. the part that changes at runtime) as well as the constant part of the comparison. Only DiagnosticCompareTypeEnum.isEqual or DiagnosticCompareTypeEnum.isNotEqual are eligible values for DiagnosticAtomicCondition.compareType.			
Base	ARObject, DiagnosticEnvCompareCondition, DiagnosticEnvConditionFormulaPart			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
modeElement	DiagnosticEnvModeElement	1	ref	This reference represents both the ModeDeclarationGroupPrototype and the ModeDeclaration relevant for the mode comparison.

Table 5.15: DiagnosticEnvModeCondition

Class	**DiagnosticEnvModeElement (abstract)**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
Note	All ModeDeclarations that are referenced in a DiagnosticEnvModeCondition must be defined as a DiagnosticEnvModeElement of this DignosticEnvironmentalCondition. This concept keeps the ARXML clean: It avoids that the DignosticEnvConditionFormula is cluttered by lengthy InstanceRef definitions. Furthermore, it allows that an InstanceRef only needs to be defined once and can be used multiple times in the different DiagnosticEnvModeConditions.			
Base	ARObject, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.16: DiagnosticEnvModeElement

Class	**DiagnosticEnvSwcModeElement**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
Note	This meta-class represents the ability to refer to a ModeDeclaration in a concrete System context.			
Base	ARObject, DiagnosticEnvModeElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
mode	ModeDeclaration	1	iref	This reference identifies both the ModeDeclarationGroupPrototype and the ModeDeclaration for the specific mode comparison.

Table 5.17: DiagnosticEnvSwcModeElement

Class	**DiagnosticEnvBswModeElement**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition			
Note	This meta-class represents the ability to refer to a specific ModeDeclaration in the scope of a BswModuleDescription.			
Base	ARObject, DiagnosticEnvModeElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
mode	ModeDeclaration	1	iref	This reference identifies both the ModeDeclarationGroupPrototype and the ModeDeclaration for the specific mode comparison.

Table 5.18: DiagnosticEnvBswModeElement

5.5 Diagnostic Services supported by AUTOSAR

The following sub-chapters describe the modeling of the collection of relevant diagnostic services as defined in the ISO 14229-1 [16]. This means that the definition of the AUTOSAR `DiagnosticExtract` does not explicitly support the total collection of diagnostic services as defined by [16].

Some of the diagnostic services compiled in this document define so-called sub-functions that need to be identified to fully specify the nature of the respective diagnostic service.

[TPS_DEXT_01045] Supported diagnostic services ⌈ The table 5.19 shows the UDS services supported by the `DiagnosticExtract`. ⌋(*RS_DEXT_00003*, *RS_DEXT_00004*, *RS_DEXT_00005*, *RS_DEXT_00006*, *RS_DEXT_00007*, *RS_DEXT_00008*, *RS_DEXT_00009*, *RS_DEXT_00010*, *RS_DEXT_00011*, *RS_DEXT_00012*, *RS_DEXT_00013*, *RS_DEXT_00014*, *RS_DEXT_00015*, *RS_DEXT_00016*, *RS_DEXT_00017*, *RS_DEXT_00018*, *RS_DEXT_00019*, *RS_DEXT_00020*, *RS_DEXT_00021*, *RS_DEXT_00022*)

ID	Service
0x10	SessionControl
0x11	EcuReset
0x14	ClearDiagnosticInformation
0x19	ClearDTCInformation
0x22	ReadDataByIdentifier
0x23	ReadMemoryByAddress
0x27	SecurityAccess
0x28	CommunicationControl
0x2A	ReadDataByPeriodicIdentifier
0x2C	DynamicallyDefineDataIdentifier
0x2E	WriteDataByIdentifier
0x2F	IOControl
0x31	RoutineControl
0x34	RequestDownload
0x35	RequestUpload
0x36	TransferData
0x37	RequestTransferExit
0x3D	WriteMemoryByAddress
0x85	ControlDTCSetting
0x86	ResponseOnEvent

Table 5.19: Supported diagnostic services

[TPS_DEXT_01013] Specification of sub-functions by means of attribute `DiagnosticServiceInstance.category` ⌈

In all cases where a diagnostic service defines a sub-function according to ISO 14229-1 [16], the value of the attribute category of the applicable sub-class of DiagnosticServiceInstance can be used to specify the applicable sub-function as a textual token.

Constraints are defined to clarify the existence of standardized values of the attribute category for the given sub-function. This implies that an instance of the given sub-class of DiagnosticServiceInstance only has a single sub-function at a time. ⌋ *(RS_DEXT_00049, RS_DEXT_00051)*

[TPS_DEXT_01014] Possible values of the category attribute for diagnostic services ⌈ AUTOSAR claims the right to standardize the possible values of the attribute category for given diagnostic services. ⌋*(RS_DEXT_00001, RS_DEXT_00051)*

If applicable, AUTOSAR allows for the usage of values of the attribute category other than the standardized values.

In this case, however, proprietary values of the attribute category shall be prefixed with a company-specific name fragment in order to avoid collisions that could occur if or when the list of possible values claimed by the AUTOSAR standard itself is extended. Example:

Listing 5.2: Example for the definition of a custom category

```
<AR-PACKAGE>
    <SHORT-NAME>DiagnosticExtractExample</SHORT-NAME>
    <ELEMENTS>
        <DIAGNOSTIC-ECU-RESET>
            <SHORT-NAME>ResetTheHardWay</SHORT-NAME>
            <CATEGORY>ACME_REALLY_HARD_RESET</CATEGORY>
        </DIAGNOSTIC-ECU-RESET>
    </ELEMENTS>
</AR-PACKAGE>
```

5.5.1 DataByIdentifier

This chapter describes the modeling of diagnostic services ReadDataByIdentifier (0x22) and WriteDataByIdentifier (0x2E).

The purpose of this diagnostic service is to enable a tester to request the values of data records from the AUTOSAR diagnostics stack. The data records are identified by a formally modeled DiagnosticDataIdentifier.

The modeling of this diagnostic service comprises the two meta-classes DiagnosticReadDataByIdentifier and DiagnosticWriteDataByIdentifier. These meta-classes both need to specify the set of DiagnosticDataIdentifiers as well as the set of applicable DiagnosticAccessPermissions.

As these properties are shared between instances of DiagnosticReadDataByIdentifier and DiagnosticWriteDataByIdentifier, an abstract base class named DiagnosticDataByIdentifier has been created that provides the ac-

tual references to DiagnosticDataIdentifier and DiagnosticAccessPermission.

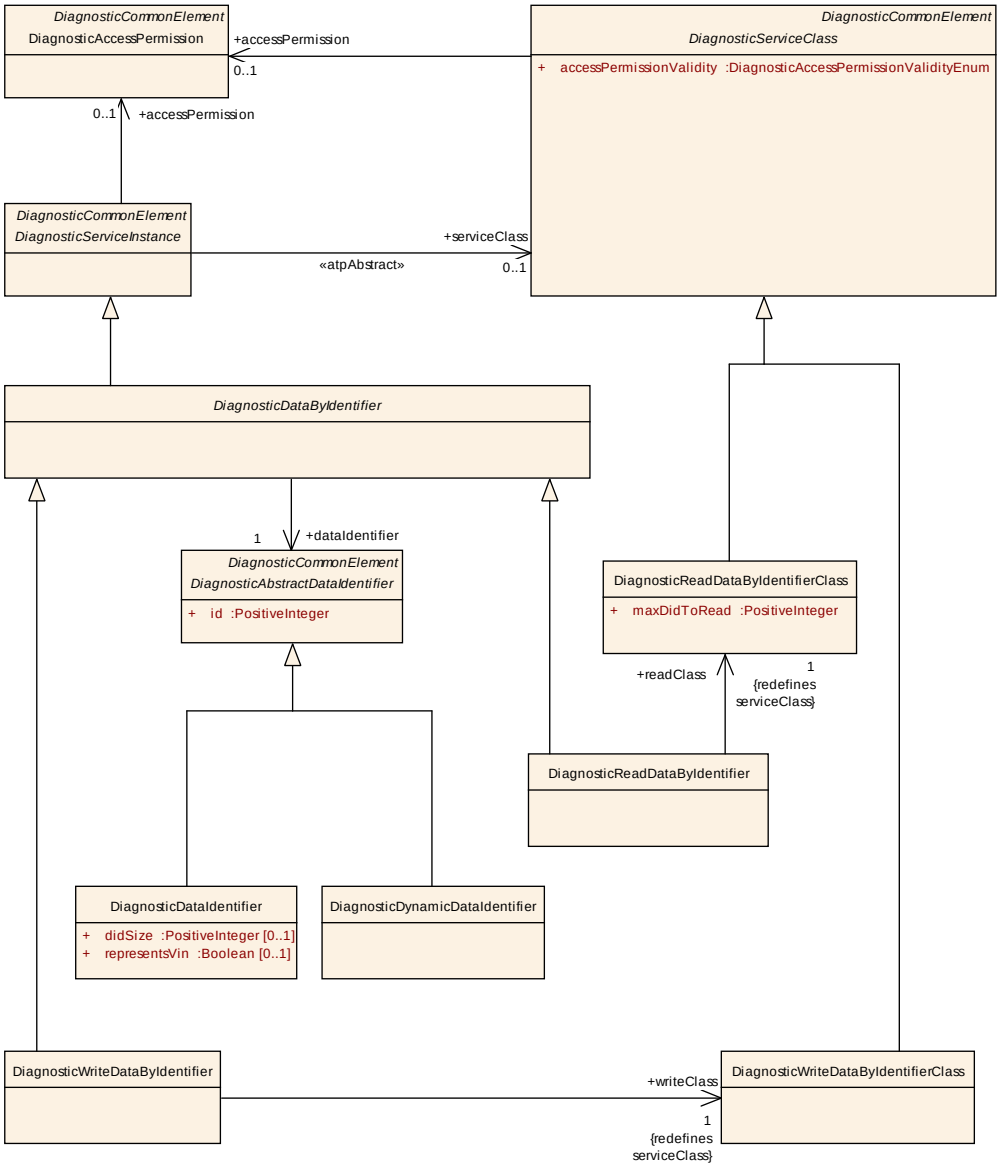

Figure 5.5: Modeling of diagnostic services `ReadDataByIdentifier` (0x22) and `WriteDataByIdentifier` (0x2E)

[TPS_DEXT_01054] Existence of `DiagnosticDataByIdentifier.dataIdentifier` ⌈ The configuration of a given DiagnosticDataByIdentifier is considered incomplete until the reference in the role DiagnosticDataByIdentifier.dataIdentifier exists. ⌋*(RS_DEXT_00007, RS_DEXT_00013, RS_DEXT_00034)*

The meaning of [TPS_DEXT_01054] is that the reference may be missing in intermediate steps of the configuration work flow. But at the point in time where ECU configuration is generated from the DiagnosticExtract the reference is needed to able to make sense of the model for the given DiagnosticDataByIdentifier.

The ability to read multiple DIDs at run-time is controlled via attribute DiagnosticReadDataByIdentifierClass.maxDidToRead and therefore it is sufficient to (at configuration-time) limit the multiplicity of attribute dataIdentifier to 1.

Please note that the reference DiagnosticDataByIdentifier.dataIdentifier goes to DiagnosticAbstractDataIdentifier. This modeling approach allows to actually reference any of the meta-classes that inherit from DiagnosticAbstractDataIdentifier.

Class	**DiagnosticReadDataByIdentifier**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier			
Note	This represents an instance of the "Read Data by Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDataByIdentifiers			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticDataByIdentifier, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
readClass	DiagnosticReadDataByIdentifierClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticReadDataByIdentifier in the given context.

Table 5.20: DiagnosticReadDataByIdentifier

Class	**DiagnosticWriteDataByIdentifier**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier			
Note	This represents an instance of the "Write Data by Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDataByIdentifiers			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticDataByIdentifier, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
writeClass	DiagnosticWriteDataByIdentifierClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticWriteDataByIdentifier in the given context.

Table 5.21: DiagnosticWriteDataByIdentifier

Class	DiagnosticWriteDataByIdentifierClass			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier			
Note	This meta-class contains attributes shared by all instances of the "Write Data by Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDataByIdentifiers			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
–	–	–	–	–

Table 5.22: DiagnosticWriteDataByIdentifierClass

Class	DiagnosticDataByIdentifier (abstract)			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier			
Note	This represents an abstract base class for all diagnostic services that access data by identifier.			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
dataIdentifier	DiagnosticAbstractDataIdentifier	1	ref	This represents the linked DiagnosticDataIdentifier.

Table 5.23: DiagnosticDataByIdentifier

The modeling of `DiagnosticDataByIdentifier` represents concrete instances of diagnostic services within a `DiagnosticExtract`. However, there are attributes that are shared among all instances of `DiagnosticReadDataByIdentifier`.

For this purpose the dedicated service class `DiagnosticReadDataByIdentifierClass` has been introduced.

Class	DiagnosticReadDataByIdentifierClass			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier			
Note	This meta-class contains attributes shared by all instances of the "Read Data by Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDataByIdentifiers			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	**Type**	**Mul.**	**Kind**	**Note**
maxDidToRead	PositiveInteger	1	attr	This attribute represents the maximum number of allowed DIDs in a single instance of DiagnosticReadDataByIdentifier.

Table 5.24: DiagnosticReadDataByIdentifierClass

Please note that it is possible to create a reference to a concrete DiagnosticDataIdentifier from different DiagnosticServiceInstances.

[TPS_DEXT_01050] Consistency of DiagnosticServiceSwMapping with respect to data IDs ⌈ For each DiagnosticServiceSwMapping that references a DiagnosticValueNeeds and a DiagnosticDataByIdentifier, the value of DiagnosticValueNeeds.didNumber shall be ignored and the value of DiagnosticDataByIdentifier.dataIdentifier.id shall be taken instead. ⌋ *(RS_DEXT_00007, RS_DEXT_00013, RS_DEXT_00034, RS_DEXT_00052)*

Class	**DiagnosticValueNeeds**			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the general needs on the configuration of the Diagnostic Communication Manager (DCM) which are not related to a particular item (e.g. a PID). The main use case is the mapping of service ports to the DCM which are not related to a particular item. In the case of using a sender receiver communicated value, the related value shall be taken via assignedData in the role "signalBasedDiagnostics". In case of using a client/server communicated value, the related value shall be communicated via the port referenced by asssignedPort. The details of this communication (e.g. appropriate naming conventions) are specified in the related software specifications (SWS).			
Base	ARObject, DiagnosticCapabilityElement, Identifiable, MultilanguageReferrable, Referrable, ServiceNeeds			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataLength	PositiveInteger	0..1	attr	This attribute is applicable only if the ServiceNeed is aggregated within BswModuleDependency. This attribute represents the length of data (in bytes) provided for this particular PID signal.
diagnosticValueAccess	DiagnosticValueAccessEnum	0..1	attr	This attribute controls whether the data can be read and written or whether it is to be handled read-only.
didNumber	PositiveInteger	0..1	attr	This represents a Data identifier for the diagnostic value. This allows to predefine the DID number if the responsible function developer has received a particular requirement from the OEM or from a standardization body.
fixedLength	Boolean	0..1	attr	This attribute controls whether the data length of the data is fixed.
processingStyle	DiagnosticProcessingStyleEnum	0..1	attr	This attribute controls whether interaction requires the software-component to react synchronously on a request or whether it processes the request in background but still the DCM has to issue the call again to eventually obtain the result of the request.

Table 5.25: DiagnosticValueNeeds

5.5.2 IOControl

This chapter describes the modeling of diagnostic services `InputOutput Control` (0x2F). The purpose of this service is to provide the tester with the ability to override values exchanged with the AUTOSAR hardware abstraction.

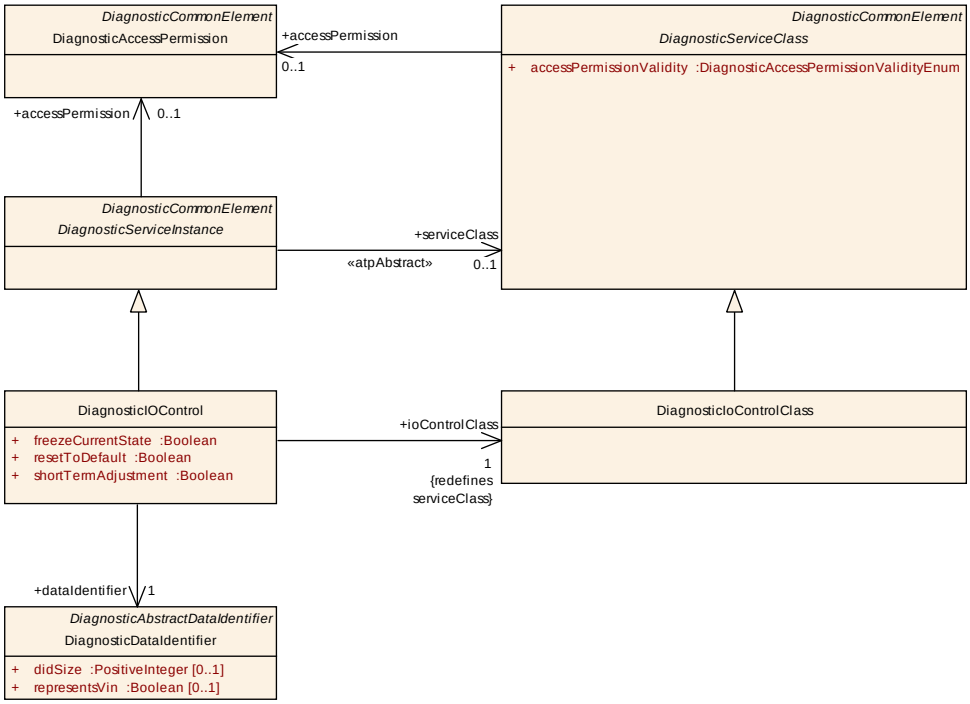

Figure 5.6: Modeling of diagnostic service `IOControl` (0x2F)

[TPS_DEXT_01015] Meaning of attributes of `DiagnosticIOControl` ⌈ The attributes `freezeCurrentState`, `resetToDefault`, and `shortTermAdjustment` represent the capabilities of the server rather than a concrete request message. ⌋ *(RS_DEXT_00014)*

[TPS_DEXT_01016] The capability returnControlToEcu ⌈ According to the statement made by [TPS_DEXT_01015], there is no formal means provided to configure the capability to execute *returnControlToEcu*. This lack of configuration is intentional because the capability is always available and cannot be revoked anyway. ⌋ *(RS_DEXT_00014)*

[TPS_DEXT_01017] Meaning of `DiagnosticIOControl.dataIdentifier` ⌈ The `DiagnosticIOControl.dataIdentifier` is taken for specifying the payload for the service.

However, in some cases `dataIdentifier` models the payload of the request message (`DiagnosticIOControl.shortTermAdjustment` is set to true) and in some cases it represents the payload of the response message. ⌋*(RS_DEXT_00014, RS_DEXT_00034)*

Please note that the referenced `dataIdentifier` itself may aggregate several `DiagnosticDataElement`s.

At run-time, only some DiagnosticDataElements may be relevant for a specific execution of the service InputOutput Control. For this purpose, the diagnostic message contains the so-called ControlEnableMaskRecord (for more information, please refer to [SWS_DCM_00581]).

[TPS_DEXT_01089] Definition of an *identifier* of a DiagnosticIOControl ⌈ The *identifier* of a DiagnosticIOControl is defined by the attribute DiagnosticIO-Control.dataIdentifier.id. ⌋*(RS_DEXT_00037)*

Class	**DiagnosticIOControl**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::IOControl			
Note	This represents an instance of the "I/O Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticIoControls			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataIdentifier	DiagnosticDataIdentifier	1	ref	This represents the corresponding DiagnosticDataIdentifier
freezeCurrentState	Boolean	1	attr	Setting this attribute to true represents the ability of the Dcm to execute a freezeCurrentState.
ioControlClass	DiagnosticIoControlClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticIOControl in the given context.
resetToDefault	Boolean	1	attr	Setting this attribute to true represents the ability of the Dcm to execute a resetToDefault.
shortTermAdjustment	Boolean	1	attr	Setting this attribute to true represents the ability of the Dcm to execute a shortTermAdjustment.

Table 5.26: DiagnosticIOControl

Class	**DiagnosticIoControlClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::IOControl			
Note	This meta-class contains attributes shared by all instances of the "IO Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticIoControls			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.27: DiagnosticIoControlClass

[TPS_DEXT_01018] `InputOutput Control` does not define any sub-functions ⌈ The diagnostic service `InputOutput Control` does not define any sub-functions, therefore the value of `DiagnosticIOControl.category` does not need to be constrained. ⌋*(RS_DEXT_00014, RS_DEXT_00051)*

[TPS_DEXT_01051] Consistency of `DiagnosticServiceSwMapping` with respect to data IDs ⌈ For each `DiagnosticServiceSwMapping` that references a `DiagnosticIoControlNeeds` and a `DiagnosticIOControl`, the value of `DiagnosticIoControlNeeds.didNumber` shall be ignored and the value of `DiagnosticIOControl.dataIdentifier.id` shall be taken instead. ⌋*(RS_DEXT_00014, RS_DEXT_00052)*

Class	**DiagnosticIoControlNeeds**			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the general needs on the configuration of the Diagnostic Communication Manager (DCM) which are not related to a particular item (e.g. a PID). The main use case is the mapping of service ports to the Dcm which are not related to a particular item.			
Base	ARObject, DiagnosticCapabilityElement, Identifiable, MultilanguageReferrable, Referrable, ServiceNeeds			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
currentValue	DiagnosticValueNeeds	0..1	ref	Reference to the DiagnosticValueNeeds indicating the access to the current value via signalBasedDiagnostics.
didNumber	PositiveInteger	0..1	attr	This represents a Data identifier for the diagnostic value. This allows to predefine the DID number if the a function developer has received a particular requirement from the OEM or from a standardization body.
freezeCurrentStateSupported	Boolean	0..1	attr	This attribute determines, if the referenced port supports temporary freezing of I/O value.
resetToDefaultSupported	Boolean	0..1	attr	This represents a flag for the existence of the ResetToDefault operation in the service interface.
shortTermAdjustmentSupported	Boolean	0..1	attr	This attribute determines, if the referenced port supports temporarily setting of I/O value to a specific value provided by the diagnostic tester.

Table 5.28: DiagnosticIoControlNeeds

5.5.3 EcuReset

This chapter describes the modeling of diagnostic services `EcuReset` (0x11).

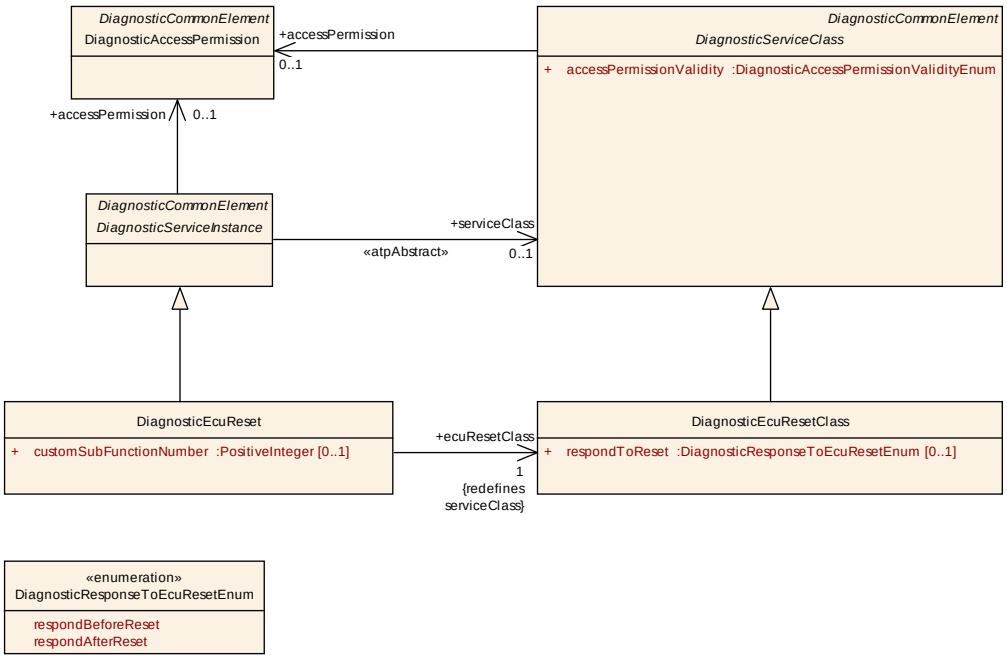

Figure 5.7: Modeling of diagnostic service `EcuReset` (0x11)

Class	**DiagnosticEcuReset**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EcuReset			
Note	This represents an instance of the "ECU Reset" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticEcuResets			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
customSubFunctionNumber	PositiveInteger	0..1	attr	This attribute shall be used to define a custom sub-function number if none of the standardized values of category shall be used.
ecuResetClass	DiagnosticEcuResetClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticEcuReset in the given context.

Table 5.29: DiagnosticEcuReset

Class	**DiagnosticEcuResetClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EcuReset			
Note	This meta-class contains attributes shared by all instances of the "Ecu Reset" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticEcuResets			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
respondToReset	DiagnosticResponseToEcuResetEnum	0..1	attr	This attribute defines whether the response to the EcuReset service shall be transmitted before or after the actual reset.

Table 5.30: DiagnosticEcuResetClass

Enumeration	**DiagnosticResponseToEcuResetEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EcuReset
Note	
Literal	***Description***
respondAfterReset	Answer to EcuReset service should come after the reset. **Tags:** atp.EnumerationValue=0
respondBeforeReset	Answer to EcuReset service should come before the reset. **Tags:** atp.EnumerationValue=1

Table 5.31: DiagnosticResponseToEcuResetEnum

Please note that (as already explained in section 5.5) the `SubFunctions` of this service are modeled by means of the `category` attribute.

[TPS_DEXT_01056] Applicable values for `DiagnosticEcuReset.category` ⌈ The following values of the attribute `DiagnosticEcuReset.category` are standardized by AUTOSAR:

- `HARD_RESET`
- `KEY_OFF_ON_RESET`
- `SOFT_RESET`
- `ENABLE_RAPID_POWER_SHUT_DOWN`
- `DISABLE_RAPID_POWER_SHUT_DOWN`

The meaning of these values is described in the applicable ISO document [16]. ⌋ *(RS_DEXT_00001, RS_DEXT_00004, RS_DEXT_00051)*

[TPS_DEXT_01019] Correspondence of `category` values to numerical values mentioned in the ISO 14229-1 ⌈ The ISO 14229-1 [16] standard document defines specific numerical values for the sub-functions of the diagnostic service `EcuReset`.

The correspondence of the numerical values to the pre-defined values of `category` according to [TPS_DEXT_01056] is pretty obvious because the definition of values defined in [TPS_DEXT_01056] has been directly inspired by the ISO 14229-1 [16] standard document. ⌋*(RS_DEXT_00001, RS_DEXT_00004, RS_DEXT_00051)*

[TPS_DEXT_01020] Manufacturer-specific values for sub-functions of service `EcuReset` ⌈ The ISO 14229-1 [16] standard document, beyond the standardized numerical values for sub-functions, reserves a numerical range of subFunction identifiers for manufacturer- or supplier-specific use.

In this case it is possible to define further values for `category`, provided that a custom prefix is used to avoid potential name clashes with further extensions of the AUTOSAR standard, namely [TPS_DEXT_01056]. ⌋*(RS_DEXT_00001, RS_DEXT_00004, RS_DEXT_00051)*

[TPS_DEXT_01021] Semantics of `DiagnosticEcuReset.customSubFunctionNumber` ⌈ The attribute `DiagnosticEcuReset.customSubFunctionNumber` has been introduced to allow for the specification of a manufacturer- or supplier-specific value to represent the custom sub-function in the diagnostic communication.

The tuple created by the the values of attributes `DiagnosticEcuReset.category` and `DiagnosticEcuReset.customSubFunctionNumber` fully specifies identification of the manufacturer- or supplier-specific sub-function. ⌋*(RS_DEXT_00004, RS_DEXT_00047, RS_DEXT_00051)*

[constr_1331] Existence of `DiagnosticEcuReset.customSubFunctionNumber` ⌈ The attribute `DiagnosticEcuReset.customSubFunctionNumber` shall only exist if the value of `DiagnosticEcuReset.category` is outside the standardized set of values as defined by [TPS_DEXT_01056]. ⌋*()*

[constr_1332] Value range for `DiagnosticEcuReset.customSubFunctionNumber` ⌈ The allowed value for `DiagnosticEcuReset.customSubFunctionNumber` shall always be within the closed interval **0x40** .. **0x7E**. ⌋*()*

5.5.4 ClearDiagnosticInformation

This chapter describes the modeling of diagnostic services `ClearDiagnosticInformation` (0x14). As the name suggests, the purpose of the service is to clear diagnostic information in the AUTOSAR diagnostics stack.

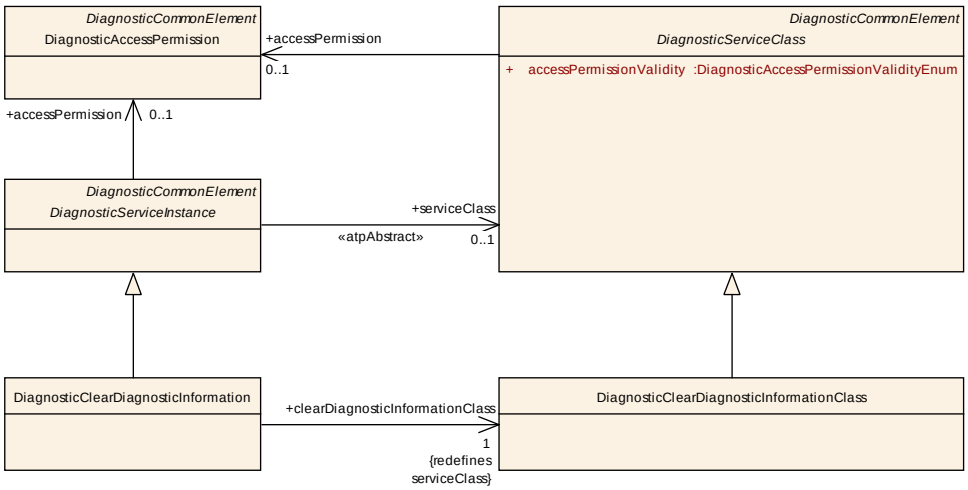

Figure 5.8: Modeling of diagnostic service `ClearDiagnosticInformation` (0x14)

Please note that there is nothing to configure for `DiagnosticClearDiagnosticInformation` beyond its mere existence.

Class	**DiagnosticClearDiagnosticInformation**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ClearDiagnosticInfo			
Note	This represents an instance of the "Clear Diagnostic Information" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticClearDiagnosticInformations			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
clearDiagnosticInformationClass	DiagnosticClearDiagnosticInformationClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticClearDiagnosticInformation in the given context.

Table 5.32: DiagnosticClearDiagnosticInformation

Class	**DiagnosticClearDiagnosticInformationClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ClearDiagnosticInfo			
Note	This meta-class contains attributes shared by all instances of the "Clear Diagnostic Information" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticClearDiagnosticInformations			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

–	–	–	–	–

Table 5.33: DiagnosticClearDiagnosticInformationClass

[TPS_DEXT_01022] `ClearDiagnosticInformation` does not define any sub-functions ⌈ The diagnostic service `ClearDiagnosticInformation` does not define any sub-functions, therefore the value of `DiagnosticClearDiagnosticInformation.category` does not need to be constrained. ⌋*(RS_DEXT_00001, RS_DEXT_00005, RS_DEXT_00051)*

5.5.5 Memory Services

This chapter describes the modeling of diagnostic services for memory access (0x23, 0x3D, 0x34-0x37). The purpose of these services is to access memory on the diagnostic stack on request of the tester.

The service description for accessing memory for diagnostic purposes is modeled by the abstract meta-class `DiagnosticMemoryByAddress`. It is supposed to provide all model properties relevant for the memory access.

The description of memory access, to some extent, requires a formal description of the memory segments to take into account. For this purpose the meta-class `DiagnosticMemoryIdentifier` has been introduced and referenced by `DiagnosticMemoryAddressableRangeAccess` in the role `memoryRange`.

The intent of this modeling was not to provide a generic memory model but to allow for the specification of memory properties just as far as diagnostics is concerned.

The aggregation of `DiagnosticMemoryIdentifier` at `DiagnosticMemoryByAddress` may or may not be relevant for an OEM. However, there is certainly a use case for adding this information to a `DiagnosticExtract` that goes back from a tier-1 supplier to an OEM as sort of a documentation of the diagnostic configuration.

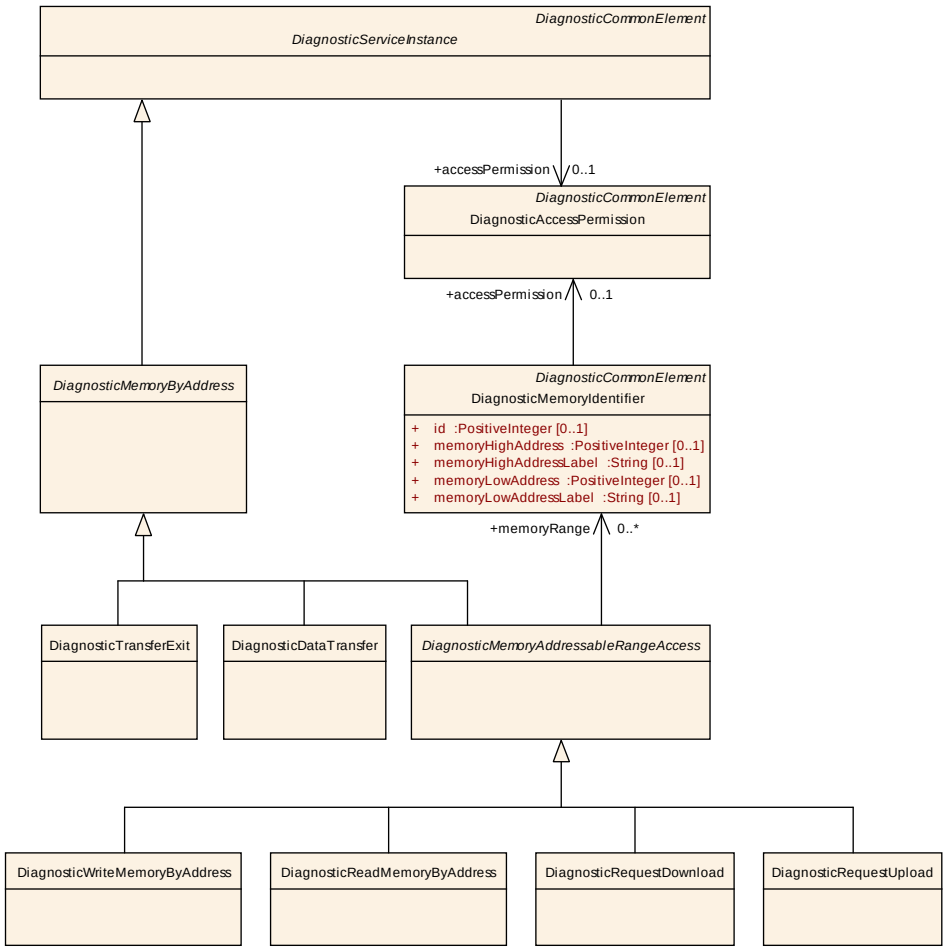

Figure 5.9: Modeling of diagnostic services `Memory` ((0x23, 0x3D, 0x34-0x37)

As `DiagnosticMemoryByAddress` represents a generic base class for all kinds of diagnostic memory access, it is also necessary to model the particular sub-classes that address specific use cases for diagnostic memory access.

These sub-classes are conceptually on the same level as other sub-classes of `DiagnosticServiceInstance`.

In other words, the case of memory access deviates from the modeling of other diagnostic services such that there is one further abstract base class involved.

[constr_1333] Existence of `DiagnosticMemoryIdentifier.memoryLowAddress` and `DiagnosticMemoryIdentifier.memoryHighAddress` ⌈ The attributes `DiagnosticMemoryIdentifier.memoryLowAddress` as well as `DiagnosticMemoryIdentifier.memoryHighAddress` shall not exist if the `DiagnosticMemoryIdentifier` referenced in the role `memoryRange` is referenced by a `DiagnosticRequestDownload` or a `DiagnosticRequestUpload`. ⌋*()*

[constr_1411] Existence of `DiagnosticMemoryIdentifier.memoryHighAddressLabel` vs. `DiagnosticMemoryIdentifier.memoryHighAddress` ⌈ At most **one** of the attributes in the following list shall exist:

- `DiagnosticMemoryIdentifier.memoryHighAddressLabel`

- DiagnosticMemoryIdentifier.memoryHighAddress

⌋()

[constr_1412] Existence of DiagnosticMemoryIdentifier.memoryLowAddressLabel vs. DiagnosticMemoryIdentifier.memoryLowAddress ⌈ At most **one** of the attributes in the following list shall exist:

- DiagnosticMemoryIdentifier.memoryLowAddressLabel
- DiagnosticMemoryIdentifier.memoryLowAddress

⌋()

Please note that it does not make sense to describe a memory address in this context **both** numerically **and** symbolically. If the address is described at all (see [constr_1333]) then it shall be done **either** symbolically or numerically. This is the motivation of the existence of [constr_1411] and [constr_1412].

Class	**DiagnosticMemoryByAddress (abstract)**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
Note	This represents an abstract base class for diagnostic services that deal with accessing memory by address.			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.34: DiagnosticMemoryByAddress

Class	**DiagnosticMemoryAddressableRangeAccess (abstract)**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
Note	This abstract base class			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMemoryByAddress, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
memoryRange	DiagnosticMemoryIdentifier	*	ref	This represents the formal description of the memory segment to which the DiagnosticMemoryByAddress applies.

Table 5.35: DiagnosticMemoryAddressableRangeAccess

Class	**DiagnosticMemoryIdentifier**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
Note	This meta-class represents the ability to define memory properties from the diagnostics point of view. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
accessPermission	DiagnosticAccessPermission	0..1	ref	This represents that access permission defined for the specific DiagnosticMemoryIdentifier.
id	PositiveInteger	0..1	attr	This represents the identification of the memory segment.
memoryHighAddress	PositiveInteger	0..1	attr	This represents the upper bound for addresses of the memory segment.
memoryHighAddressLabel	String	0..1	attr	This represents a symbolic label for the upper bound for addresses of the memory segment.
memoryLowAddress	PositiveInteger	0..1	attr	This represents the lower bound for addresses of the memory segment.
memoryLowAddressLabel	String	0..1	attr	This represents a symbolic label for the lower bound for addresses of the memory segment.

Table 5.36: DiagnosticMemoryIdentifier

Class	**DiagnosticWriteMemoryByAddress**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
Note	This represents an instance of the "Write Memory by Address" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMemoryAddressableRangeAccess, DiagnosticMemoryByAddress, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
writeClass	DiagnosticWriteMemoryByAddressClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticWritememoryByAddress in the given context.

Table 5.37: DiagnosticWriteMemoryByAddress

Class	**DiagnosticWriteMemoryByAddressClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
Note	This meta-class contains attributes shared by all instances of the "Write Memory by Address" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.38: DiagnosticWriteMemoryByAddressClass

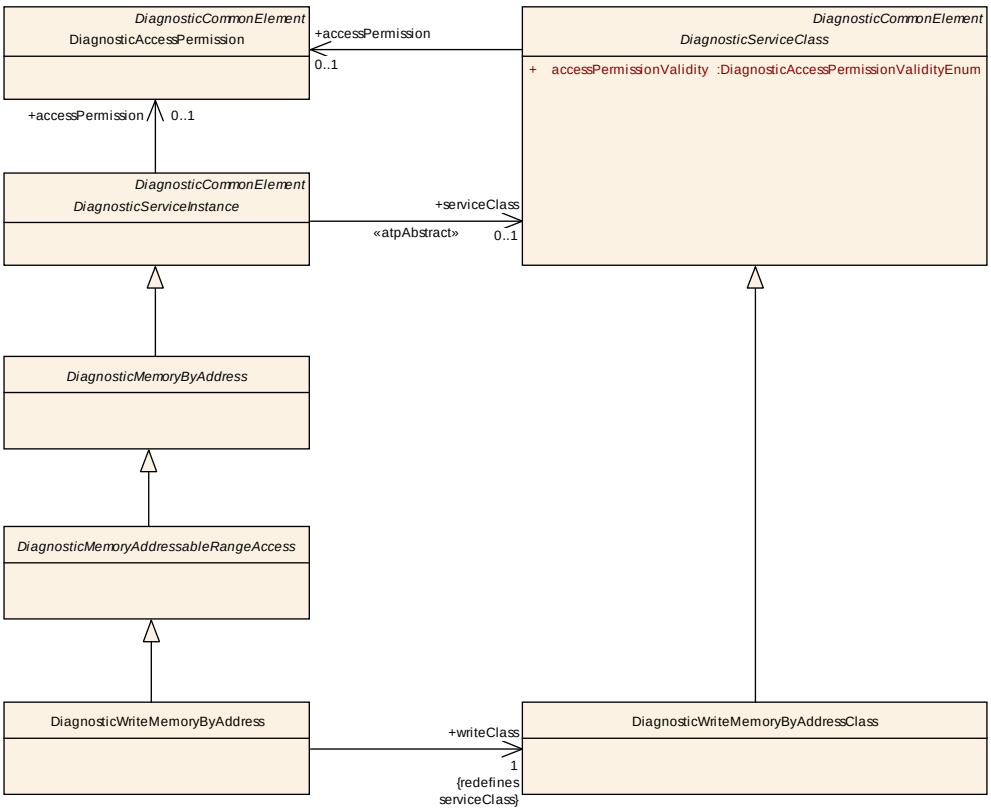

Figure 5.10: Modeling of diagnostic service `WriteMemoryByAddress` (0x3D)

[TPS_DEXT_01023] `WriteMemoryByAddress` does not define any sub-functions
⌈ The diagnostic service `WriteMemoryByAddress` does not define any sub-functions, therefore the value of `DiagnosticWriteMemoryByAddress.category` does not need to be constrained. ⌋*(RS_DEXT_00001, RS_DEXT_00020, RS_DEXT_00051)*

Class	**DiagnosticReadMemoryByAddress**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
Note	This represents an instance of the "Read Memory by Address" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMemoryAddressableRangeAccess, DiagnosticMemoryByAddress, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
readClass	DiagnosticReadMemoryByAddressClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticReadMemoryByAddresst in the given context.

Table 5.39: DiagnosticReadMemoryByAddress

Class	**DiagnosticReadMemoryByAddressClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
Note	This meta-class contains attributes shared by all instances of the "Read Memory by Address" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.40: DiagnosticReadMemoryByAddressClass

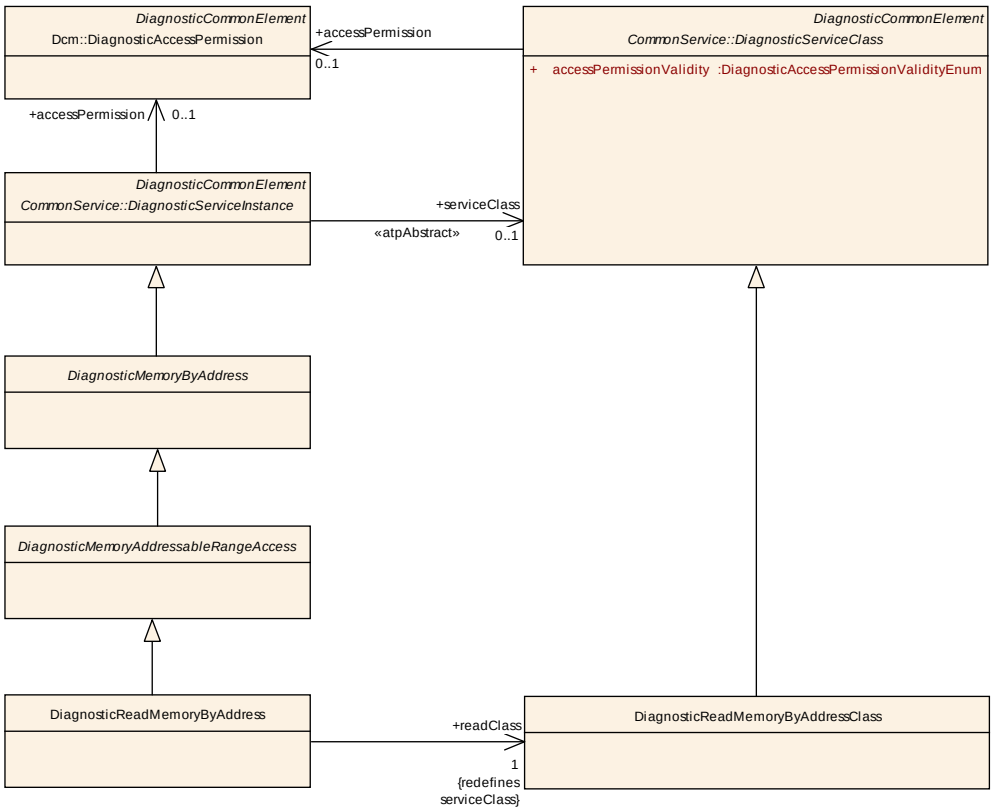

Figure 5.11: Modeling of diagnostic service `ReadMemoryByAddress` (0x23)

[TPS_DEXT_01024] `ReadMemoryByAddress` does not define any sub-functions ⌈ The diagnostic service `ReadMemoryByAddress` does not define any sub-functions, therefore the value of `DiagnosticReadMemoryByAddress.category` does not need to be constrained. ⌋*(RS_DEXT_00001, RS_DEXT_00008, RS_DEXT_00051)*

Class	**DiagnosticTransferExit**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
Note	This represents an instance of the "Transfer Exit" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMemoryByAddress, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
transferExitClass	DiagnosticTransferExitClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticTransferExit in the given context.

Table 5.41: DiagnosticTransferExit

Class	**DiagnosticTransferExitClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
Note	This meta-class contains attributes shared by all instances of the "Transfer Exit" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.42: DiagnosticTransferExitClass

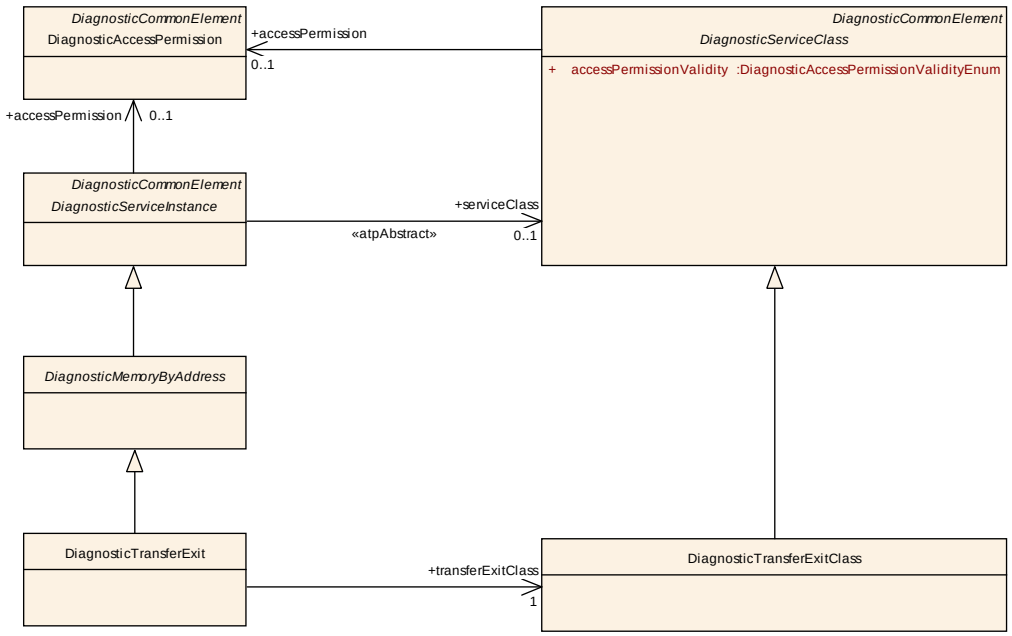

Figure 5.12: Modeling of diagnostic service `TransferExit` (0x37)

[TPS_DEXT_01025] `TransferExit` does not define any sub-functions ⌈ The diagnostic service `TransferExit` does not define any sub-functions, therefore the value of `DiagnosticTransferExit.category` does not need to be constrained. ⌋*(RS_DEXT_00001, RS_DEXT_00019, RS_DEXT_00051)*

Class	**DiagnosticDataTransfer**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
Note	This represents an instance of the "Data Transfer" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMemoryByAddress, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

dataTransferClass	DiagnosticDataTransferClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticDataTransfer in the given context.

Table 5.43: DiagnosticDataTransfer

Class	**DiagnosticDataTransferClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
Note	This meta-class contains attributes shared by all instances of the "Data Transfer" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.44: DiagnosticDataTransferClass

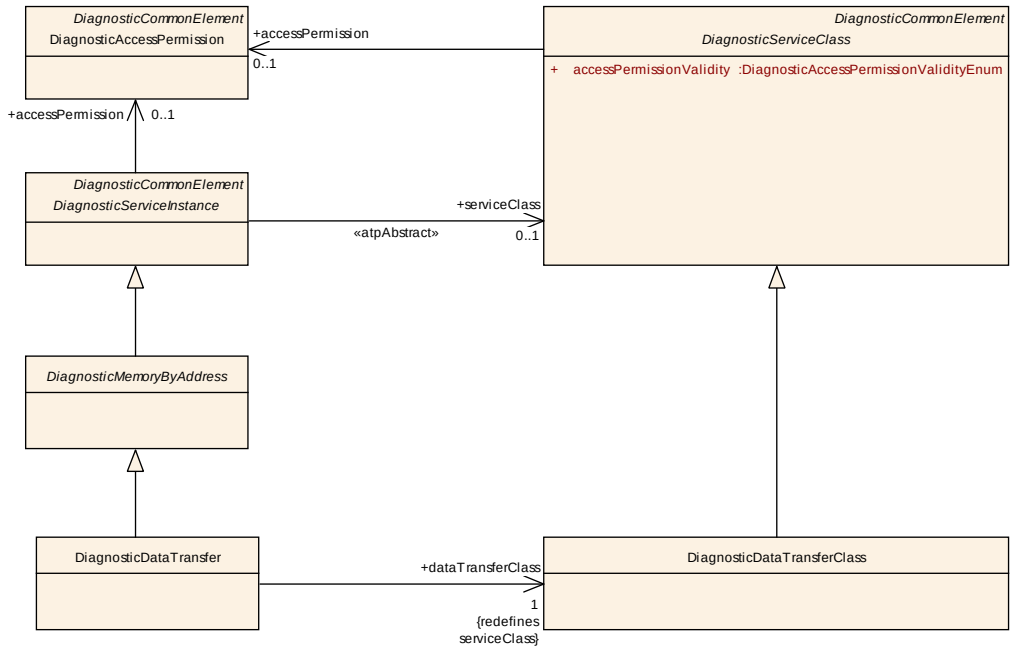

Figure 5.13: Modeling of diagnostic service `DataTransfer` (0x36)

[TPS_DEXT_01026] `DataTransfer` does not define any sub-functions ⌈ The diagnostic service `DataTransfer` does not define any sub-functions, therefore the value of `DiagnosticDataTransfer.category` does not need to be constrained. ⌋*(RS_DEXT_00001, RS_DEXT_00018, RS_DEXT_00051)*

Class	**DiagnosticRequestDownload**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
Note	This represents an instance of the "Request Download" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMemoryAddressableRangeAccess, DiagnosticMemoryByAddress, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
requestDownloadClass	DiagnosticRequestDownloadClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestDownload in the given context.

Table 5.45: DiagnosticRequestDownload

Class	**DiagnosticRequestDownloadClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
Note	This meta-class contains attributes shared by all instances of the "Request Download" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.46: DiagnosticRequestDownloadClass

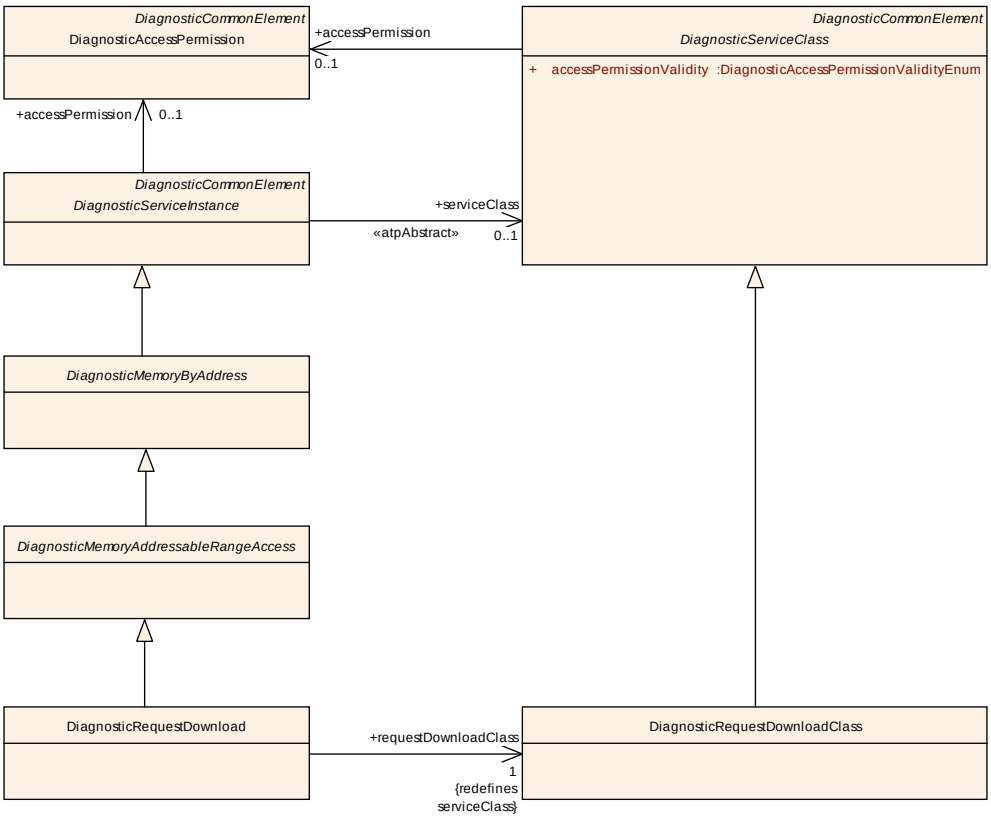

Figure 5.14: Modeling of diagnostic service `RequestDownload` (0x34)

[TPS_DEXT_01027] `RequestDownload` does not define any sub-functions ⌈ The diagnostic service `RequestDownload` does not define any sub-functions, therefore the value of `DiagnosticRequestDownload.category` does not need to be constrained. ⌋*(RS_DEXT_00001, RS_DEXT_00016, RS_DEXT_00051)*

Class	**DiagnosticRequestUpload**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress			
Note	This represents an instance of the "Request Upload" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMemoryAddressableRangeAccess, DiagnosticMemoryByAddress, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
requestUploadClass	DiagnosticRequestUploadClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestUpload in the given context.

Table 5.47: DiagnosticRequestUpload

Class	**DiagnosticRequestUploadClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::MemoryBy Address			
Note	This meta-class contains attributes shared by all instances of the "Request Upload" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticMemoryByAdresss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Diagnostic ServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.48: DiagnosticRequestUploadClass

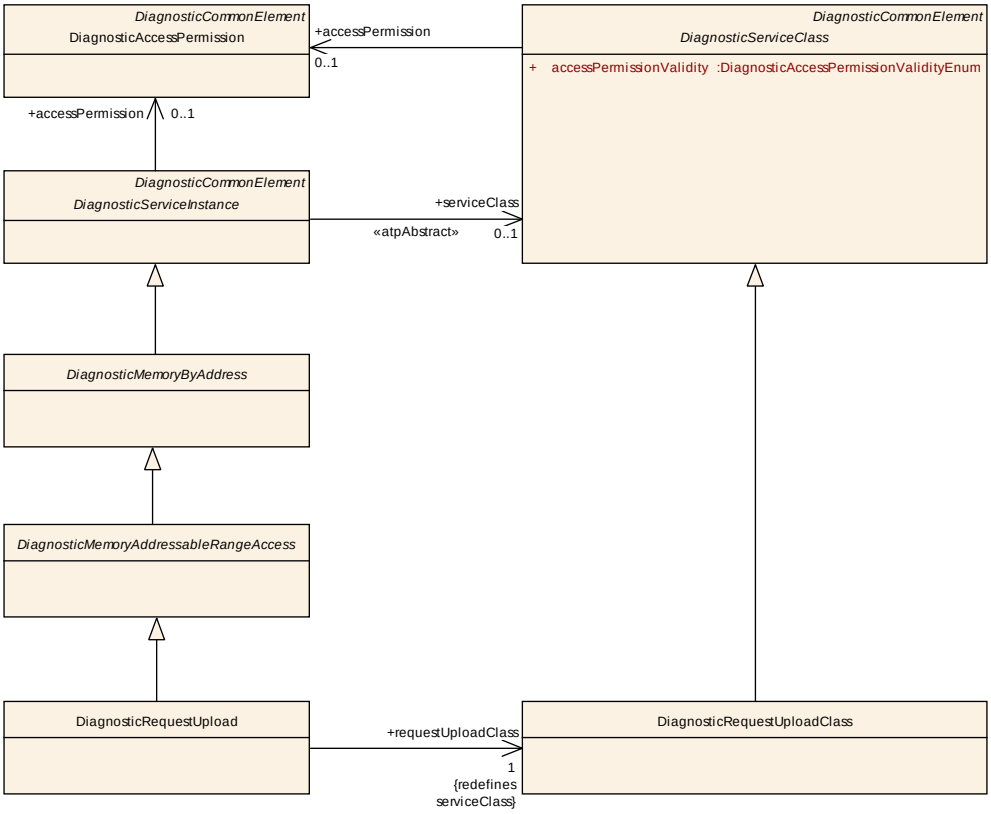

Figure 5.15: Modeling of diagnostic service `RequestDownload` (0x35)

[TPS_DEXT_01028] `RequestUpload` does not define any sub-functions ⌈ The diagnostic service `RequestUpload` does not define any sub-functions, therefore the value of `DiagnosticRequestUpload.category` does not need to be constrained. ⌋*(RS_DEXT_00001, RS_DEXT_00017, RS_DEXT_00051)*

5.5.6 CommunicationControl

This chapter describes the modeling of diagnostic services `CommunicationControl` (0x28). The purpose of this diagnostic service is to enable or disable `ISignalIPduGroup`s.

However, the actual implementation of the enabling or disabling is obviously not executed directly within the diagnostic stack. It requires some interaction with the BswM that in turn implements the enabling algorithm.

Therefore, the meta-class modeled for this purpose does not need to refer to `ISignalIPduGroup`s but implements a mode request to the BswM.

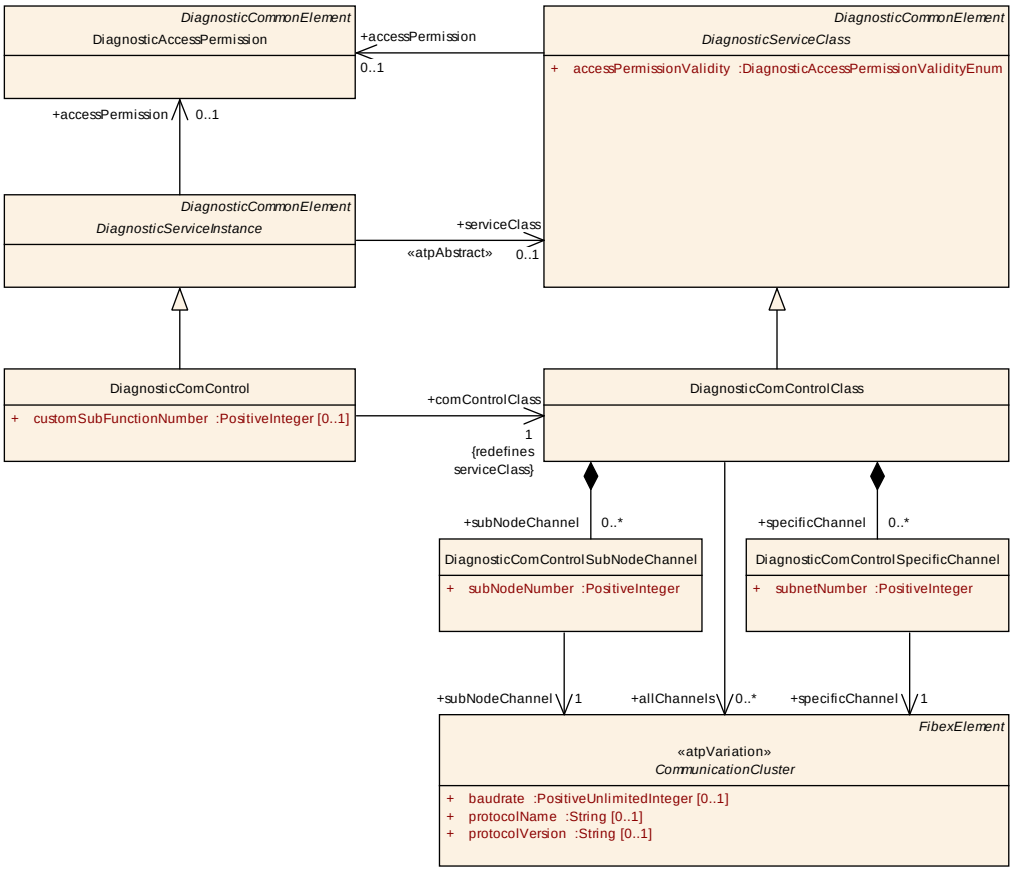

Figure 5.16: Modeling of diagnostic service `CommunicationControl` (0x28)

[TPS_DEXT_01057] Allowed values of `DiagnosticComControl.category` ⌈ The sub-functions of the diagnostic services `CommunicationControl` are identified by means of the attribute `DiagnosticComControl.category`. Standardized values of `DiagnosticComControl.category` are:

- `ENABLE_RX_AND_TX`
- `DISABLE_RX_AND_TX`
- `ENABLE_RX_AND_DISABLE_TX`

- DISABLE_RX_AND_ENABLE_TX
- ENABLE_RX_AND_DISABLE_TX_WITH_ENHANCED_ADDRESS_INFORMATION
- ENABLE_RX_AND_TX_WITH_ENHANCED_ADDRESS_INFORMATION

The meaning of these values is described in the applicable ISO document [16]. ⌋ *(RS_DEXT_00001, RS_DEXT_00010, RS_DEXT_00051)*

[TPS_DEXT_01029] Correspondence of `category` values to numerical values mentioned in the ISO 14229-1 ⌈ The ISO 14229-1 [16] standard document defines specific numerical values for the sub-functions of the diagnostic service CommunicationControl.

The correspondence of the numerical values to the pre-defined values of category according to [TPS_DEXT_01057] is pretty obvious because the definition of values defined in [TPS_DEXT_01057] has been directly inspired by the ISO 14229-1 [16] standard document. ⌋*(RS_DEXT_00001, RS_DEXT_00010, RS_DEXT_00051)*

[TPS_DEXT_01030] Manufacturer-specific values for sub-functions of service `CommunicationControl` ⌈ The ISO 14229-1 [16] standard document, beyond the standardized numerical values for sub-functions, reserves a numerical range of sub-Function identifiers for manufacturer-specific use.

In this case it is possible to define further values for category, provided that a custom prefix is used to avoid potential name clashes with further extensions of the AUTOSAR standard, namely [TPS_DEXT_01057]. ⌋*(RS_DEXT_00010, RS_DEXT_00047, RS_DEXT_00051)*

[TPS_DEXT_01031] Semantics of `DiagnosticComControl.customSubFunctionNumber` ⌈ The attribute DiagnosticComControl.customSubFunctionNumber has been introduced to allow for the specification of a manufacturer-or supplier-specific value to represent the custom sub-function in the diagnostic communication.

The tuple created by the values of attributes DiagnosticComControl.category and DiagnosticComControl.customSubFunctionNumber fully specifies identification of the manufacturer- or supplier-specific sub-function. ⌋*(RS_DEXT_00010, RS_DEXT_00047, RS_DEXT_00051)*

[constr_1334] Existence of `DiagnosticComControl.customSubFunctionNumber` ⌈ The attribute DiagnosticComControl.customSubFunctionNumber shall only exist if the value of DiagnosticComControl.category is outside the standardized set of values as defined by [TPS_DEXT_01057]. ⌋*()*

[constr_1335] Possible values for `DiagnosticComControl.customSubFunctionNumber` ⌈ Given the fulfillment of [constr_1334], the value of a given DiagnosticComControl.customSubFunctionNumber shall always be within the closed interval **0x40** .. **0x5F** (for manufacturer-specific sub-functions) or the closed interval **0x60** .. **0x7E** (for supplier-specific sub-functions). ⌋*()*

[TPS_DEXT_01032] Impact of the `DiagnosticComControlClass` on the state management for `CommunicationClusters` ⌈ The impact of the DiagnosticCom-

ControlClass on the state management for CommunicationClusters can have two alternative consequences:

- All CommunicationClusters are affected. For this purpose the attribute allChannels has the ability to identify the applicable CommunicationClusters.

  It may seem counterintuitive to require a reference to all applicable CommunicationClusters when the expected semantics is actually to define an impact on **all** of them.

  However, there could be private CommunicationClusters that are not participating in the diagnostics work-flow: These need to be kept out of scope and therefore the explicit identification of applicable CommunicationClusters makes sense.

- A selected number of CommunicationClusters is affected. This is conceptually different from the other use case in that it requires an additional attribute that keeps a subnetNumber that is typically assigned by the OEM role.

⌋(*RS_DEXT_00010*)

Class	≪**atpVariation**≫ **CommunicationCluster (abstract)**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreTopology			
Note	The CommunicationCluster is the main element to describe the topological connection of communicating ECUs. A cluster describes the ensemble of ECUs, which are linked by a communication medium of arbitrary topology (bus, star, ring, ...). The nodes within the cluster share the same communication protocol, which may be event-triggered, time-triggered or a combination of both. A CommunicationCluster aggregates one or more physical channels. **Tags:** vh.latestBindingTime=postBuild			
Base	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
baudrate	PositiveUnlimitedInteger	0..1	attr	Channels speed in bits/s.
physicalChannel	PhysicalChannel	1..*	aggr	This relationship defines which channel element belongs to which cluster. A channel must be assigned to exactly one cluster, whereas a cluster may have one or more channels. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
protocolName	String	0..1	attr	The name of the protocol used.

protocolVersion	String	0..1	attr	The version of the protocol used.

Table 5.49: CommunicationCluster

[constr_1336] Applicable value range for DiagnosticComControlSpecificChannel.subnetNumber ⌈ The value of attribute DiagnosticComControlSpecificChannel.subnetNumber shall be within the closed interval **1** .. **14**. ⌋*()*

Please note that the regulation implied by [constr_1336] has not been introduced on an arbitrary basis but gets its conceptual background from the ISO 14229-1 [16] standard document.

Obviously, a diagnostic service with the name CommunicationControl will have an impact on the enclosing ECU's mode management. This impact, however, is not defined by any further attributes or references, the DiagnosticComControl **is** the impact.

By defining a DiagnosticComControl and setting the category to one of the applicable values (e.g. ENABLE_RX_AND_TX), it is possible to express the intended semantics to the full extent.

[constr_1337] Allowed value range for attribute DiagnosticComControlSubNodeChannel.subNodeNumber ⌈ The value of attribute DiagnosticComControlSubNodeChannel.subNodeNumber shall not exceed the closed interval **0** .. **65535**. ⌋*()*

[TPS_DEXT_01074] Difference between the attributes DiagnosticComControlClass.specificChannel and DiagnosticComControlClass.subNodeChannel ⌈ The semantical difference between the attributes DiagnosticComControlClass.specificChannel and DiagnosticComControlClass.subNodeChannel is that DiagnosticComControlClass.specificChannel actually refers to a CommunicationCluster whereas DiagnosticComControlClass.subNodeChannel basically refers to a CommunicationCluster to which the nodes with the given identification numbers are connected. ⌋*(RS_DEXT_00010)*

Class	**DiagnosticComControl**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl			
Note	This represents an instance of the "Communication Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticCommunicationControls			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

comControlClass	DiagnosticComControlClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticComControl in the given context.
customSubFunctionNumber	PositiveInteger	0..1	attr	This attribute shall be used to define a custom sub-function number if none of the standardized values of category shall be used.

Table 5.50: DiagnosticComControl

Class	**DiagnosticComControlSpecificChannel**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl			
Note	This represents the ability to add further attributes to the definition of a specific channel that is subject to the diagnostic service "communication control".			
Base	ARObject			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
specificChannel	CommunicationCluster	1	ref	This represents the affected CommunicationClusters in the role specificChannel
subnetNumber	PositiveInteger	1	attr	This represents the applicable subnet number (which is an arbitrary number ranging from 1..14)

Table 5.51: DiagnosticComControlSpecificChannel

Class	**DiagnosticComControlClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl			
Note	This meta-class contains attributes shared by all instances of the "Communication Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticCommunicationControls			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
allChannels	CommunicationCluster	*	ref	This reference represents the semantics that all available channels shall be affected. It is still necessary to refer to individual CommunicatuionClusters because there could be private CommunicationClusters in the System Extract that are not subject to the service "communication control". By referring to the applicable CommunicationClusters it can be made sure that only the affected CommunicationClusters are accessed.

specificChannel	DiagnosticComControlSpecificChannel	*	aggr	This represents the ability to add additional attributes to the case that only specific channels are supposed to be considered,
subNodeChannel	DiagnosticComControlSubNodeChannel	*	aggr	This attribute represents the ability to add further attributes to the definition of a specific sub-node channel that is subject to the diagnostic service "communication control".

Table 5.52: DiagnosticComControlClass

Class	**DiagnosticComControlSubNodeChannel**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl			
Note	This represents the ability to add further attributes to the definition of a specific sub-node channel that is subject to the diagnostic service "communication control".			
Base	ARObject			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
subNodeChannel	CommunicationCluster	1	ref	This represents the affected CommunicationClusters in the role subNodeChannel
subNodeNumber	PositiveInteger	1	attr	This represents the applicable subNode number. The value corresponds to the request message parameter nodeIdentificationNumber of diagnostic service CommunicationControl (0x28).

Table 5.53: DiagnosticComControlSubNodeChannel

5.5.7 DynamicallyDefineDataIdentifier

This chapter describes the modeling of diagnostic services `DynamicallyDefineDataIdentifier` (0x2C). The purpose of the service is to allow for defining data identifiers (DID) at run-time.

By this means it is possible to combine existing diagnostic data into a single DID.

This semantics is reflected by the modeling of the meta-class `DiagnosticDynamicallyDefineDataIdentifier` that refers to a `DiagnosticDynamicDataIdentifier` in the role `dataIdentifier`.

Also, the `DiagnosticDynamicallyDefineDataIdentifier` inherits a reference to `accessPermission` from `DiagnosticServiceInstance`.

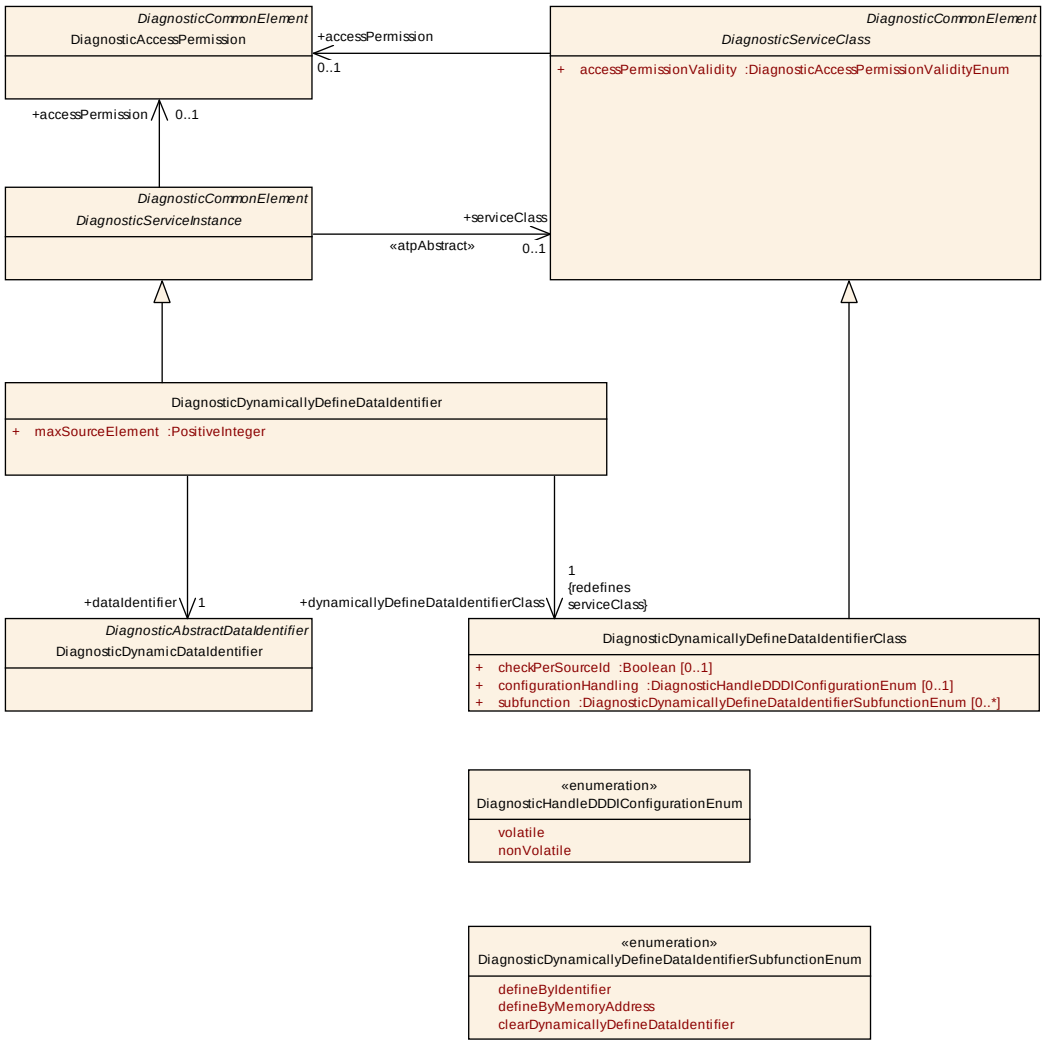

Figure 5.17: Modeling of diagnostic service `DynamicallyDefineDataIdentifier` (0x2C)

[constr_1421] Consistency of `DiagnosticDynamicallyDefineDataIdentifierClass.subfunction` ⌈ The values of `DiagnosticDynamicallyDefineDataIdentifierClass.subfunction` shall not repeat, i.e. every value of `DiagnosticDynamicallyDefineDataIdentifierSubfunctionEnum` shall at most appear once in the `subfunction` attribute. ⌋*()*

Class	**DiagnosticDynamicallyDefineDataIdentifier**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier			
Note	This represents an instance of the "Dynamically Define Data Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDynamicallyDefineDataIdentifiers			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

dataIdentifier	DiagnosticDynamicDataIdentifier	1	ref	This represents the applicable DiagnosticDynamicDataIdentfier.
dynamicallyDefineDataIdentifierClass	DiagnosticDynamicallyDefineDataIdentifierClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticDynamicallyDefineDataIdentifier in the given context.
maxSourceElement	PositiveInteger	1	attr	This represents the maximum number of source elements of the dynamically created DID.

Table 5.54: DiagnosticDynamicallyDefineDataIdentifier

Class	**DiagnosticDynamicallyDefineDataIdentifierClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier			
Note	This meta-class contains attributes shared by all instances of the "Dynamically Define Data Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticDynamicallyDefineDataIdentifiers			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
checkPerSourceId	Boolean	0..1	attr	If set to TRUE, the Dcm module shall check the session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF. If set to FALSE. the Dcm module shall not check the session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF.
configurationHandling	DiagnosticHandleDDDIConfigurationEnum	0..1	attr	This configuration switch defines whether DDDID definition is handled as non-volatile information or not.
subfunction	DiagnosticDynamicallyDefineDataIdentifierSubfunctionEnum	*	attr	This attribute contains a list of applicable subfunctions for all DiagnosticDynamicallyDefineDataIdentifier that reference the DiagnosticDynamicallyDefineDataIdentifierClass.

Table 5.55: DiagnosticDynamicallyDefineDataIdentifierClass

Enumeration	**DiagnosticHandleDDDIConfigurationEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier
Note	This meta-class represents the options for controlling how the configuration of the DynamicallyDefineDataIdentifiers is done in the given context.

Literal	*Description*
nonVolatile	This indicates that the configuration of DynamicallyDefineDataIdentifier shall be stored as non-volatile data. **Tags:** atp.EnumerationValue=0
volatile	This indicates that the configuration of DynamicallyDefineDataIdentifier shall be handled as volatile data. **Tags:** atp.EnumerationValue=1

Table 5.56: DiagnosticHandleDDDIConfigurationEnum

Enumeration	**DiagnosticDynamicallyDefineDataIdentifierSubfunctionEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier
Note	This meta-class contains a list of possible subfunctions for the UDS service 0x2C.
Literal	***Description***
clearDynamicallyDefineDataIdentifier	Clear the specified dynamic data identifier. **Tags:** atp.EnumerationValue=0
defineByIdentifier	The definition of dynamic data identifier shall be done via a reference to a diagnostic data identifier. **Tags:** atp.EnumerationValue=1
defineByMemoryAddress	The definition of dynamic data identifier shall be done via a reference to a memory address. **Tags:** atp.EnumerationValue=2

Table 5.57: DiagnosticDynamicallyDefineDataIdentifierSubfunctionEnum

5.5.8 ReadDataByPeriodicIdentifier

This chapter describes the modeling of diagnostic services `ReadDataByPeriodicIdentifier` (0x2A).

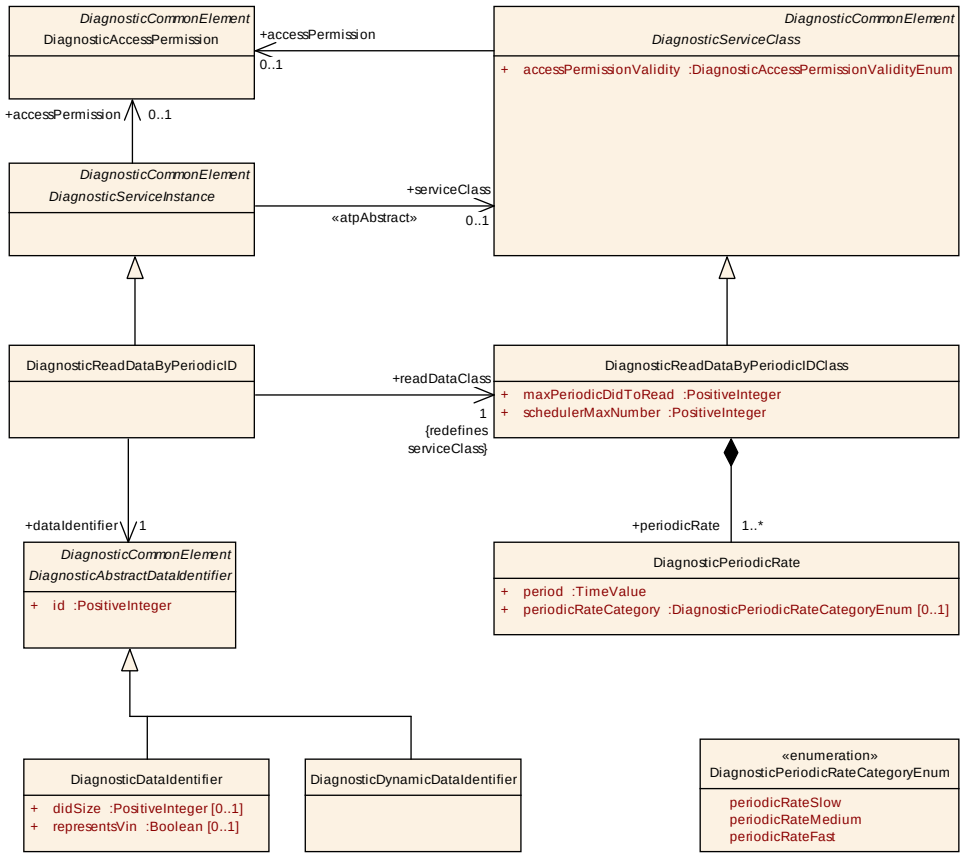

Figure 5.18: Modeling of diagnostic service `ReadDataByPeriodicIdentifier` (0x2A)

[constr_1338] Maximum number of aggregated `DiagnosticReadDataByPeriodicIDClass.periodicRate` ⌈ The number of aggregated `periodicRate` within the context of one `DiagnosticReadDataByPeriodicIDClass` shall be within the closed interval 1..3. ⌋*()*

Class	**DiagnosticReadDataByPeriodicID**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID			
Note	This represents an instance of the "Read Data by periodic Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticReadDataByPeriodicIds			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataIdentifier	DiagnosticAbstractDataIdentifier	1	ref	This represents the corresponding DiagnosticDataIdentifier.

readDataClass	DiagnosticReadDataByPeriodicIDClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the the reference represents the ability to access shared attributes among all DiagnosticReadDataByPeriodicID in the given context.

Table 5.58: DiagnosticReadDataByPeriodicID

Class	**DiagnosticReadDataByPeriodicIDClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID			
Note	This meta-class contains attributes shared by all instances of the "Read Data by periodic Identifier" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticReadDataByPeriodicIds			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
maxPeriodicDidToRead	PositiveInteger	1	attr	This represents the maximum number of data identifiers that can be included in one request.
periodicRate	DiagnosticPeriodicRate	1..*	aggr	This represents the description of a collection of periodic rates in which the service can be executed.
schedulerMaxNumber	PositiveInteger	1	attr	This represents the maximum number of periodic data identifiers that can be scheduled in parallel.

Table 5.59: DiagnosticReadDataByPeriodicIDClass

Class	**DiagnosticPeriodicRate**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID			
Note	This represents the ability to define a periodic rate for the specification of the "read data by periodic ID" diagnostic service.			
Base	ARObject			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
period	TimeValue	1	attr	This represents the period of the DiagnosticPeriodicRate in seconds.
periodicRateCategory	DiagnosticPeriodicRateCategoryEnum	0..1	attr	This attribute represents the category of the periodic rate.

Table 5.60: DiagnosticPeriodicRate

Enumeration	**DiagnosticPeriodicRateCategoryEnum**

Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID
Note	This meta-class provides possible values for the setting of the periodic rate.
Literal	***Description***
periodicRateFast	This value represents a fast periodic rate. **Tags:** atp.EnumerationValue=0
periodicRateMedium	This value represents a medium periodic rate. **Tags:** atp.EnumerationValue=1
periodicRateSlow	This value represents a slow periodic rate. **Tags:** atp.EnumerationValue=2

Table 5.61: DiagnosticPeriodicRateCategoryEnum

The meaning of the values defined in `DiagnosticPeriodicRateCategoryEnum` is described in the applicable ISO document [16].

5.5.9 ControlDTCSetting

This chapter describes the modeling of diagnostic services `ControlDTCSetting` (0x85). The purpose of the diagnostic service is to let the tester tell the diagnostic stack to either stop or resume the updating of a diagnostic trouble code.

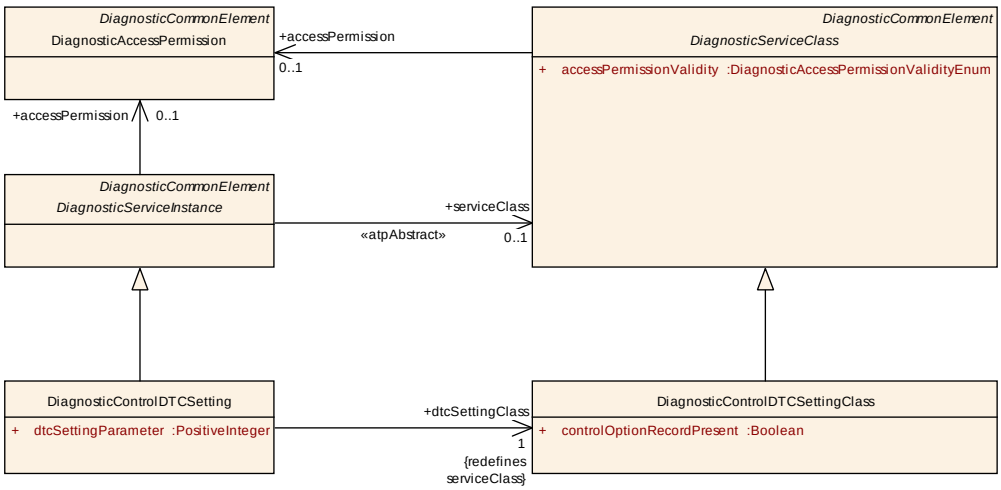

Figure 5.19: Modeling of diagnostic service `ControlDTCSetting` (0x85)

[TPS_DEXT_01075] standardized values for the attribute `DiagnosticControlDTCSetting.category` ⌈ AUTOSAR does not standardize any of the possible values for the attribute `DiagnosticControlDTCSetting.category`. ⌋ *(RS_DEXT_00001, RS_DEXT_00021, RS_DEXT_00051)*

[TPS_DEXT_01076] Identification of sub-functions of diagnostic service `ControlDTCSetting` ⌈ The identification of sub-functions (for which the attribute `cat-`

egory is used for several other DiagnosticServiceInstances) is done via the attribute DiagnosticControlDTCSetting.dtcSettingParameter.

⌋(*RS_DEXT_00021*, *RS_DEXT_00051*)

Class	**DiagnosticControlDTCSetting**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ControlDTCSetting			
Note	This represents an instance of the "Control DTC Setting" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticControlDtcSettings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dtcSettingClass	DiagnosticControlDTCSettingClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the the reference represents the ability to access shared attributes among all DiagnosticControlDTCSetting in the given context.
dtcSettingParameter	PositiveInteger	1	attr	This represents the DTCSettingType defined by ISO 14229-1. The pre-defined values are 1 (ON) and 2 (OFF).

Table 5.62: DiagnosticControlDTCSetting

Class	**DiagnosticControlDTCSettingClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ControlDTCSetting			
Note	This meta-class contains attributes shared by all instances of the "Control DTC Setting" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticControlDtcSettings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
controlOptionRecordPresent	Boolean	1	attr	This represents the decision whether the DTCSettingControlOptionRecord (see ISO 14229-1) is in general supported in the request message.

Table 5.63: DiagnosticControlDTCSettingClass

5.5.10 ResponseOnEvent

This chapter describes the modeling of diagnostic services ResponseOnEvent (0x86). The purpose of this service is to instruct the AUTOSAR diagnostic stack with respect to the starting or stopping of sending responses to a specific event to the tester.

Each DiagnosticResponseOnEvent provides the ability to define a collection of triggers (modelled by means of the abstract meta-class DiagnosticResponseOnEventTrigger) that cause the sending of a response message.

The actual trigger behavior is defined by the sub-class of DiagnosticResponseOnEventTrigger used to specify whether the trigger shall be created in response to a data change or in response to a DTC change.

