

IDTA 02007-1-0 Nameplate for Software in Manufacturing

August 2023

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
04.08.2023	1.0	Release of the official Submodel template published by IDTA.

Contents

1	General	6
1.1	About this document	6
1.2	Scope of the Submodel	6
1.3	Relevant standards for the Submodel Template	8
2	Submodel for identification of Software in Industrial Applications	9
2.1	Approach.....	9
2.2	Properties of the SMC “SoftwareNameplateType”	10
2.3	Properties of the SMC “SoftwareNameplateInstance”	13
2.4	Properties of the SMC “InstalledModules”	15
2.5	Properties of the SMC “ConfigurationPaths”	16
2.6	Properties of the SMC “ConfigurationPath”	17
2.7	Properties of the SMC “Contact”	17
2.8	Display names for sections and properties with no semanticId available	18
Annex A.	Explanations on used table formats	19
1.	General	19
2.	Tables on Submodels and SubmodelElements.....	19
	Bibliography	20

Figures

Figure 1: Software in manufacturing and the nameplate for software.....	6
Figure 2: Software as an Asset	7
Figure 3: Software as integral part of a composite component (e.g. machine).....	7
Figure 4: UML-Diagram of the Submodel Template “SoftwareNameplate”	9
Figure 5: UML-Diagram of the SMC “SoftwareNameplateType”.....	10
Figure 6: UML-Diagram of the SMC “SoftwareNameplateInstance”	13

Tables

Table 1: Properties of the Submodel Template “SoftwareNameplate”	9
Table 2: Properties of the SMC “SoftwareNameplateType”	10
Table 3: Properties of the SMC “SoftwareNameplateInstance”	13
Table 4: Properties of the SMC “InstalledModules”	15
Table 5: Properties of the SMC “ConfigurationPaths”	16
Table 6: Properties of the SMC “ConfigurationPath”	17
Table 7: Display names for sections and properties with no semanticId available	18

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]. Common terms and abbreviations can be found in [8].

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

Software becomes more and more an essential part in manufacturing value networks, in smart manufacturing, and in smart products. For an effective and efficient use and management of such software, it is necessary to gather relevant information in a uniform representation. Use cases like updates, patch management, license management, audits, etc. rely on information regarding identification, sources and features of software (Figure 1). This information shall be provided in a consistent manner in form of a "nameplate for software", derived and specialized from a common digital nameplate as defined in [9].

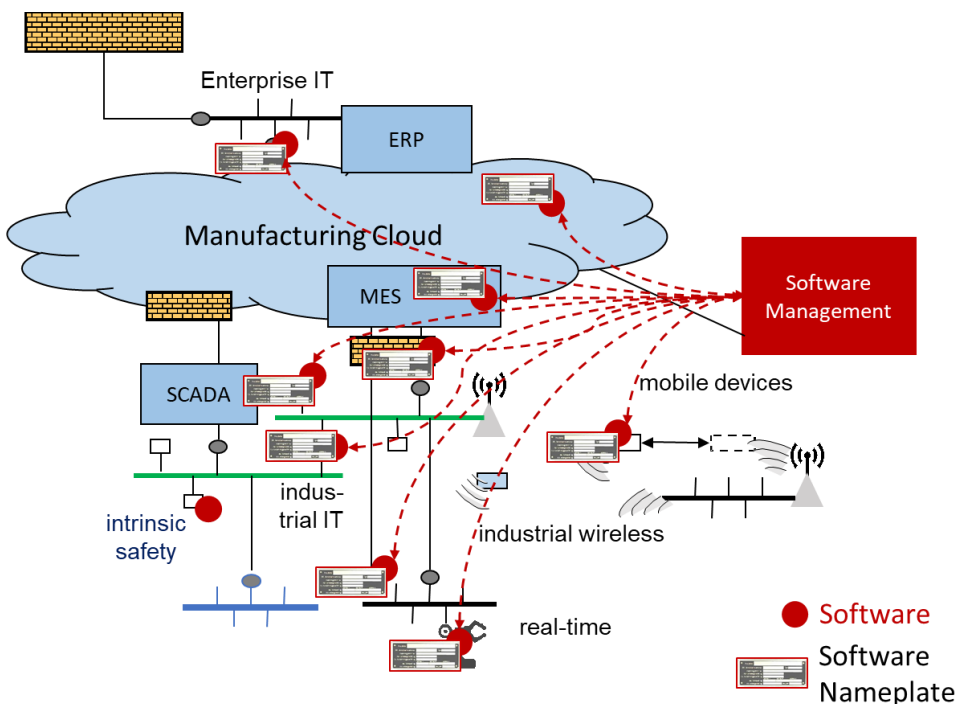


Figure 1: Software in manufacturing and the nameplate for software

The nameplate for software applies to stand-alone software Assets (Figure 2), as well as to software as integral part of a physical Asset, e.g. firmware or application software of a machine, modelled as a composite component using hierarchies according to [11] (Figure 3).

