SPECIFICATION

Submodel AI-Deployment

Version 1.0

23 06 2023

Submodel Template of the
Asset Adminstration Shell

Imprint

Publisher

Steinbeis Innovation gGmbH
Adornostr. 8
70599 Stuttgart
Germany

Source for Specification Document

Plattform Industrie 4.0
Bertolt-Brecht-Platz 3
10117 Berlin
Germany

Authors

Mike Reichardt
Daniel Buch
Philip Stricker
Dachuan Shi
Georg Güntner
Sebastian Baron
Christoph Legat
Jürgen Lenz
Nico Braunisch

Version history

2023-06-23	1.0	Submission to the InterOpera Consortium

Content

List of Figures

List of Tables

Foreword

We would like to thank all of the working group members for their help and support to develop the AI Deployment Submodel. The discussions were always helpful and we got lots of good ideas out of it.
A special thanks goes to the Interopera Consortium, that provided us with the opportunity to develop the AI Deployment Submodel.

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 [1-3] 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 [1-6].

The target group of the specification are developers and editors of ML applications and groups, that would like to deploy models, which are describing the deployment process by means of the Asset Administration Shell (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 deployment of AI models in regard to the asset of the respective Asset Administration Shell. Central element is the provision of properties [7], ideally interoperable by the means of dictionaries such as ECLASS and IEC CDD (Common Data Dictionary). 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 the assistance of deployment of AI models in a standardized way. Currently the focus is on an industrial environment.

The intended use-case is the provision of a standardized property structure for deploying AI, which enables an easier usage of AI in an industrial environment. With the assistance of two other AI Submodels (AI Dataset and AI ModelNameplate), it provides an overview of the whole AI lifecycle and allows an easier management of it.

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.

1.3 Relevant standards and sources of concepts for the Submodel template

According to [3], interoperable properties might be defined by standards, consortium specifications or manufacturer specifications. So called property dictionaries are used to 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.

Information set for Submodel AI Deployment

While defining Submodels the following three aspects must be considered as suggested in [5]:

Use and economic relevance

The Submodel AI Deployment is designed for assisting the deployment of AI models. This helps the growth and usage of AI in all kind of fields, e.g. manufacturing. An usage of AI reduces costs, because it allows to automatize processes.

One use case is the initial collection of all parameters and information needed to integrate an AI model into a usable application. The Submodel contains, among other things, information about the access possibilities to the model in the form of the storage location. In addition, information about the input and output of the AI model is available in the Submodel “AI Deployment”. Information on the hardware requirements needed is also provided by the “AI Deployment” Submodel. Alongside passive parameters that describe the configuration and access options to the model, data is also recorded in the Submodel that provides information about active use, e.g. the runtime.

Possible functions and interactions

The Submodel “AI Deployment” provides information for the deployment of AI models. Engineers in the manufacturing environment can use the Submodel to obtain information about the deployment of the AI model. This includes, for example, the storage location of the model. In addition, AI model developers can use the Submodel to provide information about the required hardware and thus actively influence the planning process of a production line, for example. In the operating case, parameters of the model currently in use are displayed in the Submodel. This allows workers on the production line to evaluate the functionality and performance of the model using the observed parameters.

The SMC "Storage" contains the information about the storage location of the deployable model. Due to the design of the SMC, it is possible to save the deployable model directly as a file in the “AI Deployment” Submodel.

The two SMCs "Input" and "Output" serve a similar purpose. Here, the SMC "Input" contains the input variables of the deployable model, while the SMC "Output" contains the output variables. Both SMCs are designed in such a way that the users of the Submodel have the freedom of representation of the input and output variables.

The SMC "Software Requirement" and the SMC "Hardware Requirement" deal with the requirements necessary for the operation of the model. Here, the SMC "SoftwareRequirements" enables the upload of a requirements file, e.g. a requriement.txt, as used in venv in Python.