[TPS_DEXT_01033] Semantics of triggers in the context of a DiagnosticResponseOnEvent ⌈ The semantics of a trigger in the context of a DiagnosticResponseOnEvent can be defined in two ways:

- The meta-class DiagnosticDataChangeTrigger allows for defining a trigger that activates on the change of the value of the referenced (in the role dataIdentifier) DiagnosticDataIdentifier.
- The meta-class DiagnosticDtcChangeTrigger allows for defining a trigger for the activation of the service. The entire proceedings of how the trigger activates and what DTCs are affected in managed at run-time and therefore no further configuration is required at this point.

⌋*(RS_DEXT_00022)*

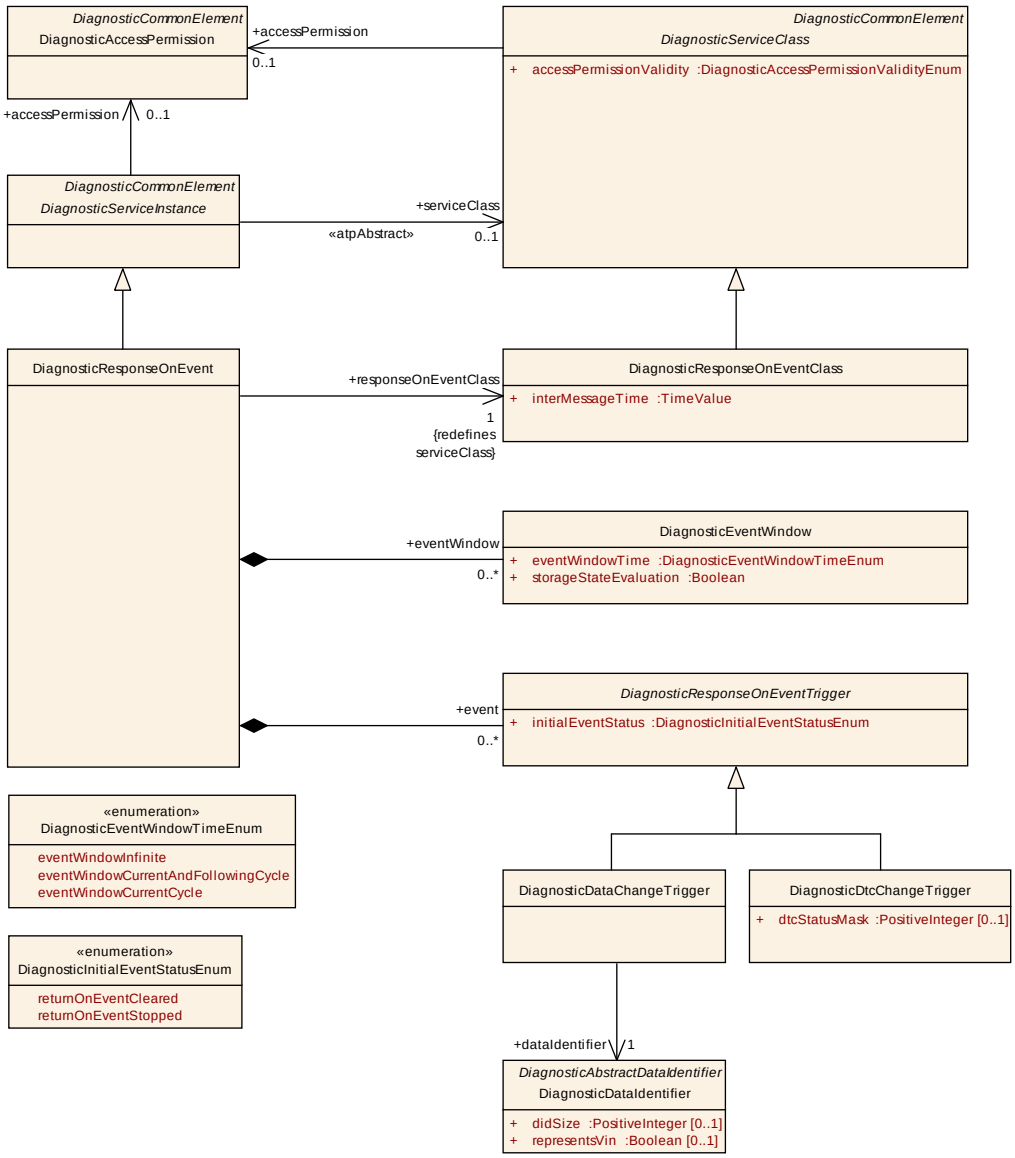

Figure 5.20: Modeling of diagnostic service `ResponseOnEvent` (0x86)

[constr_1365] Multiplicity of `DiagnosticResponseOnEvent.event` ⌈ The multiplicity of `DiagnosticResponseOnEvent.event` shall not exceed the upper bound **255**. ⌋*()*

[constr_1366] Event ID in the context of diagnostic service `ResponseOnEvent` shall be unique ⌈ The value of `DiagnosticResponseOnEvent.event.dataIdentifier.id` shall be unique within the context of a given `DiagnosticResponseOnEvent`. ⌋*()*

Class	**DiagnosticResponseOnEvent**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent			
Note	This represents an instance of the "Response on Event" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticResponseOnEvents			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
event	DiagnosticResponseOnEventTrigger	*	aggr	This represents the collection of DiagnosticResponseOnEventTriggers defined in the context of the enclosing DiagnosticResponseOnEvent.
eventWindow	DiagnosticEventWindow	*	aggr	This represents the applicable DiagnosticEventWindows
responseOnEventClass	DiagnosticResponseOnEventClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticResponseOnEvent in the given context.

Table 5.64: DiagnosticResponseOnEvent

Class	**DiagnosticResponseOnEventClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent			
Note	This represents the ability to define common properties for alle instances of the "Response on Event" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticResponseOnEvents			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
interMessageTime	TimeValue	1	attr	Provide the minimum time in seconds between two consecutive transmissions of an ROE event.

Table 5.65: DiagnosticResponseOnEventClass

Class	**DiagnosticEventWindow**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent			
Note	This represents the ability to define the characteristics of the applicable event window			
Base	ARObject			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
eventWindowTime	DiagnosticEventWindowTimeEnum	1	attr	This attribute clarifies the validity of the eventWindow

storageStateEvaluation	Boolean	1	attr	If this attribute is set to TRUE the StorageStateBit will be evaluated if this EventWindowTime is requested.

Table 5.66: DiagnosticEventWindow

Class	**DiagnosticResponseOnEventTrigger (abstract)**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent			
Note	This represents the ability to further specify the events that are associated with the execution of the diagnostic service.			
Base	ARObject			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
initialEventStatus	DiagnosticInitialEventStatusEnum	1	attr	This represents the initial status of the enclosing DiagnosticResponseOnEventTrigger.

Table 5.67: DiagnosticResponseOnEventTrigger

Class	**DiagnosticDataChangeTrigger**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent			
Note	This represents the ability to define a trigger based on the change of a given DiagnosticDataIdentifier.			
Base	ARObject, DiagnosticResponseOnEventTrigger			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataIdentifier	DiagnosticDataIdentifier	1	ref	This represents the corresponding DiagnosticDataIdentifier.

Table 5.68: DiagnosticDataChangeTrigger

Class	**DiagnosticDtcChangeTrigger**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent			
Note	This represents the ability to define a trigger that executes on the change of any DiagnosticTroubleCode.			
Base	ARObject, DiagnosticResponseOnEventTrigger			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dtcStatusMask	PositiveInteger	0..1	attr	This attribute represents the ability to define a status mask for the triggering of an ROE response on the change of a DTC.

Table 5.69: DiagnosticDtcChangeTrigger

Enumeration	**DiagnosticInitialEventStatusEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent

Note	This represents the ability to define an initial status for the ROE service.
Literal	***Description***
returnOn EventCleared	This means that the ResponseOnEvent is initially cleared. **Tags:** atp.EnumerationValue=0
return OnEvent Stopped	This means that the ResponseOnEvent is initially stopped. **Tags:** atp.EnumerationValue=1

Table 5.70: DiagnosticInitialEventStatusEnum

Enumeration	**DiagnosticEventWindowTimeEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent
Note	This represents the ability to define the semantics of the event window.
Literal	***Description***
eventWindow CurrentAnd Following Cycle	This means that the window extends to this and the following cycle. **Tags:** atp.EnumerationValue=0
eventWindow CurrentCycle	This means that the window is limited to the current cycle. **Tags:** atp.EnumerationValue=1
eventWindow Infinite	This means that the window extents without a border. **Tags:** atp.EnumerationValue=2

Table 5.71: DiagnosticEventWindowTimeEnum

5.5.11 ReadDTCInformation

This chapter describes the modeling of diagnostic services `ReadDTCInformation` (0x19). The purpose of this service is enable a tester to read a `Diagnostic Trouble Code` from the AUTOSAR `Dcm` [10] (that, in turn, fetches the information from the AUTOSAR `Dem` [11]).

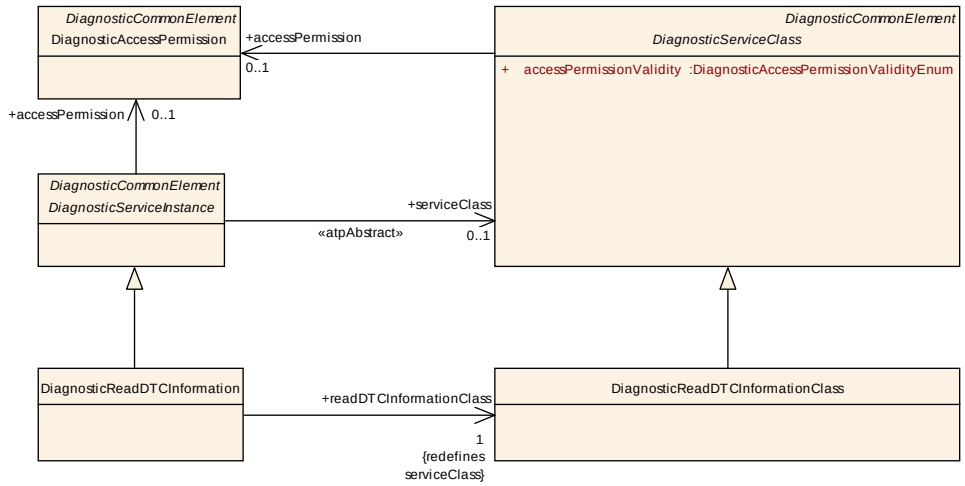

Figure 5.21: Modeling of diagnostic service `ReadDTCInformation` (0x19)

[TPS_DEXT_01034] Sub-functions of the service `ReadDTCInformation` ⌈ The diagnostics service `ReadDTCInformation` defines a number of sub-functions that are, as far as the modeling in AUTOSAR goes, identified by a textual identifier.

These sub-functions are **not** modeled explicitly but can be specified by defining a `DiagnosticReadDTCInformation` and by setting the `DiagnosticReadDTCInformation.category` to the identifier of the respective sub-function.

The possible values, as far as the AUTOSAR standard is concerned, are defined by [TPS_DEXT_01060]. ⌋*(RS_DEXT_00006, RS_DEXT_00051)*

[TPS_DEXT_01060] Applicable values for `DiagnosticReadDTCInformation.category` ⌈ The following values of the attribute `DiagnosticReadDTCInformation.category` are standardized by AUTOSAR:

- REPORT_NUMBER_OF_DTC_BY_STATUS_MASK
- REPORT_DTC_BY_STATUS_MASK
- REPORT_MIRROR_MEMORY_DTC_BY_STATUS_MASK
- REPORT_NUMBER_OF_MIRROR_MEMORY_DTC_BY_STATUS_MASK
- REPORT_NUMBER_OF_EMISSIONS_OBD_DTC_BY_STATUS_MASK
- REPORT_EMISSIONS_OBD_DTC_BY_STATUS_MASK
- REPORT_DTC_SNAPSHOT_IDENTIFICATION
- REPORT_DTC_SNAPSHOT_RECORD_BY_DTC_NUMBER
- REPORT_DTC_STORED_DATA_BY_RECORD_NUMBER
- REPORT_DTC_EXT_DATA_RECORD_BY_DTC_NUMBER
- REPORT_MIRROR_MEMORY_DTC_EXT_DATA_RECORD_BY_DTC_NUMBER
- REPORT_NUMBER_OF_DTC_BY_SEVERITY_MASK_RECORD

- REPORT_DTC_BY_SEVERITY_MASK_RECORD
- REPORT_SEVERITY_INFORMATION_OF_DTC
- REPORT_SUPPORTED_DTC
- REPORT_FIRST_TEST_FAILED_DTC
- REPORT_FIRST_CONFIRMED_DTC
- REPORT_MOST_RECENT_TEST_FAILED_DTC
- REPORT_MOST_RECENT_CONFIRMED_DTC
- REPORT_DTC_FAULT_DETECTION_COUNTER
- REPORT_DTC_WITH_PERMANENT_STATUS
- REPORT_USER_DEF_MEMORY_DTC_BY_STATUS_MASK
- REPORT_USER_DEF_MEMORY_DTC_SNAPSHOT_RECORD_BY_DTC_NUMBER
- REPORT_USER_DEF_MEMORY_DTC_EXT_DATA_RECORD_BY_DTC_NUMBER
- REPORT_WWH_OBD_DTC_BY_MASK_RECORD
- REPORT_WWH_OBD_DTC_WITH_PERMANENT_STATUS

The meanings of these values are described in the applicable ISO document (ISO 14229-1) [16]. ⌋(*RS_DEXT_00001*, *RS_DEXT_00006*, *RS_DEXT_00051*)

Please note that there is nothing to configure for DiagnosticReadDTCInformation beyond its mere existence.

Class	**DiagnosticReadDTCInformation**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::ReadDTCInformation			
Note	This represents an instance of the "Read DTC Information" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticReadDtcInformations			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
readDTCInformationClass	DiagnosticReadDTCInformationClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticReadDTCInformation in the given context.

Table 5.72: DiagnosticReadDTCInformation

5.5.12 RoutineControl

This chapter describes the modeling of diagnostic services `RoutineControl` (0x31). The purpose of this diagnostic service is to execute a piece of code, a `Diagnostic Routine`, on the diagnostic stack at the request of the tester.

`Diagnostic Routine`s consist of up to three possible components:

- `Start Routine`
- `Stop Routine`
- `Request Routine Results`

The impact of this architecture no the meta-model is described by [TPS_DEXT_01077]:

[TPS_DEXT_01077] Modeling of `DiagnosticRoutine` ⌈ From the meta-modeling point of view, the semantics of `DiagnosticRoutine` is created by aggregating three further meta-classes that represent subfunctions of service `RoutineControl` (0x31):

- `DiagnosticStartRoutine`
- `DiagnosticStopRoutine`
- `DiagnosticRequestRoutineResults`

⌋*(RS_DEXT_00015)*

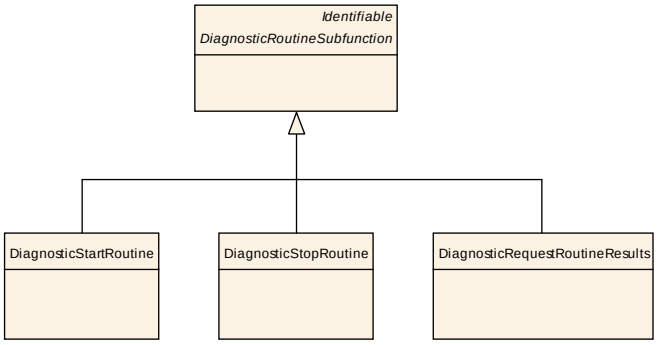

Figure 5.22: Abstract base class for subfunctions of diagnostic service `RoutineControl` (0x31)

Class	**DiagnosticRoutineSubfunction (abstract)**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class acts as an abstract base class to routine subfunctions.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
accessPermission	DiagnosticAccessPermission	0..1	ref	This reference represents the access permission of the owning routine subfunction.

Table 5.73: DiagnosticRoutineSubfunction

[TPS_DEXT_01088] Semantics of `DiagnosticRoutine.id` ⌈ The attribute `DiagnosticRoutine.id` represents the so-called *identifier* of the `DiagnosticRoutine`. ⌋*(RS_DEXT_00036)*

[TPS_DEXT_01078] Not possible to use the attribute `category` for the identification of the sub-function of diagnostic service `RoutineControl` ⌈ In the case of `DiagnosticRoutine`, it is not possible to use the attribute `category` for the identification of the sub-function. ⌋*(RS_DEXT_00015, RS_DEXT_00051)*

The sub-functions actually have different properties i.e. the arguments to a `DiagnosticRoutine`) that require a dedicated modeling for this purpose.

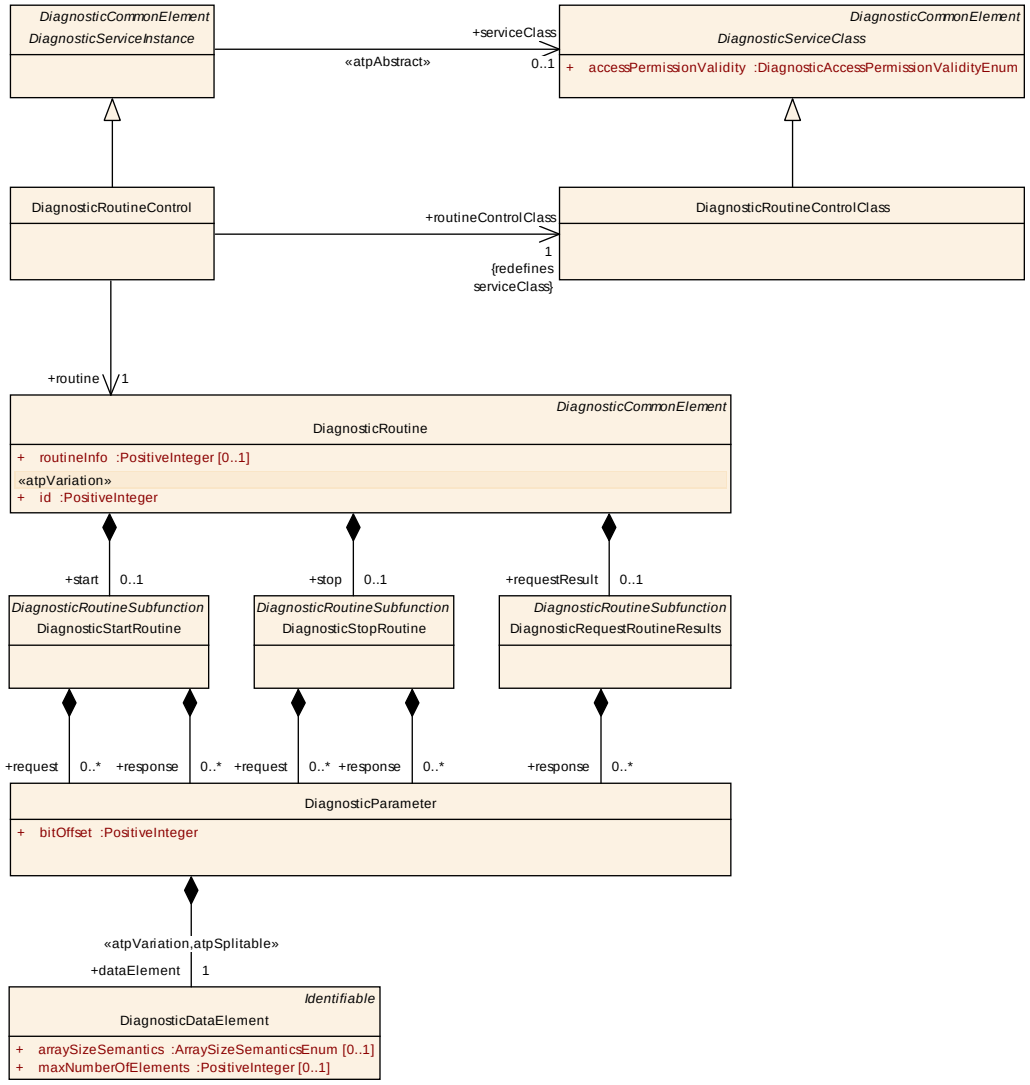

Figure 5.23: Modeling of diagnostic service `RoutineControl` (0x31)

[TPS_DEXT_01079] Modeling of the arguments to a `DiagnosticRoutine` ⌈ The arguments to a `DiagnosticRoutine` are modeled by means of `DiagnosticParameter` that is aggregated in the following roles:

- `DiagnosticStartRoutine.request`

- DiagnosticStartRoutine.response
- DiagnosticStopRoutine.request
- DiagnosticStopRoutine.response
- DiagnosticRequestRoutineResults.response

⌋(*RS_DEXT_00015*)

A DiagnosticParameter, in turn, aggregates a DiagnosticDataElement (see section 4.2) in the role dataElement.

[TPS_DEXT_01080] Diagnostic Routine needs to be started ⌈ ISO 14229-1 [16] does not foresee the existence of a Diagnostic Routine that is already executing at boot time. Therefore, a Diagnostic Routine needs to be started at some point in order to make sense of it. ⌋(*RS_DEXT_00015*)

[constr_1339] Existence of DiagnosticRoutine.start ⌈ In a complete DiagnosticExtract, the attribute DiagnosticRoutine.start shall always exist for any given DiagnosticRoutine. ⌋()

[TPS_DEXT_01035] Existence of DiagnosticRoutine.stop and DiagnosticRoutine.requestResult ⌈ In contrast to DiagnosticRoutine.start (as clarified by [constr_1339]), the existence of DiagnosticRoutine.stop and DiagnosticRoutine.requestResult is truly optional. ⌋(*RS_DEXT_00015*)

[constr_1340] Consistency of DiagnosticServiceSwMapping with respect to synchronously called DiagnosticRoutines ⌈ Each DiagnosticServiceSwMapping that references a DiagnosticRoutineControl that only aggregates a DiagnosticStartRoutine in the role start shall only reference a SwcServiceDependency or BswServiceDependency that in turn aggregates a DiagnosticRoutineNeeds with attribute diagRoutineType set to DiagnosticRoutineTypeEnum.synchronous. ⌋()

[constr_1341] Consistency of DiagnosticServiceSwMapping with respect to asynchronously called DiagnosticRoutines ⌈ Each DiagnosticServiceSwMapping that references a DiagnosticRoutineControl that aggregates a DiagnosticStopRoutine and/or DiagnosticRequestRoutineResults in the role stop resp. requestResult shall only reference a SwcServiceDependency or BswServiceDependency that in turn aggregates a DiagnosticRoutineNeeds with attribute diagRoutineType set to DiagnosticRoutineTypeEnum.asynchronous. ⌋()

[TPS_DEXT_01049] Consistency of DiagnosticServiceSwMapping with respect to routine IDs ⌈ For each DiagnosticServiceSwMapping that references a DiagnosticRoutineNeeds and a DiagnosticRoutineControl, the value of DiagnosticRoutineNeeds.ridNumber shall be ignored and the value of DiagnosticRoutineControl.routine.id shall be taken instead. ⌋(*RS_DEXT_00015, RS_DEXT_00052*)

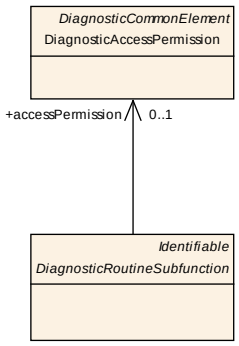

Figure 5.24: Permission is granted per subfunction

Access permissions for service `RoutineControl` are granted per subfunction, as opposed to the entire service (i.e. `DiagnosticRoutineControl`).

For this purpose, the reference from the abstract base class[1] `DiagnosticRoutineSubfunction` to `DiagnosticAccessPermission` in the role `accessPermission` exists (see Figure 5.24).

Consequently, the reference from `DiagnosticRoutineControl` (via its abstract base class `DiagnosticServiceInstance`) to meta-class `DiagnosticAccessPermission` has no meaning.

[constr_1515] Reference from `DiagnosticRoutineControl` to `DiagnosticAccessPermission` has no meaning ⌈ The reference from `DiagnosticRoutineControl` (via its abstract base class `DiagnosticServiceInstance`) in the role `accessPermission` to meta-class `DiagnosticAccessPermission` shall not be used. ⌋*()*

Class	**DiagnosticRoutine**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents the ability to define a diagnostic routine. **Tags:** atp.recommendedPackage=DiagnosticRoutines			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
id	PositiveInteger	1	attr	This is the numerical identifier used to identify the DiagnosticRoutine in the scope of diagnostic workflow **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
requestResult	DiagnosticRequestRoutineResults	0..1	aggr	This represents the ability to request the result of a running routine.

[1] of meta-classes `DiagnosticStartRoutine`, `DiagnosticStopRoutine`, and `DiagnosticRequestRoutineResults`

routineInfo	PositiveInteger	0..1	attr	This represents the routine info byte. The info byte contains a manufacturer-specific value (for the identification of record identifiers) that is reported to the tester. Other use cases for this attribute are mentioned in ISO 27145 and ISO 26021.
start	DiagnosticStart Routine	0..1	aggr	This represents the ability to start a routine
stop	DiagnosticStop Routine	0..1	aggr	This represents the ability to stop a running routine.

Table 5.74: DiagnosticRoutine

Class	**DiagnosticStartRoutine**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This represents the ability to start a diagnostic routine.			
Base	ARObject, DiagnosticRoutineSubfunction, Identifiable, MultilanguageReferrable, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
request	DiagnosticPara meter	*	aggr	This represents the request parameters.
response	DiagnosticPara meter	*	aggr	This represents the response parameters.

Table 5.75: DiagnosticStartRoutine

Class	**DiagnosticStopRoutine**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This represents the ability to stop a diagnostic routine.			
Base	ARObject, DiagnosticRoutineSubfunction, Identifiable, MultilanguageReferrable, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
request	DiagnosticPara meter	*	aggr	This represents the request parameters.
response	DiagnosticPara meter	*	aggr	This represents the response parameters.

Table 5.76: DiagnosticStopRoutine

Class	**DiagnosticRequestRoutineResults**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents the ability to define the result of a diagnostic routine execution.			
Base	ARObject, DiagnosticRoutineSubfunction, Identifiable, MultilanguageReferrable, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

response	DiagnosticParameter	*	aggr	This represents the response parameters.

Table 5.77: DiagnosticRequestRoutineResults

Class	**DiagnosticRoutineControl**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::RoutineControl			
Note	This represents an instance of the "Routine Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRoutineControls			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
routine	DiagnosticRoutine	1	ref	This refers to the applicable DiagnosticRoutine.
routineControlClass	DiagnosticRoutineControlClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRoutineControl in the given context.

Table 5.78: DiagnosticRoutineControl

Class	**DiagnosticRoutineNeeds**			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the general needs on the configuration of the Diagnostic Communication Manager (Dcm) which are not related to a particular item (e.g. a PID). The main use case is the mapping of service ports to the Dcm which are not related to a particular item.			
Base	ARObject, DiagnosticCapabilityElement, Identifiable, MultilanguageReferrable, Referrable, ServiceNeeds			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagRoutineType	DiagnosticRoutineTypeEnum	1	attr	This denotes the type of diagnostic routine which is implemented by the referenced server port.
ridNumber	PositiveInteger	0..1	attr	This represents a routine identifier for the diagnostic routine. This allows to predefine the RID number if the a function developer has received a particular requirement from the OEM or from a standardization body.

Table 5.79: DiagnosticRoutineNeeds

Enumeration	**DiagnosticRoutineTypeEnum**
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds
Note	This enumerator specifies the different types of diagnostic routines.

Literal	*Description*
asynchronous	This indicates that the diagnostic server is not blocked while the diagnostic routine is running. **Tags:** atp.EnumerationValue=0
synchronous	This indicates that the diagnostic routine blocks the diagnostic server in the ECU while the routine is running. **Tags:** atp.EnumerationValue=1

Table 5.80: DiagnosticRoutineTypeEnum

5.5.13 SecurityAccess

This chapter describes the modeling of diagnostic services `SecurityAccess` (0x27).

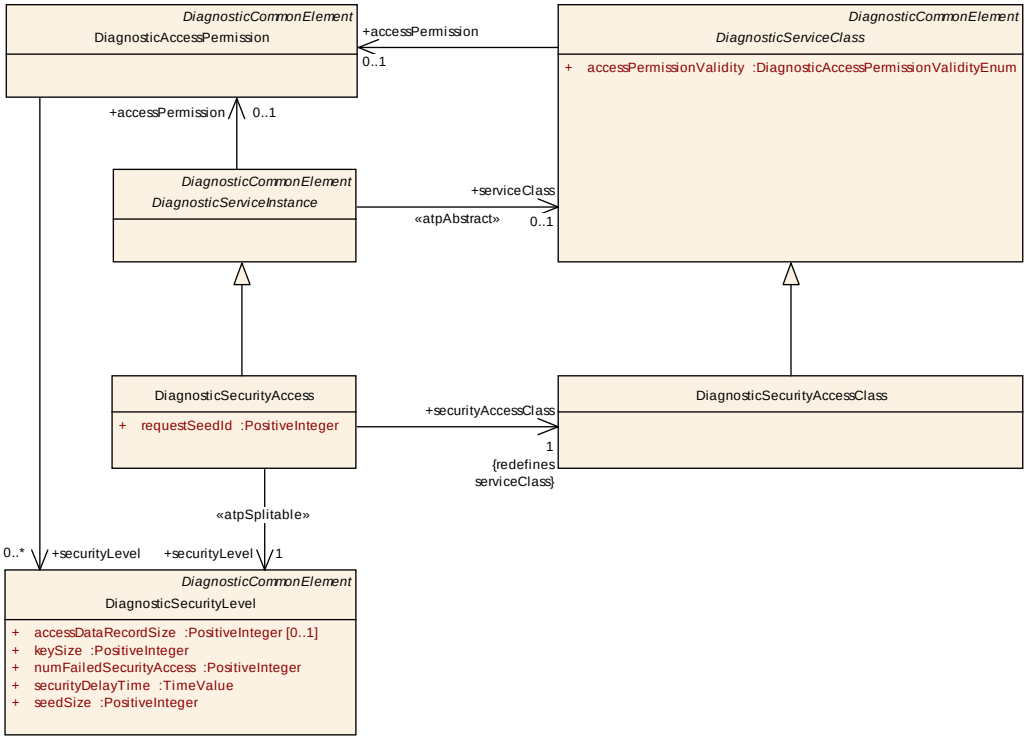

Figure 5.25: Modeling of diagnostic service `SecurityAccess` (0x27)

[TPS_DEXT_01053] Existence of `DiagnosticSecurityAccess.securityLevel`
⌈ The configuration of a given `DiagnosticSecurityAccess` is considered incomplete until the reference in the role `DiagnosticSecurityAccess.securityLevel` exists. ⌋*(RS_DEXT_00009, RS_DEXT_00042)*

The meaning of [TPS_DEXT_01053] is that the reference may be missing in intermediate steps of the configuration work flow. But at the point in time where ECU configuration is generated from the `DiagnosticExtract` the reference is needed to able to make sense of the model for the given `DiagnosticSecurityAccess`.

Please note that (as already explained in section 5.5) the sub-functions of this service are modeled by means of the category attribute.

In response to conceptual differences between many other diagnostic services and SecurityAccess, the applicable sub-functions for the diagnostic service SecurityAccess are **not** defined by means of the attribute DiagnosticSecurityAccess.category.

[TPS_DEXT_01036] Work-flow within the execution of the diagnostic service SecurityAccess ⌈ The work-flow within the execution of the diagnostic service SecurityAccess basically boils down to the tester sending the request to obtain a *seed* from the diagnostic stack and then sending back a *key* to the stack.

Thus, the sub-functions could be generically be described as *requestSeed* and *sendKey*, which is precisely what the ISO 14229-1 [16] does.

According to this logic, the *requestSeed* could get a specific number assigned to identify the sub-function and then the *sendKey* sub-function would get assigned the number of the *requestSeed* sub-function + 1. Again, this is fully in line with the ISO 14229-1 [16]. ⌋*(RS_DEXT_00009)*

However, there is further dimension to take into account, namely the DiagnosticSecurityLevel. According to ISO 14229-1 [16], different security levels make different numbers for the sub-function identifier.

[TPS_DEXT_01037] Semantics of DiagnosticSecurityAccess.requestSeedId ⌈ The attribute DiagnosticSecurityAccess.requestSeedId shall be used to define the number of the sub-function of the diagnostic service SecurityAccess according to the intended security level. ⌋*(RS_DEXT_00009)*

[constr_1342] Possible values for DiagnosticSecurityAccess.requestSeedId ⌈ The value of the attribute DiagnosticSecurityAccess.requestSeedId **shall only be set to an odd number**[2].

The supported value range consists of the following list:

- all odd numbers in the closed interval **0x01** .. **0x41**
- **0x5F** (this corresponds to the case of *end-of-life activation of on-board pyrotechnic devices according to ISO 26021-2 [17]*)
- all odd numbers in the closed interval **0x61** .. **0x7E**

⌋*()*

In contrast to a similar situation in the case of the diagnostic service SessionControl (see section 5.5.14), there is no real evidence that a DiagnosticSecurityLevel always exists before the referencing DiagnosticSecurityAccess is created in order to properly establish the reference in the role DiagnosticSecurityAccess.securityLevel.

[2]The even numbers are reserved for the identification of the corresponding *sendKey* sub-function, as explained by [TPS_DEXT_01036].

[TPS_DEXT_01038] Motivation for making the reference `DiagnosticSecurityAccess.securityLevel` ≪atpSplitable≫ ⌈ The reference `DiagnosticSecurityAccess.securityLevel` needs to be decorated with the stereotype ≪atpSplitable≫ in order to advertise the idea that the reference to a corresponding `DiagnosticSecurityLevel` is created (potentially in a different artifact) some time after the actual creation of the given `DiagnosticSecurityAccess`. ⌋ *(RS_DEXT_00002, RS_DEXT_00009, RS_DEXT_00042)*

Of course, if the `DiagnosticSecurityLevel` factually exists before the definition of `DiagnosticSecurityAccess` the reference can directly be inserted into the model.

Class	**DiagnosticSecurityAccess**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::SecurityAccess			
Note	This represents an instance of the "Security Access" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticSecurityAccesss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
requestSeedId	PositiveInteger	1	attr	This would be 0x01, 0x03, 0x05, ... The sendKey id can be computed by adding 1 to the requestSeedId
securityAccessClass	DiagnosticSecurityAccessClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticSecurityAccess in the given context.
securityLevel	DiagnosticSecurityLevel	1	ref	This reference identifies the applicable security level for the security access. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=securityLevel

Table 5.81: DiagnosticSecurityAccess

5.5.14 SessionControl

This chapter describes the modeling of diagnostic services `SessionControl` (0x10). The obvious goal of the service is to support the switching from one diagnostic session to another.

[TPS_DEXT_01081] Modeling of `DiagnosticSessionControl` ⌈ For the purpose of providing a means to specify the switching from one diagnostic session to another diagnostic session, `DiagnosticSessionControl` refers to a `DiagnosticSession` in the role `diagnosticSession`. ⌋ *(RS_DEXT_00003, RS_DEXT_00040)*

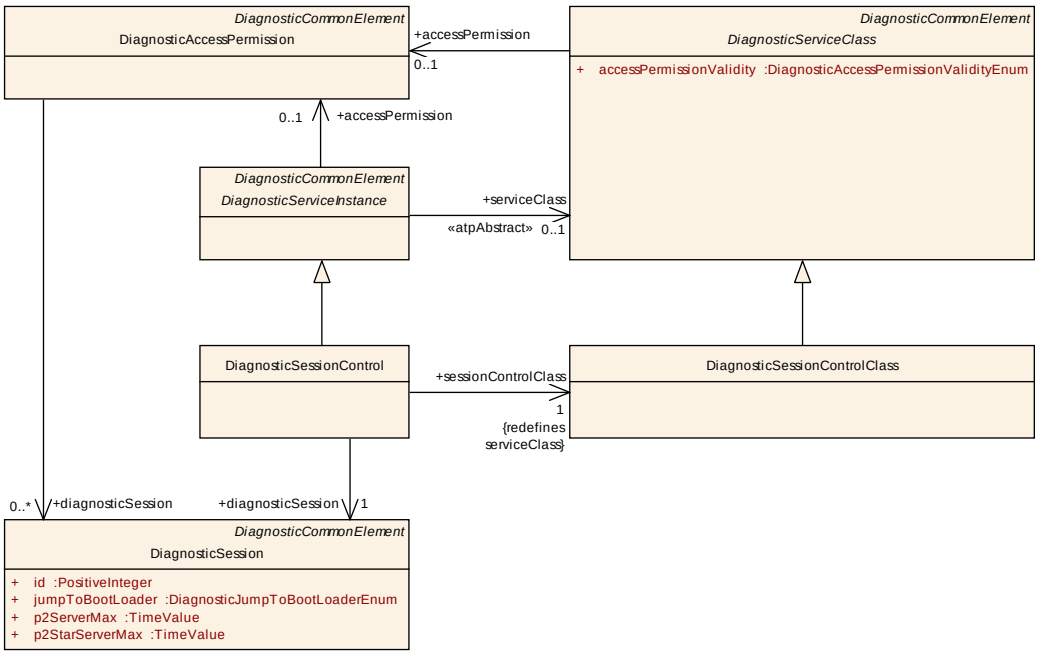

Figure 5.26: Modeling of diagnostic service `SessionControl` (0x10)

According to ISO 14229-1 [16], the diagnostic service `SessionControl` defines sub-functions.

[TPS_DEXT_01039] Identification of the sub-function of `DiagnosticSessionControl` ⌈ In the case of `DiagnosticSessionControl` it would not be a good idea to encode the applicable sub-function by means of the attribute `DiagnosticSessionControl.category`.

Actually, the possible sub-functions are strongly related to the concept of the diagnostic session, represented by the meta-class `DiagnosticSession`.

The latter, in turn, has an attribute `id` that directly corresponds to the number of the applicable sub-function for `DiagnosticSessionControl`.

In other words, the sub-function of `DiagnosticSessionControl` is identified by means of the reference `DiagnosticSessionControl.diagnosticSession`. ⌋ *(RS_DEXT_00003, RS_DEXT_00051)*

[TPS_DEXT_01082] Existence of `DiagnosticSessionControl.diagnosticSession` ⌈ The idea of modeling the sub-function of `DiagnosticSessionControl` by means of the reference `DiagnosticSessionControl.diagnosticSession` implies that the applicable `DiagnosticSession` already exists at the time when the given `DiagnosticSessionControl` is created.

It is assumed that this will always be the case because the definition of `DiagnosticSession`s is part of laying the groundwork[3] for diagnostic communication. ⌋ *(RS_DEXT_00003, RS_DEXT_00040)*

[3] This is similar to the definition of commonly used data types in a software development project

It is hard to foresee a scenario where the DiagnosticSessions are defined near the very end of the work-flow that leads to a complete DiagnosticExtract.

Class	**DiagnosticSessionControl**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::SessionControl			
Note	This represents an instance of the "Session Control" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticSessionControls			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagnosticSession	DiagnosticSession	1	ref	This represents the applicable DiagnosticSessions
sessionControlClass	DiagnosticSessionControlClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticSessionControl in the given context.

Table 5.82: DiagnosticSessionControl

5.5.15 RequestFileTransfer

This chapter describes the modeling of diagnostic services RequestFileTransfer (0x38). The purpose of the service is the triggering of the transfer of a *file* from or to the AUTOSAR diagnostic stack.

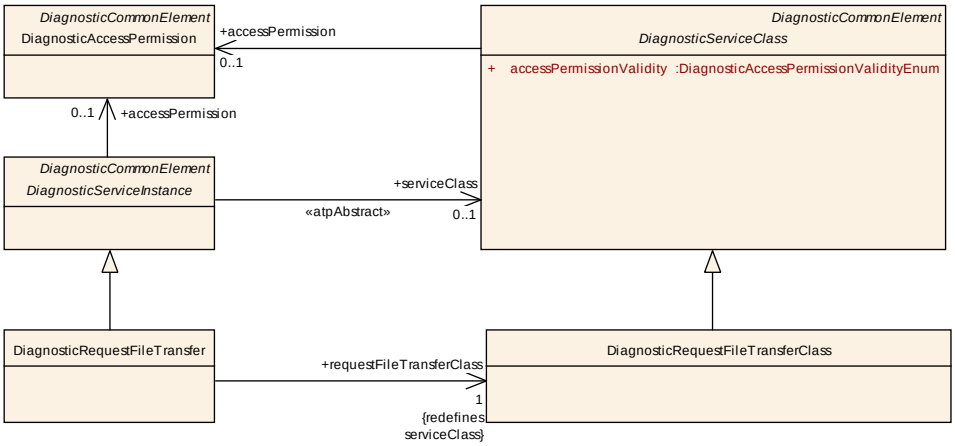

Figure 5.27: Modeling of diagnostic service RequestFileTransfer (0x38)

Please note that there is nothing to configure for DiagnosticRequestFileTransfer beyond its mere existence.

[TPS_DEXT_01090] Diagnostic service `RequestFileTransfer` does not define any sub-functions ⌈ The diagnostic service `RequestFileTransfer` does not define any sub-functions. therefore, the usage of the attribute `category` is not constrained for meta-class `DiagnosticRequestFileTransfer`. ⌋*(RS_DEXT_00057)*

Class	**DiagnosticRequestFileTransfer**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::RequestFileTransfer			
Note	This diagnostic service instance implements the UDS service 0x38. **Tags:** atp.recommendedPackage=DiagnosticRequestFileTransfers			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
requestFileTransferClass	DiagnosticRequestFileTransferClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestFileTransfer in the given context.

Table 5.83: DiagnosticRequestFileTransfer

Class	**DiagnosticRequestFileTransferClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::RequestFileTransfer			
Note	This meta-class contains attributes shared by all instances of the "Request File transfer" diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRequestFileTransfers			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.84: DiagnosticRequestFileTransferClass

5.6 OBD Diagnostic Services supported by AUTOSAR

Support for diagnostic services for on-board diagnostics (OBD) [18] requires the definition of further meta-classes similar to e.g. `DiagnosticDataIdentifier` or `DiagnosticRoutine`.

[TPS_DEXT_01092] Semantics of `DiagnosticParameterIdentifier` ⌈ The meta-class `DiagnosticParameterIdentifier` is used to reflect the concept of the

so-called Parameter Identifiers (PID) in the diagnostic extract. A DiagnosticParameterIdentifier defines the following properties:

- Each DiagnosticParameterIdentifier gets a numerical ID carried in the attribute DiagnosticParameterIdentifier.id.
- DiagnosticParameterIdentifier may also contains a so-called *Support Info Byte*, modeled as DiagnosticSupportInfoByte.
- The definition of a DiagnosticParameterIdentifier also contains a list of data associated with the PID. This list is modeled as an aggregation of DiagnosticParameter.

⌋*(RS_DEXT_00068)*

As already mentioned in [TPS_DEXT_01092], the DiagnosticParameterIdentifier shall not be confused with the DiagnosticParameter. The latter is used "inside" the definition of DiagnosticParameterIdentifier, but also in DiagnosticDataIdentifier or DiagnosticRoutine, to define one (out of potentially many) piece of information held in the scope of a DiagnosticParameterIdentifier.

[constr_1447] Restrictions for the value of DiagnosticParameterIdentifier.id ⌈ The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of DiagnosticParameterIdentifier.id. ⌋*()*

[constr_1448] Interval of DiagnosticParameterIdentifier.id ⌈ The allowed interval for values of DiagnosticParameterIdentifier.id shall not exceed [0..255]. ⌋*()*

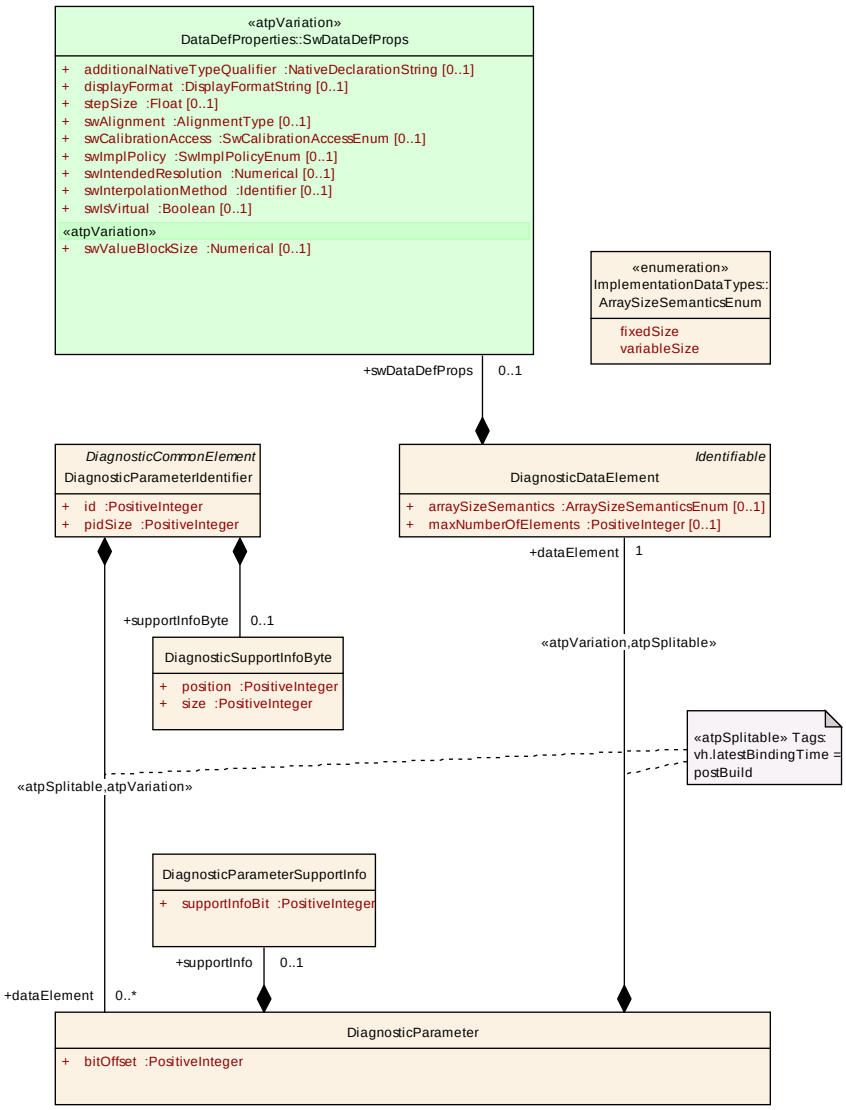

Figure 5.28: Modeling of the `DiagnosticParameterIdentifier`

[constr_1449] PID shall only carry a fixed-length collection of data ⌈ The value of `DiagnosticParameterIdentifier.dataElement.dataElement.arraySizeSemantics` shall not be set to `variableSize`. ⌋()

Class	**DiagnosticParameterIdentifier**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents the ability to model a diagnostic parameter identifier (PID) for the purpose of executing on-board diagnostics (OBD). **Tags:** atp.recommendedPackage=DiagnosticParameterIdentifiers			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

dataElement	DiagnosticParameter	*	aggr	This represents the data carried by the DiagnosticParameterIdentifier. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dataElement, variationPoint.shortLabel vh.latestBindingTime=postBuild
id	PositiveInteger	1	attr	This is the numerical identifier used to identify the DiagnosticParameterIdentifier in the scope of diagnostic workflow (see SAE J1979-DA).
pidSize	PositiveInteger	1	attr	The size of the entire PID can be greater than the sum of the data elements because padding might be applied. Unit: byte.
supportInfoByte	DiagnosticSupportInfoByte	0..1	aggr	This represents the supported information associated with the DiagnosticParameterIdentifier.

Table 5.85: DiagnosticParameterIdentifier

Class	**DiagnosticParameterSupportInfo**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This represents a way to define which bit of the supportInfo is representing this part of the PID			
Base	ARObject			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
supportInfoBit	PositiveInteger	1	attr	defines the bit in the SupportInfo byte, which represents the PID DataElement pidSize / position / size. Unit: byte.

Table 5.86: DiagnosticParameterSupportInfo

Class	**DiagnosticSupportInfoByte**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class defines the support information (typically byte A) to declare the usability of the DataElements within the so-called packeted PIDs (e.g. PID$68).			
Base	ARObject			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
position	PositiveInteger	1	attr	This represents the position of the supportInfo in the PID. Unit: byte.
size	PositiveInteger	1	attr	This represents the size of the supportInfo within the PID. Unit: byte.

Table 5.87: DiagnosticSupportInfoByte

5.6.1 OBD Mode 0x01 (RequestCurrentPowertrainDiagnosticData)

The service `RequestCurrentPowertrainDiagnosticData` is modeled in Figure 5.29.

[TPS_DEXT_01125] Support for diagnostic service `RequestCurrentPowertrainDiagnosticData` ⌈ The modeling support for the OBD diagnostic service `RequestCurrentPowertrainDiagnosticData` utilizes the new meta-class `DiagnosticParameterIdentifier` such that the meta-class `DiagnosticRequestCurrentPowertrainData` maintains a reference to `DiagnosticParameterIdentifier` in the role `pid`. ⌋*(RS_DEXT_00069)*

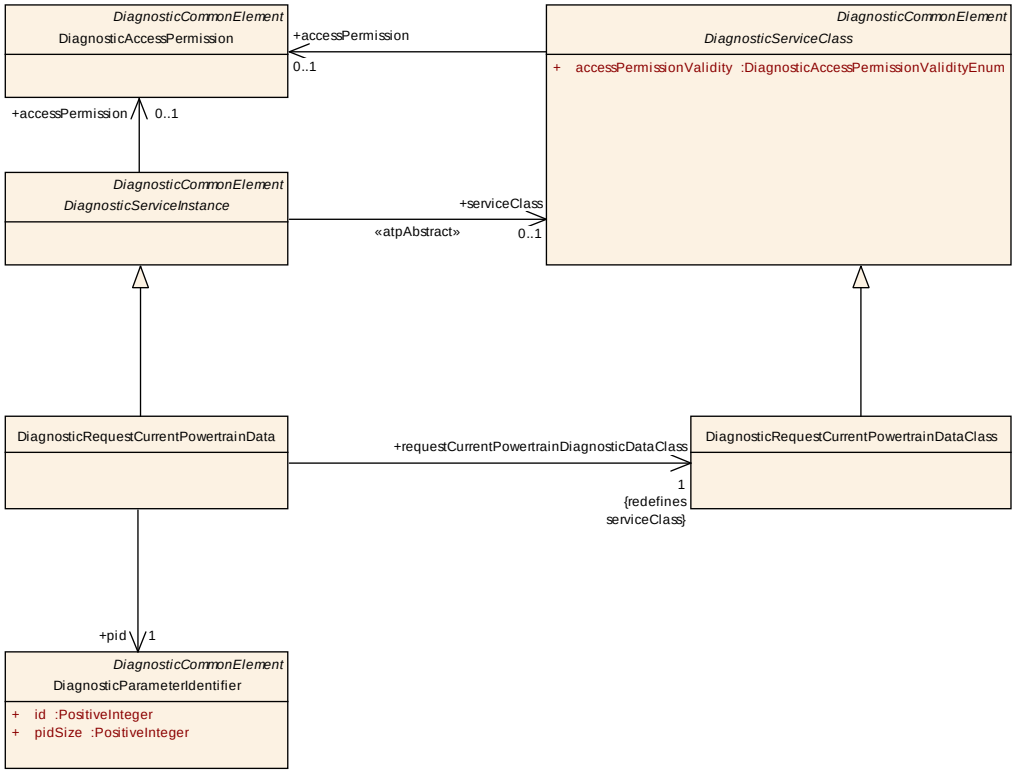

Figure 5.29: Modeling of diagnostic service for OBD Mode 0x01

Class	**DiagnosticRequestCurrentPowertrainData**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x01_RequestCurrentPowertrainDiagnosticData			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x01 service. **Tags:** atp.recommendedPackage=DiagnosticRequestCurrentPowertrainDatas			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
pid	DiagnosticParameterIdentifier	1	ref	This represents the PID associated with this instance of the OBD mode 0x01 service.

requestCurrentPowertrainDiagnosticDataClass	DiagnosticRequestCurrentPowertrainDataClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestCurrentPowertrainData in the given context.

Table 5.88: DiagnosticRequestCurrentPowertrainData

Class	**DiagnosticRequestCurrentPowertrainDataClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x01_RequestCurrentPowertrainDiagnosticData			
Note	This meta-class represents the ability to define common properties for all instances of the "Request current Powertrain Data" OBD diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRequestCurrentPowertrainDatas			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.89: DiagnosticRequestCurrentPowertrainDataClass

5.6.2 OBD Mode 0x02 (RequestPowertrainFreezeFrameData)

The service `RequestPowertrainFreezeFrameData` is modeled in Figure 5.30.

[TPS_DEXT_01126] Support of OBD service `RequestPowertrainFreeze-FrameData` ⌈ The modeling support for the ODB diagnostic service `RequestPower-trainFreezeFrameData` utilizes the new meta-class `DiagnosticParameterI-dentifier` such that the meta-class `DiagnosticRequestCurrentPowertrain-Data` maintains a reference to `DiagnosticParameterIdentifier` in the role `pid`. ⌋*(RS_DEXT_00070)*

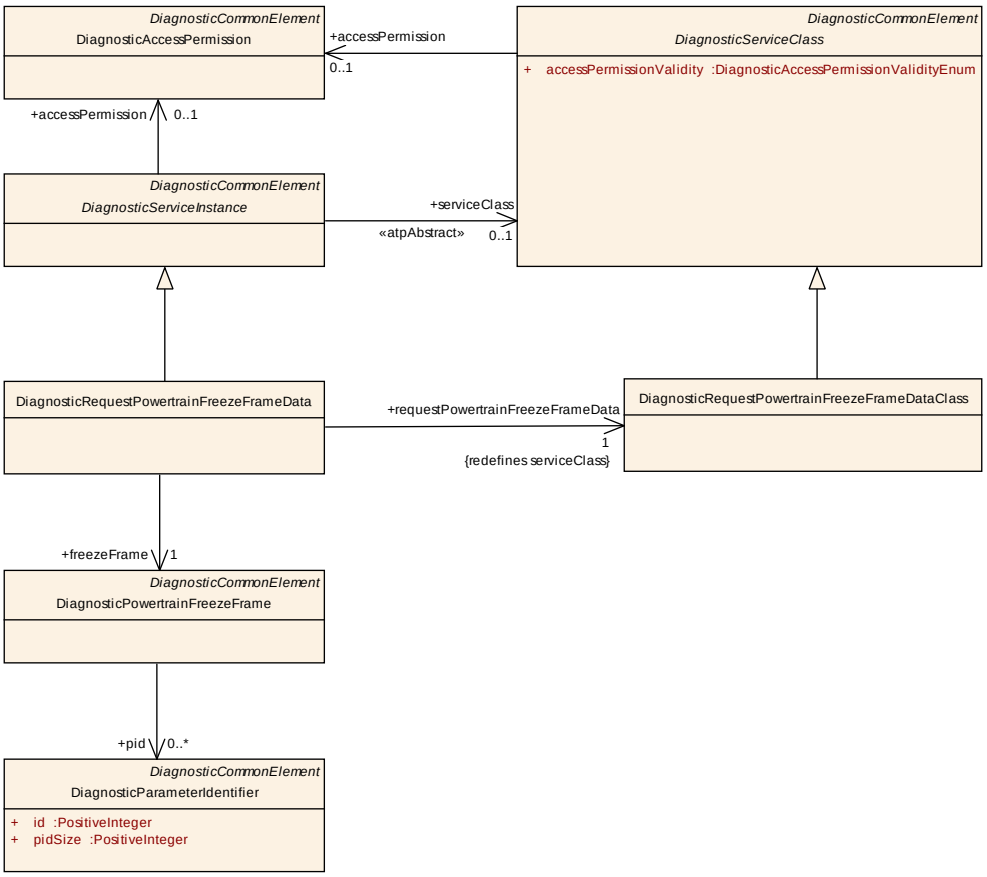

Figure 5.30: Modeling of diagnostic service for OBD Mode 0x02

Class	**DiagnosticRequestPowertrainFreezeFrameData**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x02_RequestPowertrainFreezeFrameData			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x02 service. **Tags:** atp.recommendedPackage=DiagnosticPowertrainFreezeFrames			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
freezeFrame	DiagnosticPowertrainFreezeFrame	1	ref	This represents the associated freeze-frame.
requestPowertrainFreezeFrameData	DiagnosticRequestPowertrainFreezeFrameDataClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestPowertrainFreezeFrameData in the given context.

Table 5.90: DiagnosticRequestPowertrainFreezeFrameData

Class	**DiagnosticRequestPowertrainFreezeFrameDataClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x02_RequestPowertrainFreezeFrameData			
Note	This meta-class represents the ability to define common properties for all instances of the "Request Powertrain Freeze Frame Data" OBD diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticPowertrainFreezeFrames			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.91: DiagnosticRequestPowertrainFreezeFrameDataClass

Class	**DiagnosticPowertrainFreezeFrame**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x02_RequestPowertrainFreezeFrameData			
Note	This meta-class represents a powertrain-related freeze-frame. In theory, this meta-class would need an additional id attribute. However, legal regulations requires only a single value for this attribute anyway. **Tags:** atp.recommendedPackage=DiagnosticPowertrainFreezeFrames			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
pid	DiagnosticParameterIdentifier	*	ref	This represents the PID associated with this instance of the OBD mode 0x02 service.

Table 5.92: DiagnosticPowertrainFreezeFrame

5.6.3 OBD Mode 0x03 / 0x07 (RequestEmissionRelatedDiagnosticTroubleCodes)

[TPS_DEXT_01127] Semantics of meta-class `DiagnosticRequestEmissionRelatedDTC` ⌈ Two very similar OBD services, subsumed as `RequestEmissionRelatedDiagnosticTroubleCodes` are supported by means of the meta-class `DiagnosticRequestEmissionRelatedDTC`. ⌋*(RS_DEXT_00071)*

The modeling is sketched in Figure 5.31.

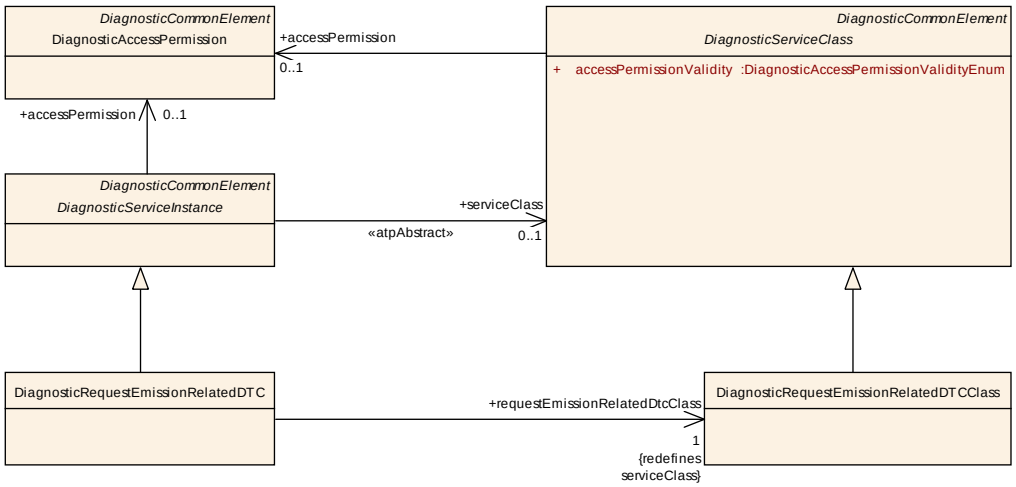

Figure 5.31: Modeling of diagnostic service for OBD Modes 0x03, 0x07

Class	**DiagnosticRequestEmissionRelatedDTC**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x03_0x07_RequestEmissionRelatedDTC			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x03/0x07 service. **Tags:** atp.recommendedPackage=DiagnosticRequestEmissionRelatedDTCs			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
requestEmissionRelatedDtcClass	DiagnosticRequestEmissionRelatedDTCClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestEmissionRelatedDTC in the given context.

Table 5.93: DiagnosticRequestEmissionRelatedDTC

Class	**DiagnosticRequestEmissionRelatedDTCClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x03_0x07_RequestEmissionRelatedDTC			
Note	This meta-class represents the ability to define common properties for all instances of the "Request Emission Related DTC" OBD diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRequestEmissionRelatedDTCs			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.94: DiagnosticRequestEmissionRelatedDTCClass

5.6.4 OBD Mode 0x04 (ClearResetEmissionRelatedDiagnosticInformation)

[TPS_DEXT_01128] Semantics of meta-class DiagnosticClearResetEmissionRelatedInfo ⌈ The OBD diagnostic service ClearResetEmissionRelatedDiagnosticInformation is supported by means of the meta-class DiagnosticClearResetEmissionRelatedInfo. ⌋*(RS_DEXT_00072)*

The modeling is sketched in Figure 5.32.

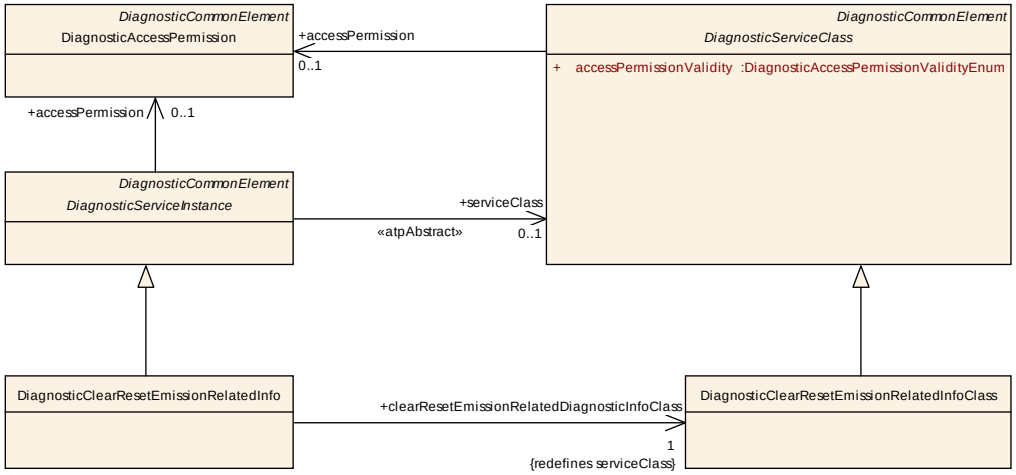

Figure 5.32: Modeling of diagnostic service for OBD Mode 0x04

Class	**DiagnosticClearResetEmissionRelatedInfo**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x04_ClearResetEmissionRelatedInfo			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x04 service. **Tags:** atp.recommendedPackage=DiagnosticClearResetEmissionRelatedInfos			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
clearResetEmissionRelatedDiagnosticInfoClass	DiagnosticClearResetEmissionRelatedInfoClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticClearResteEmissionRelatedInfo in the given context.

Table 5.95: DiagnosticClearResetEmissionRelatedInfo

Class	**DiagnosticClearResetEmissionRelatedInfoClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x04_ClearResetEmissionRelatedInfo			
Note	This meta-class represents the ability to define common properties for all instances of the "Clear Reset Emission Related Data" OBD diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticClearResetEmissionRelatedInfos			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.96: DiagnosticClearResetEmissionRelatedInfoClass

5.6.5 OBD Mode 0x06 (RequestOnBoardMonitoringTestResults)

[TPS_DEXT_01129] Support for OBD diagnostic service `RequestOnBoardMonitoringTestResults` ⌈ The OBD diagnostic service `RequestOnBoardMonitoringTestResults` is supported by manes of meta-class `DiagnosticRequestOnBoardMonitoringTestResults` that refers to the representation of the test result modeled as meta-class `DiagnosticTestResult` ⌋*(RS_DEXT_00073)*

The modeling is sketched in Figure 5.33.

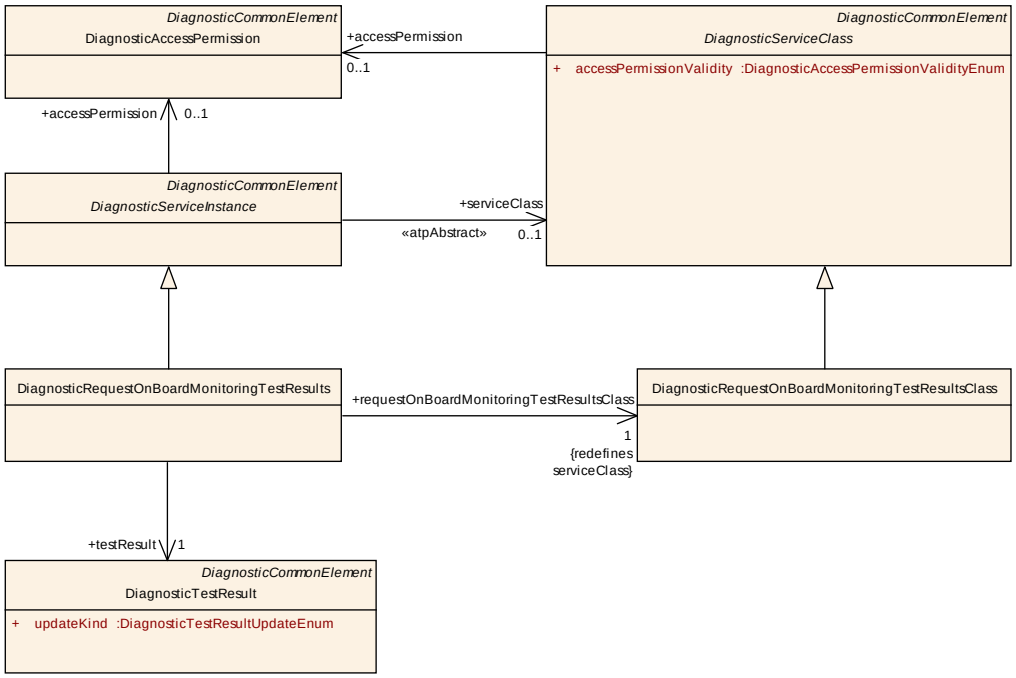

Figure 5.33: Modeling of diagnostic service for OBD Mode 0x06

In general, the mode 0x06 supports the querying of supported test identifiers from a given server. However, this functionality is not supported in AUTOSAR, hence the existence of [constr_1462].

[constr_1462] Restrictions for the value of `DiagnosticTestResult.testIdentifier.id` ⌈ The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of `DiagnosticTestResult.testIdentifier.id`. ⌋ *()*

Class	**DiagnosticRequestOnBoardMonitoringTestResults**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x06_RequestOnBoardMonitoringTestResults			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x06 service. **Tags:** atp.recommendedPackage=DiagnosticRequestOnBoardMonitoringTestResultss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
requestOnBoardMonitoringTestResultsClass	DiagnosticRequestOnBoardMonitoringTestResultsClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestOnBoardMonitoringTestResults in the given context.
testResult	DiagnosticTestResult	1	ref	This reference identifies the applicable test identifier for setting up a request message for mode 0x06.

Table 5.97: DiagnosticRequestOnBoardMonitoringTestResults

Class	**DiagnosticRequestOnBoardMonitoringTestResultsClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x06_RequestOnBoardMonitoringTestResults			
Note	This meta-class represents the ability to define common properties for all instances of the "Request On-Board Monitoring Test Results" OBD diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRequestOnBoardMonitoringTestResultss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.98: DiagnosticRequestOnBoardMonitoringTestResultsClass

Class	**DiagnosticTestResult**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTestResult			
Note	This meta-class represents the ability to define diagnostic test results. **Tags:** atp.recommendedPackage=DiagnosticTestResults			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
event	DiagnosticEvent	0..1	ref	This attribute represents the diagnostic event that is related to the diagnostic test result.
monitoredIdentifier	DiagnosticMeasurementIdentifier	1	ref	This attribute represents the related diagnostic monitored identifier.
testIdentifier	DiagnosticTestIdentifier	1	aggr	This attribute represents the applicable test identifier.
updateKind	DiagnosticTestResultUpdateEnum	1	attr	This attribute controls the update behavior of the enclosing DiagnosticTestResult.

Table 5.99: DiagnosticTestResult

5.6.6 OBD Mode 0x08 (RequestControlOfOnBoardDevice)

[TPS_DEXT_01130] Support of OBD diagnostic service `RequestControlOfOnBoardDevice` ⌈ The OBD diagnostic service `RequestControlOfOnBoardDevice` is supported by means of meta-class `DiagnosticRequestControlOfOnBoardDevice` that in turn refers to a `DiagnosticTestRoutineIdentifier` in the role `testId` ⌋*(RS_DEXT_00074)*

The modeling is sketched in Figure 5.34.

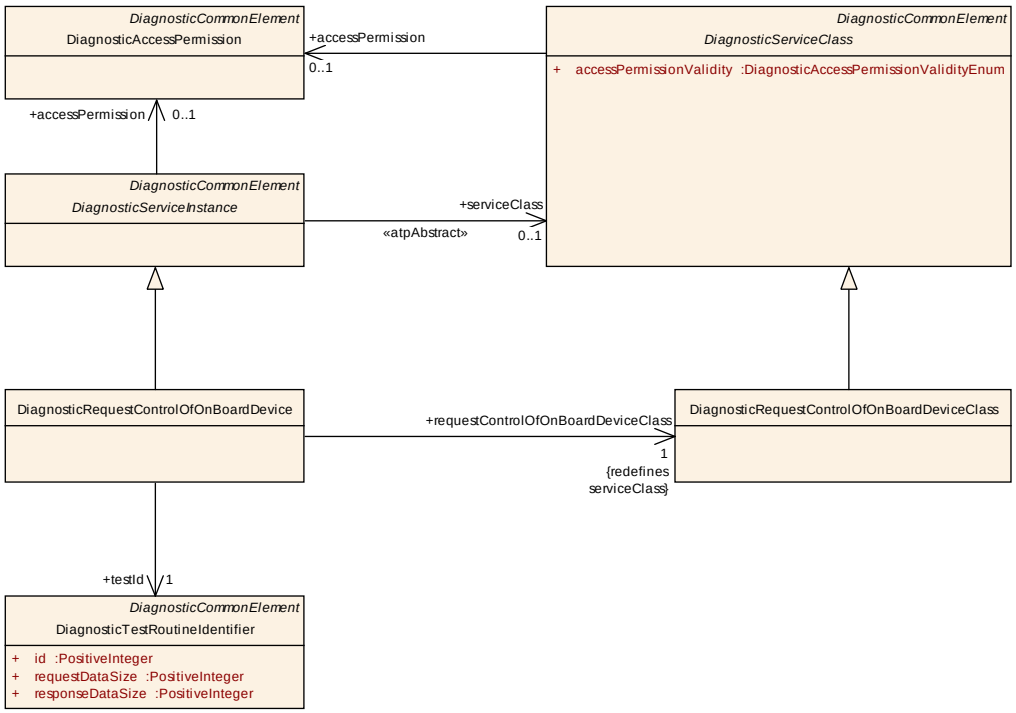

Figure 5.34: Modeling of diagnostic service for OBD Mode 0x08

In general, the mode 0x06 supports the querying of supported test routine identifiers from a given server. However, this functionality is not supported in AUTOSAR, hence the existence of [constr_1461].

[constr_1461] Restrictions for the value of `DiagnosticTestRoutineIdentifier.id` ⌈ The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of `DiagnosticTestRoutineIdentifier.id`. ⌋*()*

Class	**DiagnosticRequestControlOfOnBoardDevice**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x08_RequestControlOfOnBoardDevice			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x08 service. **Tags:** atp.recommendedPackage=DiagnosticRequestControlOfOnBoardDevices			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
requestControlOfOnBoardDeviceClass	DiagnosticRequestControlOfOnBoardDeviceClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequestControlOfOnBoardDevice in the given context.
testId	DiagnosticTestRoutineIdentifier	1	ref	This represents the test Id for the mode 0x08.

Table 5.100: DiagnosticRequestControlOfOnBoardDevice

Class	**DiagnosticRequestControlOfOnBoardDeviceClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x08_RequestControlOfOnBoardDevice			
Note	This meta-class represents the ability to define common properties for all instances of the "Request Control Of On-Board Device" OBD diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRequestControlOfOnBoardDevices			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.101: DiagnosticRequestControlOfOnBoardDeviceClass

Class	**DiagnosticTestRoutineIdentifier**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x08_RequestControlOfOnBoardDevice			
Note	This represents the test id of the DiagnosticTestIdentifier. **Tags:** atp.recommendedPackage=DiagnosticTestRoutineIdentifier			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
id	PositiveInteger	1	attr	This represents the numerical id of the DiagnosticTestIdentifier (see SAE J1979-DA).
requestDataSize	PositiveInteger	1	attr	This represents the specified data size for the request message. Unit: byte.
responseDataSize	PositiveInteger	1	attr	This represents the specified data size for the response message. Unit:byte.

Table 5.102: DiagnosticTestRoutineIdentifier

5.6.7 OBD Mode 0x09 (RequestVehicleInformation)

[TPS_DEXT_01131] Support for OBD diagnostic service `RequestVehicleInformation` ⌈ The OBD diagnostic service `RequestVehicleInformation` is supported by means of meta-class `DiagnosticRequestVehicleInfo` that in turn references a `DiagnosticInfoType` in the role `infoType`. ⌋*(RS_DEXT_00075)*

The modeling is sketched in Figure 5.35.

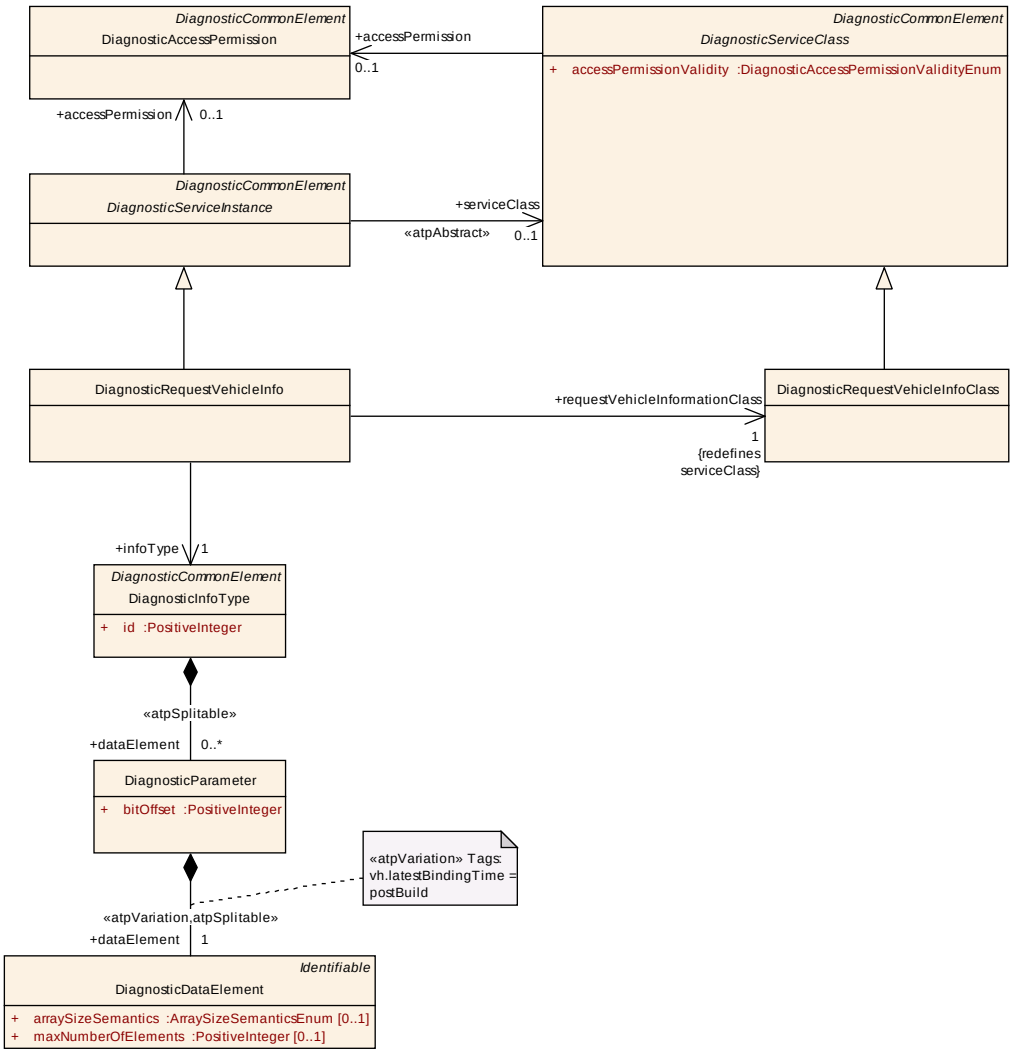

Figure 5.35: Modeling of diagnostic service for OBD Mode 0x09

[constr_1460] Restrictions for the value of `DiagnosticInfoType.id` ⌈ The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, and 0xE0 are not allowed to appear in the value of `DiagnosticInfoType.id`. ⌋*()*

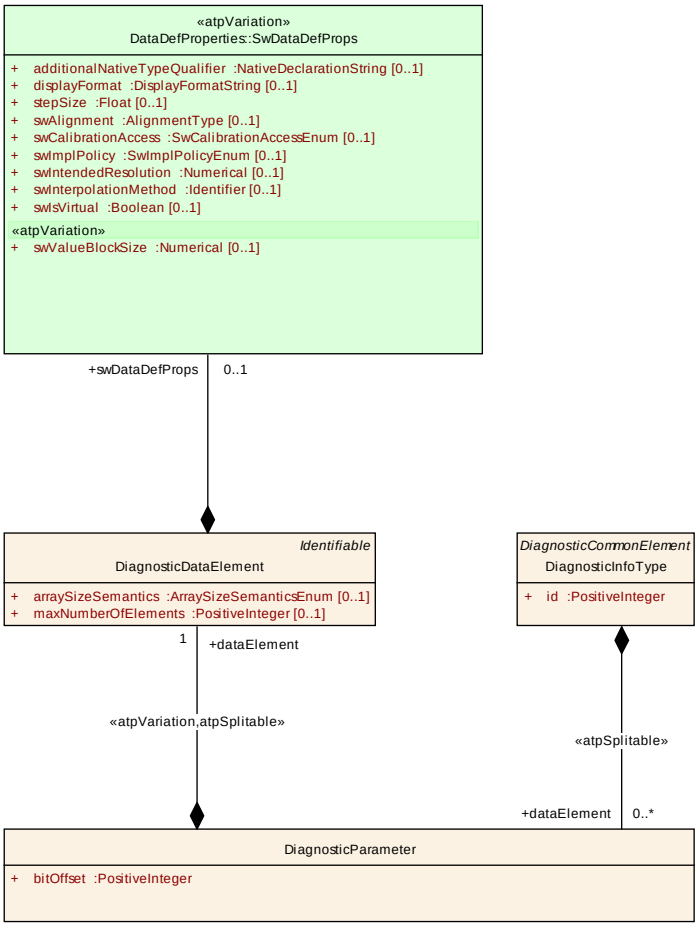

Figure 5.36: Modeling of `DiagnosticInfoType`

Class	**DiagnosticRequestVehicleInfo**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x09_RequestVehicleInformation			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x09 service. **Tags:** atp.recommendedPackage=DiagnosticRequestVehicleInfos			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
infoType	DiagnosticInfoType	1	ref	This represents the info type associated with the mode 0x09 service.
requestVehicleInformationClass	DiagnosticRequestVehicleInfoClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRequesVehicleInfo in the given context.

Table 5.103: DiagnosticRequestVehicleInfo

Class	**DiagnosticRequestVehicleInfoClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x09_RequestVehicleInformation			
Note	This meta-class represents the ability to define common properties for all instances of the "Request Vehicle Info" OBD diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRequestVehicleInfos			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
–	–	–	–	–

Table 5.104: DiagnosticRequestVehicleInfoClass

Class	**DiagnosticInfoType**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::CommonDiagnostics			
Note	This meta-class represents the ability to model an OBD info type. **Tags:** atp.recommendedPackage=DiagnosticInfoTypes			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
dataElement	DiagnosticParameter	*	aggr	This represents the data associated with the enclosing DiagnosticInfoType. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=dataElement
id	PositiveInteger	1	attr	This attribute represents the value of InfoType (see SAE J1979-DA).

Table 5.105: DiagnosticInfoType

5.6.8 OBD Mode 0x0A (RequestEmissionRelatedDiagnosticTroubleCodesPermanentStatus)

[TPS_DEXT_01132] Support for OBD diagnostic service `RequestEmissionRelatedDiagnosticTroubleCodesPermanentStatus` ⌈ The OBD diagnostic service `RequestEmissionRelatedDiagnosticTroubleCodesPermanentStatus` is supported by means of meta-class `DiagnosticRequestEmissionRelatedDTCPermanentStatus`. ⌋*(RS_DEXT_00076)*

The modeling is sketched in Figure 5.37.

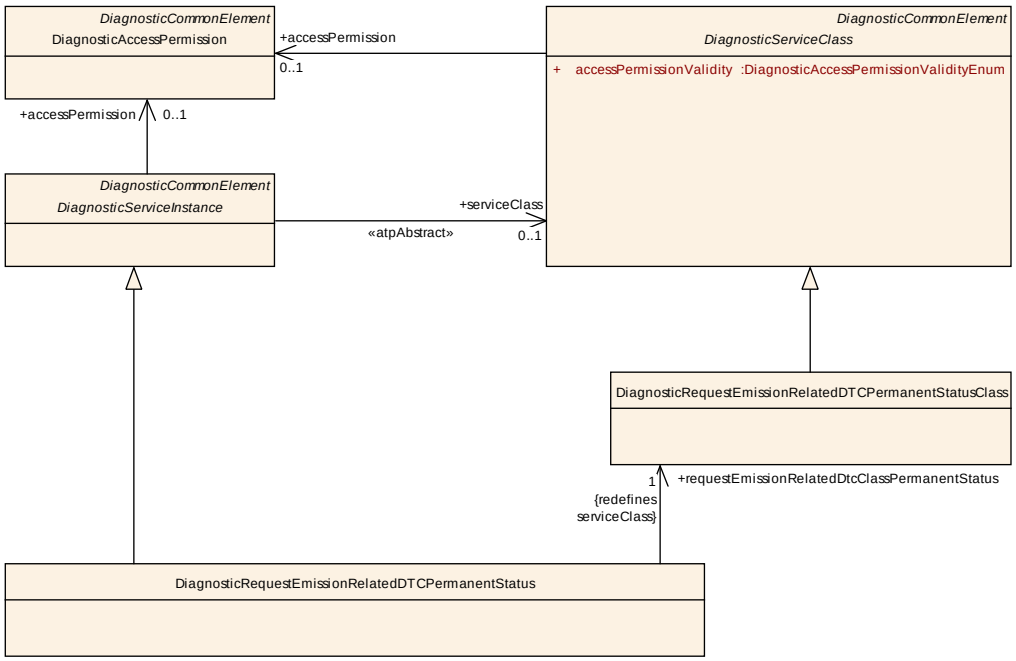

Figure 5.37: Modeling of diagnostic service for OBD Modes 0x0A

Class	**DiagnosticRequestEmissionRelatedDTCPermanentStatus**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x0A_ RequestEmissionRelatedDTCPermanentStatus			
Note	This meta-class represents the ability to model an instance of the OBD mode 0x0A service. **Tags:** atp.recommendedPackage=DiagnosticRequestEmissionRelatedDTC PermanentStatuss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Diagnostic ServiceInstance, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
requestEm issionRelat edDtcClas sPermane ntStatus	DiagnosticRequ estEmissionRel atedDTCPerma nentStatusClass	1	ref	This reference substantiates that abstract reference in the role serviceClass for this specific concrete class. Thereby, the reference represents the ability to access shared attributes among all DiagnosticRe-questEmissionRelatedDTCPermanentStatus in the given context.

Table 5.106: DiagnosticRequestEmissionRelatedDTCPermanentStatus

Class	**DiagnosticRequestEmissionRelatedDTCPermanentStatusClass**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::ObdService::Mode_0x0A_RequestEmissionRelatedDTCPermanentStatus			
Note	This meta-class represents the ability to define common properties for all instances of the "Request Emission Related DTC Permanent Status" OBD diagnostic service. **Tags:** atp.recommendedPackage=DiagnosticRequestEmissionRelatedDTCPermanentStatuss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticServiceClass, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.107: DiagnosticRequestEmissionRelatedDTCPermanentStatusClass

5.7 UDS Diagnostic Services for supporting WWH-OBD

[TPS_DEXT_01133] Support for WWH-OBD within the diagnostic extract ⌈ A support for WWH-OBD [19] within the diagnostic extract involves the usage of the following UDS services and their respective subfunctions:

DiagnosticReadDataByIdentifier (0x22) where the value of attribute DiagnosticDataIdentifier.id inside the interval defined by the OBD range, i.e.:

- F400-F4FF
- F600-F6FF
- F800-F8FF

Please note that the usage of this diagnostic service in an implementation of WWH-OBD corresponds to the existence of a DiagnosticValueNeeds on the side of an AtomicSwComponentType that interacts with the diagnostic service.

DiagnosticRoutineControl (0x31) where the value of attribute DiagnosticRoutine.id is inside the interval defined by the OBD range, i.e. E000-E0FF. Please note that the usage of this diagnostic service in an implementation of WWH-OBD corresponds to the existence of a DiagnosticRoutineNeeds on the side of an AtomicSwComponentType that implements routines executed within the diagnostic service.

DiagnosticClearDiagnosticInformation (0x14)

DiagnosticReadDTCInformation (0x19) with the subfunctions (0x4, 0x6, 0x42)

⌋*(RS_DEXT_00077)*

When used in a WWH-OBD environment, meta-class DiagnosticRoutine utilizes the attribute routineInfo to support the configuration of the diagnostic response. In a pure UDS environment (except in an application of ISO 26021 [17]) the attribute has no semantics.

Class	**DiagnosticValueNeeds**			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the general needs on the configuration of the Diagnostic Communication Manager (DCM) which are not related to a particular item (e.g. a PID). The main use case is the mapping of service ports to the DCM which are not related to a particular item. In the case of using a sender receiver communicated value, the related value shall be taken via assignedData in the role "signalBasedDiagnostics". In case of using a client/server communicated value, the related value shall be communicated via the port referenced by asssignedPort. The details of this communication (e.g. appropriate naming conventions) are specified in the related software specifications (SWS).			
Base	ARObject, DiagnosticCapabilityElement, Identifiable, MultilanguageReferrable, Referrable, ServiceNeeds			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataLength	PositiveInteger	0..1	attr	This attribute is applicable only if the ServiceNeed is aggregated within BswModuleDependency. This attribute represents the length of data (in bytes) provided for this particular PID signal.
diagnosticValueAccess	DiagnosticValueAccessEnum	0..1	attr	This attribute controls whether the data can be read and written or whether it is to be handled read-only.
didNumber	PositiveInteger	0..1	attr	This represents a Data identifier for the diagnostic value. This allows to predefine the DID number if the responsible function developer has received a particular requirement from the OEM or from a standardization body.
fixedLength	Boolean	0..1	attr	This attribute controls whether the data length of the data is fixed.
processingStyle	DiagnosticProcessingStyleEnum	0..1	attr	This attribute controls whether interaction requires the software-component to react synchronously on a request or whether it processes the request in background but still the DCM has to issue the call again to eventually obtain the result of the request.