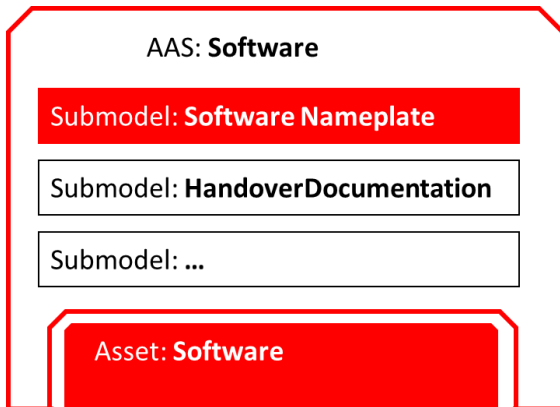


Figure 2: Software as an Asset

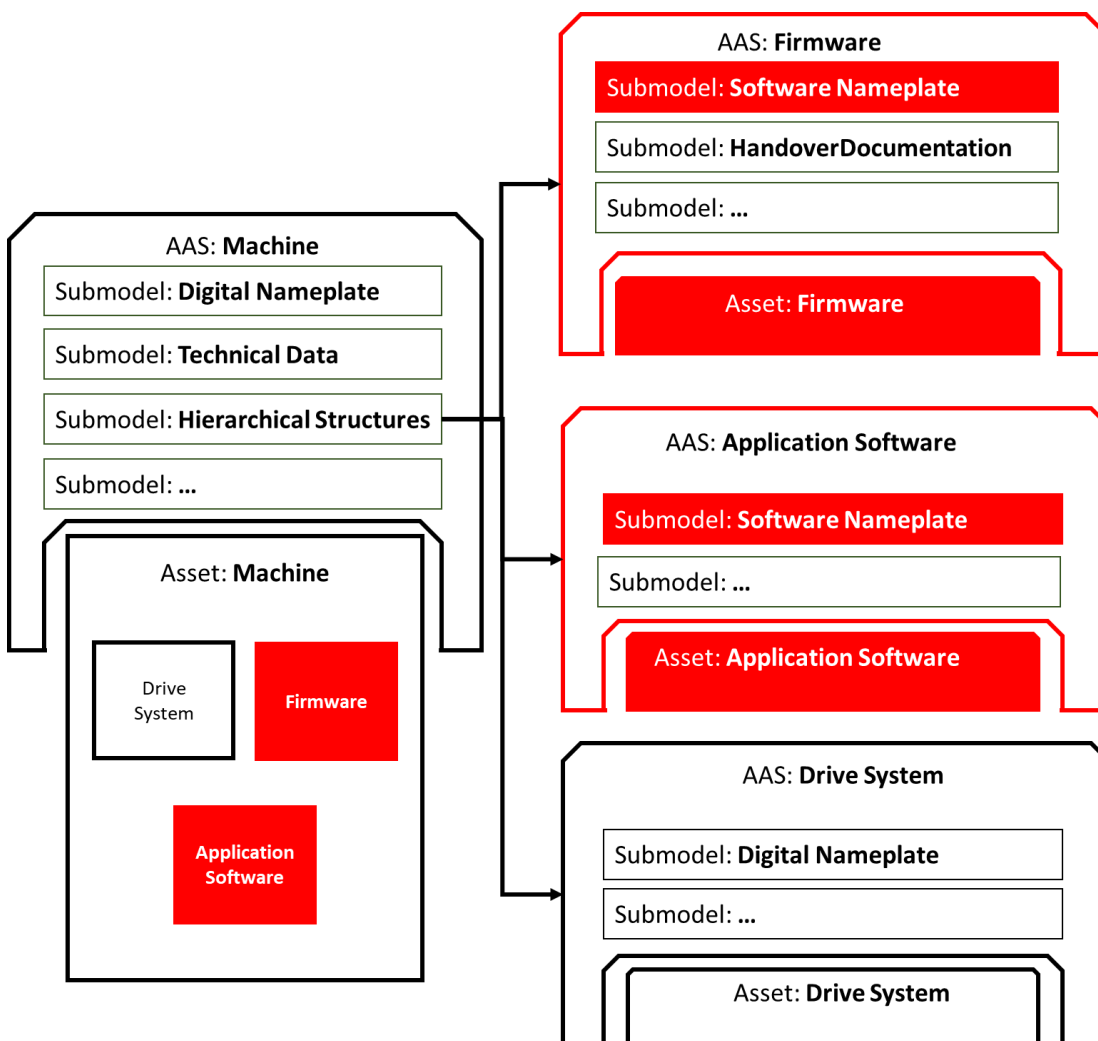


Figure 3: Software as integral part of a composite component (e.g. machine)

This Submodel Template aims at interoperable provision of software nameplate data describing 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 intended use-case is, that a manufacturer of industrial equipment describes Assets (type or instance), which are provided to the market, by the means of software nameplate data (properties), which are interoperable and unambiguously understood by the other market participants, such as system integrators or operators of industrial equipment. For providing individual industrial equipment to the market, also a supplier is covered by the use-case (for this purpose seen as functioning as manufacturer).

This Submodel Template specifies two basic sets of SubmodelElements in order to describe the necessary information according to this use-case.

1.3 Relevant standards for the Submodel Template

According to [3], interoperable properties might be defined by standards, consortium specifications or manufacturer specifications. For the former two cases, so called dictionaries, repositories or classification systems exist in the market. In the further document, these entities are summarized by "property dictionaries" (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>

2 Submodel for identification of Software in Industrial Applications

2.1 Approach

A Submodel according to this Submodel Template specification consists of properties identifying both the type of software and the specific instance. The Submodel re-uses properties of the Submodel “Digital Nameplate” [9]. Thus, the Submodel refers to corresponding concept descriptions. In addition, it uses properties defined in the Submodel “ContactInformation” [10].

