




ERJU SYSTEM PILLAR

# **System Interface Description\_SDI-GEN information model (SERA Version)**



# System Interface Description\_SDI-GEN information model (SERA Version)

Author(s)	Karl-Albrecht Klinge , Wegele, Stefan (SMO RI ML ADC I&C) , Stalin Chriss Castelino
Abstract	This document describes the as required per SPPRAMSS-349 - <a href="#">EN 50126-1:2017</a> phase 5 (Architecture and apportionment of system requirements) between the and the .
Config Item	System Interface Description
Document ID	TCCS Service Function Diagnostics _SFD_ L5/TCCS System Interface SDI-GEN information model_SERA Version#723869  System Interface Description_SDI-GEN information model (SERA Version)
Classification	Public
Status	In Review by System Pillar
Version	1.0
Revision	723869
Last Change Date	02.10.2025
Copyright	Brussels: Europe's Rail Joint Undertaking, 2025

© Europe's Rail Joint Undertaking, 2025

This document is drafted by and belongs to EU Rail.

EU Rail encourages the distribution and re-use of this document, the technical specifications and the information it contains. EU Rail holds several intellectual property rights, such as copyright and trade mark rights, which need to be considered when this document is used.

EU Rail authorises you to re-publish, re-use, copy and store this document without changing it, provided that you indicate its source and include the following: EU Rail trade mark, title of the document, year of publication, version of document.

EU Rail makes no representation or warranty as to the accuracy or completeness of the information contained within these documents. EU Rail shall have no liability to any party as a result of the use of the information contained herein. EU Rail will have no liability whatsoever for any indirect or consequential loss or damage, and any such liability is expressly excluded.

You may study, research, implement, adapt, improve and otherwise use the information, the content and the models in the this document for your own purposes. If you decide to publish or disclose any adapted, modified or improved version of this document, any amended implementation or derivative work, then you must indicate that you have modified this document, with a reference to the document name and the terms of use of this document. You may not use EU Rail's trade marks or name in any way that may state or suggest, directly or indirectly, that EU Rail is the author of your adaptations.

EU Rail cannot be held responsible for your product, even if you have used this document and its content. It is your responsibility to verify the quality, completeness and the accuracy of the information you use, for your own purposes.


**This work is currently a work in progress. The content presented is subject to change as it undergoes further review, refinement, and development. Please do not consider this version as final or authoritative.**

INFO: History table is not displayed, because this document is in status **doc\_contentApproval**.


RULE: History table is not displayed, in statuses: { draft doc\_open doc\_inprogress doc\_contentApproval doc\_contentDecision }

CONTACT: For more information contact Administrator

## Review description

Attachments	<a href="#">REMINDER_[ERJU SP] Request to review SC2.4 List of deliverables - Task 2_ Transversal Systems .pdf</a> , <a href="#">Review and Approval Jens Kilian.pdf</a> , <a href="#">Review and Approval Virgil Lostun.pdf</a>
Approvals	Kilian Jens : Waiting , SANGO Marc (SNCF / DIR TECHNOLOGIES INNOVATION ET PROJETS GROUPE / IR DIR RECHERCHE - PSF) : Waiting , DE NICOLA, Giuseppe : Waiting , KEFALAS, Georgios : Waiting , Julien Bois : Waiting , Oliver Knapp : Waiting , Wischy, Markus Alexander (SMO RI R&D F IL) : Waiting , HENON Frédéric : Waiting , TEKE, Emre : Waiting , Renato Rodrigues : Waiting , IOVINO, Salvatore : Waiting , Davinder Bhatia : Waiting , BITSCH Friedemann : Waiting , Roman R Treydel : Waiting , Golebniak, Udo (SMO RI ML ADC I&C) : Waiting , Mirko Blazic : Waiting , Benameur, Malik (SMO NEE RC-CH RI PLM SYS) : Waiting , MOTTOLA, Giuseppe Diodato : Waiting , Jack Schneider : Waiting , Zeeshan Z Ansar : Waiting , LOSTUN Virgil : Waiting , Patrick Konix : Waiting , NANNI Marco : Waiting , DE MARCO TELESE Giancarlo : Waiting , Tione, Roberto : Waiting , Andreeva-Moschen Emilia (HOLDING) : Waiting
Type of Approval	 Document Review

## Approval description

Attachments	<a href="#">REMINDER__[ERJU SP] Request to review SC2.4 List of deliverables - Task 2_ Transversal Systems .pdf</a> , <a href="#">Review and Approval Jens Kilian.pdf</a> , <a href="#">Review and Approval Virgil Lostun.pdf</a>
Approvals	LOSTUN Virgil : Waiting
Type of Approval	 Document Approval

1 Preamble	4
1.1 Scope and intended audience	4
1.2 Purpose	5
1.3 Glossary	5
2 Overview	6
2.1 Overall description	6
2.2 Non-functional characteristics / non-functional requirements	6
3 SDI-GEN interface requirements	6
3.1 SDI-GEN conceptual data description	6
3.1.1 Physical Model using the Equipment model	6
3.2 SDI-GEN Information Model	12
3.2.1 Package "SDI-GEN"	12
3.2.1.1 Package Header	12
3.2.1.2 Equipment Model	12
3.2.1.3 Subsystem Model	17
3.2.1.4 Interface Model	19
3.2.1.5 Motor Turn Data Model	21
3.2.1.6 Log Model	21
3.2.1.7 Redundancy Model	22
3.2.1.8 Meta Data Table	23

3.2.1.9 Data Model	23
3.2.1.10 NodeSet2 OPC UA Information Model	23
3.2.2 Alternative View of the SDI-GEN Information Model	23
4 Appendix	23
4.1 Input documents	23
4.2 Standards and References	24

[Figure 1. Tree View of Equipment](#)  
[Figure 2. Tree View of Subsystem](#)  
[Figure 3. Tree View of Interface](#)  
[Figure 4. Tree View of MotorTurnData](#)  
[Figure 5. Tree View of LogEvent](#)  
[Figure 6. Tree View of RedundancyStatus](#)  
[Figure 7. Tree View of RedundancyGroup](#)

## 1 Preamble

### 1.1 Scope and intended audience

This document is provided for demonstration purposes only. The TACS Interface specification SDI Generic is the applicable specification. The content, format and structure shown herein are intended solely to illustrate the toolchain to generate future versions of the applicable specification. All information is subject to change without notice and does not represent a finalised or authorised specification.

