IDTA 02031-2
Process Parameters
Part 2: Instance

Version 1.0
July 2025

SPECIFICATION

Submodel Template of the
Asset Administration Shell

Submodel Template
IDTA approved
- 100% AAS compliant
- Consistent & interoperable
- Released by the AAS experts

Imprint

Publisher
Industrial Digital Twin Association
Lyoner Strasse 18
60528 Frankfurt am Main
Germany
https://www.industrialdigitaltwin.org/

Version history

Date	Version	Comment
02.07.2025	1.0	Release of the official Submodel template published by IDTA.

Contents

Figures

Tables

1 General

1.1 About this document

This document is a part of a specification series. Each part specifies the contents of a Submodel template for the Asset Administration Shell (AAS). The AAS is described in [1], [2], [3] and [6]. First exemplary Submodel contents were described in [4], while the actual format of this document was derived by the "Administration Shell in Practice" [5]. The format aims to be very concise, giving only minimal necessary information for applying a Submodel template, while leaving deeper descriptions and specification of concepts, structures and mapping to the respective documents [1] to [6].

The target group of the specification are developers and editors of technical documentation and manufacturer information, which are describing assets in smart manufacturing by means of the Asset Administration Shell (AAS) and therefore need to create a Submodel instance with a hierarchy of Submodel Elements. This document especially details on the question, which Submodel Elements with which semantic identification shall be used for this purpose.

This document is part of a specification series for IDTA 02031 (Process Parameters), describing a framework for process input parameters (type / process parameters) and output parameters (instance / executed processes). It consists of following parts:

- Part 1: Type (Process Parameters)
- Part 2: Instance (Executed Processes)

1.2 Scope of the Submodel

This Submodel defines instance-specific information of Process Parameters and provides a framework to describe the output parameters for processes of industrial equipment. This Submodel does not require any knowledge about how a process is executed and in which order. It is solely focused on the structured description of data reported by a process.

The need for a dedicated Process Parameters Submodel is to enable the interoperable description and transfer of data between assets. The concept is based on process knowledge from various organizations in the automotive wire harness industry. Although the associated use cases are from the value chain of wire harness manufacturing, the aim of the working group is to develop a generalized Submodel that can be used for a wide variety of products and industries.

This Submodel is intended for the description of output parameters of a process and should be used in combination with the Hierarchical Structures Submodel to define the sequence of process execution and hierarchical requirements.

1.3 Use cases, requirements and design decisions

> Although the associated use cases are from the value chain of wire harness manufacturing, the aim of the working group is to develop a generalized Submodel that can be used for a wide variety of products and industries.

The intention behind the implementation of the Asset Administration Shell is to close the digital gaps in the value chain as much as possible. The data is to be made available to the value chain partners for the individual value chain areas (development, production, and assembly) in a structured and semantically described manner.

The added value in the wire harness industry can be divided into three major parts: development, production, and assembly. A special characteristic of the wire harness industry is in the description of a 150% harness in

the development phase. The 150% harness encompasses all possible variants of a wire harness and only exists in the digital world. It can be understood as the wire harness "type" in the Asset Administration Shell ecosystem. When a customer orders a vehicle with a specific configuration, the 100% harness is created from the production point of view, which ultimately represents the "instance" of the wire harness. In this case the focus stands on the product type.

Another special characteristic are the different views on an asset. An asset is defined as anything that has a value for a company. In the wire harness industry, there is no single player that provides the entire know-how about the product wire harness. Instead, a wide variety of knowledge streams flow together across different stages of the value chain, and the entire know-how ultimately results in a wire harness. The challenge is that the perspective on the asset changes depending on the value creation stage.

- For the machine supplier (Tier-3) the production resource is the asset.
- For the component supplier (Tier-2) the connector or cable is the asset.
- For the wire harness manufacturer (Tier-1) the produced wire harness is the asset.
- Finally, for the original equipment manufacturer (OEM) the entire vehicle is the asset.

These different perspectives must be considered in the implementation of the Asset Administration Shell and the development of Submodels.