Table 5.108: DiagnosticValueNeeds

Class	**DiagnosticRoutineNeeds**			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the general needs on the configuration of the Diagnostic Communication Manager (Dcm) which are not related to a particular item (e.g. a PID). The main use case is the mapping of service ports to the Dcm which are not related to a particular item.			
Base	ARObject, DiagnosticCapabilityElement, Identifiable, MultilanguageReferrable, Referrable, ServiceNeeds			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagRoutineType	DiagnosticRoutineTypeEnum	1	attr	This denotes the type of diagnostic routine which is implemented by the referenced server port.

ridNumber	PositiveInteger	0..1	attr	This represents a routine identifier for the diagnostic routine. This allows to predefine the RID number if the a function developer has received a particular requirement from the OEM or from a standardization body.

Table 5.109: DiagnosticRoutineNeeds

5.8 Diagnostic Service Mapping

Automotive diagnostics interacts with both application software and basic software in various ways that can be formalized using the AUTOSAR meta-model. This chapter contains a description of the formalization of this interaction along with the applicable constraints.

[TPS_DEXT_01040] Use case where the `DiagnosticExtract` refers to software-components ⌈ This is a list of the potential use case where the `DiagnosticExtract` refers to software-components in general and `PortPrototype`s in the context of either `CompositionSwComponentType`s or `AtomicSwComponentType`s:

- `DiagnosticExtract` refers to `PortPrototype` (for the access to `dataElement`) or `SwcServiceDependency` in the context of a `AtomicSwComponentType` embedded in the hierarchy created by the `rootSoftwareComposition`.
- `DiagnosticExtract` refers to a `PortPrototype`(for the access to `dataElement`) or `SwcServiceDependency` in the context of a `AtomicSwComponentType` embedded in the hierarchy created by a `CompositionSwComponentType` that is nowhere aggregated (for the time being).
- `DiagnosticExtract` refers to a `PortPrototype` (for the access to `dataElement`) or `SwcServiceDependency`in the context of an `AtomicSwComponentType`.
- `DiagnosticExtract` refers to a `BswServiceDependency`.

⌋*(RS_DEXT_00052)*

Class	**SwcServiceDependency**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ServiceMapping			
Note	Specialization of ServiceDependency in the context of an SwcInternalBehavior. It allows to associate ports, port groups and (in special cases) data defined for an atomic software component to a given ServiceNeeds element.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable, ServiceDependency			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

assignedData	RoleBasedDataAssignment	*	aggr	Defines the role of an associated data object of the same component. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
assignedPort	RoleBasedPortAssignment	*	aggr	Defines the role of an associated port of the same component. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=assignedPort, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
representedPortGroup	PortGroup	0..1	ref	This reference specifies an association between the ServiceNeeeds and a PortGroup, for example to request a communication mode which applies for communication via these ports. The referred PortGroup shall be local to this atomic SWC, but via the links between the PortGroups, a tool can evaluate this information such that all the ports linked via this port group on the same ECU can be found.
serviceNeeds	ServiceNeeds	1	aggr	The associated ServiceNeeds.

Table 5.110: SwcServiceDependency

Class	**BswServiceDependency**			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswBehavior			
Note	Specialization of ServiceDependency in the context of an BswInternalBehavior. It allows to associate BswModuleEntries and data defined for a BSW module or cluster to a given ServiceNeeds element.			
Base	ARObject, ServiceDependency			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
assignedData	RoleBasedDataAssignment	*	aggr	Defines the role of an associated data object (owned by this module or cluster) in the context of the ServiceNeeds element. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
assignedEntryRole	RoleBasedBswModuleEntryAssignment	*	aggr	Defines the role of an associated BswModuleEntry in the context of the ServiceNeeds element. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=assignedEntryRole, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
ident	BswServiceDependencyIdent	0..1	aggr	This adds the ability to become referrable to BswServiceDependency. **Tags:** atp.Status=shallBecomeMandatory xml.sequenceOffset=-100

serviceNeeds	ServiceNeeds	1	aggr	The associated ServiceNeeds.

Table 5.111: BswServiceDependency

[constr_1450] Service mapping for ODB mode 0x01 for DiagnosticParameterIdentifier ⌈ if a DiagnosticServiceSwMapping or DiagnosticServiceDataMapping refers to a DiagnosticRequestCurrentPowertrainData and a DiagnosticDataElement that is aggregated by a DiagnosticParameterIdentifier then then one of two alternative model configurations shall exist:

- SwcServiceDependency referenced by the same DiagnosticServiceSwMapping resp. DiagnosticServiceDataMapping shall aggregate an ObdPidServiceNeeds in the role serviceNeeds.
- The BswServiceDependencyIdent referenced by the same DiagnosticServiceSwMapping shall aggregate an ObdPidServiceNeeds in the role serviceNeeds.

⌋()

[constr_1451] Service mapping for OBD mode 0x09 for DiagnosticInfoType ⌈ if a DiagnosticServiceSwMapping refers to DiagnosticRequestVehicleInfo and a DiagnosticDataElement that is aggregated by a DiagnosticInfoType then one of two alternative model configurations shall exist:

- The SwcServiceDependency referenced by the same DiagnosticServiceSwMapping shall aggregate a ObdInfoServiceNeeds in the role serviceNeeds.
- The BswServiceDependencyIdent referenced by the same DiagnosticServiceSwMapping shall aggregate an ObdInfoServiceNeeds in the role serviceNeeds.

⌋()

[constr_1452] Service mapping for OBD mode 0x08 for DiagnosticInfoType ⌈ if a DiagnosticServiceSwMapping refers to a DiagnosticRequestControlOfOnBoardDevice then the SwcServiceDependency referenced by the same DiagnosticServiceSwMapping shall aggregate an ObdControlServiceNeeds in the role serviceNeeds. ⌋()

Class	**SwcServiceDependency**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ServiceMapping			
Note	Specialization of ServiceDependency in the context of an SwcInternalBehavior. It allows to associate ports, port groups and (in special cases) data defined for an atomic software component to a given ServiceNeeds element.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable, ServiceDependency			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
assignedData	RoleBasedDataAssignment	*	aggr	Defines the role of an associated data object of the same component. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
assignedPort	RoleBasedPortAssignment	*	aggr	Defines the role of an associated port of the same component. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=assignedPort, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
representedPortGroup	PortGroup	0..1	ref	This reference specifies an association between the ServiceNeeeds and a PortGroup, for example to request a communication mode which applies for communication via these ports. The referred PortGroup shall be local to this atomic SWC, but via the links between the PortGroups, a tool can evaluate this information such that all the ports linked via this port group on the same ECU can be found.
serviceNeeds	ServiceNeeds	1	aggr	The associated ServiceNeeds.

Table 5.112: SwcServiceDependency

Class	**ObdInfoServiceNeeds**			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the abstract needs of a component or module on the configuration of OBD Services in relation to a given InfoType (OBD Service 09) which is supported by this component or module.			
Base	ARObject, DiagnosticCapabilityElement, Identifiable, MultilanguageReferrable, Referrable, ServiceNeeds			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataLength	PositiveInteger	0..1	attr	This attribute is applicable only if the ServiceNeeds is aggregated within BswModuleDependency. This attribute represents the length of data (in bytes) provided for this InfoType.
infoType	PositiveInteger	0..1	attr	The InfoType according to ISO 15031-5

Table 5.113: ObdInfoServiceNeeds

Class	**ObdPidServiceNeeds**			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the abstract needs of a component or module on the configuration of OBD Services in relation to a particular PID (parameter identifier) which is supported by this component or module. In case of using a client/server communicated value, the related value shall be communicated via the port referenced by asssignedPort. The details of this communication (e.g. appropriate naming conventions) are specified in the related software specifications (SWS).			
Base	ARObject, DiagnosticCapabilityElement, Identifiable, MultilanguageReferrable, Referrable, ServiceNeeds			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataLength	PositiveInteger	0..1	attr	This attribute is applicable only if the ServiceNeeds is aggregated within BswModuleDependency. This attribute represents the length of data (in bytes) provided for this particular PID signal.
parameterId	PositiveInteger	0..1	attr	Standardized parameter identifier (PID) according to the OBD standard specified in attribute "standard".
standard	String	0..1	attr	Annotates the standard according to which the PID is given, e.g. "ISO15031-5" or "SAE J1979 Rev May 2007".

Table 5.114: ObdPidServiceNeeds

Class	**ObdControlServiceNeeds**			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the abstract needs of a component or module on the configuration of OBD Service 08 (request control of on-board system) in relation to a particular test-Identifier (TID) supported by this component or module.			
Base	ARObject, DiagnosticCapabilityElement, Identifiable, MultilanguageReferrable, Referrable, ServiceNeeds			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
testId	PositiveInteger	0..1	attr	Test Identifier (TID) according to ISO 15031-5.

Table 5.115: ObdControlServiceNeeds

5.8.1 Diagnostic Service Data Mapping

Please note that the Dcm is in general entitled to both read and write a `dataElement`. This applies even if the corresponding `PortPrototype` is a `PPortPrototype`. This means that the diagnostic service data mapping is limited to `SenderReceiverInterface`.

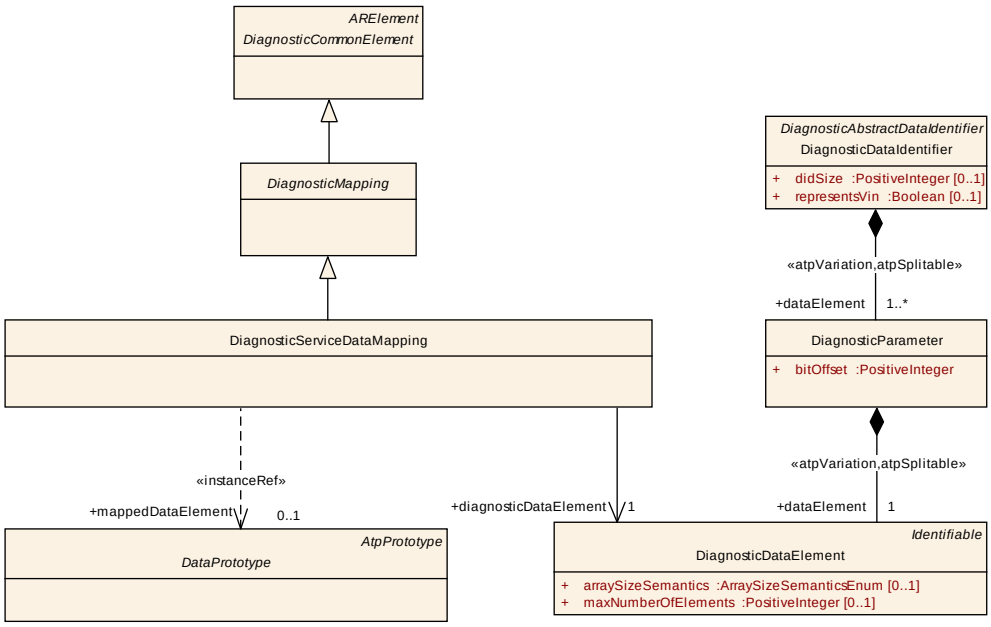

Figure 5.38: Dcm service data mapping

[TPS_DEXT_01041] Semantics of attribute DiagnosticServiceDataMapping.diagnosticDataElement ⌈ By means of the attribute DiagnosticServiceDataMapping.diagnosticDataElement it is possible to specify that the Dcm has access to a dataElement in a PortPrototype typed by a SenderReceiverInterface.

This type of data access is suitable for the diagnostic services ReadDataByIdentifier (0x22) and WriteDataByIdentifier (0x2E). ⌋*(RS_DEXT_00052)*

Class	**DiagnosticServiceDataMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::ServiceMapping			
Note	This represents the ability to define a mapping of a diagnostic service to a software-component. This kind of service mapping is applicable for the usage of SenderReceiverInterfaces. **Tags:** atp.recommendedPackage=DiagnosticServiceMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagnosticDataElement	DiagnosticDataElement	1	ref	This represents the applicable payload that corresponds to the referenced DataPrototype in the role mappedDataElement.
mappedDataElement	DataPrototype	0..1	iref	This represents the dataElement in the application software that is accessed for diagnostic purpose.

Table 5.116: DiagnosticServiceDataMapping

[constr_1343] Simultaneous existence of the attributes DiagnosticServiceDataMapping.diagnosticDataElement and DiagnosticDataByIdentifier.dataIdentifier ⌈ A DiagnosticServiceDataMapping.diagnostic-

DataElement shall also be aggregated by a DiagnosticDataByIdentifier in the role dataIdentifier.dataElement.dataElement. ⌋()

Please note that [constr_1343] shall only apply for the step in the methodology where the DiagnosticExtract is considered complete to the point that the configuration of the Dcm and Dem can be derived. Any intermediate step, e.g. hand-over from OEM to tier-1 supplier does not actually enforce [constr_1343].

In other words, [constr_1343] makes sure that there is a connection between the DiagnosticServiceDataMapping and the corresponding DiagnosticReadDataByIdentifier or DiagnosticWriteDataByIdentifier.

Only by this means the diagnostic service becomes fully usable.

[constr_1344] Condition for the identification of data types of attributes DiagnosticServiceDataMapping.mappedDataElement and DiagnosticServiceDataMapping.diagnosticDataElement ⌈ Both DiagnosticServiceDataMapping.mappedDataElement and DiagnosticServiceDataMapping.diagnosticDataElement shall be typed by either of the following options:

- ApplicationPrimitiveDataType where the value of attribute category is set to VALUE.
- ImplementationDataType where the value of attribute category is set to VALUE or to TYPE_REFERENCE that eventually resolves to an ImplementationDataType where attribute category is set to VALUE.

⌋()

Class	**ApplicationPrimitiveDataType**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Datatype::Datatypes			
Note	A primitive data type defines a set of allowed values. **Tags:** atp.recommendedPackage=ApplicationDataTypes			
Base	ARElement, ARObject, ApplicationDataType, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, AutosarDataType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.117: ApplicationPrimitiveDataType

Class	**ImplementationDataType**			
Package	M2::AUTOSARTemplates::CommonStructure::ImplementationDataTypes			
Note	Describes a reusable data type on the implementation level. This will typically correspond to a typedef in C-code. **Tags:** atp.recommendedPackage=ImplementationDataTypes			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, AutosarDataType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dynamicArraySizeProfile	String	0..1	attr	Specifies the profile which the array will follow in case this data type is a variable size array.
subElement (ordered)	ImplementationDataTypeElement	*	aggr	Specifies an element of an array, struct, or union data type. The aggregation of ImplementionDataTypeElement is subject to variability with the purpose to support the conditional existence of elements inside a ImplementationDataType representing a structure. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the SymbolProps for the ImplementationDataType. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=shortName
typeEmitter	NameToken	0..1	attr	This attribute is used to control which part of the AUTOSAR toolchain is supposed to trigger data type definitions.

Table 5.118: ImplementationDataType

In other words, [constr_1344] requires that both `DiagnosticServiceDataMapping.mappedDataElement`[4] and `DiagnosticServiceDataMapping.diagnosticDataElement` shall be typed by a primitive data type.

Please refer to [8] for a detailed explanation of the meaning of the value of a data type `category`.

[constr_1345] `DiagnosticDataElement` shall not (finally) be aggregated by a `DiagnosticRoutine` ⌈ A `DiagnosticDataElement` that is referenced by a `DiagnosticServiceDataMapping` shall not (finally) be aggregated by a `DiagnosticRoutine`. ⌋*()*

[TPS_DEXT_01042] Dem uses `DiagnosticServiceDataMapping` ⌈ There is a use case for the Dem to utilize a `DiagnosticServiceDataMapping` such that ele-

[4] `DiagnosticServiceDataMapping.mappedDataElement` can be an element of a (potentially large) composite data type. The utility of this is that this way the footprint of the data access to the payload of request and response messages can be kept as low as possible.

ments of a DiagnosticExtendedDataRecord are fetched from dataElements in an ApplicationSwComponentType.

Therefore, [constr_1345] does intentionally not exclude the aggregation of DiagnosticDataElement by DiagnosticExtendedDataRecord in the context of DiagnosticServiceDataMapping. ⌋(*RS_DEXT_00052*)

Class	**ApplicationSwComponentType**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	The ApplicationSwComponentType is used to represent the application software. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	ARElement, ARObject, AtomicSwComponentType, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, SwComponentType			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.119: ApplicationSwComponentType

5.8.2 Diagnostic Service Software Mapping

The diagnostic service software mapping is limited to ClientServerInterface or a direct function call (in the case of basic software or complex driver).

Class	**ClientServerInterface**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	A client/server interface declares a number of operations that can be invoked on a server by a client. **Tags:** atp.recommendedPackage=PortInterfaces			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, PortInterface, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
operation	ClientServerOperation	1..*	aggr	ClientServerOperation(s) of this ClientServerInterface. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=blueprintDerivationTime
possibleError	ApplicationError	*	aggr	Application errors that are defined as part of this interface.

Table 5.120: ClientServerInterface

[TPS_DEXT_01043] Purpose of DiagnosticServiceSwMapping ⌈ The meta-class DiagnosticServiceSwMapping has been introduced to support the creation of a relationship between the definition of a given diagnostic service to the SwcSer-

viceDependency (if the service applies to the application software) or BswServiceDependency (if the service applies to the basic software). ⌋(*RS_DEXT_00052*)

It is required to use the applicable form of reference to the target SwcServiceDependency depending on the context of the enclosing AtomicSwComponentType.

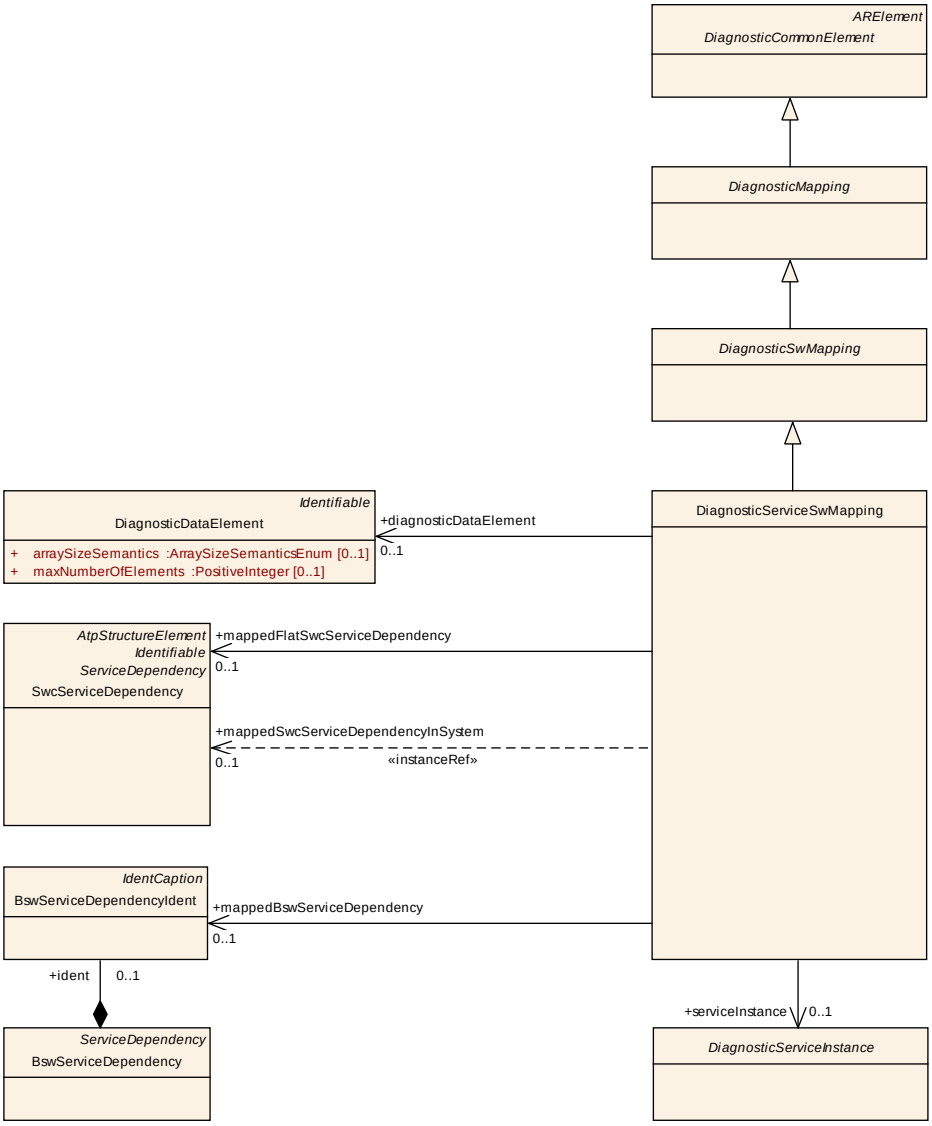

Figure 5.39: Dcm service software mapping

Class	**DiagnosticSwMapping (abstract)**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::ServiceMapping			
Note	This represents the ability to define a mapping between a diagnostic information (at this point there is no way to become more specific about the semantics) to a software-component.			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.121: DiagnosticSwMapping

Class	**DiagnosticServiceSwMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::ServiceMapping			
Note	This represents the ability to define a mapping of a diagnostic service to a software-component or a basic-software module. If the former is used then this kind of service mapping is applicable for the usage of ClientServerInterfaces. **Tags:** atp.recommendedPackage=DiagnosticServiceMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, DiagnosticSwMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagnosticDataElement	DiagnosticDataElement	0..1	ref	This represents a DiagnosticDataElement required to execute the respective diagnostic service in the context of the diagnostic service mapping,
mappedBswServiceDependency	BswServiceDependencyIdent	0..1	ref	This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.
mappedFlatSwcServiceDependency	SwcServiceDependency	0..1	ref	This represents the ability to refer to an AtomicSwComponentType that is available without the definition of how it will be emebdded into the component hiearchy.
mappedSwcServiceDependencyInSystem	SwcServiceDependency	0..1	iref	This represents the ability to point into the component hiearchy (under possible consideration of the rootSoftwareComposition)
serviceInstance	DiagnosticServiceInstance	0..1	ref	This represents the service instance that needs to be considered in this diagnostics service mapping.

Table 5.122: DiagnosticServiceSwMapping

[TPS_DEXT_01044] `BswServiceDependency` needs to act as the target of a reference ⌈ As indicated by Figure 5.39, the intention of `DiagnosticServiceSwMapping.mappedBswServiceDependency` is to refer to a `BswServiceDependency` in the same way as e.g. `DiagnosticServiceSwMapping.mappedFlatSwcServiceDependency` does.

However, `BswServiceDependency` is not derived from meta-class `Referrable` and can therefore never become the target of a reference like `DiagnosticServiceSwMapping.mappedBswServiceDependency`.

The remedy for this issue is to define meta-class `BswServiceDependencyIdent` that inherits from `IdentCaption` that in turn inherits from `Referrable`.

Then, by aggregating `BswServiceDependencyIdent` at `BswServiceDependency` in the role `ident BswServiceDependency` can **factually** become the target of the

reference and thus the original idea of DiagnosticServiceSwMapping.mappedBswServiceDependency becomes feasible. ⌋(*RS_DEXT_00052*)

Please note that the introduction [TPS_DEXT_01044], although being dangerously close to a hack, is necessary to keep the AUTOSAR XML Schema fully backwards-compatible.

In other words, if BswServiceDependency were updated to inherit from Referrable the consequence would be that all existing AUTOSAR models that contain instances BswServiceDependency would suddenly become invalid because Referrable.shortName is a mandatory attribute in the AUTOSAR XML Schema.

Class	**BswServiceDependencyIdent**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::ServiceMapping			
Note	This meta-class is created to add the ability to become the target of a reference to the non-Referrable BswServiceDependency.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, IdentCaption, Identifiable, MultilanguageReferrable, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 5.123: BswServiceDependencyIdent

[constr_1346] Allowed values of DiagnosticServiceSwMapping.serviceInstance ⌈ The applicability of the DiagnosticServiceSwMapping is limited to predefined set of diagnostic services.

By regulation of the AUTOSAR standard, DiagnosticServiceSwMapping.serviceInstance shall only point to the following sub-classes of DiagnosticServiceInstance:

- DiagnosticRoutine
- DiagnosticSecurityAccess
- DiagnosticReadDataByIdentifier
- DiagnosticWriteDataByIdentifier
- DiagnosticIOControl

⌋()

[constr_1347] Existence of attributes of DiagnosticServiceSwMapping ⌈ For any given DiagnosticServiceSwMapping, **one and only one of** the following references shall exist:

- DiagnosticServiceSwMapping.mappedFlatSwcServiceDependency
- DiagnosticServiceSwMapping.mappedSwcServiceDependencyInSystem
- DiagnosticServiceSwMapping.mappedBswServiceDependency

⌋()

[constr_1347], among further clarifications, reflects the fact that at most a single SwcServiceDependency can be referenced by a DiagnosticServiceSwMapping and this SwcServiceDependency cannot be identified by both mappedSwcServiceDependencyInSystem and mappedFlatSwcServiceDependency.

6 Diagnostic Event Handling

6.1 Introduction

This subchapter describes the meta-model elements that define the handling of and the functionality around diagnostic events.

In a standard AUTOSAR Basic Software architecture, the definitions based on the model elements described in this subchapter are realized by the Diagnostic Event Manager (Dem) module.

The following figure gives an overview on the model elements related to the diagnostic event functionality.

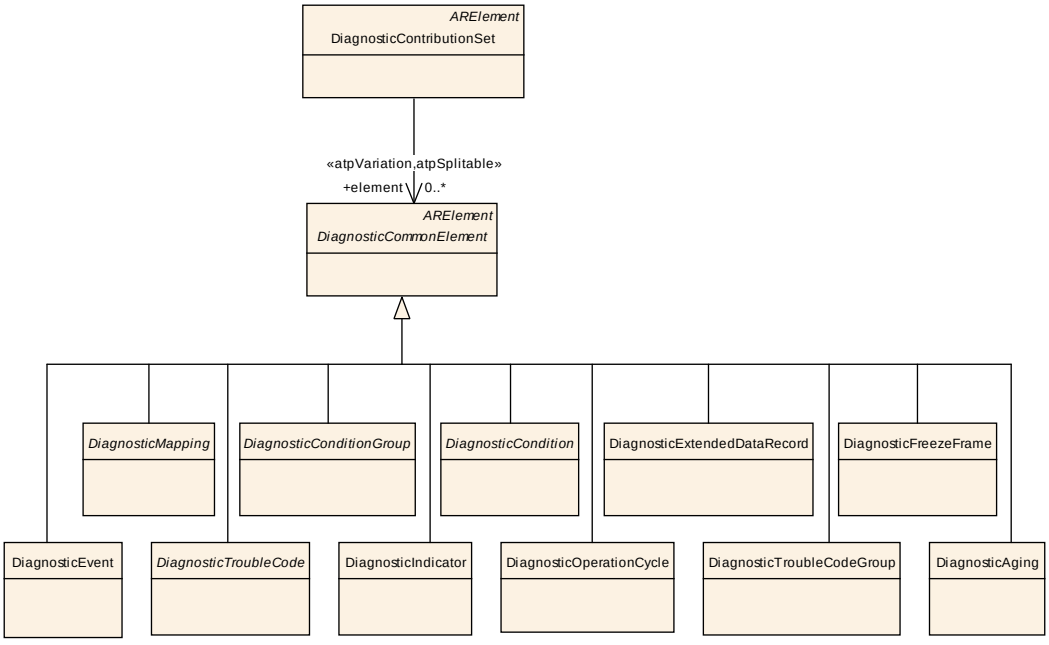

Figure 6.1: Common model elements relevant for the Dem

For the definition of the diagnostic event functionality, a number of model elements are derived from DiagnosticCommonElement. These elements are described in the following sub-chapters.

6.2 DiagnosticEvent

[TPS_DEXT_01083] Semantics of a `DiagnosticEvent` ⌈ A `DiagnosticEvent` - the atomic unit handled by the `Dem` module - has to be defined together with its properties which affect the event handling behavior and possible interfaces to software-components. ⌋*(RS_DEXT_00023)*

Figure 6.2 depicts the definition of `DiagnosticEvent` together with its properties.

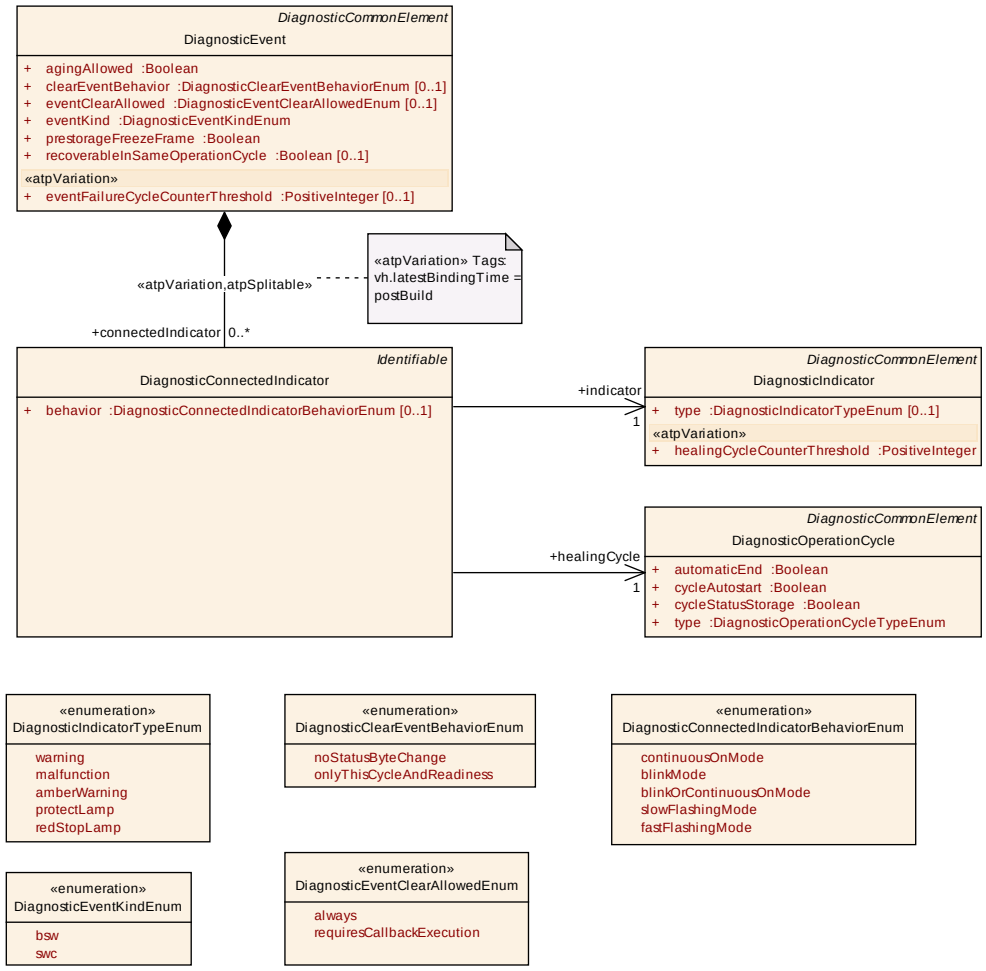

Figure 6.2: Modeling of DiagnosticEvent

The `DiagnosticExtract` allows the definition of an arbitrary number of `DiagnosticEvent`s.

Although the exchange of a `DiagnosticExtract` between companies usually involves `DiagnosticEvent`s related to `SWC` functionality, the event kind `BSW` is also supported in order to enable definition of handling of BSW events (e.g. definition of associated `DiagnosticTroubleCode`).

[TPS_DEXT_03011] Clearing request for a `DiagnosticEvent` ⌈ Furthermore, a clearing request for a `DiagnosticEvent` might require invocation of a callback to a `SWC` in order to allow or to prohibit the clearing operation.

The expectation on this callback interface can be expressed using the attribute `eventClearAllowed`:

- `always` indicates that a clearing request for the `DiagnosticEvent` **shall unconditionally be executed**.
- `never` denotes that a clearing for the `DiagnosticEvent` is **intentionally not possible**.
- In case of `requiresCallbackExecution`, the **execution of a callback shall decide whether the clearing is permitted**.

  In other words, the implementation of this decision is up to the developer of the corresponding `AtomicSwComponentType`.

  The latter shall define a `SwcServiceDependency` with appropriate `DiagnosticEventNeeds` and a `RoleBasedPortAssignment` where the value of the attribute `role` is set to `CallbackClearEventAllowed`.

⌋(*RS_DEXT_00023*)

Class	**DiagnosticEvent**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
Note	This element is used to configure DiagnosticEvents. **Tags:** atp.recommendedPackage=DiagnosticEvents			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
agingAllowed	Boolean	1	attr	This represents the decision whether aging is allowed for this DiagnosticEvent.
clearEventBehavior	DiagnosticClearEventBehaviorEnum	0..1	attr	This attribute defines the resulting UDS status byte for the related event, which shall not be cleared according to the ClearEventAllowed callback.
connectedIndicator	DiagnosticConnectedIndicator	*	aggr	Event specific description of Indicators. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
eventClearAllowed	DiagnosticEventClearAllowedEnum	0..1	attr	This attribute defines whether the Dem has access to a "ClearEventAllowed" callback.
eventFailureCycleCounterThreshold	PositiveInteger	0..1	attr	This attribute defines the number of failure cycles for the event based fault confirmation. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
eventKind	DiagnosticEventKindEnum	1	attr	This attribute is used to distinguish between SWC and BSW events.

prestorageFreezeFrame	Boolean	1	attr	This attribute describes whether the Prestorage of FreezeFrames is supported by the assigned event or not. True: Prestorage of FreezeFrames is supported False: Prestorage of FreezeFrames is not supported
recoverableInSameOperationCycle	Boolean	0..1	attr	If the attribute is set to true then reporting PASSED will reset the indication of a failed test in the current operation cycle. If the attribute is set to false then reporting PASSED will be ignored and not lead to a reset of the indication of a failed test.

Table 6.1: DiagnosticEvent

Class	**RoleBasedPortAssignment**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::SwcInternalBehavior::ServiceMapping			
Note	This class specifies an assignment of a role to a particular service port (RPortPrototype or PPortPrototype) of an AtomicSwComponentType. With this assignment, the role of the service port can be mapped to a specific ServiceNeeds element, so that a tool is able to create the correct connector.			
Base	ARObject			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
portPrototype	PortPrototype	1	ref	Service PortPrototype used in the assigned role. This PortPrototype shall either belong to the same AtomicSwComponentType as the SwcInternalBehavior which owns the ServiceDependency or to the same NvBlockSwComponentType as the NvBlockDescriptor.
role	Identifier	1	attr	This is the role of the assigned Port in the given context. The value shall be a shortName of the Blueprint of a PortInterface as standardized in the Software Specification of the related AUTOSAR Service.

Table 6.2: RoleBasedPortAssignment

[TPS_DEXT_01085] `DiagnosticEvent` can be connected to one or multiple indicators ⌈ A `DiagnosticEvent` can be connected to one or multiple indicators (modeled by means of aggregating `DiagnosticIndicator` in the role `connectedIndicator`) of a certain type and with certain behavior. ⌋*(RS_DEXT_00023)*

[TPS_DEXT_01067] Textually formulated content attached to `DiagnosticEvent` ⌈ The definition of a `DiagnosticEvent` also consists of textually formulated content that is formalized in structure but cannot be formalized in content.

The purpose of this content is to define e.g. a mature condition that relates to the specific `DiagnosticEvent`. ⌋*(RS_DEXT_00023, RS_DEXT_00045)*

[TPS_DEXT_01068] Textual description with respect to the `DiagnosticEvent` ⌈ Textual description that has the character of requirements with respect to the `DiagnosticEvent` shall be provided by means of the meta-class `StructuredReq`, i.e. by means of `introduction.structuredReq`. ⌋ *(RS_DEXT_00023, RS_DEXT_00045)*

Class	**StructuredReq**			
Package	M2::MSR::Documentation::BlockElements::RequirementsTracing			
Note	This represents a structured requirement. This is intended for a case where specific requirements for features are collected. Note that this can be rendered as a labeled list.			
Base	ARObject, DocumentViewSelectable, Identifiable, MultilanguageReferrable, Paginateable, Referrable, Traceable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
appliesTo	standardNameEnum	*	attr	This attribute represents the platform the requirement is assigned to. **Tags:** xml.namePlural=APPLIES-TO-DEPENDENCIES; xml.sequenceOffset=25
conflicts	DocumentationBlock	0..1	aggr	This represents an informal specification of conflicts. **Tags:** xml.sequenceOffset=40
date	DateTime	1	attr	This represents the date when the requirement was initiated. **Tags:** xml.sequenceOffset=5
dependencies	DocumentationBlock	0..1	aggr	This represents an informal specifiaction of dependencies. Note that upstream tracing should be formalized in the property trace provided by the superclass Traceable. **Tags:** xml.sequenceOffset=30
description	DocumentationBlock	0..1	aggr	Ths represents the general description of the requirement. **Tags:** xml.sequenceOffset=10
importance	String	1	attr	This allows to represent the importance of the requirement. **Tags:** xml.sequenceOffset=8
issuedBy	String	1	attr	This represents the person, organization or authority which issued the requirement. **Tags:** xml.sequenceOffset=6
rationale	DocumentationBlock	0..1	aggr	This represents the rationale of the requirement. **Tags:** xml.sequenceOffset=20

remark	Documentation Block	0..1	aggr	This represents an informal remark. Note that this is not modeled as annotation, since these remark is still essential part of the requirement. **Tags:** xml.sequenceOffset=60
supporting Material	Documentation Block	0..1	aggr	This represents an informal specifiaction of the supporting material. **Tags:** xml.sequenceOffset=50
testedItem	Traceable	*	ref	This assocation represents the ability to trace on the same specification level. This supports for example the of acceptance tests. **Tags:** xml.sequenceOffset=70
type	String	1	attr	This attribute allows to denote the type of requirement to denote for example is it an "enhancement", "new feature" etc. **Tags:** xml.sequenceOffset=7
useCase	Documentation Block	0..1	aggr	This describes the relevant use cases. Note that formal references to use cases should be done in the trace relation. **Tags:** xml.sequenceOffset=35

Table 6.3: StructuredReq

For more details regarding the modeling of the semi-formal text please refer to Figure 4.3.

[TPS_DEXT_01069] Standardized values of `DiagnosticEvent.introduction.structuredReq` ⌈ The following possible values of `DiagnosticEvent.introduction.structuredReq` are standardized by AUTOSAR:

- **DIAG_EVENT_MON_COND**: this value describes the monitoring condition of the corresponding `DiagnosticEvent`.
- **DIAG_EVENT_MON_TYPE**: this value describes the monitoring type of the corresponding `DiagnosticEvent`.
- **DIAG_EVENT_MON_RATE**: this value describes the monitoring rate for the corresponding `DiagnosticEvent`.
- **DIAG_EVENT_MAT_COND**: this value describes a mature condition of the `DiagnosticEvent`.
- **DIAG_EVENT_DEMAT_COND**: this value describes a de-mature condition of the `DiagnosticEvent`.
- **DIAG_EVENT_AGING**: this value describes the behavior of the `DiagnosticEvent` regarding aging.

- **DIAG_EVENT_LIMP_IN_ACT**: this value describes the associated limp-in action for the DiagnosticEvent.
- **DIAG_EVENT_MAT_TIME**: this value describes the mature time for the corresponding DiagnosticEvent, i.e. how long or how often the fault must exist.
- **DIAG_EVENT_DEMAT_TIME**: this value describes the de-mature time for the corresponding DiagnosticEvent, i.e. how long or how often must the OK conditions be fulfilled.

⌋(*RS_DEXT_00001*, *RS_DEXT_00023*, *RS_DEXT_00045*)

The following ARXML fragment exemplifies the usage of StructuredReq along with the standardized values of the attribute category to attach semi-formal textual descriptions to a DiagnosticEvent.

Listing 6.1: Example for the definition of a semi-formal textual elements in the context of a DiagnosticEvent

```
<DIAGNOSTIC-EVENT>
    <SHORT-NAME>ExampleEvent_0001</SHORT-NAME>
        <INTRODUCTION>
            <STRUCTURED-REQ>
                <SHORT-NAME>MatureCondition</SHORT-NAME>
                <CATEGORY>DIAG_EVENT_MAT_COND</CATEGORY>
                <DESCRIPTION>
                    <P>
                        <L-1 L="EN">This DTC is set if System Voltage is
                            below 9 Volts</L-1>
                    </P>
                </DESCRIPTION>
            </STRUCTURED-REQ>
            <STRUCTURED-REQ>
                <SHORT-NAME>DematureCondition</SHORT-NAME>
                <CATEGORY>DEMATURE_COND</CATEGORY>
                <DESCRIPTION>
                    <P>
                        <L-1 L="EN">This DTC is set if System Voltage is
                            above 10 Volts<XFILE><SHORT-NAME>
                            Requirement_Specification</SHORT-NAME><URL>http:
                            //autosar.org</URL></XFILE>
                        </L-1>
                    </P>
                </DESCRIPTION>
            </STRUCTURED-REQ>
        </INTRODUCTION>
    <CLEAR-EVENT-BEHAVIOR>ONLY-THIS-CYCLE-AND-READINESS</CLEAR-EVENT-
        BEHAVIOR>
    <EVENT-CLEAR-ALLOWED>ALWAYS</EVENT-CLEAR-ALLOWED>
    <EVENT-FAILURE-CYCLE-COUNTER-THRESHOLD>100</EVENT-FAILURE-CYCLE-COUNTER
        -THRESHOLD>
    <EVENT-KIND>SWC</EVENT-KIND>
    <PRESTORAGE-FREEZE-FRAME>false</PRESTORAGE-FREEZE-FRAME>
</DIAGNOSTIC-EVENT>
```

Class	**DiagnosticConnectedIndicator**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
Note	Description of indicators that are defined per DiagnosticEvent.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
behavior	DiagnosticConnectedIndicatorBehaviorEnum	0..1	attr	Behavior of the linked indicator.
healingCycle	DiagnosticOperationCycle	1	ref	The deactivation of indicators per event is defined as healing of a diagnostic event. The operation cycle in which the warning indicator will be switched off is defined here.
indicator	DiagnosticIndicator	1	ref	Reference to the used indicator.

Table 6.4: DiagnosticConnectedIndicator

Enumeration	**DiagnosticEventClearAllowedEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent
Note	Denotes whether clearing of events is allowed.
Literal	***Description***
always	The clearing is allowed unconditionally. **Tags:** atp.EnumerationValue=0
requires Callback Execution	In case the clearing of a Diagnostic Event has to be allowed or prohibited through the SWC interface CallbackClearEventAllowed, the SWC has to indicate this by defining appropriate ServiceNeeds (i.e. DiagnosticEventNeeds). **Tags:** atp.EnumerationValue=2

Table 6.5: DiagnosticEventClearAllowedEnum

Enumeration	**DiagnosticClearEventBehaviorEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent
Note	Possible behavior for clearing events.
Literal	***Description***
noStatusByte Change	The event status byte keeps unchanged. **Tags:** atp.EnumerationValue=0
onlyThis CycleAnd Readiness	The OperationCycle and readiness bits of the event status byte are reset. **Tags:** atp.EnumerationValue=1

Table 6.6: DiagnosticClearEventBehaviorEnum

Enumeration	**DiagnosticEventKindEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent
Note	Applicability of the diagnostic event.

Literal	*Description*
bsw	The event is assigned to a BSW module. **Tags:** atp.EnumerationValue=0
swc	The event is assigned to a SWC. **Tags:** atp.EnumerationValue=1

Table 6.7: DiagnosticEventKindEnum

Enumeration	**DiagnosticConnectedIndicatorBehaviorEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent
Note	Behavior of the indicator.
Literal	***Description***
blinkMode	The indicator blinks when the event has status FAILED. **Tags:** atp.EnumerationValue=0
blinkOrContinuousOnMode	The indicator is active and blinks when the event has status FAILED. **Tags:** atp.EnumerationValue=1
continuousOnMode	The indicator is active when the event has status FAILED. **Tags:** atp.EnumerationValue=2
fastFlashingMode	Flash Indicator Lamp should be set to "Fast Flash". **Tags:** atp.EnumerationValue=3
slowFlashingMode	Flash Indicator Lamp should be set to "Slow Flash". **Tags:** atp.EnumerationValue=4

Table 6.8: DiagnosticConnectedIndicatorBehaviorEnum

6.3 DiagnosticTroubleCode

DiagnosticTroubleCodes (i.e. the ECU external view on diagnostic events) are defined together with their properties and mapped to DiagnosticEvents using DiagnosticEventToTroubleCodeUdsMapping.

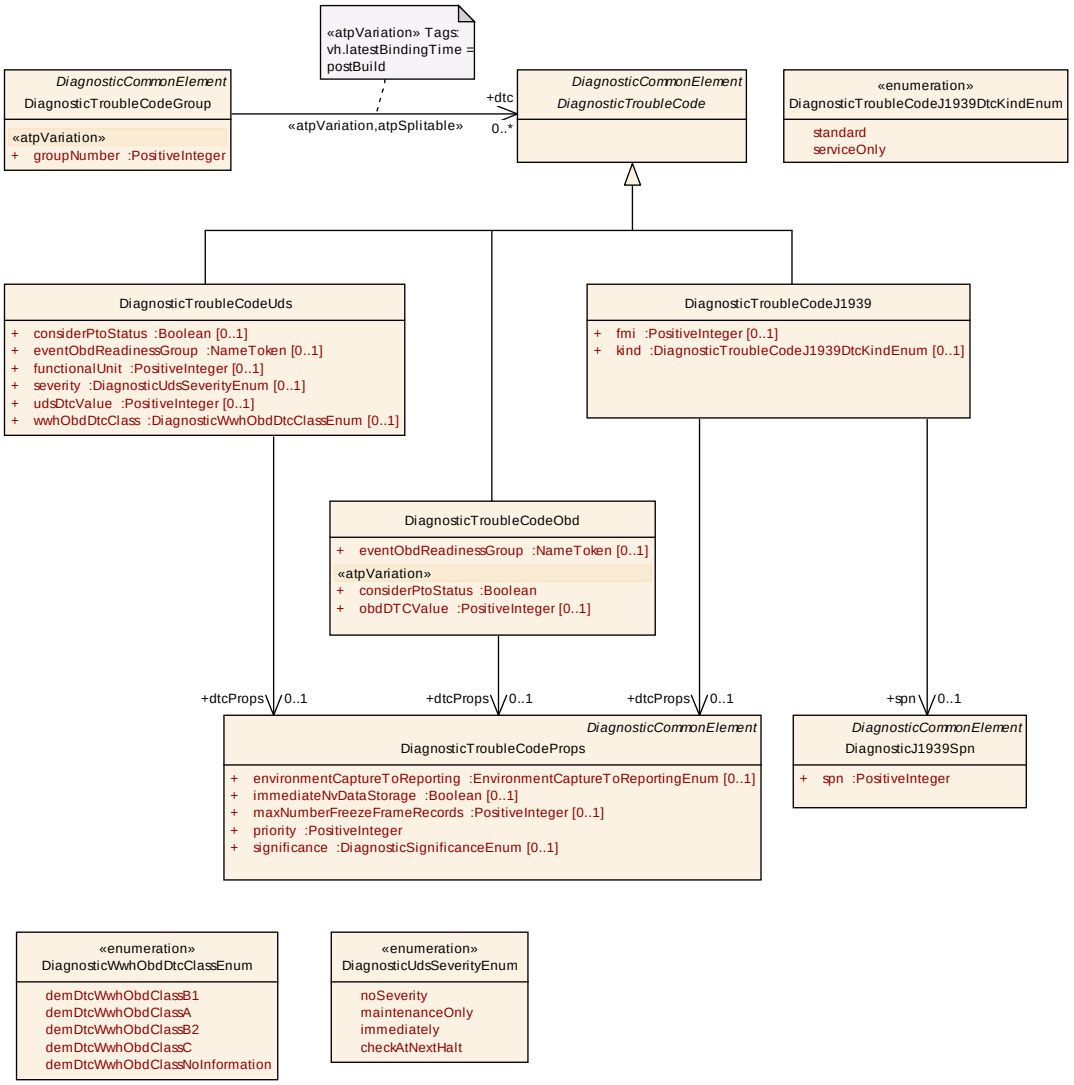

Figure 6.3: Modeling of `DiagnosticTroubleCode`

[TPS_DEXT_03012] Three kinds of DTCs ⌈ There are three kinds of DTCs represented as specializations of `DiagnosticTroubleCode`:

- non OBD relevant DTCs (`DiagnosticTroubleCodeUds`)
- OBD relevant DTCs (`DiagnosticTroubleCodeObd`)
- J1939 [20] relevant DTCs (`DiagnosticTroubleCodeJ1939`)

Properties individual to such a DTC specialization are modeled as attributes of `DiagnosticTroubleCodeUds`, `DiagnosticTroubleCodeObd` and `DiagnosticTroubleCodeJ1939`, respectively. ⌋*(RS_DEXT_00024)*

Class	**DiagnosticTroubleCodeUds**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This element is used to describe non OBD-relevant DTCs. **Tags:** atp.recommendedPackage=DiagnosticTroubleCodes			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticTroubleCode, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
considerPtoStatus	Boolean	0..1	attr	This attribute describes the affection of the event by the Dem PTO handling. True: the event is affected by the Dem PTO handling. False: the event is not affected by the Dem PTO handling.
dtcProps	DiagnosticTroubleCodeProps	0..1	ref	Defined properties associated with the DemDTC.
eventObdReadinessGroup	NameToken	0..1	attr	This attribute specifies the Event OBD Readiness group for PID $01 and PID $41 computation. This attribute is only applicable for emission-related ECUs.
functionalUnit	PositiveInteger	0..1	attr	This attribute specifies a 1-byte value which identifies the corresponding basic vehicle / system function which reports the DTC. This parameter is necessary for the report of severity information.
severity	DiagnosticUdsSeverityEnum	0..1	attr	DTC severity according to ISO 14229-1.
udsDtcValue	PositiveInteger	0..1	attr	Unique Diagnostic Trouble Code value for UDS.
wwhObdDtcClass	DiagnosticWwhObdDtcClassEnum	0..1	attr	This attribute is used to identify (if applicable) the corresponding severity class of an WWH-OBD DTC.

Table 6.9: DiagnosticTroubleCodeUds

Class	**DiagnosticTroubleCodeObd**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This element is used to define OBD-relevant DTCs. **Tags:** atp.recommendedPackage=DiagnosticTroubleCodes			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticTroubleCode, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
considerPtoStatus	Boolean	1	attr	This attribute describes the affection of the event by the Dem PTO handling. True: the event is affected by the Dem PTO handling. False: the event is not affected by the Dem PTO handling. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

dtcProps	DiagnosticTroubleCodeProps	0..1	ref	Defined properties associated with the DemDTC.
eventObdReadinessGroup	NameToken	0..1	attr	This attribute specifies the Event OBD Readiness group for PID $01 and PID $41 computation. This attribute is only applicable for emission-related ECUs.
obdDTCValue	PositiveInteger	0..1	attr	Unique Diagnostic Trouble Code value for OBD. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table 6.10: DiagnosticTroubleCodeObd

Class	**DiagnosticTroubleCodeJ1939**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This meta-class represents the ability to model specific trouble-code related properties for J1939. **Tags:** atp.recommendedPackage=DiagnosticTroubleCodes			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticTroubleCode, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dtcProps	DiagnosticTroubleCodeProps	0..1	ref	Defined properties associated with the J1939 DTC.
fmi	PositiveInteger	0..1	attr	This attribute represents the behavior of the Failure Mode Indicator.
kind	DiagnosticTroubleCodeJ1939DtcKindEnum	0..1	attr	This attribute further specifies the DTC in terms of its semantics.
node	DiagnosticJ1939Node	0..1	ref	This represents the related DiagnosticJ1939Node.
spn	DiagnosticJ1939Spn	0..1	ref	This represents the releated SPN.

Table 6.11: DiagnosticTroubleCodeJ1939

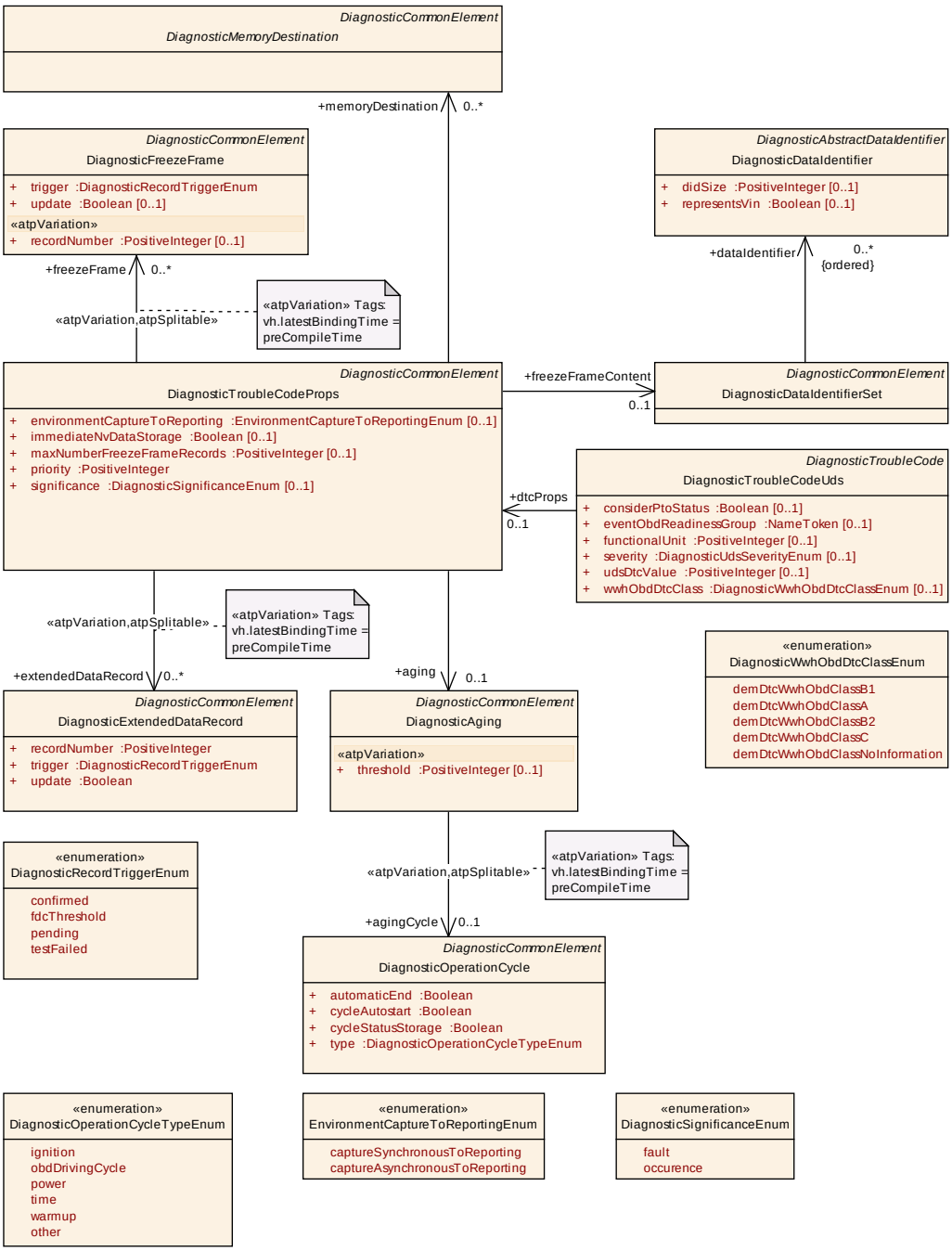

Figure 6.4: Modeling of `DiagnosticTroubleCodeUds`

[TPS_DEXT_03013] Common properties of a DTC ⌈ Properties that are often common for a group of `DiagnosticTroubleCodeUds` elements are modeled as attributes of `DiagnosticTroubleCodeProps`. ⌋*(RS_DEXT_00024)*

[constr_1349] Value of `udsDtcValue` shall be unique ⌈ The value of `udsDtcValue` shall be unique to any other DTC and DTC group value. ⌋*()*

Class	**DiagnosticTroubleCode (abstract)**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	A diagnostic trouble code defines a unique identifier that is shown to the diagnostic tester.			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 6.12: DiagnosticTroubleCode

[TPS_DEXT_03014] Semantics of `DiagnosticTroubleCodeGroup` ⌈ The `DiagnosticTroubleCodeGroup` element is used to define groups of DTCs that belong together. Each `DiagnosticTroubleCodeGroup` has its own `groupNumber` value assigned. ⌋*(RS_DEXT_00024)*

Class	**DiagnosticTroubleCodeGroup**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	The diagnostic trouble code group defines the DTCs belonging together and thereby forming a group. **Tags:** atp.recommendedPackage=DiagnosticTroubleCodes			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dtc	DiagnosticTroubleCode	*	ref	This represents the collection of DiagnosticTroubleCodes defined by this DiagnosticTroubleCodeGroup. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dtc, variationPoint.shortLabel vh.latestBindingTime=postBuild
groupNumber	PositiveInteger	1	attr	This represents the base number of the DTC group. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table 6.13: DiagnosticTroubleCodeGroup

[constr_1350] Value of `DiagnosticTroubleCodeGroup.groupNumber` shall be unique ⌈ The value of `DiagnosticTroubleCodeGroup.groupNumber` shall be unique to any other DTC and DTC group value. ⌋*()*

[constr_1351] Value of `DiagnosticTroubleCodeGroup.groupNumber` ⌈ To be compliant to ISO, the value of `DiagnosticTroubleCodeGroup.groupNumber` shall be set as defined in ISO 14229-1 [16]. ⌋*()*

[TPS_DEXT_03000] ISO 14229-1 reserves values of `DiagnosticTroubleCodeGroup.groupNumber` ⌈ Any values other than those mentioned in [constr_1351] are reserved by ISO 14229-1 [16]. ⌋*(RS_DEXT_00024)*

[constr_1352] Existence of `maxNumberFreezeFrameRecords` vs. `freezeFrame` ⌈ If the attribute `DiagnosticTroubleCodeProps.maxNumberFreezeFrameRecords` exists than the attribute `DiagnosticTroubleCodeProps.freezeFrame` shall not exist or vice versa. ⌋*()*

[constr_1353] Applicability of [1352] ⌈ [constr_1352] shall apply in the identical way (either one or the other attribute shall exist) for all `DiagnosticTroubleCodeProps` within the context of all `DiagnosticContributionSet`s of `category` `DIAGNOSTIC_ECU_EXTRACT` that refer to the same `EcuInstance`. ⌋*()*

[constr_1354] Existence of attribute `DiagnosticTroubleCodeProps.freezeFrameContent` ⌈ If one of the attributes `DiagnosticTroubleCodeProps.maxNumberFreezeFrameRecords` or `DiagnosticTroubleCodeProps.freezeFrame` exists then the attribute `DiagnosticTroubleCodeProps.freezeFrameContent` shall exist. ⌋*()*

[TPS_DEXT_01064] Textually formulated content attached to `DiagnosticTroubleCode` ⌈ The definition of a `DiagnosticTroubleCode` also consists of textually formulated content that is formalized in structure but cannot be formalized in content.

The purpose of this content is to define e.g. an error text or the possible cause that relates to the specific `DiagnosticTroubleCode`. ⌋*(RS_DEXT_00024, RS_DEXT_00045)*

[TPS_DEXT_01065] Different approaches to provide semi-formal textual content attached to a `DiagnosticTroubleCode` ⌈ There are different approaches to provide semi-formal textual content attached to a `DiagnosticTroubleCode`:

- Textual description that has the character of descriptions of the `DiagnosticTroubleCode` shall be provided by means of the meta-class `TraceableText`, i.e. by means of `introduction.trace`.
- Textual description that characterizes the `DiagnosticTroubleCode` with respect to the *ODX long name* shall be provided by means of the attribute `longName`.

⌋*(RS_DEXT_00024, RS_DEXT_00045)*

Class	**TraceableText**			
Package	M2::MSR::Documentation::BlockElements::RequirementsTracing			
Note	This meta-class represents the ability to denote a traceable text item such as requirements etc. The following approach appliles: • **shortName** represents the tag for tracing • **longName** represents the head line • **category** represents the kind of the tagged text			
Base	ARObject, DocumentViewSelectable, Identifiable, MultilanguageReferrable, Paginateable, Referrable, Traceable			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
text	Documentation Block	1	aggr	This represents the text to which the tag applies. **Tags:** xml.roleElement=false; xml.roleWrapper Element=false; xml.sequenceOffset=30; xml.type Element=false; xml.typeWrapperElement=false

Table 6.14: TraceableText

Class	**MultilanguageReferrable (abstract)**			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable			
Note	Instances of this class can be referred to by their identifier (while adhering to namespace borders). They also may have a longName. But they are not considered to contribute substantially to the overall structure of an AUTOSAR description. In particular it does not contain other Referrables.			
Base	ARObject, Referrable			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
longName	MultilanguageLongName	0..1	aggr	This specifies the long name of the object. Long name is targeted to human readers and acts like a headline.

Table 6.15: MultilanguageReferrable

For more details regarding the modeling of the semi-formal text please refer to Figure 4.3.

The usage of `TraceableText` and `StructuredReq` alone would not qualify as a semi-formal textual attachment. It is necessary to standardize the value of the `category` in order to get some level of semi-formal textual description.

[TPS_DEXT_01066] Standardized values of `DiagnosticTroubleCode.introduction.trace` ⌈ The following possible values of `DiagnosticTroubleCode.introduction.trace` are standardized by AUTOSAR:

- **DIAG_DTC_ERROR_TEXT**: this value shall be used to describe an error text.
- **DIAG_DTC_REP_ACT**: this value describes the associated repair for the corresponding `DiagnosticTroubleCode`.

- **DIAG_DTC_CUS_PER_SYMP**: this value describes the possible customer perception symptom for the corresponding DiagnosticTroubleCode.
- **DIAG_DTC_POSS_CAUSE**: This value describes the possible cause for the corresponding DiagnosticTroubleCode.

⌋(*RS_DEXT_00001*, *RS_DEXT_00024*, *RS_DEXT_00045*)

The following ARXML fragment exemplifies the usage of TraceableText along with the standardized values of the attribute category to attach semi-formal textual descriptions to a DiagnosticTroubleCodeUds.

Listing 6.2: Example for the definition of a semi-formal textual elements in the context of a DiagnosticTroubleCode

```
<DIAGNOSTIC-TROUBLE-CODE-UDS>
    <SHORT-NAME>ExampleDTC_0001</SHORT-NAME>
    <LONG-NAME>
        <L-4 L="EN">My little ODX long name</L-4>
    </LONG-NAME>
    <DESC>
        <L-2 L="EN">This DTC is a System Error DTC</L-2>
    </DESC>
    <INTRODUCTION>
        <TRACE>
            <SHORT-NAME>MyErrorText</SHORT-NAME>
            <CATEGORY>DIAG_DTC_ERROR_TEXT</CATEGORY>
            <P>
                <L-1 L="LA">Lorem ipsum dolor sit amet, consectetur
                    adipisicing elit</L-1>
            </P>
        </TRACE>
    </INTRODUCTION>
    <DTC-PROPS-REF DEST="DIAGNOSTIC-TROUBLE-CODE-PROPS">/AUTOSAR/
       UseCase_230/ExampleDTC_0001_Props</DTC-PROPS-REF>
    <FUNCTIONAL-UNIT>1</FUNCTIONAL-UNIT>
    <SEVERITY>CHECK-AT-NEXT-HALT</SEVERITY>
    <UDS-DTC-VALUE>0x000001</UDS-DTC-VALUE>
</DIAGNOSTIC-TROUBLE-CODE-UDS>
```

[constr_1376] Multiplicity of reference DiagnosticTroubleCodeProps.memoryDestination ⌈ For every given DiagnosticTroubleCodeProps, the reference in the role DiagnosticTroubleCodeProps.memoryDestination **shall not exceed** the upper multiplicity 2. [constr_1377] applies. ⌋()

[constr_1377] Existence of reference DiagnosticTroubleCodeProps.memoryDestination ⌈ The reference DiagnosticTroubleCodeProps.memoryDestination shall **only** have the upper multiplicity 2 **if one (and only one) of** the referenced DiagnosticTroubleCodeProps.memoryDestination is a DiagnosticMemoryDestinationMirror. ⌋()

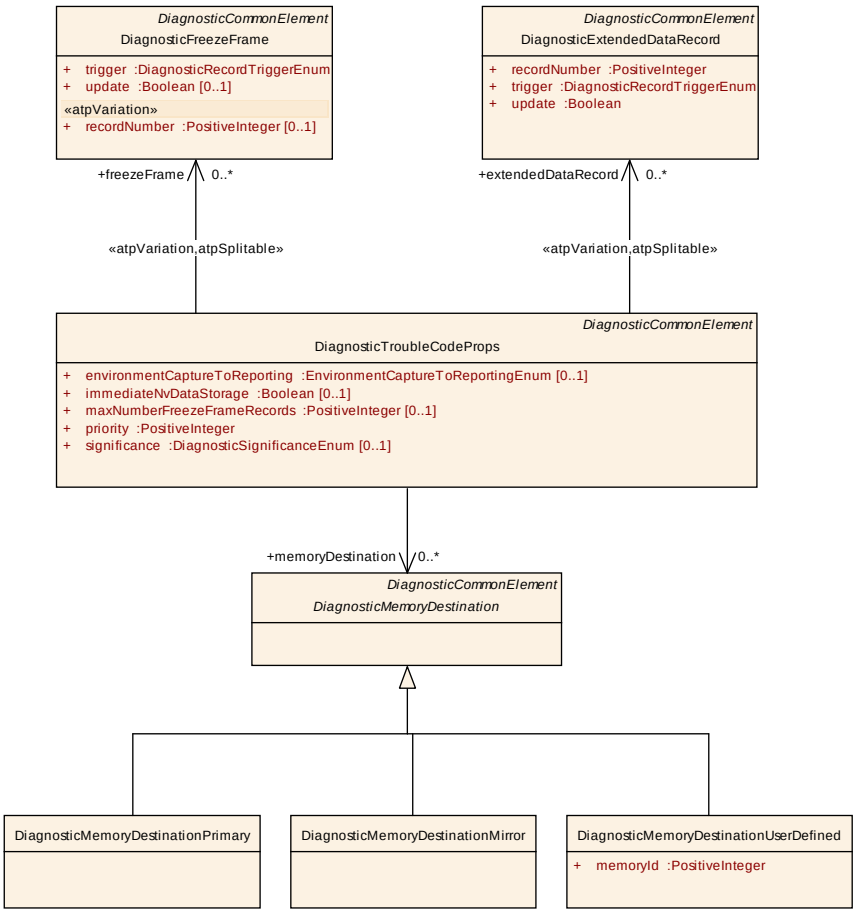

Figure 6.5: Modeling of `DiagnosticMemoryDestination`

[constr_1378] Value of `DiagnosticMemoryDestinationUserDefined.memoryId` ⌈ Within the scope of one `DiagnosticContributionSet`, no two (or more) `DiagnosticMemoryDestinationUserDefined`s shall exist that share the same value for attribute `DiagnosticMemoryDestinationUserDefined.memoryId` ⌋*()*

In other words, the value of the attribute `DiagnosticMemoryDestinationUserDefined.memoryId` shall be unique within any given `DiagnosticExtract`.

On top of that, it is necessary to make sure that only **one** *primary memory* and only **one** *mirror memory* is defined.

[constr_1379] Existence of `DiagnosticMemoryDestinationPrimary` ⌈ Within the scope of one `DiagnosticContributionSet` only one `DiagnosticMemoryDestinationPrimary` shall exist. ⌋*()*

[constr_1380] Existence of `DiagnosticMemoryDestinationMirror` ⌈ Within the scope of one `DiagnosticContributionSet` only one `DiagnosticMemoryDestinationMirror` shall exist. ⌋*()*

[TPS_DEXT_01094] Semantics of meta-class `DiagnosticTroubleCodeUdsToTroubleCodeObdMapping` ⌈ The meta-class `DiagnosticTroubleCodeUdsToTroubleCodeObdMapping` can be used to associate a `DiagnosticTroubleCodeUds` with a `DiagnosticTroubleCodeObd`. ⌋*()*

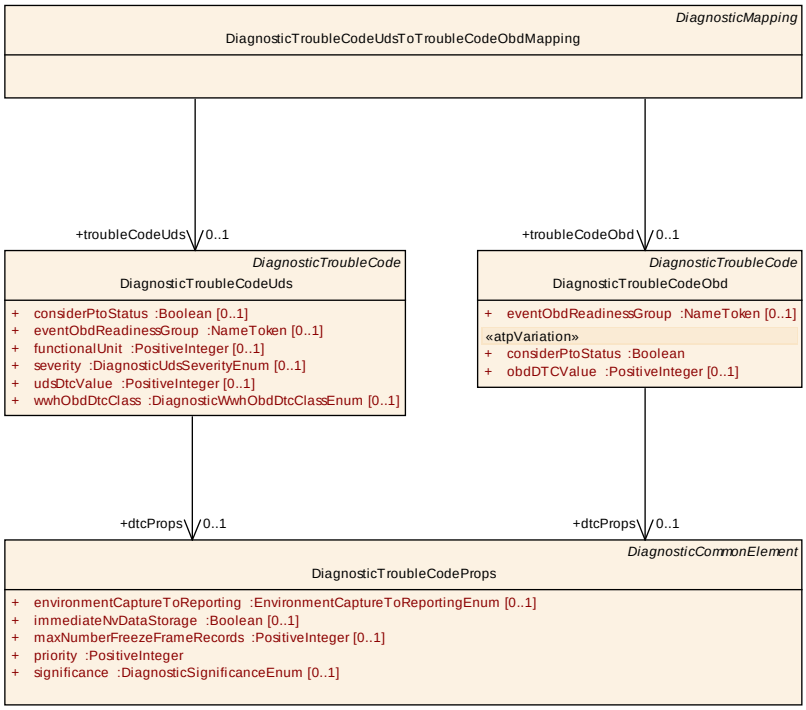

Figure 6.6: Mapping of UDS DTC to OBD DTC

Class	**DiagnosticTroubleCodeUds**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This element is used to describe non OBD-relevant DTCs. **Tags:** atp.recommendedPackage=DiagnosticTroubleCodes			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticTroubleCode, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
considerPtoStatus	Boolean	0..1	attr	This attribute describes the affection of the event by the Dem PTO handling. True: the event is affected by the Dem PTO handling. False: the event is not affected by the Dem PTO handling.
dtcProps	DiagnosticTroubleCodeProps	0..1	ref	Defined properties associated with the DemDTC.
eventObdReadinessGroup	NameToken	0..1	attr	This attribute specifies the Event OBD Readiness group for PID $01 and PID $41 computation. This attribute is only applicable for emission-related ECUs.
functionalUnit	PositiveInteger	0..1	attr	This attribute specifies a 1-byte value which identifies the corresponding basic vehicle / system function which reports the DTC. This parameter is necessary for the report of severity information.
severity	DiagnosticUdsSeverityEnum	0..1	attr	DTC severity according to ISO 14229-1.
udsDtcValue	PositiveInteger	0..1	attr	Unique Diagnostic Trouble Code value for UDS.

wwhObdDtcClass	DiagnosticWwhObdDtcClassEnum	0..1	attr	This attribute is used to identify (if applicable) the corresponding severity class of an WWH-OBD DTC.

Table 6.16: DiagnosticTroubleCodeUds

Class	**DiagnosticTroubleCodeObd**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This element is used to define OBD-relevant DTCs. **Tags:** atp.recommendedPackage=DiagnosticTroubleCodes			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticTroubleCode, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
considerPtoStatus	Boolean	1	attr	This attribute describes the affection of the event by the Dem PTO handling. True: the event is affected by the Dem PTO handling. False: the event is not affected by the Dem PTO handling. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
dtcProps	DiagnosticTroubleCodeProps	0..1	ref	Defined properties associated with the DemDTC.
eventObdReadinessGroup	NameToken	0..1	attr	This attribute specifies the Event OBD Readiness group for PID $01 and PID $41 computation. This attribute is only applicable for emission-related ECUs.
obdDTCValue	PositiveInteger	0..1	attr	Unique Diagnostic Trouble Code value for OBD. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table 6.17: DiagnosticTroubleCodeObd

Class	**DiagnosticTroubleCodeProps**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This element defines common Dtc properties that can be reused by different non OBD-relevant DTCs. **Tags:** atp.recommendedPackage=DiagnosticTroubleCodePropss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
aging	DiagnosticAging	0..1	ref	Reference to an aging algorithm in case that an aging/unlearning of the event is allowed.
environmentCaptureToReporting	EnvironmentCaptureToReportingEnum	0..1	attr	This attribute determines the point in time, when the data actually is captured.

extendedDataRecord	DiagnosticExtendedDataRecord	*	ref	Defines the links to an extended data class sampler. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
freezeFrame	DiagnosticFreezeFrame	*	ref	Define the links to a freeze frame class sampler. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
freezeFrameContent	DiagnosticDataIdentifierSet	0..1	ref	This represents the content of the a set of DiagnosticFreezeFrames.
immediateNvDataStorage	Boolean	0..1	attr	Switch to enable immediate storage triggering of an according event memory entry persistently to NVRAM. true: immediate non-volatile storage triggering enabled false: immediate non-volatile storage triggering disabled
maxNumberFreezeFrameRecords	PositiveInteger	0..1	attr	This attribute defines the number of according freeze frame records, which can maximal be stored for this event. Therefore all these freeze frame records have the same freeze frame class.
memoryDestination	DiagnosticMemoryDestination	*	ref	The event destination assigns events to none, one or multiple origins.
priority	PositiveInteger	1	attr	Priority of the event, in view of full event buffer. A lower value means higher priority.
significance	DiagnosticSignificanceEnum	0..1	attr	Significance of the event, which indicates additional information concerning fault classification and resolution.

Table 6.18: DiagnosticTroubleCodeProps

Class	**DiagnosticMemoryDestination (abstract)**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This abstract meta-class represents a possible memory destination for a diagnostic event.			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 6.19: DiagnosticMemoryDestination

Class	**DiagnosticMemoryDestinationPrimary**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This represents a primary memory for a diagnostic event. **Tags:** atp.recommendedPackage=DiagnosticMemoryDestinations			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMemoryDestination, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 6.20: DiagnosticMemoryDestinationPrimary

Class	**DiagnosticMemoryDestinationMirror**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This represents a mirror memory for a diagnostic event. **Tags:** atp.recommendedPackage=DiagnosticMemoryDestinations			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMemoryDestination, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 6.21: DiagnosticMemoryDestinationMirror

Class	**DiagnosticMemoryDestinationUserDefined**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This represents a user-defined memory for a diagnostic event. **Tags:** atp.recommendedPackage=DiagnosticMemoryDestinations			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMemoryDestination, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
memoryId	PositiveInteger	1	attr	This represents the identifier of the user-defined memory.

Table 6.22: DiagnosticMemoryDestinationUserDefined

Enumeration	**DiagnosticSignificanceEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode
Note	Significance level of a diagnostic event.
Literal	***Description***
fault	Failure, which affects the component/ECU itself. **Tags:** atp.EnumerationValue=0

occurence	Issue, which indicates additional information concerning insufficient system behavior. **Tags:** atp.EnumerationValue=1

Table 6.23: DiagnosticSignificanceEnum

Enumeration	**DiagnosticUdsSeverityEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode
Note	Severity types for a DTC according to ISO 14229-1.
Literal	***Description***
checkAtNext Halt	Check at next halt. **Tags:** atp.EnumerationValue=0
immediately	Check immediately. **Tags:** atp.EnumerationValue=1
maintenance Only	Maintenance required. **Tags:** atp.EnumerationValue=2
noSeverity	No severity information available. **Tags:** atp.EnumerationValue=3

Table 6.24: DiagnosticUdsSeverityEnum

Class	**DiagnosticDataIdentifierSet**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This represents the ability to define a list of DiagnosticDataIdentifiers that can be reused in different contexts. **Tags:** atp.recommendedPackage=DiagnosticDataIdentifierSets			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataId entifier (ordered)	DiagnosticDataI dentifier	*	ref	Reference to an orderd list of Data Identifiers.

Table 6.25: DiagnosticDataIdentifierSet

Enumeration	**DiagnosticWwhObdDtcClassEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode
Note	This meta-class represents the ability to model severity classes of an WWH-OBD DTC.
Literal	***Description***

demDtcWwhObdClassA	This attribute represents the severity class A. **Tags:** atp.EnumerationValue=0
demDtcWwhObdClassB1	This attribute represents the severity class B1. **Tags:** atp.EnumerationValue=1
demDtcWwhObdClassB2	This attribute represents the severity class B2. **Tags:** atp.EnumerationValue=2
demDtcWwhObdClassC	This attribute represents the severity class C. **Tags:** atp.EnumerationValue=3
demDtcWwhObdClassNoInformation	This attribute represents the option to intentionally not describe a dedicated severity class of an WWH-OBD DTC. **Tags:** atp.EnumerationValue=4

Table 6.26: DiagnosticWwhObdDtcClassEnum

Class	**DiagnosticTroubleCodeUdsToTroubleCodeObdMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
Note	This meta-class represents the ability to associate a UDS trouble code to an OBD trouble code. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
troubleCodeObd	DiagnosticTroubleCodeObd	0..1	ref	This represents the OBD DTC referenced in the mapping between UDS and OBD DTCs.
troubleCodeUds	DiagnosticTroubleCodeUds	0..1	ref	This represents the UDS DTC referenced in the mapping between UDS and OBD DTCs.

Table 6.27: DiagnosticTroubleCodeUdsToTroubleCodeObdMapping

6.4 DiagnosticExtendedDataRecord

[TPS_DEXT_03008] Semantics of `DiagnosticExtendedDataRecord` ⌈ A `DiagnosticExtendedDataRecord` contains `DiagnosticDataElement`s that are ordered by the `bitOffset`. ⌋*(RS_DEXT_00032)*

[constr_1355] Value of `extendedDataRecord.recordNumber` ⌈ To be compliant to ISO, the value of `extendedDataRecord.recordNumber` shall be set in the interval as defined in ISO 14229-1 [16]. ⌋*()*

[constr_1509] `extendedDataRecord.recordNumber` shall be unique within primary fault memory ⌈ For all `DiagnosticTroubleCodeProps` that refer to `Diag-`

nosticMemoryDestinationPrimary in the role memoryDestination there shall be no two extendedDataRecord.recordNumber with the same value. ⌋()

[constr_1510] extendedDataRecord.recordNumber shall be unique within mirror fault memory ⌈ For all DiagnosticTroubleCodeProps that refer to DiagnosticMemoryDestinationMirror in the role memoryDestination there shall be no two extendedDataRecord.recordNumber with the same value. ⌋()

[constr_1511] extendedDataRecord.recordNumber shall be unique within user-defined fault memory ⌈ For all DiagnosticTroubleCodeProps that refer to DiagnosticMemoryDestinationUserDefined in the role memoryDestination there shall be no two extendedDataRecord.recordNumber with the same value for any DiagnosticMemoryDestinationUserDefined referenced as DiagnosticTroubleCodeProps.memoryDestination with a given value of memoryId. ⌋()

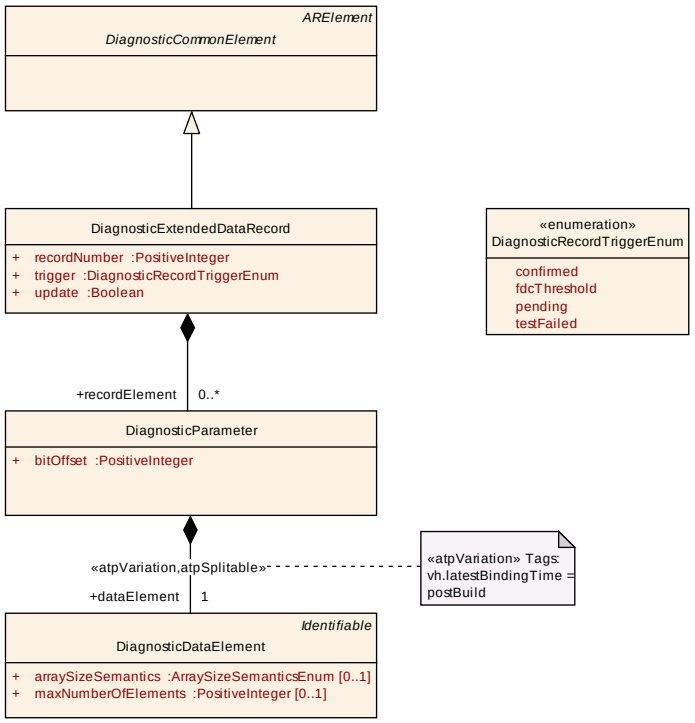

Figure 6.7: Modeling of DiagnosticExtendedDataRecord

Class	**DiagnosticExtendedDataRecord**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticExtendedDataRecord			
Note	Description of an extended data record. **Tags:** atp.recommendedPackage=DiagnosticExtendedDataRecords			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
recordElement	DiagnosticParameter	*	aggr	Defined DataElements in the extended record element.

recordNumber	PositiveInteger	1	attr	This attribute specifies an unique identifier for an extended data record.
trigger	DiagnosticRecordTriggerEnum	1	attr	This attribute specifies the primary trigger to allocate an event memory entry.
update	Boolean	1	attr	This attribute defines when an extended data record is captured. True: This extended data record is captured every time. False: This extended data record is only captured for new event memory entries.

Table 6.28: DiagnosticExtendedDataRecord

Enumeration	**DiagnosticRecordTriggerEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticFreezeFrame
Note	Triggers to allocate an event memory entry.
Literal	***Description***
confirmed	capture on "Confirmed" **Tags:** atp.EnumerationValue=0
fdcThreshold	capture on "FDC Threshold" **Tags:** atp.EnumerationValue=1
pending	capture on "Pending" **Tags:** atp.EnumerationValue=2
testFailed	capture on "Test Failed" **Tags:** atp.EnumerationValue=3

Table 6.29: DiagnosticRecordTriggerEnum

6.5 DiagnosticFreezeFrame

[TPS_DEXT_03009] Semantics of `DiagnosticFreezeFrame` ⌈ A `DiagnosticFreezeFrame` needs an ordered list of references to `DiagnosticDataIdentifier`s. However, this reference is not modeled directly but in the context of meta-class `DiagnosticTroubleCodeProps`. ⌋*(RS_DEXT_00033)*

For more details, please refer to Figure 6.3.

[constr_1357] Value of `freezeFrame.recordNumber` ⌈ To be compliant to ISO, the value of `freezeFrame.recordNumber` shall be set in the interval as defined in ISO 14229-1 [16]. ⌋*()*

[constr_1512] `freezeFrame.recordNumber` shall be unique within primary fault memory ⌈ For all `DiagnosticTroubleCodeProps` that refer to `DiagnosticMemoryDestinationPrimary` in the role `memoryDestination` there shall be no two `freezeFrame.recordNumber` with the same value. ⌋*()*

[constr_1513] `freezeFrame.recordNumber` shall be unique within mirror fault memory ⌈ For all `DiagnosticTroubleCodeProps` that refer to `DiagnosticMemoryDestinationMirror` in the role `memoryDestination` there shall be no two `freezeFrame.recordNumber` with the same value. ⌋()

[constr_1514] `freezeFrame.recordNumber` shall be unique within user-defined fault memory ⌈ For all `DiagnosticTroubleCodeProps` that refer to `DiagnosticMemoryDestinationUserDefined` in the role `memoryDestination` there shall be no two `freezeFrame.recordNumber` with the same value for any `DiagnosticMemoryDestinationUserDefined` referenced as `DiagnosticTroubleCodeProps.memoryDestination` with a given value of `memoryId`. ⌋()

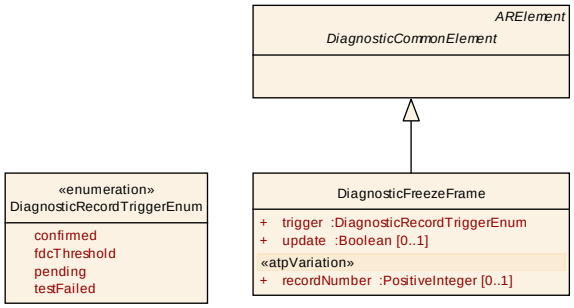

Figure 6.8: Modeling of DiagnosticFreezeFrame

Class	**DiagnosticFreezeFrame**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticFreezeFrame			
Note	This element describes combinations of DIDs for a non OBD relevant freeze frame. **Tags:** atp.recommendedPackage=DiagnosticFreezeFrames			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
recordNumber	PositiveInteger	0..1	attr	This attribute defines a record number for a freeze frame record. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
trigger	DiagnosticRecordTriggerEnum	1	attr	This attribute defines the primary trigger to allocate an event memory entry.
update	Boolean	0..1	attr	This attribute defines the approach when the freeze frame record is stored/updated. True: FreezeFrame record is captured every time. False: FreezeFrame record is only captured for new event memory entries.

Table 6.30: DiagnosticFreezeFrame

6.6 DiagnosticCondition

[TPS_DEXT_03010] Combination of DiagnosticConditions to DiagnosticConditionGroups ⌈ DiagnosticConditions are combined to DiagnosticConditionGroups and define a certain number of checks (e.g. correct voltage range) before the event report is accepted or the event gets qualified. ⌋*(RS_DEXT_00027, RS_DEXT_00028, RS_DEXT_00030, RS_DEXT_00031)*

[TPS_DEXT_03001] Different types of conditions ⌈ There are two different types of conditions: DiagnosticEnableConditions and DiagnosticStorageCondition:

- As long as the DiagnosticEnableCondition is not fulfilled, the event reports are not valid and therefore will not be accepted.
- As long as the DiagnosticStorageCondition is not fulfilled, the event is not stored in the event memory.

