IDTA 02060-1-0
Artificial Intelligence Model Nameplate

February 2025

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
25.02.2025	1.0	Release of the official Submodel template published by IDTA.

Contents

Figures

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 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 Artificial Intelligence (AI) model information 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 to assist the AI model documentation and helps to manage an AI lifecycle.

The intended use-case is the provision of a standardized property structure for documenting AI models, which enables an efficient management of the different models and helps to reuse already trained models. On top of that it contains information about the responsible person for the model and helps to gain additional information of the model from him.

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 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 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.

2 Submodel AIModelNameplate

2.1 Approach

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

2.1.1 Use and economic relevance

The Submodel Artificial Intelligence (AI) ModelNameplate is designed for documenting the training process of AI models. The documentation helps to reuse trained model and reduces the training time of models, because a retraining of a model is, in general, less time consuming than a training from scratch.

On top of that, it assists with the communication within a company. The Submodel contains information about a responsible person for this model. Therefore additional information can be asked to the responsible person directly.

One use case is the collection of the most important parameters and model variables of an AI model. The collection process is triggered at the same time as the start of the AI training. The triggered training process monitors the hyperparameters and the training and stores the desired quantities in the "AI ModelNameplate" Submodel. As a result, all the information required for the training is available in one place, namely in the Submodel.

2.1.2 Possible functions and interactions

The Submodel "AI ModelNameplate" contains information about the AI model. AI model developers and data scientists can use the Submodel to obtain detailed information about the model. Information available includes, for example, the training configuration, the storage location and the data set used. In addition to data scientists, people in the administration are also interested in the Submodel. Within the Submodel, contact information about the person responsible for the model is stored. This supports an orderly management structure even in larger companies.

By using the SMC "Inputs" and the child SMC "Dimension", information about the input variables is recorded. Analogous to the input variables, important meta information of the output variables is summarized in the SMC "Outputs".

The results of the training are recorded in the SMC "TrainingResult". The SMC is developed in such a way that the developer has the greatest possible freedom to record his specific results.

The SMC "Plots" contains all graphics assigned to the model. The SMC "Plots" is divided into three SMCs. The SMC "TrainingTesting" contains the plots created in the training and testing process. The second SMC "Structure" describes the model structure through graphics. To allow the users of the Submodel freedom of design, the SMC "GeneralPlots" has been added as a third SMC.

The SMC "Details" provides general information such as the programming language for the training or the required modules. In a model development, the data form a central element. To cover this element, the SMC "AIDataset" exists, which includes the reference and access times to the AIDataset.

In order to do justice to the different AI methods, there is the SMC "AITypeSpecificInformation". Within this SMC, there is a separate SMC for hyperparameters and transfer learning.

2.1.3 Property specification

See section 3 Submodel and Collections.