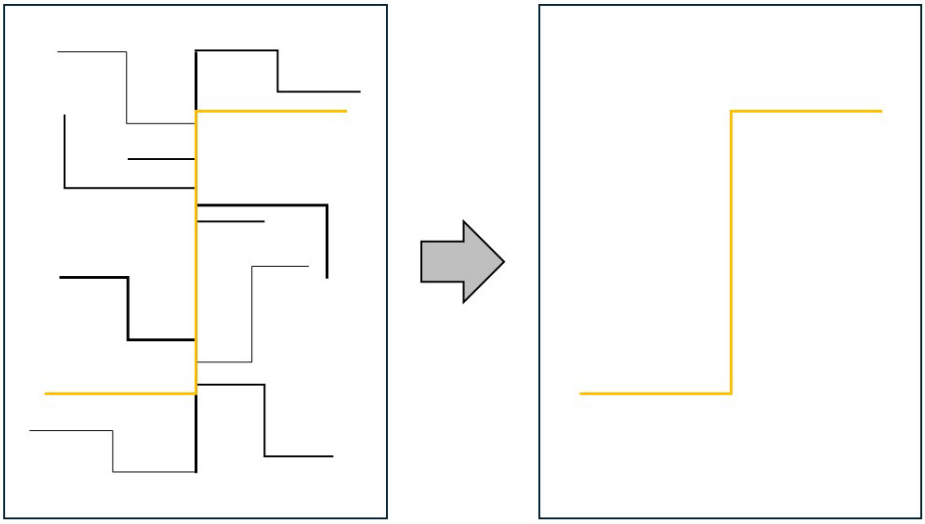

Figure 1 Derivation of a production module

Every specific wire harness is produced from thousands of individual parts. Producing such a product in one machine is currently impossible and uneconomical. For this reason, the wire harness is divided in smaller production modules, see Figure 1. A production module thus corresponds to a partial scope of the wire harness and can be produced on a production resource.

There are also many different configurations in every production module, which means that the production parameters can be different and must be verified or configured for every specific module.

The Process Parameters Submodel enables self-orchestration of production systems through standardized information structures, semantic descriptions and definition of necessary parameters for each production process.

The development of the specification for the Process Parameters Submodel is based on the use case for the value chain of the wire harness. The reference to a specific product is intended that the development of the Submodel can be carried in a targeted and realistic manner. Despite the reference to the wire harness, the aim is to develop a generic data provision for the value-adding processes that can be used across all industries and is not intended to be an isolated solution. The production of wire harnesses is composed of multiple production steps, some of them are characterized by manual activities (see Figure 2).

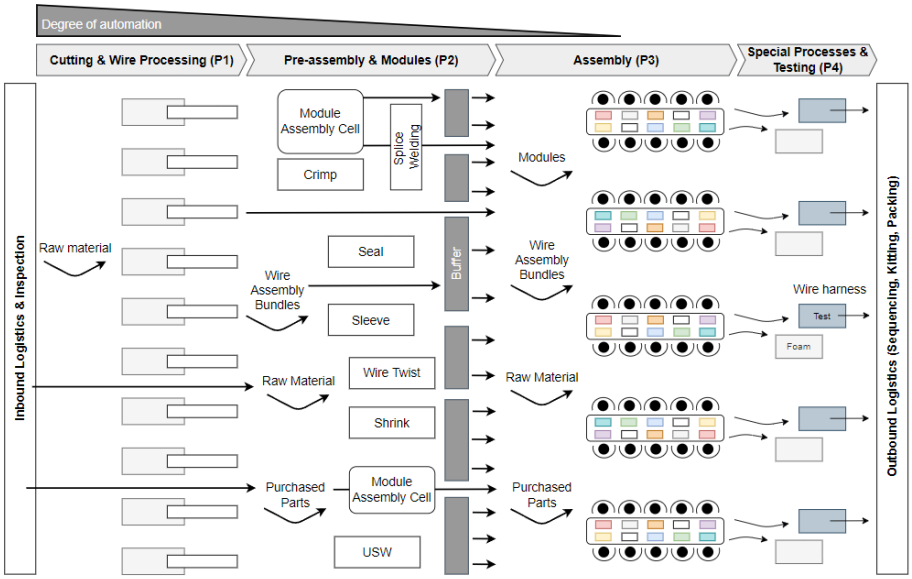

Figure 2 Overview of production steps and sections in the manufacturing of wire harnesses

Before the actual wire harness production starts, all necessary (raw) materials arrive in the inbound logistics or warehouse area, where components and raw materials are scanned.

The **cutting process and wire processing (P1)** presents a very high degree of automation and is carried out after the entry of the raw material. The cutting process is currently performed by wire processing machines, which exchange information via OPC UA and are controlled by an MES system. Generally, in this step the machines take care of the processes of cutting to length, stripping, application of seals and single strand contacts and, if necessary, marking by labeling, color coding or machine-readable markings.

The **pre-assembly process (P2)** is partially automated. In this step, different processes take place such as manual butting, ultrasonic welding (USW), seal processes, and many others as depicted in Figure 2.