⌋*(RS_DEXT_00027)*

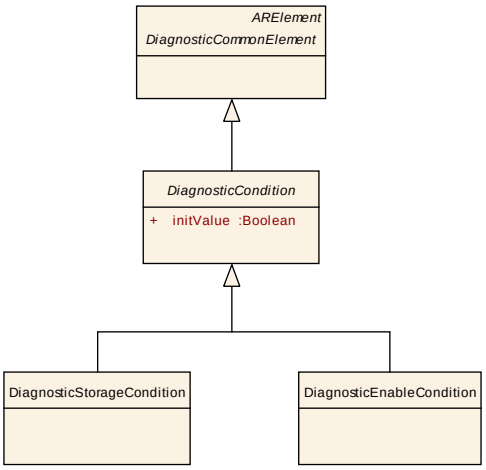

Figure 6.9: Modeling of DiagnosticCondition

Class	**DiagnosticCondition (abstract)**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticCondition			
Note	Abstract element for StorageConditions and EnableConditions.			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
initValue	Boolean	1	attr	Defines the initial status for enable or disable of acceptance/storage of event reports of a diagnostic event. The value is the initialization after power up (before this condition is reported the first time). true: acceptance/storage of a diagnostic event enabled false: acceptance/storage of a diagnostic event disabled

Table 6.31: DiagnosticCondition

Class	**DiagnosticEnableCondition**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticCondition			
Note	Specification of an enable condition. **Tags:** atp.recommendedPackage=DiagnosticConditions			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticCondition, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 6.32: DiagnosticEnableCondition

Class	**DiagnosticStorageCondition**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticCondition			
Note	Specification of a storage condition. **Tags:** atp.recommendedPackage=DiagnosticConditions			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticCondition, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 6.33: DiagnosticStorageCondition

6.7 DiagnosticConditionGroup

[TPS_DEXT_01084] Semantics of `DiagnosticConditionGroup`s ⌈ `DiagnosticConditionGroup`s are used to collect `DiagnosticCondition`s that in turn are assigned to `DiagnosticEvent`s. ⌋*(RS_DEXT_00023, RS_DEXT_00028, RS_DEXT_00029)*

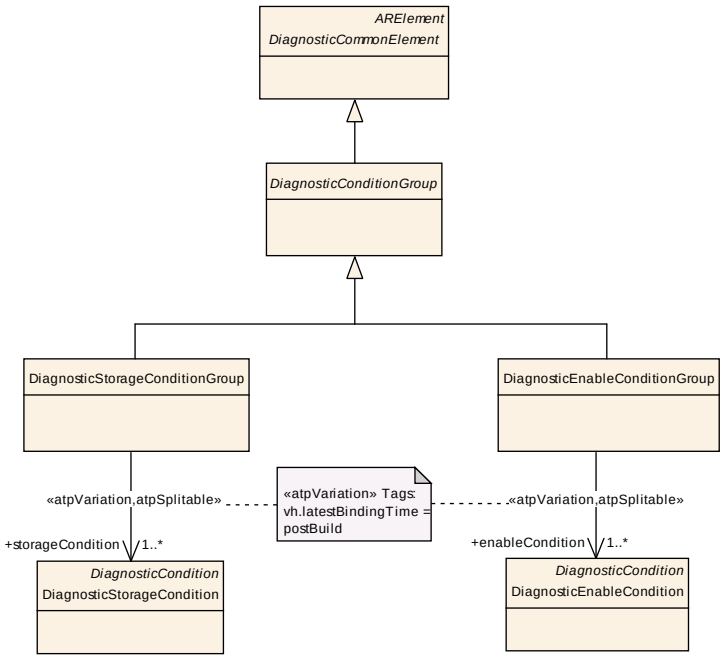

Figure 6.10: Modeling of DiagnosticConditionGroup

Class	**DiagnosticConditionGroup (abstract)**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticConditionGroup			
Note	Abstract element for StorageConditionGroups and EnableConditionGroups.			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 6.34: DiagnosticConditionGroup

Class	**DiagnosticEnableConditionGroup**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticConditionGroup			
Note	Enable condition group which includes one or several enable conditions. **Tags:** atp.recommendedPackage=DiagnosticConditions			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticConditionGroup, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
enableCondition	DiagnosticEnableCondition	1..*	ref	Reference to enableConditions that are part of the EnableConditionGroup. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=enableCondition, variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 6.35: DiagnosticEnableConditionGroup

Class	**DiagnosticStorageConditionGroup**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticConditionGroup			
Note	Storage condition group which includes one or several storage conditions. **Tags:** atp.recommendedPackage=DiagnosticConditions			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticConditionGroup, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
storageCondition	DiagnosticStorageCondition	1..*	ref	Reference to storageConditions that are part of the StorageConditionGroup. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=storageCondition, variationPoint.shortLabel vh.latestBindingTime=postBuild

Table 6.36: DiagnosticStorageConditionGroup

6.8 DiagnosticMapping

The mapping concept of the `DiagnosticExtract` template has been designed to support the decentralized and independent definition of diagnostic requirements that can be linked together at a late point during the development process.

It also supports the use of mapping contributions collected from various sources in order to reduce manual mapping work by the ECU integrator.

[TPS_DEXT_03002] Two kind of mappings ⌈ For diagnostic event handling, there are two kind of mappings:

- Mapping between a `DiagnosticEvent` and another diagnostic definition.
- Mapping between a `DiagnosticEvent` and a SWC service port.

⌋*(RS_DEXT_00023, RS_DEXT_00052)*

Figure 6.11 gives an overview on the different types of mappings available for diagnostic event handling.

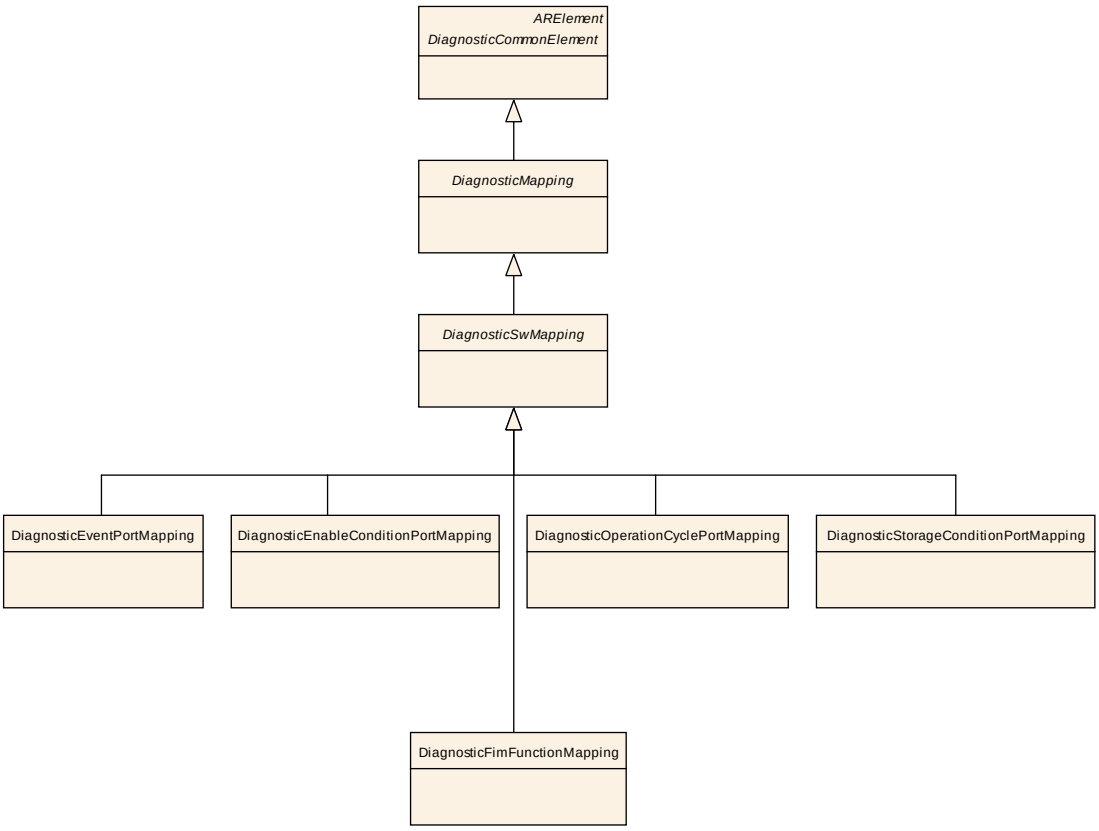

Figure 6.11: Modeling of DiagnosticMapping

Class	**DiagnosticMapping (abstract)**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
Note	Abstract element for different kinds of diagnostic mappings.			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 6.37: DiagnosticMapping

6.8.1 DiagnosticEvent to DtcUds Mapping

[TPS_DEXT_03003] Semantics of `DiagnosticEventToTroubleCodeUdsMapping` ⌈ The `DiagnosticEventToTroubleCodeUdsMapping` is used to assign one (1:1) or multiple (n:1) `DiagnosticEvent`s to a `DiagnosticTroubleCodeUds`.

In case of n:1, multiple instances of `DiagnosticEventToTroubleCodeUdsMapping` with the same reference of role `troubleCodeUds` but different references of role `diagnosticEvent` have to be defined. ⌋*(RS_DEXT_00023, RS_DEXT_00024, RS_DEXT_00025)*

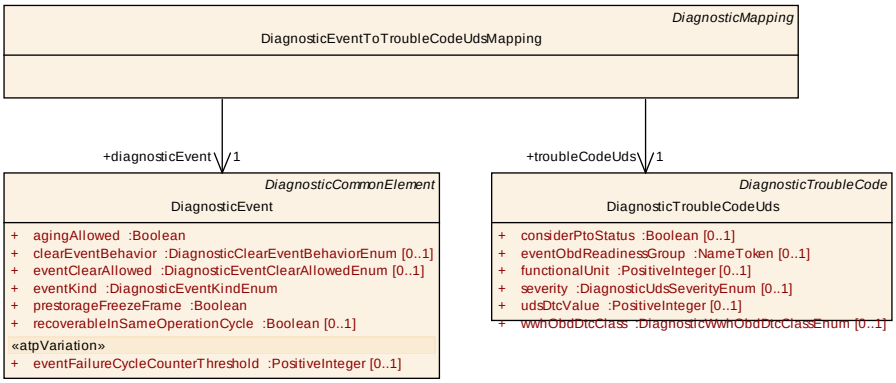

Figure 6.12: DiagnosticEventToDtcUdsMapping

Class	**DiagnosticEventToTroubleCodeUdsMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
Note	Defines which UDS Diagnostic Trouble Code is applicable for a DiagnosticEvent. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagnosticEvent	DiagnosticEvent	1	ref	Reference to a DiagnosticEvent to which a UDS Diagnostic Trouble Code is assigned.
troubleCodeUds	DiagnosticTroubleCodeUds	1	ref	Reference to an UDS Diagnostic Trouble Code assigned to a DiagnosticEvent.

Table 6.38: DiagnosticEventToTroubleCodeUdsMapping

6.8.2 DiagnosticEvent to DiagnosticOperationCycle Mapping

[TPS_DEXT_01086] Reference to `DiagnosticOperationCycle` ⌈ A `DiagnosticEvent` needs to be assigned to exactly one `DiagnosticOperationCycle`. ⌋ *(RS_DEXT_00024, RS_DEXT_00054)*

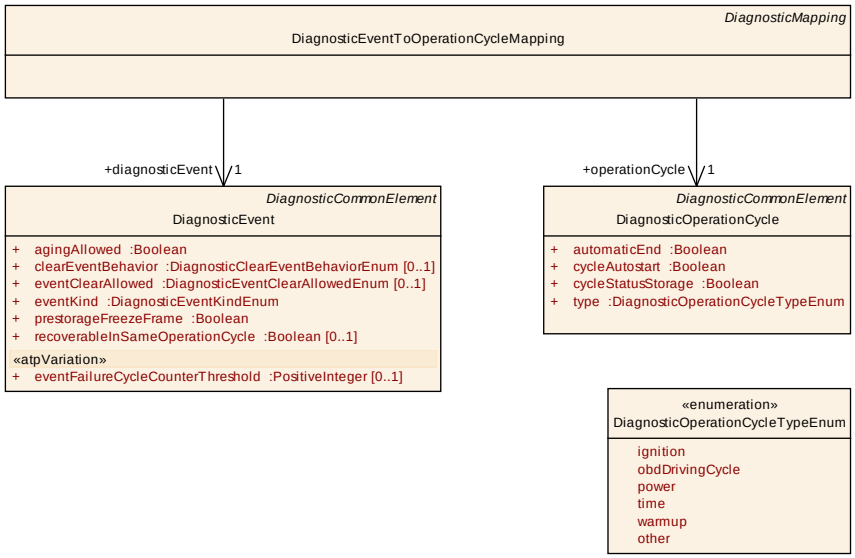

Figure 6.13: DiagnosticEventToOperationCycleMapping

Class	**DiagnosticEventToOperationCycleMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
Note	Defines which OperationCycle is applicable for a DiagnosticEvent. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagnosticEvent	DiagnosticEvent	1	ref	Reference to a DiagnosticEvent to which an OperationCycle is assigned.
operationCycle	DiagnosticOperationCycle	1	ref	Reference to an OperationCycle assigned to a DiagnosticEvent.

Table 6.39: DiagnosticEventToOperationCycleMapping

6.8.3 DiagnosticEvent to DebounceAlgorithm Mapping

[TPS_DEXT_03004] `DiagnosticEvent` and `DiagnosticDebounceAlgorithmProps` ⌈ If a `DiagnosticEvent` has to be debounced, it must be mapped to the appropriate `DiagnosticDebounceAlgorithmProps`. ⌋*(RS_DEXT_00023, RS_DEXT_00053)*

[TPS_DEXT_03005] Existence of `DiagnosticEventToDebounceAlgorithmMapping` ⌈ The `DiagnosticEventToDebounceAlgorithmMapping` shall not be created if the `DiagnosticEvent` is not debounced. ⌋*(RS_DEXT_00023, RS_DEXT_00053)*

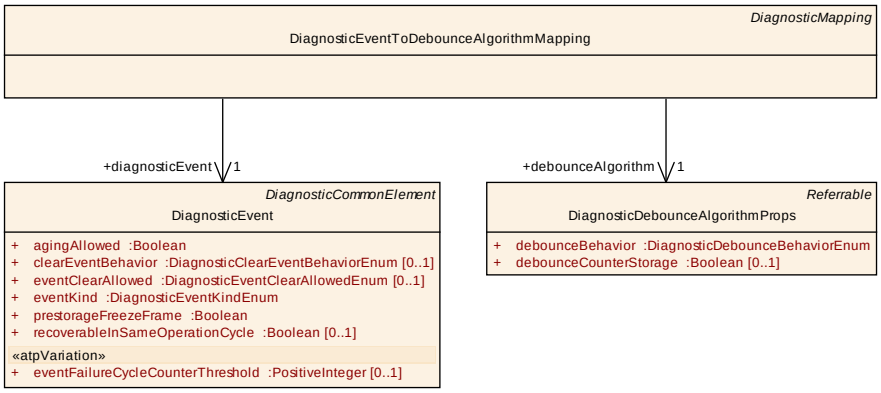

Figure 6.14: DiagnosticEventToDebounceAlgorithmMapping

Class	**DiagnosticEventToDebounceAlgorithmMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
Note	Defines which Debounce Algorithm is applicable for a DiagnosticEvent. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
debounceAlgorithm	DiagnosticDebounceAlgorithmProps	1	ref	Reference to a DebounceAlgorithm assigned to a DiagnosticEvent.
diagnosticEvent	DiagnosticEvent	1	ref	Reference to a DiagnosticEvent to which a DebounceAlgorithm is assigned.

Table 6.40: DiagnosticEventToDebounceAlgorithmMapping

Class	**DiagnosticDebounceAlgorithmProps**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticDebouncingAlgorithm			
Note	Defines properties for the debounce algorithm class.			
Base	ARObject, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
debounceAlgorithm	DiagEventDebounceAlgorithm	1	aggr	This represents the actual debounce algorithm.
debounceBehavior	DiagnosticDebounceBehaviorEnum	1	attr	This attribute defines how the event debounce algorithm will behave, if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled.
debounceCounterStorage	Boolean	0..1	attr	Switch to store the debounce counter value non-volatile or not. true: debounce counter value shall be stored non-volatile false: debounce counter value is volatile

Table 6.41: DiagnosticDebounceAlgorithmProps

The details regarding the formalization of debouncing behavior are depicted in Figure 6.15.

In particular, DiagnosticCommonProps aggregates DiagnosticDebounceAlgorithmProps in the role debounceAlgorithmProps. The DiagnosticDebounceAlgorithmProps itself does not actually represent the debouncing algorithm but provides attributes relevant for the actual debouncing algorithm.

[TPS_DEXT_01048] Actual algorithm for the diagnostic event debouncing ⌈ The actual algorithm for the debouncing is represented by subclasses of DiagEventDebounceAlgorithm aggregated in the role DiagnosticDebounceAlgorithmProps.debounceAlgorithm. ⌋*(RS_DEXT_00023, RS_DEXT_00053)*

In other words, the debouncing of diagnostic events can be formulated in two ways:

- The DiagEventDebounceCounterBased represents the ability to implement a counter-based debouncing.
- The DiagEventDebounceTimeBased represents the ability to implement a time-based debouncing.

[constr_1359] Existence of attribute DiagnosticDebounceAlgorithmProps.debounceCounterStorage ⌈ The attribute DiagnosticDebounceAlgorithmProps.debounceCounterStorage shall only exist if the aggregation DiagnosticDebounceAlgorithmProps.debounceAlgorithm actually aggregates a DiagEventDebounceCounterBased ⌋*()*

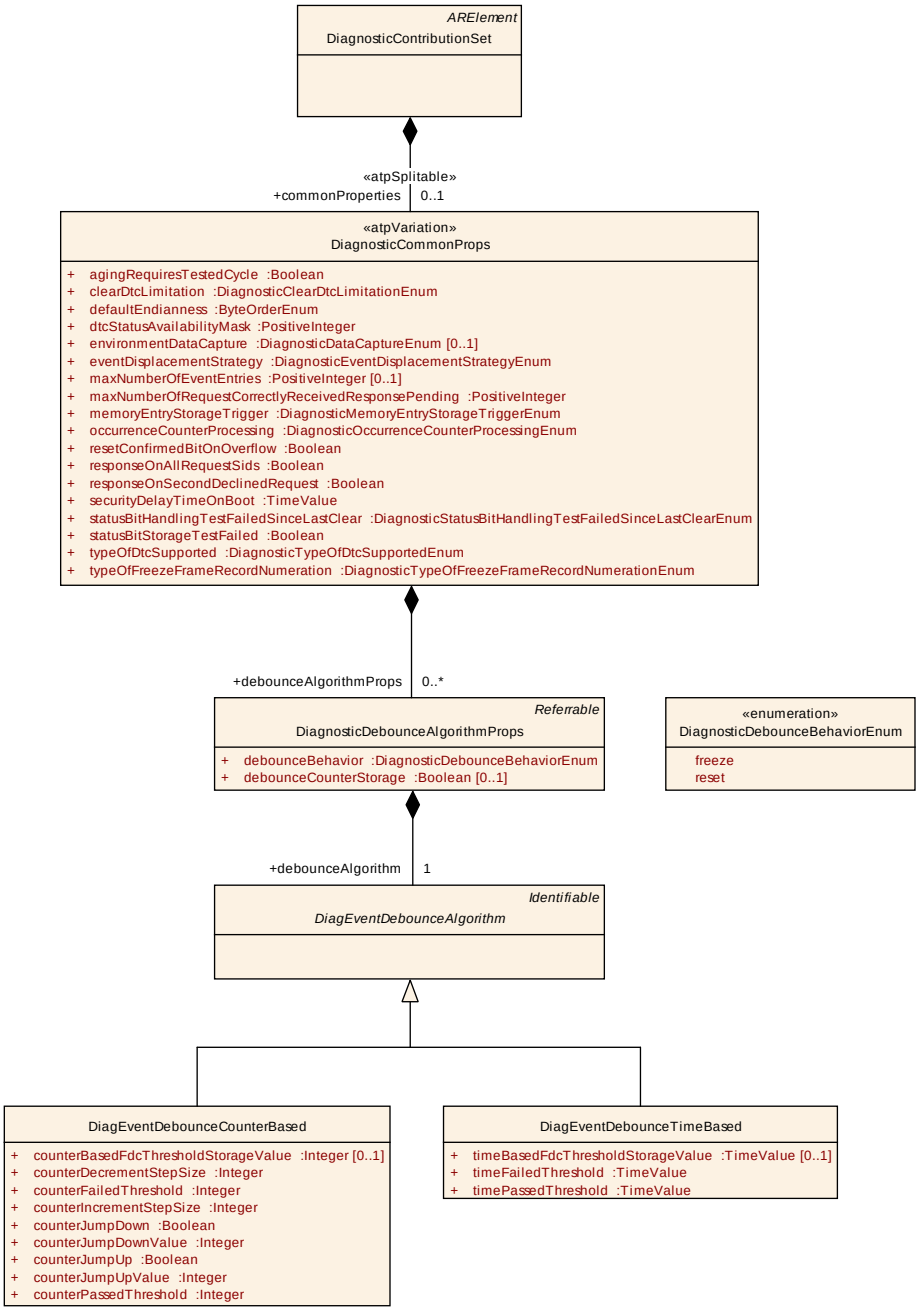

Figure 6.15: Details of `DiagnosticDebounceAlgorithmProps`

[constr_1360] Usage of `DiagEventDebounceMonitorInternal` is not supported in the context of `DiagnosticDebounceAlgorithmProps` ⌈ The usage of the meta-class `DiagEventDebounceMonitorInternal` for the aggregation in the role `DiagnosticDebounceAlgorithmProps.debounceAlgorithm` is not permitted. ⌋*()*

For clarification with respect to [constr_1360], `DiagEventDebounceMonitorInternal` is used in the context of formulating the `DiagnosticEventNeeds`, but its usage in the context of the `DiagnosticExtract` is not foreseen.

Class	**DiagnosticEventNeeds**			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the abstract needs on the configuration of the Diagnostic Event Manager for one diagnostic event. Its shortName can be regarded as a symbol identifying the diagnostic event from the viewpoint of the component or module which owns this element. In case the diagnostic event specifies a production error, the shortName shall be the name of the production error.			
Base	ARObject, DiagnosticCapabilityElement, Identifiable, MultilanguageReferrable, Referrable, ServiceNeeds			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
considerPtoStatus	Boolean	0..1	attr	PTO (Power Take Off) has an impact on the respective emission-related event (OBD). This information shall be provided by SW-C description in order to consider the PTO relevance e.g. for readiness (PID $01) computation. For events with dtcKind set to 'nonEmmissionRelatedDtc' this attribute is typically false.
deferringFid	FunctionInhibitionNeeds	*	ref	This reference contains the link to a function identifier within the FiM which is used by the monitor before delivering a result.
diagEventDebounceAlgorithm	DiagEventDebounceAlgorithm	0..1	aggr	Specifies the abstract need on the Debounce Algorithm applied by the Diagnostic Event Manager.
dtcKind	DtcKindEnum	0..1	attr	This attribute indicates the kind of the diagnostic monitor according to the SWS Diagnostic Event Manger. This attribute applies for the UDS diagnostics use case.
inhibitingFid	FunctionInhibitionNeeds	0..1	ref	This represents the primary Function Inhibition Identifier used for inhibition of the diagnostic monitor. The FID might either inhibit the monitoring of a symptom or the reporting of detected faults.
inhibitingSecondaryFid	FunctionInhibitionNeeds	*	ref	This represents the secondary Function Inhibition Identifier used for inhibition of the diagnostic monitor. Any of the FID inhibitions leads to an inhibition of the monitoring of a symptom or the reporting of detected faults.
obdDtcNumber	PositiveInteger	0..1	attr	This represents a reasonable Diagnostic Trouble Code. This allows to predefine the Diagnostic Trouble Code, e.g. if the a function developer has received a particular requirement from the OEM or from a standardization body. This attribute applies for the OBD diagnostics use case.
reportBehavior	ReportBehaviorEnum	0..1	attr	This switch indicates whether or not the BSW module is allowed to report the related Events before Dem_Init().

udsDtcNumber	PositiveInteger	0..1	attr	This represents a reasonable Diagnostic Trouble Code. This allows to predefine the Diagnostic Trouble Code, e.g. if the a function developer has received a particular requirement from the OEM or from a standardization body. This attribute applies for the UDS diagnostics use case.

Table 6.42: DiagnosticEventNeeds

Class	**DiagEventDebounceAlgorithm (abstract)**			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	This class represents the ability to specify the pre-debounce algorithm which is selected and/or required by the particular monitor. This class inherits from Identifiable in order to allow further documentation of the expected or implemented debouncing and to use the category for the identification of the expected / implemented debouncing.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 6.43: DiagEventDebounceAlgorithm

Class	**DiagEventDebounceCounterBased**			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	This meta-class represents the ability to indicate that the counter-based debounce algorithm shall be used by the DEM for this diagnostic monitor. This is related to set the ECUC choice container DemDebounceAlgorithmClass to DemDebounceCounterBased.			
Base	ARObject, DiagEventDebounceAlgorithm, Identifiable, MultilanguageReferrable, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
counterBasedFdcThresholdStorageValue	Integer	0..1	attr	Threshold to allocate an event memory entry and to capture the Freeze Frame.
counterDecrementStepSize	Integer	1	attr	This value shall be taken to decrement the internal debounce counter.
counterFailedThreshold	Integer	1	attr	This value defines the event-specific limit that indicates the "failed" counter status.
counterIncrementStepSize	Integer	1	attr	This value shall be taken to increment the internal debounce counter.
counterJumpDown	Boolean	1	attr	This value activates or deactivates the counter jump-down behavior.

counterJumpDownValue	Integer	1	attr	This value represents the initial value of the internal debounce counter if the counting direction changes from incrementing to decrementing.
counterJumpUp	Boolean	1	attr	This value activates or deactivates the counter jump-up behavior.
counterJumpUpValue	Integer	1	attr	This value represents the initial value of the internal debounce counter if the counting direction changes from decrementing to incrementing.
counterPassedThreshold	Integer	1	attr	This value defines the event-specific limit that indicates the "passed" counter status.

Table 6.44: DiagEventDebounceCounterBased

Class	**DiagEventDebounceTimeBased**			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	This meta-class represents the ability to indicate that the time-based pre-debounce algorithm shall be used by the Dem for this diagnostic monitor. This is related to set the EcuC choice container DemDebounceAlgorithmClass to DemDebounceTimeBase.			
Base	ARObject, DiagEventDebounceAlgorithm, Identifiable, MultilanguageReferrable, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
timeBasedFdcThresholdStorageValue	TimeValue	0..1	attr	Threshold to allocate an event memory entry and to capture the Freeze Frame.
timeFailedThreshold	TimeValue	1	attr	This value represents the event-specific delay indicating the "failed" status.
timePassedThreshold	TimeValue	1	attr	This value represents the event-specific delay indicating the "passed" status.

Table 6.45: DiagEventDebounceTimeBased

Enumeration	**DiagnosticDebounceBehaviorEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticDebouncing Algorithm
Note	Event debounce algorithm behavior options.
Literal	***Description***
freeze	The event debounce counter will be frozen with the current value and will not change while a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. After all related enable conditions are fulfilled and ControlDTCSetting of the related event is enabled again, the event qualification will continue with the next report of the event (i.e. SetEventStatus). **Tags:** atp.EnumerationValue=0

reset	The event debounce counter will be reset to initial value if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. The qualification of the event will be restarted with the next valid event report. **Tags:** atp.EnumerationValue=1

Table 6.46: DiagnosticDebounceBehaviorEnum

6.8.4 DiagnosticEvent to EnableConditionGroup Mapping

[TPS_DEXT_03015] `EnableConditions` have to be put into a `DiagnosticEnableConditionGroup` ⌈ `EnableConditions` that are assigned to a `DiagnosticEvent` have to be put into a `DiagnosticEnableConditionGroup` since only a group of `EnableConditions` can be mapped to a `DiagnosticEvent`. ⌋ *(RS_DEXT_00023, RS_DEXT_00026, RS_DEXT_00028)*

[constr_1361] Number of `DiagnosticEventToEnableConditionGroupMapping` elements per `DiagnosticEvent` ⌈ The mapping element `DiagnosticEventToEnableConditionGroupMapping` shall be created no more than once per `DiagnosticEvent`.

If several `DiagnosticEventToEnableConditionGroupMapping` elements referring the same `DiagnosticEvent` are defined, then the `Enable Condition Group` mapping shall be regarded as defective. ⌋*()*

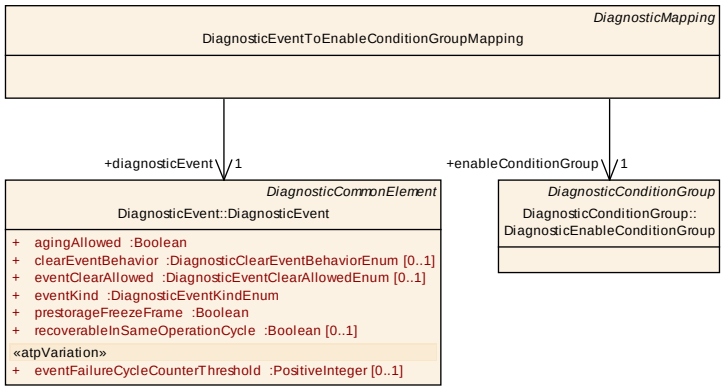

Figure 6.16: DiagnosticEventToEnableConditionGroupMapping

Class	**DiagnosticEventToEnableConditionGroupMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
Note	Defines which EnableConditionGroup is applicable for a DiagnosticEvent. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagnosticEvent	DiagnosticEvent	1	ref	Reference to a DiagnosticEvent to which an EnableConditionGroup is assigned.

enableConditionGroup	DiagnosticEnableConditionGroup	1	ref	Reference to an EnableConditionGroup assigned to a DiagnosticEvent.

Table 6.47: DiagnosticEventToEnableConditionGroupMapping

6.8.5 DiagnosticEvent to StorageConditionGroup Mapping

[TPS_DEXT_03016] StorageConditions have to be put into a DiagnosticStorageConditionGroup ⌈ StorageConditions that are assigned to a DiagnosticEvent have to be put into a DiagnosticStorageConditionGroup since only a group of StorageConditions can be mapped to a DiagnosticEvent. ⌋ *(RS_DEXT_00023, RS_DEXT_00027, RS_DEXT_00029)*

[constr_1362] Number of DiagnosticEventToStorageConditionGroupMapping elements per DiagnosticEvent ⌈ The mapping element DiagnosticEventToStorageConditionGroupMapping shall be created no more than once or once per DiagnosticEvent.

If several DiagnosticEventToStorageConditionGroupMapping elements referring the same DiagnosticEvent are defined, then the Storage Condition Group mapping shall be regarded as defective. ⌋*()*

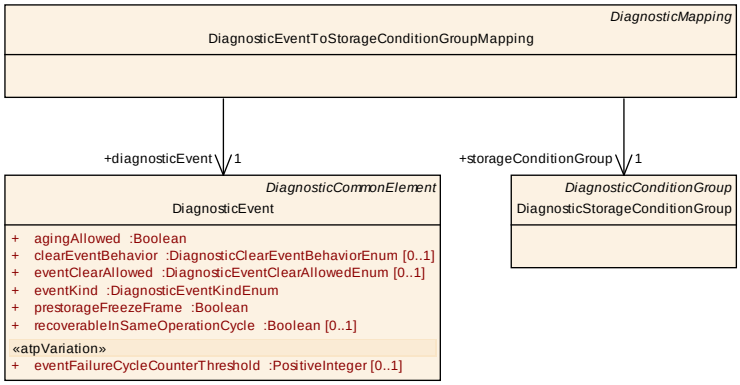

Figure 6.17: DiagnosticEventToStorageConditionGroupMapping

[TPS_DEXT_03006] Values of the individual DiagnosticStorageConditions ⌈ The values of the individual DiagnosticStorageConditions need to be algorithmically evaluated in order to find out whether or not the storage of the DiagnosticEvent is permitted.

The algorithm that is supposed to be implemented for this purpose is documented in [SWS_Dem_00459]. ⌋*(RS_DEXT_00027)*

Class	**DiagnosticEventToStorageConditionGroupMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
Note	Defines which StorageConditionGroup is applicable for a DiagnosticEvent. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagnosticEvent	DiagnosticEvent	1	ref	Reference to a DiagnosticEvent to which a StorageConditionGroup is assigned.
storageConditionGroup	DiagnosticStorageConditionGroup	1	ref	Reference to a StorageConditionGroup assigned to a DiagnosticEvent.

Table 6.48: DiagnosticEventToStorageConditionGroupMapping

6.8.6 DiagnosticEvent to Port Mapping

[TPS_DEXT_03007] Semantics of `DiagnosticEventPortMapping` ⌈ A `DiagnosticEventPortMapping` defines which `SwcServiceDependency`s of a `AtomicSwComponentType` or `BswServiceDependency` of a `BswModuleDescription` have to be connected to which `DiagnosticEvent`.

This is realized by defining a `DiagnosticEventPortMapping` referencing a `DiagnosticEvent` and (using ≪`instanceRef`≫ an instance of `SwcServiceDependency` (or `BswServiceDependency`). ⌋*(RS_DEXT_00023, RS_DEXT_00052)*

If such an instance is not yet available, an ordinary reference to `SwcServiceDependency` can be given alternatively (i.e. without specifying a certain instance).

In this way, the ECU integrator is able to directly derive the actual mapping between SWC (or BSW) service ports and the ports of the `Service Components` during ECU configuration.

[constr_1435] Debouncing in the presence of a `DiagnosticEventPortMapping` ⌈ If a `DiagnosticEventPortMapping` exists and the enclosed `DiagnosticEventPortMapping.diagnosticEvent` is also referenced by a `DiagnosticEventToDebounceAlgorithmMapping` then the concrete subclass of the respective `DiagnosticEventToDebounceAlgorithmMapping.debounceAlgorithm.debounceAlgorithm` shall be identical to the `DiagnosticEventPortMapping.swcServiceDependencyInSystem/swcFlatServiceDependency.serviceNeeds.diagEventDebounceAlgorithm`.

It is assumed that the `DiagnosticEventPortMapping.swcServiceDependencyInSystem/swcFlatServiceDependency.serviceNeeds` is a `DiagnosticEventNeeds`. ⌋*()*

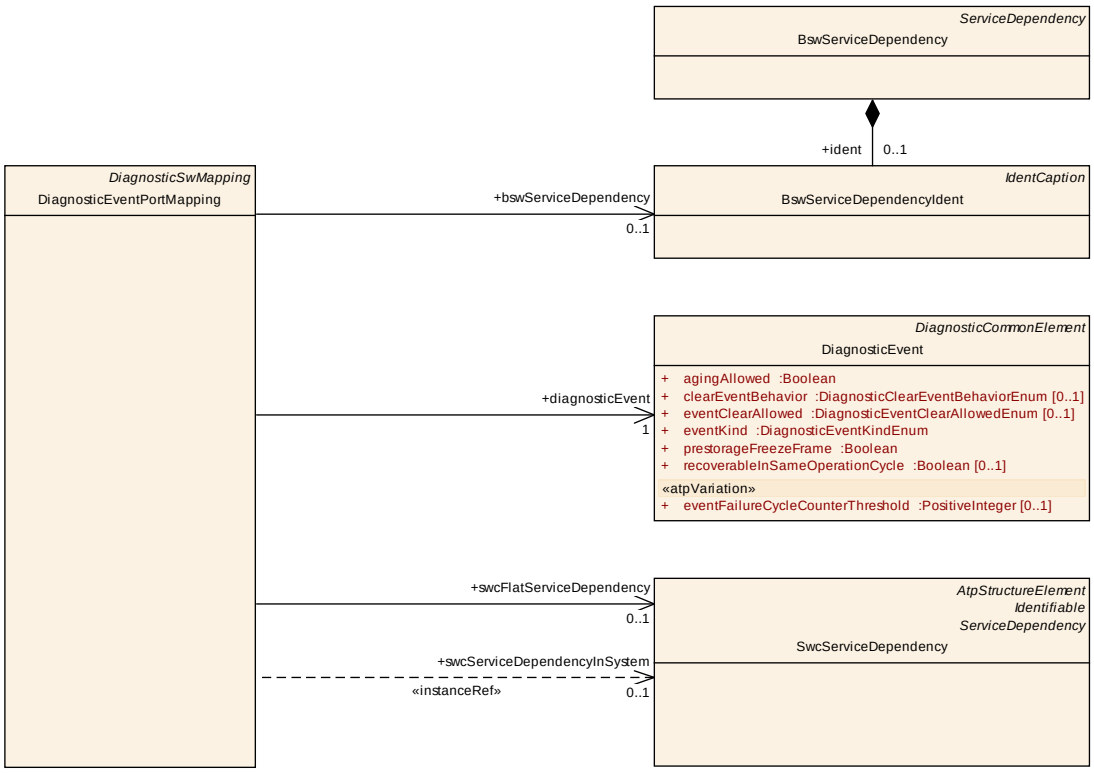

Figure 6.18: DiagnosticEventPortMapping

Class	**DiagnosticEventPortMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
Note	Defines to which SWC service ports with DiagnosticEventNeeds the DiagnosticEvent is mapped. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, DiagnosticSwMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
bswServiceDependency	BswServiceDependencyIdent	0..1	ref	Reference to a BswServiceDependency that links ServiceNeeds to BswModuleEntries.
diagnosticEvent	DiagnosticEvent	1	ref	Reference to the DiagnosticEvent that is assigned to SWC service ports with DiagnosticEventNeeds.
swcFlatServiceDependency	SwcServiceDependency	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports.
swcServiceDependencyInSystem	SwcServiceDependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports.

Table 6.49: DiagnosticEventPortMapping

6.8.7 DiagnosticOperationCycle to Port Mapping

[TPS_DEXT_03017] Semantics of `DiagnosticOperationCyclePortMapping` ⌈
A `DiagnosticOperationCyclePortMapping` defines which SWC service port(s) have to be connected to which `DiagnosticOperationCycle`.

This is realized by defining a `DiagnosticOperationCyclePortMapping` referencing a `DiagnosticOperationCycle` and an instance of `SwcServiceDependency`. ⌋*(RS_DEXT_00052, RS_DEXT_00053)*

If such an instance is not yet available, an ordinary reference to `SwcServiceDependency` can be given alternatively (i.e. without specifying a certain instance).

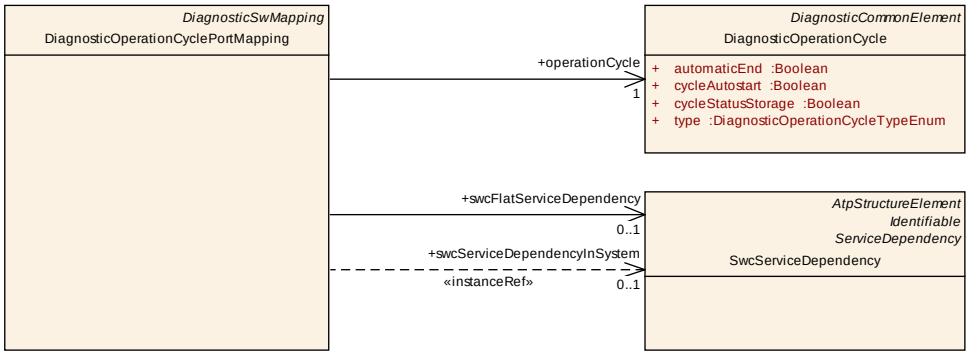

Figure 6.19: DiagnosticOperationCyclePortMapping

Class	**DiagnosticOperationCyclePortMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
Note	Defines to which SWC service ports with DiagnosticOperationCycleNeeds the DiagnosticOperationCycle is mapped. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, DiagnosticSwMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
operationCycle	DiagnosticOperationCycle	1	ref	Reference to the DiagnosticOperationCycle that is assigned to SWC service ports with DiagnosticOperationCycleNeeds.
swcFlatServiceDependency	SwcServiceDependency	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports.
swcServiceDependencyInSystem	SwcServiceDependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports.

Table 6.50: DiagnosticOperationCyclePortMapping

6.8.8 DiagnosticEnableCondition to Port Mapping

[TPS_DEXT_03018] Semantics of `DiagnosticEnableConditionPortMapping`
⌈ A `DiagnosticEnableConditionPortMapping` defines which SWC service port(s) have to be connected to which `DiagnosticEnableCondition`. This is realized by defining a `DiagnosticEnableConditionPortMapping` referencing a `DiagnosticEnableCondition` and an instance of `SwcServiceDependency`. ⌋
(RS_DEXT_00026, RS_DEXT_00052)

If such an instance is not yet available, an ordinary reference to `SwcServiceDependency` can be given alternatively (i.e. without specifying a certain instance).

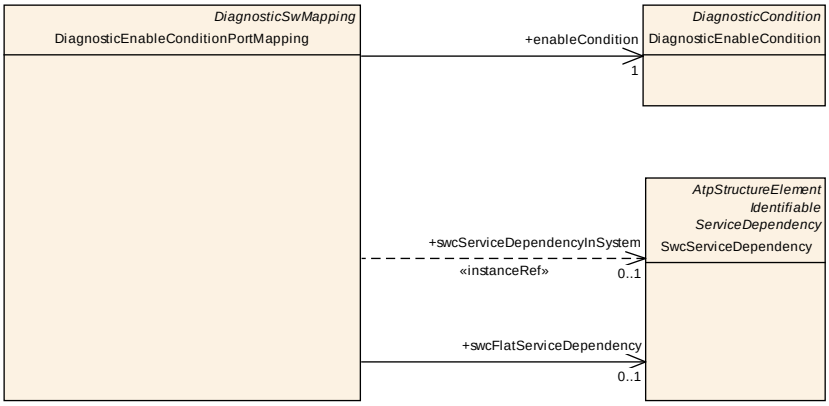

Figure 6.20: DiagnosticEnableConditionPortMapping

Class	**DiagnosticEnableConditionPortMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
Note	Defines to which SWC service ports with DiagnosticEnableConditionNeeds the DiagnosticEnableCondition is mapped. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, DiagnosticSwMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
enableCondition	DiagnosticEnableCondition	1	ref	Reference to the EnableCondition which is mapped to a SWC service port with DiagnosticEnableConditionNeeds.
swcFlatServiceDependency	SwcServiceDependency	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports. This reference can be used in early stages of the development in order to identify the SwcServiceDependency without a full System Context.
swcServiceDependencyInSystem	SwcServiceDependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports.

Table 6.51: DiagnosticEnableConditionPortMapping

6.8.9 DiagnosticStorageCondition to Port Mapping

[TPS_DEXT_03019] Semantics of `DiagnosticStorageConditionPortMapping` ⌈ A `DiagnosticStorageConditionPortMapping` defines which SWC service port(s) have to be connected to which `DiagnosticStorageCondition`. This is realized by defining a `DiagnosticStorageConditionPortMapping` referencing a `DiagnosticStorageCondition` and an instance of `SwcServiceDependency`. ⌋ *(RS_DEXT_00027, RS_DEXT_00052)*

If such an instance is not yet available, an ordinary reference to `SwcServiceDependency` can be given alternatively (i.e. without specifying a certain instance).

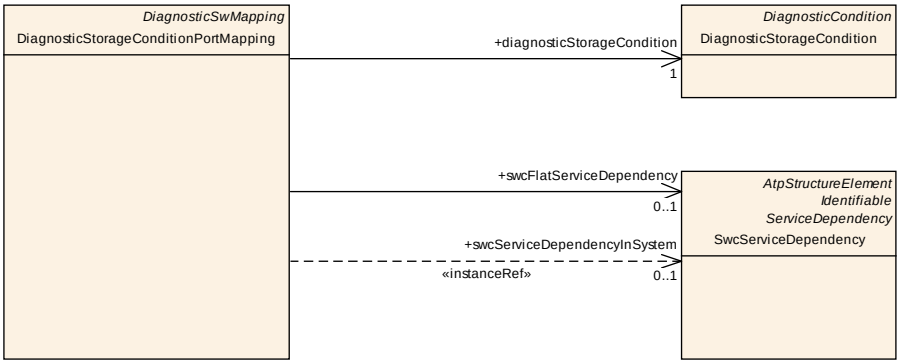

Figure 6.21: DiagnosticStorageConditionPortMapping

Class	**DiagnosticStorageConditionPortMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticMapping			
Note	Defines to which SWC service ports with DiagnosticStorageConditionNeeds the DiagnosticStorageCondition is mapped. **Tags:** atp.recommendedPackage=DiagnosticMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, DiagnosticSwMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagnosticStorageCondition	DiagnosticStorageCondition	1	ref	Reference to the StorageCondition which is mapped to a SWC service port with DiagnosticStorageConditionNeeds.
swcFlatServiceDependency	SwcServiceDependency	0..1	ref	Reference to a SwcServiceDependencyType that links ServiceNeeds to SWC service ports.
swcServiceDependencyInSystem	SwcServiceDependency	0..1	iref	Instance reference to a SwcServiceDependency that links ServiceNeeds to SWC service ports.

Table 6.52: DiagnosticStorageConditionPortMapping

6.8.10 Provided Data Mapping

[TPS_DEXT_03020] Semantics of `DiagnosticDemProvidedDataMapping` ⌈ The meta-class `DiagnosticDemProvidedDataMapping` does not seem to fulfill the condition for representing a mapping class because it only has one reference to a `DiagnosticDataElement` in the role `dataElement`.

However, the specific nature of this mapping is that the second element (the `DiagnosticDemProvidedDataMapping.dataProvider`) that is supposed to take place in the mapping cannot precisely be modeled as a single meta-class.

Therefore, there is no better way than to model the `DiagnosticDemProvidedDataMapping.dataProvider` by a `NameToken`. Of course, the collection of possible values of this attribute need to be agreed upon up-front, potentially on a project-specific basis.

The semantics of this mapping is to further qualify the access to the `DiagnosticDataElement` referenced in the role `dataElement` from within the Dem. ⌋ *(RS_DEXT_00043, RS_DEXT_00052)*

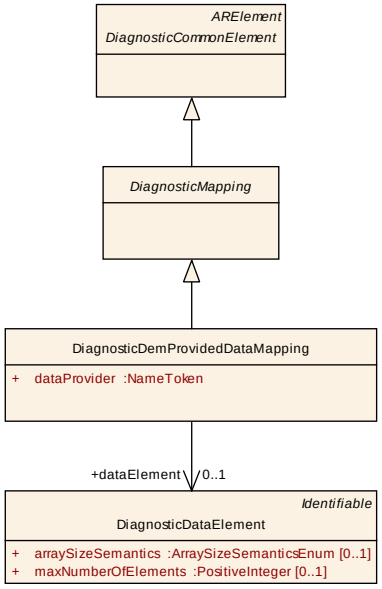

Figure 6.22: Modeling of the `DiagnosticDemProvidedDataMapping`

Class	**DiagnosticDemProvidedDataMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::ServiceMapping			
Note	This represents the ability to define the nature of a data access for a DiagnsoticDataElement in the Dem. **Tags:** atp.recommendedPackage=DiagnosticServiceMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

dataElement	DiagnosticDataElement	0..1	ref	This represents the DiagnosticDataElement for which the access is further qualified by the DiagnosticDemProvidedDataMapping.
dataProvider	NameToken	1	attr	This represents the ability to further specify the access within the Dem.

Table 6.53: DiagnosticDemProvidedDataMapping

6.9 DiagnosticOperationCycle

[TPS_DEXT_01087] Semantics of `DiagnosticOperationCycle` ⌈ Different types of `DiagnosticOperationCycle`s are supported and defined by the `type` attribute, e.g. the time between ignition on and ignition off. ⌋*(RS_DEXT_00054)*

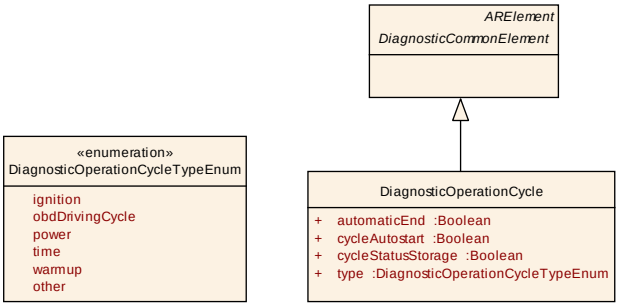

Figure 6.23: Modeling of DiagnosticOperationCycle

Class	**DiagnosticOperationCycle**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticOperationCycle			
Note	Definition of an operation cycle that is the base of the event qualifying and for Dem scheduling. **Tags:** atp.recommendedPackage=DiagnosticOperationCycles			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
automaticEnd	Boolean	1	attr	If set to true the driving cycle shall automatically end at either Dem_Shutdown() or Dem_Init().
cycleAutostart	Boolean	1	attr	This attribute defines if the operation cycles is automatically re-started during Dem_PreInit.
cycleStatusStorage	Boolean	1	attr	Defines if the operation cycle state is available over the power cycle (stored non-volatile) or not. true: the operation cycle state is stored non-volatile false: the operation cycle state is only stored volatile
type	DiagnosticOperationCycleTypeEnum	1	attr	Operation cycles types for the Dem.

Table 6.54: DiagnosticOperationCycle

Enumeration	**DiagnosticOperationCycleTypeEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticOperationCycle
Note	Operation cycles types used to identify certain Operation cycles with a certain semantics.
Literal	***Description***
ignition	Ignition ON / OFF cycle **Tags:** atp.EnumerationValue=0
obdDriving Cycle	OBD Driving cycle **Tags:** atp.EnumerationValue=1
other	further operation cycle **Tags:** atp.EnumerationValue=2
power	Power ON / OFF cycle **Tags:** atp.EnumerationValue=3; atp.Status=obsolete
time	Time based operation cycle **Tags:** atp.EnumerationValue=4; atp.Status=obsolete
warmup	OBD Warm up cycle **Tags:** atp.EnumerationValue=5

Table 6.55: DiagnosticOperationCycleTypeEnum

6.10 DiagnosticAging

[TPS_DEXT_03021] Aging ⌈ It is possible to remove a specific event from the event memory, if its fault conditions are not fulfilled for a certain period of time. This process is called as aging or unlearning.

This semantics is formalized by means of the meta-class `DiagnosticAging`. ⌋ *(RS_DEXT_00055)*

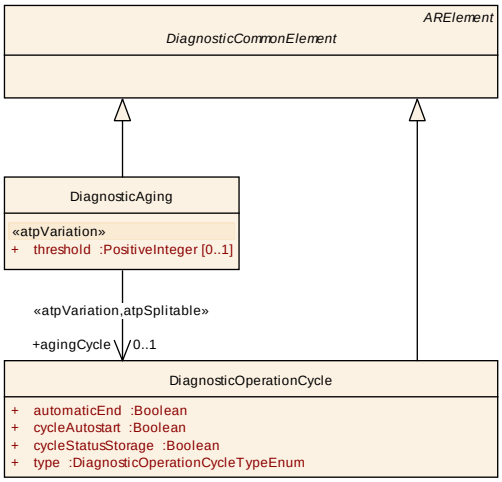

Figure 6.24: Modeling of `DiagnosticAging`

Class	**DiagnosticAging**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticAging			
Note	Defines the aging algorithm. **Tags:** atp.recommendedPackage=DiagnosticAgings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
agingCycle	DiagnosticOperationCycle	0..1	ref	This represents the applicable aging cycle. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=agingCycle, variationPoint.ShortLabel vh.latestBindingTime=preCompileTime
threshold	PositiveInteger	0..1	attr	Number of aging cycles needed to unlearn/delete the event. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table 6.56: DiagnosticAging

6.11 DiagnosticIndicator

[TPS_DEXT_03022] Different kinds of `DiagnosticIndicator`s ⌈ Different `type`s of Indicators can be defined with the `DiagnosticIndicator` element. For this, the attribute `DiagnosticIndicator.type` shall be used. ⌋*(RS_DEXT_00056)*

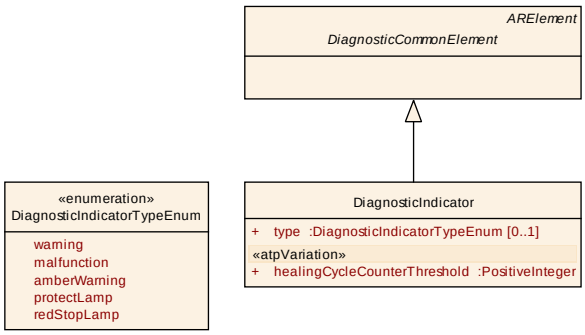

Figure 6.25: Modeling of `DiagnosticIndicator`

Class	**DiagnosticIndicator**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticIndicator			
Note	Definition of an indicator. **Tags:** atp.recommendedPackage=DiagnosticIndicators			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
healingCycleCounterThreshold	PositiveInteger	1	attr	This attribute defines the number of healing cycles for the WarningIndicatorOffCriteria **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
type	DiagnosticIndicatorTypeEnum	0..1	attr	Defines the type of the indicator.

Table 6.57: DiagnosticIndicator

Enumeration	**DiagnosticIndicatorTypeEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticIndicator
Note	Type of an indicator.
Literal	***Description***
amberWarning	Amber Warning Lamp **Tags:** atp.EnumerationValue=0
malfunction	Malfunction Indicator Lamp **Tags:** atp.EnumerationValue=1
protectLamp	Protect Lamp **Tags:** atp.EnumerationValue=2
redStopLamp	Red Stop Lamp **Tags:** atp.EnumerationValue=3
warning	Warning **Tags:** atp.EnumerationValue=4

Table 6.58: DiagnosticIndicatorTypeEnum

6.12 DiagnosticTestResult

The meta-class `DiagnosticTestResult` allows for a formal definition of a diagnostic test result. The purpose of this meta-class is to support the reporting of latest test results back to a client. This is of special importance for the OBD service mode 0x06, see chapter 5.6.5.

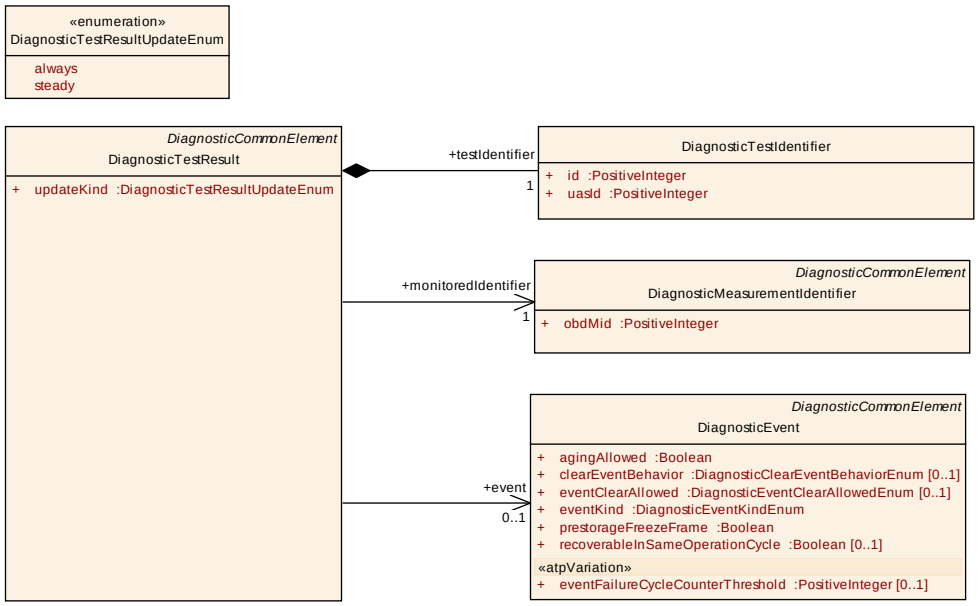

Figure 6.26: Modeling of `DiagnosticTestResult`

Class	**DiagnosticTestResult**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTestResult			
Note	This meta-class represents the ability to define diagnostic test results. **Tags:** atp.recommendedPackage=DiagnosticTestResults			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
event	DiagnosticEvent	0..1	ref	This attribute represents the diagnostic event that is related to the diagnostic test result.
monitoredIdentifier	DiagnosticMeasurementIdentifier	1	ref	This attribute represents the related diagnostic monitored identifier.
testIdentifier	DiagnosticTestIdentifier	1	aggr	This attribute represents the applicable test identifier.
updateKind	DiagnosticTestResultUpdateEnum	1	attr	This attribute controls the update behavior of the enclosing DiagnosticTestResult.

Table 6.59: DiagnosticTestResult

Enumeration	**DiagnosticTestResultUpdateEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTestResult

Note	This meta-class represents the ability to define the update behavior of a DiagnosticTestResult.
Literal	***Description***
always	Any DTR result reported by the monitor is used by the Dem. **Tags:** atp.EnumerationValue=0
steady	The Dem accepts reported DTRs only when the configured debouncing mechanism is stable at the FAIL or PASS limit. **Tags:** atp.EnumerationValue=1

Table 6.60: DiagnosticTestResultUpdateEnum

Class	**DiagnosticTestIdentifier**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTestResult			
Note	This meta-class represents the ability to create a diagnostic test identifier.			
Base	ARObject			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
id	PositiveInteger	1	attr	This represents the numerical id associated with the diagnostic test identifier.
uasId	PositiveInteger	1	attr	This represents the unit and scaling Id of the diagnostic test result.

Table 6.61: DiagnosticTestIdentifier

Class	**DiagnosticMeasurementIdentifier**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTestResult			
Note	This meta-class represents the ability to describe a measurement identifier. **Tags:** atp.recommendedPackage=DiagnosticMeasurementIdentifiers			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
obdMid	PositiveInteger	1	attr	This represents the numerical measurement Id

Table 6.62: DiagnosticMeasurementIdentifier

6.13 OBD-related aspects of Dem Configuration

The support for OBD-related [18] modeling requires the addition of some pretty global (i.e. on the level of an entire ECU) attribute that could be added to the `EcuInstance`.

However, this would not align with the idea of a decentralized configuration of the diagnostic stack where information is added at a point in time where an actual `EcuInstance` is not yet available.

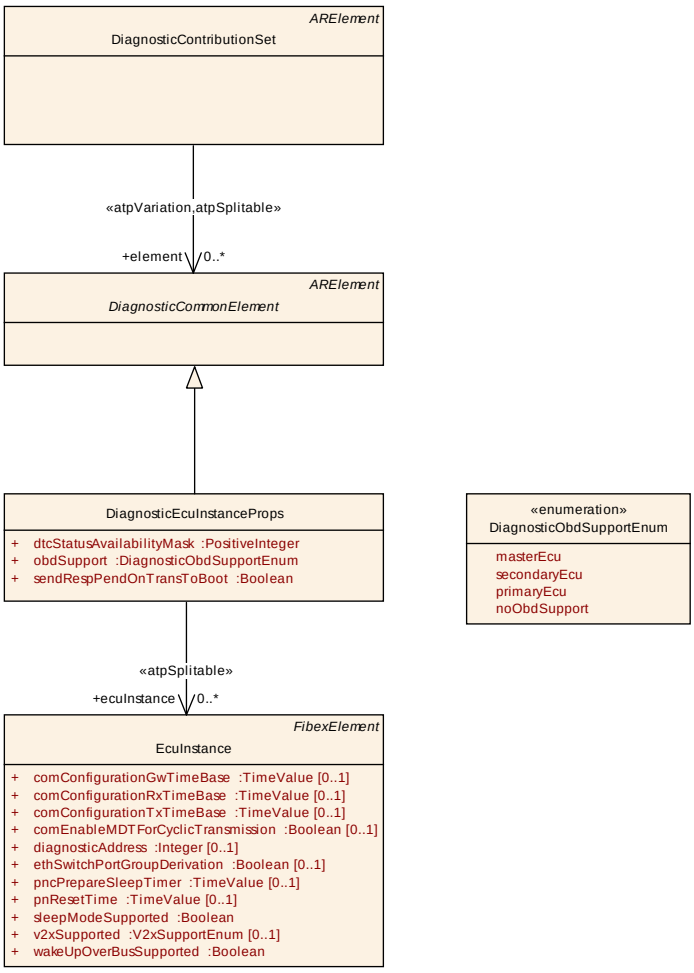

Figure 6.27: Modeling of `DiagnosticEcuInstanceProps`

Therefore, the attributes applying for the ECU-level are added to the meta-class `DiagnosticEcuInstanceProps`.

[TPS_DEXT_01122] Indication whether a `EcuInstance` supports OBD ⌈ The attribute `DiagnosticEcuInstanceProps.obdSupport` is taken to define whether or not a given `EcuInstance` shall support OBD and in which way OBD shall be supported on this Ecu. ⌋*(RS_DEXT_00058)*

Class	**DiagnosticEcuInstanceProps**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticContribution			
Note	This meta-class represents the ability to model properties that are specific for a given EcuInstance but on the other hand represent purely diagnostic-related information. In the spirit of decentralized configuration it is therefore possible to specify the diagnostic-related information related to a given EcuInstance even if the EcuInstance does not yet exist. **Tags:** atp.recommendedPackage=DiagnosticEcuInstancePropss			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

dtcStatusAvailabilityMask	PositiveInteger	1	attr	This attribute contains the value of the DTC status availability mask.
ecuInstance	EcuInstance	*	ref	This represents the actual EcuInstance to which the information conatined in the DiagnosticEcuInstance contribute. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=ecuInstance
obdSupport	DiagnosticObdSupportEnum	1	attr	This attribute is used to specify the role (if applicable) in which the DiagnosticEcuInstance supports OBD.
sendRespPendOnTransToBoot	Boolean	1	attr	The purpose of this attribute is to define whether or not the ECU should send a NRC 0x78 (response pending) before transitioning to the bootloader (in this case the attribute shall be set to "true") or if the transition shall be initiated without sending NRC 0x78 (in this case the attribute shall be set to "false"). **Tags:** atp.Status=obsolete

Table 6.63: DiagnosticEcuInstanceProps

Enumeration	**DiagnosticObdSupportEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::DiagnosticContribution
Note	This meta-class represents the ability to model the roles in which a participation in OBD is foreseen. At the moment, this applies exclusively to the Dem. However, future extension of the Dcm may require this setting as well.
Literal	***Description***
masterEcu	This represent the role "master ECU". **Tags:** atp.EnumerationValue=0
noObdSupport	This represents the ability to explicitly specify that no participation in OBD is foreseen. **Tags:** atp.EnumerationValue=1
primaryEcu	This represents the role "primary ECU". **Tags:** atp.EnumerationValue=2
secondaryEcu	This represents the role "secondary ECU". **Tags:** atp.EnumerationValue=3

Table 6.64: DiagnosticObdSupportEnum

Class	**DiagnosticTroubleCodeObd**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This element is used to define OBD-relevant DTCs. **Tags:** atp.recommendedPackage=DiagnosticTroubleCodes			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticTroubleCode, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
considerPtoStatus	Boolean	1	attr	This attribute describes the affection of the event by the Dem PTO handling. True: the event is affected by the Dem PTO handling. False: the event is not affected by the Dem PTO handling. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
dtcProps	DiagnosticTroubleCodeProps	0..1	ref	Defined properties associated with the DemDTC.
eventObdReadinessGroup	NameToken	0..1	attr	This attribute specifies the Event OBD Readiness group for PID $01 and PID $41 computation. This attribute is only applicable for emission-related ECUs.
obdDTCValue	PositiveInteger	0..1	attr	Unique Diagnostic Trouble Code value for OBD. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime

Table 6.65: DiagnosticTroubleCodeObd

Another OBD-related feature of the Dem is the support for the computation of the *In-Use-Monitor Performance Ratio* (IUMPR). It is possible to define a so-called `DiagnosticIumprGroup` that encapsulates the computation of a group of `DiagnosticEvent`s.

The modeling of the `DiagnosticIumprDenominatorGroup` supports the configuration of additional environmental conditions that shall be applied on the incrementation of the IUMPR denominator.

From the modeling point of view, neither `DiagnosticIumprGroup` nor `DiagnosticIumprDenominatorGroup` reference the associated `DiagnosticEvent` directly.

But since the `DiagnosticIumpr` referenced in the role `iumpr` in turn references at most one `DiagnosticEvent` the resulting relation effectively boils down to `DiagnosticIumprGroup` and `DiagnosticIumprDenominatorGroup` being able to precisely define the collection of affected `DiagnosticEvent`s.

[TPS_DEXT_01110] Standardized values of `DiagnosticIumprGroup.category`
⌈ The semantics of the `DiagnosticIumprGroup` are identified by means of the attribute `DiagnosticIumprGroup.category`. Standardized values of `DiagnosticIumprGroup.category` are:

- IUMPR_DENOMINATOR_NONE
- IUMPR_DENOMINATOR_PHYSICAL_API
- IUMPR_DENOMINATOR_COLDSTART
- IUMPR_DENOMINATOR_EVAP
- IUMPR_DENOMINATOR_500_MILES

⌋(*RS_DEXT_00078*)

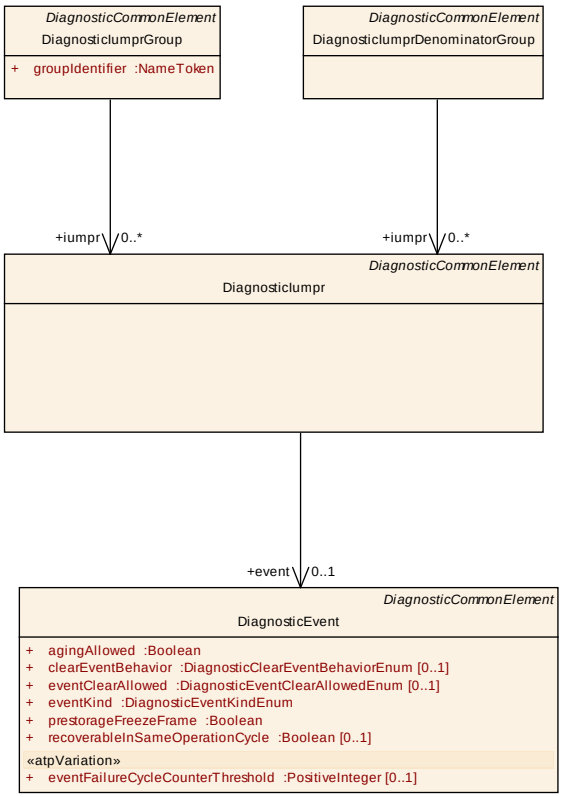

Figure 6.28: Modeling of IUMPR handling for diagnostic events

Class	**Diagnosticlumpr**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
Note	This meta-class represents he ability to model the in-us monitor performance ratio. The latter computes to the number of times a fault could have been found divided by the number of times the vehicle conditions have been properly fulfilled. **Tags:** atp.recommendedPackage=Diagnosticlumprs			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
event	DiagnosticEvent	0..1	ref	This reference represents the DiagnosticEvent that corresoponds to the IUMPR computation.

Table 6.66: Diagnosticlumpr

Class	**DiagnosticIumprGroup**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
Note	This meta-class represents the ability to model a IUMPR groups. **Tags:** atp.recommendedPackage=DiagnosticIumprGroups			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
groupIdentifier	NameToken	1	attr	This attribute shall be taken to define an identifier for the IUMPR group. Please note that the value of this identifier is driven by regulations outside the scope of AUTOSAR and can therefore not be limited to the set of characters suitable for a shortName.
iumpr	DiagnosticIumpr	*	ref	This reference collects DiagnosticIumpr to a DiagnosticIumprGroup.

Table 6.67: DiagnosticIumprGroup

Class	**DiagnosticIumprDenominatorGroup**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
Note	This meta-class represents the ability to model a IUMPR denominator groups. **Tags:** atp.recommendedPackage=DiagnosticIumprDenominatorGroup			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
iumpr	DiagnosticIumpr	*	ref	This reference collects DiagnosticIumpr to a DiagnosticIumprDenominatorGroup.

Table 6.68: DiagnosticIumprDenominatorGroup

6.13.1 Dem Configuration for OBD-II

The modeling of DTCs for the OBD-II use case is pretty similar to the modeling of DTCs for the UDS use case. In other words, `DiagnosticTroubleCodeObd` references the `DiagnosticTroubleCodeProps` in the same way that this meta-class is referenced from `DiagnosticTroubleCodeUds`.

Please note that the meta-class `DiagnosticTroubleCodeObd` is only applicable for the implementation of OBD-II.

[TPS_DEXT_01111] Legislative freeze frame for the OBD-II case ⌈ For the implementation of OBD-II, the legislative freeze frame is obtained from running the OBD Mode 0x02 service, modeled by means of meta-class `DiagnosticPowertrainFreezeFrame`. ⌋()

[constr_1459] Existence of attributes of `DiagnosticTroubleCodeProps` ⌈ The following list of attributes of meta-class `DiagnosticTroubleCodeProps` are not required and therefore shall be ignored if the `DiagnosticTroubleCodeProps` is referenced in the role `dtcProps` from a `DiagnosticTroubleCodeObd`:

- `freezeFrame`
- `freezeFrameContent`
- `memoryDestination`
- `extendedDataRecord`
- `aging`

⌋()

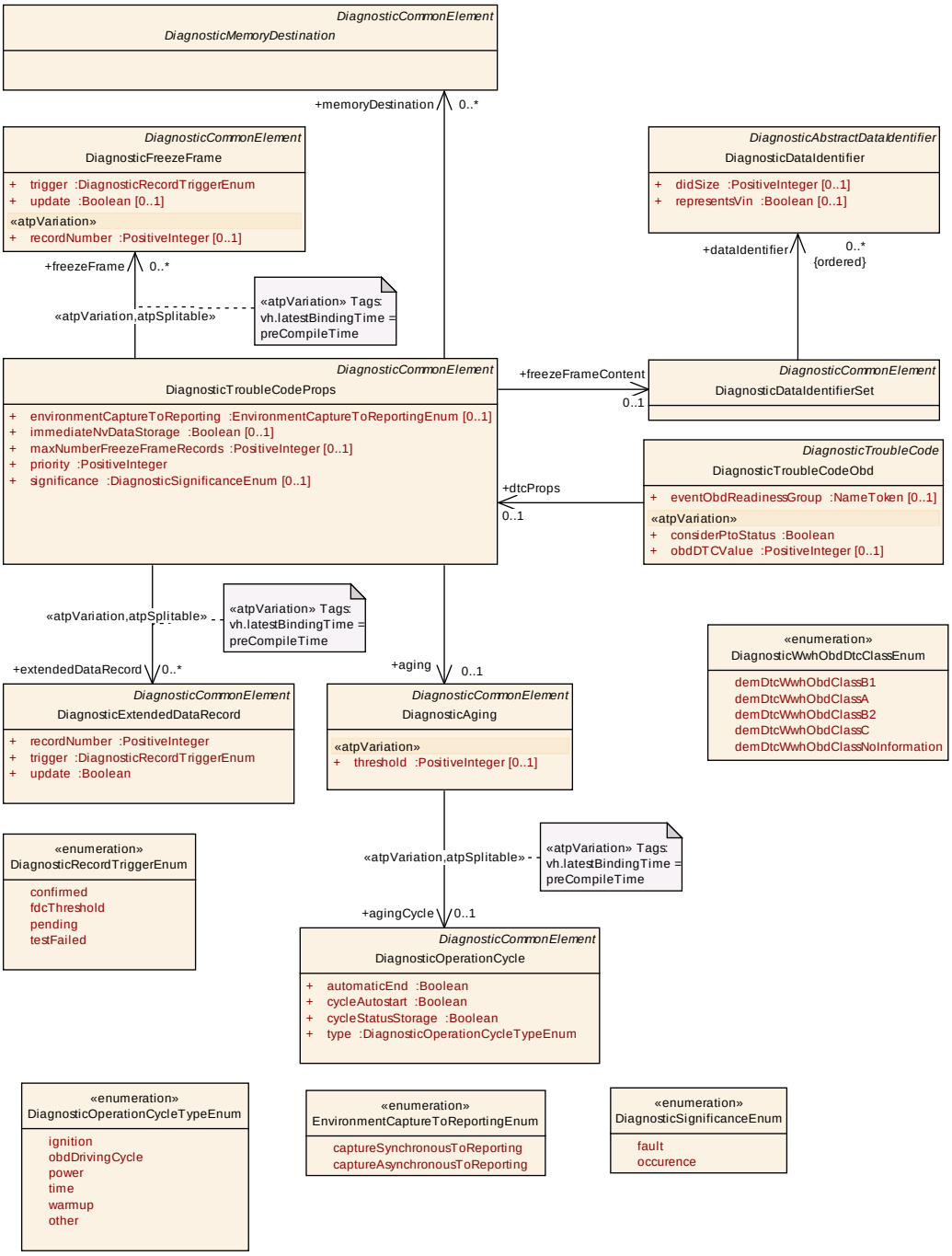

Figure 6.29: Modeling of DTCs for OBD-II

6.13.2 Dem Configuration for WWH-OBD

[TPS_DEXT_01112] Definition of a diagnostic trouble code for the implementation of WWH-OBD ⌈ The definition of a diagnostic trouble code for the implementation of WWH-OBD implies the existence of a `DiagnosticTroubleCodeUds` where attribute `wwhObdDtcClass` is set to any value other than `demDtcWwhObdClassNoInformation`. ⌋()

[TPS_DEXT_01093] Definition of legislative freeze frame for WWWH-OBD ⌈ The legislative WWH-OBD freeze frame is identified by the reference DiagnosticTroubleCodeProps.freezeFrame where attribute recordNumber is set to the value 0.

Optionally, the existence of a reference in the role DiagnosticTroubleCodeProps.extendedDataRecord with attribute recordNumber set to the value 90 is supported. ⌋()

7 Functional Inhibition

7.1 Introduction

Conceptually, the Fim [14] is closely related to the Dem since it handles the relation of functionality (expressed via the so-called *function identifier*, or in short-form: *Fid*) and linked DiagnosticEvents.

7.2 Alias Events

The close relation of Fim and Dem may have consequences in a distributed configuration: it may not be possible to configure the Fim before the Dem is configured because model elements (especially DiagnosticEvent) from the Dem configuration are required to model the Fim configuration.

This leads to the definition of DiagnosticFimAliasEvents that can be taken to model the Fim configuration even of no Dem configuration exists or if configurations with different Dem event names shall be considered for the final projects.

Please note, however, that the definition of DiagnosticFimAliasEvents is not mandatory for the configuration of the Fim. It is possible to directly take the existence of DiagnosticEvents into account and thereby bypass the definition of DiagnosticFimAliasEvents.

Class	**DiagnosticFimAliasEvent**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
Note	This meta-class is used to represent a given event semantics. However, the name of the actual events used in a specific project is sometimes not defined yet, not known or not in the responsibility of the author. Therefore, the DiagnosticFimAliasEvent has a reference to the actual DiagnosticEvent and by this the final connection is created. **Tags:** atp.recommendedPackage=DiagnosticFimAliasEvents			
Base	ARElement, ARObject, CollectableElement, DiagnosticAbstractAliasEvent, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 7.1: DiagnosticFimAliasEvent

Class	**DiagnosticAbstractAliasEvent (abstract)**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
Note	This meta-class represents an abstract base class for all diagnostic alias events.			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
–	–	–	–	–

Table 7.2: DiagnosticAbstractAliasEvent

7.3 Function Identifier

The support of the Fim as an extension of the Dem meta-model requires the modeling of additional meta-classes to describe the various ways of possible mappings between DiagnosticEvents and Fids (or, in terms of the meta-model, DiagnosticFunctionIdentifier).

[TPS_DEXT_01121] Semantics of DiagnosticFunctionIdentifier ⌈ A DiagnosticFunctionIdentifier can be inhibited by different sources, i.e. in different ways:

Event This corresponds to the DiagnosticEvent, as defined in chapter 6.2.

Fim Event Group This represents a group of events that is defined in the scope of the Fim (within the Fim terminology, this is also known as a *summary event*). In the meta-model, a *Fim Event Group* is represented by means of meta-class DiagnosticFimEventGroup.

To emphasize the locality the corresponding meta-class has been named DiagnosticFimEventGroup. DiagnosticFimEventGroup is able to reference 0..* DiagnosticEvents in the role event.

This way, the membership of DiagnosticEvents in the hypothetical group of events formed by the DiagnosticFimEventGroup is expressed.

⌋(*RS_DEXT_00060*)

Class	**DiagnosticFunctionIdentifier**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
Note	This meta-class represents a diagnostic function identifier (a.k.a. FID). **Tags:** atp.recommendedPackage=DiagnosticFunctionIdentifiers			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
–	–	–	–	–

Table 7.3: DiagnosticFunctionIdentifier

Finally all sources will refer to DiagnosticEvents.

7.4 Mapping between Inhibition Source and Diagnostic Event

The configuration of the Fim requires the clarification of the relation between a particular inhibition source (modeled as DiagnosticFunctionInhibitSource) and one or many diagnostic events (modeled as DiagnosticEvent).

The easiest way to provide this information is the usage of the DiagnosticInhibitSourceEventMapping that is able to create an association between a DiagnosticInhibitSourceEventMapping on the one hand and either a DiagnosticEvent or a DiagnosticFimEventGroup on the other hand.

This is only possible if the DiagnosticEvents referenced by the DiagnosticInhibitSourceEventMapping already exist. This existence is subject to the development workflow and may or may not apply. For more details, please refer to Figure 7.1.

Class	**DiagnosticInhibitSourceEventMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
Note	This meta-class represents the ability to map a DiagnosticFunctionInhibitSource directly to alternatively one DiagnosticEvent or one DiagnosticFimSummaryEvent. This model element shall be used if the approach via the alias events is not applicable, i.e. when diagnostic events defined by the Dem are already available at the time the Fim configuration within the diagnostic extract is created. **Tags:** atp.recommendedPackage=DiagnosticInhibitSourceEventMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
diagnosticEvent	DiagnosticEvent	0..1	ref	This represents the reference to the diagnostic event.
eventGroup	DiagnosticFimEventGroup	0..1	ref	This represents the reference to the event group
inhibitionSource	DiagnosticFunctionInhibitSource	0..1	ref	This represents the reference to the inhibition source.

Table 7.4: DiagnosticInhibitSourceEventMapping

7.5 Alias Event Mapping

[TPS_DEXT_01095] Definition of "alias" diagnostic event for the creation of a Fim configuration in the diagnostic extract ⌈ A pre-configuration of the Fim function inhibition can be created on the basis of the following meta-classes:

DiagnosticFimAliasEventMapping in this case the definition of "alias" diagnostic event corresponds to a single DiagnosticEvent

DiagnosticFimAliasEventGroupMapping in this case the definition of a group of "alias" diagnostic events corresponds to a group of single DiagnosticEvents

⌋(*RS_DEXT_00061*, *RS_DEXT_00062*)

Class	**DiagnosticFimAliasEventMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticEvent			
Note	This meta-class represents the ability to model the mapping of a DiagnosticEvent to a DiagnosticAliasEvent. By this means the "preliminary" modeling by way of a DiagnosticAliasEvent is further substantiated. **Tags:** atp.recommendedPackage=DiagnosticFimEventMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
actualEvent	DiagnosticEvent	0..1	ref	This represents the reference to the actual diagnostic event.
aliasEvent	DiagnosticFimAliasEvent	0..1	ref	This represents the reference to the alias event.

Table 7.5: DiagnosticFimAliasEventMapping

Class	**DiagnosticFimAliasEventGroup**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
Note	This meta-class represents the ability to define an alias for a Fim summarized event. This alias can be used in early phases of the configuration process until a further refinement is possible. **Tags:** atp.recommendedPackage=DiagnosticFimAliasEventGroups			
Base	ARElement, ARObject, CollectableElement, DiagnosticAbstractAliasEvent, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, Packageable Element, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
groupedAliasEvent	DiagnosticFimAliasEvent	*	ref	By means of this reference the grouping of DiagnosticAliasEvents within the DiagnosticFimSummaryEvent can be specified.

Table 7.6: DiagnosticFimAliasEventGroup

Class	**DiagnosticFimAliasEventGroupMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
Note	This meta-class represents the ability to map a DiagnosticFimEventGroup to a DiagnosticFimAliasEventGroup. By this means the "preliminary" modeling by way of a DiagnosticFimAliasEventGroup is further substantiated. **Tags:** atp.recommendedPackage=DiagnosticFimAliasEventGroupMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

actualEvent	DiagnosticFimEventGroup	0..1	ref	This represents the reference to the actual summary event.
aliasEvent	DiagnosticFimAliasEventGroup	0..1	ref	This represents the reference to the alias summary event.

Table 7.7: DiagnosticFimAliasEventGroupMapping

Ultimately, the modeling approach for the Fim starts at the definition of the concept of a function itself by means of the meta-class `DiagnosticFunctionIdentifier`.

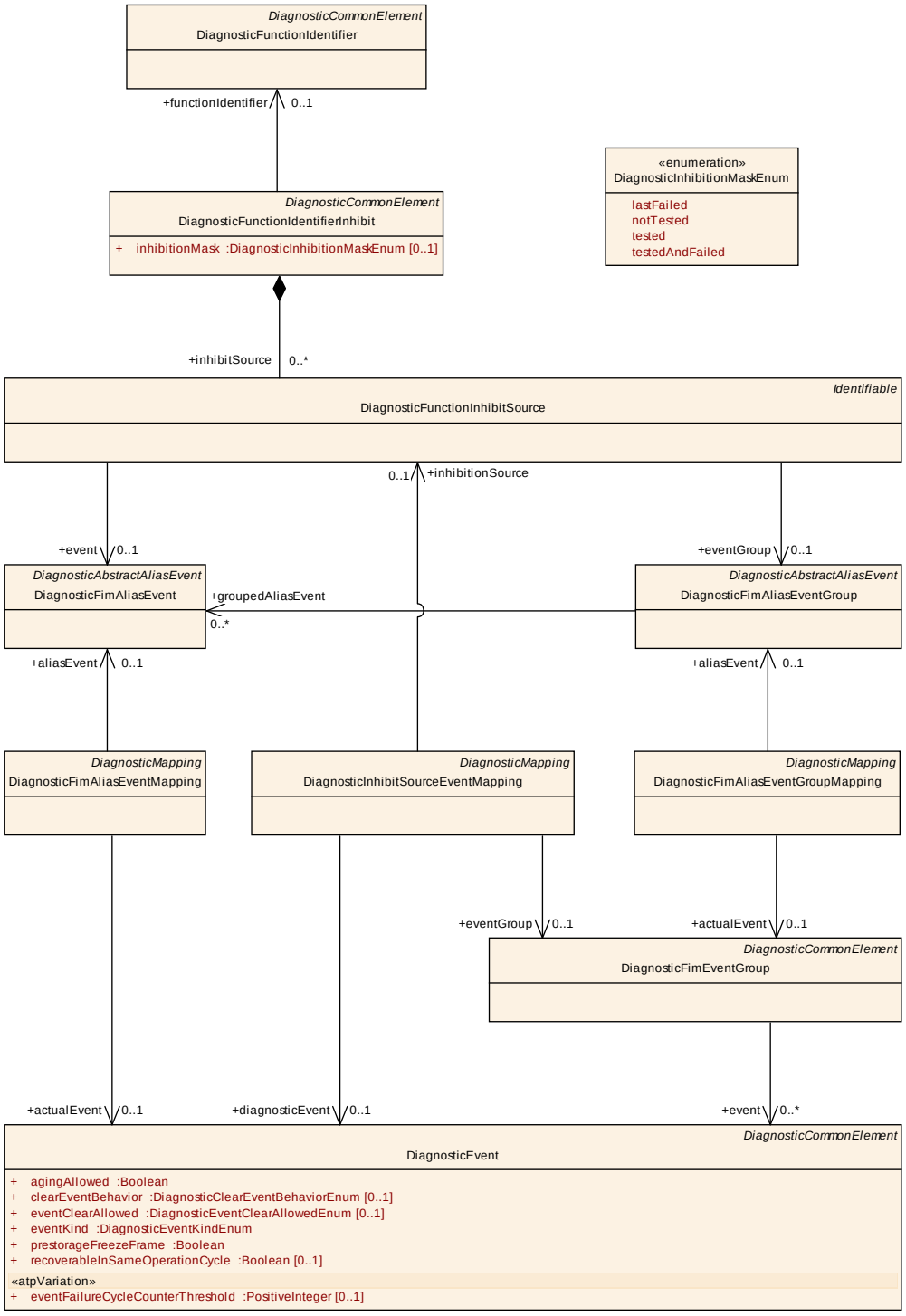

Figure 7.1: Big Picture of Fim diagnostics configuration

Class	**DiagnosticFunctionIdentifierInhibit**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
Note	This meta-class represents the ability to define the inhibition of a specific function identifier within the Fim configuration. **Tags:** atp.recommendedPackage=DiagnosticFunctionIdentifierInhibits			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
functionIdentifier	DiagnosticFunctionIdentifier	0..1	ref	This represents the corresponding function identifier.
inhibitSource	DiagnosticFunctionInhibitSource	*	aggr	This represents a collection of DiagnosticFunctionInhibitSource that contribute to the configuration of the enclosing DiagnosticFunctionIdentiferInhibit.
inhibitionMask	DiagnosticInhibitionMaskEnum	0..1	attr	This represents the value of the inhibition mask behavior.

Table 7.8: DiagnosticFunctionIdentifierInhibit

Class	**DiagnosticFunctionInhibitSource**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
Note	This meta-class represents the ability to define an inhibition source in the context of the Fim configuration.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
event	DiagnosticFimAliasEvent	0..1	ref	This represents the alias event appllicable for the referencing inhibition source.
eventGroup	DiagnosticFimAliasEventGroup	0..1	ref	This represents the event group applicable for the referencing inhibition source.

Table 7.9: DiagnosticFunctionInhibitSource

Enumeration	**DiagnosticInhibitionMaskEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Fim
Note	This meta-class represents the ability to define different kinds of inhibition mask behavior.
Literal	*Description*
lastFailed	This represents the inhibition mask behavior "last failed". **Tags:** atp.EnumerationValue=0
notTested	This represents the inhibition mask behavior "not tested". **Tags:** atp.EnumerationValue=1
tested	This represents the inhibition mask behavior "tested". **Tags:** atp.EnumerationValue=3
testedAndFailed	This represents the inhibition mask behavior "tested and failed". **Tags:** atp.EnumerationValue=2

Table 7.10: DiagnosticInhibitionMaskEnum

[TPS_DEXT_01096] Semantics of DiagnosticFunctionInhibitSource ⌈ The *function inhibition* itself is modeled by means of a different meta-class named DiagnosticFunctionIdentifierInhibit. This meta-class, in turn, references the DiagnosticFunctionIdentifier and it also aggregates 0..* instances of a further meta-class named DiagnosticFunctionInhibitSource.

DiagnosticFunctionInhibitSource inherits from Identifiable in order to be able to utilize attributes like desc, adminData, and introduction for the purpose of textually explaining the reason for defining a certain inhibit source. ⌋ *(RS_DEXT_00060)*

[TPS_DEXT_01097] Standardized value of StructuredReq.category for the modeling of DiagnosticFunctionInhibitSource ⌈ If DiagnosticFunctionInhibitSource.introduction.structuredReq is used to document the reason for a specific function inhibition then DiagnosticFunctionInhibitSource.introduction.structuredReq.category shall be set to the value InhibitReason. ⌋ *(RS_DEXT_00060)*

[constr_1453] References from DiagnosticFunctionInhibitSource ⌈ Each DiagnosticFunctionInhibitSource may either reference one of the following meta-classes in their respective roles:

- DiagnosticFimAliasEventMapping in the role event
- DiagnosticFimAliasEventGroupMapping in the role eventGroup

⌋ *()*

Class	**DiagnosticFimEventGroup**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Fim			
Note	This meta-class represents the ability to model a Fim event group, also known as a summary event in Fim terminology. This represents a group of single diagnostic events. **Tags:** atp.recommendedPackage=DiagnosticFimEventGroups			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
event	DiagnosticEvent	*	ref	This reference represents the way of grouping diagnostic events into a summary event in the context of the Fim.

Table 7.11: DiagnosticFimEventGroup

[TPS_DEXT_01098] Semantics of attribute DiagnosticFunctionInhibitSource.event ⌈ If the reference DiagnosticFunctionInhibitSource.event

exists this means the inhibition applies for a single DiagnosticEvent ⌋ *(RS_DEXT_00060, RS_DEXT_00061)*

[TPS_DEXT_01099] Semantics of attribute DiagnosticFunctionInhibitSource.eventGroup ⌈ If the reference DiagnosticFunctionInhibitSource.eventGroup exists this means that the inhibition effectively applies for a group of DiagnosticEvents where the actual grouping is defined in the scope of the Fim. ⌋*(RS_DEXT_00060, RS_DEXT_00061)*

[TPS_DEXT_01100] Consequence of the existence of DiagnosticFimAliasEventMapping ⌈ The references from meta-class DiagnosticFimAliasEventMapping in the roles actualEvent and aliasEvent do not necessarily have to exist at the same time. The following rules apply:

Only actualEvent exists: In this case there is no DiagnosticFimAliasEvent available and the actual DiagnosticEvent can directly be taken for creating the Fim configuration.

Only aliasEvent exists: In this case the configuration is considered incomplete and only defines the intended semantics from the point of view of a pre-configuration of the Fim.

