IDTA 02053-1-0 Control configuration and parametrization for NC/CNC machines

November 2024

SPECIFICATION

Submodel Template of the Asset Administration Shell

Imprint

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

Version history

Date	Version/Revision	Comment
28.11.2024	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 AAS and therefore need to create a Submodel instance with a hierarchy of SubmodelElements. This document especially details on the question, which SubmodelElements with which semantic identification shall be used for this purpose.

1.2 Scope of the Submodel

This Submodel template aims at interoperable provision of information describing the “Control configuration and parametrization for NC/CNC machines” in regard to the asset of the respective AAS. The purpose of this document is to make selected specifications of Submodels in such manner that information about assets can be exchanged in a meaningful way between partners in a value creation network. It targets people who are commissioning or integrate Computerized Numerical Control (CNC) machines. Therefore, they need data to parametrize the CNC control.

The intended use-case is the provision of a standardized property structure for Control configuration and parametrization for NC/CNC machines, which enables an easy access of data, which is needed to configure or parametrize a CNC controller.

This concept can serve as a basis for standardizing the respective Submodel. The conception is based on existing norms, studies of common practices at enterprises, directives and standards so that a far-reaching acceptance can be achieved.

Beside standardized Submodel this template also introduces standardized SubmodelElementCollections (SMC) in order to improve the interoperability while modelling aspects of Control configuration and parametrization for NC/CNC machines within other Submodels.

1.3 Relevant standards for the Submodel template

According to [3], interoperable properties might be defined by standards, consortium specifications or manufacturer specifications. Useful standards providing sources of concepts are:

AAS Submodel Digital Nameplate	The digital nameplate AAS is used as a reference to the real asset (control, axis, drive).
AAS Submodel Generic Frame for Technical Data for Industrial Equipment in Manufacturing	The technical specification AAS is used as a reference to the technical specification of the real asset (e.g., drive parameters).

AAS Submodel Software Nameplate	The software nameplate AAS is used as a reference to the software asset (control software if one exists)

Table 1: List of examplary standards defining interoperable properties

So called property dictionaries are used identify information elements (see Terms and Definitions of [6]). Such property dictionaries include:

- ECLASS, see: https://www.eclasscontent.com/
- IEC CDD, see: https://cdd.iec.ch/cdd/iec61987/iec61987.nsf and https://cdd.iec.ch/cdd/iec62683/cdddev.nsf

In this document, properties are aimed to be described by ECLASS.

1.4 Use cases, requirements and design decisions

1.4.1 Use and economic relevance

This Submodel is designed to simplify configuring and parameterizing controllers, especially CNC controllers. Commissioning a CNC control system becomes increasingly intricate due to the various CNC and component manufacturers. Therefore, the parameterization data must be located first.

Two different use cases have been identified (see Figure 1):

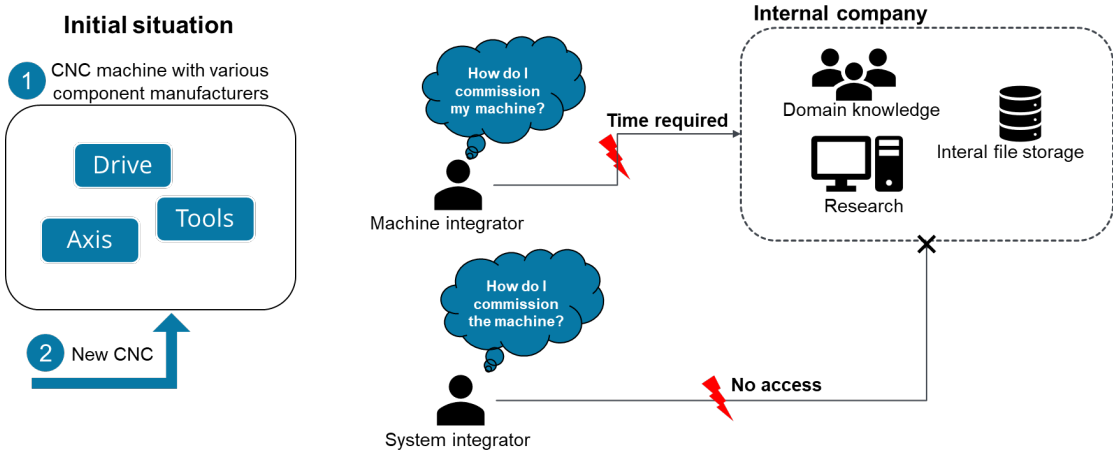

Figure 1: Use Cases

1.4.2 Use Case A:

In use case A, a machine integrator has to commission a new CNC machine that has not yet been parameterized in the company. This entails accessing domain knowledge or internal repositories, which is time-consuming. Moreover, the integrator is required to carry out their own research during this process. In order to avoid this, the information that already exists must be presented in a structured manner.

1.4.3 Use Case B:

In use case B, a system integrator is tasked with commissioning a CNC control system without access to the company's internal files. This presents an additional challenge as it significantly increases the time required to parameterize the control system.

One possible solution lies in utilizing AAS as a structured data storage mechanism to furnish relevant information for CNC parameterization and configuration. In order to achieve this, manufacturers of components and CNC equipment should produce the aforementioned administration shell. Alternatively, the machine integrator could generate the administration shell to simplify the reconfiguration of CNC machines.

1.4.4 Use Case C:

A machine integrator has many variants of a machine type. Each machine configuration will be derived based on the individual order of the customer. Then, a specific setup procedure – containing the wanted axes and components with its configuration data – needs to be defined automatically.

1.5 Semantic IDs

For one specific property relating to OPC UA concepts, the semantic Id https://admin-shell.io/idta/opcua-server-datasheet/1/0 proposed by IDTA work group “OPC UA Server Datasheet”, is used.

Another Submodel which is considered in further work is the tool description Submodel. Therefore the semantic id https://admin-shell.io/idta/machine-tool-description/1/0 is used.