The final **assembly process (P3)** is predominantly performed by manual labor, where the parts and modules from P2 serve as input for the wire harness assembly in which these inputs are assembled, routed, covered, and the layout completed.

In the stage **"Special Processes and Testing" (P4)** it is possible that special processes such as foaming are carried out. At the end of this stage the test process is performed to check the quality of the harness, where the quality of the entire harness and all its components is checked, including presence of components, electrical connections, resistors, capacitors, diodes, wire colors, pin positions, fiber optical wires, airbags, thermistors, and more.

Finally, the production process is concluded, allowing for the commissioning and packaging processes before the product leaves the facility.

Each of these areas and the production steps require and generate information about the product (components, sub-assemblies, wire harnesses, etc.), the production processes (cutting, crimping, block loading, testing, etc.) and the production resources (machines, tools, operators, etc.), see in Figure 3. For a generalized process description, the information elements must be made accessible in a standardized description. The procedure shown in Figure 2 is intended to illustrate the aggregation of the individual attributes for the execution of the process from various sources. The focus is on the dependency of the various areas.

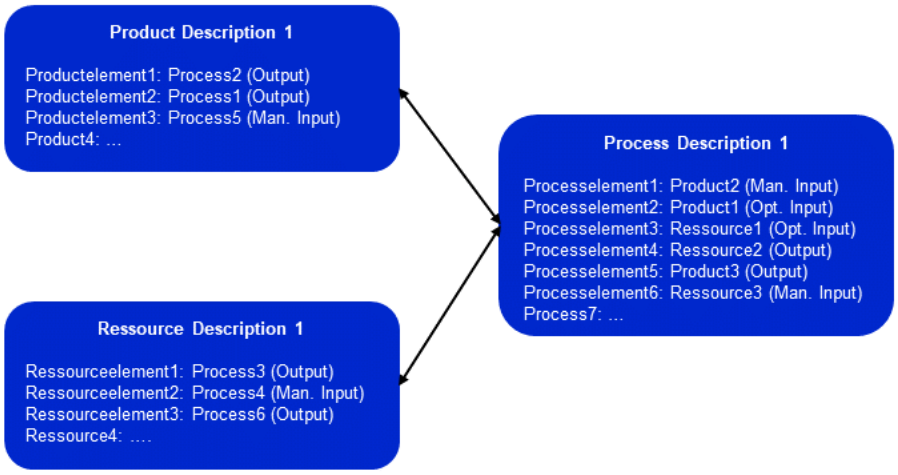

Figure 3 PPR-Model and dependencies

Table 1 Use case descriptions

Use Case	Description
Specification of Products (wire harness)	Most of the wire harness consist of many individual and intermediate parts, which can be assembled in different ways to form a complete harness for a specific vehicle. For this purpose, the entire product and its components must be described in a standardized way that allows harness manufacturers to produce parts in stages, based on specific configurations ordered by the OEM.
Specification of production resources	Different production resources from different suppliers, with different capabilities are involved in the production of product parts. Depending on the specific needs of a manufacturer, a specific subset of production resources is used. For this purpose, the resources must be described in a way that lets manufacturers select the correct production resource for their manufacturing processes.
Specification of production processes and parameters	The manufacturing of a product covers a wide range of production processes, each of them requires their own set of input and output parameters. These processes and process parameters need to be described in a generalized way that allows all manufacturers to describe the manufacturing processes of a part and exchange them with the production resources and manufacturing systems.
Traceability of the wire harness manufacturing process	A key requirement in manufacturing, is traceability, which mandates that the entire manufacturing process of a specific part can be proven. This includes raw materials, production resources, process parameters, and more. For this purpose, the manufacturer must be able to collect traceability information as the parts – or intermediate parts – undergoes individual steps in the production process.

The Process Parameters Submodel should make it possible to aggregate the necessary information (type) from various sources (product and resource description) and provide the information for process execution.

The execution of processes generates new instance information, which must also be documented and assigned to one of the 3 sources (product, process and resource), see Figure 4.