3 Submodell and collections

3.1 Properties of the Submodel "AI ModelNameplate"

The figure below shows the UML-diagram defining the relevant properties which need to be set. Table 1: Properties of Submodel "AIModelNameplate" describes the details of the Submodel structure.
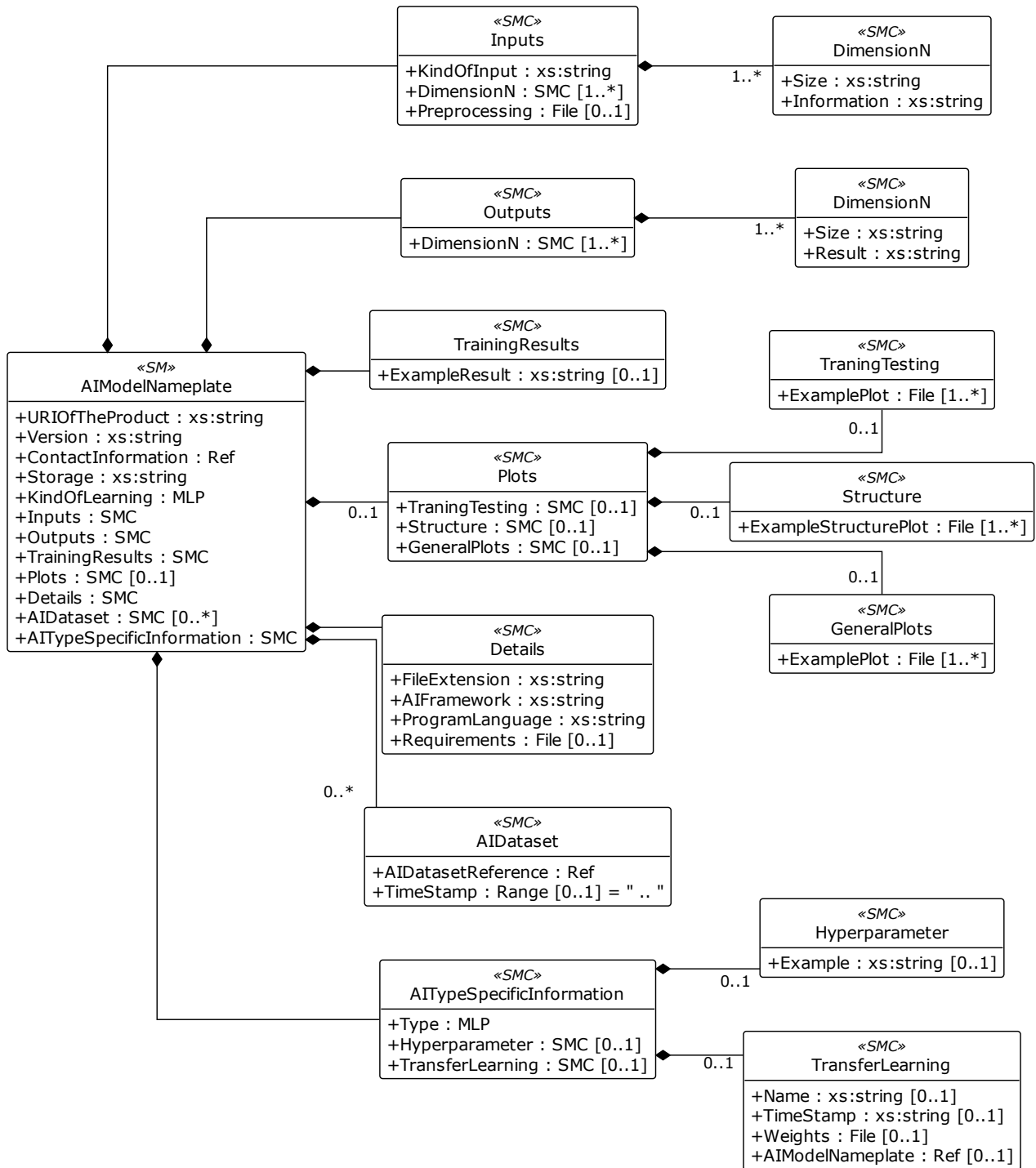
Figure 1: UML Diagram of Properties of the Submodel "AI ModelNameplate"

Table 1: Properties of Submodel "AI ModelNameplate"

idShort	AIModelNameplate
Class:	Submodel
semanticId:	[IRI] https://admin-shell.io/idta/SubmodelTemplate/AIModelNameplate/1/0
Parent:	AAS
Explanation:	Contains information about a trained AI model