2 Control configuration and parametrization for NC/CNC machines

2.1 Approach

To create an AAS for this application, the initial step involves analyzing the configuration possibilities. The subcomponents, as illustrated in Figure 2, have been identified as follows:

- Channel configuration
- Axis configuration
- Drive configuration
- Tool configuration (will be a separated Submodel in further work)

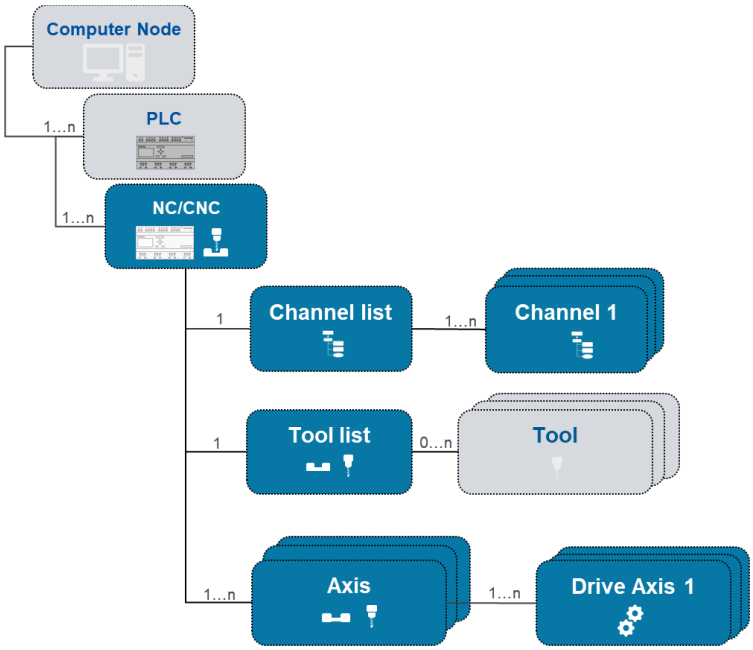

Figure 2 CNC Control sub-components

Additionally, the following configuration options are also highlighted:

- Communication configuration
- NC program configuration

Possible functions and interactions

- Consistent use of data that already exists
- Preparation of data that already exists
- Simple provision of data for configuring a CNC control system

2.2 Properties of the Submodel “Control configuration and parametrization for NC/CNC machines”

The figure below shows the UML-diagram defining the relevant properties which need to be set. Table 2 describes the details of the Submodel structure combined with examples.

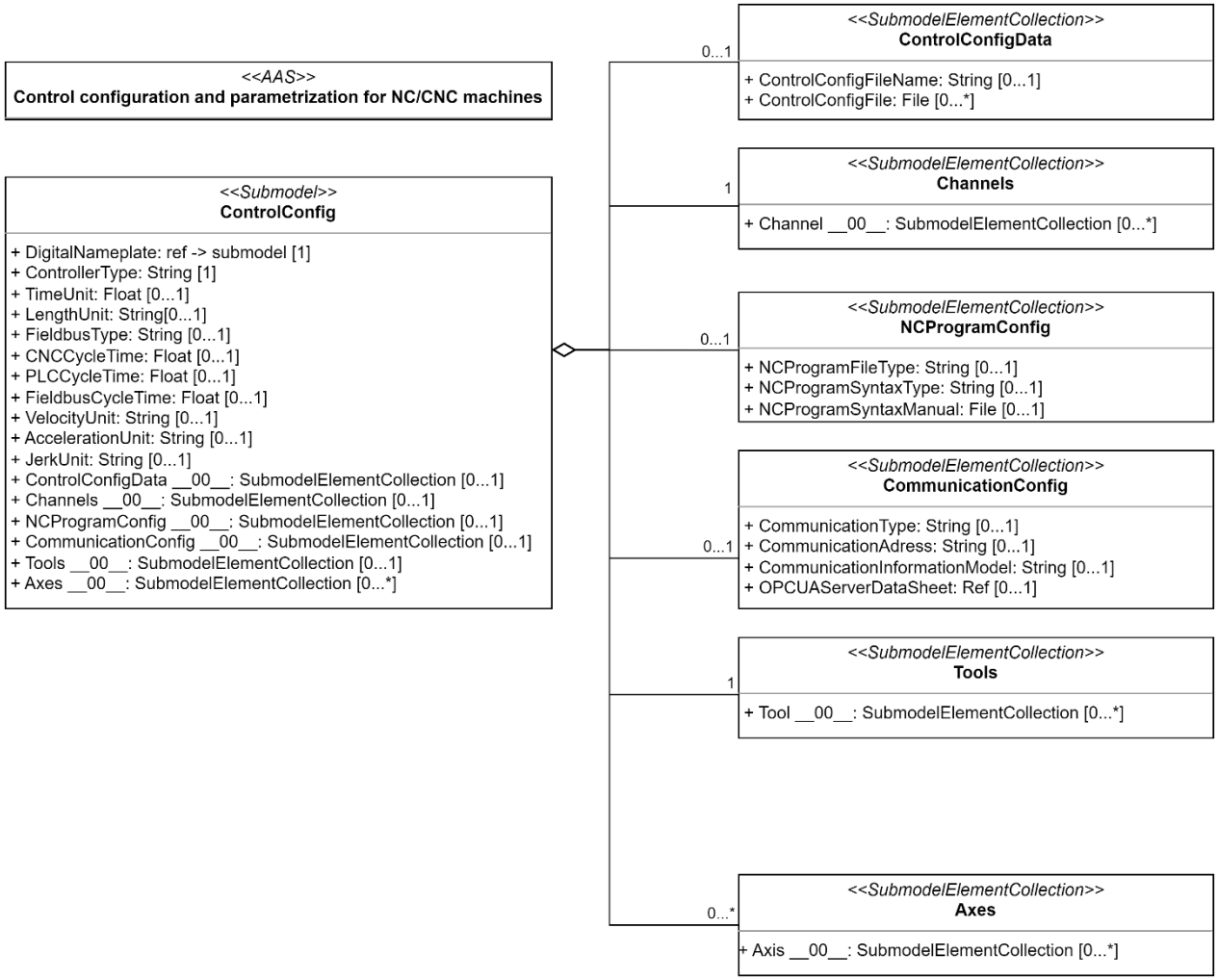

Figure 3: UML-Diagram for Submodel “Control configuration and parametrization for NC/CNC machines”

The Submodel itself is devided into different parts, which can be used to parametrize the CNC control:

- Control configuration
- Channels
- Axis configuration
- Drive configuration
- NC programming configuration
- Communication configuration
- Tool configuration (will be referenced to a Submodel, which is in planned in future work)

First, the main Submodel collection “ControllerConfig” will be introduced. In this Submodel the main configuration information of the control is defined like what type is my controller or in which cycle time the controller is operating.