Both actualEvent and aliasEvent exist: In this case it is obvious that the configuration has undergone a pre-configuration step in which a DiagnosticFimAliasEvent has been used. However, since the DiagnosticEvent in the role actualEvent exists the Diagnostic Extract is considered complete with respect to this aspect of the Fim configuration.

⌋*(RS_DEXT_00060, RS_DEXT_00061)*

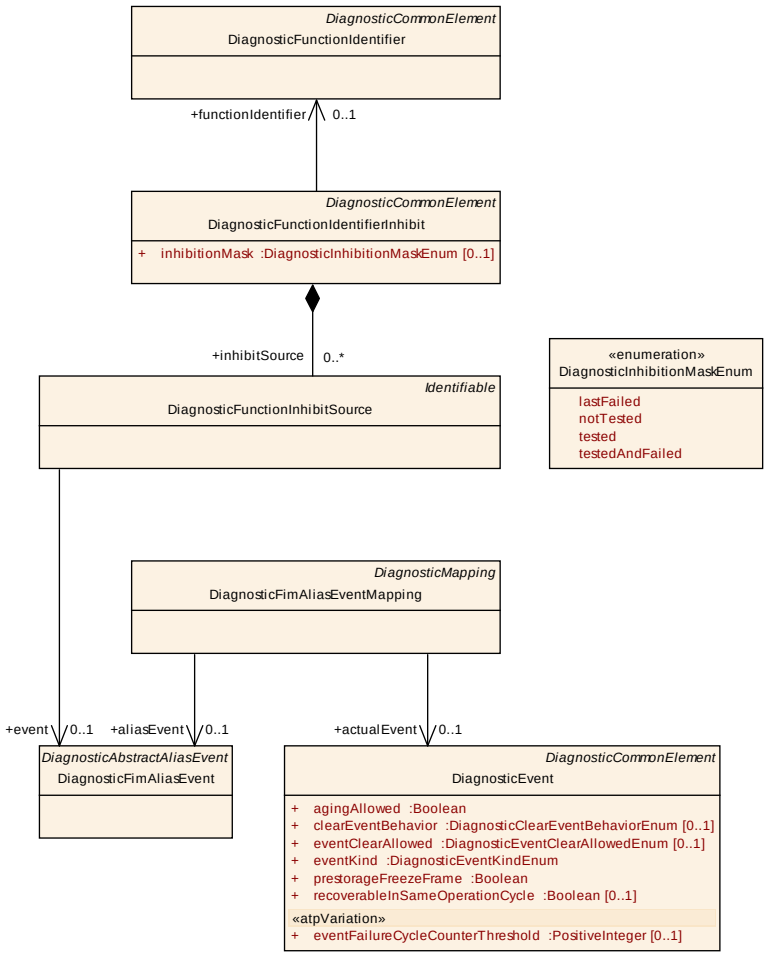

Figure 7.2: Inhibition of events in Fim for diagnostics configuration

[TPS_DEXT_01101] Consequence of the existence of `DiagnosticFimAliasEventGroupMapping` ⌈ The references from meta-class `DiagnosticFimAliasEventGroupMapping` in the roles `actualEvent` and `aliasEvent` do not necessarily have to exist at the same time. The following rules apply:

Only `actualEvent` exists: In this case there is no `DiagnosticFimAliasEventGroup` available and the actual `DiagnosticFimEventGroup` can directly be taken for creating the Fim configuration.

Only `aliasEvent` exists: In this case the configuration is considered incomplete and only defines the intended semantics from the point of view of a pre-configuration of the Fim.

Both `actualEvent` and `aliasEvent` exist: In this case it is obvious that the configuration has undergone a pre-configuration step in which a `DiagnosticFimAliasEventGroup` has been used. However, since the `DiagnosticFimEventGroup` in the role `actualEvent` exists the Diagnostic Extract is considered complete with respect to this aspect of the Fim configuration.

⌋*(RS_DEXT_00060, RS_DEXT_00061)*

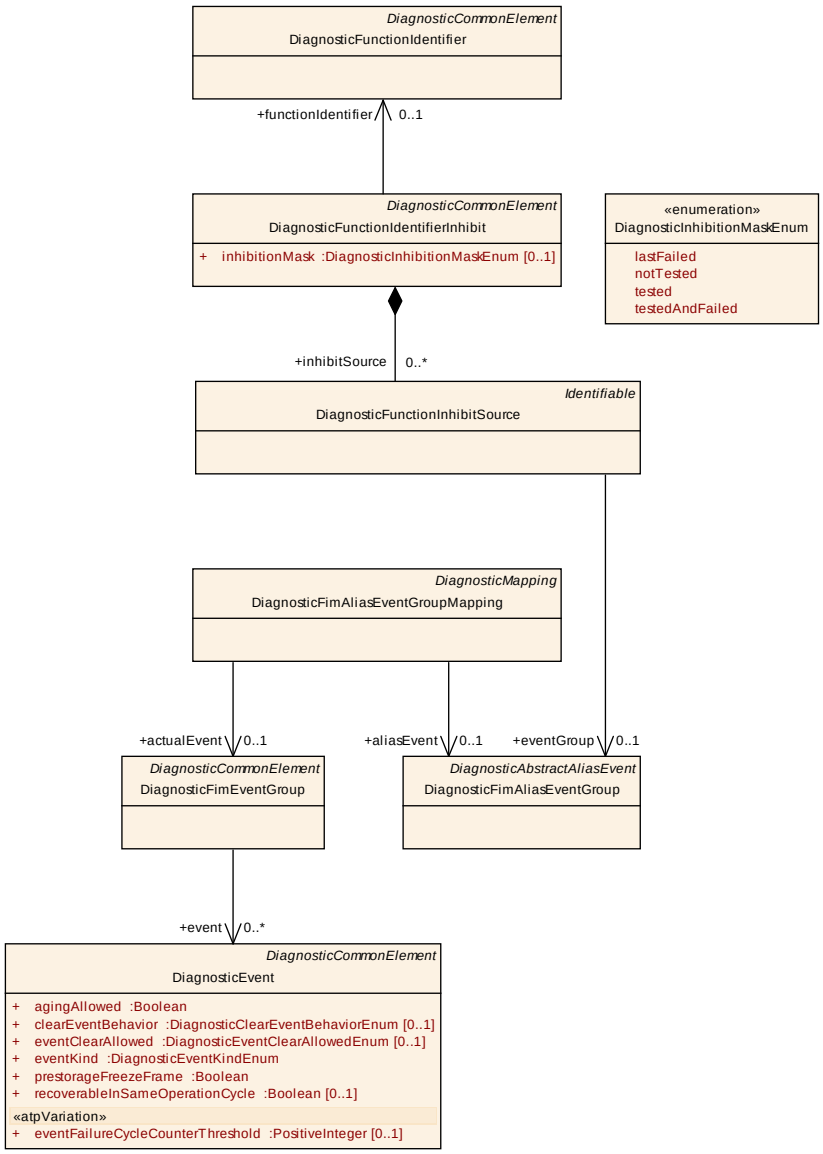

Figure 7.3: Inhibition of event groups in Fim for diagnostics configuration

7.6 Mapping of Function Identifiers to the corresponding Monitors

Beside the ability to model a function inhibition itself, it is equally important that this inhibition can be connected to the applicable monitors.

This relation can be expressed by means of the `DiagnosticFimFunctionMapping`. The details are sketched in Figure 7.4.

[TPS_DEXT_01102] Semantics of `DiagnosticFimFunctionMapping` ⌈ The meta-class `DiagnosticFimFunctionMapping` represents the ability to map a `DiagnosticFunctionIdentifier` to a `SwcServiceDependency` that addresses function inhibition. ⌋*(RS_DEXT_00063)*

[constr_1454] `DiagnosticFimFunctionMapping` shall only reference a `SwcServiceDependency` that aggregates `FunctionInhibitionNeeds` ⌈ A `DiagnosticFimFunctionMapping` shall only reference a `SwcServiceDependency` that aggregates `FunctionInhibitionNeeds` in the role `serviceNeeds`. ⌋()

The modeling of this mapping is closely related to the existing modeling of mappings that involve `SwcServiceDependency` in the context of the diagnostic extract.

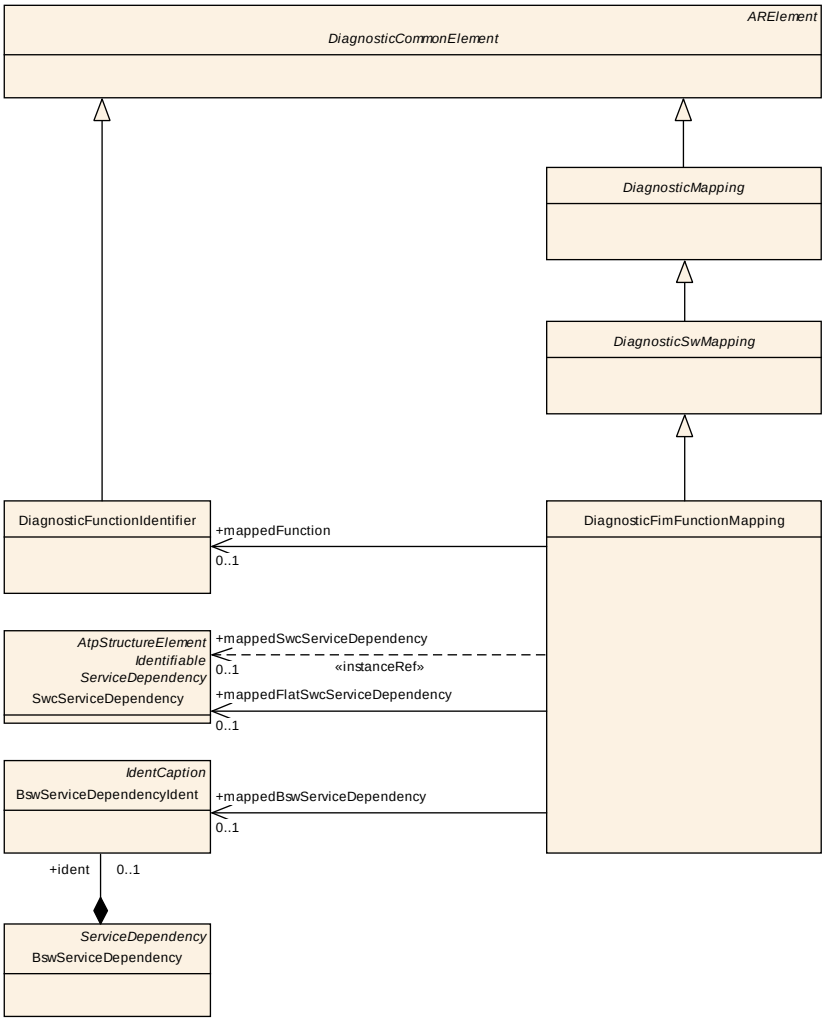

Figure 7.4: Mapping of function identifiers to the corresponding monitors

Class	**DiagnosticFimFunctionMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::ServiceMapping			
Note	This meta-class represents the ability to define a mapping between a function identifier (FID) and the corresponding SwcServiceDependency in the application software resp. basic software. **Tags:** atp.recommendedPackage=DiagnosticFimFunctionMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, DiagnosticSwMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

mappedBswServiceDependency	BswServiceDependencyIdent	0..1	ref	This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.
mappedFlatSwcServiceDependency	SwcServiceDependency	0..1	ref	This represents the ability to refer to an AtomicSwComponentType that is available without the definition of how it will be embedded into the component hierarchy.
mappedFunction	DiagnosticFunctionIdentifier	0..1	ref	This represents the mapped FID.
mappedSwcServiceDependency	SwcServiceDependency	0..1	iref	This represents the ability to point into the component hierarchy (under possible consideration of the rootSoftwareComposition).

Table 7.12: DiagnosticFimFunctionMapping

Class	**FunctionInhibitionNeeds**			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	Specifies the abstract needs on the configuration of the Function Inhibition Manager for one Function Identifier (FID). This class currently contains no attributes. Its name can be regarded as a symbol identifying the FID from the viewpoint of the component or module which owns this class.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable, ServiceNeeds			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table 7.13: FunctionInhibitionNeeds

8 Diagnostics on J1939

8.1 Introduction

Diagnostics on J1939 is to some extent similar to the “rest of the world”. In other words, J1939 uses a Dcm and a Dem in a similar way as other communication networks do.

On the other hand, there are significant differences between J1939 and the “rest of the world” that justify the creation of a separate chapter that focuses entirely on how a diagnostic extract for J1939 shall look like.

That said, a support for Dcm-related model elements for J1939 is mainly happening in the domain of the AUTOSAR System Template [6] rather than in the scope of this document.

The actual configuration of services supported by the J1939Dcm is done by assigning DMx messages in the System Description.

The mapping of the J1939DcmIPdus to the diagnostic services of J1939 Dcm happens during derivation to EcuC.

In the following sections this document focuses on explaining the modeling with respect to the Dem.

8.2 Suspect Parameter Number

In the J1939 terminology, a *Suspect Parameter Number* represents a signal identifier. The numerical value of this signal identifier is stored in the attribute DiagnosticJ1939Spn.spn.

Conceptually, the *Suspect Parameter Number* binds to the SystemSignal, i.e. by attaching a *Suspect Parameter Number* to a SystemSignal a J1939 signal is created.

At the first sight, it may seem like a good idea to attach the spn attribute to the SystemSignal itself. However, this would place a very specific J1939-related model semantics in a very prominent place.

This just doesn't seem right and thanks to the existence of meta-class DiagnosticJ1939SpnMapping it is possible to "inject" the J1939 signal identifier into a SystemSignal without actually touching the SystemSignal.

[TPS_DEXT_01103] Semantics of meta-class DiagnosticJ1939SpnMapping ⌈ The DiagnosticJ1939SpnMapping has the ability to associate a DiagnosticJ1939Spn with a SystemSignal. By this means it is possible to express that a given SystemSignal is taken to transport a J1939 *Suspect Parameter Number*. ⌋ *(RS_DEXT_00064)*

Please note that the modeling of the DiagnosticJ1939SpnMapping also implies that the same SPN can be sent by different J1939NmNodes. This ability is positively supported.

Class	**DiagnosticJ1939Spn**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::J1939			
Note	This meta-class represents the ability to model a J1939 Suspect Parameter Number (SPN). **Tags:** atp.recommendedPackage=DiagnosticJ1939Spns			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
spn	PositiveInteger	1	attr	This attribute represents the concrete numerical identification for the enclosing SPN.

Table 8.1: DiagnosticJ1939Spn

8.3 J1939Dcm-related Modeling

The modeling of the J1939Dcm-related configuration within the Diagnostic Extract extends to the definition of a new attribute of meta-class `J1939Cluster`. The latter provides an attribute named `networkId`. For more information, please refer to [6].

The actual configuration of the services supported by J1939Dcm is done by assigning DMx messages in the System Description.

8.4 Dem-related Modeling

The modeling of Dem-related support for J1939 is centered around the `DiagnosticJ1939Node`. This meta-class literally acts as a sort of "inverted"[1] hub for all the Dem-related model elements. The `DiagnosticJ1939Node` represents a specific function, or in terms of the J1939 terminology, a *Controller Application*.

This aspect is stressed in Figure 8.1.

Since, according to the J1939 concept, each *controller application* represents an independent entity in terms of network management, the following constraint applies:

[constr_1455] Relation of `DiagnosticJ1939Node` to `J1939NmNode` ⌈ Each `J1939NmNode` shall only be referenced in the role `nmNode` by a single `DiagnosticJ1939Node`. ⌋()

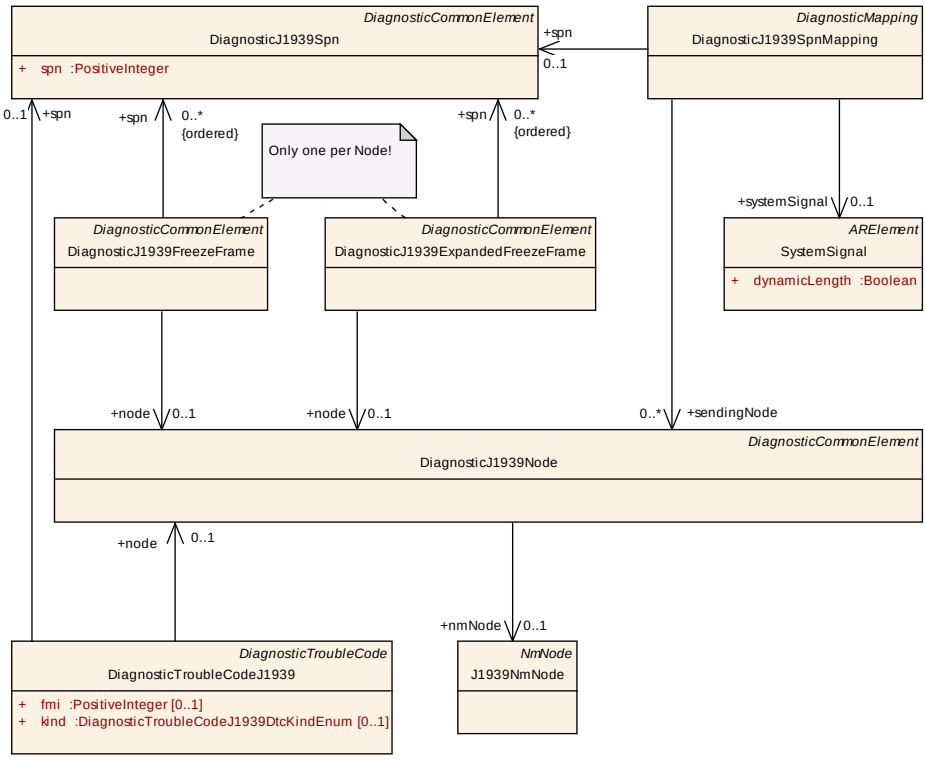

Figure 8.1: Big Picture of J1939 diagnostics configuration

[1] Meaning: several meta-classes are referencing the `DiagnosticJ1939Node`.

J1939 supports the capturing of information similar to the definition of DiagnosticFreezeFrame, as explained in chapter 6.5. In the case of J1939, however, two different kinds of freeze frames can be defined:

[TPS_DEXT_01104] Difference between DiagnosticJ1939FreezeFrame and DiagnosticJ1939ExpandedFreezeFrame ⌈ A DiagnosticJ1939FreezeFrame is transmitted via J1939 **DM04** while a DiagnosticJ1939ExpandedFreezeFrame is transmitted via J1939 **DM24**/**DM25**. ⌋*(RS_DEXT_00065)*

[TPS_DEXT_01105] Relation of DiagnosticJ1939Spn to DiagnosticJ1939FreezeFrame and DiagnosticJ1939ExpandedFreezeFrame ⌈ It is possible that a given DiagnosticJ1939Spn is referenced by both a DiagnosticJ1939FreezeFrame and a DiagnosticJ1939ExpandedFreezeFrame.

In other words, the *Suspect Parameter Number* can be part of a normal freeze frame and, at the same time, an expanded freeze frame. ⌋*(RS_DEXT_00065)*

Class	**DiagnosticJ1939FreezeFrame**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::J1939			
Note	This meta-class represents the ability to model a J1939 Freeze Frame. **Tags:** atp.recommendedPackage=DiagnosticJ1939FreezeFrames			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
node	DiagnosticJ1939Node	0..1	ref	This represents the DiagnosticJ1939Node to which the J1939 freeze frame is associated.
spn (ordered)	DiagnosticJ1939Spn	*	ref	This represents the collection of SPNs that make the J1939 Freeze Frame.

Table 8.2: DiagnosticJ1939FreezeFrame

Class	**DiagnosticJ1939ExpandedFreezeFrame**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::J1939			
Note	This meta-class represents the ability to model an expanded J1939 Freeze Frame. **Tags:** atp.recommendedPackage=DiagnosticJ1939ExpandedFreezeFrames			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
node	DiagnosticJ1939Node	0..1	ref	This represents the DiagnosticJ1939Node to which the J1939 expanded freeze frame is associated.
spn (ordered)	DiagnosticJ1939Spn	*	ref	This represents the collection of SPNs that make the expanded J1939 Freeze Frame.

Table 8.3: DiagnosticJ1939ExpandedFreezeFrame

Enumeration	**DiagnosticTroubleCodeJ1939DtcKindEnum**

Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode
Note	This meta-class represents the ability to further specify a J1939 DTC in terms of its semantics.
Literal	***Description***
serviceOnly	this represents a DTC that is only relevant for service in a garage, reported by e.g. DM53. **Tags:** atp.EnumerationValue=0
standard	This represents a non-specific DTC reported by e.g. DM1. **Tags:** atp.EnumerationValue=1

Table 8.4: DiagnosticTroubleCodeJ1939DtcKindEnum

[TPS_DEXT_01106] Relation of Controller Application to SPN ⌈ It is technically possible that several *Controller Applications*, formally represented by the meta-class `DiagnosticJ1939Node`, can send the same *Suspect Parameter Number*.

In response to this specific characteristic of the J1939 approach, the multiplicity of the reference `DiagnosticJ1939SpnMapping.sendingNode` has been set to 0..*. ⌋ *(RS_DEXT_00064)*

Class	**DiagnosticJ1939SpnMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::J1939			
Note	This meta-class represents the ability to define a mapping between an SPN and a SystemSignal. The existence of a mapping means that neither the SPN nor the SystemSignal need to be updated if the relation between the two changes. **Tags:** atp.recommendedPackage=DiagnosticJ1939SpnMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticMapping, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
sendingNode	DiagnosticJ1939Node	*	ref	This additional reference has a supporting role in that it identifies all sending nodes of a given SPN. It is positively possible thata given SPN is sent by more than one node. Even tough the reference targets the DiagnosticJ1939Node the semantics of the reference is bound to the J1939NmNode that is in turn referenced by the DiagnosticJ1939Node.
spn	DiagnosticJ1939Spn	0..1	ref	This reference goes to the SPN that shall be associated with a SystemSignal.
systemSignal	SystemSignal	0..1	ref	This reference goes to the SystemSignal that shall be associated with an SPN.

Table 8.5: DiagnosticJ1939SpnMapping

Class	**DiagnosticJ1939Node**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::J1939			
Note	This meta-class represents the diagnostic configuration of a J1939 Nm node, which in turn represents a "virtual Ecu" on the J1939 communication bus. **Tags:** atp.recommendedPackage=DiagnosticJ1939Nodes			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
nmNode	J1939NmNode	0..1	ref	This represents the reference to the "virtual Ecu" to which the enclosing DiagnosticJ1939Node is associated.

Table 8.6: DiagnosticJ1939Node

[constr_1456] Valid interval for attribute `DiagnosticTroubleCodeJ1939.fmi` ⌈ The value of the attribute `DiagnosticTroubleCodeJ1939.fmi` shall be in the interval 0..31. ⌋*()*

Please note that the rationale for the existence of [constr_1456] as well as the meaning of the individual values of the attribute `fmi` can be found in the respective SAE J1939 [20] specification.

Class	**DiagnosticTroubleCodeJ1939**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dem::DiagnosticTroubleCode			
Note	This meta-class represents the ability to model specific trouble-code related properties for J1939. **Tags:** atp.recommendedPackage=DiagnosticTroubleCodes			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, DiagnosticTroubleCode, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dtcProps	DiagnosticTroubleCodeProps	0..1	ref	Defined properties associated with the J1939 DTC.
fmi	PositiveInteger	0..1	attr	This attribute represents the behavior of the Failure Mode Indicator.
kind	DiagnosticTroubleCodeJ1939DtcKindEnum	0..1	attr	This attribute further specifies the DTC in terms of its semantics.
node	DiagnosticJ1939Node	0..1	ref	This represents the related DiagnosticJ1939Node.
spn	DiagnosticJ1939Spn	0..1	ref	This represents the releated SPN.

Table 8.7: DiagnosticTroubleCodeJ1939

[TPS_DEXT_01107] Definition of service-only DTC ⌈ It is possible to specify whether given DTCs on J1939 are only relevant for a service session in a garage or workshop. The common property of service-only DTCs is that they shall be located in a memory section that is exclusively used by service-only DTCs. ⌋*(RS_DEXT_00067)*

The statement made by [TPS_DEXT_01107] needs to be secured by two constraints:

[constr_1457] Service-only DTCs shall refer to a common memory section ⌈ All DiagnosticTroubleCodeJ1939 with attribute kind set to the value serviceOnly that reference the same DiagnosticJ1939Node shall also reference the same DiagnosticTroubleCodeProps.memoryDestination. ⌋*()*

[constr_1458] Reference to DiagnosticMemoryDestination ⌈ A DiagnosticMemoryDestination that is referenced by a DiagnosticTroubleCodeJ1939.dtcProps.memoryDestination where the value of attribute DiagnosticTroubleCodeJ1939.kind is set to serviceOnly shall **not be referenced by any other** DiagnosticTroubleCodeJ1939 where attribute kind is set to any other value than serviceOnly. ⌋*()*

8.5 Mapping between Software-Components and Controller Applications

Another aspect of the modeling of J1939 diagnostics is that a mapping between the DiagnosticJ1939Node (which formally represents a *Controller Application*) and the AUTOSAR way of specifying a “function” (by means of the meta-class SwComponentPrototype) is required.

This leads to the definition of the DiagnosticJ1939SwMapping.

[TPS_DEXT_01108] Purpose of the DiagnosticJ1939SwMapping ⌈ The purpose of the DiagnosticJ1939SwMapping is to associate a SwComponentPrototype with a DiagnosticJ1939Node. By this means a concrete functionality is mapped to the abstract J1939 *Controller Application*. ⌋*(RS_DEXT_00066)*

Please note that the basis for this mapping is the existence of a CompositionSwComponentType rather than a System. The mapping can therefore (and this is the main motivation for this kind of modeling) be done independently of the deployment to a concrete project.

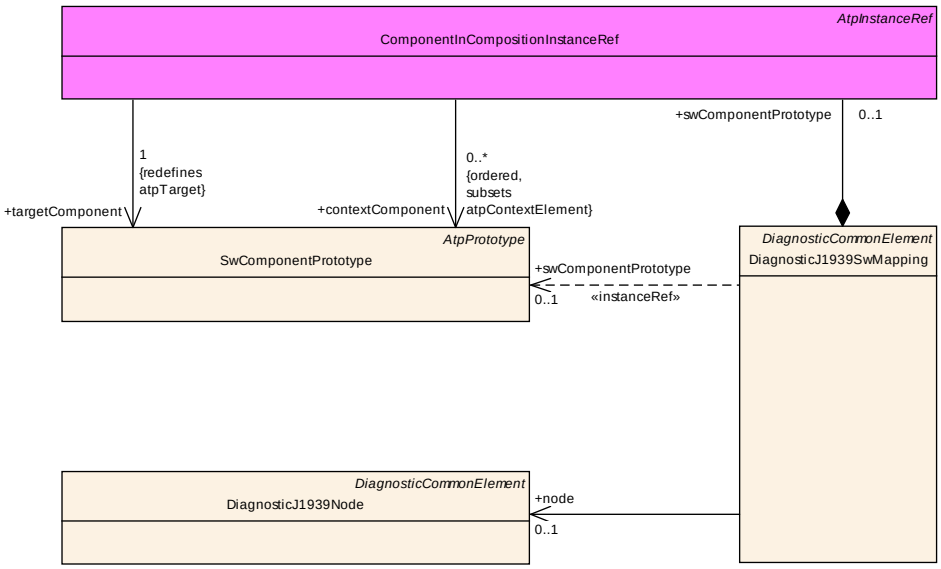

Figure 8.2: Mapping between Software-Components and Controller Applications

Class	**DiagnosticJ1939SwMapping**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::SwMapping			
Note	This meta-class represents the ability to map a piece of application software to a J1939DiagnosticNode. By this means the diagnostic configuration can be associated with the application software. **Tags:** atp.recommendedPackage=DiagnosticJ1939SwMappings			
Base	ARElement, ARObject, CollectableElement, DiagnosticCommonElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
node	DiagnosticJ1939Node	0..1	ref	This represents the mapped DiagnosticJ1939Node.
swComponentPrototype	SwComponentPrototype	0..1	iref	This represents the mapped SwComponentPrototype.

Table 8.8: DiagnosticJ1939SwMapping

A 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.

Class	**AtomicSwComponentType (abstract)**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	An atomic software component is atomic in the sense that it cannot be further decomposed and distributed across multiple ECUs.			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, SwComponentType			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
internalBehavior	SwcInternalBehavior	0..1	aggr	The SwcInternalBehaviors owned by an AtomicSwComponentType can be located in a different physical file. Therefore the aggregation is «atpSplitable». **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=internalBehavior, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
symbolProps	SymbolProps	0..1	aggr	This represents the SymbolProps for the AtomicSwComponentType. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=shortName

Table A.1: AtomicSwComponentType

Class	**AtpInstanceRef (abstract)**			
Package	M2::AUTOSARTemplates::GenericStructure::AbstractStructure			
Note	An M0 instance of a classifier may be represented as a tree rooted at that instance, where under each node come the sub-trees representing the instances which act as features under that node. An instance ref specifies a navigation path from any M0 tree-instance of the base (which is a classifier) to a leaf (which is an instance of the target).			
Base	ARObject			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
atpBase	AtpClassifier	1	ref	This is the base from which the navigaion path starts. **Stereotypes:** atpAbstract; atpDerived
atpContextElement (ordered)	AtpPrototype	*	ref	This is one particular step in the navigation path. **Stereotypes:** atpAbstract
atpTarget	AtpFeature	1	ref	This is the target of the instance ref. In other words it is the terminal of the navigation path. **Stereotypes:** atpAbstract

Table A.2: AtpInstanceRef

Class	**BaseType (abstract)**			
Package	M2::MSR::AsamHdo::BaseTypes			
Note	This abstract meta-class represents the ability to specify a platform dependant base type.			
Base	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
baseType Definition	BaseTypeDefinition	1	aggr	This is the actual definition of the base type. **Tags:** xml.roleElement=false; xml.roleWrapperElement=false; xml.sequenceOffset=20; xml.typeElement=false; xml.typeWrapperElement=false

Table A.3: BaseType

Class	**BaseTypeDirectDefinition**			
Package	M2::MSR::AsamHdo::BaseTypes			
Note	This BaseType is defined directly (as opposite to a derived BaseType)			
Base	ARObject, BaseTypeDefinition			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
baseType Encoding	BaseTypeEncodingString	1	attr	This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **Tags:** xml.sequenceOffset=90
baseType Size	PositiveInteger	0..1	attr	Describes the length of the data type specified in the container in bits. **Tags:** xml.sequenceOffset=70
byteOrder	ByteOrderEnum	0..1	attr	This attribute specifies the byte order of the base type. **Tags:** xml.sequenceOffset=110
maxBaseTypeSize	PositiveInteger	0..1	attr	Describes the maximum length of the BaseType in bits. **Tags:** atp.Status=obsolete xml.sequenceOffset=80
memAlignment	PositiveInteger	0..1	attr	This attribute describes the alignment of the memory object in bits. E.g. "8" specifies, that the object in question is aligned to a byte while "32" specifies that it is aligned four byte. If the value is set to "0" the meaning shall be interpreted as "unspecified". **Tags:** xml.sequenceOffset=100

nativeDeclaration	NativeDeclarationString	0..1	attr	This attribute describes the declaration of such a base type in the native programming language, primarily in the Programming language C. This can then be used by a code generator to include the necessary declarations into a header file. For example
				
				BaseType with
				`shortName: "MyUnsignedInt"`
				`nativeDeclaration: "unsigned short"`
				
				Results in
				`typedef unsigned short MyUnsignedInt;`
				
				If the attribute is not defined the referring ImplementationDataTypes will not be generated as a typedef by RTE.
				
				If a nativeDeclaration type is given it shall fulfill the characteristic given by basetypeEncoding and baseTypeSize.
				
				This is required to ensure the consistent handling and interpretation by software components, RTE, COM and MCM systems.
				
				Tags: xml.sequenceOffset=120

Table A.4: BaseTypeDirectDefinition

Class	**BswModuleDescription**			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswOverview			
Note	Root element for the description of a single BSW module or BSW cluster. In case it describes a BSW module, the short name of this element equals the name of the BSW module. **Tags:** atp.recommendedPackage=BswModuleDescriptions			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpFeature, AtpStructureElement, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
bswModuleDependency	BswModuleDependency	*	aggr	Describes the dependency to another BSW module. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=20

bswModuleDocumentation	SwComponentDocumentation	0..1	aggr	This adds a documentation to the BSW module. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=bswModuleDocumentation, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=6
expectedEntry	BswModuleEntry	*	ref	Indicates an entry which is required by this module. Replacement of outgoingCallback / requiredEntry. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=expectedEntry, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
implementedEntry	BswModuleEntry	*	ref	Specifies an entry provided by this module which can be called by other modules. This includes "main" functions, interrupt routines, and callbacks. Replacement of providedEntry / expectedCallback. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=implementedEntry, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
internalBehavior	BswInternalBehavior	*	aggr	The various BswInternalBehaviors associated with a BswModuleDescription can be distributed over several physical files. Therefore the aggregation is «atpSplitable». **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=shortName xml.sequenceOffset=65
moduleId	PositiveInteger	0..1	attr	Refers to the BSW Module Identifier defined by the AUTOSAR standard. For non-standardized modules, a proprietary identifier can be optionally chosen. **Tags:** xml.sequenceOffset=5
providedClientServerEntry	BswModuleClientServerEntry	*	aggr	Specifies that this module provides a client server entry which can be called from another parition or core.This entry is declared locally to this context and will be connected to the requiredClientServerEntry of another or the same module via the configuration of the BSW Scheduler. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=45

providedData	VariableDataPrototype	*	aggr	Specifies a data prototype provided by this module in order to be read from another partition or core.The providedData is declared locally to this context and will be connected to the requiredData of another or the same module via the configuration of the BSW Scheduler. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=55
providedModeGroup	ModeDeclarationGroupPrototype	*	aggr	A set of modes which is owned and provided by this module or cluster. It can be connected to the requiredModeGroups of other modules or clusters via the configuration of the BswScheduler. It can also be synchronized with modes provided via ports by an associated ServiceSwComponentType, EcuAbstractionSwComponentType or ComplexDeviceDriverSwComponentType. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=25
releasedTrigger	Trigger	*	aggr	A Trigger released by this module or cluster. It can be connected to the requiredTriggers of other modules or clusters via the configuration of the BswScheduler. It can also be synchronized with Triggers provided via ports by an associated ServiceSwComponentType, EcuAbstractionSwComponentType or ComplexDeviceDriverSwComponentType. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=35
requiredClientServerEntry	BswModuleClientServerEntry	*	aggr	Specifies that this module requires a client server entry which can be implemented on another parition or core.This entry is declared locally to this context and will be connected to the providedClientServerEntry of another or the same module via the configuration of the BSW Scheduler. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=50

requiredData	VariableDataPrototype	*	aggr	Specifies a data prototype required by this module in oder to be provided from another partition or core.The requiredData is declared locally to this context and will be connected to the providedData of another or the same module via the configuration of the BswScheduler. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=60
requiredModeGroup	ModeDeclarationGroupPrototype	*	aggr	Specifies that this module or cluster depends on a certain mode group. The requiredModeGroup is local to this context and will be connected to the providedModeGroup of another module or cluster via the configuration of the BswScheduler. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=30
requiredTrigger	Trigger	*	aggr	Specifies that this module or cluster reacts upon an external trigger.This requiredTrigger is declared locally to this context and will be connected to the providedTrigger of another module or cluster via the configuration of the BswScheduler. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=40

Table A.5: BswModuleDescription

Class	**CompositionSwComponentType**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	A CompositionSwComponentType aggregates SwComponentPrototypes (that in turn are typed by SwComponentTypes) as well as SwConnectors for primarily connecting SwComponentPrototypes among each others and towards the surface of the CompositionSwComponentType. By this means hierarchical structures of software-components can be created. **Tags:** atp.recommendedPackage=SwComponentTypes			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable, SwComponentType			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

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 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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variation Point.shortLabel vh.latestBindingTime=postBuild
connector	SwConnector	*	aggr	SwConnectors have the principal ability to establish a connection among PortPrototypes. They can have many roles in the context of a CompositionSwComponentType. Details are refined by subclasses. 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 AssemblySwConnectors between ApplicationSwComponentTypes and ServiceSwComponentTypes during the ECU integration. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variation Point.shortLabel vh.latestBindingTime=postBuild
constantValueMapping	ConstantSpecificationMappingSet	*	ref	Reference to the ConstantSpecificationMapping to be applied for initValues of PPortComSpecs and RPortComSpec. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=constantValueMapping

dataTypeMapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMapping to be applied for the used ApplicationDataTypes in PortInterfaces. Background: when developing subsystems it may happen that ApplicationDataTypes are used on the surface of CompositionSwComponentTypes. In this case it would be reasonable to be able to also provide the intended mapping to the ImplementationDataTypes. However, this mapping shall be informal and not technically binding for the implementers mainly because the RTE generator is not concerned about the CompositionSwComponentTypes. Rationale: if the mapping of ApplicationDataTypes on the delegated and inner PortPrototype matches then the mapping to ImplementationDataTypes is not impacting compatibility. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=dataTypeMapping
instantiationRTEEventProps	InstantiationRTEEventProps	*	aggr	This allows to define instantiation specific properties for RTE Events, in particular for instance specific scheduling. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortLabel, variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime

Table A.6: CompositionSwComponentType

Class	**CompuMethod**			
Package	M2::MSR::AsamHdo::ComputationMethod			
Note	This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant. **Tags:** atp.recommendedPackage=CompuMethods			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
compuInternalToPhys	Compu	0..1	aggr	This specifies the computation from internal values to physical values. **Tags:** xml.sequenceOffset=80
compuPhysToInternal	Compu	0..1	aggr	This represents the computation from physical values to the internal values. **Tags:** xml.sequenceOffset=90

displayFormat	DisplayFormatString	0..1	attr	This property specifies, how the physical value shall be displayed e.g. in documents or measurement and calibration tools. **Tags:** xml.sequenceOffset=20
unit	Unit	0..1	ref	This is the physical unit of the Physical values for which the CompuMethod applies. **Tags:** xml.sequenceOffset=30

Table A.7: CompuMethod

Class	**DataConstr**			
Package	M2::MSR::AsamHdo::Constraints::GlobalConstraints			
Note	This meta-class represents the ability to specify constraints on data. **Tags:** atp.recommendedPackage=DataConstrs			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataConstrRule	DataConstrRule	*	aggr	This is one particular rule within the data constraints. **Tags:** xml.roleElement=true; xml.roleWrapperElement=true; xml.sequenceOffset=30; xml.typeElement=false; xml.typeWrapperElement=false

Table A.8: DataConstr

Class	**DiagEventDebounceMonitorInternal**			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	This meta-class represents the ability to indicate that the pre-debounce algorithm shall be used by the Dem for this diagnostic monitor. This is related to setting the EcuC choice container DemDebounceAlgorithmClass to DemDebounceMonitorInternal. If the FaultDetectionAlogrithm is already known to be implemented by a specific BswModuleEntry the reference bswModuleEntry points to the function specification. If the FaultDetectionCounter value is accessible at a PortPrototype this PortPrototype shall be referenced by an assignedPort.			
Base	ARObject, DiagEventDebounceAlgorithm, Identifiable, MultilanguageReferrable, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.9: DiagEventDebounceMonitorInternal

Enumeration	**DiagnosticCompareTypeEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition
Note	Enumeration for the type of a comparison of values usually expressed by the following operators: ==, !=, <, <=, >, >=
Literal	***Description***
isEqual	equal **Tags:** atp.EnumerationValue=0
isGreaterOrEqual	greater than or equal **Tags:** atp.EnumerationValue=5
isGreaterThan	greater than **Tags:** atp.EnumerationValue=4
isLessOrEqual	less than or equal **Tags:** atp.EnumerationValue=3
isLessThan	less than **Tags:** atp.EnumerationValue=2
isNotEqual	not equal **Tags:** atp.EnumerationValue=1

Table A.10: DiagnosticCompareTypeEnum

Enumeration	**DiagnosticLogicalOperatorEnum**
Package	M2::AUTOSARTemplates::DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition
Note	Logical AND and OR operation (&&, ||)
Literal	***Description***
logicalAnd	Logical AND **Tags:** atp.EnumerationValue=0
logicalOr	Logical OR **Tags:** atp.EnumerationValue=1

Table A.11: DiagnosticLogicalOperatorEnum

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. **Tags:** atp.recommendedPackage=EcuInstances			
Base	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
associated ComIPduGroup	ISignalIPduGroup	*	ref	With this reference it is possible to identify which ISignalIPduGroups are applicable for which CommunicationConnector/ ECU. Only top level ISignalIPduGroups shall be referenced by an EcuInstance. If an ISignalIPduGroup contains other ISignalIPduGroups than these contained ISignalIPduGroups shall not be referenced by the EcuInstance. Contained ISignalIPduGroups are associated to an EcuInstance via the top level ISignalIPduGroup.
associated PdurIPduGroup	PdurIPduGroup	*	ref	With this reference it is possible to identify which PduR IPdu Groups are applicable for which CommunicationConnector/ ECU.
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.
comConfigurationGwTimeBase	TimeValue	0..1	attr	The period between successive calls to Com_MainFunctionRouteSignals of the AUTOSAR COM module in seconds.
comConfigurationRxTimeBase	TimeValue	0..1	attr	The period between successive calls to Com_MainFunctionRx of the AUTOSAR COM module in seconds.
comConfigurationTxTimeBase	TimeValue	0..1	attr	The period between successive calls to Com_MainFunctionTx of the AUTOSAR COM module in seconds.
comEnableMDTForCyclicTransmission	Boolean	0..1	attr	Enables for the Com module of this EcuInstance the minimum delay time monitoring for cyclic and repeated transmissions (TransmissionModeTiming has cyclicTiming assigned or eventControlledTiming with numberOfRepetitions > 0).
commController	CommunicationController	1..*	aggr	CommunicationControllers of the ECU.
connector	CommunicationConnector	*	aggr	All channels controlled by a single controller.
diagnosticAddress	Integer	0..1	attr	An ECU specific ID for responses of diagnostic routines.

diagnostic Props	DiagnosticEcuP rops	0..1	aggr	This represents the diagnostic-related properties of an entire ECU. **Tags:** atp.Status=obsolete
ethSwitchP ortGroupD erivation	Boolean	0..1	attr	Defines whether the derivation of SwitchPortGroups based on VLAN and/or CouplingPort.pncMapping shall be performed for this EcuInstance. If not defined the derivation shall not be done.
partition	EcuPartition	*	aggr	Optional definition of Partitions within an Ecu.
pnResetTi me	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.
pncPrepar eSleepTim er	TimeValue	0..1	attr	Time in seconds the PNC state machine shall wait in PNC_PREPARE_SLEEP.
sleepMode Supported	Boolean	1	attr	Specifies whether the ECU instance may be put to a "low power mode" • true: sleep mode is supported • false: sleep mode is not supported Note: This flag may only be set to "true" if the feature is supported by both hardware and basic software.
v2xSuppor ted	V2xSupportEnu m	0..1	attr	This attribute is used to control the existence of the V2X stack on the given EcuInstance.
wakeUpOv erBusSupp orted	Boolean	1	attr	Driver support for wakeup over Bus.

Table A.12: EcuInstance

Class	**ISignal**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Signal of the Interaction Layer. The RTE supports a "signal fan-out" where the same System Signal is sent in different SignalIPdus to multiple receivers. To support the RTE "signal fan-out" each SignalIPdu contains ISignals. If the same System Signal is to be mapped into several SignalIPdus there is one ISignal needed for each ISignalToIPduMapping. ISignals describe the Interface between the Precompile configured RTE and the potentially Postbuild configured Com Stack (see ECUC Parameter Mapping). In case of the SystemSignalGroup an ISignal must be created for each SystemSignal contained in the SystemSignalGroup. **Tags:** atp.recommendedPackage=ISignals			
Base	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataTransformation	DataTransformation	0..1	ref	Optional reference to a DataTransformation which represents the transformer chain that is used to transform the data that shall be placed inside this ISignal. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=dataTransformation, variationPoint.shortLabel vh.latestBindingTime=codeGenerationTime
dataTypePolicy	DataTypePolicyEnum	1	attr	With the aggregation of SwDataDefProps an ISignal specifies how it is represented on the network. This representation follows a particular policy. Note that this causes some redundancy which is intended and can be used to support flexible development methodology as well as subsequent integrity checks. If the policy "networkRepresentationFromComSpec" is chosen the network representation from the ComSpec that is aggregated by the PortPrototype shall be used. If the "override" policy is chosen the requirements specified in the PortInterface and in the ComSpec are not fulfilled by the networkRepresentationProps. In case the System Description doesn't use a complete Software Component Description (VFB View) the "legacy" policy can be chosen.
iSignalProps	ISignalProps	0..1	aggr	Additional optional ISignal properties that may be stored in different files. **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=iSignalProps

iSignalType	ISignalTypeEnum	0..1	attr	This attribute defines whether this iSignal is an array that results in a UINT8_N / UINT8_DYN ComSignalType in the COM configuration or a primitive type.
initValue	ValueSpecification	0..1	aggr	Optional definition of a ISignal's initValue in case the System Description doesn't use a complete Software Component Description (VFB View). This supports the inclusion of legacy system signals. This value can be used to configure the Signal's "InitValue". If a full DataMapping exist for the SystemSignal this information may be available from a configured SenderComSpec and ReceiverComSpec. In this case the initvalues in SenderComSpec and/or ReceiverComSpec override this optional value specification. Further restrictions apply from the RTE specification.
length	Integer	1	attr	Size of the signal in bits. The size needs to be derived from the mapped VariableDataPrototype according to the mapping of primitive DataTypes to BaseTypes as used in the RTE. Indicates maximum size for dynamic length signals. The ISignal length of zero bits is allowed.
networkRepresentationProps	SwDataDefProps	0..1	aggr	Specification of the actual network representation. The usage of SwDataDefProps for this purpose is restricted to the attributes compuMethod and baseType. The optional baseType attributes "memAllignment" and "byteOrder" shall not be used. The attribute "dataTypePolicy" in the SystemTemplate element defines whether this network representation shall be ignored and the information shall be taken over from the network representation of the ComSpec. If "override" is chosen by the system integrator the network representation can violate against the requirements defined in the PortInterface and in the network representation of the ComSpec. In case that the System Description doesn't use a complete Software Component Description (VFB View) this element is used to configure "ComSignalDataInvalidValue" and the Data Semantics.
systemSignal	SystemSignal	1	ref	Reference to the System Signal that is supposed to be transmitted in the ISignal.

timeoutSubstitutionValue	ValueSpecification	0..1	aggr	Defines and enables the ComTimeoutSubstituition for this ISignal.
transformationISignalProps	TransformationISignalProps	*	aggr	A transformer chain consists of an ordered list of transformers. The ISignal specific configuration properties for each transformer are defined in the TransformationISignalProps class. The transformer configuration properties that are common for all ISignals are described in the TransformationTechnology class.

Table A.13: ISignal

Class	**ISignalIPduGroup**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	The AUTOSAR COM Layer is able to start and to stop sending and receiving configurable groups of I-Pdus during runtime. An ISignalIPduGroup contains either ISignalIPdus or ISignalIPduGroups. **Tags:** atp.recommendedPackage=ISignaliPduGroup			
Base	ARObject, CollectableElement, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
communicationDirection	CommunicationDirectionType	1	attr	This attribute determines in which direction IPdus that are contained in this IPduGroup will be transmitted (communication direction can be either In or Out).
communicationMode	String	1	attr	This attribute defines the use-case for this ISignalIPduGroup (e.g. diagnostic, debugging etc.). For example, in a diagnostic mode all IPdus - which are not involved in diagnostic - are disabled. The use cases are not limited to a fixed enumeration and can be specified as a string.
containedISignalIPduGroup	ISignalIPduGroup	*	ref	An I-Pdu group can be included in other I-Pdu groups. Contained I-Pdu groups shall not be referenced by the EcuInstance.
iSignalIPdu	ISignalIPdu	*	ref	Reference to a set of Signal I-Pdus, which are contained in the ISignal I-Pdu Group. atpVariation: The content of a ISignal I-Pdu group can vary (->vehicle modes). **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
nmPdu	NmPdu	*	ref	Reference to a set of NmPdus with NmUserData, which are contained in the ISignalIPduGroup. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild

Table A.14: ISignalIPduGroup

Class	**IdentCaption (abstract)**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::RPTScenario			
Note	This meta-class represents the caption. This allows having some meta classes optionally identifiable.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.15: IdentCaption

Class	**Identifiable (abstract)**			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable			
Note	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.			
Base	ARObject, MultilanguageReferrable, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
desc	MultiLanguage OverviewParagraph	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". **Tags:** xml.sequenceOffset=-60
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. **Tags:** xml.sequenceOffset=-50
adminData	AdminData	0..1	aggr	This represents the administrative data for the identifiable object. **Tags:** 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. **Tags:** xml.sequenceOffset=-25

introduction	Documentation Block	0..1	aggr	This represents more information about how the object in question is built or is used. Therefore it is a DocumentationBlock. **Tags:** 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. **Tags:** xml.attribute=true

Table A.16: Identifiable

Class	**InternalBehavior (abstract)**			
Package	M2::AUTOSARTemplates::CommonStructure::InternalBehavior			
Note	Common base class (abstract) for the internal behavior of both software components and basic software modules/clusters.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

constantMemory	ParameterDataPrototype	*	aggr	Describes a read only memory object containing characteristic value(s) implemented by this InternalBehavior. The shortName of ParameterDataPrototype has to be equal to the ”C’ identifier of the described constant. The characteristic value(s) might be shared between SwComponentPrototypes 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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
constantValueMapping	ConstantSpecificationMappingSet	*	ref	Reference to the ConstanSpecificationMapping to be applied for the particular InternalBehavior **Stereotypes:** atpSplitable **Tags:** atp.Splitkey=constantValueMapping
dataTypeMapping	DataTypeMappingSet	*	ref	Reference to the DataTypeMapping to be applied for the particular InternalBehavior **Stereotypes:** atpSplitable **Tags:** 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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
exclusiveAreaNestingOrder	ExclusiveAreaNestingOrder	*	aggr	This represents the set of ExclusiveAreaNestingOrder owned by the InternalBehavior. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

staticMemory	VariableDataPrototype	*	aggr	Describes a read and writeable static memory object representing measurerment 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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime

Table A.17: InternalBehavior

Class	≪`atpVariation`≫ **J1939Cluster**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::Fibex4Can::CanTopology			
Note	J1939 specific cluster attributes. **Tags:** atp.recommendedPackage=CommunicationClusters			
Base	ARObject, AbstractCanCluster, CollectableElement, CommunicationCluster, FibexElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
networkId	PositiveInteger	0..1	attr	This represents the network ID for the J1939 cluster.
request2Support	Boolean	0..1	attr	Enables support for the Request2 PGN (RQST2).
usesAddressArbitration	Boolean	0..1	attr	Defines whether the nodes attached to this channel use an initial address claim, and whether they react to contending address claims of other nodes. True: The initial address claim is sent, and the node reacts to address claims of other nodes. False: The node only sends an address claim upon request, and does not care for contending address claims.

Table A.18: J1939Cluster

Class	**J1939DcmIPdu**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	Represents the IPdus handled by J1939Dcm. **Tags:** atp.recommendedPackage=Pdus			
Base	ARObject, CollectableElement, FibexElement, IPdu, Identifiable, Multilanguage Referrable, PackageableElement, Pdu, Referrable			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
diagnostic MessageType	PositiveInteger	0..1	attr	This attribute is used to identify the actual DMx message, e.g 1 means DM01, etc.

Table A.19: J1939DcmIPdu

Class	**J1939NmNode**			
Package	M2::AUTOSARTemplates::SystemTemplate::NetworkManagement			
Note	J1939 specific NM Node attributes.			
Base	ARObject, Identifiable, MultilanguageReferrable, NmNode, Referrable			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
nodeName	J1939NodeName	0..1	aggr	NodeName configuration

Table A.20: J1939NmNode

Class	**ModeDeclaration**			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	Declaration of one Mode. The name and semantics of a specific mode is not defined in the meta-model.			
Base	ARObject, AtpClassifier, AtpFeature, AtpStructureElement, Identifiable, MultilanguageReferrable, Referrable			
Attribute	*Type*	*Mul.*	*Kind*	*Note*
value	PositiveInteger	0..1	attr	The RTE shall take the value of this attribute for generating the source code representation of this ModeDeclaration.

Table A.21: ModeDeclaration

Class	**ModeDeclarationGroup**			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	A collection of Mode Declarations. Also, the initial mode is explicitly identified. **Tags:** atp.recommendedPackage=ModeDeclarationGroups			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	*Type*	*Mul.*	*Kind*	*Note*

initialMode	ModeDeclaration	1	ref	The initial mode of the ModeDeclarationGroup. This mode is active before any mode switches occurred.
modeDeclaration	ModeDeclaration	1..*	aggr	The ModeDeclarations collected in this ModeDeclarationGroup. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=blueprintDerivationTime
modeManagerErrorBehavior	ModeErrorBehavior	0..1	aggr	This represents the ability to define the error behavior expected by the mode manager in case of errors on the mode user side (e.g. terminated mode user).
modeTransition	ModeTransition	*	aggr	This represents the avaliable ModeTransitions of the ModeDeclarationGroup
modeUserErrorBehavior	ModeErrorBehavior	0..1	aggr	This represents the definition of the error behavior expected by the mode user in case of errors on the mode manager side (e.g. terminated mode manager).
onTransitionValue	PositiveInteger	0..1	attr	The value of this attribute shall be taken into account by the RTE generator for programmatically representing a value used for the transition between two statuses.

Table A.22: ModeDeclarationGroup

Class	**ModeDeclarationGroupPrototype**			
Package	M2::AUTOSARTemplates::CommonStructure::ModeDeclaration			
Note	The ModeDeclarationGroupPrototype specifies a set of Modes (ModeDeclarationGroup) which is provided or required in the given context.			
Base	ARObject, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
swCalibrationAccess	SwCalibrationAccessEnum	0..1	attr	This allows for specifying whether or not the enclosing ModeDeclarationGroupPrototype can be measured at run-time.
type	ModeDeclarationGroup	1	tref	The "collection of ModeDeclarations" (= ModeDeclarationGroup) supported by a component **Stereotypes:** isOfType

Table A.23: ModeDeclarationGroupPrototype

Class	**ModeInBswModuleDescriptionInstanceRef**			
Package	M2::AUTOSARTemplates::BswModuleTemplate::BswOverview::InstanceRefs			
Note				
Base	ARObject,AtpInstanceRef			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

base	BswModuleDescription	1	ref	**Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
contextModeDeclarationGroup	ModeDeclarationGroupPrototype	1	ref	**Tags:** xml.sequenceOffset=20
targetMode	ModeDeclaration	1	ref	**Tags:** xml.sequenceOffset=30

Table A.24: ModeInBswModuleDescriptionInstanceRef

Primitive	**NameToken**
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::PrimitiveTypes
Note	This is an identifier as used in xml, e.g. xml-names. Basic difference to Identifier is the fact that it can contain "-". **Tags:** xml.xsd.customType=NMTOKEN-STRING; xml.xsd.type=NMTOKEN

Table A.25: NameToken

Class	**PPortPrototype**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	Component port providing a certain port interface.			
Base	ARObject, AbstractProvidedPortPrototype, AtpBlueprintable, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, PortPrototype, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
providedInterface	PortInterface	1	tref	The interface that this port provides. **Stereotypes:** isOfType

Table A.26: PPortPrototype

Class	**PortInterface (abstract)**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	Abstract base class for an interface that is either provided or required by a port of a software component.			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

isService	Boolean	1	attr	This flag is set if the PortInterface is to be used for communication between an • ApplicationSwComponentType or • ServiceProxySwComponentType or • SensorActuatorSwComponentType or • ComplexDeviceDriverSwComponentType • ServiceSwComponentType • EcuAbstractionSwComponentType and a ServiceSwComponentType (namely an AUTOSAR Service) located on the same ECU. Otherwise the flag is not set.
serviceKind	ServiceProviderEnum	0..1	attr	This attribute provides further details about the nature of the applied service.

Table A.27: PortInterface

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			
Attribute	***Type***	***Mul.***	***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 voilatile data port.
parameterPortAnnotation	ParameterPortAnnotation	*	aggr	Annotations on this parameter port.
senderReceiverAnnotation	SenderReceiverAnnotation	*	aggr	Collection of annotations of this ports sender/receiver communication.

triggerPort Annotation	TriggerPortAnn otation	*	aggr	Annotations on this trigger port.

Table A.28: PortPrototype

Class	**Referrable (abstract)**			
Package	M2::AUTOSARTemplates::GenericStructure::GeneralTemplateClasses::Identifiable			
Note	Instances of this class can be referred to by their identifier (while adhering to namespace borders).			
Base	ARObject			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
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. **Tags:** xml.enforceMinMultiplicity=true; xml.sequenceOffset=-100
shortName Fragment	ShortNameFrag ment	*	aggr	This specifies how the Referrable.shortName is composed of several shortNameFragments. **Tags:** xml.sequenceOffset=-90

Table A.29: Referrable

Class	**SenderReceiverInterface**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::PortInterface			
Note	A sender/receiver interface declares a number of data elements to be sent and received. **Tags:** atp.recommendedPackage=PortInterfaces			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, DataInterface, Identifiable, MultilanguageReferrable, PackageableElement, PortInterface, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dataEleme nt	VariableDataPr ototype	1..*	aggr	The data elements of this SenderReceiverInterface.
invalidation Policy	InvalidationPolic y	*	aggr	InvalidationPolicy for a particular dataElement

Table A.30: SenderReceiverInterface

Class	**ServiceNeeds (abstract)**			
Package	M2::AUTOSARTemplates::CommonStructure::ServiceNeeds			
Note	This expresses the abstract needs that a Software Component or Basic Software Module has on the configuration of an AUTOSAR Service to which it will be connected. "Abstract needs" means that the model abstracts from the Configuration Parameters of the underlying Basic Software.			
Base	ARObject, Identifiable, MultilanguageReferrable, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.31: ServiceNeeds

Class	**SwBaseType**			
Package	M2::MSR::AsamHdo::BaseTypes			
Note	This meta-class represents a base type used within ECU software. **Tags:** atp.recommendedPackage=BaseTypes			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, BaseType, Collectable Element, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
–	–	–	–	–

Table A.32: SwBaseType

Class	**SwComponentPrototype**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Composition			
Note	Role of a software component within a composition.			
Base	ARObject, AtpFeature, AtpPrototype, Identifiable, MultilanguageReferrable, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
type	SwComponentType	1	tref	Type of the instance. **Stereotypes:** isOfType

Table A.33: SwComponentPrototype

Class	**SwComponentType (abstract)**			
Package	M2::AUTOSARTemplates::SWComponentTemplate::Components			
Note	Base class for AUTOSAR software components.			
Base	ARElement, ARObject, AtpBlueprint, AtpBlueprintable, AtpClassifier, AtpType, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

consistencyNeeds	ConsistencyNeeds	*	aggr	This represents the collection of ConsistencyNeeds owned by the enclosing SwComponentType. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
port	PortPrototype	*	aggr	The PortPrototypes through which this SwComponentType can communicate. The aggregation of PortPrototype is subject to variability with the purpose to support the conditional existence of PortPrototypes. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=preCompileTime
portGroup	PortGroup	*	aggr	A port group being part of this component. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=preCompileTime
swComponentDocumentation	SwComponentDocumentation	0..1	aggr	This adds a documentation to the SwComponentType. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=swComponentDocumentation, variationPoint.shortLabel vh.latestBindingTime=preCompileTime xml.sequenceOffset=-10
unitGroup	UnitGroup	*	ref	This allows for the specification of which UnitGroups are relevant in the context of referencing SwComponentType.

Table A.34: SwComponentType

Class	**System**			
Package	M2::AUTOSARTemplates::SystemTemplate			
Note	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. **Tags:** atp.recommendedPackage=Systems			
Base	ARElement, ARObject, AtpClassifier, AtpFeature, AtpStructureElement, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

clientIdDefinitionSet	ClientIdDefinitionSet	*	ref	Set of Client Identifiers that are used for inter-ECU client-server communication in the System.
containerIPduHeaderByteOrder	ByteOrderEnum	0..1	attr	Defines the byteOrder of the header in ContainerIPdus.
ecuExtractVersion	RevisionLabelString	0..1	attr	Version number of the Ecu Extract.
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 FibexElements can be optional. **Stereotypes:** atpVariation **Tags:** vh.latestBindingTime=postBuild
j1939SharedAddressCluster	J1939SharedAddressCluster	*	aggr	Collection of J1939Clusters that share a common address space for the routing of messages. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
mapping	SystemMapping	*	aggr	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 SystemMapping 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. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=postBuild
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	RootSwCompositionPrototype	0..1	aggr	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. atpVariation: The RootSwCompositionPrototype can vary. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=systemDesignTime
systemDocumentation	Chapter	*	aggr	Possibility to provide additional documentation while defining the System. The System documentation can be composed of several chapters. **Stereotypes:** atpSplitable; atpVariation **Tags:** atp.Splitkey=shortName, variationPoint.shortLabel vh.latestBindingTime=systemDesignTime xml.sequenceOffset=-10
systemVersion	RevisionLabelString	1	attr	Version number of the System Description.

Table A.35: System

Class	**SystemSignal**			
Package	M2::AUTOSARTemplates::SystemTemplate::Fibex::FibexCore::CoreCommunication			
Note	The system signal represents the communication system's view of data exchanged between SW components which reside on different ECUs. The system signals allow to represent this communication in a flattened structure, with exactly one system signal defined for each data element prototype sent and received by connected SW component instances. **Tags:** atp.recommendedPackage=SystemSignals			
Base	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
dynamicLength	Boolean	1	attr	The length of dynamic length signals is variable in run-time. Only a maximum length of such a signal is specified in the configuration (attribute length in ISignal element).
physicalProps	SwDataDefProps	0..1	aggr	Specification of the physical representation.

Table A.36: SystemSignal

Class	**Unit**			
Package	M2::MSR::AsamHdo::Units			

Note	This is a physical measurement unit. All units that might be defined should stem from SI units. In order to convert one unit into another factor and offset are defined.
	
	For the calculation from SI-unit to the defined unit the factor (factorSiToUnit) and the offset (offsetSiToUnit) are applied as follows:
	`x [{unit}] := y * [{siUnit}] * factorSiToUnit [[unit]/{siUnit}] + offsetSiToUnit [{unit}]`
	
	For the calculation from a unit to SI-unit the reciprocal of the factor (factorSiToUnit) and the negation of the offset (offsetSiToUnit) are applied.
	`y {siUnit} := (x*{unit} - offsetSiToUnit [{unit}]) / (factorSiToUnit [[unit]/{siUnit}]`
	
	Tags: atp.recommendedPackage=Units			

Base	ARElement, ARObject, CollectableElement, Identifiable, MultilanguageReferrable, PackageableElement, Referrable			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
displayName	SingleLanguageUnitNames	0..1	aggr	This specifies how the unit shall be displayed in documents or in user interfaces of tools.The displayName corresponds to the Unit.Display in an ASAM MCD-2MC file. **Tags:** xml.sequenceOffset=20
factorSiToUnit	Float	0..1	attr	This is the factor for the conversion from SI Units to units. The inverse is used for conversion from units to SI Units. **Tags:** xml.sequenceOffset=30
offsetSiToUnit	Float	0..1	attr	This is the offset for the conversion from and to siUnits. **Tags:** xml.sequenceOffset=40
physicalDimension	PhysicalDimension	0..1	ref	This association represents the physical dimension to which the unit belongs to. Note that only values with units of the same physical dimensions might be converted. **Tags:** xml.sequenceOffset=50

Table A.37: Unit

Class	**ValueSpecification (abstract)**			
Package	M2::AUTOSARTemplates::CommonStructure::Constants			
Note	Base class for expressions leading to a value which can be used to initialize a data object.			
Base	ARObject			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
shortLabel	Identifier	0..1	attr	This can be used to identify particular value specifications for human readers, for example elements of a record type.

Table A.38: ValueSpecification

B History of Constraints and Specification Items

B.1 Constraint History of this Document according to AUTOSAR R4.2.1

B.1.1 Added Specification Items in R4.2.1

Number	Heading
[TPS_DEXT_01000]	AUTOSAR diagnostics supports two kinds of data identifiers
[TPS_DEXT_01001]	Definition of a fixed-sized array
[TPS_DEXT_01002]	Definition of a variable-sized array
[TPS_DEXT_01003]	`DiagnosticContributionSet` is the central part of the `DiagnosticExtract`
[TPS_DEXT_01004]	`DiagnosticContributionSet` defines the scope of the `DiagnosticExtract`
[TPS_DEXT_01005]	`DiagnosticContributionSet` can exist independently
[TPS_DEXT_01006]	The role of `DiagnosticServiceTable`s in the context of a `DiagnosticContributionSet`
[TPS_DEXT_01007]	Common properties of a `DiagnosticExtract`
[TPS_DEXT_01008]	`DiagnosticContributionSet` defines the scope for the application of the common diagnostic properties
[TPS_DEXT_01009]	Limited support for the configuration of custom diagnostic services
[TPS_DEXT_01010]	Configuration of custom diagnostic services
[TPS_DEXT_01011]	Semantics of `DiagnosticSession.id`
[TPS_DEXT_01012]	Rationale for the modeling of the multiplicity of `DiagnosticAccessPermission.securityLevel`
[TPS_DEXT_01013]	Specification of sub-functions by means of attribute `DiagnosticServiceInstance.category`
[TPS_DEXT_01014]	Possible values of the `category` attribute for diagnostic services
[TPS_DEXT_01015]	Meaning of attributes of `DiagnosticIOControl`
[TPS_DEXT_01016]	The capability returnControlToEcu
[TPS_DEXT_01017]	Meaning of `DiagnosticIOControl.dataIdentifier`

Number	Heading
[TPS_DEXT_01018]	InputOutput Control does not define any sub-functions
[TPS_DEXT_01019]	Correspondence of category values to numerical values mentioned in the ISO 14229-1
[TPS_DEXT_01020]	Manufacturer-specific values for sub-functions of service EcuReset
[TPS_DEXT_01021]	Semantics of DiagnosticEcuReset.customSubFunctionNumber
[TPS_DEXT_01022]	ClearDiagnosticInformation does not define any sub-functions
[TPS_DEXT_01023]	WriteMemoryByAddress does not define any sub-functions
[TPS_DEXT_01024]	ReadMemoryByAddress does not define any sub-functions
[TPS_DEXT_01025]	TransferExit does not define any sub-functions
[TPS_DEXT_01026]	DataTransfer does not define any sub-functions
[TPS_DEXT_01027]	RequestDownload does not define any sub-functions
[TPS_DEXT_01028]	RequestUpload does not define any sub-functions
[TPS_DEXT_01029]	Correspondence of category values to numerical values mentioned in the ISO 14229-1
[TPS_DEXT_01030]	Manufacturer-specific values for sub-functions of service CommunicationControl
[TPS_DEXT_01031]	Semantics of DiagnosticComControl.customSubFunctionNumber
[TPS_DEXT_01032]	Impact of the DiagnosticComControlClass on the state management for CommunicationClusters
[TPS_DEXT_01033]	Semantics of triggers in the context of a DiagnosticResponseOnEvent
[TPS_DEXT_01034]	Sub-functions of the service ReadDTCInformation
[TPS_DEXT_01035]	Existence of DiagnosticRoutine.stop and DiagnosticRoutine.requestResult
[TPS_DEXT_01036]	Work-flow within the execution of the diagnostic service SecurityAccess
[TPS_DEXT_01037]	Semantics of DiagnosticSecurityAccess.requestSeedId
[TPS_DEXT_01038]	Motivation for making the reference DiagnosticSecurityAccess.securityLevel≪atpSplitable≫
[TPS_DEXT_01039]	Identification of the sub-function of DiagnosticSessionControl
[TPS_DEXT_01040]	Use case where the DiagnosticExtract refers to software-components
[TPS_DEXT_01041]	Semantics of attribute DiagnosticServiceDataMapping.diagnosticDataElement
[TPS_DEXT_01042]	Dem uses DiagnosticServiceDataMapping
[TPS_DEXT_01043]	Purpose of DiagnosticServiceSwMapping
[TPS_DEXT_01044]	BswServiceDependency needs to act as the target of a reference
[TPS_DEXT_01045]	Supported diagnostic services
[TPS_DEXT_01046]	ECU configuration is not suitable to be exchanged between partners in an ECU development project
[TPS_DEXT_01047]	Differences in the development processes for diagnostics at automotive OEMs and ECU suppliers
[TPS_DEXT_01048]	Actual algorithm for the diagnostic event debouncing
[TPS_DEXT_01049]	Consistency of DiagnosticServiceSwMapping with respect to routine IDs
[TPS_DEXT_01050]	Consistency of DiagnosticServiceSwMapping with respect to data IDs
[TPS_DEXT_01051]	Consistency of DiagnosticServiceSwMapping with respect to data IDs

Number	Heading
[TPS_DEXT_01052]	Existence of attribute DiagnosticServiceInstance.accessPermission
[TPS_DEXT_01053]	Existence of DiagnosticSecurityAccess.securityLevel
[TPS_DEXT_01054]	Existence of DiagnosticDataByIdentifier.dataIdentifier
[TPS_DEXT_01055]	Standardized values of DiagnosticContributionSet.category
[TPS_DEXT_01056]	Applicable values for DiagnosticEcuReset.category
[TPS_DEXT_01057]	Allowed values of DiagnosticComControl.category
[TPS_DEXT_01058]	Standardized values for DiagnosticDynamicallyDefineDataIdentifier.category
[TPS_DEXT_01059]	Applicable values for DiagnosticPeriodicRate.category
[TPS_DEXT_01060]	Applicable values for DiagnosticReadDTCInformation.category
[TPS_DEXT_01061]	Supported scenarios for the definition of *access permission*
[TPS_DEXT_01062]	Existence of DiagnosticServiceClass.accessPermissionValidity in an incomplete model
[TPS_DEXT_01063]	Existence of DiagnosticServiceClass.accessPermissionValidity in a complete model
[TPS_DEXT_01064]	Textually formulated content attached to DiagnosticTroubleCode
[TPS_DEXT_01065]	Different approaches to provide semi-formal textual content attached to a DiagnosticTroubleCode
[TPS_DEXT_01066]	Standardized values of DiagnosticTroubleCode.introduction.trace
[TPS_DEXT_01067]	Textually formulated content attached to DiagnosticEvent
[TPS_DEXT_01068]	Textual description with respect to the DiagnosticEvent
[TPS_DEXT_01069]	Standardized values of DiagnosticEvent.introduction.structuredReq
[TPS_DEXT_01070]	Description of textually semi-formal formulated pre- and post-conditions for the validity of DiagnosticAccessPermission
[TPS_DEXT_01071]	Standardized values of DiagnosticAccessPermission.introduction.structuredReq
[TPS_DEXT_01072]	Purpose of attribute DiagnosticDataIdentifier.representsVin
[TPS_DEXT_01073]	Diagnostic properties that are specific to an individual EcuInstance
[TPS_DEXT_01074]	Difference between the attributes DiagnosticComControl.specificChannel and DiagnosticComControl.subNodeChannel
[TPS_DEXT_01075]	standardized values for the attribute DiagnosticControlDTCSetting.category
[TPS_DEXT_01076]	Identification of sub-functions of diagnostic service ControlDTCSetting
[TPS_DEXT_01077]	Modeling of DiagnosticRoutine
[TPS_DEXT_01078]	Not possible to use the attribute category for the identification of the sub-function of diagnostic service RoutineControl
[TPS_DEXT_01079]	Modeling of the arguments to a DiagnosticRoutine
[TPS_DEXT_01080]	Diagnostic Routine needs to be started
[TPS_DEXT_01081]	Modeling of DiagnosticSessionControl
[TPS_DEXT_01082]	Existence of DiagnosticSessionControl.diagnosticSession
[TPS_DEXT_01083]	Semantics of a DiagnosticEvent
[TPS_DEXT_01084]	DiagnosticEvent can be connected to one or multiple indicators

Number	Heading
[TPS_DEXT_01085]	Semantics of DiagnosticConditionGroups
[TPS_DEXT_01086]	Reference to DiagnosticOperationCycle
[TPS_DEXT_01087]	Semantics of DiagnosticOperationCycle
[TPS_DEXT_01088]	Semantics of DiagnosticRoutine.id
[TPS_DEXT_01089]	Definition of an *identifier* of a DiagnosticIOControl
[TPS_DEXT_01090]	Diagnostic service RequestFileTransfer does not define any sub-functions
[TPS_DEXT_03000]	ISO 14229-1 reserves values of DiagnosticTroubleCodeGroup.group-Number
[TPS_DEXT_03001]	Different types of conditions
[TPS_DEXT_03002]	Two kind of mappings
[TPS_DEXT_03003]	Semantics of DiagnosticEventToTroubleCodeUdsMapping
[TPS_DEXT_03004]	DiagnosticEvent and DiagnosticDebounceAlgorithmProps
[TPS_DEXT_03005]	Existence of DiagnosticEventToDebounceAlgorithmMapping
[TPS_DEXT_03006]	Values of the individual DiagnosticStorageConditions
[TPS_DEXT_03007]	Semantics of DiagnosticEventPortMapping
[TPS_DEXT_03008]	Semantics of DiagnosticExtendedDataRecord
[TPS_DEXT_03009]	Semantics of DiagnosticFreezeFrame
[TPS_DEXT_03010]	Combination of DiagnosticConditions to DiagnosticCondition-Groups
[TPS_DEXT_03011]	Clearing request for a DiagnosticEvent
[TPS_DEXT_03012]	Three kinds of DTCs
[TPS_DEXT_03013]	Common properties of a DTC
[TPS_DEXT_03014]	Semantics of DiagnosticTroubleCodeGroup
[TPS_DEXT_03015]	EnableConditions have to be put into a DiagnosticEnableCondition-Group
[TPS_DEXT_03016]	StorageConditions have to be put into a DiagnosticStorageCondi-tionGroup
[TPS_DEXT_03017]	Semantics of DiagnosticOperationCyclePortMapping
[TPS_DEXT_03018]	Semantics of DiagnosticEnableConditionPortMapping
[TPS_DEXT_03019]	Semantics of DiagnosticStorageConditionPortMapping
[TPS_DEXT_03020]	Semantics of DiagnosticDemProvidedDataMapping
[TPS_DEXT_03021]	Aging
[TPS_DEXT_03022]	Different kinds of DiagnosticIndicators

Table B.1: Added Specification Items in 4.2.1

B.1.2 Added Constraints in R4.2.1

Number	Heading
[constr_1324]	Existence of attribute DiagnosticDataIdentifier.representsVin
[constr_1325]	Allowed attributes of SwDataDefProps for DiagnosticDataElement.sw-DataDefProps
[constr_1326]	Existence of a variable-sized array

Number	Heading
[constr_1327]	Multiplicity of DiagnosticContributionSet.ecuInstance
[constr_1328]	Consistency of DiagnosticContributionSet.ecuInstance and DiagnosticServiceTable.ecuInstance
[constr_1329]	Existence of concrete sub-classes of DiagnosticServiceClass in the context created by a DiagnosticContributionSet
[constr_1330]	Custom service identifier shall not overlap with standardized service identifiers
[constr_1331]	Existence of DiagnosticEcuReset.customSubFunctionNumber
[constr_1332]	Value range for DiagnosticEcuReset.customSubFunctionNumber
[constr_1333]	Existence of DiagnosticMemoryIdentifier.memoryLowAddress and DiagnosticMemoryIdentifier.memoryHighAddress
[constr_1334]	Existence of DiagnosticComControl.customSubFunctionNumber
[constr_1335]	Possible values for DiagnosticComControl.customSubFunctionNumber
[constr_1336]	Applicable value range for DiagnosticComControlSpecificChannel.subnetNumber
[constr_1337]	Allowed value range for attribute DiagnosticComControlSubNodeChannel.subNodeNumber
[constr_1338]	Maximum number of aggregated DiagnosticReadDataByPeriodicIDClass.periodicRate
[constr_1339]	Existence of DiagnosticRoutine.start
[constr_1340]	Consistency of DiagnosticServiceSwMapping with respect to synchronously called DiagnosticRoutines
[constr_1341]	Consistency of DiagnosticServiceSwMapping with respect to asynchronously called DiagnosticRoutines
[constr_1342]	Possible values for DiagnosticSecurityAccess.requestSeedId
[constr_1343]	Simultaneous existence of the attributes DiagnosticServiceDataMapping.diagnosticDataElement and DiagnosticDataByIdentifier.dataIdentifier
[constr_1344]	Condition for the identification of data types of attributes DiagnosticServiceDataMapping.mappedDataElement and DiagnosticServiceDataMapping.diagnosticDataElement
[constr_1345]	DiagnosticDataElement shall not (finally) be aggregated by a DiagnosticRoutine
[constr_1346]	Allowed values of DiagnosticServiceSwMapping.serviceInstance
[constr_1347]	Existence of attributes of DiagnosticServiceSwMapping
[constr_1349]	Value of udsDtcValue shall be unique
[constr_1350]	Value of DiagnosticTroubleCodeGroup.groupNumber shall be unique
[constr_1351]	Value of DiagnosticTroubleCodeGroup.groupNumber
[constr_1352]	Existence of maxNumberFreezeFrameRecords vs. freezeFrame
[constr_1353]	Applicability of [constr_1352]
[constr_1354]	Existence of attribute DiagnosticTroubleCodeProps.freezeFrameContent
[constr_1355]	Value of recordNumber
[constr_1356]	Value of recordNumber shall be unique
[constr_1357]	Value of recordNumber
[constr_1358]	Value of recordNumber shall be unique

Number	Heading
[constr_1359]	Existence of attribute DiagnosticDebounceAlgorithmProps.debounceCounterStorage
[constr_1360]	Usage of DiagEventDebounceMonitorInternal is not supported in the context of DiagnosticDebounceAlgorithmProps
[constr_1361]	Number of DiagnosticEventToEnableConditionGroupMapping elements per DiagnosticEvent
[constr_1362]	Number of DiagnosticEventToStorageConditionGroupMapping elements per DiagnosticEvent
[constr_1365]	Multiplicity of DiagnosticResponseOnEvent.event
[constr_1366]	Event ID in the context of diagnostic service ResponseOnEvent shall be unique
[constr_1376]	Multiplicity of reference DiagnosticTroubleCodeProps.memoryDestination
[constr_1377]	Existence of reference DiagnosticTroubleCodeProps.memoryDestination
[constr_1378]	Value of DiagnosticMemoryDestinationUserDefined.memoryId
[constr_1379]	Existence of DiagnosticMemoryDestinationPrimary
[constr_1380]	Existence of DiagnosticMemoryDestinationMirror

Table B.2: Added Constraints in R4.2.1

B.2 Constraint History of this Document according to AUTOSAR R4.2.2

B.2.1 Added Traceables in R4.2.2

none

B.2.2 Changed Traceables in R4.2.2

none

B.2.3 Deleted Traceables in R4.2.2

none

B.2.4 Added Constraints in R4.2.2

Number	Heading
[constr_1394]	Value of DiagnosticDataElement.maxNumberOfElements depending on its existence

Table B.3: Added Constraints in R4.2.2

B.2.5 Changed Constraints in R4.2.2

none

B.2.6 Deleted Constraints in R4.2.2

none

B.3 Constraint History of this Document according to AUTOSAR R4.3.0

B.3.1 Added Traceables in R4.3.0

Number	Heading
[TPS_DEXT_01091]	Relation between a DiagnosticServiceTable and one or more DiagnosticConnections
[TPS_DEXT_01092]	Semantics of DiagnosticParameterIdentifier
[TPS_DEXT_01093]	Definition of legislative freeze frame for WWWH-OBD
[TPS_DEXT_01094]	Semantics of meta-class DiagnosticTroubleCodeUdsToTroubleCodeObdMapping
[TPS_DEXT_01095]	Definition of “alias” diagnostic event for the creation of a Fim configuration in the diagnostic extract
[TPS_DEXT_01096]	Semantics of DiagnosticFunctionInhibitSource
[TPS_DEXT_01097]	Standardized value of StructuredReq.category for the modeling of DiagnosticFunctionInhibitSource
[TPS_DEXT_01098]	Semantics of attribute DiagnosticFunctionInhibitSource.event
[TPS_DEXT_01099]	Semantics of attribute DiagnosticFunctionInhibitSource.eventGroup
[TPS_DEXT_01100]	Consequence of the existence of DiagnosticFimAliasEventMapping
[TPS_DEXT_01101]	Consequence of the existence of DiagnosticFimAliasEventGroupMapping
[TPS_DEXT_01102]	Semantics of DiagnosticFimFunctionMapping
[TPS_DEXT_01103]	Semantics of meta-class DiagnosticJ1939SpnMapping
[TPS_DEXT_01104]	Difference between DiagnosticJ1939FreezeFrame and DiagnosticJ1939ExpandedFreezeFrame
[TPS_DEXT_01105]	Relation of DiagnosticJ1939Spn to DiagnosticJ1939FreezeFrame and DiagnosticJ1939ExpandedFreezeFrame
[TPS_DEXT_01106]	Relation of Controller Application to SPN
[TPS_DEXT_01107]	Definition of service-only DTC
[TPS_DEXT_01108]	Purpose of the DiagnosticJ1939SwMapping
[TPS_DEXT_01110]	Standardized values of DiagnosticIumprGroup.category
[TPS_DEXT_01111]	Legislative freeze frame for the OBD-II case

Number	Heading
[TPS_DEXT_01112]	Definition of a diagnostic trouble code for the implementation of WWH-OBD
[TPS_DEXT_01113]	Evaluation of a DiagnosticEnvConditionFormula
[TPS_DEXT_01114]	DiagnosticEnvConditionFormula that has no parts
[TPS_DEXT_01115]	DiagnosticEnvConditionFormula that has one part
[TPS_DEXT_01116]	DiagnosticEnvConditionFormula that has more than one part
[TPS_DEXT_01117]	Semantics of DiagnosticEnvConditionFormula.nrcValue
[TPS_DEXT_01118]	Semantics of DiagnosticEnvDataCondition
[TPS_DEXT_01119]	Semantics of DiagnosticEnvModeCondition
[TPS_DEXT_01120]	Comparison of the value of a ModeDeclarationGroupPrototype with a ModeDeclaration
[TPS_DEXT_01121]	Semantics of DiagnosticFunctionIdentifier
[TPS_DEXT_01122]	Indication whether a EcuInstance supports OBD
[TPS_DEXT_01124]	Semantics of meta-class DiagnosticProtocol
[TPS_DEXT_01125]	Support for diagnostic service RequestCurrentPowertrainDiagnostic-Data
[TPS_DEXT_01126]	Support of OBD service RequestPowertrainFreezeFrameData
[TPS_DEXT_01127]	Semantics of meta-class DiagnosticRequestEmissionRelatedDTC
[TPS_DEXT_01128]	Semantics of meta-class DiagnosticClearResetEmissionRelatedInfo
[TPS_DEXT_01129]	Support for OBD diagnostic service RequestOnBoardMonitoringTestRe-sults
[TPS_DEXT_01130]	Support of OBD diagnostic service RequestControlOfOnBoardDevice
[TPS_DEXT_01131]	Support for OBD diagnostic service RequestVehicleInformation
[TPS_DEXT_01132]	Support for OBD diagnostic service RequestEmissionRelatedDiagnos-ticTroubleCodesPermanentStatus
[TPS_DEXT_01133]	Support for WWH-OBD within the diagnostic extract
[TPS_DEXT_01134]	Definition of a DiagnosticDataElement used in the context of a DID ob-tained by sender-receiver communication
[TPS_DEXT_01135]	Definition of a DiagnosticDataElement used in the context of a DID ob-tained by client/server communication
[TPS_DEXT_01136]	Definition of a DiagnosticDataElement used in the context of a diagnostic routine
[TPS_DEXT_01137]	Applicability of DiagnosticDataIdentifier.didSize
[TPS_DEXT_01138]	Applicability of DiagnosticDataIdentifier.supportInfoByte
[TPS_DEXT_01139]	Semantics of the references from DiagnosticAccessPermission

Table B.4: Added Traceabless in 4.3.0

B.3.2 Changed Traceables in R4.3.0

Number	Heading
[TPS_DEXT_01006]	The role of DiagnosticServiceTables
[TPS_DEXT_01052]	Existence of attribute DiagnosticServiceInstance.accessPermission
[TPS_DEXT_01060]	Applicable values for DiagnosticReadDTCInformation.category

Number	Heading
[TPS_DEXT_01074]	Difference between the attributes DiagnosticComControlClass.specificChannel and DiagnosticComControlClass.subNodeChannel
[TPS_DEXT_03003]	Semantics of DiagnosticEventToTroubleCodeUdsMapping

Table B.5: Changed Traceables in R4.3.0

B.3.3 Deleted Traceables in R4.3.0

Number	Heading
[TPS_DEXT_01058]	Standardized values for DiagnosticDynamicallyDefineDataIdentifier.category
[TPS_DEXT_01059]	Applicable values for DiagnosticPeriodicRate.category

Table B.6: Deleted Traceables in R4.3.0

B.3.4 Added Constraints in R4.3.0

Number	Heading
[constr_1405]	Value of DiagnosticProtocol.serviceTable vs. DiagnosticServiceTable.protocolKind
[constr_1406]	DiagnosticServiceTable.diagnosticConnection vs. DiagnosticProtocol.diagnosticConnection
[constr_1411]	Existence of DiagnosticMemoryIdentifier.memoryHighAddressLabel vs. DiagnosticMemoryIdentifier.memoryHighAddress
[constr_1412]	Existence of DiagnosticMemoryIdentifier.memoryLowAddressLabel vs. DiagnosticMemoryIdentifier.memoryLowAddress
[constr_1419]	Value of DiagnosticSecurityLevel.accessDataRecordSize
[constr_1421]	Consistency of DiagnosticDynamicallyDefineDataIdentifierClass.subfunction
[constr_1435]	Debouncing in the presence of a DiagnosticEventPortMapping
[constr_1447]	Restrictions for the value of DiagnosticParameterIdentifier.id
[constr_1448]	Interval of DiagnosticParameterIdentifier.id
[constr_1449]	PID shall only carry a fixed-length collection of data
[constr_1450]	Service mapping for ODB mode 0x01 for DiagnosticParameterIdentifier
[constr_1451]	Service mapping for OBD mode 0x09 for DiagnosticInfoType
[constr_1452]	Service mapping for OBD mode 0x08 for DiagnosticInfoType
[constr_1453]	References from DiagnosticFunctionInhibitSource
[constr_1454]	DiagnosticFimFunctionMapping shall only reference a SwcServiceDependency that aggregates FunctionInhibitionNeeds
[constr_1455]	Relation of DiagnosticJ1939Node to J1939NmNode
[constr_1456]	Valid interval for attribute DiagnosticTroubleCodeJ1939.fmi
[constr_1457]	Service-only DTCs shall refer to a common memory section
[constr_1458]	Reference to DiagnosticMemoryDestination
[constr_1459]	Existence of attributes of DiagnosticTroubleCodeProps

Number	Heading
[constr_1460]	Restrictions for the value of DiagnosticInfoType.id
[constr_1461]	Restrictions for the value of DiagnosticTestRoutineIdentifier.id
[constr_1462]	Restrictions for the value of DiagnosticTestResult.testIdentifier.id
[constr_1464]	Allowed value range of DiagnosticEnvConditionFormula.nrcValue
[constr_1465]	Allowed values of compareType in the context of a DiagnosticEnvDataCondition
[constr_1466]	Allowed values of compareType in the context of a DiagnosticEnvModeCondition
[constr_1467]	References in DiagnosticEnvModeCondition
[constr_1470]	Value of DiagnosticParameter.bitOffset
[constr_1471]	Existence of DiagnosticDataIdentifier.didSize
[constr_1472]	Existence of DiagnosticDataIdentifier.supportInfoByte

Table B.7: Added Constraints in R4.3.0

B.3.5 Changed Constraints in R4.3.0

Number	Heading
[constr_1325]	Allowed attributes of SwDataDefProps for DiagnosticDataElement.swDataDefProps
[constr_1327]	Multiplicity of DiagnosticEcuInstanceProps.ecuInstance
[constr_1328]	Consistency of DiagnosticEcuInstanceProps.ecuInstance and DiagnosticServiceTable.ecuInstance

Table B.8: Changed Constraints in R4.3.0

B.3.6 Deleted Constraints in R4.3.0

none

B.4 Constraint History of this Document according to AUTOSAR R4.3.1

B.4.1 Added Traceables in 4.3.1

Number	Heading
[TPS_DEXT_01140]	Values contained in DiagnosticExtract shall be taken for the derivation of basic software modules

Table B.9: Added Traceables in 4.3.1

B.4.2 Changed Traceables in 4.3.1

Number	Heading
[TPS_DEXT_01087]	Semantics of DiagnosticOperationCycle

Table B.10: Changed Traceables in 4.3.1

B.4.3 Deleted Traceables in 4.3.1

Number	Heading
[TPS_DEXT_01073]	Diagnostic properties that are specific to an individual EcuInstance

Table B.11: Deleted Traceables in 4.3.1

B.4.4 Added Constraints in 4.3.1

Number	Heading
[constr_1509]	extendedDataRecord.recordNumber shall be unique within primary fault memory
[constr_1510]	extendedDataRecord.recordNumber shall be unique within mirror fault memory
[constr_1511]	extendedDataRecord.recordNumber shall be unique within user-defined fault memory
[constr_1512]	freezeFrame.recordNumber shall be unique within primary fault memory
[constr_1513]	freezeFrame.recordNumber shall be unique within mirror fault memory
[constr_1514]	freezeFrame.recordNumber shall be unique within user-defined fault memory
[constr_1515]	Reference from DiagnosticRoutineControl to DiagnosticAccessPermission has no meaning

Table B.12: Added Constraints in 4.3.1

B.4.5 Changed Constraints in 4.3.1

Number	Heading
[constr_1325]	Allowed attributes of SwDataDefProps for DiagnosticDataElement.swDataDefProps
[constr_1450]	Service mapping for ODB mode 0x01 for DiagnosticParameterIdentifier
[constr_1451]	Service mapping for OBD mode 0x09 for DiagnosticInfoType

Table B.13: Changed Constraints in 4.3.1

B.4.6 Deleted Constraints in 4.3.1

Number	Heading
[constr_1356]	Value of recordNumber shall be unique
[constr_1358]	Value of recordNumber shall be unique

Table B.14: Deleted Constraints in 4.3.1

C Glossary

Artifact This is a Work Product Definition that provides a description and definition for tangible work product types. Artifacts may be composed of other artifacts ([21]).