The information model presented in this document is subject to change due to ongoing development of the TCCS toolchain, Polarion template and formatting. The information model could be analysed by reading this document. For a comprehensive analysis of the artefacts generated by the TCCS toolchain, please refer to the alternative view of the information model.

This document defines the requirements for the Standard Diagnostic Interface (SDI) of a generic diagnosable building block (SDI-GEN) to the Service Function Diagnostics based on the OPC UA protocol, as outlined in Part 1 of the SDI Companion Specification.

This specification is derived from and aligned with the following key documents:

- The System Definition, which describes the overall architecture and objectives of the SDI-based integration approach.
- The System Requirements Specification, which outlines functional, operational, and non-functional expectations for SDI implementations.
- The System Architecture Definition, which provides the structural blueprint for how system components interact, communicate, and are organized.

Intended Audience:

- System Architects and Integrators, responsible for the design and alignment of interoperable interfaces and data models across systems.
- Product Manufacturers and Suppliers, especially those contributing equipment or software components that expose OPC UA-based interfaces according to SDI standards.

- OPC UA Modelers and Developers, implementing companion models in NodeSet2 XML, transforming UML diagrams into OPC UA Information Models, and maintaining semantic consistency.
- Operators and Maintainers, who rely on standardized and interpretable data interfaces for diagnostics, monitoring, and operational decision-making.
- Standards Bodies and Certification Authorities, reviewing and validating SDI-based models for compliance, compatibility, and long-term maintainability.

The scope of this document covers the detailed structure of the NodeSet2 file and the information model for the generic diagnosable building block (SDI-GEN). This document builds on general modeling and communication requirements, including semantic standardization, reference modeling, versioning strategies, and structural adaptation rules for mapping domain-specific models into OPC UA, which are provided in separate parts of the companion specification (e.g., SDI-XX and Part 1 SDI Specification documents).

## 1.2 Purpose

This document describes the Standard Diagnostic Interface - Generic (SDI-GEN) in accordance with the requirements of – EN 50126-1:2017, Phase 5: Architecture and Apportionment of System Requirements. The described interface defines the communication and data exchange between the service function diagnostics and a diagnosable building block, forming a fundamental part of the system architecture.

The purpose of this document is to provide:

- A general and sufficiently detailed description of the system interface;
- Justification for the architectural and design decisions taken;
- A clear explanation of how the interface contributes to meeting the overall system requirements;
- A basis for consistent implementation, integration, and verification activities across different suppliers or subsystems.

This document aims to ensure a shared understanding of the architectural principles and interface structure, focusing on the rationale that led to the selected approach. It supports traceability from high-level system requirements to concrete technical solutions, particularly those realized using OPC UA.

Note: This Interface Definition may cover specific layer(s) of the overall system interface stack where appropriate, allowing reuse across multiple subsystems or interface definitions.

Depending on the context:

- For external interfaces, it serves as an extension of the System Definition, describing interactions from the perspective of the System Under Consideration (SuC).
- For internal interfaces, it complements the System Architecture Document, detailing the internal communication structure within the SuC.

This document is a key input for the development of product-specific OPC UA companion specifications and information models (e.g., SDI-P), and provides the foundation for implementing semantically consistent OPC UA-based interfaces across the system.

## 1.3 Glossary

This section provides definitions for key terms and abbreviations used throughout this document. All definitions are based on the official System Pillar Glossary. To ensure consistency and traceability, terms

referenced here are aligned with existing definition work items and maintained according to the Glossary Usage Guidelines.

<b>SDI – Standard Diagnostic Interface</b>	A standardized OPC UA-based interface framework for the integration of diagnostic, condition monitoring, and asset-related data across different subsystems and suppliers.
<b>OPC UA – OPC Unified Architecture</b>	A machine-to-machine communication protocol for industrial automation, designed for interoperability, scalability, and platform-independence.
<b>NodeSet2</b>	The XML-based format used to describe OPC UA Information Models for exchange between tools and implementations.
<b>RedundancyGroup</b>	A functional entity used to organize and monitor multiple redundant system components, ensuring availability through redundancy strategies.

## 2 Overview

### 2.1 Overall description

The interface is based on OPC UA defining the application layer and using the lower levels of the TCP/IP stack.

The SDI-GEN information model consists of the Equipment Model, Interface model, Subsystem model, MotorTurnData model, Log model and Redundancy Model.

The information model is described by:

1. A class diagram of the model
2. CCS/TMS data model format representation of the model
3. Table representation of the model meta data for the attributes and classes
4. Combined NodeSet2 XML OPC UA Information Model

The SDI-XX requirements apply.

### 2.2 Non-functional characteristics / non-functional requirements

Decisions and rationales

## 3 SDI-GEN interface requirements

The  SPT2TS-129409 - [System Structure and Interface Modeling](#) apply [SPT2TS-130217 ]

### 3.1 SDI-GEN conceptual data description

#### 3.1.1 Physical Model using the Equipment model

An equipment refers to a procurable unit provided by a manufacturer, uniquely identified by its serial number. This unique identification ensures traceability and precise management of individual units across their lifecycle.

Key Characteristics of an Equipment:

### 1. Procurable Unit:

- equipment is a tangible unit that can be procured directly from the manufacturer.
- example: A track switch motor or a signaling device.

### 2. Serial Number:

- Each equipment is assigned a unique serial number by the manufacturer.
- This identifier supports traceability, warranty claims, and maintenance history tracking.