Figure 4 shows the more detailed UML diagram and the properties of the Submodel ControlConfig.

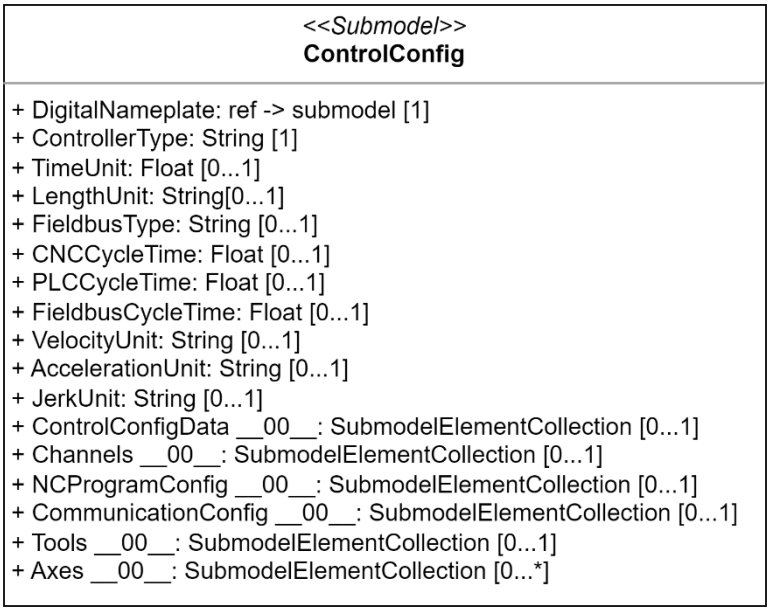

Figure 4: UML-diagram for the Submodel "ControllerConfig"

Table 2: Properties of Submodel “ControlConfig”

idShort:	ControlConfig __00__ Note: a different idShort might be used, as long as it is unqiue in the Submodel.		
Class:	Submodel (SM)		
semanticId:	[IRI] https://admin-shell.io/idta/ControlConfig/1/0		
Parent:	AAS		
Explanation:	The Submodel “ControlConfig” is the collection for various parametrizations for the control of CNC machines		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Reference] DigitalNameplate	[IRI] https://admin-shell.io/idta/nameplate/3/0/Nameplate Reference to the digital nameplate of the controller.	n/a	[1]
[Property] ControllerType	[IRI] https://admin-shell.io/idta/ControlConfig/ControllerType/1/0 The Controller Type describes the type of controller	[String]	[1]
[Property] TimeUnit	[IRI] https://admin-shell.io/idta/ControlConfig/TimeUnit/1/0 Time unit of the control	[String] s, ms, us…	[0…1]
[Property] LengthUnit	[IRI] https://admin-shell.io/idta/ControlConfig/LengthUnit/1/0 Length Unit of the control	[string] m, mm, cm, inch …	[0…1]
[Property]	[IRI] https://admin-shell.io/idta/ControlConfig/FieldbusType/1/0	[String]	[0…1]

FieldbusType	Fieldbus Type of the control	EtherCAT, Profinet, Profibus, EtherNet/IP, SERCOS Interface …	
[Property] CNCCycleTime	[IRI] https://admin-shell.io/idta/ControlConfig/CNCCycleTime/1/0 Defines the cycle time of the CNC control	[Float]	[0…1]
[Property] PLCCycleTime	[IRI] https://admin-shell.io/idta/ControlConfig/PLCCycleTime/1/0 Defines the cycle time of the PLC control	[Float]	[0…1]
[Property] FieldbusCycleTime	[IRI] https://admin-shell.io/idta/ControlConfig/FieldbusCycleTime/1/0 Defines the cycle time of the fieldbus control	[Float]	[0…1]
[Property] VelocityUnit	[IRI] https://admin-shell.io/idta/ControlConfig/VelocityType/1/0 Defines the velocity unit of the CNC control	[String] mm/s	[0…1]
[Property] AccelerationUnit	[IRI] https://admin-shell.io/idta/ControlConfig/ AccelerationUnit /1/0 Defines the acceleration unit unit of the CNC control	[String] mm/s²	[0…1]
[Property] JerkUnit	[IRI] https://admin-shell.io/idta/ControlConfig/JerkUnit/1/0 Defines the jerk unit of the CNC control	[String] m/s³	[0…1]
[SMC] ControlConfigData __00__	[IRI] https://admin-shell.io/idta/ControlConfig/ControlConfigData/1/0 The SMC “ControlConfigData” is a collection to store configuration data for the main part of the control of CNC machines.	n/a	[0…1]
[SMC] Channels __00__	[IRI] https://admin-shell.io/idta/ControlConfig/Channels/1/0 The SMC “Channels” is a collection of channels of the CNC controller.	n/a	[0…1]
[SMC] NCProgramConfig __00__	[IRI] https://admin-shell.io/idta/ControlConfig/NCProgramConfig/1/0 The SMC “NCProgramConfig” is a collection of NC program configuration information.	n/a	[0…1]
[SMC] Communication Config __00__	[IRI] https://admin-shell.io/idta/ControlConfig/CommunicationConfig/1/0 The SMC “CommunicationConfig” is a collection of communication configuration information.	n/a	[0…1]
[SMC] Tools __00__	[IRI] https://admin-shell.io/idta/ControlConfig/Tools/1/0 Explanation: The SMC “Tools” is a collection of tools of the CNC controller.	n/a	[0…1]
[SMC] Axes __00__	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/1/0 The SMC “Axes” is a collection of axes of the CNC controller.	n/a	[0…*]