At a high level, an artifact is represented as a single conceptual file.

AUTOSAR Tool This is a software tool which supports one or more tasks defined as AUTOSAR tasks in the methodology. Depending on the supported tasks, an AUTOSAR tool can act as an authoring tool, a converter tool, a processor tool or as a combination of those (see separate definitions).

AUTOSAR Authoring Tool An AUTOSAR Tool used to create and modify AUTOSAR XML Descriptions. Example: System Description Editor.

AUTOSAR Converter Tool An AUTOSAR Tool used to create AUTOSAR XML files by converting information from other AUTOSAR XML files. Example: ECU Flattener

AUTOSAR Definition This is the definition of parameters which can have values. One could say that the parameter values are Instances of the definitions. But in the meta model hierarchy of AUTOSAR, definitions are also instances of the meta model and therefore considered as a description. Examples for AUTOSAR definitions are: `EcucParameterDef`, `PostBuildVariantCriterion`, `SwSystemconst`.

AUTOSAR XML Description In AUTOSAR this means "filled Template". In fact an AUTOSAR XML description is the XML representation of an AUTOSAR model.

The AUTOSAR XML description can consist of several files. Each individual file represents an AUTOSAR partial model and shall validate successfully against the AUTOSAR XML schema.

AUTOSAR Meta-Model This is an UML2.0 model that defines the language for describing AUTOSAR systems. The AUTOSAR meta-model is an UML representation of the AUTOSAR templates. UML2.0 class diagrams are used to describe the attributes and their interrelationships. Stereotypes, UML tags and OCL expressions (object constraint language) are used for defining specific semantics and constraints.

AUTOSAR Meta-Model Tool The AUTOSAR Meta-Model Tool is the tool that generates different views (class tables, list of constraints, diagrams, XML Schema etc.) on the AUTOSAR meta-model.

AUTOSAR Model This is a representation of an AUTOSAR product. The AUTOSAR model represents aspects suitable to the intended use according to the AUTOSAR methodology.

Strictly speaking, this is an instance of the AUTOSAR meta-model. The information contained in the AUTOSAR model can be anything that is representable according to the AUTOSAR meta-model.

AUTOSAR Partial Model In AUTOSAR, the possible partitioning of models is marked in the meta-model by ≪atpSplitable≫. One partial model is represented in an AUTOSAR XML description by one file. The partial model does not need to fulfill all semantic constraints applicable to an AUTOSAR model.

AUTOSAR Processor Tool An AUTOSAR Tool used to create non-AUTOSAR files by processing information from AUTOSAR XML files. Example: RTE Generator

AUTOSAR Specification Element An AUTOSAR Specification Element is a named element that is part of an AUTOSAR specification. Examples: requirement, constraint, specification item, class or attribute in the meta model, methodology, deliverable, methodology activity, model element, bsw module etc.

AUTOSAR Template The term "Template" is used in AUTOSAR to describe the format different kinds of descriptions. The term template comes from the idea, that AUTOSAR defines a kind of form which shall be filled out in order to describe a model. The filled form is then called the description.

In fact the AUTOSAR templates are now defined as a meta-model.

AUTOSAR Validation Tool A specialized AUTOSAR Tool which is able to check an AUTOSAR model against the rules defined by a profile.

AUTOSAR XML Schema This is a W3C XML schema that defines the language for exchanging AUTOSAR models. This Schema is derived from the AUTOSAR meta-model. The AUTOSAR XML Schema defines the AUTOSAR data exchange format.

Blueprint This is a model from which other models can be derived by copy and refinement. Note that in contrast to meta model resp. types, this process is *not* an instantiation.

Instance Generally this is a particular exemplar of a model or of a type.

Life Cycle Life Cycle is the course of development/evolutionary stages of a model element during its life time.

Meta-Model This defines the building blocks of a model. In that sense, a Meta-Model represents the language for building models.

Meta-Data This includes pertinent information about data, including information about the authorship, versioning, access-rights, timestamps etc.

Model A Model is an simplified representation of reality. The model represents the aspects suitable for an intended purpose.

Partial Model This is a part of a model which is intended to be persisted in one particular artifact.

Pattern in GST : This is an approach to simplify the definition of the meta model by applying a model transformation. This transformation creates an enhanced model out of an annotated model.

Profile Authoring Support Data Data that is used for efficient authoring of a profile. E.g. list of referable constraints, meta-classes, meta-attributes or other reusable model assets (blueprints)

Profile Authoring Tool A specialized AUTOSAR Tool which focuses on the authoring of profiles for data exchange points. It e.g. provides support for the creation of profiles from scratch, modification of existing profiles or composition of existing profiles.

Profile Compatibility Checker Tool A specialized AUTOSAR Tool which focuses on checking the compatibility of profiles for data exchange. Note that this compatibility check includes manual compatibility checks by engineers and automated assistance using more formal algorithms.

Profile Consistency Checker Tool A specialized AUTOSAR Tool which focuses on checking the consistency of profiles.

Property A property is a structural feature of an object. As an example a “connector” has the properties “receive port” and “send port”

Properties are made variant by the ≪atpVariation≫.

Prototype This is the implementation of a role of a type within the definition of another type. In other words a type may contain Prototypes that in turn are typed by "Types". Each one of these prototypes becomes an instance when this type is instantiated.

Type A type provides features that can appear in various roles of this type.

Value This is a particular value assigned to a “Definition”.

Variability Variability of a system is its quality to describe a set of variants. These variants are characterized by variant specific property settings and / or selections. As an example, such a system property selection manifests itself in a particular “receive port” for a connection.

This is implemented using the ≪atpVariation≫.

Variant A system variant is a concrete realization of a system, so that all its properties have been set respectively selected. The software system has no variability anymore with respect to the binding time.

This is implemented using EvaluatedVariantSet.

Variation Binding A variant is the result of a variation binding process that resolves the variability of the system by assigning particular values/selections to all the system’s properties.

This is implemented by VariationPoint.

Variation Binding Time The variation binding time determines the step in the methodology at which the variability given by a set of variable properties is resolved.

This is implemented by `vh.LatestBindingtime` at the related properties .

Variation Definition Time The variation definition time determines the step in the methodology at which the variation points are defined.

Variation Point A variation point indicates that a property is subject to variation. Furthermore, it is associated with a condition and a binding time which define the system context for the selection / setting of a concrete variant.

This is implemented by `VariationPoint`.

D Modeling of InstanceRef

D.1 Introduction

The existence of so-called `InstanceRef`s is a direct consequence to the usage of the `type-prototype` pattern for modeling within AUTOSAR. When referencing a `prototype` it is also necessary to include a reference to the `prototype`s typed by their corresponding `type`s that in turn aggregate further `prototype`s to set up the context.

In other words, `InstanceRef`s are representing **structured references** that, on the one hand, consist of references to context `prototype`s (indicated by a subsetting or redefinition of `atpContextElement`) and finally a reference to the applicable target `prototype` (indicated by a redefinition of `atpTarget`).

Note that it is not uncommon to have more than a single context in the modeling of particular `InstanceRef`s.

For the reader of specifications, the modeling of `InstanceRef`s manifests as a UML dependency stereotyped ≪`instanceRef`≫ drawn from one meta-class to another.

This is a simplified indication that the source of the dependency implements an `InstanceRef` to the meta-class at the target of the dependency. Again, in most cases this is everything a reader needs to understand in order to figure out the modeling.

The formal modeling of `InstanceRef`s is done by creating subclasses of the abstract meta-class `AtpInstanceRef`.

Wherever a more detailed understanding of the modeling is advised in the context of the specific chapter of this document, the modeling of a specific subclasses of `AtpInstanceRef` is explained directly in the context of the corresponding chapter.

In all other cases, a deeper understanding of the modeling of particular subclasses of `AtpInstanceRef`s can be obtained from reading this chapter.

Class tables included in this chapter are not fully filled out in the sense that most of the notes inside the class tables are missing.

The **primary** purpose of these class tables is to **provide information about the intended order** in which `InstanceRef`s are **serialized in M1 AUTOSAR models**.

In particular, the information about the order in serialized M1 models can be obtained from the value of the tag `xml.sequenceOffset` of each attribute of an `InstanceRef` meta-class.

For more information about the general concept of modeling `AtpInstanceRef` (e.g. the conceptual background of redefining or subsetting an association from a subclass of `AtpInstanceRef` to other meta-classes) please refer to [22].

D.2 Modeling

Class	**DataPrototypeInSystemInstanceRef**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::InstanceRefs			
Note				
Base	ARObject,AtpInstanceRef			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
base	System	1	ref	This represents the base of the InstanceRef **Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
contextComponent	SwComponentPrototype	*	ref	**Tags:** xml.sequenceOffset=30
contextDataPrototype (ordered)	ApplicationCompositeElementDataPrototype	*	ref	**Tags:** xml.sequenceOffset=50
contextPort	PortPrototype	1	ref	This represents the PortPrototype that is contained in the InstanceRef. **Tags:** xml.sequenceOffset=40
contextRootComposition	RootSwCompositionPrototype	0..1	ref	**Tags:** xml.sequenceOffset=20
targetDataPrototype	DataPrototype	1	ref	This represents the target of the InstanceRef **Tags:** xml.sequenceOffset=60

Table D.1: DataPrototypeInSystemInstanceRef

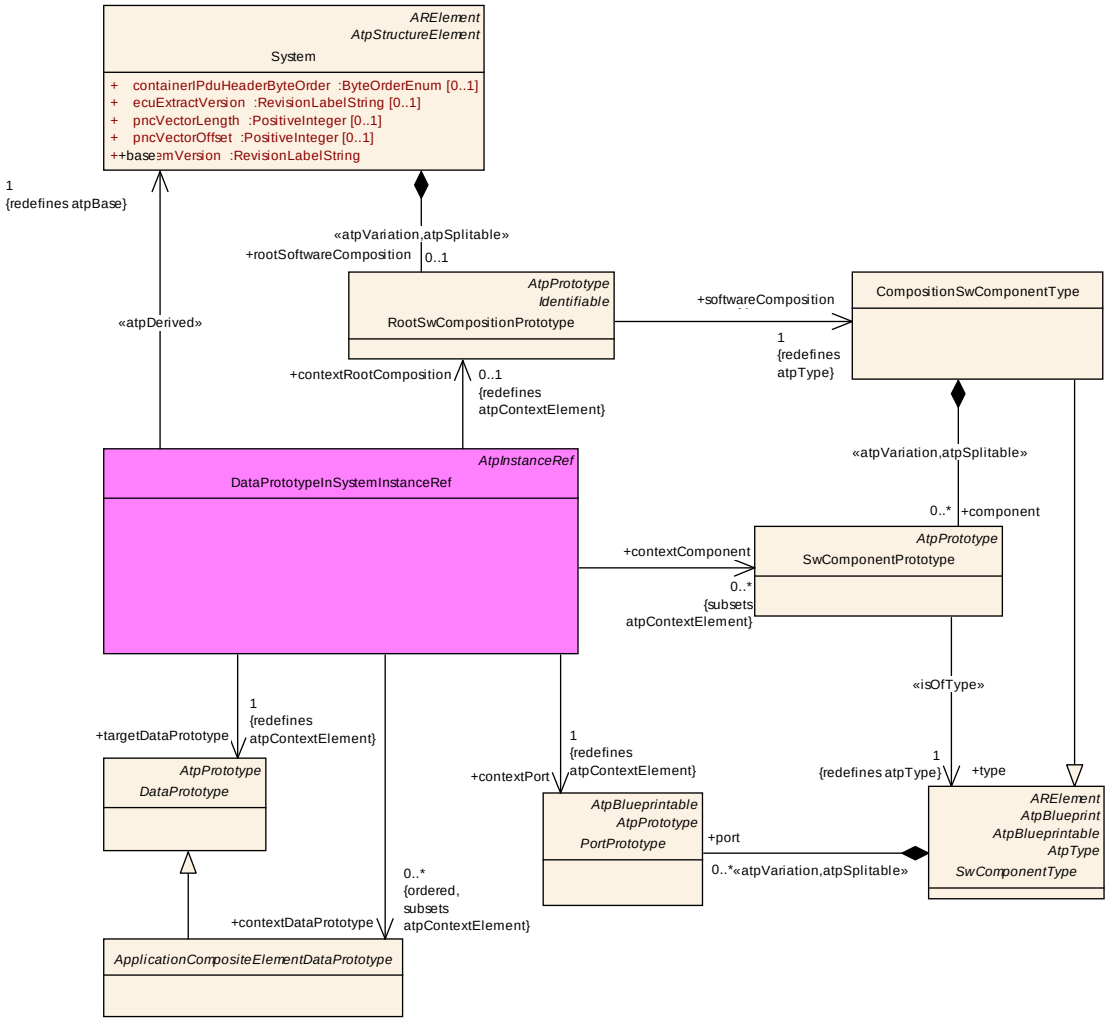

Figure D.1: Modeling of `DataPrototypeInSystemInstanceRef`

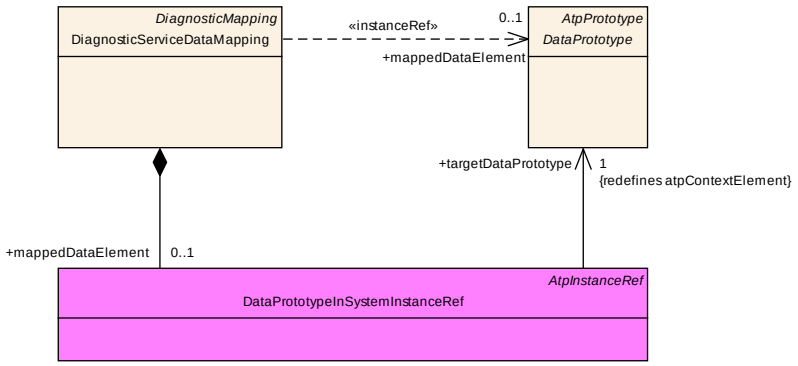

Figure D.2: Modeling of `DiagnosticServiceDataMapping`

Class	**SwcServiceDependencyInSystemInstanceRef**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::InstanceRefs			
Note				
Base	ARObject,AtpInstanceRef			
Attribute	***Type***	***Mul.***	***Kind***	***Note***

base	System	0..1	ref	**Tags:** atp.Status=obsolete
contextRootSwCompositionPrototype	RootSwCompositionPrototype	1	ref	
contextSwComponentPrototype	SwComponentPrototype	*	ref	
targetSwcServiceDependency	SwcServiceDependency	1	ref	

Table D.2: SwcServiceDependencyInSystemInstanceRef

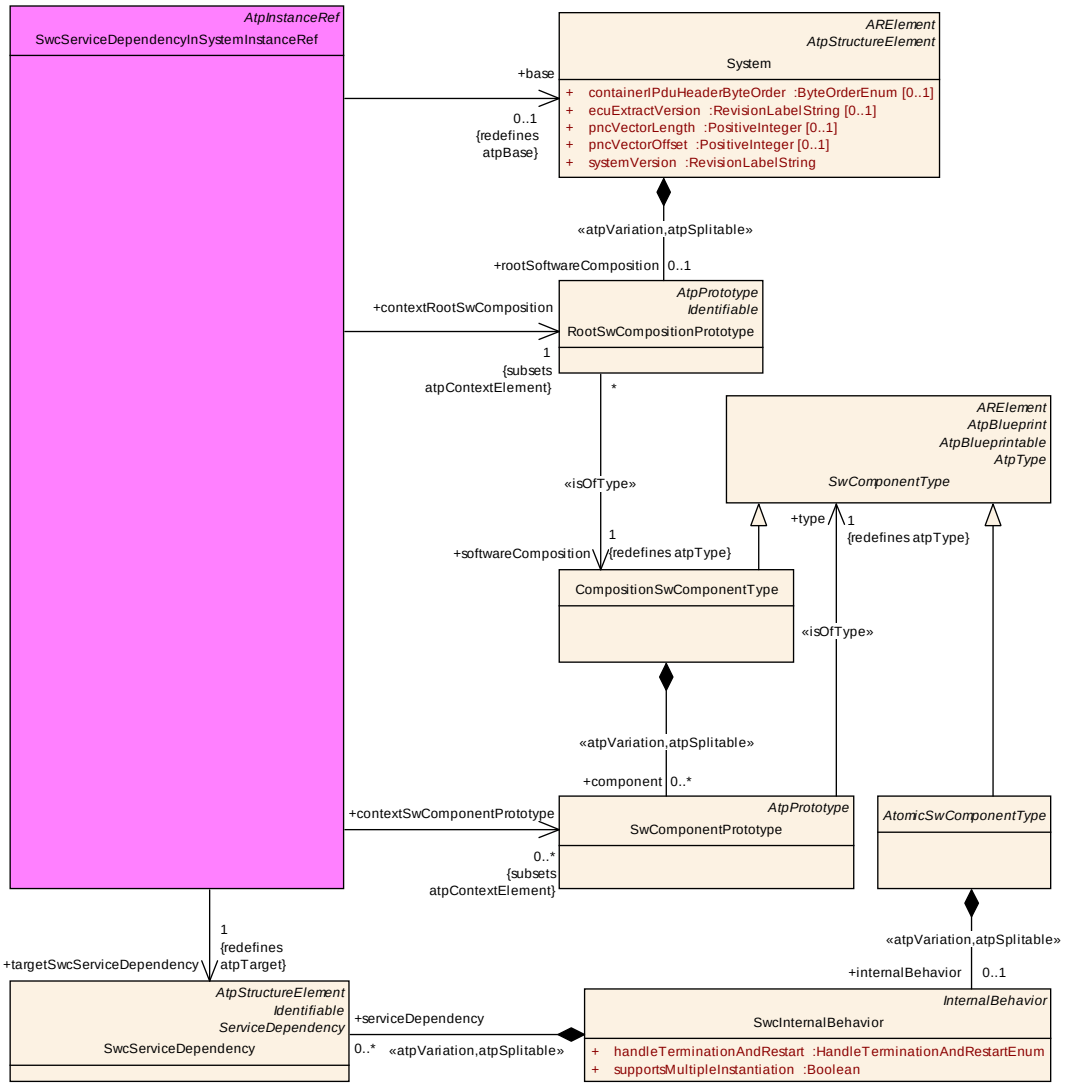

Figure D.3: Modeling of `SwcServiceDependencyInSystemInstanceRef`

Class	**PModeInSystemInstanceRef**			
Package	M2::AUTOSARTemplates::DiagnosticExtract::InstanceRefs			
Note				
Base	ARObject,AtpInstanceRef			
Attribute	***Type***	***Mul.***	***Kind***	***Note***
base	System	1	ref	**Stereotypes:** atpDerived **Tags:** xml.sequenceOffset=10
contextComponent	SwComponentPrototype	1..*	ref	**Tags:** xml.sequenceOffset=30
contextComposition	RootSwCompositionPrototype	1	ref	**Tags:** xml.sequenceOffset=20
contextModeDeclarationGroup	ModeDeclarationGroupPrototype	1	ref	**Tags:** xml.sequenceOffset=50
contextPPort	AbstractProvidedPortPrototype	1	ref	**Tags:** xml.sequenceOffset=40
targetMode	ModeDeclaration	1	ref	**Tags:** xml.sequenceOffset=60

Table D.3: PModeInSystemInstanceRef

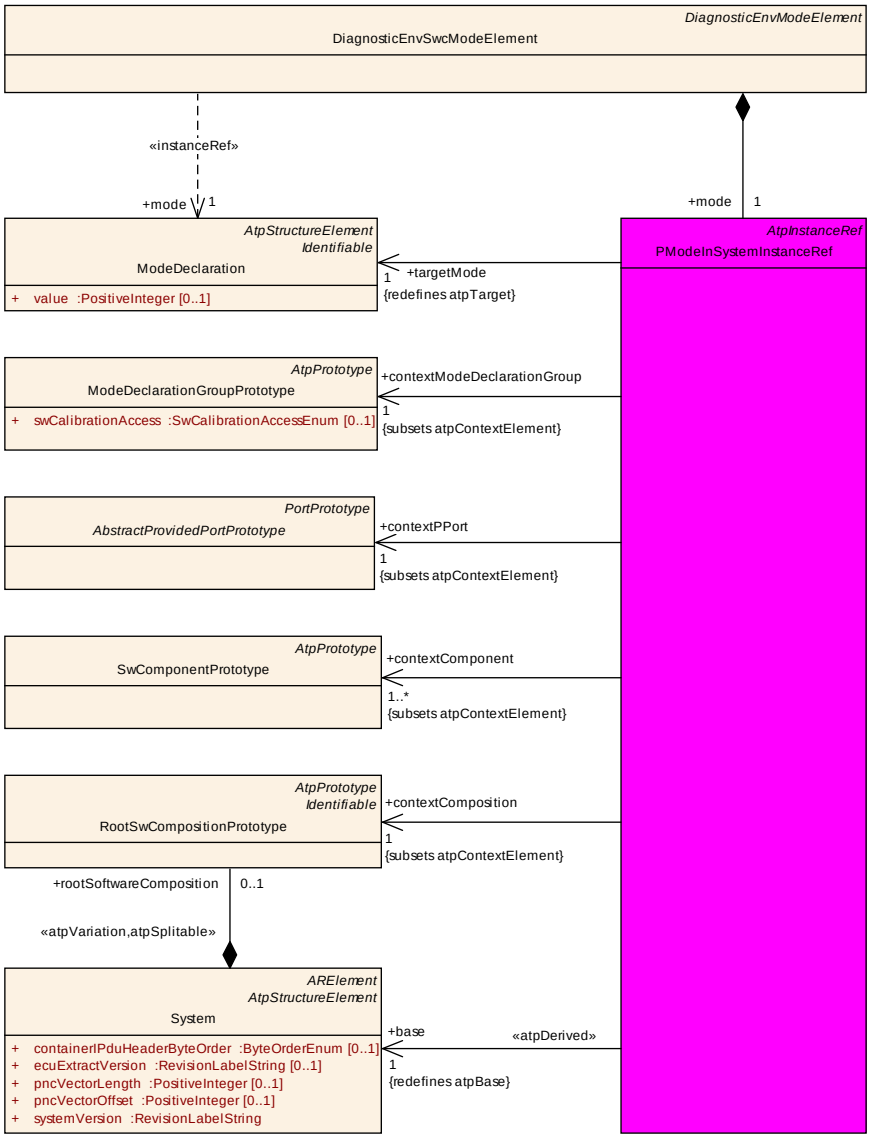

Figure D.4: Formal modeling of the comparison of a `ModeDeclarationGroupPrototype` with a `ModeDeclaration`

E Upstream Mapping

E.1 Introduction

This chapter describes the mapping of the ECU Configuration parameters (M1 model) onto the meta-classes and attributes of the AUTOSAR upstream templates (System Template, SW Component Template and ECU Resource Template).

The relationships between upstream templates and ECU Configuration are described in order to answer typical questions like:

- How shall a supplier use the information in a System Description in order to fulfill the needs defined by the systems engineer?
- How is a tool vendor supposed to generate an ECU Configuration Description out of ECU Extract of System Description?

Please note that the tables contain the following columns:

bsw module: Name of BSW module

bsw context: Reference to parameter container

bsw type: Type of parameter

bsw param: Name of the BSW parameter

bsw desc: Description from the configuration document

m2 template: System Template, SW Component Template, ECU Resource Template

m2 param: Name of the upstream template parameter

m2 description: Description from the upstream template definition

mapping rule: Textual description on how to transform between M2 and BSW domains

mapping type:

- local: no mapping needed since parameter local to BSW
- partial: some data can be automatically mapped but not all
- full: all data can be automatically mapped

E.2 Dcm

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsd	
BSW Parameter		**BSW Type**
DcmDsdServiceTable		EcucParamConfContainerDef

BSW Description
This container contains the configuration (DSD parameters) for a Service Identifier Table. Note: It is allowed to add OBD services to a DcmDsdServiceTable related to a UDS Protocol. But it is not allowed to add UDS services to a DcmDsdServiceTable related to an OBD Protocol.
Template Description
This meta-class represents a model of a diagnostic service table, i.e. the UDS services applicable for a given ECU.
M2 Parameter
DiagnosticExtract::DiagnosticContribution::DiagnosticServiceTable

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00040

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService

BSW Parameter	BSW Type
DcmDsdSidTabSecurityLevelRef	EcucReferenceDef

BSW Description
Reference to a Security Level in which the service is allowed to be executed. Multiple references are allowed for a service. Please refer to ISO 14229-1, ISO 15031-5 and chapter "Verification of the Service Security Access levels." If there is no reference configured, no service security verification shall be performed.
Template Description
This represents the associated DiagnosticSecurityLevels
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticAccessPermission.securityLevel

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00041

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService

BSW Parameter	BSW Type
DcmDsdSidTabServiceId	EcucIntegerParamDef

BSW Description
Identifier of the service. The possible service identifiers are defined in ISO 14229-1 and ISO 15031-5.
Template Description
This meta-class provides the ability to define common properties that are shared among all instances of sub-classes of DiagnosticServiceInstance.
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::CommonService::DiagnosticServiceClass

Mapping Rule	Mapping Type
Service identifiers of the used DiagnosticServiceClass	full
Mapping Status	**Mapping ID**

valid	up_Dcm_00042

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService
BSW Parameter	**BSW Type**
DcmDsdSidTabSessionLevelRef	EcucReferenceDef
BSW Description	
Reference to a Session Level in which the service is allowed to be executed. Multiple references are allowed for a service. Please refer to ISO 14229-1, ISO 15031-5 and chapter "Verification of the Diagnostic Session". If there is no reference configured, no diagnostic session verification shall be performed.	
Template Description	
This represents the associated DiagnosticSessions	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00043

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService
BSW Parameter	**BSW Type**
DcmDsdSidTabSubfuncAvail	EcucBooleanParamDef
BSW Description	
Information about whether the service has subfunctions or not. This parameter is used for the handling of the "suppressPosRspMsgIndicationBit" as defined in ISO 14229-1, which can be used as a reference for the configuration. true - service has subfunctions, suppressPosRspMsgIndicationBit is available false - service has no subfunctions, suppressPosRspMsgIndicationBit is not available	
Template Description	
The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.	
M2 Parameter	
GenericStructure::GeneralTemplateClasses::Identifiable::Identifiable.category	
Mapping Rule	**Mapping Type**
Stanadardized sub-functions of diagnostic services are mainly identified by the category. There are further specific attributes in the meta-model that allow for handling custom subfunctions,	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00044

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService/DcmDsdSubService
BSW Parameter	**BSW Type**

DcmDsdSubServiceFnc	EcucFunctionNameDef
BSW Description	
Callback function of the ECU Supplier specific component for the particular service. The function's prototype is as described for <Module>_<DiagnosticService>_<SubService>. If this parameter is not configured, the subservice is handled Dcm-internally.	
Template Description	
This attribute shall be used to define a custom sub-function number if none of the standardized values of category shall be used.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl::DiagnosticComControl.customSubFunctionNumber	
Mapping Rule	**Mapping Type**
The existence of a custom subfunction number shall trigger the creation of a custom processor.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00286

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService/DcmDsdSubService
BSW Parameter	**BSW Type**
DcmDsdSubServiceId	EcucIntegerParamDef
BSW Description	
Identifier of the subservice. The possible subservice identifiers are defined in ISO 14229-1 and ISO 15031-5.	
Template Description	
The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.	
M2 Parameter	
GenericStructure::GeneralTemplateClasses::Identifiable::Identifiable.category	
Mapping Rule	**Mapping Type**
Numerical values of diagnostic service according to ISO 14229 correspond to values of DiagnosticServiceInstance.category.	partial
Mapping Status	**Mapping ID**
valid	up_Dcm_00045

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService/DcmDsdSubService
BSW Parameter	**BSW Type**
DcmDsdSubServiceSecurityLevelRef	EcucReferenceDef
BSW Description	
Reference to a Security Level in which the subservice is allowed to be executed. Multiple references are allowed for a subservice. Please refer to ISO 14229-1, ISO 15031-5 and chapter "Verification of the Service Security Access levels." If there is no reference configured, no subservice security verification shall be performed.	
Template Description	

This represents the associated DiagnosticSecurityLevels	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticAccessPermission.securityLevel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00046

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsd/DcmDsdServiceTable/DcmDsdService/DcmDsdSubService
BSW Parameter	**BSW Type**
DcmDsdSubServiceSessionLevelRef	EcucReferenceDef
BSW Description	
Reference to a Session Level in which the subservice is allowed to be executed. Multiple references are allowed for a subservice. Please refer to ISO 14229-1, ISO 15031-5 and chapter "Verification of the Diagnostic Session". If there is no reference configured, no diagnostic session verification shall be performed.	
Template Description	
This represents the associated DiagnosticSessions	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00047

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslDiagResp
BSW Parameter	**BSW Type**
DcmDslDiagRespMaxNumRespPend	EcucIntegerParamDef
BSW Description	
Maximum number of negative responses with response code 0x78 (requestCorrectlyReceivedResponsePending) allowed for a request. If Dcm reaches this limit, an automatic 0x10 (generalReject) final response will be trasmitted and the service processing will be cancelled.	
Template Description	
Maximum number of negative responses with response code 0x78 (requestCorrectlyReceived-ResponsePending) allowed per request. DCM will send a negative response with response code 0x10 (generalReject), in case the limit value gets reached. Value 0xFF means that no limit number of NRC 0x78 response apply.	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.maxNumberOfRequestCorrectlyReceivedResponsePending	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00048

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslDiagResp
BSW Parameter	**BSW Type**
DcmDslDiagRespOnSecondDeclinedRequest	EcucBooleanParamDef
BSW Description	
Defines the reaction upon a second request (ClientB) that can not be processed (e.g. due to priority assessment). TRUE: when the second request (Client B) can not be processed, it shall be answered with NRC21 BusyRepeatRequest. FALSE: when the second request (Client B) can not be processed, it shall not be responded.	
Template Description	
Defines the reaction upon a second request (ClientB) that can not be processed (e.g. due to priority assessment). TRUE: when the second request (Client B) can not be processed, it shall be answered with NRC21 BusyRepeatRequest. FALSE: when the second request (Client B) can not be processed, it shall not be responded.	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.responseOnSecondDeclinedRequest	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00049

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol
BSW Parameter	**BSW Type**
DcmDslProtocolRow	EcucParamConfContainerDef
BSW Description	
This container contains the configuration of one particular diagnostic protocol used in Dcm.	
Template Description	
This meta-class represents the ability to define a diagnostic protocol.	
M2 Parameter	
DiagnosticExtract::DiagnosticContribution::DiagnosticProtocol	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00050

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection
BSW Parameter	**BSW Type**
DcmDslProtocolComMChannelRef	EcucSymbolicNameReferenceDef
BSW Description	
Reference to the ComMChannel on which the DcmDslProtocolRxPdu is received and the DcmDslProtocolTxPdu is transmitted.	
Template Description	

The CommunicationCluster is the main element to describe the topological connection of communicating ECUs. A cluster describes the ensemble of ECUs, which are linked by a communication medium of arbitrary topology (bus, star, ring, ...). The nodes within the cluster share the same communication protocol, which may be event-triggered, time-triggered or a combination of both. A CommunicationCluster aggregates one or more physical channels.

M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::CommunicationCluster	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00051

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection/DcmDslProtocolRx/DcmDslProtocolRxAddrType

BSW Parameter	**BSW Type**
DCM_FUNCTIONAL_TYPE	EcucEnumerationLiteralDef
BSW Description	
FUNCTIONAL = 1 to n communication	
Template Description	
Reference to functional request messages.	
M2 Parameter	
SystemTemplate::DiagnosticConnection::DiagnosticConnection.functionalRequest	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00052

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection/DcmDslProtocolRx/DcmDslProtocolRxAddrType

BSW Parameter	**BSW Type**
DCM_PHYSICAL_TYPE	EcucEnumerationLiteralDef
BSW Description	
PHYSICAL = 1 to 1 communications using physical addressing	
Template Description	
Reference to a physical request message.	
M2 Parameter	
SystemTemplate::DiagnosticConnection::DiagnosticConnection.physicalRequest	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00053

BSW Module	**BSW Context**

Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection/DcmDslProtocolRx

BSW Parameter	**BSW Type**
DcmDslProtocolRxPduRef	EcucReferenceDef
BSW Description	
Reference to a Pdu in EcuC that is used for this reception channel.	
Template Description	
The IPdu (Interaction Layer Protocol Data Unit) element is used to sum up all Pdus that are routed by the PduR.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::IPdu	
Mapping Rule	**Mapping Type**
Reference to IPdu of xxxTpConnection for DiagnosticConnection.physicalRequest / DiagnosticConnection.functionalRequest	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00054

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection

BSW Parameter	**BSW Type**
DcmDslProtocolRxConnectionId	EcucIntegerParamDef
BSW Description	
Unique identifier of the tester which uses this connection for diagnostic communication.	
Template Description	
An ECU specific ID for responses of diagnostic routines.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::EcuInstance.diagnosticAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00055

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection

BSW Parameter	**BSW Type**
DcmDslProtocolRxTesterSourceAddr	EcucIntegerParamDef
BSW Description	
Tester source address uniquely describes a client and will be used e.g within the jump to Bootloader interfaces. This parameter is not required for generic connections (DcmPdus with MetaDataLength >= 1).	
Template Description	
An ECU specific ID for responses of diagnostic routines.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreTopology::EcuInstance.diagnosticAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection

BSW Parameter	BSW Type
DcmDslProtocolTx	EcucParamConfContainerDef

BSW Description

This container contains the configuration parameters of a transmission channel in a diagnostic connection.

The PDU referenced by this transmission channel can produce meta data items of type TARGET_ADDRESS_16 and SOURCE_ADDRESS_16.

Template Description

In the vast majority of cases a response is required. However, there are also cases where providing the response is not possible and/or not allowed.

M2 Parameter

SystemTemplate::DiagnosticConnection::DiagnosticConnection.response

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00056

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslMainConnection/DcmDslProtocolTx

BSW Parameter	BSW Type
DcmDslProtocolTxPduRef	EcucReferenceDef

BSW Description

Reference to a Pdu in EcuC that is used for this transmission channel.

Template Description

The IPdu (Interaction Layer Protocol Data Unit) element is used to sum up all Pdus that are routed by the PduR.

M2 Parameter

SystemTemplate::Fibex::FibexCore::CoreCommunication::IPdu

Mapping Rule	Mapping Type
Reference to IPdu of xxxTpConnection for DiagnosticConnection.response	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00057

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslPeriodicTransmission/DcmDslPeriodicConnection

BSW Parameter	BSW Type
DcmDslPeriodicTxPduRef	EcucReferenceDef

BSW Description

Reference to a Pdu in EcuC that is used for this periodic transmission channel.

Template Description

The IPdu (Interaction Layer Protocol Data Unit) element is used to sum up all Pdus that are routed by the PduR.

M2 Parameter

SystemTemplate::Fibex::FibexCore::CoreCommunication::IPdu

Mapping Rule	Mapping Type

Reference to IPdu of xxxTpConnection in case of DiagnosticConnection.periodicResponseTp or IPdu of PduTriggering in case of DiagnosticConnection.periodicResponseUudt	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00058

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslConnection/DcmDslResponseOnEvent
BSW Parameter	**BSW Type**
DcmDslRoeTxPduRef	EcucReferenceDef
BSW Description	
Reference to a Pdu in EcuC that is used for this ResponseOnEvent transmission connection.	
Template Description	
The IPdu (Interaction Layer Protocol Data Unit) element is used to sum up all Pdus that are routed by the PduR.	
M2 Parameter	
SystemTemplate::Fibex::FibexCore::CoreCommunication::IPdu	
Mapping Rule	**Mapping Type**
Reference to IPdu of xxxTpConnection for DiagnosticConnection.responseOnEvent	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00059

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolID
BSW Parameter	**BSW Type**
DCM_PERIODICTRANS_ON_CAN	EcucEnumerationLiteralDef
BSW Description	
Template Description	
AbstractCanPhysicalChannel: Abstract class that is used to collect the common TtCAN and CAN PhysicalChannel attributes. **DiagnosticConnection.periodicResponseUudt**: Reference to UUDT responses.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::AbstractCanPhysicalChannel, SystemTemplate::DiagnosticConnection::DiagnosticConnection.periodicResponseUudt,	
Mapping Rule	**Mapping Type**
If DiagnosticConnection.periodicResponseUudt exists and PhysicalChannel given as AbstractCanPhysicalChannel.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00060

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolID
BSW Parameter	**BSW Type**
DCM_PERIODICTRANS_ON_IP	EcucEnumerationLiteralDef

BSW Description
Template Description
SocketConnection: The SoAd serves as a (De)Multiplexer between different PDU sources and the TCP/IP stack. **DiagnosticConnection.periodicResponseUudt**: Reference to UUDT responses. **EthernetPhysicalChannel**: The EthernetPhysicalChannel represents a VLAN or an untagged channel. An untagged channel is modeled as an EthernetPhysicalChannel without an aggregated VLAN.
M2 Parameter
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection, SystemTemplate::DiagnosticConnection::DiagnosticConnection.periodicResponseUudt, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetPhysicalChannel

Mapping Rule	Mapping Type
If DiagnosticConnection.periodicResponseUudt exists and PhysicalChannel given as EthernetPhysicalChannel.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00061

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDsl ProtocolID

BSW Parameter	BSW Type
DCM_ROE_ON_CAN	EcucEnumerationLiteralDef

BSW Description
Template Description
DiagnosticConnection.responseOnEvent: Reference to a ROE message. **CanTpConnection**: A connection identifies the sender and the receiver of this particular communication. The CanTp module routes a Pdu through this connection. atpVariation: Derived, because TpNode can vary.
M2 Parameter
SystemTemplate::DiagnosticConnection::DiagnosticConnection.responseOnEvent, SystemTemplate::TransportProtocols::CanTpConnection

Mapping Rule	Mapping Type
In case DiagnosticConnection.responseOnEvent exists and TpConnection Ident.ident belongs to a CanTpConnection.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00062

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDsl ProtocolID

BSW Parameter	BSW Type
DCM_ROE_ON_FLEXRAY	EcucEnumerationLiteralDef

BSW Description
Template Description
DiagnosticConnection.responseOnEvent: Reference to a ROE message. **FlexrayTpConnection**: A connection identifies the sender and the receiver of this particular communication. The FlexRayTp module routes a Pdu through this connection. In a System Description the references to the PduPools are mandatory. In an ECU Extract these references can be optional: On unicast connections these references are always mandatory. On multicast the txPduPool is mandatory on the sender side. The rxPduPool is mandatory on the receiver side. On Gateway ECUs both references are mandatory.
M2 Parameter
SystemTemplate::DiagnosticConnection::DiagnosticConnection.responseOnEvent, SystemTemplate::TransportProtocols::FlexrayTpConnection

Mapping Rule	Mapping Type
In case DiagnosticConnection.responseOnEvent exists and TpConnection Ident.ident belongs to FlexRayTpConnection	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00063

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolID

BSW Parameter	BSW Type
DCM_ROE_ON_IP	EcucEnumerationLiteralDef
BSW Description	
Template Description	
DiagnosticConnection.responseOnEvent: Reference to a ROE message. **SocketConnection**: The SoAd serves as a (De)Multiplexer between different PDU sources and the TCP/IP stack.	
M2 Parameter	
SystemTemplate::DiagnosticConnection::DiagnosticConnection.responseOnEvent, SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection	

Mapping Rule	Mapping Type
In case DiagnosticConnection.responseOnEvent exists and TpConnection Ident.ident belongs to SocketConnection	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00064

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolID

BSW Parameter	BSW Type
DCM_UDS_ON_CAN	EcucEnumerationLiteralDef
BSW Description	

UDS on CAN (ISO15765-3; ISO14229-1)	
Template Description	
DiagnosticConnection.physicalRequest: Reference to a physical request message. **CanTpConnection**: A connection identifies the sender and the receiver of this particular communication. The CanTp module routes a Pdu through this connection. atpVariation: Derived, because TpNode can vary.	
M2 Parameter	
SystemTemplate::DiagnosticConnection::DiagnosticConnection.physicalRequest, SystemTemplate::TransportProtocols::CanTpConnection	
Mapping Rule	**Mapping Type**
In case DiagnosticConnection.physicalRequest exists and TpConnection Ident.ident belongs to CanTpConnection	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00065

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDsl ProtocolID
BSW Parameter	**BSW Type**
DCM_UDS_ON_FLEXRAY	EcucEnumerationLiteralDef
BSW Description	
UDS on FlexRay (Manufacturer specific; ISO14229-1)	
Template Description	
DiagnosticConnection.physicalRequest: Reference to a physical request message. **FlexrayTpConnection**: A connection identifies the sender and the receiver of this particular communication. The FlexRayTp module routes a Pdu through this connection. In a System Description the references to the PduPools are mandatory. In an ECU Extract these references can be optional: On unicast connections these references are always mandatory. On multicast the txPduPool is mandatory on the sender side. The rxPduPool is mandatory on the receiver side. On Gateway ECUs both references are mandatory.	
M2 Parameter	
SystemTemplate::DiagnosticConnection::DiagnosticConnection.physicalRequest, SystemTemplate::TransportProtocols::FlexrayTpConnection	
Mapping Rule	**Mapping Type**
In case DiagnosticConnection.physicalRequest exists and TpConnection Ident.ident belongs to FlexRayTpConnection	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00066

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDsl ProtocolID
BSW Parameter	**BSW Type**

DCM_UDS_ON_IP	EcucEnumerationLiteralDef
BSW Description	
Template Description	
DiagnosticConnection.physicalRequest: Reference to a physical request message. **SocketConnection**: The SoAd serves as a (De)Multiplexer between different PDU sources and the TCP/IP stack.	
M2 Parameter	
SystemTemplate::DiagnosticConnection::DiagnosticConnection.physicalRequest, SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection	
Mapping Rule	**Mapping Type**
In case DiagnosticConnection.physicalRequest exists and TpConnection Ident.ident belongs to a SocketConnection	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00067

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow
BSW Parameter	**BSW Type**
DcmDslProtocolPriority	EcucIntegerParamDef
BSW Description	
Protocol priority used during protocol preemption. A higher priority protocol may preempt a lower priority protocol. Lower numeric values represent higher protocol priority: 0 - Highest protocol priority 255 - Lowest protocol priority	
Template Description	
This represents the priority of the diagnostic protocol in comparison to other diagnostic protocols. Lower numeric values represent higher protocol priority: - 0 - Highest protocol priority - 255 - Lowest protocol priority	
M2 Parameter	
DiagnosticExtract::DiagnosticContribution::DiagnosticProtocol.priority	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00068

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDsl ProtocolTransType
BSW Parameter	**BSW Type**
TYPE1	EcucEnumerationLiteralDef
BSW Description	
Messages on the DcmTxPduId already used for normal diagnostic responses. The outgoing messages must be synchronized with 'normal outgoing messages', which have a higher priority.	
Template Description	

Reference to a ROE message.	
M2 Parameter	
SystemTemplate::DiagnosticConnection::DiagnosticConnection.responseOnEvent	
Mapping Rule	**Mapping Type**
TYPE1 : periodicResponseTp / responseOnEvent using same reference as the normal response	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00069

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolTransType
BSW Parameter	**BSW Type**
TYPE2	EcucEnumerationLiteralDef
BSW Description	
Messages on a separate DcmTxPduId.	
Template Description	
Reference to a ROE message.	
M2 Parameter	
SystemTemplate::DiagnosticConnection::DiagnosticConnection.responseOnEvent	
Mapping Rule	**Mapping Type**
TYPE2: periodicResponseTp / responseOnEvent using other reference as the normal response	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00070

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolType
BSW Parameter	**BSW Type**
DCM_PERIODICTRANS_ON_CAN	EcucEnumerationLiteralDef
BSW Description	
Template Description	
AbstractCanPhysicalChannel: Abstract class that is used to collect the common TtCAN and CAN PhysicalChannel attributes. **DiagnosticConnection.periodicResponseUudt**: Reference to UUDT responses.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::AbstractCanPhysicalChannel, SystemTemplate::DiagnosticConnection::DiagnosticConnection.periodicResponseUudt,	
Mapping Rule	**Mapping Type**
If DiagnosticConnection.periodicResponseUudt exists and PhysicalChannel given as AbstractCanPhysicalChannel.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00060

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolType

BSW Parameter	BSW Type
DCM_PERIODICTRANS_ON_IP	EcucEnumerationLiteralDef
BSW Description	
Template Description	
SocketConnection: The SoAd serves as a (De)Multiplexer between different PDU sources and the TCP/IP stack. **DiagnosticConnection.periodicResponseUudt**: Reference to UUDT responses. **EthernetPhysicalChannel**: The EthernetPhysicalChannel represents a VLAN or an untagged channel. An untagged channel is modeled as an EthernetPhysicalChannel without an aggregated VLAN.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection, SystemTemplate::DiagnosticConnection::DiagnosticConnection.periodicResponseUudt, SystemTemplate::Fibex::Fibex4Ethernet::EthernetTopology::EthernetPhysicalChannel	
Mapping Rule	**Mapping Type**
If DiagnosticConnection.periodicResponseUudt exists and PhysicalChannel given as EthernetPhysicalChannel.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00061

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDsl ProtocolType

BSW Parameter	BSW Type
DCM_ROE_ON_CAN	EcucEnumerationLiteralDef
BSW Description	
Template Description	
DiagnosticConnection.responseOnEvent: Reference to a ROE message. **CanTpConnection**: A connection identifies the sender and the receiver of this particular communication. The CanTp module routes a Pdu through this connection. atpVariation: Derived, because TpNode can vary.	
M2 Parameter	
SystemTemplate::DiagnosticConnection::DiagnosticConnection.responseOnEvent, SystemTemplate::TransportProtocols::CanTpConnection	
Mapping Rule	**Mapping Type**
In case DiagnosticConnection.responseOnEvent exists and TpConnection Ident.ident belongs to a CanTpConnection.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00062

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDsl ProtocolType

BSW Parameter	BSW Type
DCM_ROE_ON_FLEXRAY	EcucEnumerationLiteralDef
BSW Description	
Template Description	
DiagnosticConnection.responseOnEvent: Reference to a ROE message. **FlexrayTpConnection**: A connection identifies the sender and the receiver of this particular communication. The FlexRayTp module routes a Pdu through this connection. In a System Description the references to the PduPools are mandatory. In an ECU Extract these references can be optional: On unicast connections these references are always mandatory. On multicast the txPduPool is mandatory on the sender side. The rxPduPool is mandatory on the receiver side. On Gateway ECUs both references are mandatory.	
M2 Parameter	
SystemTemplate::DiagnosticConnection::DiagnosticConnection.responseOnEvent, SystemTemplate::TransportProtocols::FlexrayTpConnection	
Mapping Rule	**Mapping Type**
In case DiagnosticConnection.responseOnEvent exists and TpConnection Ident.ident belongs to FlexRayTpConnection	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00063

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDsl ProtocolType
BSW Parameter	**BSW Type**
DCM_ROE_ON_IP	EcucEnumerationLiteralDef
BSW Description	
Template Description	
DiagnosticConnection.responseOnEvent: Reference to a ROE message. **SocketConnection**: The SoAd serves as a (De)Multiplexer between different PDU sources and the TCP/IP stack.	
M2 Parameter	
SystemTemplate::DiagnosticConnection::DiagnosticConnection.responseOnEvent, SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection	
Mapping Rule	**Mapping Type**
In case DiagnosticConnection.responseOnEvent exists and TpConnection Ident.ident belongs to SocketConnection	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00064

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDsl ProtocolType
BSW Parameter	**BSW Type**

DCM_UDS_ON_CAN	EcucEnumerationLiteralDef
BSW Description	
UDS on CAN (ISO15765-3; ISO14229-1)	
Template Description	
DiagnosticConnection.physicalRequest: Reference to a physical request message. **CanTpConnection**: A connection identifies the sender and the receiver of this particular communication. The CanTp module routes a Pdu through this connection. atpVariation: Derived, because TpNode can vary.	
M2 Parameter	
SystemTemplate::DiagnosticConnection::DiagnosticConnection.physicalRequest, SystemTemplate::TransportProtocols::CanTpConnection	
Mapping Rule	**Mapping Type**
In case DiagnosticConnection.physicalRequest exists and TpConnection Ident.ident belongs to CanTpConnection	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00065

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDsl ProtocolType
BSW Parameter	**BSW Type**
DCM_UDS_ON_FLEXRAY	EcucEnumerationLiteralDef
BSW Description	
UDS on FlexRay (Manufacturer specific; ISO14229-1)	
Template Description	
DiagnosticConnection.physicalRequest: Reference to a physical request message. **FlexrayTpConnection**: A connection identifies the sender and the receiver of this particular communication. The FlexRayTp module routes a Pdu through this connection. In a System Description the references to the PduPools are mandatory. In an ECU Extract these references can be optional: On unicast connections these references are always mandatory. On multicast the txPduPool is mandatory on the sender side. The rxPduPool is mandatory on the receiver side. On Gateway ECUs both references are mandatory.	
M2 Parameter	
SystemTemplate::DiagnosticConnection::DiagnosticConnection.physicalRequest, SystemTemplate::TransportProtocols::FlexrayTpConnection	
Mapping Rule	**Mapping Type**
In case DiagnosticConnection.physicalRequest exists and TpConnection Ident.ident belongs to FlexRayTpConnection	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00066

BSW Module	**BSW Context**

Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow/DcmDslProtocolType	
BSW Parameter		**BSW Type**
DCM_UDS_ON_IP		EcucEnumerationLiteralDef
BSW Description		
Template Description		
DiagnosticConnection.physicalRequest: Reference to a physical request message. **SocketConnection**: The SoAd serves as a (De)Multiplexer between different PDU sources and the TCP/IP stack.		
M2 Parameter		
SystemTemplate::DiagnosticConnection::DiagnosticConnection.physicalRequest, SystemTemplate::Fibex::Fibex4Ethernet::EthernetCommunication::SocketConnection		
Mapping Rule		**Mapping Type**
In case DiagnosticConnection.physicalRequest exists and TpConnectionIdent.ident belongs to a SocketConnection		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00067

BSW Module	**BSW Context**	
Dcm	Dcm/DcmConfigSet/DcmDsl/DcmDslProtocol/DcmDslProtocolRow	
BSW Parameter		**BSW Type**
DcmSendRespPendOnTransToBoot		EcucBooleanParamDef
BSW Description		
Parameter specifying if the ECU should send a NRC 0x78 (response pending) before transitioning to the bootloader (parameter set to TRUE) or if the transition shall be initiated without sending NRC 0x78 (parameter set to FALSE).		
Template Description		
The purpose of this attribute is to define whether or not the ECU should send a NRC 0x78 (response pending) before transitioning to the bootloader (in this case the attribute shall be set to "true") or if the transition shall be initiated without sending NRC 0x78 (in this case the attribute shall be set to "false").		
M2 Parameter		
DiagnosticExtract::DiagnosticContribution::DiagnosticProtocol.sendRespPendOnTransToBoot		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00294

BSW Module	**BSW Context**	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspComControl/DcmDspComControlAllChannel	
BSW Parameter		**BSW Type**
DcmDspAllComMChannelRef		EcucSymbolicNameReferenceDef
BSW Description		
Reference to ComM channel.		
Template Description		

This reference represents the semantics that all available channels shall be affected. It is still necessary to refer to individual CommunicatuionClusters because there could be private CommunicationClusters in the System Extract that are not subject to the service "communication control". By referring to the applicable CommunicationClusters it can be made sure that only the affected CommunicationClusters are accessed.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl::DiagnosticComControlClass.allChannels	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00072

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspComControl/DcmDspComControlSpecificChannel
BSW Parameter	**BSW Type**
DcmDspSpecificComMChannelRef	EcucSymbolicNameReferenceDef
BSW Description	
Reference to ComM channel.	
Template Description	
This represents the ability to add additional attributes to the case that only specific channels are supposed to be considered,	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl::DiagnosticComControlClass.specificChannel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00073

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspComControl/DcmDspComControlSpecificChannel
BSW Parameter	**BSW Type**
DcmDspSubnetNumber	EcucIntegerParamDef
BSW Description	
Subnet Number which controls the specific ComMChannel.	
Template Description	
This represents the applicable subnet number (which is an arbitrary number ranging from 1..14)	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl::DiagnosticComControlSpecificChannel.subnetNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00074

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspComControl

BSW Parameter	BSW Type
DcmDspComControlSubNode	EcucParamConfContainerDef
BSW Description	
This container gives information about the node identification number and the ComM channel used to address a sub-network.	
Template Description	
This attribute represents the ability to add further attributes to the definition of a specific sub-node channel that is subject to the diagnostic service "communication control".	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl::DiagnosticComControlClass.subNodeChannel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00075

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspComControl/DcmDspComControlSubNode
BSW Parameter	**BSW Type**
DcmDspComControlSubNodeComMChannelRef	EcucSymbolicNameReferenceDef
BSW Description	
This parameter references a ComM channel where this node is connected to.	
Template Description	
This represents the affected CommunicationClusters in the role subNodeChannel	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl::DiagnosticComControlSubNodeChannel.subNodeChannel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00076

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspComControl/DcmDspComControlSubNode
BSW Parameter	**BSW Type**
DcmDspComControlSubNodeId	EcucIntegerParamDef
BSW Description	
The node identification number DcmDspComControlSubNodeId is addressed by the CommunicationControl (0x28) request.	
Template Description	
This represents the applicable subNode number. The value corresponds to the request message parameter nodeIdentificationNumber of diagnostic service CommunicationControl (0x28).	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::CommunicationControl::DiagnosticComControlSubNodeChannel.subNodeNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00077

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp	
BSW Parameter		**BSW Type**
DcmDspCommonAuthorization		EcucParamConfContainerDef
BSW Description		
This container contains the configuration (parameters) for the common Authorization being equal for several services / sub-services.		
Template Description		
This represents an instance of the "Routine Control" diagnostic service.		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::RoutineControl::DiagnosticRoutineControl		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00254

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspControlDTCSetting	
BSW Parameter		**BSW Type**
DcmSupportDTCSettingControlOptionRecord		EcucBooleanParamDef
BSW Description		
This configuration switch defines if the DTCSettingControlOptionRecord is in general supported in the request message or not.		
Template Description		
This represents the decision whether the DTCSettingControlOptionRecord (see ISO 14229-1) is in general supported in the request message.		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::ControlDTCSetting::DiagnosticControlDTCSetting Class.controlOptionRecordPresent		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp	
BSW Parameter		**BSW Type**
DcmDspDDDIDcheckPerSourceDID		EcucBooleanParamDef
BSW Description		
Defines the check for session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22). true: Dcm module shall check the session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF false: Dcm module shall not check the session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF		
Template Description		

If set to TRUE, the Dcm module shall check the session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF. If set to FALSE. the Dcm module shall not check the session, security and mode dependencies per source DIDs with a ReadDataByIdentifier (0x22) with DID in the range 0xF200 to 0xF3FF.
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier::DiagnosticDynamicallyDefineDataIdentifierClass.checkPerSourceId

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00079

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData

BSW Parameter	**BSW Type**
DcmDspDataByteSize	EcucIntegerParamDef

BSW Description
Defines the array length in bytes or the the maximum array length for variable datalengths.
Template Description
DiagnosticDataElement.maxNumberOfElements: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.
M2 Parameter
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	**Mapping Type**
S/R via array: DcmDspDataByteSize= maxNumberOfElements * (baseTypeSize / 8) C/S of FNC callback: DcmDspDataByteSize= maxNumberOfElements Note: 8 is the baseTypeSize of UINT8	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00085

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData

BSW Parameter	**BSW Type**
DcmDspDataConditionCheckReadFnc	EcucFunctionNameDef

BSW Description
Function name to demand application if the conditions (e.g. System state) to read the DID are correct. (ConditionCheckRead-function). Multiplicity shall be equal to parameter DcmDspDataReadFnc. This parameter is related to the interface Xxx_ConditionCheckRead.
Template Description

This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00080

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspDataEndianness	EcucEnumerationParamDef
BSW Description	
Defines the endianness of the data belonging to a DID in a diagnostic request or response message.	
Template Description	
This attribute specifies the byte order of the base type.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder	
Mapping Rule	**Mapping Type**
baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticDataIdentifier.dataElement	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00081

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspDataFreezeCurrentStateFnc	EcucFunctionNameDef
BSW Description	
Function name to request to application to freeze the current state of an IOControl. (FreezeCurrentState-function). This parameter is related to the interface Xxx_FreezeCurrentState.	
Template Description	
DiagnosticServiceSwMapping.mappedBswServiceDependency: This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference. **DiagnosticIoControlNeeds.freezeCurrentStateSupported**: This attribute determines, if the referenced port supports temporary freezing of I/O value.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency, CommonStructure::ServiceNeeds::DiagnosticIoControlNeeds.freezeCurrentStateSupported	
Mapping Rule	**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00004

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData	
BSW Parameter		**BSW Type**
DcmDspDataReadDataLengthFnc		EcucFunctionNameDef
BSW Description		
Function name to request from application the data length of a DID. (ReadDataLength-function). This parameter is related to the interface Xxx_ReadDataLength.		
Template Description		
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.		
M2 Parameter		
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency		
Mapping Rule		**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00082

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData	
BSW Parameter		**BSW Type**
DcmDspDataReadFnc		EcucFunctionNameDef
BSW Description		
Function name to request from application the data value of a DID. (ReadData-function). This parameter is related to the interface Xxx_ReadData.		
Template Description		
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.		
M2 Parameter		
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency		
Mapping Rule		**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00083

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData	
BSW Parameter		**BSW Type**
DcmDspDataResetToDefaultFnc		EcucFunctionNameDef
BSW Description		
Function name to request to application to reset an IOControl to default value. (ResetToDefault-function). This parameter is related to the interface Xxx_ResetToDefault.		
Template Description		

DiagnosticServiceSwMapping.mappedBswServiceDependency: This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference. **DiagnosticIoControlNeeds.resetToDefaultSupported**: This represents a flag for the existence of the ResetToDefault operation in the service interface.
M2 Parameter
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency, CommonStructure::ServiceNeeds::DiagnosticIoControlNeeds.resetToDefaultSupported

Mapping Rule	**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00005

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData

BSW Parameter	**BSW Type**
DcmDspDataReturnControlToEcuFnc	EcucFunctionNameDef

BSW Description
Function name to request to application to return control to ECU of an IOControl. (ReturnControlToECU-function). This parameter is related to the interface Xxx_ReturnControlToECU.
Template Description
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.
M2 Parameter
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency

Mapping Rule	**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00084

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData

BSW Parameter	**BSW Type**
DcmDspDataShortTermAdjustmentFnc	EcucFunctionNameDef

BSW Description
Function name to request to application to adjust the IO signal. (ShortTermAdjustment-function). This parameter is related to the interface Xxx_ShortTermAdjustment.
Template Description

DiagnosticServiceSwMapping.mappedBswServiceDependency: This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference. **DiagnosticIoControlNeeds.shortTermAdjustmentSupported**: This attribute determines, if the referenced port supports temporarily setting of I/O value to a specific value provided by the diagnostic tester.
M2 Parameter
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency, CommonStructure::ServiceNeeds::DiagnosticIoControlNeeds.shortTermAdjustmentSupported

Mapping Rule	**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00006

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType

BSW Parameter	**BSW Type**
BOOLEAN	EcucEnumerationLiteralDef

BSW Description
Type of the data is boolean.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength**: This attribute controls whether the data length of the data is fixed.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength

Mapping Rule	**Mapping Type**
baseTypeEncoding = BOOLEAN baseTypeSize = 1 maxNumberOfElements shall not exist arraySizeSemantics shall not exist Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00008

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType

BSW Parameter	**BSW Type**
SINT16	EcucEnumerationLiteralDef

BSW Description

Type of the data is sint16.
Template Description
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **BaseTypeDirectDefinition.baseTypeEncoding**: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **DiagnosticValueNeeds.fixedLength**: This attribute controls whether the data length of the data is fixed.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength

Mapping Rule	**Mapping Type**
baseTypeEncoding = 2C baseTypeSize = 16 maxNumberOfElements shall not exist arraySizeSemantics shall not exist Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00012

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType

BSW Parameter	**BSW Type**
SINT16_N	EcucEnumerationLiteralDef
BSW Description	
Type of the data is sint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **BaseTypeDirectDefinition.baseTypeEncoding**: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticValueNeeds.fixedLength**: This attribute controls whether the data length of the data is fixed.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	

Mapping Rule	Mapping Type
baseTypeEncoding = 2C baseTypeSize = 16 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001) Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00018

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
SINT32	EcucEnumerationLiteralDef
BSW Description	
Type of the data is sint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **BaseTypeDirectDefinition.baseTypeEncoding**: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **DiagnosticValueNeeds.fixedLength**: This attribute controls whether the data length of the data is fixed.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	
Mapping Rule	**Mapping Type**
baseTypeEncoding = 2C baseTypeSize = 32 maxNumberOfElements shall not exist arraySizeSemantics shall not exist Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00014

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
SINT32_N	EcucEnumerationLiteralDef
BSW Description	
Type of the data is sint32 array.	
Template Description	

BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **BaseTypeDirectDefinition.baseTypeEncoding**: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticValueNeeds.fixedLength**: This attribute controls whether the data length of the data is fixed.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength

Mapping Rule	**Mapping Type**
baseTypeEncoding = 2C baseTypeSize = 32 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001) Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00020

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType

BSW Parameter	**BSW Type**
SINT8	EcucEnumerationLiteralDef

BSW Description
Type of the data is sint8.
Template Description
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **BaseTypeDirectDefinition.baseTypeEncoding**: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **DiagnosticValueNeeds.fixedLength**: This attribute controls whether the data length of the data is fixed.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength

Mapping Rule	Mapping Type
baseTypeEncoding = 2C baseTypeSize = 8 maxNumberOfElements shall not exist arraySizeSemantics shall not exist Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00010

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
SINT8_N	EcucEnumerationLiteralDef
BSW Description	
Type of the data is sint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **BaseTypeDirectDefinition.baseTypeEncoding**: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticValueNeeds.fixedLength**: This attribute controls whether the data length of the data is fixed.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	
Mapping Rule	**Mapping Type**
baseTypeEncoding = 2C baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001) Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00016

BSW Module	BSW Context

Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType	
BSW Parameter		**BSW Type**
UINT16		EcucEnumerationLiteralDef
BSW Description		
Type of the data is uint16.		
Template Description		
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength**: This attribute controls whether the data length of the data is fixed.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength		
Mapping Rule		**Mapping Type**
baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16 maxNumberOfElements shall not exist arraySizeSemantics shall not exist Derivation from DiagnosticValueNeeds.fixedLength=1 possible.		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00011

BSW Module	**BSW Context**	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType	
BSW Parameter		**BSW Type**
UINT16_N		EcucEnumerationLiteralDef
BSW Description		
Type of the data is uint16 array.		
Template Description		
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticValueNeeds.fixedLength**: This attribute controls whether the data length of the data is fixed.		
M2 Parameter		

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	
Mapping Rule	**Mapping Type**
baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001) Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00017

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
UINT32	EcucEnumerationLiteralDef
BSW Description	
Type of the data is uint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength**: This attribute controls whether the data length of the data is fixed.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	
Mapping Rule	**Mapping Type**
baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32 maxNumberOfElements shall not exist arraySizeSemantics shall not exist Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00013

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
UINT32_N	EcucEnumerationLiteralDef
BSW Description	

Type of the data is uint32 array.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticValueNeeds.fixedLength**: This attribute controls whether the data length of the data is fixed.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength

Mapping Rule	**Mapping Type**
baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001) Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00019

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType

BSW Parameter	**BSW Type**
UINT8	EcucEnumerationLiteralDef

BSW Description
Type of the data is uint8.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticValueNeeds.fixedLength**: This attribute controls whether the data length of the data is fixed.
M2 Parameter

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	
Mapping Rule	**Mapping Type**
baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8 maxNumberOfElements shall not exist arraySizeSemantics shall not exist Derivation from DiagnosticValueNeeds.fixedLength=1 possible.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00009

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType
BSW Parameter	**BSW Type**
UINT8_DYN	EcucEnumerationLiteralDef
BSW Description	
Type of the data is uint8 array with dynamic length.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticValueNeeds.fixedLength**: This attribute controls whether the data length of the data is fixed.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength	
Mapping Rule	**Mapping Type**
baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01002) arraySizeSemantics exists and is set to ArraySizeSemanticsEnum.variableSize (cf. TPS_DEXT_01002) Derivation from DiagnosticValueNeeds.fixedLength=0 possible.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00007

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataType	
BSW Parameter		**BSW Type**
UINT8_N		EcucEnumerationLiteralDef
BSW Description		
Type of the data is uint8 array.		
Template Description		
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticValueNeeds.fixedLength**: This attribute controls whether the data length of the data is fixed.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.fixedLength		
Mapping Rule		**Mapping Type**
baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001) Derivation from DiagnosticValueNeeds.fixedLength=1 possible.		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00015

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspDataUsePort	EcucEnumerationParamDef
BSW Description	
Defines which interface shall be used to access the data.	
Template Description	
This attribute controls whether interaction requires the software-component to react synchronously on a request or whether it processes the request in background but still the DCM has to issue the call again to eventually obtain the result of the request.	
M2 Parameter	
CommonStructure::ServiceNeeds::DiagnosticValueNeeds.processingStyle	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00001

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort
BSW Parameter	**BSW Type**
USE_DATA_ASYNCH_CLIENT_SERVER	EcucEnumerationLiteralDef
BSW Description	
The DCM will access the Data using an R-Port requiring a asynchronous ClientServertInterface DataServices_{Data}. The R-Port is named DataServices_{Data} where {Data} is the name of the container DcmDspData.	
Template Description	
The software-component processes the request in background but still the Dcm has to issue the call again to eventually obtain the result of the request.	
M2 Parameter	
CommonStructure::ServiceNeeds::DiagnosticProcessingStyleEnum.processingStyleAsynchronous	
Mapping Rule	**Mapping Type**
DiagnosticServiceSwMapping is having a SwcServiceDependency and ServiceNeeds::DiagnosticProcessingStyleEnum is equal to processingStyleAsynchronous	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00022

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort
BSW Parameter	**BSW Type**
USE_DATA_ASYNCH_CLIENT_SERVER_ERROR	EcucEnumerationLiteralDef
BSW Description	
The Dcm will access the Data using an R-Port requiring a asynchronous ClientServertInterface DataServices_{Data}. The parameter ErrorCode can be returned to allow the application to trigger a negative response during the operation. The R-Port is named DataServices_{Data} where {Data} is the name of the container DcmDspData.	
Template Description	
The software-component processes the request in background but still the Dcm has to issue the call again to eventually obtain the result of the request or handle error code.	
M2 Parameter	
CommonStructure::ServiceNeeds::DiagnosticProcessingStyleEnum.processingStyleAsynchronous WithError	
Mapping Rule	**Mapping Type**
DiagnosticServiceSwMapping is having a SwcServiceDependency and ServiceNeeds::DiagnosticProcessingStyleEnum is equal to processingStyleAsynchronousWithError	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00023

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort
BSW Parameter	**BSW Type**
USE_DATA_ASYNCH_FNC	EcucEnumerationLiteralDef

BSW Description	
The DCM will access the Data using the functions that are defined in parameters of type EcucFunctionNameDef (but without DcmDspDataReadDataLengthFnc) in the DcmDspData container. DCM_E_PENDING return is allowed. OpStatus is existing as IN parameter.	
Template Description	
The software-component processes the request in background but still the Dcm has to issue the call again to eventually obtain the result of the request.	
M2 Parameter	
CommonStructure::ServiceNeeds::DiagnosticProcessingStyleEnum.processingStyleAsynchronous	
Mapping Rule	**Mapping Type**
DiagnosticServiceSwMapping is having a BswServiceDependency and ServiceNeeds::DiagnosticProcessingStyleEnum is equal to processingStyleAsynchronous	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00250

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort
BSW Parameter	**BSW Type**
USE_DATA_SENDER_RECEIVER	EcucEnumerationLiteralDef
BSW Description	
The DCM will access the Data using an Port requiring a SenderReceiverInteface (with isService=false) DataServices_{Data}. The Port is namedDataServices_{Data} where {Data} is the name of the container DcmDspData.	
Template Description	
This represents the ability to define a mapping of a diagnostic service to a software-component. This kind of service mapping is applicable for the usage of SenderReceiverInterfaces.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00087

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDataUsePort
BSW Parameter	**BSW Type**
USE_DATA_SENDER_RECEIVER_AS_SERVICE	EcucEnumerationLiteralDef
BSW Description	
The DCM will access the Data using an service Port requiring a SenderReceiverInteface (with isService=true) DataServices_{Data} . The Port is namedDataServices_{Data} where {Data} is the name of the container DcmDspData.	
Template Description	
This represents the ability to define a mapping of a diagnostic service to a software-component. This kind of service mapping is applicable for the usage of SenderReceiverInterfaces.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00088

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData	
BSW Parameter		**BSW Type**
DcmDspDataWriteFnc		EcucFunctionNameDef
BSW Description		
Function name to request application to write the data value of a DID. (WriteData-function). This parameter is related to the interface Xxx_WriteData.		
Template Description		
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.		
M2 Parameter		
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency		
Mapping Rule		**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00090

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDiagnosisScaling/DcmDspAlternativeDataType	
BSW Parameter		**BSW Type**
DcmDspTextTableMapping		EcucParamConfContainerDef
BSW Description		
The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.		
Template Description		
This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.		
M2 Parameter		
AsamHdo::ComputationMethod::CompuMethod		
Mapping Rule		**Mapping Type**
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00097

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDiagnosisScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping	
BSW Parameter		**BSW Type**

DcmDspDiagnosisRepresentationDataValue	EcucIntegerParamDef
BSW Description	
The data value in the diagnosis representation.	
Template Description	
This represents a textual constant in the computation method.	
M2 Parameter	
AsamHdo::ComputationMethod::CompuConstTextContent.vt	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00098

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspDiagnosisScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspInternalDataValue	EcucIntegerParamDef
BSW Description	
The ECU internal data value.	
Template Description	
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit**: This specifies the upper limit of a of the scale.	
M2 Parameter	
AsamHdo::ComputationMethod::CompuScale.lowerLimit, AsamHdo::ComputationMethod::CompuScale.upperLimit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00099

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspDidDataSupportInfo	EcucParamConfContainerDef
BSW Description	
This container defines the supported information.	
Template Description	
DiagnosticParameter.supportInfo: This attribute represents the ability to define which bit of the support info byte is representing this part of the PID. **DiagnosticDataIdentifier.supportInfoByte**: This attribute represents the supported information associated with the DiagnosticDataIdentifier.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.supportInfo, DiagnosticExtract::CommonDiagnostics::DiagnosticDataIdentifier.supportInfoByte	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**

valid	up_Dcm_00282

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspExternalSRDataElementClass/DcmDataElementInstance
BSW Parameter	**BSW Type**
DcmDataElementInstanceRef	EcucInstanceReferenceDef
BSW Description	
Instance Reference to the primitive or array data which shall be read or written. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationPrimitiveDataType of category VALUE or BOOLEAN or ApplicationArrayDataType or if the AutosarDataPrototype is typed with a ImplementationDataType of category VALUE, ARRAY or TYPE_REFERENCE that in turn boils down to VALUE or ARRAY	
Template Description	
This represents the dataElement in the application software that is accessed for diagnostic purpose.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping.mappedDataElement	
Mapping Rule	**Mapping Type**
DiagnosticServiceDataMapping maps to a primitive data.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00100

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspExternalSRDataElementClass/DcmSubElementInDataElementInstance
BSW Parameter	**BSW Type**
DcmSubElementInDataElementInstanceRef	EcucInstanceReferenceDef
BSW Description	
Instance Reference to the primitve sub-element (at any level) of composite data in a port which shall be read. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationCompositeDataType.	
Template Description	
This represents the dataElement in the application software that is accessed for diagnostic purpose.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping.mappedDataElement	
Mapping Rule	**Mapping Type**
DiagnosticServiceDataMapping maps to a primitive element within a composite data, where the AutosarDataPrototype is typed with a ApplicationCompositeDataType.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00101

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData/DcmDspExternalSRDataElementClass/DcmSubElementInImplDataElementInstance
BSW Parameter	**BSW Type**

DcmSubElementInImplDataElementInstanceRef	EcucInstanceReferenceDef
BSW Description	
Instance Reference to the primitve sub-element (at any level) of composite data in a port which shall be read. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ImplementationDataType of category STRUCTURE or ARRAY. Please note that in case of ARRAY the index attribute in the target reference has to be set to select a single array element.	
Template Description	
This represents the dataElement in the application software that is accessed for diagnostic purpose.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping.mappedDataElement	
Mapping Rule	**Mapping Type**
DiagnosticServiceDataMapping maps to a primitive element within a composite data, where the AutosarDataPrototype is typed with a ApplicationCompositeDataType ImplementationDataType of category STRUCTURE or ARRAY.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00102

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspData
BSW Parameter	**BSW Type**
DcmDspOdxDataDescription	EcucAddInfoParamDef
BSW Description	
Defines additional description for ODX documentation	
Template Description	
This specifies the long name of the object. Long name is targeted to human readers and acts like a headline.	
M2 Parameter	
GenericStructure::GeneralTemplateClasses::Identifiable::MultilanguageReferrable.longName	
Mapping Rule	**Mapping Type**
Textual description that characterizes the DID element with respect to the ODX long name can be provided by means of the attribute long-Name.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00103

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp
BSW Parameter	**BSW Type**
DcmDspDataDefaultEndianness	EcucEnumerationParamDef
BSW Description	
Defines the default endianness belonging to a DID, RID or PID if the corresponding data does not define an endianness.	
Template Description	
Defines the default endianness of the data belonging to a DID or RID which is applicable if the DiagnosticDataElement does not define the endianness via the swDataDefProps.baseType attribute.	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.defaultEndianness	
Mapping Rule	**Mapping Type**
1:1 mapping	full

Mapping Status	Mapping ID
valid	up_Dcm_00104

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp
BSW Parameter	**BSW Type**
DcmDspDid	EcucParamConfContainerDef
BSW Description	
This container contains the configuration (parameters) of the DID.	
Template Description	
DiagnosticDataIdentifier: This meta-class represents the ability to model a diagnostic data identifier (DID) that is fully specified regarding the payload at configuration-time. **DiagnosticDynamicDataIdentifier**: This meta-class represents the ability to define a diagnostic data identifier (DID) at run-time.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticDataIdentifier DiagnosticExtract::CommonDiagnostics::DiagnosticDynamicDataIdentifier	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00178

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid
BSW Parameter	**BSW Type**
DcmDspDidIdentifier	EcucIntegerParamDef
BSW Description	
2 byte Identifier of the DID Within each DcmConfigSet all DcmDspDidIdentifier values shall be unique.	
Template Description	
DiagnosticAbstractDataIdentifier.id: This is the numerical identifier used to identify the DiagnosticAbstractDataIdentifier in the scope of diagnostic workflow **DiagnosticValueNeeds.didNumber**: This represents a Data identifier for the diagnostic value. This allows to predefine the DID number if the responsible function developer has received a particular requirement from the OEM or from a standardization body.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticAbstractDataIdentifier.id, CommonStructure::ServiceNeeds::DiagnosticValueNeeds.didNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00002

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid/DcmDspDidSignal

BSW Parameter	BSW Type
DcmDspDidByteOffset	EcucIntegerParamDef
BSW Description	
Defines the absolute byte offset of the data defined by DcmDspDidDataRef reference to DcmDspData container in the DID.	
Template Description	
This represents the bitOffset of the DiagnosticParameter	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.bitOffset	
Mapping Rule	**Mapping Type**
bitOffset / 8	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00283

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid
BSW Parameter	**BSW Type**
DcmDspDidSize	EcucIntegerParamDef
BSW Description	
Length of a DID in byte(s).	
Template Description	
This attribute indicates the size of the DiagnosticDataIdentifier.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticDataIdentifier.didSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00280

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDid
BSW Parameter	**BSW Type**
DcmDspDidSupportInfo	EcucParamConfContainerDef
BSW Description	
This container defines the support information to declare the usability of the data bytes within the DIDs	
Template Description	
This attribute represents the supported information associated with the DiagnosticDataIdentifier.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticDataIdentifier.supportInfoByte	
Mapping Rule	**Mapping Type**
	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00281

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo
BSW Parameter	**BSW Type**
DcmDspDDDIDMaxElements	EcucIntegerParamDef
BSW Description	

Maximum number of source elements of a DDDID.	
Template Description	
This represents the maximum number of source elements of the dynamically created DID.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier::DiagnosticDynamicallyDefineDataIdentifier.maxSourceElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00107

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl
BSW Parameter	**BSW Type**
DcmDspDidControlSecurityLevelRef	EcucReferenceDef
BSW Description	
Reference to DcmDspSecurityRow Security levels allowed to control this DID. If there is no reference, no check of security level shall be done.	
Template Description	
This represents the associated DiagnosticSecurityLevels	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticAccessPermission.securityLevel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00108

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl
BSW Parameter	**BSW Type**
DcmDspDidControlSessionRef	EcucReferenceDef
BSW Description	
Reference to DcmDspSessionRow Sessions allowed to control this DID. If there is no reference, no check of session level shall be done.	
Template Description	
This represents the associated DiagnosticSessions	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00109

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl
BSW Parameter	**BSW Type**
DcmDspDidFreezeCurrentState	EcucBooleanParamDef
BSW Description	
This indicates the presence of "FreezeCurrentState".	

Template Description
DiagnosticIOControl.freezeCurrentState: Setting this attribute to true represents the ability of the Dcm to execute a freezeCurrentState. **DiagnosticIoControlNeeds.freezeCurrentStateSupported**: This attribute determines, if the referenced port supports temporary freezing of I/O value.
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::IOControl::DiagnosticIOControl.freezeCurrentState, CommonStructure::ServiceNeeds::DiagnosticIoControlNeeds.freezeCurrentStateSupported

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00035

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl

BSW Parameter	BSW Type
DcmDspDidResetToDefault	EcucBooleanParamDef

BSW Description
This indicates the presence of "ResetToDefault".
Template Description
DiagnosticIOControl.resetToDefault: Setting this attribute to true represents the ability of the Dcm to execute a resetToDefault. **DiagnosticIoControlNeeds.resetToDefaultSupported**: This represents a flag for the existence of the ResetToDefault operation in the service interface.
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::IOControl::DiagnosticIOControl.resetToDefault, CommonStructure::ServiceNeeds::DiagnosticIoControlNeeds.resetToDefaultSupported

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00036

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidControl

BSW Parameter	BSW Type
DcmDspDidShortTermAdjustment	EcucBooleanParamDef

BSW Description
This indicates the presence of "ShortTermAdjustment".
Template Description
DiagnosticIOControl.shortTermAdjustment: Setting this attribute to true represents the ability of the Dcm to execute a shortTermAdjustment. **DiagnosticIoControlNeeds.shortTermAdjustmentSupported**: This attribute determines, if the referenced port supports temporarily setting of I/O value to a specific value provided by the diagnostic tester.
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::IOControl::DiagnosticIOControl.shortTermAdjustment, CommonStructure::ServiceNeeds::DiagnosticIoControlNeeds.shortTermAdjustmentSupported

Mapping Rule	Mapping Type
1:1 mapping	full

Mapping Status	Mapping ID
valid	up_Dcm_00037

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo
BSW Parameter	**BSW Type**
DcmDspDidDynamicallyDefined	EcucBooleanParamDef
BSW Description	
Indicates if this DID can be dynamically defined true = DID can be dynamically defined false = DID can not be dynamically defined	
Template Description	
DiagnosticDataIdentifier: This meta-class represents the ability to model a diagnostic data identifier (DID) that is fully specified regarding the payload at configuration-time. **DiagnosticDynamicDataIdentifier**: This meta-class represents the ability to define a diagnostic data identifier (DID) at run-time.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticDataIdentifier, DiagnosticExtract::CommonDiagnostics::DiagnosticDynamicDataIdentifier	
Mapping Rule	**Mapping Type**
true: in case the DiagnosticAbstractDataIdentifier for the DID value is aggregated by DiagnosticDynamicDataIdentifier false: in case the DiagnosticAbstract DataIdentifier for the DID value is aggregated by DiagnosticDataIdentifier	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00110

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidRead
BSW Parameter	**BSW Type**
DcmDspDidReadSecurityLevelRef	EcucReferenceDef
BSW Description	
Reference to DcmDspSecurityRow Referenced security levels are allowed to read this DID. If there is no reference, no check of security level shall be done.	
Template Description	
This represents the associated DiagnosticSecurityLevels	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticAccessPermission.securityLevel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00111

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidRead
BSW Parameter	**BSW Type**
DcmDspDidReadSessionRef	EcucReferenceDef
BSW Description	

Reference to DcmDspSessionRow Referenced sessions are allowed to read this DID. If there is no reference, no check of session level shall be done.
Template Description
This represents the associated DiagnosticSessions
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00112

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidWrite

BSW Parameter	**BSW Type**
DcmDspDidWriteSecurityLevelRef	EcucReferenceDef

BSW Description
Reference to DcmDspSecurityRow Referenced security levels are allowed to wirte this DID. If there is no reference, no check of security level shall be done.
Template Description
This represents the associated DiagnosticSecurityLevels
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticAccessPermission.securityLevel

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00113

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspDidInfo/DcmDspDidWrite

BSW Parameter	**BSW Type**
DcmDspDidWriteSessionRef	EcucReferenceDef

BSW Description
Reference to DcmDspSessionRow Referenced sessions are allowed to write this DID. If there is no reference, no check of session level shall be done.
Template Description
This represents the associated DiagnosticSessions
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00114

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspEcuReset/DcmDspEcuResetRow

BSW Parameter	**BSW Type**
DcmDspEcuResetId	EcucIntegerParamDef

BSW Description	
Defines the subfunction ID	
Template Description	
This represents the maximum number of source elements of the dynamically created DID.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier::DiagnosticDynamicallyDefineDataIdentifier.maxSourceElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00107

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspEcuReset/DcmDspEcuResetRow
BSW Parameter	**BSW Type**
DcmResponseToEcuReset	EcucEnumerationParamDef
BSW Description	
Defines the answer to EcuReset service should come: Before or after the reset.	
Template Description	
This attribute defines whether the response to the EcuReset service shall be transmitted before or after the actual reset.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EcuReset::DiagnosticEcuResetClass.respondToReset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00244

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspEcuReset/DcmDspEcuResetRow/DcmResponseToEcuReset
BSW Parameter	**BSW Type**
AFTER_RESET	EcucEnumerationLiteralDef
BSW Description	
Answer to EcuReset service should come after the reset.	
Template Description	
Answer to EcuReset service should come after the reset.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EcuReset::DiagnosticResponseToEcuResetEnum.respondAfterReset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00245

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspEcuReset/DcmDspEcuResetRow/DcmResponseToEcuReset
BSW Parameter	**BSW Type**
BEFORE_RESET	EcucEnumerationLiteralDef

BSW Description	
Answer to EcuReset service should come before the reset.	
Template Description	
Answer to EcuReset service should come before the reset.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EcuReset::DiagnosticResponseToEcuReset Enum.respondBeforeReset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00246

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp
BSW Parameter	**BSW Type**
DcmDspMaxDidToRead	EcucIntegerParamDef
BSW Description	
Indicates the maximum allowed DIDs in a single "ReadDataByIdentifier" request.	
Template Description	
This attribute represents the maximum number of allowed DIDs in a single instance of DiagnosticReadDataByIdentifier.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::DataByIdentifier::DiagnosticReadDataByIdentifier Class.maxDidToRead	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00115

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp
BSW Parameter	**BSW Type**
DcmDspMaxPeriodicDidToRead	EcucIntegerParamDef
BSW Description	
Indicates the maximum allowed periodicDIDs which can be read in a single "ReadDataByPeriodicIdentifier" request.	
Template Description	
This represents the maximum number of data identifiers that can be included in one request.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID::DiagnosticReadDataByPeriodicIDClass.maxPeriodicDidToRead	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00116

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo
BSW Parameter	**BSW Type**
DcmDspMemoryIdValue	EcucIntegerParamDef

BSW Description	
Value of the memory device identifier used. Each DcmDspMemoryIdInfo should have a unique ID. The MemoryIdValue is retrieved from the request messages (RMBA,WMBA,RD,RU,DDDI) according to ISO-14229-1.	
Template Description	
This represents the identification of the memory segment.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.id	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00117

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeByLabelInfo
BSW Parameter	**BSW Type**
DcmDspReadMemoryRangeByLabelHigh	EcucStringParamDef
BSW Description	
High memory address as label (string) of a range allowed for reading.	
Template Description	
This represents a symbolic label for the upper bound for addresses of the memory segment.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.memoryHighAddressLabel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00118

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeByLabelInfo
BSW Parameter	**BSW Type**
DcmDspReadMemoryRangeByLabelLow	EcucStringParamDef
BSW Description	
Low memory address as label (string) of a range allowed for reading.	
Template Description	
This represents a symbolic label for the lower bound for addresses of the memory segment.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.memoryLowAddressLabel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00119

BSW Module	BSW Context

Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeByLabelInfo

BSW Parameter	BSW Type
DcmDspReadMemoryRangeSecurityLevelRef	EcucReferenceDef

BSW Description
Link to the Security Access Levels needed for read access on this memory address. If there is no reference, no check of security level shall be done.