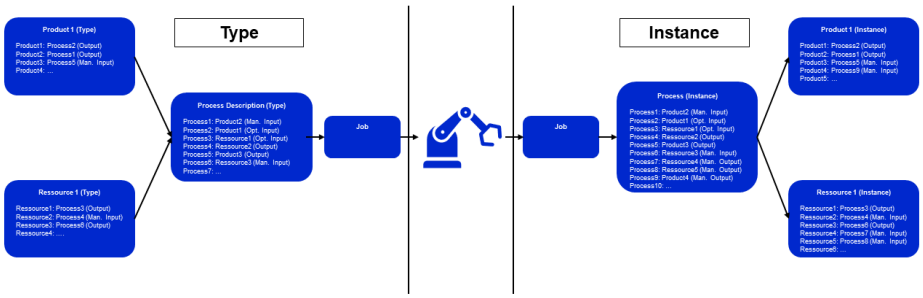

Figure 4 Data aggregation and structuring the data after process execution

2 Submodel Template “Executed Processes”

The Submodel Template “Executed Processes” describes the outcome of a resource that executed the processes described by the Submodel Template “Process Parameters”. It allows the description of one or more Runs, each of which represents an execution of the Process Parameters. For example, when executing a job order with multiple items.

Each Run contains information for each process defined in Process Parameters Submodel. The input parameters are duplicated for traceability reasons and to allow resources to change values when needed. Additional output values are documented.

2.1 Approach

The Submodel is intended to aggregate all the instance information after the execution of production processes from product, process and resources. For this purpose, an independent Submodel was created that describes the processes by “Run” SubmodelElementCollections on the second level, Figure 5.

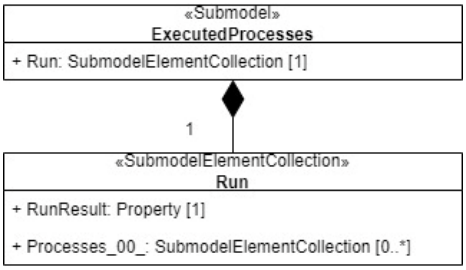

Figure 5 Structure of the ExecutedProcesses Submodel and Run SubmodelElementCollection

2.2 Elements of Submodel “ExecutedProcesses”

Table 2 Elements of Submodel ExecutedProcesses

idShort:	ExecutedProcesses Note: the above idShort shall always be as stated.		
Class:	Submodel		
semanticId:	[IRI] https://admin-shell.io/idta/ExecutedProcesses/1/0		
Parent:	Asset Administration Shell		
Explanation:	The Submodel describes the input and output parameters for the execution of processes.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**Example**	
[SMC] Run__00__	[IRI] https://admin-shell.io/idta/ExecutedProcesses/Run/1/0 Represents a single execution of the processes defined in the process.	n/a	1..*

2.3 Elements of SMC “Run__00__

Each "Run" represents a process sequence that has been executed using the information from the ProcessParameters Submodel.

Table 3 Elements of SMC Run__00__

idShort:	Run__00__ Note: the above idShort shall always be as stated.		
Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/ExecutedProcesses/Run/1/0		
Parent:	Submodel ExecutedProcesses		
Explanation:	Represents a single execution of the processes defined in the process.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**Example**	
[Property] RunResult	[IRI] https://admin-shell.io/idta/ExecutedProcesses/RunResult/1/0 Indicates the outcome of a run.	[int]	1
[SMC] Process__00__	[IRI] https://admin-shell.io/idta/ExecutedProcesses/Process/1/0 Describes the values of a process that was executed.	n/a	0..*

2.4 Elements of SMC “Process”

The individual process attributes of the SMC are described in the SMC "*Process*". Product-unspecific modeling was chosen for this so that the widest possible range of products can be manufactured. It is important to mention at this point that the attributes contained in the SMC contain both the actual and nominal values from the individual production processes carried out. The additional SMCs "*ProcessStages*" and "*ProcessErrors*" contain several attributes that go into more detail about the sub-processes executed.