2.3 Properties of the SMC “ControlConfigData”

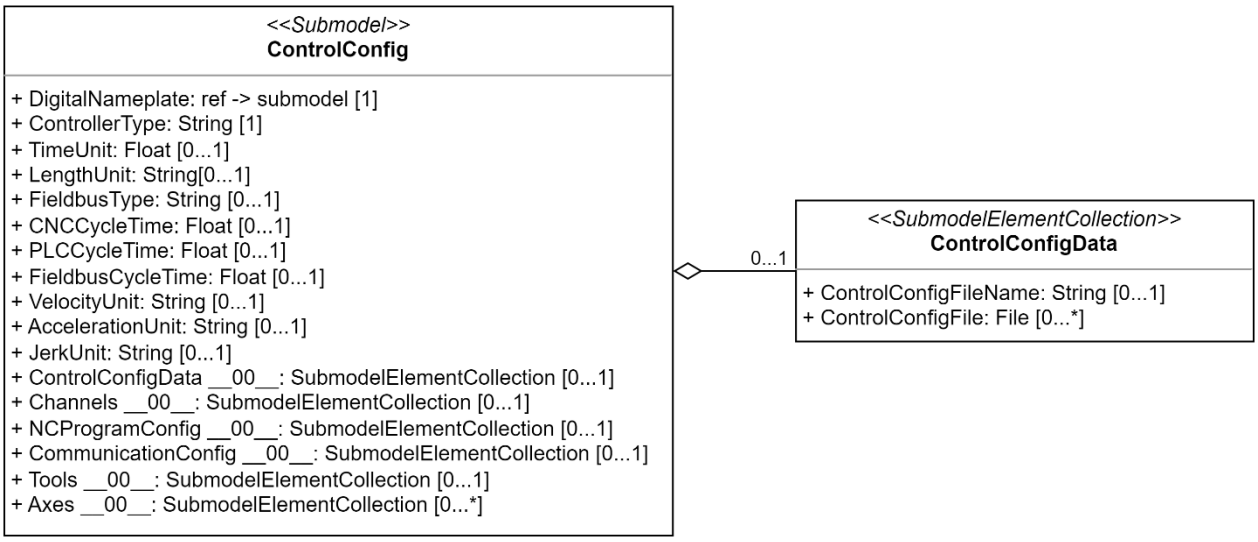

Figure 5: UML-diagram for the SMC "ControlConfigData"

Table 3: Properties of Submodel element collection “ControlConfigData”

idShort:	ControlConfigData __00__ Note: a different idShort might be used, as long as it is unqiue in the Submodel.		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/ControlConfig/ControlConfigData/1/0		
Parent:	Submodel “ControlConfig”		
Explanation:	The SMC “ControlConfigData” is a collection to store configuration data for the main part of the control of CNC machines.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] ControlConfigFileName	[IRI] https://admin-shell.io/idta/ControlConfig/ControlConfigFileName /1/0 Name of the control configuration file	[String]	[0…1]
[File][1] ControlConfigFile	[IRI] https://admin-shell.io/idta/ControlConfig/ControlConfigFile/1/0 File to configure the control	[File]	[0…*]

[1] Recommendation: property declaration as MLP is required by its semantic definition. As the property value is language independent, users are recommended to provide maximal 1 string in any language of the user’s choice.

2.4 Properties of the SMC “Channels”

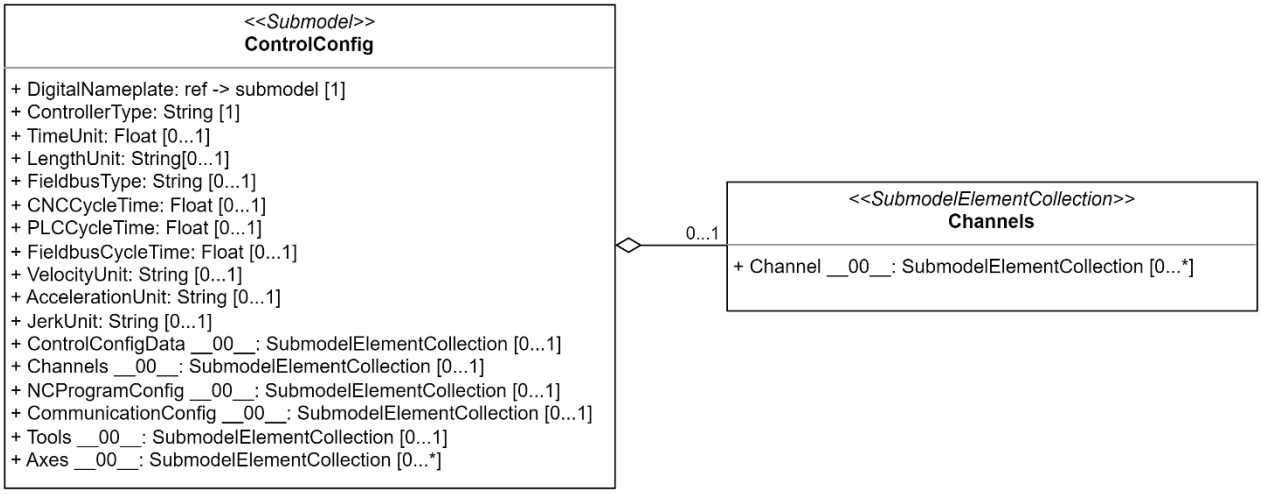

Figure 6: UML-diagram for the SMC "Channels"

Table 4: Properties of SMC "Channels"

idShort:	Channels __00__ Note: a different idShort might be used, as long as it is unqiue in the Submodel.		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/ControlConfig/Channels/1/0		
Parent:	Submodel “ControlConfig”		
Explanation:	The SMC “Channels” is a collection of channels of the CNC controller.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[SMC] Channel __00__	[IRI] https://admin-shell.io/idta/ControlConfig/Channel/1/0 Configuration information of a specific channel	n/a	[0…*]

2.5 Properties of the SMC “Channel”

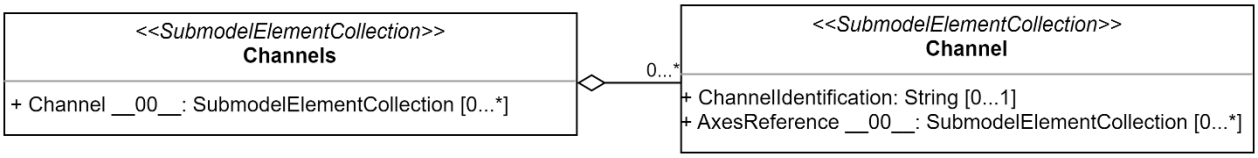

Figure 7: UML-diagram for the SMC "Channel"