Template Description
This represents the associated DiagnosticSecurityLevels

M2 Parameter
DiagnosticExtract::Dcm::DiagnosticAccessPermission.securityLevel

Mapping Rule	Mapping Type
DiagnosticMemoryIdentifier referenced in the role memoryRange is referenced by a DiagnosticReadMemoryByAddress The accessPermission holds the security level information.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00120

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeByLabelInfo

BSW Parameter	BSW Type
DcmDspReadMemoryRangeSessionLevelRef	EcucReferenceDef

BSW Description
Link to the session level needed for access to this memory address range. If there is no reference, no check of session level shall be done.

Template Description
This represents the associated DiagnosticSessions

M2 Parameter
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00278

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeInfo

BSW Parameter	BSW Type
DcmDspReadMemoryRangeHigh	EcucIntegerParamDef

BSW Description
High memory address of a range allowed for reading

Template Description
This represents the upper bound for addresses of the memory segment.

M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.memoryHighAddress

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00121

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeInfo
BSW Parameter	**BSW Type**
DcmDspReadMemoryRangeLow	EcucIntegerParamDef
BSW Description	
Low memory address of a range allowed for reading	
Template Description	
This represents the lower bound for addresses of the memory segment.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.memoryLowAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00122

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeInfo
BSW Parameter	**BSW Type**
DcmDspReadMemoryRangeSecurityLevelRef	EcucReferenceDef
BSW Description	
Link to the Security Access Levels needed for read access on this memory address. If there is no reference, no check of security level shall be done.	
Template Description	
This represents the associated DiagnosticSecurityLevels	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticAccessPermission.securityLevel	
Mapping Rule	**Mapping Type**
DiagnosticMemoryIdentifier referenced in the role memoryRange is referenced by a DiagnosticReadMemoryByAddress The accessPermission holds the security level information.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00179

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspReadMemoryRangeInfo
BSW Parameter	**BSW Type**
DcmDspReadMemoryRangeSessionLevelRef	EcucReferenceDef
BSW Description	
Link to the session level needed for access to this memory address range. If there is no reference, no check of session level shall be done.	
Template Description	
This represents the associated DiagnosticSessions	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession	
Mapping Rule	**Mapping Type**
1:1 mapping	full

Mapping Status	Mapping ID
valid	up_Dcm_00276

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeByLabelInfo
BSW Parameter	**BSW Type**
DcmDspWriteMemoryRangeByLabelHigh	EcucStringParamDef
BSW Description	
High memory address as label (string) of a range allowed for writing.	
Template Description	
This represents a symbolic label for the upper bound for addresses of the memory segment.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.memoryHighAddressLabel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00123

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeByLabelInfo
BSW Parameter	**BSW Type**
DcmDspWriteMemoryRangeByLabelLow	EcucStringParamDef
BSW Description	
Low memory address as label (string) of a range allowed for writing.	
Template Description	
This represents a symbolic label for the lower bound for addresses of the memory segment.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.memoryLowAddressLabel	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00124

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeByLabelInfo
BSW Parameter	**BSW Type**
DcmDspWriteMemoryRangeSecurityLevelRef	EcucReferenceDef
BSW Description	
Link to the Security Access Levels needed for write access on this memory address. If there is no reference, no check of security level shall be done.	
Template Description	
This represents the associated DiagnosticSecurityLevels	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticAccessPermission.securityLevel	
Mapping Rule	**Mapping Type**

DiagnosticMemoryIdentifier referenced in the role memoryRange is referenced by a DiagnosticWriteMemoryByAddress The accessPermission holds the security level information.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00125

BSW Module	**BSW Context**	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeByLabelInfo	
BSW Parameter		**BSW Type**
DcmDspWriteMemoryRangeSessionLevelRef		EcucReferenceDef
BSW Description		
Link to the session level needed for access to this memory address range. If there is no reference, no check of session level shall be done.		
Template Description		
This represents the associated DiagnosticSessions		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00279

BSW Module	**BSW Context**	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeInfo	
BSW Parameter		**BSW Type**
DcmDspWriteMemoryRangeHigh		EcucIntegerParamDef
BSW Description		
High memory address of a range allowed for writing.		
Template Description		
This represents the upper bound for addresses of the memory segment.		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.memoryHighAddress		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00126

BSW Module	**BSW Context**	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeInfo	
BSW Parameter		**BSW Type**
DcmDspWriteMemoryRangeLow		EcucIntegerParamDef
BSW Description		
Low memory address of a range allowed for writing		
Template Description		
This represents the lower bound for addresses of the memory segment.		

M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::MemoryByAddress::DiagnosticMemoryIdentifier.memoryLowAddress	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00127

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeInfo	
BSW Parameter		**BSW Type**
DcmDspWriteMemoryRangeSecurityLevelRef		EcucReferenceDef
BSW Description		
Link to the Security Access Levels needed for write access on this memory address. If there is no reference, no check of security level shall be done.		
Template Description		
This represents the associated DiagnosticSecurityLevels		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticAccessPermission.securityLevel		
Mapping Rule		**Mapping Type**
DiagnosticMemoryIdentifier referenced in the role memoryRange is referenced by a DiagnosticWriteMemoryByAddress The accessPermission holds the security level information.		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00180

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspMemory/DcmDspMemoryIdInfo/DcmDspWriteMemoryRangeInfo	
BSW Parameter		**BSW Type**
DcmDspWriteMemoryRangeSessionLevelRef		EcucReferenceDef
BSW Description		
Link to the session level needed for access to this memory address range. If there is no reference, no check of session level shall be done.		
Template Description		
This represents the associated DiagnosticSessions		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticAccessPermission.diagnosticSession		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00277

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPeriodicTransmission	
BSW Parameter		**BSW Type**
DcmDspMaxPeriodicDidScheduler		EcucIntegerParamDef
BSW Description		

Defines the maximum number of periodicDataIdentifiers that can be scheduled concurrently.	
Template Description	
This represents the maximum number of periodic data identifiers that can be scheduled in parallel.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID::DiagnosticReadDataByPeriodicIDClass.schedulerMaxNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00128

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPeriodicTransmission
BSW Parameter	**BSW Type**
DcmDspPeriodicTransmissionFastRate	EcucFloatParamDef
BSW Description	
This parameter give the transmission rate of the requested periodicDataIdentifiers to be used if the parameter transmissionMode given in the ReadDataByPeriodicID request is equal to 0x03 ("sendAtFastRate"). This parameter value in seconds have to be configured as a multiple of DcmTaskTime. min: A negative value and zero is not allowed.	
Template Description	
DiagnosticPeriodicRate.period: This represents the period of the DiagnosticPeriodicRate in seconds. **DiagnosticPeriodicRate.periodicRateCategory**: This attribute represents the category of the periodic rate.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID::DiagnosticPeriodicRate.period DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID::DiagnosticPeriodicRate.periodicRateCategory	
Mapping Rule	**Mapping Type**
The parameter shall exist if DiagnosticPeriodicRate.periodicRateCategory is set to DiagnosticPeriodicRateCategoryEnum.periodicRateFast.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00129

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPeriodicTransmission
BSW Parameter	**BSW Type**
DcmDspPeriodicTransmissionMediumRate	EcucFloatParamDef
BSW Description	
This parameter give the transmission rate of the requested periodicDataIdentifiers to be used if the parameter transmissionMode given in the ReadDataByPeriodicID request is equal to 0x02 ("sendAtMediumRate"). This parameter value in seconds have to be configured as a multiple of DcmTaskTime. min: A negative value and zero is not allowed.	
Template Description	

DiagnosticPeriodicRate.period: This represents the period of the DiagnosticPeriodicRate in seconds. **DiagnosticPeriodicRate.periodicRateCategory**: This attribute represents the category of the periodic rate.
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID::DiagnosticPeriodicRate.period DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID::DiagnosticPeriodic Rate.periodicRateCategory

Mapping Rule	**Mapping Type**
The parameter shall exist if DiagnosticPeriodicRate.periodicRateCategory is set to DiagnosticPeriodicRateCategoryEnum.periodicRateMedium.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00130

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPeriodicTransmission

BSW Parameter	**BSW Type**
DcmDspPeriodicTransmissionSlowRate	EcucFloatParamDef

BSW Description
This parameter give the transmission rate of the requested periodicDataIdentifiers to be used if the parameter transmissionMode given in the ReadDataByPeriodicID request is equal to 0x01 ("sendAtSlowRate"). This parameter value in seconds have to be configured as a multiple of DcmTaskTime. min: A negative value and zero is not allowed.
Template Description
DiagnosticPeriodicRate.period: This represents the period of the DiagnosticPeriodicRate in seconds. **DiagnosticPeriodicRate.periodicRateCategory**: This attribute represents the category of the periodic rate.
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID::DiagnosticPeriodicRate.period DiagnosticExtract::Dcm::DiagnosticService::ReadDataByPeriodicID::DiagnosticPeriodic Rate.periodicRateCategory

Mapping Rule	**Mapping Type**
The parameter shall exist if DiagnosticPeriodicRate.periodicRateCategory is set to DiagnosticPeriodicRateCategoryEnum.periodicRateSlow.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00131

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp

BSW Parameter	**BSW Type**
DcmDspPid	EcucParamConfContainerDef

BSW Description
This container defines the availability of a PID to the DCM.
Template Description
This meta-class represents the ability to model a diagnostic parameter identifier (PID) for the purpose of executing on-board diagnostics (OBD).

M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameterIdentifier	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid
BSW Parameter	**BSW Type**
DcmDspPidData	EcucParamConfContainerDef
BSW Description	
This container defines the parameter for a Signal in the PID.	
Template Description	
This represents the data carried by the DiagnosticParameterIdentifier.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameterIdentifier.dataElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData
BSW Parameter	**BSW Type**
DcmDspPidByteOffset	EcucIntegerParamDef
BSW Description	
This is the position in bytes of the PID structure and will not start at position 0 in case a support information is available (for packeted PIDs).	
Template Description	
This represents the bitOffset of the DiagnosticParameter	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.bitOffset	
Mapping Rule	**Mapping Type**
bitOffset / 8	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00284

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData
BSW Parameter	**BSW Type**
DcmDspPidDataByteSize	EcucIntegerParamDef
BSW Description	
Defines the array length in bytes or the the maximum array length for variable datalengths.	
Template Description	
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	

M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
S/R via array: DcmDspPidDataByteSize= maxNumberOfElements * (baseTypeSize / 8) C/S of FNC callback: DcmDspPidDataByteSize= maxNumberOfElements Note: 8 is the baseTypeSize of UINT8	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00285

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData
BSW Parameter	**BSW Type**
DcmDspPidDataSupportInfo	EcucParamConfContainerDef
BSW Description	
This container defines the supported information.	
Template Description	
This attribute represents the ability to define which bit of the support info byte is representing this part of the PID.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.supportInfo	
Mapping Rule	**Mapping Type**
Applicable if the DiagnosticParameter is owned by a DiagnosticParameterIdentifier	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidDataSupportInfo
BSW Parameter	**BSW Type**
DcmDspPidDataSupportInfoBit	EcucIntegerParamDef
BSW Description	
Referenced Bit of the SupportInfo	
Template Description	
defines the bit in the SupportInfo byte, which represents the PID DataElement pidSize / position / size. Unit: byte.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameterSupportInfo.supportInfoBit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidDataSupportInfo
BSW Parameter	**BSW Type**

DcmDspPidDataSupportInfoRef	EcucReferenceDef
BSW Description	
Reference to DcmDspPidSupportInfo	
Template Description	
This represents the supported information associated with the DiagnosticParameterIdentifier.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameterIdentifier.supportInfoByte	
Mapping Rule	**Mapping Type**
Shall refer to the DiagnosticParameterIdentifier.supportInfoByte of the enclosing DiagnosticParameterIdentifier	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspDiagnosisScaling/DcmDspAlternativeDataType
BSW Parameter	**BSW Type**
DcmDspTextTableMapping	EcucParamConfContainerDef
BSW Description	
The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.	
Template Description	
This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.	
M2 Parameter	
AsamHdo::ComputationMethod::CompuMethod	
Mapping Rule	**Mapping Type**
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00097

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspDiagnosisScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspDiagnosisRepresentationDataValue	EcucIntegerParamDef
BSW Description	
The data value in the diagnosis representation.	
Template Description	
This represents a textual constant in the computation method.	

M2 Parameter	
AsamHdo::ComputationMethod::CompuConstTextContent.vt	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00098

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspDiagnosisScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspInternalDataValue	EcucIntegerParamDef
BSW Description	
The ECU internal data value.	
Template Description	
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit**: This specifies the upper limit of a of the scale.	
M2 Parameter	
AsamHdo::ComputationMethod::CompuScale.lowerLimit, AsamHdo::ComputationMethod::CompuScale.upperLimit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00099

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataEndianness
BSW Parameter	**BSW Type**
BIG_ENDIAN	EcucEnumerationLiteralDef
BSW Description	
Most significant byte shall be stored at the lowest address.	
Template Description	
This attribute specifies the byte order of the base type.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder	
Mapping Rule	**Mapping Type**
BaseTypeDirectDefinition.byteOrder == ByteOrderEnum.mostSignificantByteFirst	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataEndianness
BSW Parameter	**BSW Type**
LITTLE_ENDIAN	EcucEnumerationLiteralDef

BSW Description	
Most significant byte shall be stored at the highest address	
Template Description	
This attribute specifies the byte order of the base type.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder	
Mapping Rule	**Mapping Type**
BaseTypeDirectDefinition.byteOrder ==ByteOrderEnum.mostSignificantByteLast	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataEndianness
BSW Parameter	**BSW Type**
OPAQUE	EcucEnumerationLiteralDef
BSW Description	
Opaque data endianness	
Template Description	
This attribute specifies the byte order of the base type.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder	
Mapping Rule	**Mapping Type**
BaseTypeDirectDefinition.byteOrder == ByteOrderEnum.opaque	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01
BSW Parameter	**BSW Type**
DcmDspPidDataReadFnc	EcucFunctionNameDef
BSW Description	
Function name for reading PID data value. This is only relevant if DcmDspPidDataUsePort==USE_DATA_SYNCH_FNC. This parameter is related to the interface Xxx_ReadData.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
The BswServiceDependency should have aRoleBasedBswModuleEntryAssignment that in turn has attribute role set to xxx_ReadData and points to a Bsw ModuleEntry.	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType

BSW Parameter	BSW Type
BOOLEAN	EcucEnumerationLiteralDef

BSW Description
Type of the data is boolean.

Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.

M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement baseTypeEncoding = BOOLEAN baseTypeSize = 1	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType

BSW Parameter	BSW Type
SINT16	EcucEnumerationLiteralDef

BSW Description
Type of the data is sint16.

Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.

M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement baseTypeEncoding = 2C baseTypeSize = 16	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00154

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType

BSW Parameter	BSW Type
SINT16_N	EcucEnumerationLiteralDef

BSW Description

Type of the data is sint16 array.

Template Description

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize,
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding,
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics,
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements,

Mapping Rule	Mapping Type
baseTypeEncoding = 2C baseTypeSize = 16 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00160

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType

BSW Parameter	BSW Type
SINT32	EcucEnumerationLiteralDef

BSW Description

Type of the data is sint32.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

M2 Parameter

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding,
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement baseTypeEncoding = 2C baseTypeSize = 32	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType
BSW Parameter	**BSW Type**
SINT32_N	EcucEnumerationLiteralDef
BSW Description	
Type of the data is sint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **BaseTypeDirectDefinition.baseTypeEncoding**: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements,	
Mapping Rule	**Mapping Type**
baseTypeEncoding = 2C baseTypeSize = 32 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00166

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType
BSW Parameter	**BSW Type**
SINT8	EcucEnumerationLiteralDef
BSW Description	
Type of the data is sint8.	

Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement baseTypeEncoding = 2C baseTypeSize = 8	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType
BSW Parameter	**BSW Type**
SINT8_N	EcucEnumerationLiteralDef
BSW Description	
Type of the data is sint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeSize: Describes the length of the data type specified in the container in bits. **BaseTypeDirectDefinition.baseTypeEncoding**: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements,	
Mapping Rule	**Mapping Type**
baseTypeEncoding = 2C baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**

valid	up_Dcm_00153

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType

BSW Parameter	BSW Type
UINT16	EcucEnumerationLiteralDef

BSW Description

Type of the data is uint16.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

M2 Parameter

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding,
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00205

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType

BSW Parameter	BSW Type
UINT16_N	EcucEnumerationLiteralDef

BSW Description

Type of the data is uint16 array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements,	
Mapping Rule	**Mapping Type**
baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00132

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType
BSW Parameter	**BSW Type**
UINT32	EcucEnumerationLiteralDef
BSW Description	
Type of the data is uint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType
BSW Parameter	**BSW Type**
UINT32_N	EcucEnumerationLiteralDef
BSW Description	
Type of the data is uint32 array.	
Template Description	

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements,

Mapping Rule	**Mapping Type**
baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00133

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType

BSW Parameter	**BSW Type**
UINT8	EcucEnumerationLiteralDef
BSW Description	
Type of the data is uint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticParameterIdentifier.dataElement baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType

BSW Parameter	BSW Type
UINT8_DYN	EcucEnumerationLiteralDef

BSW Description

Type of the data is uint8 array with dynamic length.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

M2 Parameter

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding,
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize,
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics,
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements,

Mapping Rule	Mapping Type
baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01002) arraySizeSemantics exists and is set to ArraySizeSemanticsEnum.variableSize (cf. TPS_DEXT_01002)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00134

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataType

BSW Parameter	BSW Type
UINT8_N	EcucEnumerationLiteralDef

BSW Description

Type of the data is uint8 array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements,

Mapping Rule	**Mapping Type**
baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00172

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort

BSW Parameter	**BSW Type**
USE_DATA_SENDER_RECEIVER	EcucEnumerationLiteralDef

BSW Description

Template Description
This represents the ability to define a mapping of a diagnostic service to a software-component. This kind of service mapping is applicable for the usage of SenderReceiverInterfaces.
M2 Parameter
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping

Mapping Rule	**Mapping Type**
DiagnosticServiceDataMapping.diagnosticDataElement refers to a DiagnosticDataElement that in turn is aggregated in the PID definition where the PortPrototype referenced in the role mappedDataElement.contextPort refers to a DataInterface where attribute isService is set to false.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00256

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort

BSW Parameter	**BSW Type**

USE_DATA_SENDER_RECEIVER_AS_SERVICE	EcucEnumerationLiteralDef
BSW Description	
Template Description	
This represents the ability to define a mapping of a diagnostic service to a software-component. This kind of service mapping is applicable for the usage of SenderReceiverInterfaces.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping	
Mapping Rule	**Mapping Type**
DiagnosticServiceDataMapping.diagnosticDataElement refers to a Diagnostic DataElement that in turn is aggregated in the PID definition where the PortPrototype referenced in the role mappedDataElement.contextPort refers to a Data Interface where attribute isService is set to true.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00257

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort
BSW Parameter	**BSW Type**
USE_DATA_SYNCH_CLIENT_SERVER	EcucEnumerationLiteralDef
BSW Description	
Template Description	
This represents the ability to define a mapping of a diagnostic service to a software-component or a basic-software module. If the former is used then this kind of service mapping is applicable for the usage of ClientServerInterfaces.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping	
Mapping Rule	**Mapping Type**
DiagnosticServiceSwMapping.diagnosticDataElement refers to a Diagnostic DataElement that in turn is aggregated in the PID definition. DiagnosticServiceSwMapping.mappedSwcServiceDependency shall exist.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00255

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidDataUsePort
BSW Parameter	**BSW Type**
USE_DATA_SYNCH_FNC	EcucEnumerationLiteralDef
BSW Description	
Template Description	
This represents the ability to define a mapping of a diagnostic service to a software-component or a basic-software module. If the former is used then this kind of service mapping is applicable for the usage of ClientServerInterfaces.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping	
Mapping Rule	**Mapping Type**

DiagnosticServiceSwMapping.diagnosticDataElement refers to a Diagnostic DataElement that in turn is aggregated in the PID definition and that also defines the role mappedBswServiceDependency.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00258

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidService01ExternalSRDataElementClass/DcmDataElementInstance
BSW Parameter	**BSW Type**
DcmDataElementInstanceRef	EcucInstanceReferenceDef
BSW Description	
Instance Reference to the primitive or array data which shall be read or written. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationPrimitiveDataType of category VALUE or BOOLEAN or ApplicationArrayDataType or if the AutosarDataPrototype is typed with a ImplementationDataType of category VALUE, ARRAY or TYPE_REFERENCE that in turn boils down to VALUE or ARRAY	
Template Description	
This represents the dataElement in the application software that is accessed for diagnostic purpose.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping.mappedDataElement	
Mapping Rule	**Mapping Type**
DiagnosticServiceDataMapping maps to a primitive data.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00100

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidService01ExternalSRDataElementClass/DcmSubElementInDataElementInstance
BSW Parameter	**BSW Type**
DcmSubElementInDataElementInstanceRef	EcucInstanceReferenceDef
BSW Description	
Instance Reference to the primitve sub-element (at any level) of composite data in a port which shall be read. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationCompositeDataType.	
Template Description	
This represents the dataElement in the application software that is accessed for diagnostic purpose.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping.mappedDataElement	
Mapping Rule	**Mapping Type**
DiagnosticServiceDataMapping maps to a primitive element within a composite data, where the AutosarDataPrototype is typed with a ApplicationCompositeDataType.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00101

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService01/DcmDspPidService01ExternalSRDataElementClass/DcmSubElementInImplDataElementInstance

BSW Parameter	BSW Type
DcmSubElementInImplDataElementInstanceRef	EcucInstanceReferenceDef

BSW Description

Instance Reference to the primitve sub-element (at any level) of composite data in a port which shall be read.
Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only).
This reference is applicable if the AutosarDataPrototype is typed with a ImplementationDataType of category STRUCTURE or ARRAY.
Please note that in case of ARRAY the index attribute in the target reference has to be set to select a single array element.

Template Description

This represents the dataElement in the application software that is accessed for diagnostic purpose.

M2 Parameter

DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping.mappedDataElement

Mapping Rule	Mapping Type
DiagnosticServiceDataMapping maps to a primitive element within a composite data, where the AutosarDataPrototype is typed with a ApplicationCompositeDataType ImplementationDataType of category STRUCTURE or ARRAY.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00102

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidData/DcmDspPidService02

BSW Parameter	BSW Type
DcmDspPidDataDemRef	EcucReferenceDef

BSW Description

Reference to DemPidDataElement in DEM configuration. Allows to link the DCM PID and DEM PID configuration for Mode $02.

Template Description

This represents the PID associated with this instance of the OBD mode 0x02 service.

M2 Parameter

DiagnosticExtract::Dcm::ObdService::Mode_0x02_RequestPowertrainFreezeFrameData::DiagnosticPowertrainFreezeFrame.pid

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid

BSW Parameter	BSW Type
DcmDspPidIdentifier	EcucIntegerParamDef

BSW Description

1 byte Identifier of the PID

Within each DcmConfigSet all DcmDspPidIdentifier values shall be unique.

Template Description
ObdPidServiceNeeds.parameterId: Standardized parameter identifier (PID) according to the OBD standard specified in attribute "standard". **DiagnosticParameterIdentifier.id**: This is the numerical identifier used to identify the DiagnosticParameterIdentifier in the scope of diagnostic workflow (see SAE J1979-DA).
M2 Parameter
CommonStructure::ServiceNeeds::ObdPidServiceNeeds.parameterId, DiagnosticExtract::CommonDiagnostics::DiagnosticParameterIdentifier.id

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00028

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidService
BSW Parameter	**BSW Type**
DCM_SERVICE_01	EcucEnumerationLiteralDef
BSW Description	
A PID is used with service $01 only.	
Template Description	
This represents the PID associated with this instance of the OBD mode 0x01 service.	
M2 Parameter	
DiagnosticExtract::Dcm::ObdService::Mode_0x01_RequestCurrentPowertrainDiagnosticData:: DiagnosticRequestCurrentPowertrainData.pid	

Mapping Rule	Mapping Type
Applicable if the DiagnosticParameterIdentifier is only referenced by a DiagnosticRequestCurrentPowertrainData	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidService
BSW Parameter	**BSW Type**
DCM_SERVICE_01_02	EcucEnumerationLiteralDef
BSW Description	
A PID is used with service $01 and $02. Allowed with a PID configuration containing data elements on byte basis.	
Template Description	
DiagnosticRequestCurrentPowertrainData.pid: This represents the PID associated with this instance of the OBD mode 0x01 service. **DiagnosticPowertrainFreezeFrame.pid**: This represents the PID associated with this instance of the OBD mode 0x02 service.	
M2 Parameter	
DiagnosticExtract::Dcm::ObdService::Mode_0x01_RequestCurrentPowertrainDiagnosticData::DiagnosticRequestCurrentPowertrainData.pid, DiagnosticExtract::Dcm::ObdService::Mode_0x02_RequestPowertrainFreezeFrameData::DiagnosticPowertrainFreezeFrame.pid	

Mapping Rule	Mapping Type

Applicable if the DiagnosticParameterIdentifier is referenced by both a DiagnosticRequestCurrentPowertrainData and a DiagnosticPowertrainFreezeFrame	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidService
BSW Parameter	**BSW Type**
DCM_SERVICE_02	EcucEnumerationLiteralDef
BSW Description	
A PID is used with service $02 only. Allowed with a PID configuration containing data elements on byte basis.	
Template Description	
This represents the PID associated with this instance of the OBD mode 0x02 service.	
M2 Parameter	
DiagnosticExtract::Dcm::ObdService::Mode_0x02_RequestPowertrainFreezeFrameData::DiagnosticPowertrainFreezeFrame.pid	
Mapping Rule	**Mapping Type**
Applicable if the DiagnosticParameterIdentifier is only referenced by a DiagnosticPowertrainFreezeFrame	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid
BSW Parameter	**BSW Type**
DcmDspPidSize	EcucIntegerParamDef
BSW Description	
Length of a PID in byte(s).	
Template Description	
The size of the entire PID can be greater than the sum of the data elements because padding might be applied. Unit: byte.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameterIdentifier.pidSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid
BSW Parameter	**BSW Type**
DcmDspPidSupportInfo	EcucParamConfContainerDef
BSW Description	
This container defines the support information (typically byte A) to declare the usability of the data bytes within the so-called packeted PIDs (e.g. PID$68).	
Template Description	
This represents the supported information associated with the DiagnosticParameterIdentifier.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameterIdentifier.supportInfoByte	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidSupportInfo
BSW Parameter	**BSW Type**
DcmDspPidSupportInfoLen	EcucIntegerParamDef
BSW Description	
Length of the support information in bytes.	
Template Description	
This represents the size of the supportInfo within the PID. Unit: byte.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticSupportInfoByte.size	
Mapping Rule	**Mapping Type**
	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspPid/DcmDspPidSupportInfo
BSW Parameter	**BSW Type**
DcmDspPidSupportInfoPos	EcucIntegerParamDef
BSW Description	
Position of the support information in bytes.	
Template Description	
This represents the position of the supportInfo in the PID. Unit: byte.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticSupportInfoByte.position	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp
BSW Parameter	**BSW Type**
DcmDspRequestControl	EcucParamConfContainerDef
BSW Description	
This container contains the configuration (parameters) of the "Request control of on-board system, test or component" service (Service $08). The DCM will request the control using an R-Port requiring a PortInteface RequestControlServices_{Tid}. The R-Port is named RequestControlServices_{Tid} where {Tid} is the name of the container DcmDspRequestControl.	
Template Description	
This meta-class represents the ability to model an instance of the OBD mode 0x08 service.	
M2 Parameter	
DiagnosticExtract::Dcm::ObdService::Mode_0x08_RequestControlOfOnBoardDevice::DiagnosticRequestControlOfOnBoardDevice	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRequestControl
BSW Parameter	**BSW Type**
DcmDspRequestControlInBufferSize	EcucIntegerParamDef
BSW Description	
Number of bytes to be provided in the input buffer of the interface RequestControlServices_{Tid} for OBD Service $08	
Template Description	
This represents the specified data size for the request message. Unit: byte.	
M2 Parameter	
DiagnosticExtract::Dcm::ObdService::Mode_0x08_RequestControlOfOnBoardDevice::DiagnosticTestRoutineIdentifier.requestDataSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRequestControl
BSW Parameter	**BSW Type**
DcmDspRequestControlOutBufferSize	EcucIntegerParamDef
BSW Description	
Number of bytes to be provided in the output buffer of the interface RequestControlServices_{Tid} for OBD Service $08	
Template Description	
This represents the specified data size for the response message. Unit:byte.	
M2 Parameter	
DiagnosticExtract::Dcm::ObdService::Mode_0x08_RequestControlOfOnBoardDevice::DiagnosticTestRoutineIdentifier.responseDataSize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRequestControl
BSW Parameter	**BSW Type**
DcmDspRequestControlTestId	EcucIntegerParamDef
BSW Description	
Test Id for Service $08	
Template Description	
ObdControlServiceNeeds.testId: Test Identifier (TID) according to ISO 15031-5. **DiagnosticTestRoutineIdentifier.id**: This represents the numerical id of the DiagnosticTestIdentifier (see SAE J1979-DA).	

M2 Parameter	
CommonStructure::ServiceNeeds::ObdControlServiceNeeds.testId, DiagnosticExtract::Dcm::ObdService::Mode_0x08_RequestControlOfOnBoardDevice::DiagnosticTestRoutineIdentifier.id	
Mapping Rule	**Mapping Type**
The value shall be taken from DiagnosticRequestControlOfOnBoardDevice.testId.id if available.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00030

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp	
BSW Parameter		**BSW Type**
DcmDspRequestFileTransfer		EcucParamConfContainerDef
BSW Description		
This container contains the configuration for RequestFileTransfer. This container only exists if RequestFileTransfer is configured.		
Template Description		
This diagnostic service instance implements the UDS service 0x38.		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::RequestFileTransfer::DiagnosticRequestFileTransfer		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00135

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEvent/DcmDspRoeEventProperties/DcmDspRoeOnChangeOfDataIdentifier	
BSW Parameter		**BSW Type**
DcmDspRoeDidRef		EcucReferenceDef
BSW Description		
Reference to a Did which is watched.		
Template Description		
This represents the corresponding DiagnosticDataIdentifier.		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticDataChangeTrigger.dataIdentifier		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00136

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEvent/DcmDspRoeEventProperties/DcmDspRoeOnDTCStatusChange	
BSW Parameter		**BSW Type**
DcmDspRoeDTCStatusMask		EcucIntegerParamDef
BSW Description		
Value of the relevant DTCStatusMask		

Template Description	
This attribute represents the ability to define a status mask for the triggering of an ROE response on the change of a DTC.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticDtcChangeTrigger.dtc StatusMask	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00295

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEvent
BSW Parameter	**BSW Type**
DcmDspRoeInitialEventStatus	EcucEnumerationParamDef
BSW Description	
Initial Roe status of this RoeEvent	
Template Description	
This represents the initial status of the enclosing DiagnosticResponseOnEventTrigger.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticResponseOnEvent Trigger.initialEventStatus	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00137

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEvent/DcmDspRoe InitialEventStatus
BSW Parameter	**BSW Type**
DCM_ROE_CLEARED	EcucEnumerationLiteralDef
BSW Description	
Template Description	
This means that the ResponseOnEvent is initially cleared.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticInitialEventStatus Enum.returnOnEventCleared	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00138

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEvent/DcmDspRoe InitialEventStatus
BSW Parameter	**BSW Type**
DCM_ROE_STOPPED	EcucEnumerationLiteralDef
BSW Description	

Template Description	
This means that the ResponseOnEvent is initially stopped.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticInitialEventStatusEnum.returnOnEventStopped	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00139

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe
BSW Parameter	**BSW Type**
DcmDspRoeEventWindowTime	EcucParamConfContainerDef
BSW Description	
This container configures the available EventWindowTime in this Ecu. This container contains a sub-set of EventWindowTimes supported by the Dcm, to limit the Ecu resources.	
Template Description	
This attribute clarifies the validity of the eventWindow	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticEventWindow.eventWindowTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00140

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEventWindowTime/DcmDspRoeEventWindowTime
BSW Parameter	**BSW Type**
DCM_ROE_EVENT_WINDOW_CURRENT_AND_FOLLOWING_CYCLE	EcucEnumerationLiteralDef
BSW Description	
Template Description	
This means that the window extends to this and the following cycle.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticEventWindowTimeEnum.eventWindowCurrentAndFollowingCycle	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00141

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEventWindowTime/DcmDspRoeEventWindowTime

BSW Parameter	BSW Type
DCM_ROE_EVENT_WINDOW_CURRENT_CYCLE	EcucEnumerationLiteralDef
BSW Description	
Template Description	
This means that the window is limited to the current cycle.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticEventWindowTime Enum.eventWindowCurrentCycle	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00142

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEventWindowTime/ DcmDspRoeEventWindowTime
BSW Parameter	**BSW Type**
DCM_ROE_EVENT_WINDOW_INFINITE	EcucEnumerationLiteralDef
BSW Description	
Template Description	
This means that the window extents without a border.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticEventWindowTime Enum.eventWindowInfinite	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00143

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe/DcmDspRoeEventWindowTime
BSW Parameter	**BSW Type**
DcmDspRoeStorageState	EcucBooleanParamDef
BSW Description	
If this parameter is set to TRUE the StorageStateBit will be evaluated if this EventWindowTime is requested.	
Template Description	
If this attribute is set to TRUE the StorageStateBit will be evaluated if this EventWindowTime is requested.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticEventWindow.storage StateEvaluation	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00144

BSW Module	BSW Context

Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoe	
BSW Parameter		**BSW Type**
DcmDspRoeInterMessageTime		EcucFloatParamDef
BSW Description		
Provide the minimum time in seconds between two transmissions of ROE event. It is used for the delay between two different consecutive Roe transmissions.		
Template Description		
Provide the minimum time in seconds between two consecutive transmissions of an ROE event.		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::ResponseOnEvent::DiagnosticResponseOnEvent Class.interMessageTime		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00145

BSW Module	**BSW Context**	
Dcm	Dcm/DcmConfigSet/DcmDsp	
BSW Parameter		**BSW Type**
DcmDspRoutine		EcucParamConfContainerDef
BSW Description		
This container contains the configuration (parameters) for Routines		
Template Description		
This meta-class represents the ability to define a diagnostic routine.		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticRoutine		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00181

BSW Module	**BSW Context**	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine	
BSW Parameter		**BSW Type**
DcmDspRequestRoutineResults		EcucParamConfContainerDef
BSW Description		
Provides the configuration of RequestResult subservice for RoutineControl service. Existence indicates that the RequestRoutineResults in the RoutineControl is supported.		
Template Description		
DiagnosticRoutine.requestResult: This represents the ability to request the result of a running routine. **DiagnosticRoutineNeeds.diagRoutineType**: This denotes the type of diagnostic routine which is implemented by the referenced server port.		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticRoutine.requestResult, CommonStructure::ServiceNeeds::DiagnosticRoutineNeeds.diagRoutineType		
Mapping Rule		**Mapping Type**

1:1 mapping for DiagnosticRoutine.requestResult OR DiagnosticRoutineNeeds.diagRoutineTyoe == asynchronous	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00026

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults
BSW Parameter	**BSW Type**
DcmDspRequestRoutineResultsFnc	EcucFunctionNameDef
BSW Description	
Function name for request to application the results of a routine. (Routine_RequestResults-function) This parameter is related to the interface Xxx_RequestResults.	
Template Description	
Specialization of ServiceDependency in the context of an BswInternalBehavior. It allows to associate BswModuleEntries and data defined for a BSW module or cluster to a given ServiceNeeds element.	
M2 Parameter	
BswModuleTemplate::BswBehavior::BswServiceDependency	
Mapping Rule	**Mapping Type**
It could be possible to get the FNC name via BswServiceDependency	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00147

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults
BSW Parameter	**BSW Type**
DcmDspRequestRoutineResultsOut	EcucParamConfContainerDef
BSW Description	
Provide description of output parameter of RequestResult subservice for RoutineControl service.	
Template Description	
This represents the response parameters.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticRequestRoutineResults.response	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00148

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut
BSW Parameter	**BSW Type**
DcmDspRequestRoutineResultsOutSignal	EcucParamConfContainerDef
BSW Description	

Provides description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataOutN elements in the XXX_RequestResult function call.
Template Description
This represents the related dataElement of the DiagnosticParameter
M2 Parameter
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.dataElement

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00149

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType

BSW Parameter	**BSW Type**
DcmDspTextTableMapping	EcucParamConfContainerDef
BSW Description	
The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.	
Template Description	
This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.	
M2 Parameter	
AsamHdo::ComputationMethod::CompuMethod	

Mapping Rule	**Mapping Type**
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00097

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping

BSW Parameter	**BSW Type**
DcmDspDiagnosisRepresentationDataValue	EcucIntegerParamDef
BSW Description	
The data value in the diagnosis representation.	

Template Description	
This represents a textual constant in the computation method.	
M2 Parameter	
AsamHdo::ComputationMethod::CompuConstTextContent.vt	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00098

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspInternalDataValue	EcucIntegerParamDef
BSW Description	
The ECU internal data value.	
Template Description	
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit**: This specifies the upper limit of a of the scale.	
M2 Parameter	
AsamHdo::ComputationMethod::CompuScale.lowerLimit, AsamHdo::ComputationMethod::CompuScale.upperLimit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00099

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal
BSW Parameter	**BSW Type**
DcmDspRoutineParameterSize	EcucIntegerParamDef
BSW Description	
Provide the size of a RoutineControl parameter in bytes	
Template Description	
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00189

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal	
BSW Parameter		**BSW Type**
DcmDspRoutineSignalEndianness		EcucEnumerationParamDef
BSW Description		
Defines the endianness of the data belonging to a Routine Out Signal for RequestResult subfunction.		
Template Description		
This attribute specifies the byte order of the base type.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder		
Mapping Rule		**Mapping Type**
baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnostictRequestRoutineResult.response		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00185

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal	
BSW Parameter		**BSW Type**
DcmDspRoutineSignalPos		EcucIntegerParamDef
BSW Description		
Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.		
Template Description		
This represents the bitOffset of the DiagnosticParameter		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.bitOffset		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00193

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType	
BSW Parameter		**BSW Type**
BOOLEAN		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is boolean.		
Template Description		
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.		
M2 Parameter		

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response baseTypeEncoding = BOOLEAN baseTypeSize = 1	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00199

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response baseTypeEncoding = 2C baseTypeSize = 16	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00156

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint16 array.	
Template Description	

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response baseTypeEncoding = 2C baseTypeSize = 16 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00156

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
SINT32	EcucEnumerationLiteralDef

BSW Description
Type of the signal is sint32.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	**Mapping Type**

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response baseTypeEncoding = 2C baseTypeSize = 32	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00163

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
SINT32_N	EcucEnumerationLiteralDef

BSW Description
Type of the signal is sint32 array.

Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response baseTypeEncoding = 2C baseTypeSize = 32 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00163

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT8	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response baseTypeEncoding = 2C baseTypeSize = 8	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00201

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT8_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response baseTypeEncoding = 2C baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00201

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00207

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint16 array.	
Template Description	

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00207

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
UINT32	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00210

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
UINT32_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response baseTypeEncoding = NONE, UTF-32, UTF-32 baseTypeSize = 32 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00210

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
UINT8	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00167

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT8_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf.TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00167

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspRequestRoutineResults/DcmDspRequestRoutineResultsOut/DcmDspRequestRoutineResultsOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
VARIABLE_LENGTH	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint8[DcmDspRoutineParameterSize]. This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticRequestRoutineResults.response baseTypeEncoding = NONE baseTypeSize = 8 arraySizeSemantics = variableSize maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01002) arraySizeSemantics exists and is set to ArraySizeSemanticsEnum.variableSize (cf. TPS_DEXT_01002)	full

Mapping Status	Mapping ID
valid	up_Dcm_00217

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine
BSW Parameter	**BSW Type**
DcmDspRoutineIdentifier	EcucIntegerParamDef
BSW Description	
2 bytes Identifier of the RID Within each DcmConfigSet all DcmDspRoutineIdentifier values shall be unique.	
Template Description	
DiagnosticRoutine.id: This is the numerical identifier used to identify the DiagnosticRoutine in the scope of diagnostic workflow **DiagnosticRoutineNeeds.ridNumber**: This represents a routine identifier for the diagnostic routine. This allows to predefine the RID number if the a function developer has received a particular requirement from the OEM or from a standardization body.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticRoutine.id, CommonStructure::ServiceNeeds::DiagnosticRoutineNeeds.ridNumber	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00003

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine
BSW Parameter	**BSW Type**
DcmDspRoutineUsePort	EcucBooleanParamDef
BSW Description	
If this parameter is set to true, the DCM uses a port requiring a PortInterface RoutineServices_{RoutineName}. The R-Port is named RoutineServices_{RoutineName} where {RoutineName} is the name of the container DcmDspRoutine In that case, the configuration must not provide function names in DcmDspStartRoutineFnc, DcmDspStopRoutineFnc or DcmDspRequestResultsRoutineFnc. If this is false, the DCM expects to find the names of the functions to be used in DcmDspStartRoutineFnc, DcmDspStopRoutineFnc or DcmDspRequestResultsRoutineFnc.	
Template Description	
This represents the ability to define a mapping of a diagnostic service to a software-component or a basic-software module. If the former is used then this kind of service mapping is applicable for the usage of ClientServerInterfaces.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping	
Mapping Rule	**Mapping Type**
TRUE: DiagnosticServiceSwMapping is having a SwcServiceDependency FALSE: DiagnosticServiceSwMapping is having a BswServiceDependency	full
Mapping Status	**Mapping ID**

valid	up_Dcm_00174

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine	
BSW Parameter		**BSW Type**
DcmDspStartRoutine		EcucParamConfContainerDef
BSW Description		
Provides the configuration of Start subservice for RoutineControl service.		
Template Description		
DiagnosticRoutine.start: This represents the ability to start a routine **DiagnosticRoutineNeeds**: Specifies the general needs on the configuration of the Diagnostic Communication Manager (Dcm) which are not related to a particular item (e.g. a PID). The main use case is the mapping of service ports to the Dcm which are not related to a particular item.		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticRoutine.start, CommonStructure::ServiceNeeds::DiagnosticRoutineNeeds		
Mapping Rule		**Mapping Type**
A routine always comes with a start routine, independently of whether the execution is done synchronously or asynchronously.		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00024

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine	
BSW Parameter		**BSW Type**
DcmDspStartRoutineFnc		EcucFunctionNameDef
BSW Description		
Function name for request to application to start a routine. (Routine_Start-function) This parameter is related to the interface Xxx_Start.		
Template Description		
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.		
M2 Parameter		
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00175

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine	
BSW Parameter		**BSW Type**
DcmDspStartRoutineIn		EcucParamConfContainerDef
BSW Description		
Provide description of input parameter of Start subservice for RoutineControl service		

Template Description	
This represents the request parameters.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticStartRoutine.request	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00176

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn
BSW Parameter	**BSW Type**
DcmDspStartRoutineInSignal	EcucParamConfContainerDef
BSW Description	
Provide description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataInN elements in the XXX_Start function call.	
Template Description	
This represents the related dataElement of the DiagnosticParameter	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.dataElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00177

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType
BSW Parameter	**BSW Type**
DcmDspTextTableMapping	EcucParamConfContainerDef
BSW Description	
The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.	
Template Description	
This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.	
M2 Parameter	
AsamHdo::ComputationMethod::CompuMethod	

Mapping Rule	Mapping Type
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00097

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspDiagnosisRepresentationDataValue	EcucIntegerParamDef
BSW Description	
The data value in the diagnosis representation.	
Template Description	
This represents a textual constant in the computation method.	
M2 Parameter	
AsamHdo::ComputationMethod::CompuConstTextContent.vt	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00098

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspInternalDataValue	EcucIntegerParamDef
BSW Description	
The ECU internal data value.	
Template Description	
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit**: This specifies the upper limit of a of the scale.	
M2 Parameter	
AsamHdo::ComputationMethod::CompuScale.lowerLimit, AsamHdo::ComputationMethod::CompuScale.upperLimit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00099

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal
BSW Parameter	**BSW Type**
DcmDspRoutineParameterSize	EcucIntegerParamDef
BSW Description	

Provide the size of a RoutineControl parameter in bytes	
Template Description	
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
"Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00186

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal
BSW Parameter	**BSW Type**
DcmDspRoutineSignalEndianness	EcucEnumerationParamDef
BSW Description	
Defines the endianness of the data belonging to a Routine In Signal for Start subfunction.	
Template Description	
This attribute specifies the byte order of the base type.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder	
Mapping Rule	**Mapping Type**
baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00182

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal
BSW Parameter	**BSW Type**
DcmDspRoutineSignalPos	EcucIntegerParamDef
BSW Description	
Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.	
Template Description	
This represents the bitOffset of the DiagnosticParameter	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.bitOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00190

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**

BOOLEAN	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is boolean.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request baseTypeEncoding = BOOLEAN baseTypeSize = 1	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00196

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request baseTypeEncoding = 2C baseTypeSize = 16	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00155

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**

SINT16_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request baseTypeEncoding = 2C baseTypeSize = 16 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00155

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT32	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request baseTypeEncoding = 2C baseTypeSize = 32	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00164

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT32_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request baseTypeEncoding = 2C baseTypeSize = 32 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00164

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT8	EcucEnumerationLiteralDef

BSW Description	
Type of the signal is sint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request baseTypeEncoding = 2C baseTypeSize = 8	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00200

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT8_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request baseTypeEncoding = 2C baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either doesnot exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00200

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00204

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint16 array.	
Template Description	

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00204

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
UINT32	EcucEnumerationLiteralDef

BSW Description
Type of the signal is uint32.

Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32	full

Mapping Status	Mapping ID
valid	up_Dcm_00213

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT32_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request baseTypeEncoding = NONE, UTF-32 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00213

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT8	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint8.	
Template Description	

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00169

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
UINT8_N	EcucEnumerationLiteralDef

BSW Description
Type of the signal is uint8 array.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements

Mapping Rule	**Mapping Type**

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00169

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineIn/DcmDspStartRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
VARIABLE_LENGTH	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint8[DcmDspRoutineParameterSize]. This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.request baseTypeEncoding = NONE baseTypeSize = 8 arraySizeSemantics = variableSize maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01002) arraySizeSemantics exists and is set to ArraySizeSemanticsEnum.variableSize (cf. TPS_DEXT_01002)	full
Mapping Status	**Mapping ID**

valid	up_Dcm_00219

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine	
BSW Parameter		**BSW Type**
DcmDspStartRoutineOut		EcucParamConfContainerDef
BSW Description		
Provide description of output parameter of Start subservice for RoutineControl service.		
Template Description		
This represents the response parameters.		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticStartRoutine.response		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00243

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDsp StartRoutineOut	
BSW Parameter		**BSW Type**
DcmDspStartRoutineOutSignal		EcucParamConfContainerDef
BSW Description		
Provide description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataOutN elements in the XXX_Start function call.		
Template Description		
This represents the related dataElement of the DiagnosticParameter		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.dataElement		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00242

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDsp StartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspArgumentScaling/ DcmDspAlternativeDataType	
BSW Parameter		**BSW Type**
DcmDspTextTableMapping		EcucParamConfContainerDef
BSW Description		

The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.
Template Description
This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.
M2 Parameter
AsamHdo::ComputationMethod::CompuMethod

Mapping Rule	**Mapping Type**
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00097

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping

BSW Parameter	**BSW Type**
DcmDspDiagnosisRepresentationDataValue	EcucIntegerParamDef
BSW Description	
The data value in the diagnosis representation.	
Template Description	
This represents a textual constant in the computation method.	
M2 Parameter	
AsamHdo::ComputationMethod::CompuConstTextContent.vt	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00098

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping

BSW Parameter	**BSW Type**
DcmDspInternalDataValue	EcucIntegerParamDef
BSW Description	
The ECU internal data value.	
Template Description	

CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit**: This specifies the upper limit of a of the scale.
M2 Parameter
AsamHdo::ComputationMethod::CompuScale.lowerLimit, AsamHdo::ComputationMethod::CompuScale.upperLimit

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00099

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal

BSW Parameter	**BSW Type**
DcmDspRoutineParameterSize	EcucIntegerParamDef
BSW Description	
Provide the size of a RoutineControl parameter in bytes	
Template Description	
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	

Mapping Rule	**Mapping Type**
Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00151

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal

BSW Parameter	**BSW Type**
DcmDspRoutineSignalEndianness	EcucEnumerationParamDef
BSW Description	
Defines the endianness of the data belonging to a Routine Out Signal for Start subfunction.	
Template Description	
This attribute specifies the byte order of the base type.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder	

Mapping Rule	**Mapping Type**
baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00150

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal

BSW Parameter	BSW Type
DcmDspRoutineSignalPos	EcucIntegerParamDef
BSW Description	
Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.	
Template Description	
This represents the bitOffset of the DiagnosticParameter	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.bitOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00152

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDsp StartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
BOOLEAN	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is boolean.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response baseTypeEncoding = BOOLEAN baseTypeSize = 1	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00198

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDsp StartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint16.	
Template Description	

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response baseTypeEncoding = 2C baseTypeSize = 16	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00157

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response baseTypeEncoding = 2C baseTypeSize = 16 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full

Mapping Status	Mapping ID
valid	up_Dcm_00157

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT32	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response baseTypeEncoding = 2C baseTypeSize = 32	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00161

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT32_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size.	
M2 Parameter	

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response baseTypeEncoding = 2C baseTypeSize = 32 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00161

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT8	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response baseTypeEncoding = 2C baseTypeSize = 8	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00202

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT8_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint8 array.	
Template Description	

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response baseTypeEncoding = 2C baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00202

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
UINT16	EcucEnumerationLiteralDef

BSW Description
Type of the signal is uint16.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16	full

Mapping Status	Mapping ID
valid	up_Dcm_00206

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemanticseither does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00206

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT32	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint32.	
Template Description	

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00215

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
UINT32_N	EcucEnumerationLiteralDef

BSW Description
Type of the signal is uint32 array.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemanticseither does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full

Mapping Status	Mapping ID
valid	up_Dcm_00215

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT8	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00168

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT8_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00168

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStartRoutine/DcmDspStartRoutineOut/DcmDspStartRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
VARIABLE_LENGTH	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint8[DcmDspRoutineParameterSize]. This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStartRoutine.response baseTypeEncoding = NONE baseTypeSize = 8 arraySizeSemantics = variableSize maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01002) arraySizeSemantics exists and is set to ArraySizeSemanticsEnum.variableSize (cf. TPS_DEXT_01002)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00216

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine
BSW Parameter	**BSW Type**
DcmDspStopRoutine	EcucParamConfContainerDef
BSW Description	
Provides the configuration of Stop subservice for RoutineControl service. Existence indicates that the StopRoutine in the RoutineControl is supported.	
Template Description	
DiagnosticRoutine.stop: This represents the ability to stop a running routine. **DiagnosticRoutineNeeds.diagRoutineType**: This denotes the type of diagnostic routine which is implemented by the referenced server port.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticRoutine.stop, CommonStructure::ServiceNeeds::DiagnosticRoutineNeeds.diagRoutineType	
Mapping Rule	**Mapping Type**
1:1 mapping for DiagnosticRoutine.stop OR DiagnosticRoutineNeeds.diagRoutineType == asynchronous	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00025

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine
BSW Parameter	**BSW Type**
DcmDspStopRoutineFnc	EcucFunctionNameDef
BSW Description	
Function name for request to application to stop a routine. (Routine_Stop-function) This parameter is related to the interface Xxx_Stop.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	

DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00221

BSW Module	**BSW Context**	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine	
BSW Parameter		**BSW Type**
DcmDspStopRoutineIn		EcucParamConfContainerDef
BSW Description		
Provide description of input parameter of Stop subservice for RoutineControl service.		
Template Description		
This represents the request parameters.		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticStopRoutine.request		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00220

BSW Module	**BSW Context**	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn	
BSW Parameter		**BSW Type**
DcmDspStopRoutineInSignal		EcucParamConfContainerDef
BSW Description		
Provide description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataInN elements in the XXX_Stop function call.		
Template Description		
This represents the related dataElement of the DiagnosticParameter		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.dataElement		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00222

BSW Module	**BSW Context**	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType	
BSW Parameter		**BSW Type**
DcmDspTextTableMapping		EcucParamConfContainerDef
BSW Description		

The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.
Template Description
This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.
M2 Parameter
AsamHdo::ComputationMethod::CompuMethod

Mapping Rule	**Mapping Type**
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00097

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping

BSW Parameter	**BSW Type**
DcmDspDiagnosisRepresentationDataValue	EcucIntegerParamDef
BSW Description	
The data value in the diagnosis representation.	
Template Description	
This represents a textual constant in the computation method.	
M2 Parameter	
AsamHdo::ComputationMethod::CompuConstTextContent.vt	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00098

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping

BSW Parameter	**BSW Type**
DcmDspInternalDataValue	EcucIntegerParamDef
BSW Description	
The ECU internal data value.	
Template Description	

CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit**: This specifies the upper limit of a of the scale.
M2 Parameter
AsamHdo::ComputationMethod::CompuScale.lowerLimit, AsamHdo::ComputationMethod::CompuScale.upperLimit

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00099

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal

BSW Parameter	**BSW Type**
DcmDspRoutineParameterSize	EcucIntegerParamDef

BSW Description
Provide the size of a RoutineControl parameter in bytes
Template Description
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements

Mapping Rule	**Mapping Type**
Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00188

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal

BSW Parameter	**BSW Type**
DcmDspRoutineSignalEndianness	EcucEnumerationParamDef

BSW Description
Defines the endianness of the data belonging to a Routine In Signal for Stop subfunction.
Template Description
This attribute specifies the byte order of the base type.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder

Mapping Rule	**Mapping Type**
baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00183

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal

BSW Parameter	BSW Type
DcmDspRoutineSignalPos	EcucIntegerParamDef
BSW Description	
Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.	
Template Description	
This represents the bitOffset of the DiagnosticParameter	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.bitOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00192

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
BOOLEAN	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is boolean.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request baseTypeEncoding = BOOLEAN baseTypeSize = 1	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00194

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT16	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint16.	
Template Description	

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request baseTypeEncoding = 2C baseTypeSize = 16	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00158

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
SINT16_N	EcucEnumerationLiteralDef

BSW Description
Type of the signal is sint16 array.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request baseTypeEncoding = 2C baseTypeSize = 16 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either doesnot exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full

Mapping Status	Mapping ID
valid	up_Dcm_00158

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT32	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint32.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request baseTypeEncoding = 2C baseTypeSize = 32	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00165

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT32_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request baseTypeEncoding = 2C baseTypeSize = 32 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either doesnot exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00165

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT8	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint8.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request baseTypeEncoding = 2C baseTypeSize = 8	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00247

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT8_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint8 array.	
Template Description	

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request baseTypeEncoding = 2C baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00247

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
UINT16	EcucEnumerationLiteralDef

BSW Description
Type of the signal is uint16.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16	full

Mapping Status	Mapping ID
valid	up_Dcm_00208

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint16 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00208

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT32	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint32.	
Template Description	

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	**Mapping Type**
baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00214

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
UINT32_N	EcucEnumerationLiteralDef

BSW Description
Type of the signal is uint32 array.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements

Mapping Rule	**Mapping Type**
baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00214

BSW Module	**BSW Context**

Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType	
BSW Parameter		**BSW Type**
UINT8		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is uint8.		
Template Description		
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize		
Mapping Rule		**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00171

BSW Module	**BSW Context**	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType	
BSW Parameter		**BSW Type**
UINT8_N		EcucEnumerationLiteralDef
BSW Description		
Type of the signal is uint8 array.		
Template Description		
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.		
M2 Parameter		
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements		
Mapping Rule		**Mapping Type**

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00171

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineIn/DcmDspStopRoutineInSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
VARIABLE_LENGTH	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint8[DcmDspRoutineParameterSize]. This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.request baseTypeEncoding = NONE baseTypeSize = 8 arraySizeSemantics = variableSize maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01002) arraySizeSemantics exists and is set to ArraySizeSemanticsEnum.variableSize (cf. TPS_DEXT_01002)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00218

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine	
BSW Parameter		**BSW Type**
DcmDspStopRoutineOut		EcucParamConfContainerDef
BSW Description		
Provide description of output parameter of Stop subservice for RoutineControl service.		
Template Description		
This represents the response parameters.		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticStopRoutine.response		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00224

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDsp StopRoutineOut	
BSW Parameter		**BSW Type**
DcmDspStopRoutineOutSignal		EcucParamConfContainerDef
BSW Description		
Provide description of a routine signal used in RoutineControl service. The ordering defined via the index attribute of the subcontainers in this list represents the order of the dataOutN elements in the XXX_Stop function call.		
Template Description		
This represents the related dataElement of the DiagnosticParameter		
M2 Parameter		
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.dataElement		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00223

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDsp StopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspArgumentScaling/ DcmDspAlternativeDataType	
BSW Parameter		**BSW Type**
DcmDspTextTableMapping		EcucParamConfContainerDef
BSW Description		
The purpose of the DcmDspTextTableMapping is to associate a texttable value defined in the context of the Dcm to a texttable value defined in the context of a CompuMethod referenced by a DataType that shall be taken to create a dataElement in a SenderReceiverInterface. By this means it is possible to create a primitive version of a TexttableMapping (which can only be applied if a dataElement already exists). In other words, the DcmDspTextTableMapping provides a similar mechanism to the TexttableMapping in a situation where the TexttableMapping cannot be applied since the SenderReceiverInterface for the PortPrototype on the Dcm ServiceComponent does not yet exist.		
Template Description		

This meta-class represents the ability to express the relationship between a physical value and the mathematical representation. Note that this is still independent of the technical implementation in data types. It only specifies the formula how the internal value corresponds to its physical pendant.	
M2 Parameter	
AsamHdo::ComputationMethod::CompuMethod	
Mapping Rule	**Mapping Type**
This mapping applies if the CompuMethod.category is set to values TEXTTABLE or SCALE_LINEAR_AND_TEXTTABLE.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00097

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspDiagnosisRepresentationDataValue	EcucIntegerParamDef
BSW Description	
The data value in the diagnosis representation.	
Template Description	
This represents a textual constant in the computation method.	
M2 Parameter	
AsamHdo::ComputationMethod::CompuConstTextContent.vt	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00098

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspArgumentScaling/DcmDspAlternativeDataType/DcmDspTextTableMapping
BSW Parameter	**BSW Type**
DcmDspInternalDataValue	EcucIntegerParamDef
BSW Description	
The ECU internal data value.	
Template Description	
CompuScale.lowerLimit: This specifies the lower limit of the scale. **CompuScale.upperLimit**: This specifies the upper limit of a of the scale.	
M2 Parameter	
AsamHdo::ComputationMethod::CompuScale.lowerLimit, AsamHdo::ComputationMethod::CompuScale.upperLimit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00099

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal
BSW Parameter	**BSW Type**
DcmDspRoutineParameterSize	EcucIntegerParamDef
BSW Description	
Provide the size of a RoutineControl parameter in bytes	
Template Description	
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
"Only in case of variable length required (according to constr_6008). Calculation: DcmDspRoutineSignalLength = maxNumberOfElements * 8	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00187

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal
BSW Parameter	**BSW Type**
DcmDspRoutineSignalEndianness	EcucEnumerationParamDef
BSW Description	
Defines the endianness of the data belonging to a Routine Out Signal for Stop subfunction.	
Template Description	
This attribute specifies the byte order of the base type.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.byteOrder	
Mapping Rule	**Mapping Type**
baseType.baseTypeDefinition.byteOrder referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00184

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal
BSW Parameter	**BSW Type**
DcmDspRoutineSignalPos	EcucIntegerParamDef
BSW Description	
Provide the position of the signal in the RoutineControl request/response. The position is defined in bits.	
Template Description	
This represents the bitOffset of the DiagnosticParameter	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.bitOffset	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00191

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
BOOLEAN	EcucEnumerationLiteralDef

BSW Description
Type of the signal is boolean.

Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.

M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response baseTypeEncoding = BOOLEAN baseTypeSize = 1	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00195

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT16	EcucEnumerationLiteralDef

BSW Description
Type of the signal is sint16.

Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.

M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response baseTypeEncoding = 2C baseTypeSize = 16	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00159

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT16_N	EcucEnumerationLiteralDef

BSW Description

Type of the signal is sint16 array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding,
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize,
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements,
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics

Mapping Rule	Mapping Type
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response baseTypeEncoding = 2C baseTypeSize = 16 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full

Mapping Status	Mapping ID
valid	up_Dcm_00159

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	BSW Type
SINT32	EcucEnumerationLiteralDef

BSW Description

Type of the signal is sint32.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response baseTypeEncoding = 2C baseTypeSize = 32	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00162

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
SINT32_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is sint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response baseTypeEncoding = 2C baseTypeSize = 32 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00162

BSW Module	BSW Context

Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
SINT8	EcucEnumerationLiteralDef

BSW Description

Type of the signal is sint8.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

M2 Parameter

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding,
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response baseTypeEncoding = 2C baseTypeSize = 8	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00203

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
SINT8_N	EcucEnumerationLiteralDef

BSW Description

Type of the signal is sint8 array.

Template Description

BaseTypeDirectDefinition.baseTypeEncoding:
This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence.

BaseTypeDirectDefinition.baseTypeSize:
Describes the length of the data type specified in the container in bits.

DiagnosticDataElement.maxNumberOfElements:
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.

DiagnosticDataElement.arraySizeSemantics:
This attribute controls the meaning of the value of the array size.

M2 Parameter

AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding,
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize,
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements,
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics

Mapping Rule	**Mapping Type**

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response baseTypeEncoding = 2C baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either doesnot exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00203

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint16.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00209

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT16_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint16 array.	
Template Description	

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response baseTypeEncoding = NONE, UTF-16 baseTypeSize = 16 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00209

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType

BSW Parameter	**BSW Type**
UINT32	EcucEnumerationLiteralDef

BSW Description
Type of the signal is uint32.
Template Description
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.
M2 Parameter
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize

Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32	full

Mapping Status	Mapping ID
valid	up_Dcm_00212

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT32_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint32 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response baseTypeEncoding = NONE, UTF-32 baseTypeSize = 32 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00212

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT8	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint8.	
Template Description	

BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00170

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
UINT8_N	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint8 array.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements	
Mapping Rule	**Mapping Type**

referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response baseTypeEncoding = NONE, WINDOWS-1252, UTF-8, BCD-P, BCD-UP baseTypeSize = 8 maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01001) arraySizeSemantics either does not exist or exists and is set to ArraySizeSemanticsEnum.fixedSize (cf. TPS_DEXT_01001)	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00170

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspRoutine/DcmDspStopRoutine/DcmDspStopRoutineOut/DcmDspStopRoutineOutSignal/DcmDspRoutineSignalType
BSW Parameter	**BSW Type**
VARIABLE_LENGTH	EcucEnumerationLiteralDef
BSW Description	
Type of the signal is uint8[DcmDspRoutineParameterSize]. This is only valid for the last signal and when DcmDspRoutineSignalType is set to VARIABLE_LENGTH.	
Template Description	
BaseTypeDirectDefinition.baseTypeEncoding: This specifies, how an object of the current BaseType is encoded, e.g. in an ECU within a message sequence. **BaseTypeDirectDefinition.baseTypeSize**: Describes the length of the data type specified in the container in bits. **DiagnosticDataElement.maxNumberOfElements**: The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take. **DiagnosticDataElement.arraySizeSemantics**: This attribute controls the meaning of the value of the array size.	
M2 Parameter	
AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeEncoding, AsamHdo::BaseTypes::BaseTypeDirectDefinition.baseTypeSize, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements, DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.arraySizeSemantics	
Mapping Rule	**Mapping Type**
referenced by swDataDefProps of the DiagnosticParameter with the role DiagnosticStopRoutine.response baseTypeEncoding = NONE baseTypeSize = 8 arraySizeSemantics = variableSize maxNumberOfElements exists and value is greater than 0 (cf. TPS_DEXT_01002) arraySizeSemantics exists and is set to ArraySizeSemanticsEnum.variableSize (cf. TPS_DEXT_01002)	full
Mapping Status	**Mapping ID**

valid	up_Dcm_00173

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow	
BSW Parameter		**BSW Type**
DcmDspSecurityADRSize		EcucIntegerParamDef
BSW Description		
Size in bytes of the AccessDataRecord used in GetSeed		
Template Description		
This represents the size of the AccessDataRecord used in GetSeed. Unit:byte.		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticSecurityLevel.accessDataRecordSize		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00241

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow	
BSW Parameter		**BSW Type**
DcmDspSecurityAttemptCounterEnabled		EcucBooleanParamDef
BSW Description		
Allows to enable the external handling of the security attempt counter (e.g. to survive a reset of the ECU).		
Template Description		
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.		
M2 Parameter		
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00293

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow	
BSW Parameter		**BSW Type**
DcmDspSecurityCompareKeyFnc		EcucFunctionNameDef
BSW Description		
Function name to request the result of a key comparison. Parameter is only relevant if DcmDspSecurityUsePort=="USE_ ASYNCH_FNC". This parameter is related to the interface Xxx_CompareKey.		
Template Description		
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.		
M2 Parameter		

DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00240

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**
DcmDspSecurityDelayTime	EcucFloatParamDef
BSW Description	
Delay time after failed security access in seconds. This is started after DcmDspSecurityNumAttDelay number of failed security accesses. min: A negative value is not allowed.	
Template Description	
This represents the delay time after a failed security access. Unit: second.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticSecurityLevel.securityDelayTime	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00238

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**
DcmDspSecurityDelayTimeOnBoot	EcucFloatParamDef
BSW Description	
Value of the delay timer in case of 'power on' in seconds. This delay indicates the time at ECU boot power-on time during which the Dcm does not accept a security access. min: A negative value is not allowed.	
Template Description	
Start delay timer on power on in seconds. This delay indicates the time at ECU boot power-on time where the Dcm remains in the default session and does not accept a security access.	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.securityDelayTimeOnBoot	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00237

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**

DcmDspSecurityGetAttemptCounterFnc	EcucFunctionNameDef
BSW Description	
Function name to request the value of an attempt counter. This parameter is related to the interface Xxx_ GetSecurityAttemptCounter.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00291

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**
DcmDspSecurityGetSeedFnc	EcucFunctionNameDef
BSW Description	
Callout function name used to request a seed. This parameter is related to the interface Xxx_GetSeed.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00239

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**
DcmDspSecurityKeySize	EcucIntegerParamDef
BSW Description	
size of the security key (in Bytes).	
Template Description	
This represents the size of the security key. Unit: byte.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticSecurityLevel.keySize	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00236

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow	
BSW Parameter		**BSW Type**
DcmDspSecurityLevel		EcucIntegerParamDef
BSW Description		
Value of Security level. The locked state cannot be configured explicitly. 1,2,3...63: configuration dependent - Conversion formula to calculate SecurityLevel out of tester requested SecurityAccessType parameter: SecurityLevel = (SecurityAccessType (requestSeed) + 1) / 2 Type: Dcm_SecLevelType		
Template Description		
This would be 0x01, 0x03, 0x05, ... The sendKey id can be computed by adding 1 to the requestSeedId		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticService::SecurityAccess::DiagnosticSecurityAccess.requestSeedId		
Mapping Rule		**Mapping Type**
DcmDspSecurityLevel=(requestSeedId+1)/2		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00235

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow	
BSW Parameter		**BSW Type**
DcmDspSecurityNumAttDelay		EcucIntegerParamDef
BSW Description		
Number of failed security accesses after which the delay time is activated		
Template Description		
This represents the number of failed security accesses after which the delay time is activated.		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticSecurityLevel.numFailedSecurityAccess		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00234

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow	
BSW Parameter		**BSW Type**
DcmDspSecuritySeedSize		EcucIntegerParamDef
BSW Description		
size of the security seed (in Bytes).		
Template Description		
This represents the size of the security seed. Unit: byte.		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticSecurityLevel.seedSize		

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00233

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow
BSW Parameter	**BSW Type**
DcmDspSecuritySetAttemptCounterFnc	EcucFunctionNameDef
BSW Description	
Function name to set the value of an attempt counter. This parameter is related to the interface Xxx_ SetSecurityAttemptCounter.	
Template Description	
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00292

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow/DcmDspSecurityUsePort
BSW Parameter	**BSW Type**
USE_ASYNCH_CLIENT_SERVER	EcucEnumerationLiteralDef
BSW Description	
The DCM will access the data using an R-Port requiring a asynchronous ClientServertInterface SecurityAccess_{SecurityLevel}. The R-Port is described in DcmDspSecurityRow description.	
Template Description	
This represents the ability to point into the component hiearchy (under possible consideration of the rootSoftwareComposition)	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedSwcServiceDependencyInSystem	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00226

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSecurity/DcmDspSecurityRow/DcmDspSecurityUsePort
BSW Parameter	**BSW Type**
USE_ASYNCH_FNC	EcucEnumerationLiteralDef
BSW Description	

The DCM will access the data using the functions that are defined in the parameters DcmDspSecurityGetSeedFnc and DcmDspSecurityCompareKeyFnc as well as the functions defined in DcmDspSecurityGetAttemptCounterFnc and DcmDspSecuritySetAttemptCounterFnc, if enabled by the parameter DcmDspSecurityAttemptCounterEnabled. DCM_E_PENDING return is allowed and OpStatus is existing as IN parameter.
Template Description
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.
M2 Parameter
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00225

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow/DcmDspSessionForBoot

BSW Parameter	**BSW Type**
DCM_NO_BOOT	EcucEnumerationLiteralDef
BSW Description	
This diagnostic session doesn't allow to jump to Bootloader.	
Template Description	
This diagnostic session doesn't allow to jump to Bootloader.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticJumpToBootLoaderEnum.noBoot	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00231

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow/DcmDspSessionForBoot

BSW Parameter	**BSW Type**
DCM_OEM_BOOT	EcucEnumerationLiteralDef
BSW Description	
This diagnostic session allows to jump to OEM Bootloader and bootloader sends final response.	
Template Description	
This diagnostic session allows to jump to OEM Bootloader. In this case the bootloader send the final response.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticJumpToBootLoaderEnum.oemBoot	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00227

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow/DcmDspSessionForBoot
BSW Parameter	**BSW Type**
DCM_OEM_BOOT_RESPAPP	EcucEnumerationLiteralDef
BSW Description	
This diagnostic session allows to jump to OEM Bootloader and application sends final response.	
Template Description	
This diagnostic session allows to jump to OEM Bootloader and application sends final response.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticJumpToBootLoaderEnum.oemBootRespApp	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00228

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow/DcmDspSessionForBoot
BSW Parameter	**BSW Type**
DCM_SYS_BOOT	EcucEnumerationLiteralDef
BSW Description	
This diagnostic session allows to jump to System Supplier Bootloader and bootloader sends final response.	
Template Description	
This diagnostic session allows to jump to System Supplier Bootloader. In this case the bootloader send the final response.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticJumpToBootLoaderEnum.systemSupplierBoot	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00229

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow/DcmDspSessionForBoot
BSW Parameter	**BSW Type**
DCM_SYS_BOOT_RESPAPP	EcucEnumerationLiteralDef
BSW Description	
This diagnostic session allows to jump to System Supplier Bootloader and application sends final response.	
Template Description	
This diagnostic session allows to jump to System Supplier Bootloader and application sends final response.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticJumpToBootLoaderEnum.systemSupplierBootRespApp	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00230

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow	
BSW Parameter	**BSW Type**	
DcmDspSessionLevel	EcucIntegerParamDef	
BSW Description		
subFunction value of the DiagnosticSession. 0, 127 and all values above 127 are reserved by ISO		
Template Description		
This is the numerical identifier used to identify the DiagnosticSession in the scope of diagnostic workflow		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticSession.id		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00232

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow	
BSW Parameter	**BSW Type**	
DcmDspSessionP2ServerMax	EcucFloatParamDef	
BSW Description		
This is the session value for P2ServerMax in seconds (per Session). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds. DCM configuration tools must convert this float value to the appropriate value format for the use in the software implementation of DCM. This value is reported to the tester within the response to the 'Session Control' service.		
Template Description		
This is the session value for P2ServerMax in seconds (per Session Control). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds.		
M2 Parameter		
DiagnosticExtract::Dcm::DiagnosticSession.p2ServerMax		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dcm_00251

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspSession/DcmDspSessionRow	
BSW Parameter	**BSW Type**	
DcmDspSessionP2StarServerMax	EcucFloatParamDef	
BSW Description		
This is the session value for P2*ServerMax in seconds (per Session). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds. DCM configuration tools must convert this float value to the appropriate value format for the use in the software implementation of DCM. This value is reported to the tester within the response to the 'Session Control' service.		
Template Description		

This is the session value for P2*ServerMax in seconds (per Session Control). The AUTOSAR configuration standard is to use SI units, so this parameter is defined as a float value in seconds.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticSession.p2StarServerMax	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00252

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp
BSW Parameter	**BSW Type**
DcmDspVehInfo	EcucParamConfContainerDef
BSW Description	
This container contains the configuration (parameters) for one single VehicleInfoType of service $09	
Template Description	
This meta-class represents the ability to model an instance of the OBD mode 0x09 service.	
M2 Parameter	
DiagnosticExtract::Dcm::ObdService::Mode_0x09_RequestVehicleInformation::DiagnosticRequestVehicleInfo	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo
BSW Parameter	**BSW Type**
DcmDspVehInfoData	EcucParamConfContainerDef
BSW Description	
Data Item of an InfoType; ShortName is post-fix of the port interface name.	
Template Description	
This meta-class represents the ability to describe a concrete piece of data to be taken into account for diagnostic purposes.	
M2 Parameter	
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement	
Mapping Rule	**Mapping Type**
The value shall be taken from DiagnosticRequestVehicleInfo.infoType.dataElement.dataElement.	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData
BSW Parameter	**BSW Type**
DcmDspVehInfoDataOrder	EcucIntegerParamDef
BSW Description	

Defines the order of the data item in the InfoType; values: 0..255; first data item having the order number 0; the next 1 and so on. The configuration of order needs to be unique per InfoType.
Template Description
This represents the bitOffset of the DiagnosticParameter
M2 Parameter
DiagnosticExtract::CommonDiagnostics::DiagnosticParameter.bitOffset

Mapping Rule	**Mapping Type**
Value shall be taken from DiagnosticRequestVehicleInfo.infoType.dataElement.bitOffset.	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData

BSW Parameter	**BSW Type**
DcmDspVehInfoDataReadFnc	EcucFunctionNameDef

BSW Description
Callout function name for reading InfoType data item. Only required in case parameter 'DcmDspVehInfoDataUsePort' is set to 'false'
Template Description
This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference.
M2 Parameter
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency

Mapping Rule	**Mapping Type**
The BswServiceDependency should have aRoleBasedBswModuleEntryAssignment that in turn has attribute role set to Xxx_GetInfotypeValueData and points to a BswModuleEntry.	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData

BSW Parameter	**BSW Type**
DcmDspVehInfoDataSize	EcucIntegerParamDef

BSW Description
Size in bytes of the InfoType data item.
Template Description
The existence of this attribute turns the data instance into an array of data. The attribute determines the size of the array in terms of how many elements the array can take.
M2 Parameter
DiagnosticExtract::CommonDiagnostics::DiagnosticDataElement.maxNumberOfElements

Mapping Rule	**Mapping Type**
Value shall be taken from DiagnosticRequestVehicleInfo.infoType.dataElement.dataElement.maxNumberOfElements.	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo/DcmDspVehInfoData	
BSW Parameter		**BSW Type**
DcmDspVehInfoDataUsePort		EcucBooleanParamDef
BSW Description		
When this parameter is set to true the DCM will access the Data using an R-Port requiring a PortInterface IInfotypeServices_{VehInfoData}. The R-Port is named InfotypeServices_{VehInfoData} where {VEHINFODATA} is the name of the container DcmDspVehInfoData. In that case, the DcmDspVehInfoDataReadFnc is ignored and the RTE APIs are used. When this parameter is set to false, the DCM calls the function defined in DcmDspVehInfoDataReadFnc.		
Template Description		
DiagnosticServiceSwMapping.mappedBswServiceDependency: This is supposed to represent a reference to a BswServiceDependency. the latter is not derived from Referrable and therefore this detour needs to be implemented to still let BswServiceDependency become the target of a reference. **DiagnosticServiceSwMapping.mappedSwcServiceDependencyInSystem**: This represents the ability to point into the component hiearchy (under possible consideration of the rootSoftwareComposition)		
M2 Parameter		
DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedBswServiceDependency, DiagnosticExtract::ServiceMapping::DiagnosticServiceSwMapping.mappedSwcServiceDependencyInSystem		
Mapping Rule		**Mapping Type**
Shall be set to TRUE if the reference DiagnosticServiceSwMapping.mappedSwcServiceDependency exists. Shall be set to FALSE if the reference DiagnosticServiceSwMapping.mappedBswServiceDependency exists.		full
Mapping Status		**Mapping ID**
valid		

BSW Module	BSW Context	
Dcm	Dcm/DcmConfigSet/DcmDsp/DcmDspVehInfo	
BSW Parameter		**BSW Type**
DcmDspVehInfoInfoType		EcucIntegerParamDef
BSW Description		
value of InfoType. Within each DcmConfigSet all DcmDspVehInfoInfoType values shall be unique.		
Template Description		
ObdInfoServiceNeeds.infoType: The InfoType according to ISO 15031-5 **DiagnosticInfoType.id**: This attribute represents the value of InfoType (see SAE J1979-DA).		
M2 Parameter		
CommonStructure::ServiceNeeds::ObdInfoServiceNeeds.infoType, DiagnosticExtract::CommonDiagnostics::DiagnosticInfoType.id		
Mapping Rule		**Mapping Type**

If DiagnosticRequestVehicleInfo, us DiagnosticRequestVehicleInfo.infoType.id.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00029

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions
BSW Parameter	**BSW Type**
DcmModeCondition	EcucParamConfContainerDef
BSW Description	
This container contains the configuration of a mode condition or an environmental conditions which can be used as argument in DcmModeRules. One DcmModeCondition shall contain either one DcmSwcModeRef or one DcmBswModeRef or one DcmSwcSRDataElementRef. Please note that the Dcm acts as well as mode manager. Therefore the references DcmSwcModeRef or one DcmBswModeRef. might point to provided ModeDeclarationGroupPrototypes of the Dcm itself as well as to provided ModeDeclarationGroupPrototypes of other Bsw Modules or software components. In case of a configured DcmSwcModeRef or DcmBswModeRef only the DcmConditionType DCM_EQUALS or DCM_EQUALS_NOT are applicable. In case of DcmSwcSRDataElementRef all literals of DcmConditionType are possible.	
Template Description	
DiagnosticCompareConditions are atomic conditions. They are based on the idea of a comparison at runtime of some variable data with something constant. The type of the comparison (==, !=, <, <=, ...) is specified in DiagnosticCompareCondition.compareType.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvCompareCondition	
Mapping Rule	**Mapping Type**
Depending on the reference a DcmModeCondition is mapped to a DiagnosticEnvModeCondition if only one reference is present and reference is a DcmSwcModeRef or a DcmBswModeRef. If two references are present, a DcmSwcSRDataElementRef and a DcmSwcSRDataElementValueRef, then DcmModeCondition is mapped to a DiagnosticEnvDataCondition.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00271

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition
BSW Parameter	**BSW Type**
DcmBswModeRef	EcucInstanceReferenceDef
BSW Description	
This parameter references a mode of a ModeDeclarationGroupPrototype provided by a Basic Software Module used for the condition. Please note that such ModeDeclarationGroupPrototype are owned by a Basic Software Module Description in the role providedModeGroup.	
Template Description	

This reference represents both the ModeDeclarationGroupPrototype and the ModeDeclaration relevant for the mode comparison.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvModeCondition.modeElement	
Mapping Rule	**Mapping Type**
For DcmModeRef a new DiagnosticEnvBswModeElement is used, pointing to the ModeDeclaration via ModeInModuleDescriptionInstanceRef. This new DiagnosticEnvModeElement shall be aggregated by the same DiagnosticEnvironmentalConfition as the DiagnosticEnvModeCondition, in which the target of the reference DiagnosticEnvModeCondition.modeElement points to the this DiagnosticEnvModeElement.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00273

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType
BSW Parameter	**BSW Type**
DCM_EQUALS	EcucEnumerationLiteralDef
BSW Description	
Template Description	
DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticCompareTypeEnum.isEqual**: equal	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isEqual	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00265

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType
BSW Parameter	**BSW Type**
DCM_EQUALS_NOT	EcucEnumerationLiteralDef
BSW Description	
Template Description	

DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticCompareTypeEnum.isNotEqual**: not equal
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isNotEqual

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00266

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType

BSW Parameter	**BSW Type**
DCM_GREATER_OR_EQUAL	EcucEnumerationLiteralDef

BSW Description

Template Description
DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticCompareTypeEnum.isGreaterOrEqual**: greater than or equal
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isGreaterOrEqual

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00269

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType

BSW Parameter	**BSW Type**
DCM_GREATER_THAN	EcucEnumerationLiteralDef

BSW Description

Template Description

DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticCompareTypeEnum.isGreaterThan**: greater than
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isGreaterThan

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00267

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType

BSW Parameter	BSW Type
DCM_LESS_OR_EQUAL	EcucEnumerationLiteralDef

BSW Description

Template Description
DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticCompareTypeEnum.isLessOrEqual**: less than or equal
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isLessOrEqual

Mapping Rule	Mapping Type
	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00270

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition/DcmConditionType

BSW Parameter	BSW Type
DCM_LESS_THAN	EcucEnumerationLiteralDef

BSW Description

Template Description

DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticCompareTypeEnum.isLessThan**: less than
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isLessThan

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00268

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition

BSW Parameter	**BSW Type**
DcmSwcModeRef	EcucInstanceReferenceDef

BSW Description
This parameter references a mode in a particular mode request port of a software component that is used for the condition.
Template Description
This reference represents both the ModeDeclarationGroupPrototype and the ModeDeclaration relevant for the mode comparison.
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvModeCondition.modeElement

Mapping Rule	**Mapping Type**
For DcmModeRef a new DiagnosticEnvSwcModeElement is used, pointing to the ModeDeclaration via PModeInSystemInstanceRef. This new Diagnostic EnvModeElement shall be aggregated by the same DiagnosticEnvironmental Confition as the DiagnosticEnvModeCondition, in which the target of the reference DiagnosticEnvModeCondition.modeElement points to the this Diagnostic EnvModeElement.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00272

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition

BSW Parameter	**BSW Type**
DcmSwcSRDataElementRef	EcucChoiceReferenceDef

BSW Description
Reference to environmental conditions. It is possible to reference a S/R Receiver-Port to read physical values and compare (equal, greater, less,...) them with a configured value that is defined by DcmSwcSRDataElementValueRef.
Template Description
This reference represents the related diagnostic data element.
M2 Parameter

DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvDataCondition.dataElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00274

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeCondition
BSW Parameter	**BSW Type**
DcmSwcSRDataElementValueRef	EcucForeignReferenceDef
BSW Description	
Reference to a constant specification defining the compare value for environmental condition.	
Template Description	
This attribute represents a fixed compare value taken to evaluate the compare condition.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvDataCondition.compareValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00275

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions
BSW Parameter	**BSW Type**
DcmModeRule	EcucParamConfContainerDef
BSW Description	
This container contains the configuration of a mode rule which represents a logical expression with DcmModeConditions or other DcmModeRules as arguments. All arguments are processed with the operator defined by DcmLogicalOperator, for instance: Argument_A AND Argument_B AND Argument_C	
Template Description	
A DiagnosticEnvConditionFormula embodies the computation instruction that is to be evaluated at runtime to determine if the DiagnosticEnvironmentalCondition is currently present (i.e. the formula is evaluated to true) or not (otherwise). The formula itself consists of parts which are combined by the logical operations specified by DiagnosticEnvConditionFormula.op. If a diagnostic functionality cannot be executed because an environmental condition fails then the diagnostic stack shall send a negative response code (NRC) back to the client. The value of the NRC is directly related to the specific formula and is therefore formalized in the attribute DiagnosticEnvConditionFormula.nrcValue.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula	
Mapping Rule	**Mapping Type**

A DcmModeRule is mapped to a DiagnosticEnvConditionFormula, if this DcmModeRule is a subrule, i.e. it is referenced by a DcmArgumentRef. In addition, a new DiagnosticEnvironmentalCondition shall be created with DiagnosticEnvironmentalCondition.formula containing a DiagnosticEnvConditionFormula. In both cases, if no DcmLogicalOperator is present in this DcmModeRule, then DiagnosticEnvConditionFormula shall be set to DiagnosticLogicalOperatorEnum.logicalAnd.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00259

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeRule
BSW Parameter	**BSW Type**
DcmArgumentRef	EcucChoiceReferenceDef
BSW Description	
This is a choice reference either to a mode condition or a an other mode rule serving as subexpression.	
Template Description	
A DiagnosticEnvConditionFormulaPart can either be a atomic condition, e.g. a DiagnosticEnvCompareCondition, or a DiagnosticEnvConditionFormula, again, which allows arbitrary nesting.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvCondition FormulaPart	
Mapping Rule	**Mapping Type**
Depending on the destination, one DcmArgumentRef is mapped to a Diagnostic EnvConditionFormula if "destination" is a DcmModeRule, and to a Diagnostic EnvCompareCondition, if "destination" is a DcmModeCondition. The order of the aggregation of the DiagnosticEnvConditionFormulaParts shall correspond to the ordering of the index of the DcmArgumentRefs.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00264

BSW Module	**BSW Context**
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeRule/DcmLogical Operator
BSW Parameter	**BSW Type**
DCM_AND	EcucEnumerationLiteralDef
BSW Description	
Template Description	
DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticLogicalOperatorEnum.logicalAnd**: Logical AND	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticLogicalOperator Enum.logicalAnd	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00261

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeRule/DcmLogical Operator
BSW Parameter	**BSW Type**
DCM_OR	EcucEnumerationLiteralDef
BSW Description	
Template Description	
DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticLogicalOperatorEnum.logicalOr**: Logical OR	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticLogicalOperatorEnum.logicalOr	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00262

BSW Module	BSW Context
Dcm	Dcm/DcmConfigSet/DcmProcessingConditions/DcmModeRule
BSW Parameter	**BSW Type**
DcmModeRuleNrcValue	EcucIntegerParamDef
BSW Description	
Optional parameter which defines the NRC to be sent in case the mode rule condition is not valid.	
Template Description	
This attribute represents the concrete NRC value that shall be returned if the condition fails.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.nrcValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00263

BSW Module	BSW Context
Dcm	Dcm/DcmGeneral
BSW Parameter	**BSW Type**
DcmDDDIDStorage	EcucBooleanParamDef
BSW Description	

This configuration switch defines, whether DDDID definition is stored non-volatile or not. true: DDDID are stored non-volatile false: DDDID are only maintained volatile
Template Description
This configuration switch defines whether DDDID definition is handled as non-volatile information or not.
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::DynamicallyDefineDataIdentifier::DiagnosticDynamicallyDefineDataIdentifierClass.configurationHandling

Mapping Rule	**Mapping Type**
volatile -> false, nonVolatile -> true	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00253

BSW Module	**BSW Context**
Dcm	Dcm/DcmGeneral

BSW Parameter	**BSW Type**
DcmRespondAllRequest	EcucBooleanParamDef

BSW Description
If set to FALSE the Dcm will not respond to diagnostic request that contains a service ID which is in the range from 0x40 to 0x7F or in the range from 0xC0 to 0xFF (Response IDs).
Template Description
If set to FALSE the DCM will not respond to diagnostic request that contains a service ID which is in the range from 0x40 to 0x7F or in the range from 0xC0 to 0xFF (Response IDs).
M2 Parameter
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.responseOnAllRequestSids

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00249

BSW Module	**BSW Context**
Dcm	Dcm/DcmGeneral

BSW Parameter	**BSW Type**
DcmVinRef	EcucReferenceDef

BSW Description
Reference to the Did containing the VIN Information. This parameter is needed for function Dcm_GetVin
Template Description
This meta-class represents the ability to model a diagnostic data identifier (DID) that is fully specified regarding the payload at configuration-time.
M2 Parameter
DiagnosticExtract::CommonDiagnostics::DiagnosticDataIdentifier

Mapping Rule	**Mapping Type**
This reference shall only apply to a DiagnosticDataIdentifier where the attribute representsVin is set to true.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00248

E.3 Dem

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDTC	
BSW Parameter		**BSW Type**
DemDTCFunctionalUnit		EcucIntegerParamDef
BSW Description		
DTCFuncitonalUnit is a 1-byte value which identifies the corresponding basic vehicle / system function which reports the DTC. This parameter is necessary for the report of severity information. If this parameter is configured for no DTC, the Dem provides no DTC functional unit information.		
Template Description		
This attribute specifies a 1-byte value which identifies the corresponding basic vehicle / system function which reports the DTC. This parameter is necessary for the report of severity information.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticTroubleCodeUds.functionalUnit		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00081

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDTC	
BSW Parameter		**BSW Type**
DemDTCSeverity		EcucEnumerationParamDef
BSW Description		
DTC severity according to ISO 14229-1. This parameter depends on the automotive manufacturer. If it is not configured, the value is counted as 'no severity'. If this parameter is configured for no DTC, the Dem provides no DTC severity information.		
Template Description		
DTC severity according to ISO 14229-1.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticTroubleCodeUds.severity		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00082

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDTC	
BSW Parameter		**BSW Type**
DemWWHOBDDTCClass		EcucEnumerationParamDef
BSW Description		
DTC Class according to ISO 14229-1 [2013 version]. This parameter depends on the automotive manufacturer. If it is not configured, the value is marked as 'unclassified'. If this parameter is configured for no DTC, the Dem provides no DTC WWHOBD class information.		
Template Description		
DTC severity according to ISO 14229-1.		