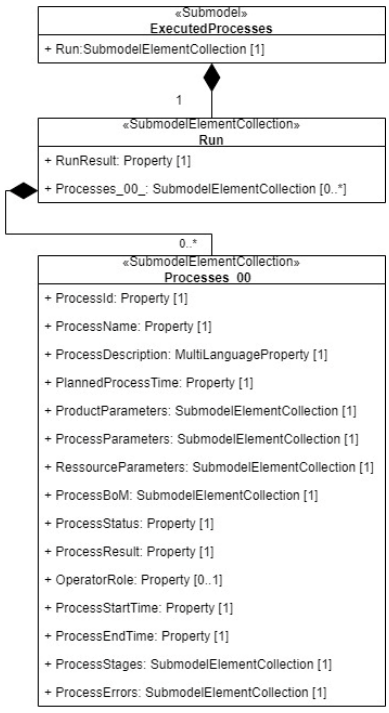

Figure 6 Structure of the SMC Process__00__

Table 4 Elements of SMC Process__00__

idShort:	Process__00__ Note: the above idShort shall always be as stated.		
Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/ExecutedProcesses/Process/1/0		
Parent:	SMC Run		
Explanation:	Describes the values of a process that was executed.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**Example**	
[Property] ProcessId	[IRI] https://admin-shell.io/idta/ProcessParameters/ProcessId/1/0 Describes the ID of the process	[string]	1
[Property] ProcessName	[IRI] https://admin-shell.io/idta/ProcessParameters/ProcessName/1/0 Describes the Name of the Process	[string]	1
[MLP] ProcessDescription	[IRI] https://admin-shell.io/idta/ProcessParameters/ProcessDescription/1/0 Describes the process.	[string]	1
[Property] PlannedProcessTime	[IRI] https://admin-shell.io/idta/ProcessParameters/PlannedProcessTime/1/0 Planned processing time for process execution (without set-up time)	[duration]	1
[SMC] ProductParameters	[IRI] https://admin-shell.io/idta/ProcessParameters/ProductParameters/1/0 Mandatory product parameters for process execution	n/a	1
[SMC] ProcessParameters	[IRI] https://admin-shell.io/idta/ProcessParameters/ProcessParameters/1/0 Mandatory process parameters for process execution	n/a	1
[SMC] ResourceParameters	[IRI] https://admin-shell.io/idta/ProcessParameters/ResourceParameters/1/0 Mandatory resource parameters for process execution	n/a	1
[SMC] ProcessBoM	[IRI] https://admin-shell.io/idta/ProcessParameters/ProcessBoM/1/0 describes the products or semi-finished products to be involved in the process	n/a	1
[Property] ProcessStatus	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessStatus/1/0 Describes the process status.	[int]	1
[Property] ProcessResult	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessResult/1/0 Result of the process.	[int]	1
[Property] OperatorRole	[IRI] https://admin-shell.io/idta/ExecutedProcesses/OperatorRole/1/0 Role of the operator in the process.	[string]	0..1

[Property] ProcessStartTime	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessStartTime/1/0 Start time of the process.	[time]	1
[Property] ProcessEndTime	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessEndTime/1/0 End time of the process.	[time]	1
[SMC] ProcessStages	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessStages/1/0 List of stages that the process went through.	n/a	1
[SMC] ProcessErrors	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessErrors/1/0 List of errors that caused the process to fail.	n/a	1

2.5 Elements of SMC “ProcessStages”

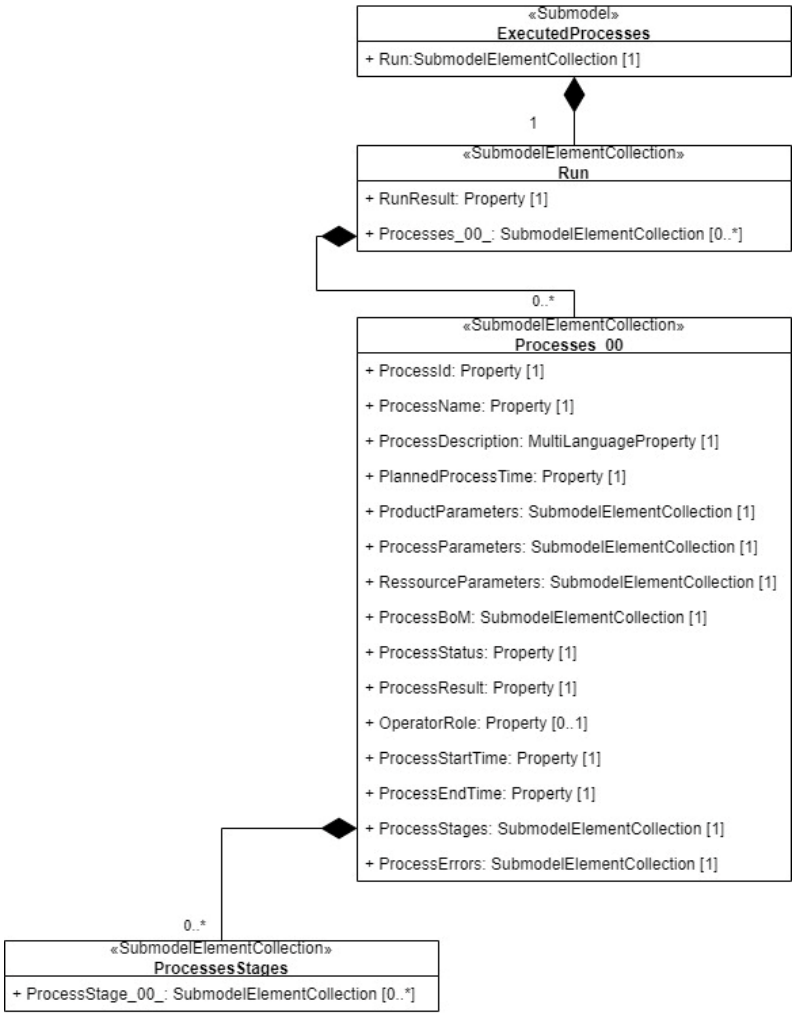

Figure 7 Structure of the SMC ProcessStages

Table 5 Elements of SMC ProcessStages

idShort:	ProcessStages Note: the above idShort shall always be as stated.

Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessStages/1/0		
Parent:	SMC Process		
Explanation:	List of stages that the process went through.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**Example**	
[SMC] ProcessStage__00__	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessStage/1/0 A stage that the process went through.	n/a	0..*

2.6 Elements of SMC “ProcessStage”

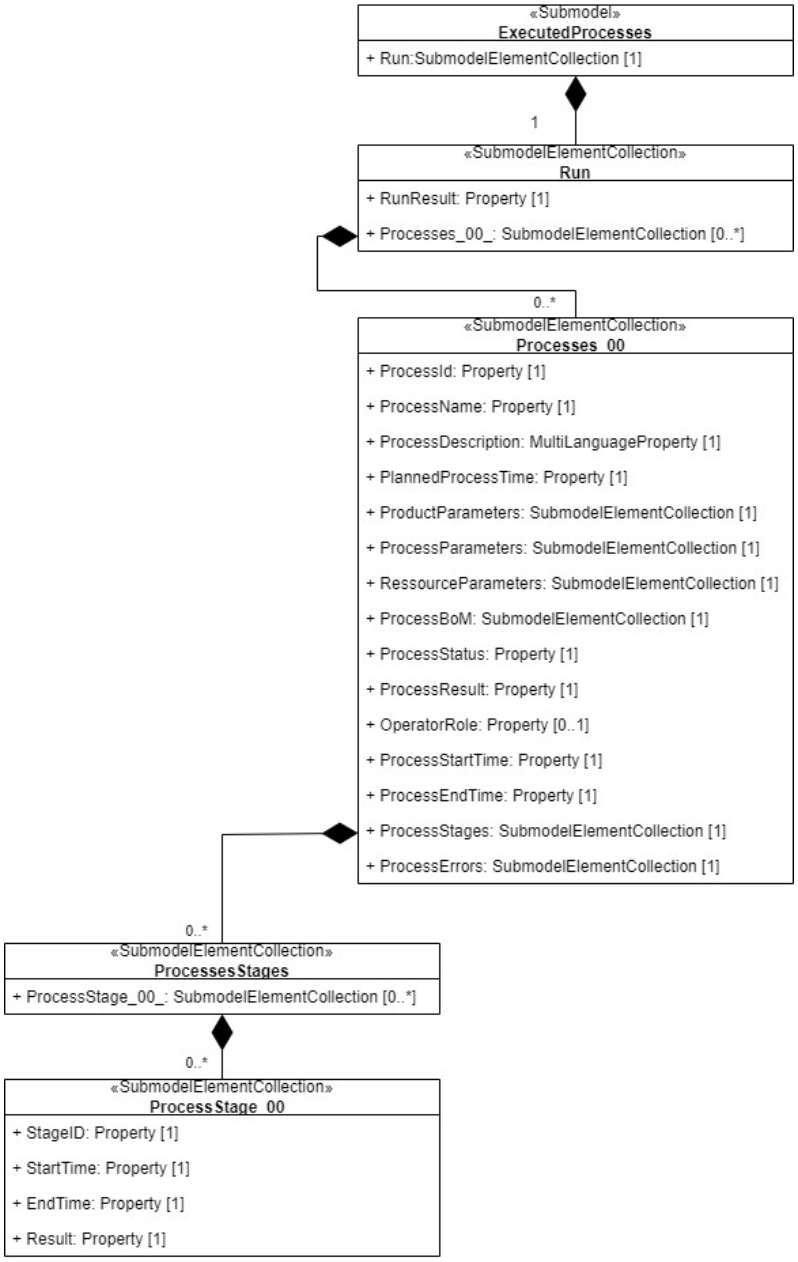

Figure 8 Structure of the SMC ProcessStage

Table 6 Elements of SMC ProcessStage

idShort:	ProcessStage Note: the above idShort shall always be as stated.		
Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessStage/1/0		
Parent:	SMC ProcessStages		
Explanation:	A stage that the process went through.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**Example**	
[Property] StageId	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessStage/StageId/1/0 ID of the stage.	[string]	1
[Property] StartTime	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessStage/StartTime/1/0 Time when the stage started.	[time]	1
[Property] EndTime	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessStage/EndTime/1/0 Time when the stage ended.	[time]	1
[Property] Result	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessStage/Result/1/0 Stage result.	[int]	1