The Submodel Template contains two separate SMCs. While SMC “SoftwareNameplateType” contains properties of the type of software (indicated by attribute “assetKind=type”), the SMC “SoftwareNameplateInstance” contains properties of the specific instance of software (“assetKind=instance”). Figure 4 shows the corresponding UML-diagram, Table 1 defines the properties of the Submodel Template.

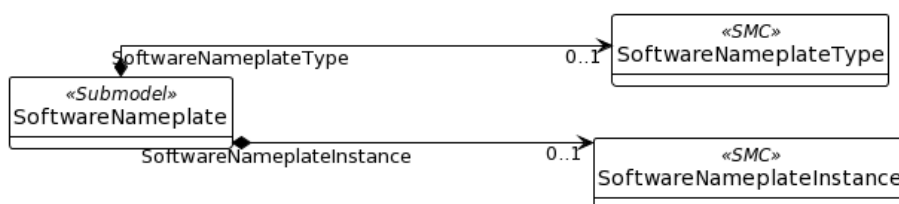


Figure 4: UML-Diagram of the Submodel Template “SoftwareNameplate”

Table 1: Properties of the Submodel Template “SoftwareNameplate”

idShort:	SoftwareNameplate Note: The above idShort shall always be as stated.		
Class:	Submodel		
semanticId:	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0		
Parent:	Asset Administration Shell with Asset, which is software		
Explanation:	Submodel containing the nameplate information for software Asset and associated product classifications.		
[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	Description@en	example	
[SubmodelElementCollection] SoftwareNameplate_Type	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplateType	n/a	[0..1]
	SMC defining type-related properties of a software Asset		
[SubmodelElementCollection] SoftwareNameplate_Instance	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplateInstance	n/a	[0..1]
	SMC defining instance-related properties of a software Asset		

2.2 Properties of the SMC “SoftwareNameplateType”

Figure 5 shows the UML-diagram of the SMC “SoftwareNameplateType”, defining the relevant properties which need to be set to describe a type of software. The properties in red color are re-used from the Submodel “Digital Nameplate” [9].