The SMC "PerformanceInformation" and its child "InferenceTime" allow inference times to be recorded for specific hardware configurations. This allows the selection of specific hardware whom the deployable model must run with a specific inference time in the application. In order to ensure the monitoring of the running model, there exists the SMC LiveMonitoring. It contains the SMC "HardwareWorkload", which shows the current utilization of the hardware. For detecting changes in the data and in the model, there is the SMC "DriftMetrics".

Property specification

See section 3 Submodel and Collections.

Submodel and Collections

3.1 Properties of the Submodel “Deployment”

The figure below shows the UML-diagram defining the relevant properties which need to be set. Table 1: Properties of Submodel “Deployment" describes the details of the Submodel structure combined with examples.

Figure 1: UML-Diagram for Submodel "Deployment"

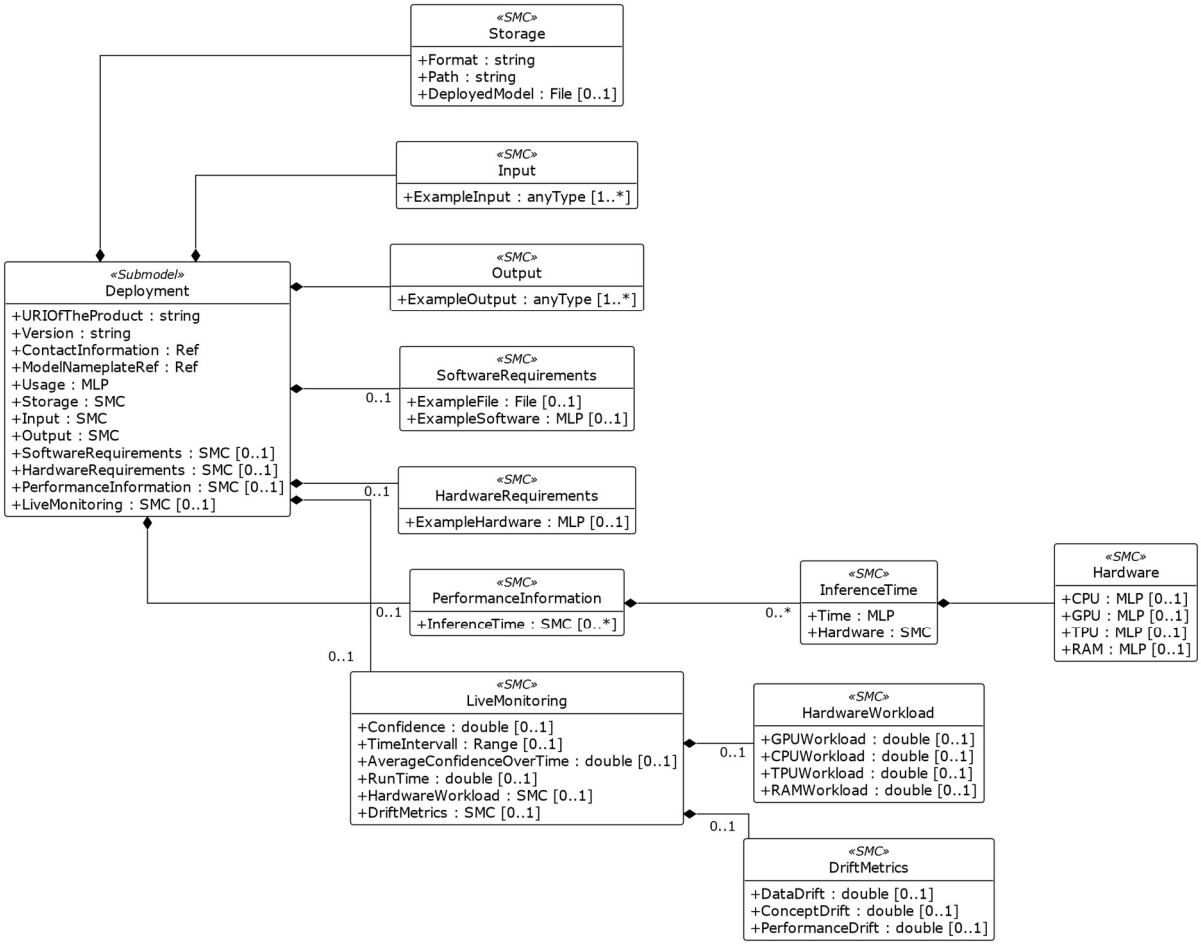

Table 1: Properties of Submodel “Deployment"