[SME type]	semanticId	[valueType]	card.
idShort	**Description@en**	**example**	
[Prop] URIOfTheProduct	[IRDI] 0112/2///61987#ABN590#002 unique global identification of the product instance using an universal resource identifier (URI)	[String] https://www.example.com/search?q=model	[1]
[Prop] Version	[IRDI] 0173-1#02-AAS354#002 Version of the nameplate	[String] 1.0	[1]
[Ref] ContactInformation	[IRI] https://admin-shell.io/idta/AIModelNameplate/Contactinformation/1/0 Reference to the Contact Information IDTA Submodel to describe the responsible person for the Submodel	[-]	[1]
[Prop] Storage	[IRI] https://admin-shell.io/idta/AIModelNameplate/Storage/1/0 Path to the model (e.g. local path, serverpath,...)	[String] C:\Users\example\Desktop\Models\pytorch	[1]
[MLP] KindOfLearning	[IRI] https://admin-shell.io/idta/AIModelNameplate/KindOfLearning/1/0 This property describes the learningmethod of the model (supervised, unsupervised, reinforcement, ...)	[-] Supervised@en	[1]
[SMC] Inputs	[IRI] https://admin-shell.io/idta/AIModelNameplate/Inputs/1/0 Collection of necessary information about the model input	[-] 3 elements	[1]
[SMC] Outputs	[IRI] https://admin-shell.io/idta/AIModelNameplate/Outputs/1/0 Collection of necessary information about the model output	[-] 1 elements	[1]
[SMC] TrainingResults	[IRI] https://admin-shell.io/idta/AIModelNameplate/TrainingResults/1/0 Collection of different training results of the model	[-] 1 elements	[1]

[SMC] Plots	[IRI] https://admin- shell.io/idta/AIModelNameplate/Plots/1/0 Collection of different plots (e.g. about the training or the model itself)	[-] 3 elements	[0..1]
[SMC] Details	[IRI] https://admin- shell.io/idta/AIModelNameplate/Details/1/0 Collection of additional details about the model	[-] 4 elements	[1]
[SMC] AIDataset	[IRI] https://admin- shell.io/idta/AIModelNameplate/AIDataset/1/0 Collection of information about the used AIDataset	[-] 2 elements	[0..1]
[SMC] AITypeSpecificInform ation	[IRI] https://admin- shell.io/idta/AIModelNameplate/AITypeSpecificI nformation/1/0 Collection containing different kinds of AI method specific parameters	[-] 1 elements	[1]

3.2 Properties of SMC "Inputs"

Table 2: Properties of SMC "Inputs"

idShort:	Inputs		
Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/AIModelNameplate/Inputs/1/0		
Parent:	SM AIModelNameplate		
Explanation:	Collection of necessary information about the model input@en		
[SME type]	**semanticId**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Prop] KindOfInput	[IRI] https://admin-shell.io/idta/AIModelNameplate/Inputs/KindOfInput/1/0 Type of data (e.g. Images, Audio)	[String] Images	[1]
[SMC] DimensionN	[IRI] https://admin-shell.io/idta/AIModelNameplate/Inputs/DimensionN/1/0 Collection of information about a single input dimension	[-] 2 elements	[0..*]
[File] Preprocessing	[IRI] https://admin-shell.io/idta/AIModelNameplate/Inputs/Preprocessing/1/0 Preprocessing pipeline as a file	[-]	[0..1]

3.2.1 Properties of SMC "DimensionN"

Table 3: Properties of SMC "DimensionN"

idShort:	DimensionN		
Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/AIModelNameplate/Inputs/DimensionN/1/0		
Parent:	SMC Inputs		
Explanation:	Collection of information about a single input dimension@en		
[SME type]	**semanticId**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Prop] Size	[IRI] https://admin-shell.io/idta/AIModelNameplate/Inputs/DimensionN/Size/1/0 Size of the input dimension	[String] 1	[1]
[Prop] Information	[IRI] https://admin-shell.io/idta/AIModelNameplate/Inputs/DimensionN/Information/1/0 Information about the usage of this dimension	[String]	[0..1]