M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticTroubleCodeUds.severity	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00083

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTCAttributes
BSW Parameter	**BSW Type**
DemAgingAllowed	EcucBooleanParamDef
BSW Description	
Switch to allow aging/unlearning of the event or not. true: aging allowed false: aging not allowed	
Template Description	
This represents the decision whether aging is allowed for this DiagnosticEvent.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticEvent.agingAllowed	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00069

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTCAttributes
BSW Parameter	**BSW Type**
DemAgingCycleCounterThreshold	EcucIntegerParamDef
BSW Description	
Number of aging cycles needed to unlearn/delete the event.	
Template Description	
Number of aging cycles needed to unlearn/delete the event.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticAging::DiagnosticAging.threshold	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00072

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTCAttributes
BSW Parameter	**BSW Type**
DemAgingCycleRef	EcucReferenceDef
BSW Description	
Reference to the cycle which is triggering the aging of the event.	
Template Description	
This represents the applicable aging cycle.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticAging::DiagnosticAging.agingCycle	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00073

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTCAttributes
BSW Parameter	**BSW Type**
DemDTCPriority	EcucIntegerParamDef
BSW Description	
Priority of the event/dtc, in view of full event memory. A lower value means higher priority.	
Template Description	
Priority of the event, in view of full event buffer. A lower value means higher priority.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticTroubleCodeProps.priority	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00099

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTCAttributes
BSW Parameter	**BSW Type**
DemDTCSignificance	EcucEnumerationParamDef
BSW Description	
Significance of the event, which indicates additional information concerning fault classification and resolution. It can be mapped as Dem-internal data element. It shall be configured, if it is a part of event related data.	
Template Description	
Significance of the event, which indicates additional information concerning fault classification and resolution.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticTroubleCodeProps.significance	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00091

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTCAttributes/DemDTCSignificance
BSW Parameter	**BSW Type**
DEM_EVENT_SIGNIFICANCE_OCCURRENCE	EcucEnumerationLiteralDef
BSW Description	
issue, which indicates additional information concerning insufficient system behavior	
Template Description	
Issue, which indicates additional information concerning insufficient system behavior.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticSignificanceEnum.occurence	

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00042

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTCAttributes
BSW Parameter	**BSW Type**
DemMaxNumberFreezeFrameRecords	EcucIntegerParamDef
BSW Description	
This parameter defines the number of according freeze frame records, which can maximal be stored for this event. Therefore all these freeze frame records have the same freeze frame class. This parameter is only required for calculated record numeration (refer to DemTypeOfFreeze-FrameRecordNumeration).	
Template Description	
This attribute defines the number of according freeze frame records, which can maximal be stored for this event. Therefore all these freeze frame records have the same freeze frame class.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticTroubleCodeProps.maxNumberFreeze FrameRecords	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00106

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDTCAttributes
BSW Parameter	**BSW Type**
DemMemoryDestinationRef	EcucChoiceReferenceDef
BSW Description	
The memory destination assigns DTCs to one or two memory destinations. If more than one memory destination is assigned to a specific DTC, the DTC can be present in the corresponding event memories. In this case one of the references has to be DemMirrorMemory (SWS_Dem_CONSTR_6104).	
Template Description	
The event destination assigns events to none, one or multiple origins.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticTroubleCodeProps.memoryDestination	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00088

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass
BSW Parameter	**BSW Type**
DemCounterBasedFdcThresholdStorageValue	EcucIntegerParamDef
BSW Description	
Threshold to allocate an event memory entry and to capture the Freeze Frame.	
Template Description	

Threshold to allocate an event memory entry and to capture the Freeze Frame.	
M2 Parameter	
CommonStructure::ServiceNeeds::DiagEventDebounceCounterBased.counterBasedFdcThreshold StorageValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00097

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass
BSW Parameter	**BSW Type**
DemDebounceBehavior	EcucEnumerationParamDef
BSW Description	
This parameter defines how the event debounce algorithm will behave, if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled.	
Template Description	
This attribute defines how the event debounce algorithm will behave, if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticDebouncingAlgorithm::DiagnosticDebounceAlgorithm Props.debounceBehavior	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00101

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass/DemDebounceBehavior
BSW Parameter	**BSW Type**
DEM_DEBOUNCE_FREEZE	EcucEnumerationLiteralDef
BSW Description	
The event debounce counter will be frozen with the current value and will not change while a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. After all related enable conditions are fulfilled and ControlDTCSetting of the related event is enabled again, the event qualification will continue with the next report of the event (i.e. SetEventStatus).	
Template Description	
The event debounce counter will be frozen with the current value and will not change while a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. After all related enable conditions are fulfilled and ControlDTCSetting of the related event is enabled again, the event qualification will continue with the next report of the event (i.e. SetEventStatus).	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticDebouncingAlgorithm::DiagnosticDebounceBehavior Enum.freeze	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00066

BSW Module	**BSW Context**

Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass/DemDebounceBehavior	
BSW Parameter		**BSW Type**
DEM_DEBOUNCE_RESET		EcucEnumerationLiteralDef
BSW Description		
The event debounce counter will be reset to initial value if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. The qualification of the event will be restarted with the next valid event report.		
Template Description		
The event debounce counter will be reset to initial value if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. The qualification of the event will be restarted with the next valid event report.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticDebouncingAlgorithm::DiagnosticDebounceBehavior Enum.reset		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00068

BSW Module	**BSW Context**	
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass	
BSW Parameter		**BSW Type**
DemDebounceCounterDecrementStepSize		EcucIntegerParamDef
BSW Description		
Defines the step size for decrementation of the internal debounce counter (PREPASSED).		
Template Description		
This value shall be taken to decrement the internal debounce counter.		
M2 Parameter		
CommonStructure::ServiceNeeds::DiagEventDebounceCounterBased.counterDecrementStepSize		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00028

BSW Module	**BSW Context**	
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass	
BSW Parameter		**BSW Type**
DemDebounceCounterFailedThreshold		EcucIntegerParamDef
BSW Description		
Defines the value of the internal debounce counter, which indicates the failed status.		
Template Description		
This value defines the event-specific limit that indicates the "failed" counter status.		
M2 Parameter		
CommonStructure::ServiceNeeds::DiagEventDebounceCounterBased.counterFailedThreshold		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00015

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass	
BSW Parameter		**BSW Type**
DemDebounceCounterIncrementStepSize		EcucIntegerParamDef
BSW Description		
Defines the step size for incrementation of the internal debounce counter (PREFAILED).		
Template Description		
This value shall be taken to increment the internal debounce counter.		
M2 Parameter		
CommonStructure::ServiceNeeds::DiagEventDebounceCounterBased.counterIncrementStepSize		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00016

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass	
BSW Parameter		**BSW Type**
DemDebounceCounterJumpDown		EcucBooleanParamDef
BSW Description		
Switch for the activation of Jump-Down. true: Jump-Down activated false: Jump-Down deactivated		
Template Description		
This value activates or deactivates the counter jump-down behavior.		
M2 Parameter		
CommonStructure::ServiceNeeds::DiagEventDebounceCounterBased.counterJumpDown		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00018

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass	
BSW Parameter		**BSW Type**
DemDebounceCounterJumpDownValue		EcucIntegerParamDef
BSW Description		
Jump-Down value of the internal debounce counter which is taken as initialization value for the counter when the respective step-down occurs.		
Template Description		
This value represents the initial value of the internal debounce counter if the counting direction changes from incrementing to decrementing.		
M2 Parameter		
CommonStructure::ServiceNeeds::DiagEventDebounceCounterBased.counterJumpDownValue		
Mapping Rule		**Mapping Type**
		full
Mapping Status		**Mapping ID**
valid		up_Dem_00017

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass	
BSW Parameter		**BSW Type**
DemDebounceCounterJumpUp		EcucBooleanParamDef
BSW Description		
Switch for the activation of Jump-Up. true: Jump-Up activated false: Jump-Up deactivated		
Template Description		
This value activates or deactivates the counter jump-up behavior.		
M2 Parameter		
CommonStructure::ServiceNeeds::DiagEventDebounceCounterBased.counterJumpUp		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00019

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass	
BSW Parameter		**BSW Type**
DemDebounceCounterJumpUpValue		EcucIntegerParamDef
BSW Description		
Jump-Up value of the internal debounce counter which is taken as initialization value for the counter when the respective step-up occurs.		
Template Description		
This value represents the initial value of the internal debounce counter if the counting direction changes from decrementing to incrementing.		
M2 Parameter		
CommonStructure::ServiceNeeds::DiagEventDebounceCounterBased.counterJumpUpValue		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00020

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass	
BSW Parameter		**BSW Type**
DemDebounceCounterPassedThreshold		EcucIntegerParamDef
BSW Description		
Defines the value of the internal debounce counter, which indicates the passed status.		
Template Description		
This value defines the event-specific limit that indicates the "passed" counter status.		
M2 Parameter		
CommonStructure::ServiceNeeds::DiagEventDebounceCounterBased.counterPassedThreshold		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00021

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDebounceCounterBasedClass	
BSW Parameter		**BSW Type**
DemDebounceCounterStorage		EcucBooleanParamDef
BSW Description		
Switch to store the debounce counter value non-volatile or not. true: debounce counter value shall be stored non-volatile false: debounce counter value is volatile		
Template Description		
Switch to store the debounce counter value non-volatile or not. true: debounce counter value shall be stored non-volatile false: debounce counter value is volatile		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticDebouncingAlgorithm::DiagnosticDebounceAlgorithmProps.debounceCounterStorage		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00108

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDebounceTimeBaseClass	
BSW Parameter		**BSW Type**
DemDebounceBehavior		EcucEnumerationParamDef
BSW Description		
This parameter defines how the event debounce algorithm will behave, if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled.		
Template Description		
This attribute defines how the event debounce algorithm will behave, if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticDebouncingAlgorithm::DiagnosticDebounceAlgorithmProps.debounceBehavior		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00100

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDebounceTimeBaseClass/DemDebounceBehavior	
BSW Parameter		**BSW Type**
DEM_DEBOUNCE_FREEZE		EcucEnumerationLiteralDef
BSW Description		
The event debounce timer will be frozen with the current value and will not change while a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. After all related enable conditions are fulfilled and ControlDTCSetting of the related event is enabled again, the event qualification will continue with the next report of the event (i.e. SetEventStatus).		
Template Description		

The event debounce counter will be frozen with the current value and will not change while a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. After all related enable conditions are fulfilled and ControlDTCSetting of the related event is enabled again, the event qualification will continue with the next report of the event (i.e. SetEventStatus).
M2 Parameter
DiagnosticExtract::Dem::DiagnosticDebouncingAlgorithm::DiagnosticDebounceBehavior Enum.freeze

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00065

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDebounceTimeBaseClass/DemDebounceBehavior

BSW Parameter	**BSW Type**
DEM_DEBOUNCE_RESET	EcucEnumerationLiteralDef

BSW Description
The event debounce timer will be reset to initial value if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. The qualification of the event will be restarted with the next valid event report.
Template Description
The event debounce counter will be reset to initial value if a related enable condition is not fulfilled or ControlDTCSetting of the related event is disabled. The qualification of the event will be restarted with the next valid event report.
M2 Parameter
DiagnosticExtract::Dem::DiagnosticDebouncingAlgorithm::DiagnosticDebounceBehavior Enum.reset

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00067

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemDebounceTimeBaseClass

BSW Parameter	**BSW Type**
DemDebounceTimeFailedThreshold	EcucFloatParamDef

BSW Description
Defines the time out duration for "Event Failed" qualification. The AUTOSAR configuration standard is to use SI units, so this parameter is defined as float value in seconds. Dem configuration tools must convert this float value to the appropriate value format for the use in the software implementation of Dem.
Template Description
This value represents the event-specific delay indicating the "failed" status.
M2 Parameter
CommonStructure::ServiceNeeds::DiagEventDebounceTimeBased.timeFailedThreshold

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00118

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDebounceTimeBaseClass	
BSW Parameter		**BSW Type**
DemDebounceTimePassedThreshold		EcucFloatParamDef
BSW Description		
Defines the time out duration for "Event Passed" qualification. The AUTOSAR configuration standard is to use SI units, so this parameter is defined as float value in seconds. Dem configuration tools must convert this float value to the appropriate value format for the use in the software implementation of Dem.		
Template Description		
This value represents the event-specific delay indicating the "passed" status.		
M2 Parameter		
CommonStructure::ServiceNeeds::DiagEventDebounceTimeBased.timePassedThreshold		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00119

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDebounceTimeBaseClass	
BSW Parameter		**BSW Type**
DemTimeBasedFdcThresholdStorageValue		EcucFloatParamDef
BSW Description		
Threshold to allocate an event memory entry and to capture the Freeze Frame.		
Template Description		
Threshold to allocate an event memory entry and to capture the Freeze Frame.		
M2 Parameter		
CommonStructure::ServiceNeeds::DiagEventDebounceTimeBased.timeBasedFdcThresholdStorageValue		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00098

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDtrs/DemDtr	
BSW Parameter		**BSW Type**
DemDtrEventRef		EcucReferenceDef
BSW Description		
Reference to the DemEventParameter this DTR is related to. If the related event is not configured, the Dem cannot ensure consistency between the DTR and the event.		
Template Description		
This attribute represents the diagnostic event that is related to the diagnostic test result.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticTestResult::DiagnosticTestResult.event		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDtrs/DemDtr	
BSW Parameter		**BSW Type**
DemDtrId		EcucIntegerParamDef
BSW Description		
The index identifier value assigned to this DTR. The value is generated during the Dem configuration process.		
Template Description		
This represents the numerical id associated with the diagnostic test identifier.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticTestResult::DiagnosticTestIdentifier.id		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDtrs/DemDtr	
BSW Parameter		**BSW Type**
DemDtrMid		EcucIntegerParamDef
BSW Description		
The OBDMID of the DTR. The values 0x00, 0x20, 0x40, 0x60, 0x80, 0xA0, 0xC0, 0xE0 are reserved.		
Template Description		
This represents the numerical measurement Id		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticTestResult::DiagnosticMeasurementIdentifier.obdMid		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemDtrs/DemDtr	
BSW Parameter		**BSW Type**
DemDtrTid		EcucIntegerParamDef
BSW Description		
The OBDTID of the DTR.		
Template Description		
This represents the numerical id associated with the diagnostic test identifier.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticTestResult::DiagnosticTestIdentifier.id		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDtrs/DemDtr

BSW Parameter	BSW Type
DemDtrUasid	EcucIntegerParamDef
BSW Description	
The UaSId the DTR data shall be scaled to, and reported together with the rescaled DTR data.	
Template Description	
This represents the unit and scaling Id of the diagnostic test result.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTestResult::DiagnosticTestIdentifier.uasId	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDtrs/DemDtr
BSW Parameter	**BSW Type**
DemDtrUpdateKind	EcucEnumerationParamDef
BSW Description	
Update conditions applied by the Dem to reports of DTR values. Only supported if a related Event is configured	
Template Description	
This attribute controls the update behavior of the enclosing DiagnosticTestResult.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTestResult::DiagnosticTestResult.updateKind	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDtrs/DemDtr/DemDtrUpdateKind
BSW Parameter	**BSW Type**
DEM_DTR_UPDATE_ALWAYS	EcucEnumerationLiteralDef
BSW Description	
Any DTR result reported by the monitor is used by the Dem.	
Template Description	
Any DTR result reported by the monitor is used by the Dem.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTestResult::DiagnosticTestResultUpdateEnum.always	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemDtrs/DemDtr/DemDtrUpdateKind
BSW Parameter	**BSW Type**
DEM_DTR_UPDATE_STEADY	EcucEnumerationLiteralDef
BSW Description	

The Dem accepts reported DTRs only when the configured debouncing mechanism is stable at the FAIL or PASS limit.
Template Description
The Dem accepts reported DTRs only when the configured debouncing mechanism is stable at the FAIL or PASS limit.
M2 Parameter
DiagnosticExtract::Dem::DiagnosticTestResult::DiagnosticTestResultUpdateEnum.steady

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemEventParameter/DemCallbackClearEventAllowed

BSW Parameter	**BSW Type**
DemCallbackClearEventAllowedFnc	EcucFunctionNameDef

BSW Description
Function name of prototype "ClearEventAllowed".
Template Description
This attribute defines whether the Dem has access to a "ClearEventAllowed" callback.
M2 Parameter
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticEvent.eventClearAllowed

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00074

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemEventParameter/DemCallbackClearEventAllowed

BSW Parameter	**BSW Type**
DemClearEventAllowedBehavior	EcucEnumerationParamDef

BSW Description
Defines the resulting UDS status byte for the related event, which must not be cleared according to the ClearEventAllowed callback.
Template Description
This attribute defines the resulting UDS status byte for the related event, which shall not be cleared according to the ClearEventAllowed callback.
M2 Parameter
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticEvent.clearEventBehavior

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00136

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemEventParameter/DemCallbackClearEventAllowed/ DemClearEventAllowedBehavior

BSW Parameter	**BSW Type**
DEM_NO_STATUS_BYTE_CHANGE	EcucEnumerationLiteralDef
BSW Description	

The UDS status byte keeps unchanged.	
Template Description	
The event status byte keeps unchanged.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticClearEventBehaviorEnum.noStatusByte Change	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00137

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemEventParameter/DemCallbackClearEventAllowed/ DemClearEventAllowedBehavior
BSW Parameter	**BSW Type**
DEM_ONLY_THIS_CYCLE_AND_READINESS	EcucEnumerationLiteralDef
BSW Description	
The <...>ThisOperationCycle and readiness bits of the UDS status byte are reset.	
Template Description	
The OperationCycle and readiness bits of the event status byte are reset.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticClearEventBehaviorEnum.onlyThisCycleAnd Readiness	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00050

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemEventParameter
BSW Parameter	**BSW Type**
DemDebounceAlgorithmClass	EcucChoiceContainerDef
BSW Description	
Debounce algorithm class: counter based, time based, or monitor internal.	
Template Description	
This class represents the ability to specify the pre-debounce algorithm which is selected and/or required by the particular monitor. This class inherits from Identifiable in order to allow further documentation of the expected or implemented debouncing and to use the category for the identification of the expected / implemented debouncing.	
M2 Parameter	
CommonStructure::ServiceNeeds::DiagEventDebounceAlgorithm	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00022

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass

BSW Parameter	BSW Type
DemDebounceCounterBased	EcucParamConfContainerDef
BSW Description	
This container contains the configuration (parameters) for counter based debouncing.	
Template Description	
This meta-class represents the ability to indicate that the counter-based debounce algorithm shall be used by the DEM for this diagnostic monitor. This is related to set the ECUC choice container DemDebounceAlgorithmClass to DemDebounceCounterBased.	
M2 Parameter	
CommonStructure::ServiceNeeds::DiagEventDebounceCounterBased	
Mapping Rule	**Mapping Type**
There are two ways to derive the existence of DemDebounceCounterBased: 1. DiagEventNeeds,diagEventDebounceAlgoritm exists and is modeled as a DiagEventDebounceCounterBased. 2. DiagnosticContributionSet.commonProperties.debounceAlgorithm Props.debounceAlgorithm exists and is modeled as a DiagEventDebounce CounterBased If both alternatives exist at the same time then the definition ot DiagnosticContributionSet.commonProperties.debounceAlgorithmProps.debounce Algorithm shall be handled with priority.	full
Mapping Status	**Mapping ID**
valid	up_Dem_00014

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass
BSW Parameter	**BSW Type**
DemDebounceMonitorInternal	EcucParamConfContainerDef
BSW Description	
This container contains the configuration (parameters) for monitor internal debouncing.	
Template Description	
This meta-class represents the ability to indicate that the pre-debounce algorithm shall be used by the Dem for this diagnostic monitor. This is related to setting the EcuC choice container DemDebounceAlgorithmClass to DemDebounceMonitorInternal. If the FaultDetectionAlogrithm is already known to be implemented by a specific BswModuleEntry the reference bswModuleEntry points to the function specification. If the FaultDetectionCounter value is accessible at a PortPrototype this PortPrototype shall be referenced by an assignedPort.	
M2 Parameter	
CommonStructure::ServiceNeeds::DiagEventDebounceMonitorInternal	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00023

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemEventParameter/DemDebounceAlgorithmClass	
BSW Parameter		**BSW Type**
DemDebounceTimeBase		EcucParamConfContainerDef
BSW Description		
This container contains the configuration (parameters) for time based debouncing.		
Template Description		
This meta-class represents the ability to indicate that the time-based pre-debounce algorithm shall be used by the Dem for this diagnostic monitor. This is related to set the EcuC choice container DemDebounceAlgorithmClass to DemDebounceTimeBase.		
M2 Parameter		
CommonStructure::ServiceNeeds::DiagEventDebounceTimeBased		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00024

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemEventParameter	
BSW Parameter		**BSW Type**
DemEventConfirmationThreshold		EcucIntegerParamDef
BSW Description		
Defines the operation cycle threshold of the DTC confirmation status according "Confirmation Threshold" of ISO 14229-1.		
Template Description		
This attribute defines the number of failure cycles for the event based fault confirmation.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticEvent.eventFailureCycleCounterThreshold		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00096

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemEventParameter	
BSW Parameter		**BSW Type**
DemEventKind		EcucEnumerationParamDef
BSW Description		
This parameter is used to distinguish between SW-C and BSW events.		
Template Description		
This attribute is used to distinguish between SWC and BSW events.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticEvent.eventKind		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00095

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemEventParameter/DemEventKind	
BSW Parameter		**BSW Type**
DEM_EVENT_KIND_BSW		EcucEnumerationLiteralDef
BSW Description		
The event is a assigned to a BSW module		
Template Description		
The event is assigned to a BSW module.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticEventKindEnum.bsw		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00047

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemEventParameter/DemEventKind	
BSW Parameter		**BSW Type**
DEM_EVENT_KIND_SWC		EcucEnumerationLiteralDef
BSW Description		
The event is a assigned to a SW-C		
Template Description		
The event is assigned to a SWC.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticEventKindEnum.swc		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00049

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemEventParameter	
BSW Parameter		**BSW Type**
DemEventRecoverableInSameOperationCycle		EcucBooleanParamDef
BSW Description		
If parameter is configured to FALSE, reporting of PASSED will be ignored if the event is already "testfailed this operation cycle".		
Template Description		
If the attribute is set to true then reporting PASSED will reset the indication of a failed test in the current operation cycle. If the attribute is set to false then reporting PASSED will be ignored and not lead to a reset of the indication of a failed test.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticEvent.recoverableInSameOperationCycle		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00134

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter

BSW Parameter	BSW Type
DemFFPrestorageSupported	EcucBooleanParamDef
BSW Description	
If this parameter is set to true, then the Prestorage of FreezeFrames is supported by the assigned event. This parameter is useful to calculate the buffer size.	
Template Description	
This attribute describes whether the Prestorage of FreezeFrames is supported by the assigned event or not. True: Prestorage of FreezeFrames is supported False: Prestorage of FreezeFrames is not supported	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticEvent.prestorageFreezeFrame	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00103

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemIndicatorAttribute
BSW Parameter	**BSW Type**
DemIndicatorBehaviour	EcucEnumerationParamDef
BSW Description	
Behaviour of the linked indicator	
Template Description	
Behavior of the linked indicator.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticConnectedIndicator.behavior	
Mapping Rule	**Mapping Type**
	full
Mapping Status	**Mapping ID**
valid	up_Dem_00124

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemIndicatorAttribute/DemIndicator Behaviour
BSW Parameter	**BSW Type**
DEM_INDICATOR_BLINKING	EcucEnumerationLiteralDef
BSW Description	
The indicator blinks when the event has status FAILED Not relevant with J1939.	
Template Description	
The indicator blinks when the event has status FAILED.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticConnectedIndicatorBehaviorEnum.blinkMode	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00051

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemIndicatorAttribute/DemIndicatorBehaviour
BSW Parameter	**BSW Type**
DEM_INDICATOR_BLINK_CONT	EcucEnumerationLiteralDef
BSW Description	
The indicator is active and blinks when the event has status FAILED Not relevant with J1939.	
Template Description	
The indicator is active and blinks when the event has status FAILED.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticConnectedIndicatorBehaviorEnum.blinkOrContinuousOnMode	
Mapping Rule	**Mapping Type**
	full
Mapping Status	**Mapping ID**
valid	up_Dem_00054

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemIndicatorAttribute/DemIndicatorBehaviour
BSW Parameter	**BSW Type**
DEM_INDICATOR_CONTINUOUS	EcucEnumerationLiteralDef
BSW Description	
The indicator is active when the even has status FAILED	
Template Description	
The indicator is active when the event has status FAILED.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticConnectedIndicatorBehaviorEnum.continuousOnMode	
Mapping Rule	**Mapping Type**
	full
Mapping Status	**Mapping ID**
valid	up_Dem_00055

BSW Module	BSW Context
Dem	Dem/DemConfigSet/DemEventParameter/DemIndicatorAttribute/DemIndicatorBehaviour
BSW Parameter	**BSW Type**
DEM_INDICATOR_FAST_FLASH	EcucEnumerationLiteralDef
BSW Description	
Flash Indicator Lamp should be set to 'Fast Flash'	
Template Description	
Flash Indicator Lamp should be set to "Fast Flash".	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticConnectedIndicatorBehaviorEnum.fastFlashingMode	
Mapping Rule	**Mapping Type**
	full
Mapping Status	**Mapping ID**
valid	up_Dem_00052

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemEventParameter/DemIndicatorAttribute/DemIndicatorBehaviour	
BSW Parameter		**BSW Type**
DEM_INDICATOR_SLOW_FLASH		EcucEnumerationLiteralDef
BSW Description		
Flash Indicator Lamp should be set to 'Slow Flash'		
Template Description		
Flash Indicator Lamp should be set to "Slow Flash".		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticConnectedIndicatorBehaviorEnum.slowFlashingMode		
Mapping Rule		**Mapping Type**
		full
Mapping Status		**Mapping ID**
valid		up_Dem_00053

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemEventParameter/DemIndicatorAttribute	
BSW Parameter		**BSW Type**
DemIndicatorHealingCycleCounterThreshold		EcucIntegerParamDef
BSW Description		
Defines the number of healing cycles for the WarningIndicatorOffCriteria.		
Template Description		
This attribute defines the number of healing cycles for the WarningIndicatorOffCriteria		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticIndicator::DiagnosticIndicator.healingCycleCounterThreshold		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00087

BSW Module	BSW Context	
Dem	Dem/DemConfigSet	
BSW Parameter		**BSW Type**
DemObdDTC		EcucParamConfContainerDef
BSW Description		
This container contains the configuration (parameters) for DemObdDTC.		
Template Description		
Unique Diagnostic Trouble Code value for OBD.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticTroubleCodeObd.obdDTCValue		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00080

BSW Module	BSW Context	
Dem	Dem/DemConfigSet/DemObdDTC	
BSW Parameter		**BSW Type**

DemConsiderPtoStatus	EcucBooleanParamDef
BSW Description	
This parameter is TRUE, when the event is affected by the Dem PTO handling.	
Template Description	
This attribute describes the affection of the event by the Dem PTO handling. True: the event is affected by the Dem PTO handling. False: the event is not affected by the Dem PTO handling.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticTroubleCodeObd.considerPtoStatus	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00078

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemObdDTC
BSW Parameter	**BSW Type**
DemEventOBDReadinessGroup	EcucEnumerationParamDef
BSW Description	
This parameter specifies the Event OBD Readiness group for PID $01 and PID $41 computation. This parameter is only applicable for emission-related ECUs.	
Template Description	
DiagnosticTroubleCodeObd.eventObdReadinessGroup: This attribute specifies the Event OBD Readiness group for PID $01 and PID $41 computation. This attribute is only applicable for emission-related ECUs. **DiagnosticTroubleCodeUds.eventObdReadinessGroup**: This attribute specifies the Event OBD Readiness group for PID $01 and PID $41 computation. This attribute is only applicable for emission-related ECUs.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticTroubleCodeObd.eventObdReadiness Group, DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticTroubleCodeUds.eventObdReadiness Group	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00090

BSW Module	**BSW Context**
Dem	Dem/DemConfigSet/DemObdDTC
BSW Parameter	**BSW Type**
DemJ1939DTCValue	EcucIntegerParamDef
BSW Description	
Unique Diagnostic Trouble Code value for J1939 (consisting of SPN and FMI)	
Template Description	
DiagnosticTroubleCodeJ1939.spn: This represents the releated SPN. **DiagnosticTroubleCodeJ1939.fmi**: This attribute represents the behavior of the Failure Mode Indicator.	

M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticTroubleCodeJ1939.spn, DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticTroubleCodeJ1939.fmi	
Mapping Rule	**Mapping Type**
The value is created out of a combination of the two attribute fmi and spn. The details are explained in the J1939-73 document	full
Mapping Status	**Mapping ID**
valid	up_Dem_00133

BSW Module	BSW Context
Dem	Dem/DemGeneral
BSW Parameter	**BSW Type**
DemAgingRequiresTestedCycle	EcucBooleanParamDef
BSW Description	
Defines if the aging cycle counter is processed every aging cycles or if only tested aging cycle are considered. true: only tested aging cycle are considered for aging cycle counter false: aging cycle counter is processed every aging cycle	
Template Description	
Defines whether the aging cycle counter is processed every aging cycles or else only tested aging cycle are considered. If the attribute is set to TRUE: only tested aging cycle are considered for aging cycle counter. If the attribute is set to FALSE: aging cycle counter is processed every aging cycle.	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.agingRequiresTestedCycle	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00070

BSW Module	BSW Context
Dem	Dem/DemGeneral
BSW Parameter	**BSW Type**
DemClearDTCBehavior	EcucEnumerationParamDef
BSW Description	
Defines the clearing process of diagnostic information for volatile and non-volatile memory and the positive response handling for the Dcm module.	
Template Description	
This attribute defines the resulting UDS status byte for the related event, which shall not be cleared according to the ClearEventAllowed callback.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticEvent.clearEventBehavior	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00071

BSW Module	BSW Context	
Dem	Dem/DemGeneral	
BSW Parameter		**BSW Type**
DemClearDTCLimitation		EcucEnumerationParamDef
BSW Description		
Defines the supported Dem_<...>ClearDTC API scope.		
Template Description		
Defines the scope of the DEM_ClearDTC Api.		
M2 Parameter		
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.clearDtcLimitation		
Mapping Rule		**Mapping Type**
		full
Mapping Status		**Mapping ID**
valid		up_Dem_00105

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemClearDTCLimitation	
BSW Parameter		**BSW Type**
DEM_ALL_SUPPORTED_DTCS		EcucEnumerationLiteralDef
BSW Description		
Dem_<...>ClearDTC accepts all supported DTC values, as well as all DTC values which are configured in DemGroupDTCs and DEM_DTC_GROUP_ALL_DTCS.		
Template Description		
DEM_ClearDtc API accepts all supported DTC values.		
M2 Parameter		
DiagnosticExtract::DiagnosticCommonProps::DiagnosticClearDtcLimitationEnum.allSupportedDtcs		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00063

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemClearDTCLimitation	
BSW Parameter		**BSW Type**
DEM_ONLY_CLEAR_ALL_DTCS		EcucEnumerationLiteralDef
BSW Description		
Dem_<...>ClearDTC accepts ClearAllDTCs only.		
Template Description		
DEM_ClearDtc API accepts ClearAllDTCs only.		
M2 Parameter		
DiagnosticExtract::DiagnosticCommonProps::DiagnosticClearDtcLimitationEnum.clearAllDtcs		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00064

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemDataElementClass/DemInternalDataElementClass	
BSW Parameter		**BSW Type**
DemInternalDataElement		EcucEnumerationParamDef

BSW Description	
This parameter defines the Dem-internal data value, which is mapped to the data element.	
Template Description	
This represents the ability to further specify the access within the Dem.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticDemProvidedDataMapping.dataProvider	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00107

BSW Module	BSW Context
Dem	Dem/DemGeneral
BSW Parameter	**BSW Type**
DemEnvironmentDataCapture	EcucEnumerationParamDef
BSW Description	
DemEnvironmentDataCapture defines the point in time, when the data actually is captured.	
Template Description	
This attribute determines the point in time, when the data actually is captured.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticTroubleCodeProps.environmentCaptureToReporting	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00034

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEnvironmentDataCapture
BSW Parameter	**BSW Type**
DEM_CAPTURE_ASYNCHRONOUS_TO_REPORTING	EcucEnumerationLiteralDef
BSW Description	
The data capturing is postponed to the next cycle of the Dem_Mainfunction. (This means that there is a minimum delay between report of the failure and capturing the data).	
Template Description	
The data capturing is postponed to the next cycle of the Dem_Mainfunction. (This means that there is a minimum delay between report of the failure and capturing the data.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTroubleCode::EnvironmentCaptureToReportingEnum.captureAsynchronousToReporting	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00032

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEnvironmentDataCapture
BSW Parameter	**BSW Type**
DEM_CAPTURE_SYNCHRONOUS_TO_REPORTING	EcucEnumerationLiteralDef

BSW Description	
The data is captured immediately within the context of Dem_SetEventStatus.	
Template Description	
The data is captured immediately within the reporting function (i.e. in the context of the setEventStatus/reportErrorStatus function).	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTroubleCode::EnvironmentCaptureToReportingEnum.captureSynchronousToReporting	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00033

BSW Module	BSW Context
Dem	Dem/DemGeneral
BSW Parameter	**BSW Type**
DemEventDisplacementStrategy	EcucEnumerationParamDef
BSW Description	
This configuration switch defines, whether support for event displacement is enabled or not, and which displacement strategy is followed.	
Template Description	
This attribute defines, whether support for event displacement is enabled or not, and which displacement strategy is followed.	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.eventDisplacementStrategy	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00117

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventDisplacementStrategy
BSW Parameter	**BSW Type**
DEM_DISPLACEMENT_FULL	EcucEnumerationLiteralDef
BSW Description	
Event memory entry displacement is enabled, by consideration of priority active/passive status, and occurrence.	
Template Description	
Event memory entry displacement is enabled, by consideration of priority active/passive status, and occurrence.	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticEventDisplacementStrategyEnum.full	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00126

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventDisplacementStrategy

BSW Parameter	BSW Type
DEM_DISPLACEMENT_NONE	EcucEnumerationLiteralDef
BSW Description	
Event memory entry displacement is disabled.	
Template Description	
Event memory entry displacement is disabled.	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticEventDisplacementStrategyEnum.none	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00125

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventDisplacementStrategy
BSW Parameter	**BSW Type**
DEM_DISPLACEMENT_PRIO_OCC	EcucEnumerationLiteralDef
BSW Description	
Event memory entry displacement is enabled, by consideration of priority and occurrence (but without active/passive status).	
Template Description	
Event memory entry displacement is enabled, by consideration of priority and occurrence (but without active/passive status).	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticEventDisplacementStrategyEnum.prioOcc	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00060

BSW Module	BSW Context
Dem	Dem/DemGeneral
BSW Parameter	**BSW Type**
DemEventMemoryEntryStorageTrigger	EcucEnumerationParamDef
BSW Description	
Configures the primary trigger to allocate an event memory entry.	
Template Description	
Describes the primary trigger to allocate an event memory entry.	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.memoryEntryStorageTrigger	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00089

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet
BSW Parameter	**BSW Type**
DemDtcStatusAvailabilityMask	EcucIntegerParamDef

BSW Description	
Mask for the supported DTC status bits by the Dem. This mask is used by UDS service 0x19.	
Template Description	
Mask for the supported DTC status bits by the Dem.	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.dtcStatusAvailabilityMask	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00084

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet
BSW Parameter	**BSW Type**
DemMILIndicatorRef	EcucReferenceDef
BSW Description	
This parameter defines the indicator representing the MIL. This parameter is mandatory for ECUs supporting OBD (refer to DemOBDSupport).	
Template Description	
Defines the type of the indicator.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticIndicator::DiagnosticIndicator.type	
Mapping Rule	**Mapping Type**
DignosticIndicator.type == DiagnosticIndicatorTypeEnum.malfunction	full
Mapping Status	**Mapping ID**
valid	up_Dem_00006

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet
BSW Parameter	**BSW Type**
DemMirrorMemory	EcucParamConfContainerDef
BSW Description	
This container contains the mirror event memory specific parameters of the Dem module.	
Template Description	
This represents a mirror memory for a diagnostic event.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticMemoryDestinationMirror	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00075

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemMirrorMemory
BSW Parameter	**BSW Type**
DemMaxNumberEventEntryMirror	EcucIntegerParamDef
BSW Description	
Maximum number of events which can be stored in the mirror memory	
Template Description	

This attribute fixes the maximum number of event entries in the fault memory.	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.maxNumberOfEventEntries	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00109

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet
BSW Parameter	**BSW Type**
DemPrimaryMemory	EcucParamConfContainerDef
BSW Description	
This container contains the primary event memory specific parameters of the Dem module.	
Template Description	
This represents a primary memory for a diagnostic event.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticMemoryDestinationPrimary	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00076

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet/DemPrimaryMemory
BSW Parameter	**BSW Type**
DemMaxNumberEventEntryPrimary	EcucIntegerParamDef
BSW Description	
Maximum number of events which can be stored in the primary memory	
Template Description	
This attribute fixes the maximum number of event entries in the fault memory.	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.maxNumberOfEventEntries	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00110

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemEventMemorySet
BSW Parameter	**BSW Type**
DemTypeOfDTCSupported	EcucEnumerationParamDef
BSW Description	
This parameter defines the format returned by Dem_GetTranslationType and does not relate to/influence the supported Dem functionality.	
Template Description	
This attribute defines the format returned by Dem_DcmGetTranslationType and does not relate to/influence the supported Dem functionality.	

M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.typeOfDtcSupported	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00123

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemTypeOfDTCSupported
BSW Parameter	**BSW Type**
DEM_DTC_TRANSLATION_ISO11992_4	EcucEnumerationLiteralDef
BSW Description	
ISO11992-4 DTC format	
Template Description	
ISO11992-4 DTC format	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticTypeOfDtcSupportedEnum.iso11992_4	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00056

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemTypeOfDTCSupported
BSW Parameter	**BSW Type**
DEM_DTC_TRANSLATION_ISO14229_1	EcucEnumerationLiteralDef
BSW Description	
ISO14229-1 DTC format (3 byte format)	
Template Description	
ISO14229-1 DTC format (3 byte format)	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticTypeOfDtcSupportedEnum.iso14229_1	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00059

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemEventMemorySet/DemTypeOfDTCSupported
BSW Parameter	**BSW Type**
DEM_DTC_TRANSLATION_SAEJ1939_73	EcucEnumerationLiteralDef
BSW Description	
SAEJ1939-73 DTC format	
Template Description	
SAEJ1939-73 DTC format	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticTypeOfDtcSupportedEnum.saeJ1939_73	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**

valid	up_Dem_00057

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemEventMemorySet/DemTypeOfDTCSupported	
BSW Parameter		**BSW Type**
DEM_DTC_TRANSLATION_SAE_J2012_DA_DTCFORMAT_04		EcucEnumerationLiteralDef
BSW Description		
SAE_J2012-DA_DTCFormat_00 (3 byte format)		
Template Description		
SAE_J2012-DA_DTCFormat_00 (3 byte format)		
M2 Parameter		
DiagnosticExtract::DiagnosticCommonProps::DiagnosticTypeOfDtcSupportedEnum.saeJ2012_da		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00058

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemEventMemorySet	
BSW Parameter		**BSW Type**
DemUserDefinedMemory		EcucParamConfContainerDef
BSW Description		
This container contains the user defined event memory specific parameters of the Dem module.		
Template Description		
This represents a user-defined memory for a diagnostic event.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticMemoryDestinationUserDefined		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00077

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory	
BSW Parameter		**BSW Type**
DemMaxNumberEventEntryUserDefined		EcucIntegerParamDef
BSW Description		
Maximum number of events which can be stored in the user defined memory.		
Template Description		
This attribute fixes the maximum number of event entries in the fault memory.		
M2 Parameter		
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.maxNumberOfEventEntries		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00111

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemEventMemorySet/DemUserDefinedMemory	
BSW Parameter	**BSW Type**	
DemUserDefinedMemoryIdentifier	EcucIntegerParamDef	
BSW Description		
Identifier used by external tester to identify the User defined event memory.		
Template Description		
This represents the identifier of the user-defined memory.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticMemoryDestinationUserDefined.memoryId		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00112

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemExtendedDataRecordClass	
BSW Parameter	**BSW Type**	
DemExtendedDataRecordNumber	EcucIntegerParamDef	
BSW Description		
This configuration parameter specifies an unique identifier for an extended data record. One or more extended data records can be assigned to one diagnostic event/DTC. 0x00 is reserved by ISO (therefore the minimal value equals 1) 0xF0 to 0xFF are reserved by ISO (therefore the maximal value equals 239)		
Template Description		
This attribute specifies an unique identifier for an extended data record.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticExtendedDataRecord::DiagnosticExtendedDataRecord.recordNumber		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00102

BSW Module	BSW Context	
Dem	Dem/DemGeneral/DemExtendedDataRecordClass	
BSW Parameter	**BSW Type**	
DemExtendedDataRecordTrigger	EcucEnumerationParamDef	
BSW Description		
Defines the trigger to store the ExtendedDataRecord.		
Template Description		
This attribute specifies the primary trigger to allocate an event memory entry.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticExtendedDataRecord::DiagnosticExtendedDataRecord.trigger		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00092

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemExtendedDataRecordClass/DemExtendedDataRecordTrigger

BSW Parameter	BSW Type
DEM_TRIGGER_ON_CONFIRMED	EcucEnumerationLiteralDef

BSW Description
ExtendedDataRecord will be stored when the UDS status confirmed bit changes from 0 to 1.
Template Description
capture on "Confirmed"
M2 Parameter
DiagnosticExtract::Dem::DiagnosticFreezeFrame::DiagnosticRecordTriggerEnum.confirmed

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00044

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemExtendedDataRecordClass/DemExtendedDataRecordTrigger

BSW Parameter	BSW Type
DEM_TRIGGER_ON_FDC_THRESHOLD	EcucEnumerationLiteralDef

BSW Description
ExtendedDataRecord will be stored when the FDC reaches its threshold.
Template Description
capture on "FDC Threshold"
M2 Parameter
DiagnosticExtract::Dem::DiagnosticFreezeFrame::DiagnosticRecordTriggerEnum.fdcThreshold

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00046

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemExtendedDataRecordClass/DemExtendedDataRecordTrigger

BSW Parameter	BSW Type
DEM_TRIGGER_ON_PENDING	EcucEnumerationLiteralDef

BSW Description
ExtendedDataRecord will be stored when the UDS status pending bit changes from 0 to 1.
Template Description
capture on "Pending"
M2 Parameter
DiagnosticExtract::Dem::DiagnosticFreezeFrame::DiagnosticRecordTriggerEnum.pending

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00045

BSW Module	BSW Context

Dem	Dem/DemGeneral/DemExtendedDataRecordClass/DemExtendedDataRecordTrigger	
BSW Parameter		**BSW Type**
DEM_TRIGGER_ON_TEST_FAILED		EcucEnumerationLiteralDef
BSW Description		
ExtendedDataRecord will be stored when the UDS status test failed bit changes from 0 to 1.		
Template Description		
capture on "Test Failed"		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticFreezeFrame::DiagnosticRecordTriggerEnum.testFailed		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00043

BSW Module	**BSW Context**	
Dem	Dem/DemGeneral/DemExtendedDataRecordClass	
BSW Parameter		**BSW Type**
DemExtendedDataRecordUpdate		EcucEnumerationParamDef
BSW Description		
This extended data record is captured if the configured trigger condition in "DemExtended-DataRecordTrigger" is fulfilled.		
Template Description		
This attribute defines when an extended data record is captured. True: This extended data record is captured every time. False: This extended data record is only captured for new event memory entries.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticExtendedDataRecord::DiagnosticExtendedDataRecord.update		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00104

BSW Module	**BSW Context**	
Dem	Dem/DemGeneral/DemFreezeFrameRecordClass	
BSW Parameter		**BSW Type**
DemFreezeFrameRecordNumber		EcucIntegerParamDef
BSW Description		
This parameter defines a record number for a freeze frame record. This record number is unique per freeze frame record number class.		
Template Description		
This attribute defines a record number for a freeze frame record.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticFreezeFrame::DiagnosticFreezeFrame.recordNumber		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00039

BSW Module	**BSW Context**

Dem	Dem/DemGeneral/DemFreezeFrameRecordClass	
BSW Parameter		**BSW Type**
DemFreezeFrameRecordTrigger		EcucEnumerationParamDef
BSW Description		
Defines the trigger to store the FreezeFrameRecord.		
Template Description		
This attribute defines the primary trigger to allocate an event memory entry.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticFreezeFrame::DiagnosticFreezeFrame.trigger		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00093

BSW Module	**BSW Context**	
Dem	Dem/DemGeneral/DemFreezeFrameRecordClass	
BSW Parameter		**BSW Type**
DemFreezeFrameRecordUpdate		EcucEnumerationParamDef
BSW Description		
This parameter defines the case, when the freeze frame record is stored/updated.		
Template Description		
This attribute defines the approach when the freeze frame record is stored/updated. True: FreezeFrame record is captured every time. False: FreezeFrame record is only captured for new event memory entries.		
M2 Parameter		
DiagnosticExtract::Dem::DiagnosticFreezeFrame::DiagnosticFreezeFrame.update		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00085

BSW Module	**BSW Context**	
Dem	Dem/DemGeneral	
BSW Parameter		**BSW Type**
DemOBDSupport		EcucEnumerationParamDef
BSW Description		
This configuration switch defines OBD support and kind of OBD ECU.		
Template Description		
This attribute is used to specify the role (if applicable) in which the DiagnosticEcuInstance supports OBD.		
M2 Parameter		
DiagnosticExtract::DiagnosticContribution::DiagnosticEcuInstanceProps.obdSupport		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		

BSW Module	**BSW Context**	
Dem	Dem/DemGeneral/DemOBDSupport	
BSW Parameter		**BSW Type**

DEM_OBD_DEP_SEC_ECU	EcucEnumerationLiteralDef
BSW Description	
Kind of OBD ECU: OBD Dependend / Secondary ECU	
Template Description	
This represents the role "secondary ECU".	
M2 Parameter	
DiagnosticExtract::DiagnosticContribution::DiagnosticObdSupportEnum.secondaryEcu	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00129

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemOBDSupport
BSW Parameter	**BSW Type**
DEM_OBD_MASTER_ECU	EcucEnumerationLiteralDef
BSW Description	
Kind of OBD ECU: Master ECU	
Template Description	
This represent the role "master ECU".	
M2 Parameter	
DiagnosticExtract::DiagnosticContribution::DiagnosticObdSupportEnum.masterEcu	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00131

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemOBDSupport
BSW Parameter	**BSW Type**
DEM_OBD_NO_OBD_SUPPORT	EcucEnumerationLiteralDef
BSW Description	
OBD is not supported within this ECU	
Template Description	
This represents the ability to explicitly specify that no participation in OBD is foreseen.	
M2 Parameter	
DiagnosticExtract::DiagnosticContribution::DiagnosticObdSupportEnum.noObdSupport	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00128

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemOBDSupport
BSW Parameter	**BSW Type**
DEM_OBD_PRIMARY_ECU	EcucEnumerationLiteralDef
BSW Description	
Kind of OBD ECU: Pimary ECU	
Template Description	
This represents the role "primary ECU".	

M2 Parameter	
DiagnosticExtract::DiagnosticContribution::DiagnosticObdSupportEnum.primaryEcu	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00130

BSW Module	BSW Context
Dem	Dem/DemGeneral
BSW Parameter	**BSW Type**
DemOccurrenceCounterProcessing	EcucEnumerationParamDef
BSW Description	
This configuration switch defines the consideration of the fault confirmation process for the occurrence counter. For OBD and mixed systems (OBD/non OBD, refer to DemOBDSupport) configuration switch shall always set to DEM_PROCESS_OCCCTR_TF.	
Template Description	
This attribute defines the consideration of the fault confirmation process for the occurrence counter.	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.occurrenceCounterProcessing	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00040

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemOccurrenceCounterProcessing
BSW Parameter	**BSW Type**
DEM_PROCESS_OCCCTR_CDTC	EcucEnumerationLiteralDef
BSW Description	
the occurrence counter is triggered by the TestFailed bit if the fault confirmation was successful (ConfirmedDTC bit is set)	
Template Description	
The occurrence counter is triggered by the TestFailed bit if the fault confirmation was successful (ConfirmedDTC bit is set).	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticOccurrenceCounterProcessingEnum.confirmedDtcBit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00037

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemOccurrenceCounterProcessing
BSW Parameter	**BSW Type**
DEM_PROCESS_OCCCTR_TF	EcucEnumerationLiteralDef
BSW Description	
the occurrence counter is only triggered by the TestFailed bit (and the fault confirmation is not considered) This parameter is mandatory in case of J1939.	

Template Description	
The occurrence counter is only triggered by the TestFailed bit (and the fault confirmation is not considered).	
M2 Parameter	
DiagnosticExtract::DiagnosticCommonProps::DiagnosticOccurrenceCounterProcessingEnum.testFailedBit	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00038

BSW Module	BSW Context
Dem	Dem/DemGeneral
BSW Parameter	**BSW Type**
DemOperationCycle	EcucParamConfContainerDef
BSW Description	
This container holds all parameters that are relevant to configure an operation cycle.	
Template Description	
Definition of an operation cycle that is the base of the event qualifying and for Dem scheduling.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticOperationCycle::DiagnosticOperationCycle	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00113

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemOperationCycle
BSW Parameter	**BSW Type**
DemOperationCycleAutostart	EcucBooleanParamDef
BSW Description	
The autostart property defines if the operation cycles is automatically (re-)started during Dem_PreInit.	
Template Description	
This attribute defines if the operation cycles is automatically re-started during Dem_PreInit.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticOperationCycle::DiagnosticOperationCycle.cycleAutostart	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00114

BSW Module	BSW Context
Dem	Dem/DemGeneral
BSW Parameter	**BSW Type**
DemOperationCycleStatusStorage	EcucBooleanParamDef
BSW Description	
Defines if the operation cycle state is available over the power cycle (stored non-volatile) or not. true: the operation cycle state is stored non-volatile false: the operation cycle state is only stored volatile	

Template Description	
Defines if the operation cycle state is available over the power cycle (stored non-volatile) or not. true: the operation cycle state is stored non-volatile false: the operation cycle state is only stored volatile	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticOperationCycle::DiagnosticOperationCycle.cycleStatusStorage	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00115

BSW Module	BSW Context
Dem	Dem/DemGeneral
BSW Parameter	**BSW Type**
DemRatio	EcucParamConfContainerDef
BSW Description	
This container contains the OBD-specific in-use-monitor performance ratio configuration. It is related to a specific event, a FID, and an IUMPR group.	
Template Description	
ObdRatioServiceNeeds: Specifies the abstract needs of a component or module on the configuration of OBD Services in relation to a particular "ratio monitoring" which is supported by this component or module. **DiagnosticIumprGroup**: This meta-class represents the ability to model a IUMPR groups.	
M2 Parameter	
CommonStructure::ServiceNeeds::ObdRatioServiceNeeds, DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticIumprGroup	
Mapping Rule	**Mapping Type**
In case the owner of the ObdRatioServiceNeeds is a BSW module then the DemRatio.shortName = {capitalizedMip}_{ServiceDependency.symbolicNameProps.symbol}. For the DiagnosticIumprGroup the mapping rule is 1:1	full
Mapping Status	**Mapping ID**
valid	up_Dem_00001

BSW Module	BSW Context
Dem	Dem/DemGeneral/DemRatio
BSW Parameter	**BSW Type**
DemDiagnosticEventRef	EcucReferenceDef
BSW Description	
This reference contains the link to a diagnostic event.	
Template Description	
DiagnosticIumprGroup.iumpr: This reference collects DiagnosticIumpr to a DiagnosticIumprGroup. **DiagnosticIumpr.event**: This reference represents the DiagnosticEvent that corresoponds to the IUMPR computation.	
M2 Parameter	

DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticIumprGroup.iumpr, DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticIumpr.event	
Mapping Rule	**Mapping Type**
foreach DiagnosticIumprGroup, follow the iumpr reference and then pick the target of the event reference	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio
BSW Parameter	**BSW Type**
DemIUMPRDenGroup	EcucEnumerationParamDef
BSW Description	
This parameter specifies the assigned denominator type which is applied in addition to the DEM_IUMPR_GENERAL_INDIVIDUAL_DENOMINATOR conditions.	
Template Description	
This reference collects DiagnosticIumpr to a DiagnosticIumprDenominatorGroup.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticIumprDenominatorGroup.iumpr,	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio/DemIUMPRDenGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_DEN_500MILL	EcucEnumerationLiteralDef
BSW Description	
Additional condition based on definition of 500miles conditions as defined for OBD2.	
Template Description	
DiagnosticIumprDenominatorGroup.iumpr: This reference collects DiagnosticIumpr to a DiagnosticIumprDenominatorGroup. **Identifiable.category**: The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticIumprDenominatorGroup.iumpr, GenericStructure::GeneralTemplateClasses::Identifiable::Identifiable.category	
Mapping Rule	**Mapping Type**
DiagnosticIumprDenominatorGroup.category ==IUMPR_DENOMINATOR_500_MILES	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio/DemIUMPRDenGroup
BSW Parameter	**BSW Type**
DEM_IUMPR_DEN_COLDSTART	EcucEnumerationLiteralDef
BSW Description	

Additional condition based on definition of "cold start" as defined for EU5+
Template Description
DiagnosticIumprDenominatorGroup.iumpr: This reference collects DiagnosticIumpr to a DiagnosticIumprDenominatorGroup. **Identifiable.category**: The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.
M2 Parameter
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticIumprDenominatorGroup.iumpr, GenericStructure::GeneralTemplateClasses::Identifiable::Identifiable.category

Mapping Rule	**Mapping Type**
DiagnosticIumprDenominatorGroup == IUMPR_DENOMINATOR_COLDSTART	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio/DemIUMPRDenGroup

BSW Parameter	**BSW Type**
DEM_IUMPR_DEN_EVAP	EcucEnumerationLiteralDef

BSW Description
Additional condition based on definition of "EVAP" conditions as defined for OBD2.
Template Description
DiagnosticIumprDenominatorGroup.iumpr: This reference collects DiagnosticIumpr to a DiagnosticIumprDenominatorGroup. **Identifiable.category**: The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.
M2 Parameter
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticIumprDenominatorGroup.iumpr, GenericStructure::GeneralTemplateClasses::Identifiable::Identifiable.category

Mapping Rule	**Mapping Type**
DiagnosticIumprDenominatorGroup.category == IUMPR_DENOMINATOR_EVAP	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio/DemIUMPRDenGroup

BSW Parameter	**BSW Type**
DEM_IUMPR_DEN_NONE	EcucEnumerationLiteralDef

BSW Description
No further condition. Denominator increments based on GENERAL_INDIVIDUAL_DENOMINATOR only.
Template Description

DiagnosticIumprDenominatorGroup.iumpr: This reference collects DiagnosticIumpr to a DiagnosticIumprDenominatorGroup. **Identifiable.category**: The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.
M2 Parameter
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticIumprDenominatorGroup.iumpr, GenericStructure::GeneralTemplateClasses::Identifiable::Identifiable.category

Mapping Rule	**Mapping Type**
DiagnosticIumprDenominatorGroup.category == IUMPR_DENOMINATOR_NONE	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio/DemIUMPRDenGroup

BSW Parameter	**BSW Type**
DEM_IUMPR_DEN_PHYS_API	EcucEnumerationLiteralDef

BSW Description
Additional physical condition (component activity) computed wihtin the SW-C and reported via Dem_RepIUMPRDenLock / Dem_RepIUMPRDenRelease.
Template Description
DiagnosticIumprDenominatorGroup.iumpr: This reference collects DiagnosticIumpr to a DiagnosticIumprDenominatorGroup. **Identifiable.category**: The category is a keyword that specializes the semantics of the Identifiable. It affects the expected existence of attributes and the applicability of constraints.
M2 Parameter
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticIumprDenominatorGroup.iumpr, GenericStructure::GeneralTemplateClasses::Identifiable::Identifiable.category

Mapping Rule	**Mapping Type**
DiagnosticIumprDenominatorGroup.category == IUMPR_DENOMINATOR_PHYSICAL_API	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
Dem	Dem/DemGeneral/DemRatio

BSW Parameter	**BSW Type**
DemIUMPRGroup	EcucEnumerationParamDef

BSW Description
This parameter specifies the assigned IUMPR group of the ratio Id.
Template Description
This reference collects DiagnosticIumpr to a DiagnosticIumprGroup.
M2 Parameter
DiagnosticExtract::Dem::DiagnosticEvent::DiagnosticIumprGroup.iumpr

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context	
Dem	Dem/DemGeneral	
BSW Parameter		**BSW Type**
DemResetConfirmedBitOnOverflow		EcucBooleanParamDef
BSW Description		
This configuration switch defines, whether the confirmed bit is reset or not while an event memory entry will be displaced.		
Template Description		
This attribute defines, whether the confirmed bit is reset or not while an event memory entry will be displaced.		
M2 Parameter		
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.resetConfirmedBitOnOverflow		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00121

BSW Module	BSW Context	
Dem	Dem/DemGeneral	
BSW Parameter		**BSW Type**
DemStatusBitHandlingTestFailedSinceLastClear		EcucEnumerationParamDef
BSW Description		
This configuration switch defines, whether the aging and displacement mechanism shall be applied to the "TestFailedSinceLastClear" status bits.		
Template Description		
This attribute defines, whether the aging and displacement mechanism shall be applied to the "TestFailedSinceLastClear" status bits.		
M2 Parameter		
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.statusBitHandlingTestFailedSinceLastClear		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00122

BSW Module	BSW Context	
Dem	Dem/DemGeneral	
BSW Parameter		**BSW Type**
DemStatusBitStorageTestFailed		EcucBooleanParamDef
BSW Description		
Activate/Deactivate the permanent storage of the "TestFailed" status bits. true: storage activated false: storage deactivated		
Template Description		
This parameter is used to activate/deactivate the permanent storage of the "TestFailed" status bits. true: storage activated false: storage deactivated		
M2 Parameter		
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.statusBitStorageTestFailed		
Mapping Rule		**Mapping Type**

1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00116

BSW Module	**BSW Context**	
Dem	Dem/DemGeneral	
BSW Parameter		**BSW Type**
DemTypeOfFreezeFrameRecordNumeration		EcucEnumerationParamDef
BSW Description		
This parameter defines the type of assigning freeze frame record numbers for event-specific freeze frame records.		
Template Description		
This attribute defines the type of assigning freeze frame record numbers for event-specific freeze frame records.		
M2 Parameter		
DiagnosticExtract::DiagnosticCommonProps::DiagnosticCommonProps.typeOfFreezeFrameRecordNumeration		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00041

BSW Module	**BSW Context**	
Dem	Dem/DemGeneral/DemTypeOfFreezeFrameRecordNumeration	
BSW Parameter		**BSW Type**
DEM_FF_RECNUM_CALCULATED		EcucEnumerationLiteralDef
BSW Description		
freeze frame records will be numbered consecutive starting by 1 in their chronological order		
Template Description		
Freeze frame records will be numbered consecutive starting by 1 in their chronological order.		
M2 Parameter		
DiagnosticExtract::DiagnosticCommonProps::DiagnosticTypeOfFreezeFrameRecordNumerationEnum.calculated		
Mapping Rule		**Mapping Type**
1:1 mapping		full
Mapping Status		**Mapping ID**
valid		up_Dem_00035

BSW Module	**BSW Context**	
Dem	Dem/DemGeneral/DemTypeOfFreezeFrameRecordNumeration	
BSW Parameter		**BSW Type**
DEM_FF_RECNUM_CONFIGURED		EcucEnumerationLiteralDef
BSW Description		
freeze frame records will be numbered based on the given configuration in their chronological order		
Template Description		
Freeze frame records will be numbered based on the given configuration in their chronological order.		
M2 Parameter		
DiagnosticExtract::DiagnosticCommonProps::DiagnosticTypeOfFreezeFrameRecordNumerationEnum.configured		
Mapping Rule		**Mapping Type**

1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dem_00036

E.4 Fim

BSW Module	**BSW Context**	
FiM	FiM/FiMConfigSet	
BSW Parameter		**BSW Type**
FiMFID		EcucParamConfContainerDef
BSW Description		
This container includes symbolic names of all FIDs.		
Template Description		
FunctionInhibitionNeeds: Specifies the abstract needs on the configuration of the Function Inhibition Manager for one Function Identifier (FID). This class currently contains no attributes. Its name can be regarded as a symbol identifying the FID from the viewpoint of the component or module which owns this class. **DiagnosticFunctionIdentifier**: This meta-class represents a diagnostic function identifier (a.k.a. FID).		
M2 Parameter		
CommonStructure::ServiceNeeds::FunctionInhibitionNeeds DiagnosticExtract::Fim::DiagnosticFunctionIdentifier		
Mapping Rule		**Mapping Type**
In case the owner of the FunctionInhibitionNeeds is a BSW module then the FiMFID.shortName= {capitalizedMip}_{ServiceDependency.symbolicNameProps.symbol}.		full
Mapping Status		**Mapping ID**
valid		up_FiM_00001

BSW Module	**BSW Context**	
FiM	FiM/FiMConfigSet	
BSW Parameter		**BSW Type**
FiMInhibitionConfiguration		EcucParamConfContainerDef
BSW Description		
This container includes all configuration parameters concerning the relationship between event and FID.		
Template Description		
This meta-class represents the ability to define the inhibition of a specific function identifier within the Fim configuration.		
M2 Parameter		
DiagnosticExtract::Fim::DiagnosticFunctionIdentifierInhibit		
Mapping Rule		**Mapping Type**
		full
Mapping Status		**Mapping ID**
valid		

BSW Module	**BSW Context**	
FiM	FiM/FiMConfigSet/FiMInhibitionConfiguration	
BSW Parameter		**BSW Type**

FiMInhEventRef	EcucSymbolicNameReferenceDef
BSW Description	
Selection of an single DEM Event.	
Template Description	
This represents the alias event appllicable for the referencing inhibition source.	
M2 Parameter	
DiagnosticExtract::Fim::DiagnosticFunctionInhibitSource.event	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
FiM	FiM/FiMConfigSet/FiMInhibitionConfiguration
BSW Parameter	**BSW Type**
FiMInhInhibitionMask	EcucEnumerationParamDef
BSW Description	
The configuration parameter is used to specify the inhibition mask for an event - FID relation.	
Template Description	
This represents the value of the inhibition mask behavior.	
M2 Parameter	
DiagnosticExtract::Fim::DiagnosticFunctionIdentifierInhibit.inhibitionMask	
Mapping Rule	**Mapping Type**
	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
FiM	FiM/FiMConfigSet/FiMInhibitionConfiguration/FiMInhInhibitionMask
BSW Parameter	**BSW Type**
FIM_LAST_FAILED	EcucEnumerationLiteralDef
BSW Description	
Last Failed - DEM_UDS_STATUS_TF flag of Dem Eventstatus is set Use case: Re-configuration, avoiding follow-up errors	
Template Description	
This represents the inhibition mask behavior "last failed".	
M2 Parameter	
DiagnosticExtract::Fim::DiagnosticInhibitionMaskEnum.lastFailed	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
FiM	FiM/FiMConfigSet/FiMInhibitionConfiguration/FiMInhInhibitionMask
BSW Parameter	**BSW Type**
FIM_NOT_TESTED	EcucEnumerationLiteralDef
BSW Description	
Not Tested this cycle - DEM_UDS_STATUS_TNCTOC flag of Dem Eventstatus is set. Use case: Scheduling of monitors.	

Template Description	
This represents the inhibition mask behavior "not tested".	
M2 Parameter	
DiagnosticExtract::Fim::DiagnosticInhibitionMaskEnum.notTested	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
FiM	FiM/FiMConfigSet/FiMInhibitionConfiguration/FiMInhInhibitionMask
BSW Parameter	**BSW Type**
FIM_TESTED	EcucEnumerationLiteralDef
BSW Description	
Tested - DEM_UDS_STATUS_TNCTOC flag of Dem Eventstatus is not set. Use case: Self deactivation, check during driving cycle.	
Template Description	
This represents the inhibition mask behavior "tested".	
M2 Parameter	
DiagnosticExtract::Fim::DiagnosticInhibitionMaskEnum.tested	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
FiM	FiM/FiMConfigSet/FiMInhibitionConfiguration/FiMInhInhibitionMask
BSW Parameter	**BSW Type**
FIM_TESTED_AND_FAILED	EcucEnumerationLiteralDef
BSW Description	
Tested and Failed - DEM_UDS_STATUS_TF flag of Dem Eventstatus is set and DEM_UDS_STATUS_TNCTOC flag is not set Use case: Avoiding deadlocks, repeated monitoring.	
Template Description	
This represents the inhibition mask behavior "tested and failed".	
M2 Parameter	
DiagnosticExtract::Fim::DiagnosticInhibitionMaskEnum.testedAndFailed	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
FiM	FiM/FiMConfigSet
BSW Parameter	**BSW Type**
FiMSummaryEvent	EcucParamConfContainerDef
BSW Description	

The summarized EventId definition record consists of a summarized event ID and specific Dem Events. This record means that a particular FID that has to be disabled in case of summarized event (defined above) is to be disabled in any of the specific events. A possible solution could be assigning events as summarized events along with a list of specific events. During the configuration process the summarized event substitutes the referenced single events. However, it is not outlined how this requirement is solved - whether by configuration process or by implementation within the FiM. The FiM configuration tool could also build up a suitable data structure for summarized events and deal with it in the FiM implementation.
Template Description
This meta-class represents the ability to model a Fim event group, also known as a summary event in Fim terminology. This represents a group of single diagnostic events.
M2 Parameter
DiagnosticExtract::Fim::DiagnosticFimEventGroup

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	

E.5 J1939 Dcm

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmChannel

BSW Parameter	**BSW Type**
J1939DcmBusType	EcucEnumerationParamDef
BSW Description	
Identifies the communication port	
Template Description	
This represents the network ID for the J1939 cluster.	
M2 Parameter	
SystemTemplate::Fibex::Fibex4Can::CanTopology::J1939Cluster.networkId	

Mapping Rule	**Mapping Type**
value 1 maps to J1939DCM_J1939_NETWORK_1 value 2 maps to J1939DCM_J1939_NETWORK_2 value 3 maps to J1939DCM_J1939_NETWORK_3 value 4 maps to J1939DCM_J1939_NETWORK_4	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmDspExternalSRDataElementClass/ J1939DcmDataElementInstance

BSW Parameter	**BSW Type**
J1939DcmDataElementInstanceRef	EcucInstanceReferenceDef
BSW Description	

Instance Reference to the primitive data which shall be read or written. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationPrimitiveDataType of category VALUE or BOOLEAN or if the AutosarDataPrototype is typed with a ImplementationDataType of category VALUE or TYPE_REFERENCE that in turn boils down to VALUE
Template Description
This represents the dataElement in the application software that is accessed for diagnostic purpose.
M2 Parameter
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping.mappedDataElement

Mapping Rule	**Mapping Type**
DiagnosticServiceDataMapping maps to a primitive data.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00100

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmDspExternalSRDataElementClass/J1939DcmSubElementInDataElementInstance

BSW Parameter	**BSW Type**
J1939DcmSubElementInDataElementInstanceRef	EcucInstanceReferenceDef

BSW Description
Instance Reference to the primitve sub-element (at any level) of composite data in a port which shall be read. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ApplicationCompositeDataType.
Template Description
This represents the dataElement in the application software that is accessed for diagnostic purpose.
M2 Parameter
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping.mappedDataElement

Mapping Rule	**Mapping Type**
DiagnosticServiceDataMapping maps to a primitive element within a composite data, where the AutosarDataPrototype is typed with a ApplicationCompositeDataType.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00101

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmDspExternalSRDataElementClass/J1939DcmSubElementInImplDataElementInstance

BSW Parameter	**BSW Type**
J1939DcmSubElementInImplDataElementInstanceRef	EcucInstanceReferenceDef

BSW Description
Instance Reference to the primitve sub-element (at any level) of composite data in a port which shall be read. Supported are VariableDataPrototypes in SenderReceiverInterfaces and NvDataInterfaces and ParameterDataPrototypes in ParameterInterfaces (read only). This reference is applicable if the AutosarDataPrototype is typed with a ImplementationDataType of category STRUCTURE or ARRAY. Please note that in case of ARRAY the index attribute in the target reference has to be set to select a single array element.

Template Description	
This represents the dataElement in the application software that is accessed for diagnostic purpose.	
M2 Parameter	
DiagnosticExtract::ServiceMapping::DiagnosticServiceDataMapping.mappedDataElement	
Mapping Rule	**Mapping Type**
DiagnosticServiceDataMapping maps to a primitive element within a composite data, where the AutosarDataPrototype is typed with a ApplicationCompositeDataType ImplementationDataType of category STRUCTURE or ARRAY.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00102

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmNode
BSW Parameter	**BSW Type**
J1939DcmServiceOnlyDTCsMemoryDestinationRef	EcucSymbolicNameReferenceDef
BSW Description	
Reference to the user defined memory used for the Service Only DTCs handled by DM53, DM54, and DM55.	
Template Description	
The event destination assigns events to none, one or multiple origins.	
M2 Parameter	
DiagnosticExtract::Dem::DiagnosticTroubleCode::DiagnosticTroubleCodeProps.memoryDestination	
Mapping Rule	**Mapping Type**
This upstream mapping shall only exist for a DiagnosticTroubleCodeJ1939 with attribute kind set to serviceOnly that references the DiagnosticTroubleCodeProps that owns the memoryDestination.	full
Mapping Status	**Mapping ID**
valid	up_J1939Dcm_00005

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions
BSW Parameter	**BSW Type**
J1939DcmModeCondition	EcucParamConfContainerDef
BSW Description	
This container contains the configuration of a mode condition or an environmental conditions which can be used as argument in J1939DcmModeRules. One J1939DcmModeCondition shall contain either one J1939DcmSwcModeRef or one J1939DcmBswModeRef or one J1939DcmSwcSRDataElementRef. Please note that the J1939Dcm acts as well as mode manager. Therefore the references J1939DcmSwcModeRef or one J1939DcmBswModeRef might point to provided ModeDeclarationGroupPrototypes of the J1939Dcm itself as well as to provided ModeDeclarationGroupPrototypes of other Bsw Modules or software components. In case of a configured J1939DcmSwcModeRef or J1939DcmBswModeRef only the J1939DcmConditionType J1939DCM_EQUALS or J1939DCM_EQUALS_NOT are applicable. In case of J1939DcmSwcSRDataElementRef all literals of J1939DcmConditionType are possible.	
Template Description	

DiagnosticCompareConditions are atomic conditions. They are based on the idea of a comparison at runtime of some variable data with something constant. The type of the comparison (==, !=, <, <=, ...) is specified in DiagnosticCompareCondition.compareType.
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvCompareCondition

Mapping Rule	**Mapping Type**
Depending on the reference a DcmModeCondition is mapped to a DiagnosticEnvModeCondition if only one reference is present and reference is a DcmSwcModeRef or a DcmBswModeRef. If two references are present, a DcmSwcSRDataElementRef and a DcmSwcSRDataElementValueRef, then DcmModeCondition is mapped to a DiagnosticEnvDataCondition.	full
Mapping Status	**Mapping ID**
valid	

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition

BSW Parameter	**BSW Type**
J1939DcmBswModeRef	EcucInstanceReferenceDef
BSW Description	
This parameter references a mode of a ModeDeclarationGroupPrototype provided by a Basic Software Module used for the condition. Please note that such ModeDeclarationGroupPrototype are owned by a Basic Software Module Description in the role providedModeGroup.	
Template Description	
This reference represents both the ModeDeclarationGroupPrototype and the ModeDeclaration relevant for the mode comparison.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvModeCondition.modeElement	

Mapping Rule	**Mapping Type**
For DcmModeRef a new DiagnosticEnvBswModeElement is used, pointing to the ModeDeclaration via ModeInModuleDescriptionInstanceRef. This new DiagnosticEnvModeElement shall be aggregated by the same DiagnosticEnvironmentalConfition as the DiagnosticEnvModeCondition, in which the target of the reference DiagnosticEnvModeCondition.modeElement points to the this DiagnosticEnvModeElement.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00273

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType

BSW Parameter	**BSW Type**
J1939DCM_EQUALS	EcucEnumerationLiteralDef
BSW Description	
Template Description	

DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticCompareTypeEnum.isEqual**: equal
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isEqual

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00265

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType

BSW Parameter	**BSW Type**
J1939DCM_EQUALS_NOT	EcucEnumerationLiteralDef
BSW Description	
Template Description	
DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticCompareTypeEnum.isNotEqual**: not equal	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isNotEqual	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00266

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType

BSW Parameter	**BSW Type**
J1939DCM_GREATER_OR_EQUAL	EcucEnumerationLiteralDef
BSW Description	
Template Description	

DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticCompareTypeEnum.isGreaterOrEqual**: greater than or equal
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isGreaterOrEqual

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00269

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType

BSW Parameter	**BSW Type**
J1939DCM_GREATER_THAN	EcucEnumerationLiteralDef
BSW Description	
Template Description	
DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticCompareTypeEnum.isGreaterThan**: greater than	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isGreaterThan	

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00267

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType

BSW Parameter	**BSW Type**
J1939DCM_LESS_OR_EQUAL	EcucEnumerationLiteralDef
BSW Description	
Template Description	

DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticCompareTypeEnum.isLessOrEqual**: less than or equal
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isLessOrEqual

Mapping Rule	**Mapping Type**
	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00270

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition/J1939DcmConditionType

BSW Parameter	**BSW Type**
J1939DCM_LESS_THAN	EcucEnumerationLiteralDef
BSW Description	

Template Description
DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticCompareTypeEnum.isLessThan**: less than
M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticCompareTypeEnum.isLessThan

Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00268

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition

BSW Parameter	**BSW Type**
J1939DcmSwcModeRef	EcucInstanceReferenceDef
BSW Description	
This parameter references a mode in a particular mode request port of a software component that is used for the condition.	
Template Description	
This reference represents both the ModeDeclarationGroupPrototype and the ModeDeclaration relevant for the mode comparison.	
M2 Parameter	

DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvModeCondition.modeElement	
Mapping Rule	**Mapping Type**
For DcmModeRef a new DiagnosticEnvSwcModeElement is used, pointing to the ModeDeclaration via PModeInSystemInstanceRef. This new DiagnosticEnvModeElement shall be aggregated by the same DiagnosticEnvironmentalConfition as the DiagnosticEnvModeCondition, in which the target of the reference DiagnosticEnvModeCondition.modeElement points to the this DiagnosticEnvModeElement.	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00272

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition
BSW Parameter	**BSW Type**
J1939DcmSwcSRDataElementRef	EcucReferenceDef
BSW Description	
Reference to environmental conditions. It is possible to reference a S/R Receiver-Port to read physical values and compare (equal, greater, less,...) them with a configured value that is defined by J1939DcmSwcSRDataElementValueRef.	
Template Description	
This reference represents the related diagnostic data element.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvDataCondition.dataElement	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00274

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939DcmModeCondition
BSW Parameter	**BSW Type**
J1939DcmSwcSRDataElementValueRef	EcucForeignReferenceDef
BSW Description	
Reference to a constant specification defining the compare value for environmental condition.	
Template Description	
This attribute represents a fixed compare value taken to evaluate the compare condition.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvDataCondition.compareValue	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00275

BSW Module	**BSW Context**
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions

BSW Parameter	BSW Type
J1939DcmModeRule	EcucParamConfContainerDef
BSW Description	
This container contains the configuration of a mode rule which represents a logical expression with J1939DcmModeConditions or other J1939DcmModeRules as arguments. All arguments are processed with the operator defined by DcmLogicalOperator, for instance: Argument_A AND Argument_B AND Argument_C	
Template Description	
A DiagnosticEnvConditionFormula embodies the computation instruction that is to be evaluated at runtime to determine if the DiagnosticEnvironmentalCondition is currently present (i.e. the formula is evaluated to true) or not (otherwise). The formula itself consists of parts which are combined by the logical operations specified by DiagnosticEnvConditionFormula.op. If a diagnostic functionality cannot be executed because an environmental condition fails then the diagnostic stack shall send a negative response code (NRC) back to the client. The value of the NRC is directly related to the specific formula and is therefore formalized in the attribute DiagnosticEnvConditionFormula.nrcValue.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvCondition Formula	
Mapping Rule	**Mapping Type**
A DcmModeRule is mapped to a DiagnosticEnvConditionFormula, if this Dcm ModeRule is a subrule, i.e. it is referenced by a DcmArgumentRef. In addition, a new DiagnosticEnvironmentalCondition shall be created with Diagnostic EnvironmentalCondition.formula containing a DiagnosticEnvConditionFormula. In both cases, if no DcmLogicalOperator is present in this DcmModeRule, then DiagnosticEnvConditionFormula shall be set to DiagnosticLogicalOperatorEnum.logicalAnd.	full
Mapping Status	**Mapping ID**
valid	

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939Dcm ModeRule
BSW Parameter	**BSW Type**
J1939DcmArgumentRef	EcucChoiceReferenceDef
BSW Description	
This is a choice reference either to a mode condition or a an other mode rule serving as subexpression.	
Template Description	
A DiagnosticEnvConditionFormulaPart can either be a atomic condition, e.g. a DiagnosticEnvCompareCondition, or a DiagnosticEnvConditionFormula, again, which allows arbitrary nesting.	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvCondition FormulaPart	
Mapping Rule	**Mapping Type**
Depending on the destination, one DcmArgumentRef is mapped to a Diagnostic EnvConditionFormula if "destination" is a DcmModeRule, and to a Diagnostic EnvCompareCondition, if "destination" is a DcmModeCondition. The order of the aggregation of the DiagnosticEnvConditionFormulaParts shall correspond to the ordering of the index of the DcmArgumentRefs.	full

Mapping Status	Mapping ID
valid	up_Dcm_00264

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939Dcm ModeRule/J1939DcmLogicalOperator
BSW Parameter	**BSW Type**
J1939DCM_AND	EcucEnumerationLiteralDef
BSW Description	
Template Description	
DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticLogicalOperatorEnum.logicalAnd**: Logical AND	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticLogicalOperatorEnum.logicalAnd	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00261

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939Dcm ModeRule/J1939DcmLogicalOperator
BSW Parameter	**BSW Type**
J1939DCM_OR	EcucEnumerationLiteralDef
BSW Description	
Template Description	
DiagnosticEnvConditionFormula.op: This attribute represents the concrete operator (supported operators: and, or) of the condition formula. **DiagnosticLogicalOperatorEnum.logicalOr**: Logical OR	
M2 Parameter	
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvConditionFormula.op, DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticLogicalOperatorEnum.logicalOr	
Mapping Rule	**Mapping Type**
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00262

BSW Module	BSW Context
J1939Dcm	J1939Dcm/J1939DcmConfigSet/J1939DcmProcessingConditions/J1939Dcm ModeRule

BSW Parameter	BSW Type
J1939DcmModeRuleNrcValue	EcucIntegerParamDef

BSW Description
Optional parameter which defines the NRC to be sent in case the mode rule condition is not valid.

Template Description
This attribute represents the concrete NRC value that shall be returned if the condition fails.

M2 Parameter
DiagnosticExtract::Dcm::DiagnosticService::EnvironmentalCondition::DiagnosticEnvCondition Formula.nrcValue

Mapping Rule	Mapping Type
1:1 mapping	full
Mapping Status	**Mapping ID**
valid	up_Dcm_00263

F Splitable Elements in the Scope of this Document

This chapter contains a table of all model elements stereotyped ≪atpSplitable≫ in the scope of this document.

Each entry in Table F.1 consists of the identification of the specific model element itself and the applicable value of the tagged value atp.Splitkey.

For more information about the concept of splitable model elements and how these shall be treated please refer to [22].

Name of splitable element	Splitkey
DiagnosticAging.agingCycle	agingCycle, variationPoint.ShortLabel
DiagnosticContributionSet.commonProperties	commonProperties
DiagnosticContributionSet.element	element, variationPoint.shortLabel
DiagnosticContributionSet.serviceTable	serviceTable, variationPoint.shortLabel
DiagnosticDataIdentifier.dataElement	dataElement, variationPoint.shortLabel
DiagnosticEcuInstanceProps.ecuInstance	ecuInstance
DiagnosticEnableConditionGroup.enableCondition	enableCondition, variationPoint.shortLabel
DiagnosticEvent.connectedIndicator	shortName, variationPoint.shortLabel
DiagnosticInfoType.dataElement	dataElement
DiagnosticParameter.dataElement	shortName, variationPoint.shortLabel
DiagnosticParameterIdentifier.dataElement	dataElement, variationPoint.shortLabel
DiagnosticProtocol.diagnosticConnection	diagnosticConnection, variationPoint.shortLabel
DiagnosticProtocol.serviceTable	serviceTable, variationPoint.shortLabel
DiagnosticSecurityAccess.securityLevel	securityLevel
DiagnosticServiceTable.diagnosticConnection	diagnosticConnection, variationPoint.shortLabel
DiagnosticStorageConditionGroup.storageCondition	storageCondition, variationPoint.shortLabel
DiagnosticTroubleCodeGroup.dtc	dtc, variationPoint.shortLabel
DiagnosticTroubleCodeProps.extendedDataRecord	shortName, variationPoint.shortLabel
DiagnosticTroubleCodeProps.freezeFrame	shortName, variationPoint.shortLabel

Table F.1: Usage of splitable elements

G Variation Points in the Scope of this Document

This chapter contains a table of all model elements stereotyped ≪atpVariation≫ in the scope of this document.

Each entry in Table G.1 consists of the identification of the model element itself and the applicable value of the tagged value vh.latestBindingTime.

For more information about the concept of variation points and how model elements that contain variation points shall be treated please refer to [22].

Variation Point	Latest Binding Time
DiagnosticAging.agingCycle	preCompileTime
DiagnosticAging.threshold	preCompileTime
DiagnosticCommonProps	codeGenerationTime
DiagnosticContributionSet.element	postBuild
DiagnosticContributionSet.serviceTable	postBuild
DiagnosticDataIdentifier.dataElement	postBuild
DiagnosticEnableConditionGroup.enableCondition	postBuild
DiagnosticEvent.connectedIndicator	postBuild
DiagnosticEvent.eventFailureCycleCounterThreshold	postBuild
DiagnosticFreezeFrame.recordNumber	preCompileTime
DiagnosticIndicator.healingCycleCounterThreshold	preCompileTime
DiagnosticParameter.dataElement	postBuild
DiagnosticParameterIdentifier.dataElement	postBuild
DiagnosticProtocol.diagnosticConnection	postBuild
DiagnosticProtocol.serviceTable	postBuild
DiagnosticRoutine.id	preCompileTime
DiagnosticServiceTable.diagnosticConnection	postBuild
DiagnosticStorageConditionGroup.storageCondition	postBuild
DiagnosticTroubleCodeGroup.dtc	postBuild
DiagnosticTroubleCodeGroup.groupNumber	preCompileTime
DiagnosticTroubleCodeObd.considerPtoStatus	preCompileTime
DiagnosticTroubleCodeObd.obdDTCValue	preCompileTime
DiagnosticTroubleCodeProps.extendedDataRecord	preCompileTime
DiagnosticTroubleCodeProps.freezeFrame	preCompileTime

Table G.1: Usage of variation points