### 3. Component Hierarchy (Parent-Child Relationship):

- Equipments can consist of other equipments as their components, if the components themselves are also procurable units.
- This hierarchical structure supports modular maintenance and replacement strategies.
- Example:
  - Parent Equipment: Point
  - Child Equipments: Point Motor
  - Child Equipments: Gearbox, mechanical joints, ... (if all separately procurable!).

### 4. Material Type:

- The type of equipment is defined by its material, which represents a standard or catalogued item that can be ordered.
- Material type serves as the blueprint for the equipment, specifying its technical specifications and functional characteristics.
- Example:
  - Material Type: "Hydraulic Actuator Type A."
  - Equipments: All hydraulic actuators of this type, each with a unique serial number.

### [SPT2TS-130551 ]

Many widespread standard components of equipment have already been modeled to ensure interoperability, consistency, and reusability across different systems. These components represent common building blocks found in most equipment and are essential for standardization.

### Common Standardized Components:

#### 1. Controller

- Role: The central processing unit responsible for controlling the equipment's functionality and interfacing with other systems.
- Examples: Embedded microcontrollers, programmable logic controllers (PLCs), or custom control units.

#### 2. Physical Network Interface

- Role: Provides connectivity to communication networks for data exchange.
- Examples: Ethernet ports, Wi-Fi modules, Bluetooth modules, or fieldbus interfaces (e.g., CAN, PROFINET).

#### 3. Storage Medium

- Role: Enables data storage for operational logs, configuration files, or temporary data buffers.
- Examples: Flash memory, solid-state drives (SSDs), or removable storage like SD cards.

#### 4. Power Supply

- Role: Supplies power to the equipment and its components, often including conversion or regulation functions.
- Examples: AC-to-DC converters, battery packs, or uninterruptible power supplies (UPS).

## 5. Physical Outputs

- Role: Convert digital or analog signals into real-world actions or states.
- Types:
  - Digital Outputs: Binary signals, such as activating relays or indication LEDs.
  - Analog Outputs: Continuous signals, such as adjusting motor speed or controlling valve positions.

## 6. Physical Inputs

- Role: Accept external signals for processing or decision-making by the equipment.
- Types:
  - Digital Inputs: Binary states, such as detecting limit switches or push buttons.
  - Analog Inputs: Variable signals, such as temperature sensors or pressure transducers.

## 7. Input Switch or Button

- Role: Manual control elements for triggering actions or settings within the equipment.
- Examples: Push buttons, toggle switches, or rotary selectors.

### Benefits of Pre-Modeled Standard Components:

1. Interoperability: Ensures seamless integration with other systems and equipment following the same standards.
2. Efficiency in Design and Deployment: Reusing standardized components reduces development and integration time.
3. Scalability: Allows for easy addition or replacement of components without re-engineering the entire system.
4. Maintenance Simplification: Standard components with consistent interfaces make troubleshooting, replacement, and upgrades straightforward. A maintainer could bring an equipment of the same material that has failed

### [SPT2TS-130552 ]

The Equipment model offers flexibility to represent systems at different levels of granularity based on how components are designed to be replaced or managed. Below are examples to illustrate its application:

#### Example 1: Object Controller with Three Integrated Controllers (Non-Replaceable Individually)

- Scenario: An object controller contains three Controllers that are tightly integrated and replaced as a single unit.
- Modeling Approach:
  - The object controller is modeled as a single Equipment with a HasComponent relationship to three Controllers.
  - The Controllers are not separate Equipments but are part of the object controller's internal structure.

Visualization:

```

Object Controller (Equipment)
├── Controller 1 (HasComponent)
├── Controller 2 (HasComponent)
└── Controller 3 (HasComponent)
  
```



## Example 2: Object Controller with Three Replaceable Controllers (Individually Replaceable)

- Scenario: An object controller is designed with three Controllers that can be replaced individually (Line Replaceable Unit - LRU) without replacing the entire unit.
- Modeling Approach:
  - The object controller is modeled as a parent Equipment.
  - Each of the three Controllers is modeled as a child Equipment with its own serial number and potentially its own material type.

Visualization:

```

Object Controller (Parent Equipment)
├── Controller Unit 1 (Child Equipment)
│   └── Controller (HasComponent)
├── Controller Unit 2 (Child Equipment)
│   └── Controller (HasComponent)
└── Controller Unit 3 (Child Equipment)
    └── Controller (HasComponent)
  