3.3 Properties of SMC "Outputs"

Table 4: Properties of SMC "Outputs"

idShort:	Outputs		
Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/AIModelNameplate/Outputs/1/0		
Parent:	SM AIModelNameplate		
Explanation:	Collection of necessary information about the model output@en		
[SME type]	**semanticId**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[SMC] DimensionN	[IRI] https://admin-shell.io/idta/AIModelNameplate/Outputs/DimensionN/1/0 Collection of information about a single output dimension	[-] 2 elements	[1..*]

3.3.1 Properties of SMC "DimensionN"

Table 5: Properties of SMC "DimensionN"

idShort:	DimensionN
Class:	SubmodelElementCollection
semanticId:	[IRI] https://admin-shell.io/idta/AIModelNameplate/Outputs/DimensionN/1/0

Parent:	SMC Outputs		
Explanation:	Collection of information about a single output dimension@en		
[SME type]	**semanticId**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Prop] Size	[IRI] https://admin-shell.io/idta/AIModelNameplate/Outputs/DimensionN/Size/1/0 Size of the output dimension	[String] 3	[1]
[MLP] Result	[IRI] https://admin-shell.io/idta/AIModelNameplate/Outputs/DimensionN/Result/1/0 Meaning of the output dimension (e.g. class)	[-] bbox;labels;confidences@en	[1]

3.4 Properties of SMC "TrainingResults"

Table 6: Properties of SMC "TrainingResults"

idShort:	TrainingResults		
Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/AIModelNameplate/TrainingResults/1/0		
Parent:	SM AIModelNameplate		
Explanation:	Collection of different training results of the model@en		
[SME type]	**semanticId**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Prop] ExampleResult	[IRI] https://admin-shell.io/idta/AIModelNameplate/TrainingResults/Result/1/0 Result of the trained AI	[String] 0.96848	[0..1]

3.5 Properties of SMC "Plots"

Table 7: Properties of SMC "Plots"

idShort:	Plots		
Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/AIModelNameplate/Plots/1/0		
Parent:	SM AIModelNameplate		
Explanation:	Collection of different plots (e.g. about the training or the model itself)@en		
[SME type]	**semanticId**	**[valueType]**	**card.**

idShort	Description@en	example	
[SMC] TrainingTesting	[IRI] https://admin-shell.io/idta/AIModelNameplate/Plots/TrainingTesting/1/0 Collection of plots about training results and test results	[-] 1 elements	[0..1]
[SMC] Structure	[IRI] https://admin-shell.io/idta/AIModelNameplate/Plots/Structure/1/0 Collection describing the model architecture (e.g. of a neural net)	[-] 1 elements	[0..1]
[SMC] GeneralPlots	[IRI] https://admin-shell.io/idta/AIModelNameplate/Plots/GeneralPlots/1/0 Collection of plots not fitting in existing collections	[-] 1 elements	[0..1]

3.5.1 Properties of SMC "TrainingTesting"

Table 8: Properties of SMC "TrainingTesting"

idShort:	TrainingTesting		
Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/AIModelNameplate/Plots/TrainingTesting/1/0		
Parent:	SMC Plots		
Explanation:	Collection of plots about training results and test results@en		
[SME type]	**semanticId**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[File] ExamplePlot	[IRI] https://admin-shell.io/idta/AIModelNameplate/Plots/TrainingTesting/Plot/1/0 Training or test result plot	[-]	[1..*]

3.5.2 Properties of SMC "Structure"

Table 9: Properties of SMC "Structure"

idShort:	Structure		
Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/AIModelNameplate/Plots/Structure/1/0		
Parent:	SMC Plots		
Explanation:	Collection describing the model architecture (e.g. of a neural net)@en		
[SME type]	**semanticId**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[File] ExampleStructurePlot	[IRI] https://admin-shell.io/idta/AIModelNameplate/Plots/Structure/StructurePlot/1/0 Image of a model architecture	[-]	[1..*]