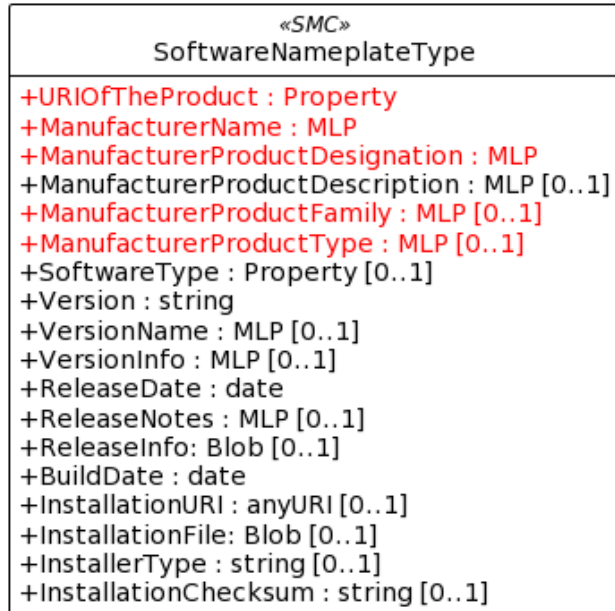


Figure 5: UML-Diagram of the SMC “SoftwareNameplateType”.

Table 2 describes the details of the Submodel structure combined with examples.

Table 2: Properties of the SMC “SoftwareNameplateType”

idShort:	SoftwareNameplate_Type Note: The above idShort shall always be as stated.		
Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplateType		
Parent:	SoftwareNameplate		
Explanation:	SMC containing the nameplate information for a type of a software Asset.		
[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	Description@en	example	
[Property] URIOfTheProduct	[IRDI] 0173-1#02-AAy811#001	string	[1]
	Unique global identification of the product using a universal resource identifier (URI)	ZVEI.I40.ITinAutomation.DemoSW_123456	
[MLP] ManufacturerName	[IRDI] 0173-1#02-AAO677#002	langString	[1]

	Legally valid designation of the natural or judicial person which is directly responsible for the design, production, packaging and labeling of a product in respect to its being brought into circulation	ZVEI AK IT in Automation	
[MLP] ManufacturerProductDesignation	[IRDI] 0173-1#02-AAW338#001	langString	[1]
	The name of the product, provided by the manufacturer	My Software Package for Demonstration	
[MLP] ManufacturerProductDescription	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateType/ManufacturerProductDescription	langString	[0..1]
	Description of the product, it's technical features and implementation if needed (long text)	A first software installation to be used for demo purpose only.	
[MLP] ManufacturerProductFamily	[IRDI] 0173-1#02-AAU731#001	langString	[0..1]
	2nd level of a 3 level manufacturer specific product hierarchy	Demo Products for IT in Automation	
[MLP] ManufacturerProductType	[IRDI] 0173-1#02-AAO057#002	langString	[0..1]
	Characteristic to differentiate between different products of a product family or special variants	DP-AKIT-A	
[Property] SoftwareType	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateType/SoftwareType	string	[0..1]
	The type of the software (category, e.g. Runtime, Application, Firmware, Driver, etc.)	PLC Runtime	
[Property] Version	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateType/Version	string	[1]
	The complete version information consisting of Major Version, Minor Version, Revision and Build Number	0.9.1.0	
[MLP] VersionName	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateType/VersionName	langString	[0..1]
	The name this particular version is given	R2021 beta	
[MLP] VersionInfo	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateType/VersionInfo	langString	[0..1]
	Provides a textual description of most relevant characteristics of the version of the software	Please do not install in productive environments!	

[Property] ReleaseDate	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateType/ReleaseDate	date	[1]
	The moment in time, when this version of the software was made publicly available	20220207	
[MLP] ReleaseNotes	[[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateType/ReleaseNotes	langString	[0..1]
	Contains information about this release	This release requires special configuration.	
[Property] BuildDate	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateType/BuildDate	date	[1]
	The moment in time, when this particular build of software was created	20201119	
[Property] InstallationURI	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateType/InstallationURI	anyURI	[0..1]
	Indicates the resource, where the software is being provided by the manufacturer	https://tud.de/inf/pk/demo-sw/download/DemoFirmware_09.zip	
[Property] InstallationFile	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateType/InstallationFile	Blob [0..1]	[0..1]
	Contains the installation code as BLOB.		
[Property] InstallerType	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateType/InstallerType	string	[0..1]
	Indicates the type of installation package	MSI	
[Property] InstallationChecksum	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateType/InstallationChecksum	string	[0..1]
	Provides the checksum for the software available at InstallationURI	0x2783	

2.3 Properties of the SMC “SoftwareNameplateInstance”

Figure 6 shows the UML-diagram of the SMC “SoftwareNameplateInstance”, defining the relevant properties which need to be set to describe a specific instance of software. Table 3 shows the relevant properties which need to be set. The properties in red color are re-used from the Submodel “Digital Nameplate” [9], the SMC “Contact” (in green color) is defined in the Submodel “ContactInformation” [10].