2.7 Elements of SMC “ProcessErrors”

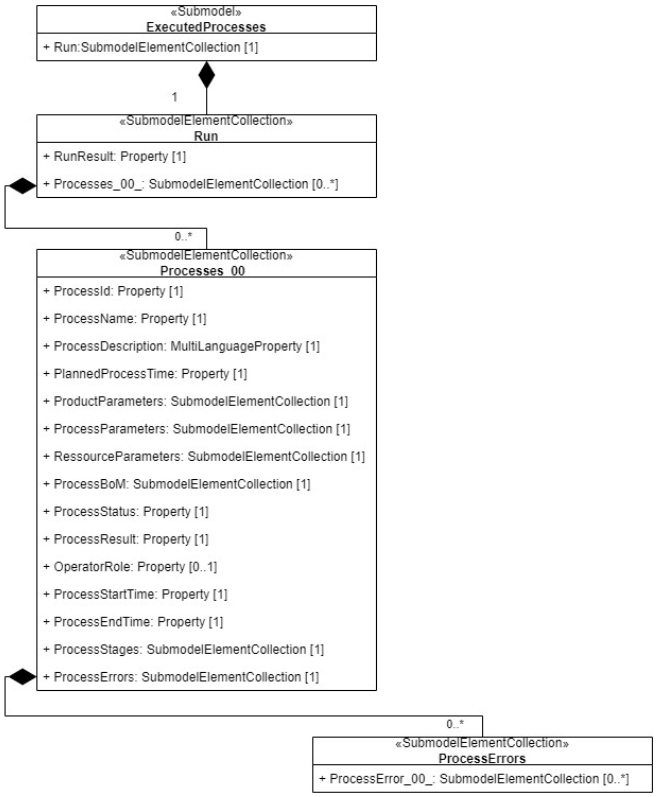

Figure 9 Structure of SMC ProcessErrors

Table 7 Elements of SMC ProcessErrors

idShort:	ProcessErrors Note: the above idShort shall always be as stated.		
Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessErrors/1/0		
Parent:	SMC Process		
Explanation:	List of errors that caused the process to fail.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**Example**	
[SMC] ProcessError__00__	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessError/1/0 An error that caused the process to fail.	n/a	0..*