```

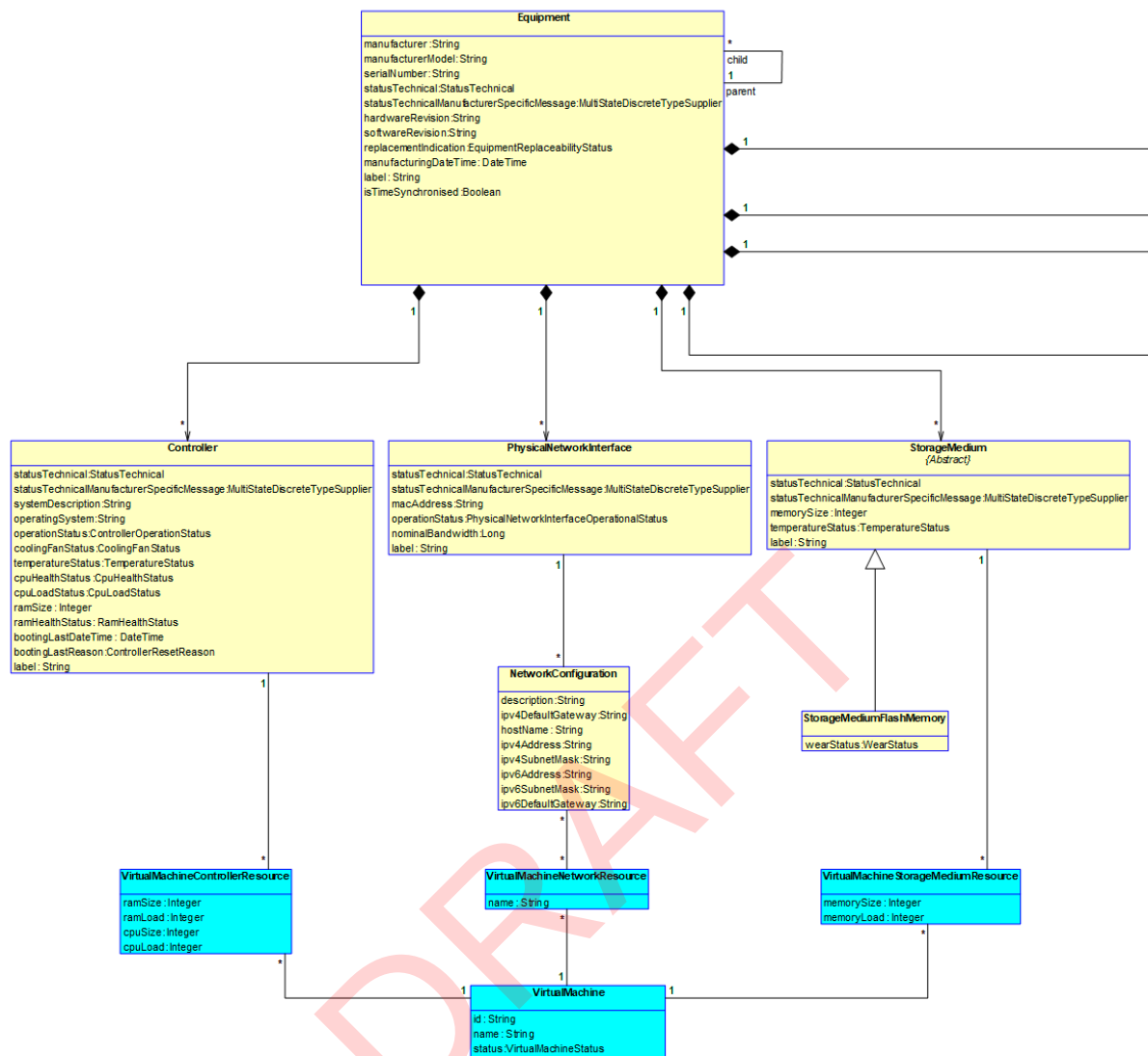
### Practical Considerations:

- Procurement and Inventory:
  - In Example 1, the object controller is ordered as a single unit.
  - In Example 2, each Controller can be procured and stocked individually, reducing spare part costs.
- Maintenance Planning:
  - Example 1 simplifies maintenance procedures but may require longer downtime.
  - Example 2 allows for more targeted repairs, reducing overall system downtime.
- Traceability and Asset Management:
  - Example 1 tracks the object controller as a single asset.
  - Example 2 tracks each Controller independently, enabling detailed performance and maintenance analytics.

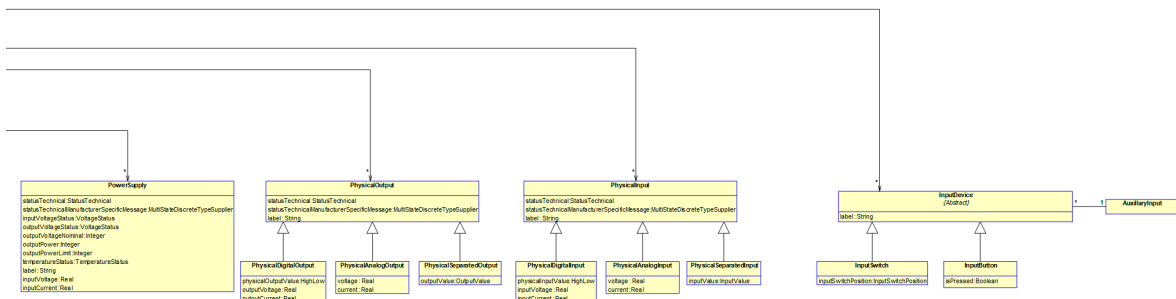
**The Equipment model allows to describe different system architectures without a need to standardize the architecture itself.**

[SPT2TS-130546 ]

## Static class diagram of the equipment model



more standard components of equipments:



[SPT2TS-130547 ]

## Virtual Machines

A Virtual Machine (VM) represents a logical partition of the resources of a physical computer, enabling

multiple isolated systems to run concurrently on shared hardware. A hypervisor is responsible for managing the resource allocation and isolating each virtual machine. The model for Virtual Machines includes distinct resource components assigned to a Virtual Computer.

### Key Concepts:

#### 1. Virtual Machine (VM):

- A virtualized instance of a computer system.
- Operates as if it were an independent computer with dedicated resources.

#### 2. Physical Computer:

- The underlying hardware providing computational, storage, and network resources to VMs.

#### 3. Hypervisor:

- A software layer that manages the creation, allocation, and isolation of resources for VMs.
- Assigns and controls resource slices for each VM.

#### 4. Virtual Computer:

- Represents the abstracted “machine” formed by combining allocated resources.
- Functions as an operational unit for applications and processes.

### Resource Components Assigned by the Hypervisor (see blue classes in the class diagram):

#### 1. VirtualMachineControllerResource:

- Represents the virtualized processing capabilities of the VM.
- Includes allocated CPU cores, thread scheduling, and associated processing power and RAM.
- Example: A VM might be assigned 4 out of 16 physical CPU cores.

#### 2. VirtualMachineNetworkResource:

- Virtualized network interfaces allowing communication over physical or virtual networks.
- Includes bandwidth, IP address management, and routing capabilities.
- Example: A VM is assigned a virtual NIC connected to a specific VLAN or subnet.