Table 5: Properties of SMC "Channel"

idShort:	Channel __00__ Note: a different idShort might be used, as long as it is unqiue in the Submodel.		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/ControlConfig/Channels/Channel/1/0		
Parent:	Channels		
Explanation:	The SMC “Channel” is a collection of channel informations of the CNC controller.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] ChannelIdentification	[IRI] https://admin-shell.io/idta/ControlConfig/Channels/ChannelIdentification/1/0 This describes the unique channel identification of the specific channel	[String] Channel1	[0…1]
[SMC] AxesReference __00__	[IRI] https://admin-shell.io/idta/ControlConfig/Channels/AxesReference/1/0 The SMC “AxesReference” is a collection of referenced axes for the specific channel.	n/a	[0…*]

2.6 Properties of the SMC “AxesReference”

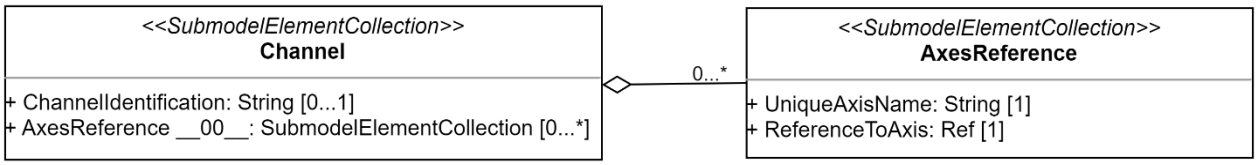

Figure 8: UML-diagram for the SMC " AxesReference "

Table 6: Properties of SMC " AxesReference "

idShort:	AxesReference __00__ Note: a different idShort might be used, as long as it is unqiue in the Submodel.		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/ControlConfig/Channels/AxesReference/1/0		
Parent:	Channel		
Explanation:	The SMC “AxesReference” is a collection of referenced axes for the specific channel.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] AxisName	[IRI] https://admin-shell.io/idta/ControlConfig/Channels/AxesReference/1/0 Unique axis name	String	[1]
[Ref] ReferenceToAxis	[IRI] https://admin-shell.io/idta/ControlConfig/Channels/ReferenceToAxis/1/0 Reference to the axis	n/a	[1]

2.7 Properties of the SMC “Axes”

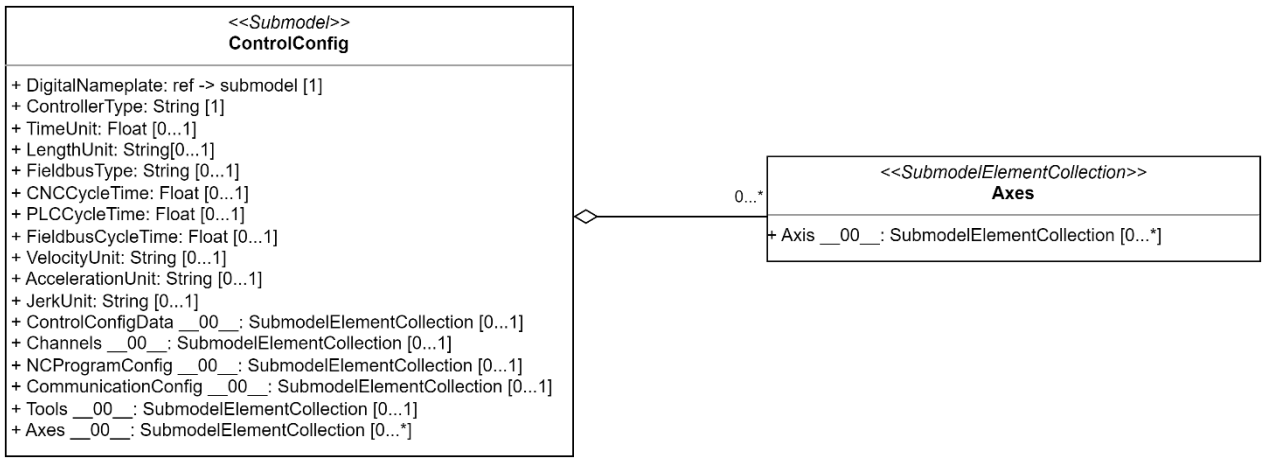

Figure 9: UML-diagram for the SMC "Axes"

Table 7: Properties of SMC "Axes”

idShort:	Axes __00__ Note: a different idShort might be used, as long as it is unqiue in the Submodel.		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/1/0		
Parent:	Channel		
Explanation:	The SMC “Axes” is a collection of axes of the CNC controller.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[SMC] Axis __00__	[IRI] https://admin-shell.io/idta/ControlConfig/Axis/1/0 The SMC “Axis” is a collection of axis configuration information.	n/a	[0…*]

2.8 Properties of the SMC “Axis”

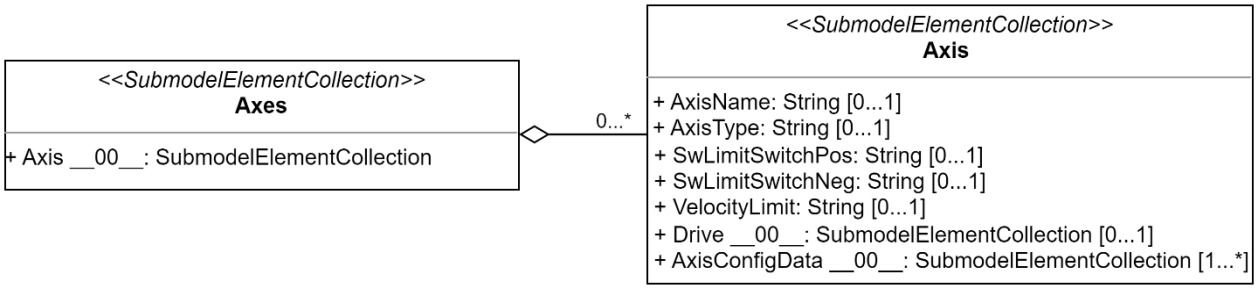

Figure 10: UML-diagram for the SMC “Axis”