2.8 Elements of SMC “ProcessError__00__”

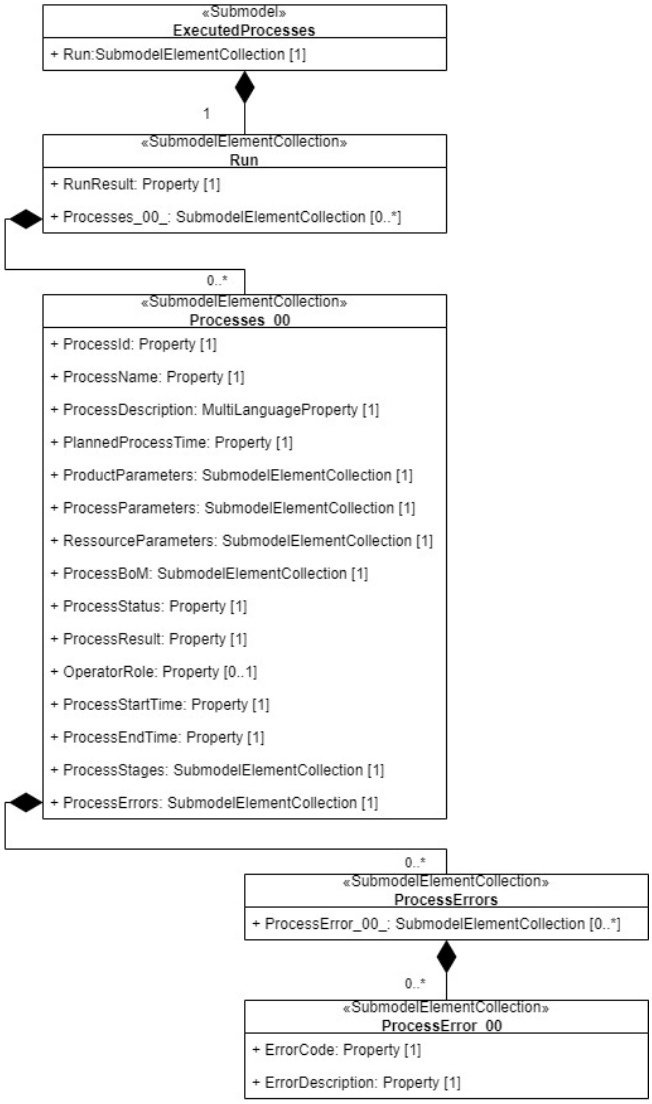

Figure 10 Structure SMC ProcessError

Table 8 Elements of SMC ProcessError__00__

idShort:	ProcessError__00__ Note: the above idShort shall always be as stated.		
Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessStage/1/0		
Parent:	SMC ProcessErrors		
Explanation:	An error that caused the process to fail.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**Example**	
[Property] ErrorCode	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessError/ErrorCode/1/0 Error code.	[string]	1
[Property] ErrorDescription	[IRI] https://admin-shell.io/idta/ExecutedProcesses/ProcessError/ErrorDescription/1/0 Error description.	[string]	1

Annex A. Explanations on used table formats

1. General

The used tables in this document try to outline information as concise as possible. They do not convey all information on Submodels and SubmodelElements. For this purpose, the definitive definitions are given by a separate file in form of an AASX file of the Submodel template and its elements.

2. Tables on Submodels and SubmodelElements

For clarity and brevity, a set of rules is used for the tables for describing Submodels and SubmodelElements.

- The tables follow in principle the same conventions as in [5].
- The table heads abbreviate 'cardinality' with 'card'.
- The tables often place two informations in different rows of the same table cell. In this case, the first information is marked out by sharp brackets [] form the second information. A special case are the semanticIds, which are marked out by the format: (type)(local)[idType]value.
- The types of SubmodelElements are abbreviated:

SME type	SubmodelElement type
Property	Property
MLP	MultiLanguageProperty
Range	Range
File	File
Blob	Blob
Ref	ReferenceElement
Rel	RelationshipElement
SMC	SubmodelElementCollection

- If an idShort ends with '__00__', this indicates a suffix of the respective length (here: 2) of decimal digits, in order to make the idShort unique. A different idShort might be choosen, as long as it is unique in the parent's context.
- The Keys of semanticId in the main section feature only idType and value, such as: [IRI]https://admin-shell.io/vdi/2770/1/0/DocumentId/Id. The attributes "type" and "local" (typically "ConceptDescription" and "(local)" or "GlobalReference" and (no-local)") need to be set accordingly; see [6].
- If a table does not contain a column with "parent" heading, all represented attributes share the same parent. This parent is denoted in the head of the table.
- Multi-language strings are represented by the text value, followed by '@'-character and the ISO 639 language code: example@EN.
- The [valueType] is only given for Properties.

Annex B. Bibliography

[1] "Recommendations for implementing the strategic initiative INDUSTRIE 4.0", acatech, April 2013. [Online]. Available https://www.acatech.de/Publikation/recommendations-for-implementing-the-strategic-initiative-industrie-4-0-final-report-of-the-industrie-4-0-working-group/

[2] "Implementation Strategy Industrie 4.0: Report on the results of the Industrie 4.0 Platform"; BITKOM e.V. / VDMA e.V., /ZVEI e.V., April 2015. [Online]. Available: https://www.bitkom.org/noindex/Publikationen/2016/Sonstiges/Implementation-Strategy-Industrie-40/2016-01-Implementation-Strategy-Industrie40.pdf

[3] "The Structure of the Administration Shell: TRILATERAL PERSPECTIVES from France, Italy and Germany", March 2018, [Online]. Available: https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/hm-2018-trilaterale-coop.html

[4] "Beispiele zur Verwaltungsschale der Industrie 4.0-Komponente – Basisteil (German)"; ZVEI e.V., Whitepaper, November 2016. [Online]. Available: https://www.zvei.org/presse-medien/publikationen/beispiele-zur-verwaltungsschale-der-industrie-40-komponente-basisteil/

[5] "Verwaltungsschale in der Praxis. Wie definiere ich Teilmodelle, beispielhafte Teilmodelle und Interaktion zwischen Verwaltungsschalen (in German)", Version 1.0, April 2019, Plattform Industrie 4.0 in Kooperation mit VDE GMA Fachausschuss 7.20, Federal Ministry for Economic Affairs and Energy (BMWi), Available: https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/2019-verwaltungsschale-in-der-praxis.html

[6] "Details of the Asset Administration Shell; Part 1 - The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC01)", November 2020, [Online]. Available: Asset Administration Shell Specification - Part 1: Metamodel (industrialdigitaltwin.org)

www.industrialdigitaltwin.org