idShort:	Deployment		
Class:	Submodel		
semanticId:	[https://admin-shell.io/id/InterOpera/Deployment]		
Parent:			
Explanation:			
[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[Property] URIOfTheProduct	[IRDI]0173-1#02-ABH173#001 URI of the product	[string]	1

[Property] Version	[IRDI]0173-1#02-AAS354#002 Version	[string]	1
[Ref] ContactInformation	[IRI]https://admin-shell.io/id/InterOpera/AIDataset/Contactinformation Reference to the Contact Information IDTA Submodel to describe the responsible person for the Submodel	[] 	1
[Ref] ModelNameplateRef	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/ModelNameplateRef Reference to the ModelNameplate Submodel of the deployed model	[] 	1
[MLP] Usage	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/Usage Usage of the deployed application	[] @en	1
[SMC] Storage	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/Storage Collection of storage information about the deployable AI-model	[] 3 elements	1
[SMC] Input	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/Input Collection of information about the input data for deployment	[] 1 elements	1
[SMC] Output	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/Output Collection of information about the output data for deployment	[] 1 elements	1
[SMC] SoftwareRequirements	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/SoftwareRequirements Collection about required software for deployment	[] 2 elements	0..1
[SMC] HardwareRequirements	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/HardwareRequirements	[] 1 elements	0..1

	Collection about required hardware for deployment		
[SMC] PerformanceInformation	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/PerformanceInformation Collection of performance measurements	[] 1 elements	0..1
[SMC] LiveMonitoring	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/LiveMonitoring Collection containing live values for monitoring	[] 6 elements	0..1

3.2 Properties of the SMC “Storage”

Figure 1: UML-Diagram for Submodel "Deployment" shows the UML-diagram defining the relevant properties which need to be set. The following table describes the details of the SMC structure.

Table 2: Properties of SMC " Storage"

idShort:	Storage		
Class:	SubmodelElementCollection		
semanticId:	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/Storage		
Parent:	AIDeployment		
Explanation:	Collection of storage information about the deployable AI-model@en		
[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[Property] Format	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/Storage/Format Data format of the deployable model (e.g. onnx, docker-image ,...)	[string]	1
[Property] Path	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/Storage/Path Path to the deployable model	[string]	1
[File] DeployedModel	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/Storage/DeployedModel The deployable model	[]	0..1

3.3 Properties of the SMC “Input”

Figure 1: UML-Diagram for Submodel "Deployment" shows the UML-diagram defining the relevant properties which need to be set. The following table describes the details of the SMC structure.

Table 3: Properties of SMC " Input"

idShort:	Input		
Class:	SubmodelElementCollection		
semanticId:	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/Input		
Parent:	AIDeployment		
Explanation:	Collection of information about the input data for deployment@en		
[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[Property] ExampleInput	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/Input/InputProperty Input information of the deployment	[anyType]	1..*

3.4 Properties of the SMC “Output”

Figure 1: UML-Diagram for Submodel "Deployment" shows the UML-diagram defining the relevant properties which need to be set. The following table describes the details of the SMC structure.

Table 4: Properties of SMC " Output"

idShort:	Output		
Class:	SubmodelElementCollection		
semanticId:	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/Output		
Parent:	AIDeployment		
Explanation:	Collection of information about the output data for deployment@en		
[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[Property] ExampleOutput	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/Output/OutputProperty Output information of the deployment	[anyType]	1..*

3.5 Properties of the SMC “SoftwareRequirements”

Figure 1: UML-Diagram for Submodel "Deployment" shows the UML-diagram defining the relevant properties which need to be set. The following table describes the details of the SMC structure.