3.5.3 Properties of SMC "GeneralPlots"

Table 10: Properties of SMC "General Plots"

idShort:	GeneralPlots
Class:	SubmodelElementCollection
semanticId:	[IRI] https://admin-shell.io/idta/AIModelNameplate/Plots/GeneralPlots/1/0
Parent:	Plots
Explanation:	Collection of plots not fitting in existing collections@en

[SME type]	semanticId		[valueType]	card.
idShort	**Description@en**		**example**	
[File] ExamplePlot	[IRI] https://admin-shell.io/idta/AIModelNameplate/Plots/GeneralPlots/Plot/1/0 Image of an model related plot		[-]	[1..*]

3.6 Properties of SMC "Details"

Table 11: Properties of SMC "Details"

idShort:	Details
Class:	SubmodelElementCollection
semanticId:	[IRI] https://admin-shell.io/idta/AIModelNameplate/Details/1/0
Parent:	SM AIModelNameplate
Explanation:	Collection of additional details about the model@en

[SME type]	semanticId	[valueType]	card.
idShort	**Description@en**	**example**	
[Prop] FileExtension	[IRI] https://admin-shell.io/idta/AIModelNameplate/Details/DataEnding/1/0 File extension of the model (e.g. .pth)	[String] .pth	[1]
[Prop] AIFramework	[IRI] https://admin-shell.io/idta/AIModelNameplate/Details/AIFramework/1/0 AI framework and its version (e.g. pytorch 11.3)	[String] pyTorch 1.13.0+cu116	[1]
[Prop] ProgramLanguage	[IRI] https://admin-shell.io/idta/AIModelNameplate/Details/ProgramLanguage/1/0 Programming language and version (e.g. python 3.9)	[String] Python 3.9.13	[1]
[File] Requirements	[IRI] https://admin-shell.io/idta/AIModelNameplate/Details/Requirements/1/0 Requirements file containing the required software dependencies (e.g. Python Requirements.txt file)	[-]	[0..1]

3.7 Properties of SMC "AIDataset"

Table 12: Properties of SMC "AIDataset"

idShort:	AIDataset		
Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/AIModelNameplate/AIDataset/1/0		
Parent:	SM AIModelNameplate		
Explanation:	Collection of information about the used AIDataset@en		
[SME type]	semanticId	[valueType]	card.
idShort	Description@en	example	
[Ref] AIDatasetReference	[IRI] https://admin-shell.io/idta/AIModelNameplate/AIDataset/AIDatasetReference/1/0 Reference to the used AIDataset-Submodel	[-]	[1]
[Range] TimeStamp	[IRI] https://admin-shell.io/idta/AIModelNameplate/AIDataset/TimeStamp/1/0 Determining the used data, when AIDataset differs over time	[dateTime] ..	[0..1]

3.8 Properties of SMC "AITypeSpecificInformation"

Table 13: Properties of SMC "AITypeSpecificInformation"

idShort:	AITypeSpecificInformation		
Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/AIModelNameplate/AITypeSpecificInformation/1/0		
Parent:	SM AIModelNameplate		
Explanation:	Collection containing different kinds of AI method specific parameters@en.		
[SME type]	semanticId	[valueType]	card.
idShort	Description@en	example	
[SMC] TransferLearning	[IRI] https://admin-shell.io/idta/AIModelNameplate/AITypeSpecificInformation/TransferLearning/1/0 Collection containing information of the origin model, if the model is a result of transfer learning	[-] 4 elements	[0..1]
[SMC] Hyperparameter	[IRI] https://admin-shell.io/idta/AIModelNameplate/AITypeSpecificInformation/Hyperparameter/1/0 Collection containing relevant hyperparamter	[-] 1 elements	[0..1]
[MLP] Type	[IRI] https://admin-shell.io/idta/AIModelNameplate/AITypeSpecificInformation/Type/1/0 Type of the AI(e.g. CNN, RNN, LSTM, ...)	[-] faster RCNN@en	[0..1]