Table 8: Properties of SMC “Axis”

idShort:	Axis __00__ Note: a different idShort might be used, as long as it is unqiue in the Submodel.		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/Axis/1/0		
Parent:	Axes		
Explanation:	The SMC “Axis” is a collection of axis configuration information.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] AxisName	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/AxisName/1/0 Name of the axis	[String] Axis1 @en	[0…1]
[Property] AxisType	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/AxisType/1/0 Type of the axis	[String] Linear axis, Rotary axis	[0…1]
[Property] SimulationAxis	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/SimulationAxis /1/0 Axis is a simulated or real one	[Boolean] True, False	[0…1]
[Property] SwLimitSwitchPos	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/SwLimitSwitchPos/1/0 Positive software limit switch	[Float]	[0…1]
[Float] SwLimitSwitchNeg	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/SwLimitSwitchNeg/1/0	[Float]	[0…1]

	Negative software limit switch		
[Float] VelocityLimit	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/VelocityLimit/1/0 Velocity limit of the axis	[Float]	[0…1]
[Float] AccelerationLimit	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/AccelerationLimit/1/0 Accelerationlimit of the axis	[Float]	[0…1]
[Float] JerkLimit	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/JerkLimit/1/0 Jerk limit of the axis	[Float]	[0…1]
[SMC] Drive __00__	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/Drive/1/0 The SMC “Drive” is a collection of drive configuration information.	n/a	[0…1]
[SMC] AxisConfigData __00__	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/AxisConfigData/1/0 The SMC “AxisConfigData” is a collection of axis configuration data.	n/a	[1…*]

2.9 Properties of the SMC “AxisConfigData”

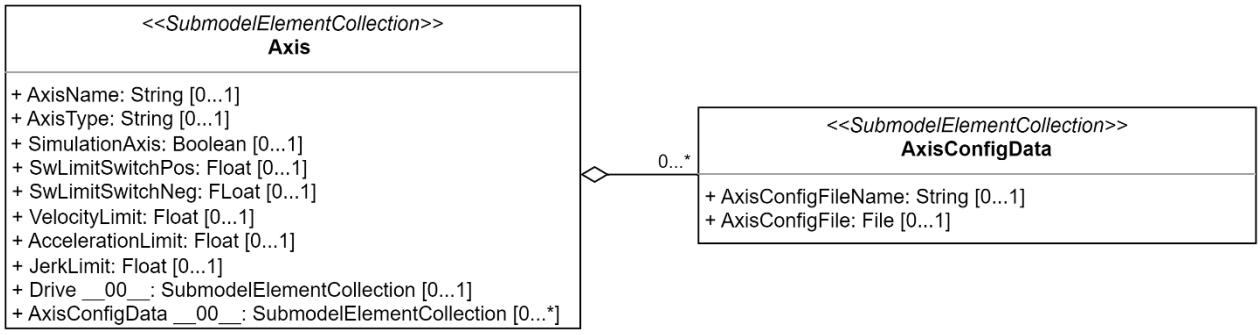

Figure 11: UML-diagram for the SMC “AxisConfigData”

Table 9: Properties of SMC “AxisConfigData”

idShort:	AxisConfigData __00__ Note: a different idShort might be used, as long as it is unqiue in the Submodel.		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/AxisConfigData/1/0		
Parent:	Axis		
Explanation:	The SMC “AxisConfigData” is a collection of axis configuration data.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] AxisConfigFileName	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/AxisConfigFileName/1/0 Axis configuration file name	[String]	[0…1]
[File] AxisConfigFile	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/AxisConfigFile/1/0 Configuration file of the axis	n/a	[0…1]

2.10 Properties of the SMC “Drive”

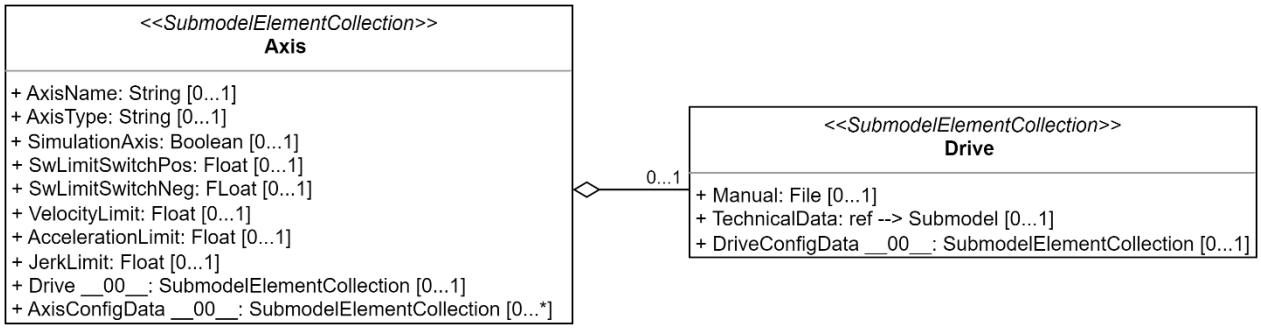

Figure 12: UML-diagram for the SMC “Drive”

Table 10 : Properties of SMC “Drive”

idShort:	Drive __00__ Note: a different idShort might be used, as long as it is unqiue in the Submodel.		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/Axis/Drive/1/0		
Parent:	Axis		
Explanation:	The SMC “Drive” is a collection of drive configuration information.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[File] Manual	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/Axis/Manual/1/0 The manual file of the drive	n/a	[0…1]
[Ref] TechnicalData	[IRI] https://admin-shell.io/IDTA/TechnicalData/Submodel/2/0 Reference to the Submodel Technical Specification of the axis to gather more informations about e.g.: • Drive transmission • Nominal Torque • Nominal Speed • Firmware File …	n/a	[0…1]
[SMC] DriveConfigData __00__	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/Axis/DriveConfigData/1/0 The SMC “DriveConfigData” is a collection of drive configuration data.	n/a	[0…1]