Table 5: Properties of SMC " SoftwareRequirements"

idShort:	SoftwareRequirements		
Class:	SubmodelElementCollection		
semanticId:	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/SoftwareRequirements		
Parent:	AIDeployment		
Explanation:	Collection about required software for deployment@en		
[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[File] ExampleFile	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/SoftwareRequirements/RequirementsFile Requirement file for software requirements	[]	0..1
[MLP] ExampleSoftware	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/SoftwareRequirements/SoftwareDescription Description of the required software	[] @en	0..1

3.6 Properties of the SMC “HardwareRequirements”

Figure 1: UML-Diagram for Submodel "Deployment" shows the UML-diagram defining the relevant properties which need to be set. The following table describes the details of the SMC structure.

Table 6: Properties of SMC " HardwareRequirements"

idShort:	HardwareRequirements		
Class:	SubmodelElementCollection		
semanticId:	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/HardwareRequirements		
Parent:	AIDeployment		
Explanation:	Collection about required hardware for deployment@en		
[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	

[MLP] ExampleHardware	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/HardwareRequirements/HardwareDescription Description of the required hardware	[] @en	0..1

3.7 Properties of the SMC “PerformanceInformation”

Figure 1: UML-Diagram for Submodel "Deployment" shows the UML-diagram defining the relevant properties which need to be set. The following table describes the details of the SMC structure.

Table 7: Properties of SMC " Storage"

idShort:	PerformanceInformation		
Class:	SubmodelElementCollection		
semanticId:	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/PerformanceInformation		
Parent:	AIDeployment		
Explanation:	Collection of performance measurements@en		
[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[SMC] InferenceTime	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/PerformanceInformation/InferenceTime Collection of information about the measured inference time and the used equipment to achieve it	[] 2 elements	0..*

3.8 Properties of the SMC “InferenceTime”

Figure 1: UML-Diagram for Submodel "Deployment" shows the UML-diagram defining the relevant properties which need to be set. The following table describes the details of the SMC structure.

Table 8: Properties of SMC " InferenceTime"

idShort:	InferenceTime		
Class:	SubmodelElementCollection		
semanticId:	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/PerformanceInformation/InferenceTime		
Parent:	PerformanceInformation		
Explanation:	Collection of information about the measured inference time and the used equipment to achieve it@en		

[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[MLP] Time	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/PerformanceInformation/InferenceTime/Time Measured inference time	[] @en	1
[SMC] Hardware	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/PerformanceInformation/InferenceTime/Hardware Collection of hardware used for the measurement	[] 4 elements	1

3.9 Properties of the SMC “Hardware”

Figure 1: UML-Diagram for Submodel "Deployment" shows the UML-diagram defining the relevant properties which need to be set. The following table describes the details of the SMC structure.

Table 9: Properties of SMC " Hardware"

idShort:	Hardware		
Class:	SubmodelElementCollection		
semanticId:	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/PerformanceInformation/InferenceTime/Hardware		
Parent:	Hardware		
Explanation:	Collection of hardware used for the measurement@en		
[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[MLP] CPU	[IRDI]0173-1#01-ACL481#008 Processor (computer module) Information about the CPU.	[] @en	0..1
[MLP] GPU	[IRDI]0173-1#01-ACL476#008 Graphic card (computer module) Electronic assembly as a component of a computer that is responsible for generating the signals for the display or the external monitor Information about the GPU.	[] @en	0..1
[MLP] TPU	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/PerformanceInformation/InferenceTime/Hardware/TPU Name of the used tensor processor unit (TPU)	[] @en	0..1

[MLP] RAM	[IRDI]0173-1#01-ACL340#008 Random access memory (computer) Memory in a computer in which the data to be processed and the running programs (software) are stored or kept Information about the RAM.	[] @en	0..1

3.10 Properties of the SM “LiveMonitoring”

Figure 1: UML-Diagram for Submodel "Deployment" shows the UML-diagram defining the relevant properties which need to be set. The following table describes the details of the SMC structure.