3.8.1 Properties of SMC "TransferLearning"

Table 14: Properties of SMC "TransferLearning"

idShort:	TransferLearning
Class:	SubmodelElementCollection
semanticId:	[IRI] https://admin-shell.io/idta/AIModelNameplate/AITypeSpecificInformation/NeuralNetwork/TransferLearning/1/0
Parent:	SMC AITypeSpecificInformation
Explanation:	Collection containing information of the origin model, if the model is a result of transfer learning@en

[SME type]	semanticId		[valueType]	card.
idShort	**Description@en**		**example**	
[Prop] Name	[IRI] https://admin-shell.io/idta/AIModelNameplate/AITypeSpecificInformation/NeuralNetwork/TransferLearning/Name/1/0 Name of the origin neural network (e.g. resnet18)		[String] torchvision.models.detection.fasterrcnn_resnet50_fpn	[0..1]
[Prop] TimeStamp	[IRI] https://admin-shell.io/idta/AIModelNameplate/AITypeSpecificInformation/NeuralNetwork/TransferLearning/TimeStamp/1/0 Date of access of the origin neural network		[dateTime] 22.03.2023	[0..1]
[File] Weights	[IRI] https://admin-shell.io/idta/AIModelNameplate/AITypeSpecificInformation/NeuralNetwork/TransferLearning/Weights/1/0 File containing the weights of the origin model (e.g. .txt)		[-] default	[0..1]
[Ref] AIModelNameplate	[IRI] https://admin-shell.io/idta/AIModelNameplate/AITypeSpecificInformation/NeuralNetwork/TransferLearning/AIModelNameplateRef/1/0 Reference to the AIModelNameplate Submodel of the origin neural network		[-]	[0..1]

3.8.2 Properties of SMC "Hyperparameter"

Table 15: Properties of SMC "Hyperparameter"

idShort:	Hyperparameter
Class:	SubmodelElementCollection
semanticId:	[IRI] https://admin-shell.io/idta/AIModelNameplate/AITypeSpecificInformation/NeuralNetwork/Hyperparameter/1/0
Parent:	SMC AITypeSpecificInformation
Explanation:	Collection containing relevant hyperparamter@en

[SME type]	semanticId	[valueType]	card.
idShort	**Description@en**	**example**	
[Prop] Example	[IRI] https://admin-shell.io/idta/AIModelNameplate/AITypeSpecificInformation/Hyperparameter/LearningRate/1/0 Example hyperparameter (e.g. learning rate of the model)	[String] 0.005	[0..1]

[SME type]	semanticId	[valueType]	card.
idShort	**Description@en**	**example**	
[Prop] Example	[IRI] https://admin-shell.io/idta/AIModelNameplate/AITypeSpecificInformation/Hyperparameter/LearningRate/1/0 Example hyperparameter (e.g. learning rate of the model)	[String] 0.005	[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
          https://www.acatech.de/publikation/umsetzungsempfehlungen-fuer-das-zukunftsprojekt-
          industrie-4-0-abschlussbericht-des-arbeitskreises-industrie-4-0/

[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/Bitkom/Publikationen/Implementation-Strategy-Industrie-40-
          Report-on-the-results-of-the-Industrie-40-Platform.html

[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]   "Examples of the Asset Administration Shell for Industrie 4.0 Components – Basic Part";
          ZVEI e.V., Whitepaper, April 2017. [Online]. Available:
          ZVEI_WP_Verwaltungschale_Englisch_21.03.17.indd

[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]   "Specification of the Asset Administration Shell Part 1: Metamodel (V3.0.1)", Juni 2024,
          [Online]. Available:Asset Administration Shell Specification - Part 1: Metamodel

[7]   "Semantic IDTAbility: challenges in the digital transformation age"; IEC, International
          Electronical Commission; 2019. [Online]. Available:
          https://www.iec.ch/basecamp/semantic-interoperability-challenges-digital-transformation-
          age

www.industrialdigitaltwin.org