#### 3. VirtualMachineStorageMediumResource:

- Represents the allocated virtual storage for the VM.
- Includes virtual hard drives (VHD), storage quotas, and access to shared storage pools.
- Example: A VM might have 100 GB of virtual storage mapped to an SSD pool on the physical computer.

[SPT2TS-130548 ]

ToDo

VM > Discuss with Computing Environment

Discuss model to be used for CCS on-board application [SPT2TS-130549 ]

## 3.2 SDI-GEN Information Model

### 3.2.1 Package "SDI-GEN"

#### 3.2.1.1 Package Header

##### SPT2TS-131276 - Package specification

```
{
  "$schema": "prefix",
  "isDefinedBy": "http://rail-research.europa.eu/eu-rail.sdi-generic.4.3.1-r02/",
  "name": "sdi_g",
  "prefix": "sdi_g",
  "version": "4.3.1-r02",
  "intId": 20,
  "enums": [], "structs": []
}
```

---

#### 3.2.1.2 Equipment Model


##### SPMS-7352 - Equipment

The equipment class is used to represent the physical view of the system. Equipment classes represent unique instances down to at least the line replaceable units (hierarchical structure of equipment classes, parent-child). Linking multiple equipment classes allows manufacturers to represent their specific system.


DRAFT

## Properties


The object owns the properties listed below:

**controller** :  SPMS-7351 - Controller [ 0 .. \* ]


No description available.

**subEquipment** :  SPMS-7352 - Equipment [ 0 .. \* ]


children - parent relation

**inputButton** :  SPMS-7354 - InputButton [ 0 .. \* ]


No description available.

**inputSwitch** :  SPMS-7355 - InputSwitch [ 0 .. \* ]


No description available.

**physicalAnalogInput** :  SPMS-7369 - PhysicalAnalogInput [ 0 .. \* ]


No description available.

**physicalDigitalInput** :  SPMS-7370 - PhysicalDigitalInput [ 0 .. \* ]


No description available.

**physicalSeparatedInput** :  SPMS-7371 - PhysicalSeparatedInput [ 0 .. \* ]


No description available.

**physicalNetworkInterface** :  SPMS-7372 - PhysicalNetworkInterface [ 0 .. \* ]


No description available.

**physicalAnalogOutput** :  SPMS-7374 - PhysicalAnalogOutput [ 0 .. \* ]


No description available.

**physicalDigitalOutput** :  SPMS-7375 - PhysicalDigitalOutput [ 0 .. \* ]

No description available.

**physicalSeparatedOutput** :  SPMS-7376 - PhysicalSeparatedOutput [ 0 .. \* ]

No description available.

**powerSupply** :  SPMS-7377 - PowerSupply [ 0 .. \* ]

No description available.

**storageMediumFlashMemory** :  SPMS-7382 - StorageMediumFlashMemory [ 0 .. \* ]

No description available.

**replacementIndication** : EquipmentReplaceabilityStatus

Indicates when the equipment requests a replacement. The decision to act on this indication is up to the operator, in accordance with the equipment manual.

EquipmentReplaceabilityStatus enumeration values:

Value	Enumeration Literal	Description
0	Unknown	The status unknown is used when the state is not yet established e.g. if connection to the system is lost
1	ReplacementNot Needed	Equipment does not need to be replaced
2	ReplaceableAtOperation	Equipment could be replaced during operation
3	ReplaceableMaintenance	Equipment should be replaced during maintenance
4	ReplaceableRevalidation	Equipment should be replaced during revalidation

This property has the following additional attributes:

Property	Value
type	Diagnosis : ContentType
category	Equipment : ContentCategory
update	OnChange : ClientUpdateSubscriptions

Property	Value
temporalAccuracy	1 s
samplingFrequency	1 Hz
dataCollectionInterval	1 s
dataBufferingTime	1 s

**statusTechnical** : StatusTechnical

Technical Status of the system, that represents the aggregated status of all hierarchical lower systems. This allows to have a top level information on the status of the system which can be drilled down if the system is in any other state than "Ok".

StatusTechnical enumeration values:

Value	Enumeration Literal	Description
0	Unknown	The status unknown is used when the state is not yet established e.g. if connection to the system is lost
1	Ok	System serves all primary functions and has no deviations, errors or failures
2	Warning	All subsystems are working as intended, but the system detects unexpected behaviour (e.g. deviation from expected values).
3	FailureNonCritical	At least one error in one of the (sub)systems, but on this system level all functions are available
4	FailureCritical	At least one function is not available; operational consequences possible

This property has the following additional attributes:

Property	Value
type	Diagnosis : ContentType
category	Equipment : ContentCategory
update	OnChange : ClientUpdateSubscriptions
temporalAccuracy	1 s
samplingFrequency	1 Hz
dataCollectionInterval	1 s
dataBufferingTime	1 s

**hardwareRevision** : string

The data point hardwareRevision indicates the hardware revision level of the equipment. Hardware can only be changed by replacing an instance of Equipment.

This property has the following additional attributes:

Property	Value
type	ConfigurationParameter : ContentType
category	Equipment : ContentCategory

Property	Value
update	OnChange : ClientUpdateSubscriptions
temporalAccuracy	1 s
samplingFrequency	1 Hz
dataCollectionInterval	1 s
dataBufferingTime	1 s

**label** : string [ 0 .. 1 ]

It is assigned to all classes representing physically identifiable entities. This string, corresponding to a physically identifiable label, facilitates consistent reference between the physical entities in the field and their digital representations within the model

This property has the following additional attributes:

Property	Value
type	ConfigurationParameter : ContentType
category	Equipment : ContentCategory
update	OnChange : ClientUpdateSubscriptions
temporalAccuracy	1 s
samplingFrequency	1 Hz
dataCollectionInterval	1 s
dataBufferingTime	1 s

**manufacturer** : string

The name of the manufacturer of the equipment.