Table 10: Properties of SMC " LiveMonitoring"

idShort:	LiveMonitoring		
Class:	SubmodelElementCollection		
semanticId:	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/LiveMonitoring		
Parent:	AIDeployment		
Explanation:	Collection containing live values for monitoring@en		
[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[Property] Confidence	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/LiveMonitoring/Confidence Live confidence	[double]	0..1
[Range] TimeIntervall	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/LiveMonitoring/TimeIntervall Time interval for calculating the average confidence	[] ..	0..1
[Property] AverageConfidenceOverTime	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/LiveMonitoring/AverageConfidence Average confidence over a defined time interval	[double]	0..1
[Property] RunTime	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/LiveMonitoring/RunTime	[double]	0..1

	Time the deployed model is running without a break or downtime		
[SMC] HardwareWorkload	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/LiveMonitoring/HardwareWorkload Collection containing the live hardware workload	[] 4 elements	0..1
[SMC] DriftMetrics	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/LiveMonitoring/DriftMetrics Collection of live monitoring of different drift metrics	[] 3 elements	0..1

3.11 Properties of the SMC “HardwareWorkload”

Figure 1: UML-Diagram for Submodel "Deployment" shows the UML-diagram defining the relevant properties which need to be set. The following table describes the details of the SMC structure.

Table 11: Properties of SMC " HardwareWorkload"

idShort:	HardwareWorkload		
Class:	SubmodelElementCollection		
semanticId:	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/LiveMonitoring/HardwareWorkload		
Parent:	LiveMonitoring		
Explanation:	Collection containing the live hardware workload@en		
[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[Property] GPUWorkload	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/LiveMonitoring/HardwareWorkload/GPUWorkload Live GPU workload	[double]	0..1
[Property] CPUWorkload	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/LiveMonitoring/HardwareWorkload/CPUWorkload Live CPU workload	[double]	0..1
[Property] TPUWorkload	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/LiveMonitoring/HardwareWorkload/TPUWorkload	[double]	0..1

	Live TPU workload		
[Property] RAMWorkload	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/LiveMonitoring/HardwareWorkload/RAMWorkload Live RAM workload	[double]	0..1

3.12 Properties of the SMC “DriftMetrics”

Figure 1: UML-Diagram for Submodel "Deployment" shows the UML-diagram defining the relevant properties which need to be set. The following table describes the details of the SMC structure.

Table 12: Properties of SMC " DriftMetrics "

idShort:	DriftMetrics		
Class:	SubmodelElementCollection		
semanticId:	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/LiveMonitoring/DriftMetrics		
Parent:	LiveMonitoring		
Explanation:	Collection of live monitoring of different drift metrics@en		
[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[Property] DataDrift	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/LiveMonitoring/DriftMetrics/DataDrift Live data drift	[double]	0..1
[Property] ConceptDrift	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/LiveMonitoring/DriftMetrics/ConceptDrift Live concept drift	[double]	0..1
[Property] PerformanceDrift	[IRI]https://admin-shell.io/id/InterOpera/AIDeployment/LiveMonitoring/DriftMetrics/PerformanceDrift Live performance drift	[double]	0..1

Annex A: Explanations on used table formats

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.

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

SME type Submodel	Element 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@de.
- The [valueType] is only given for Properties.

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: https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/Details-of-the-Asset-Administration-Shell-Part1.html

[7] “Semantic interoperability: challenges in the digital transformation age”; IEC, International Electronical Commission; 2019. [Online]. Available:https://basecamp.iec.ch/download/iec-white-paper-semantic-nteroperability-challenges-in-the-digital-transformation-age-en/