2.11 Properties of the SMC “DriveConfigData”

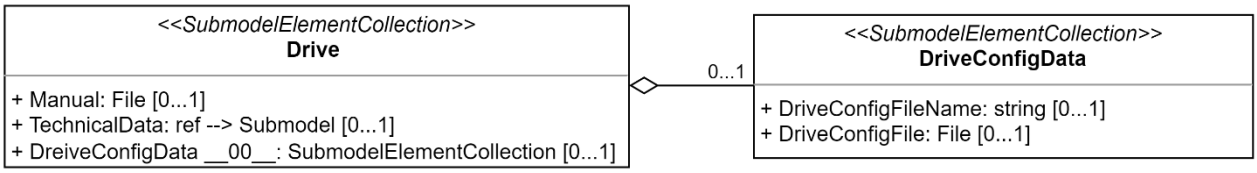

Figure 13: UML-diagram for the SMC “DriveConfigData”

Table 11: Properties of SMC “DriveConfigData”

idShort:	DriveConfigData __00__ Note: a different idShort might be used, as long as it is unqiue in the Submodel.		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/Axis/Drive/DriveConfigData/1/0		
Parent:	Drive		
Explanation:	The SMC “DriveConfigData” is a collection of drive configuration data.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] DriveConfigFileName	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/Axis/Drive/DriveConfigFileName/1/0 Name of the configuration file of the drive	[String]	[0…1]
[File] DriveConfigFile	[IRI] https://admin-shell.io/idta/ControlConfig/Axes/Axis/Drive/DriveConfigFile/1/0 Configuration File of the drive	n/a	[0…1]

2.12 Properties of the SMC “NCProgramConfig”

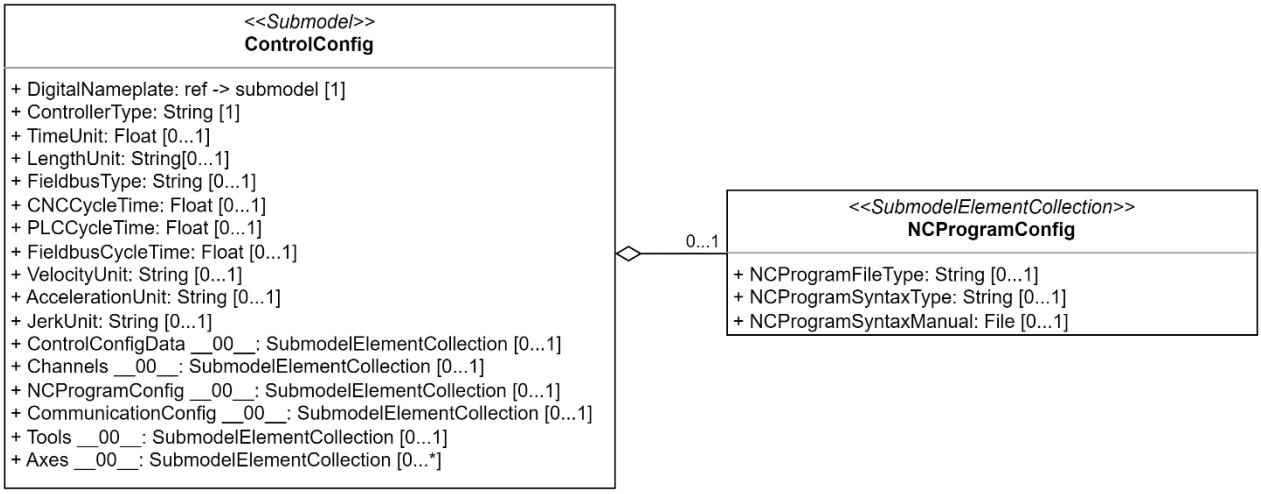

Figure 14: UML-diagram for the SMC "NCProgramConfig"

Table 12: Properties of SMC "NCProgramConfig”

idShort:	NCProgramConfig __00__ Note: a different idShort might be used, as long as it is unqiue in the Submodel.		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/ControlConfig/NCProgramConfig/1/0		
Parent:	ControlConfig		
Explanation:	The SMC “NCProgramConfig” is a collection of NC program configuration information.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] NCProgramFileType	[IRI] https://admin-shell.io/idta/ControlConfig/NCProgramFileType/1/0 NC program file type, which the CNC control can work with	[String] .nc	[0…1]
[Property] NCProgramSyntaxType	[IRI] https://admin-shell.io/idta/ControlConfig/NCProgramSyntaxType/1/0 NC program syntax type	[String] DIN 66025	[0…1]
[File] NCProgramSyntaxManual	[IRI] https://admin-shell.io/idta/ControlConfig/NCProgramSyntaxManual/1/0 Manual file of the syntax of the NC program	n/a	[0…1]

2.13 Properties of the SMC “CommunicationConfig”

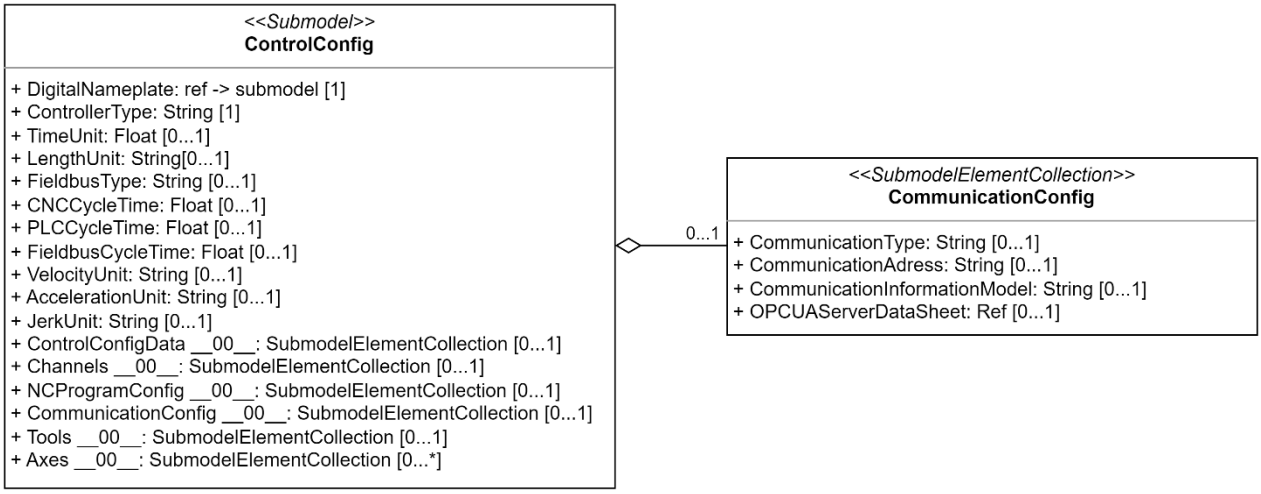

Figure 15: UML-diagram for the SMC "CommunicationConfig"