This property has the following additional attributes:

Property	Value
type	ConfigurationParameter : ContentType
category	Equipment : ContentCategory
update	OnChange : ClientUpdateSubscriptions
temporalAccuracy	1 s
samplingFrequency	1 Hz
dataCollectionInterval	1 s
dataBufferingTime	0 s

**manufacturerModel** : string

The name of the equipment model

This property has the following additional attributes:

Property	Value
type	ConfigurationParameter : ContentType
category	Equipment : ContentCategory
update	OnChange : ClientUpdateSubscriptions
temporalAccuracy	1 s
samplingFrequency	1 Hz

Property	Value
dataCollectionInterval	1 s
dataBufferingTime	0 s

**serialNumber** : string

Number defined and provided by the manufacturer. The serial number combined with the manufacturer information must be unique.

This property has the following additional attributes:

Property	Value
type	ConfigurationParameter : ContentType
category	Equipment : ContentCategory
update	OnChange : ClientUpdateSubscriptions
temporalAccuracy	1 s
samplingFrequency	1 Hz
dataCollectionInterval	1 s
dataBufferingTime	0 s

**softwareRevision** : string

The data point softwareRevision (not interface revision) indicates the software revision level of the equipment. It contains the information to identify all software components, including firmware. It does not include changes in the configuration data. The format and semantics are defined by the manufacturer.

This property has the following additional attributes:

Property	Value
dataCollectionInterval	1 s
dataBufferingTime	1 s
temporalAccuracy	1 s
samplingFrequency	1 Hz
type	ConfigurationParameter : ContentType
category	Equipment : ContentCategory
update	Undefined : ClientUpdateSubscriptions

**manufacturingDateTime** : timestamp

Indicates the production date of the equipment

This property has the following additional attributes:

Property	Value
type	ConfigurationParameter : ContentType
category	Equipment : ContentCategory
update	OnChange : ClientUpdateSubscriptions
temporalAccuracy	1 s
samplingFrequency	1 Hz
dataCollectionInterval	1 s



Property	Value
dataBufferingTime	0 s

**statusTechnicalManufacturerSpecificMessage** : uint16 [ 0 .. 1 ]

Must be used by the supplier to describe the reasons for a StatusTechnical != OK, that cannot be explained by existing datapoints (NOT including IM and manufacturer specific diagnostic messages). This Information MUST be provided from the supplier. This should provide flexibility for future uses. Multiple states can be indicated at the same time if multiple diagnosis have not been included in the model during the design phase. The supplier specific reason may not overlap with reasons already covered in other attributes.

This property has the following additional attributes:

Property	Value
dataCollectionInterval	1 s
dataBufferingTime	1 s
temporalAccuracy	1 s
samplingFrequency	1 Hz
type	Diagnosis : ContentType
category	Equipment : ContentCategory
update	Undefined : ClientUpdateSubscriptions

**isTimeSynchronised** : boolean [ 0 .. 1 ]

Indicates whether the last time synchronisation was successful or not. This is important for the subsystem communication and diagnostic data gathering. TRUE: current time of this subsystem is synchronised. This property has the following additional attributes:

Property	Value
type	Diagnosis : ContentType
category	Equipment : ContentCategory
update	OnChange : ClientUpdateSubscriptions
temporalAccuracy	1 s
samplingFrequency	1 Hz
dataCollectionInterval	1 s
dataBufferingTime	1 s

Tree View  
Diagram



Figure 1 Tree View of Equipment

### 3.2.1.3 Subsystem Model

#### SPMS-7383 - Subsystem

a subsystem for operational or service functions

## Properties

The object owns the properties listed below:

**statusTechnical** : StatusTechnical

Technical Status of the system, that represents the aggregated status of all hierarchical lower systems. This allows to have a top level information on the status of the system which can be drilled down if the system is in any other state than "Ok"

StatusTechnical enumeration values:

Value	Enumeration Literal	Description
0	Unknown	The status unknown is used when the state is not yet established e.g. if connection to the system is lost
1	Ok	System serves all primary functions and has no deviations, errors or failures
2	Warning	All subsystems are working as intended, but the system detects unexpected behaviour (e.g. deviation from expected values).
3	FailureNonCritical	At least one error in one of the (sub)systems, but on this system level all functions are available
4	FailureCritical	At least one function is not available; operational consequences possible

**subsystemIdentification** : string

The technical identifier of the subsystem or adjacent systems (see Eu.SAS.77 in [Eu.Doc.16]).

**statusTechnicalManufacturerSpecificMessage** : uint16 [ 0 .. 1 ]

Must be used by the supplier to describe the reasons for a StatusTechnical != OK, that cannot be explained by existing datapoints (NOT including IM and manufacturer specific diagnostic messages). This Information MUST be provided from the supplier. This should provide flexibility for future uses. Multiple states can be indicated at the same time if multiple diagnosis have not been included in the model during the design phase. The supplier specific reason may not overlap with reasons already covered in other attributes.

**isTimeSynchronised** : boolean [ 0 .. 1 ]

Indicates whether the last time synchronisation was successful or not. This is important for the subsystem communication and diagnostic data gathering. TRUE: Current time of this subsystem is synchronised

## Tree View Diagram

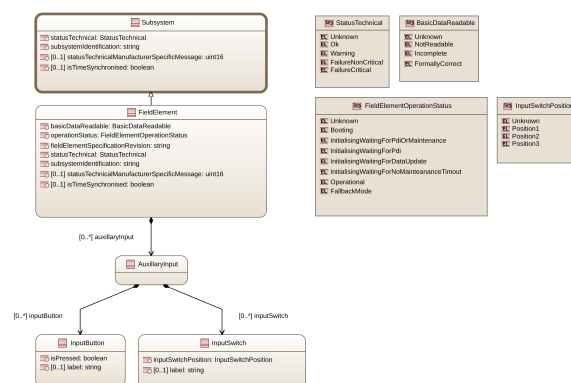


Figure 2 Tree View of Subsystem

### 3.2.1.4 Interface Model

#### **SPMS-7356 - Interface**

An interface between logical components

DRAFT

## Properties

The object owns the properties listed below:

**connectionStatus** : ConnectionStatus

Indicates the overall connection status of the endpoint

ConnectionStatus enumeration values:

Value	Enumeration Literal	Description
0	Unknown	The status unknown is used when the state is not yet established e.g. if connection to the system is lost
1	NotAvailableNotConnected	The subsystem is neither available nor connected
2	AvailableNotConnected	The subsystem is available but not connected
3	Connected	The subsystem is connected
4	NotConnectedDisturbed	The subsystem is not connected and disturbed

**statusTechnical** : StatusTechnical

Technical Status of the system, that represents the aggregated status of all hierarchical lower systems. This allows to have a top level information on the status of the system which can be drilled down if the system is in any other state than "Ok".

StatusTechnical enumeration values:

Value	Enumeration Literal	Description
0	Unknown	The status unknown is used when the state is not yet established e.g. if connection to the system is lost
1	Ok	System serves all primary functions and has no deviations, errors or failures
2	Warning	All subsystems are working as intended, but the system detects unexpected behaviour (e.g. deviation from expected values).
3	FailureNonCritical	At least one error in one of the (sub)systems, but on this system level all functions are available
4	FailureCritical	At least one function is not available; operational consequences possible

**statusTechnicalManufacturerSpecificMessage** : uint16 [ 0 .. 1 ]

Must be used by the supplier to describe the reasons for a StatusTechnical != OK, that cannot be explained by existing datapoints (NOT including IM and manufacturer specific diagnostic messages). This Information MUST be provided from the supplier. This should provide flexibility for future uses. Multiple states can be indicated at the same time if multiple diagnosis have not been included in the model during the design phase. The supplier specific reason may not overlap with reasons already covered in other attributes.

Tree View  
Diagram

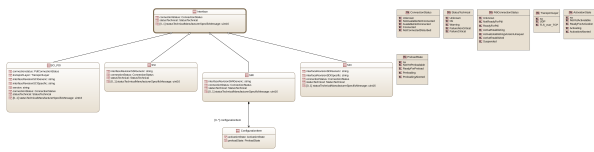


Figure 3 Tree View of Interface

3.2.1.5 Motor Turn Data Model

SPMS-7337 - MotorTurnData

To be implemented by one of the alternative underlying classes, depending on the implementation of the point machine.

Properties

The object owns the properties listed below:

**index** : string  
Index of the motor. Must be identical to the index in the logical class representing the motor (e.g. PointMachine).  
This property has the following additional attributes:

Property	Value
type	Prognosis : ContentType
category	Network : ContentCategory
update	OnThreshold : ClientUpdateSubscriptions
temporalAccuracy	2 s
samplingFrequency	50 Hz
dataCollectionInterval	10 s
dataBufferingTime	20 s

Tree View  
Diagram

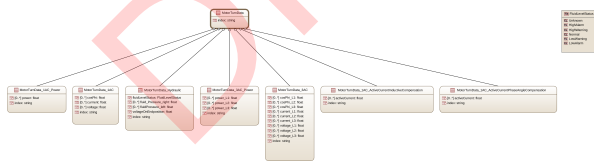


Figure 4 Tree View of MotorTurnData

3.2.1.6 Log Model

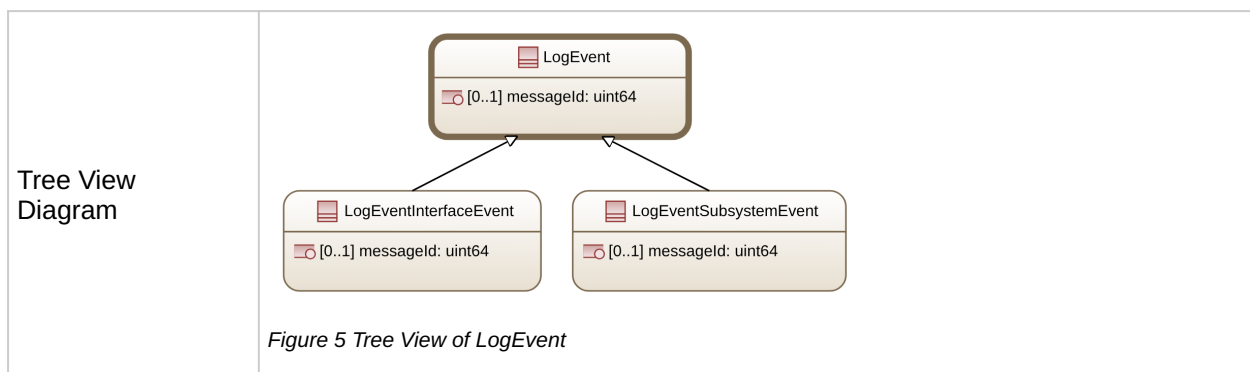
SPMS-7346 - LogEvent

Logging the events

Properties

The object owns the properties listed below:

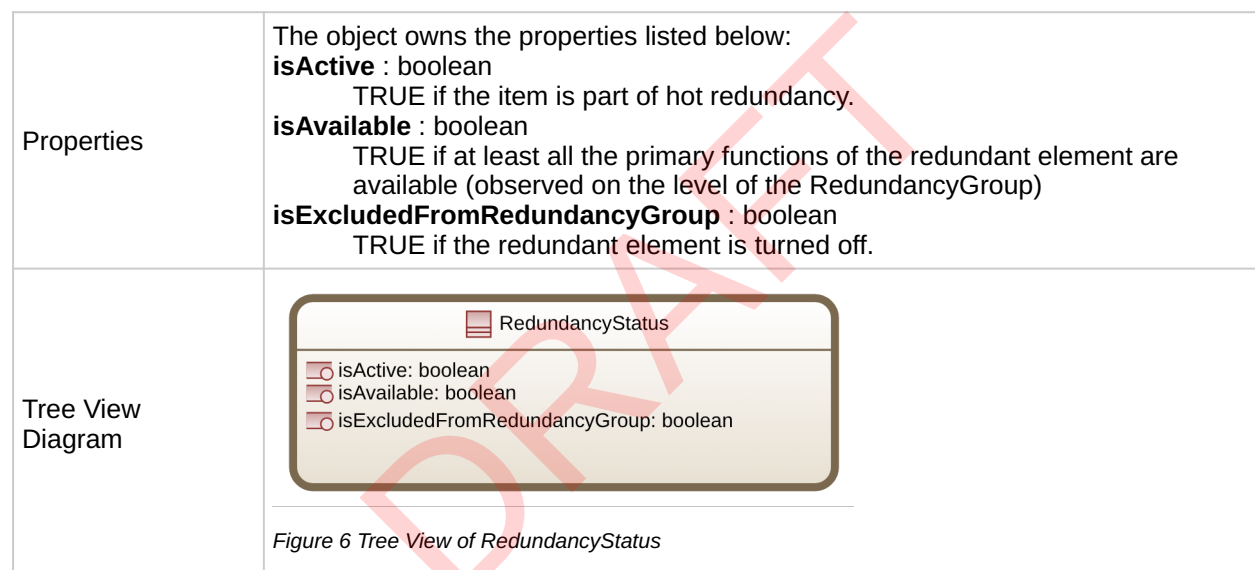
**messageId** : uint64 [ 0 .. 1 ]  
Unique ID linked to a localized Text, available in the service function Diagnostics collector.



### 3.2.1.7 Redundancy Model

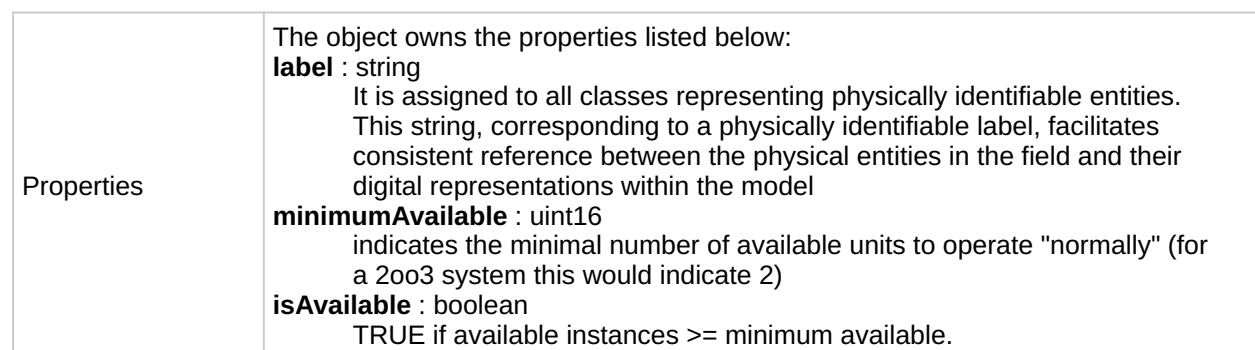
#### SPMS-7379 - RedundancyStatus

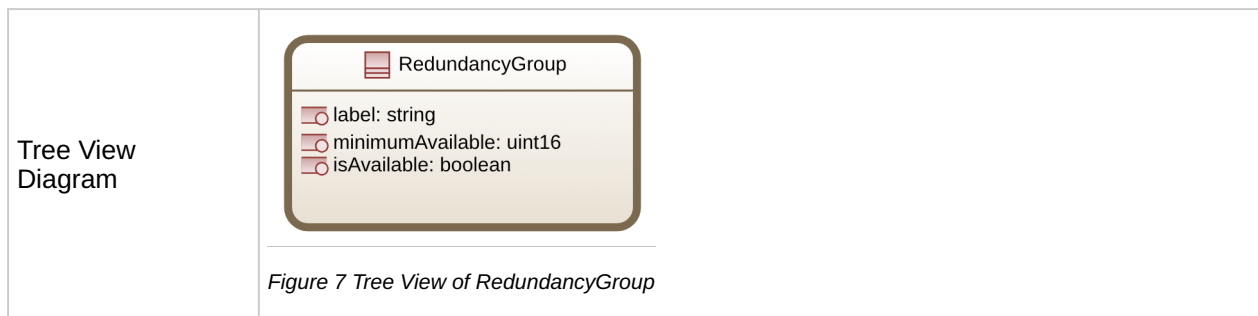
This object is added as a child object to the item (Equipment, Controller or other class) that is part of a redundancy. Use `isPartOfRedundancyGroup` (non hierarchical) as a to connect to the RedundancyGroup.



#### SPMS-7378 - RedundancyGroup

Watches the combined status of the instances that perform redundancy.





### 3.2.1.8 Meta Data Table



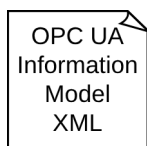
Meta Data Table is present here. [SPT2TS-131341 ]

### 3.2.1.9 Data Model

**POLARION**  
Data Model

Data Model is present here. [SPT2TS-131343 ]

### 3.2.1.10 NodeSet2 OPC UA Information Model



NodeSet2 OPC UA Information Model is present here. [SPT2TS-131347 ]

## 3.2.2 Alternative View of the SDI-GEN Information Model

**GENERATED CONTENT FOR INFORMATION MODEL ARE PRESENT HERE AS ARTEFACTS**

**The data present here is described in three artefacts:**

- **A UML class diagram representing the static structure of the types used including the hierarchical and non-hierarchical (functional relationships (references)).**
- **A table according to the following meta data attributes**
- **A Nodeset2 XML OPC UA information model**

[SPT2TS-130225 ]

## 4 Appendix

### 4.1 Input documents

## 4.2 Standards and References

DRAFT