Table 13: Properties of SMC "CommunicationConfig”

idShort:	CommunicationConfig __00__ Note: a different idShort might be used, as long as it is unqiue in the Submodel.		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/ControlConfig/CommunicationConfig/1/0		
Parent:	ControlConfig		
Explanation:	The SMC “CommunicationConfig” is a collection of communication configuration information.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] Communication Type	[IRI] https://admin-shell.io/idta/ControlConfig/CommunicationType/1/0 Type of communication to the control	[String] MQTT	[0…1]
[Property] Communication Adress	[IRI] https://admin-shell.io/idta/ControlConfig/CommunicationAdress/1/0 Communication address into the control	[String]	[0…1]
[Property] CommunicationInformationModel	[IRI] https://admin-shell.io/idta/ControlConfig/CommunicationInformationModel/1/0 Communication information model if ones exist like OPCUA	[String] OPCUA	[0…1]
[Ref]	[IRI] https://admin-shell.io/idta/opcua-server-datasheet/1/0	n/a	[0…1]

OPCUAServerDataSheet	Reference to the OPCUA server datasheet Submodel		

2.14 Properties of the SMC “Tools”

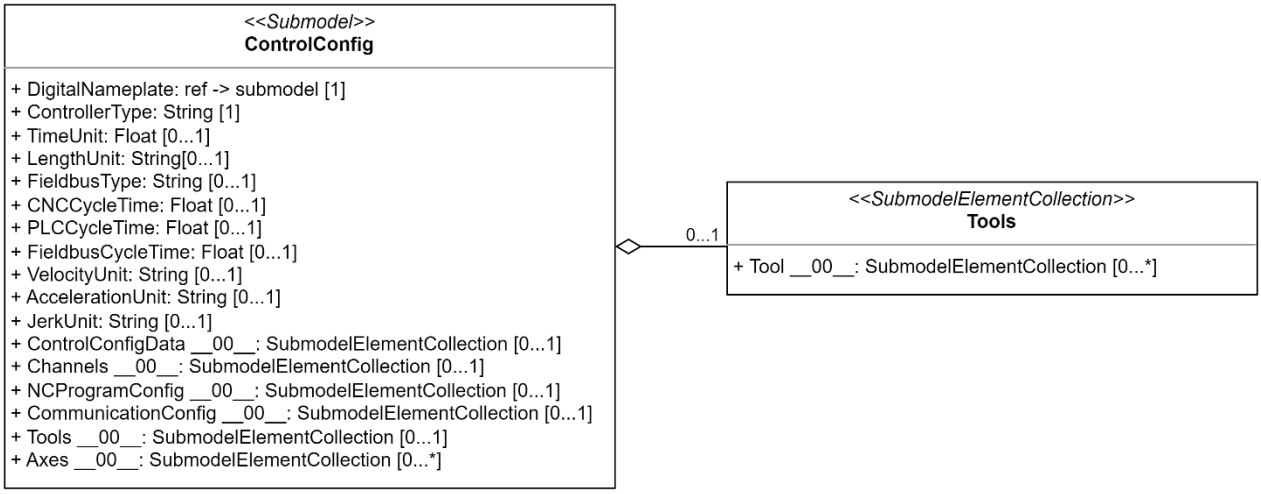

Figure 16: UML-diagram for the SMC "Tools"

Table 14: Properties of SMC "Tools”

idShort:	Tools __00__ Note: a different idShort might be used, as long as it is unqiue in the Submodel.		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/ControlConfig/Tools/1/0		
Parent:	ControlConfig		
Explanation:	The SMC “Tools” is a collection of tools of the CNC controller.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[SMC] Tool __00__	[IRI] https://admin-shell.io/idta/ControlConfig/Tool/1/0 The SMC “Tool” is a collection of tool configuration information.	n/a	[0…*]

2.15 Properties of the SMC “Tool”

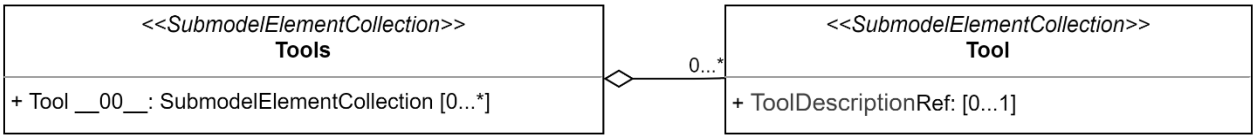

Figure 17: UML-diagram for the SMC "Tool"

Table 15: Properties of SMC "Tool”

idShort:	Tool __00__ Note: a different idShort might be used, as long as it is unqiue in the Submodel.		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/ControlConfig/Tool/1/0		
Parent:	Tools		
Explanation:	The SMC “Tool” is a collection of tool configuration information.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Ref] ToolDescription	[IRI] https://admin-shell.io/idta/ToolDescription/1/0 Reference to the tool description Submodel of the specific tool.	n/a	[0…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.

Bibliography

[1] "Recommendations for implementing the strategic initiative INDUSTRIE 4.0", acatech, April 2013. [Online]. Available Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Final report of the Industrie 4.0 Working Group - acatech - National Academy of Science and Engineering

[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: Implementation Strategy Industrie 4.0: Report on the results of the Industrie 4.0 Platform | Leitfaden 2016 | Bitkom e. V.

[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 Asset Administration Shell Specification - Part 1: Metamodel (industrialdigitaltwin.org)

www.industrialdigitaltwin.org