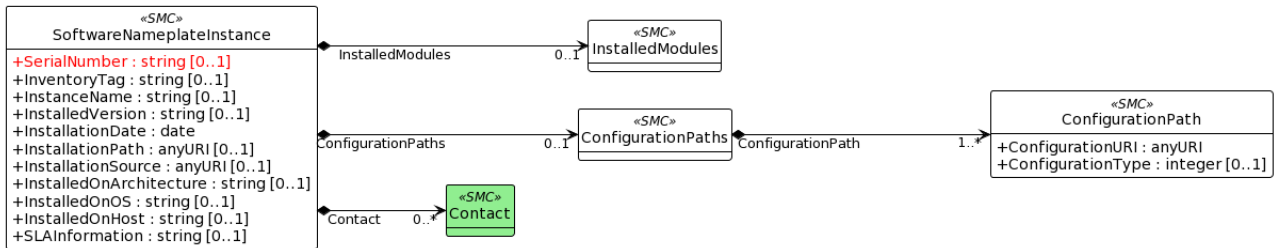


Figure 6: UML-Diagram of the SMC “SoftwareNameplateInstance”.

Table 3: Properties of the SMC “SoftwareNameplateInstance”

idShort:	SoftwareNameplate_Instance Note: The above idShort shall always be as stated.		
Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplateInstance		
Parent:	Asset Administration Shell with Asset, which is software		
Explanation:	SMC containing the nameplate information for an instance of a software Asset.		
[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	Description@en	Example	
[Property] SerialNumber	[IRDI] 0173-1#02-AAM556#002 ([IRDI] 0112/2///61987#ABA951#007 serial number)	String	[0..1]
	Unique combination of numbers and letters used to identify the software instance	123456	
[Property] InstanceName	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/InstanceName	String	[0..1]
	The name of the software instance	My Software Instance	
[Property] InstalledVersion	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/InstalledVersion	String	[0..1]
	The version information of the installed instance, consisting of Major Version, Minor Version, Revision and Build Number indicates the actual version of the instance	0.9.1.0	
[Property] InstallationDate	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/InstallationDate	Date	[0..1]
	Date of Installation	20201119T09:30:20	

[Property] InstallationPath	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/InstallationPath	anyURI	[0..1]
	Indicates the path to the installed instance of the software	C:\Windows\Program Files\Demo\Firmware	
[Property] InstallationSource	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/InstallationSource	anyURI	[0..1]
	Indicates the path to the installation files used in this instance of the software	https://tud.de/inf/pk/installation/firmware/src	
[Property] InstalledOnArchitecture	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/InstalledOnArchitecture	String	[0..1]
	Indicates the processor architecture this instance is installed on	x86-32	
[Property] InstalledOnOS	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/InstalledOnOS	String	[0..1]
	Indicates the operating system this instance is installed on	Windows 10	
[Property] InstalledOnHost	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/InstalledOnHost	String	[0..1]
	Indicates the host system in case of a virtual environment	IPC_42	
[SubmodelElementCollection] InstalledModules	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/InstalledModules	n/a	[0..1]
	Collection of installed modules		
[SubmodelElementCollection] ConfigurationPaths	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/ConfigurationPaths	n/a	[0..1]
	Indicates the path to the configuration information		
[Property] SLAInformation	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/SLAInformation	string	[0..1]
	Indicates the actual service level agreements	Service level GOLD USER.	
[SubmodelElementCollection] Contact	[IRI] https://admin-shell.io/zvei/nameplate/1/0/ContactInformations/ContactInformation	n/a	[0..1]
	Collection for general contact data		

[Property] InventoryTag	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/InventoryTag	string	[0..*]
	Specifies an information used for inventory of the software	TU3-88D5	

2.4 Properties of the SMC “InstalledModules”

Figure 6 shows the UML-diagram defining the relevant properties which need to be set. Table 4 describes the details of the SMC structure combined with examples.

Table 4: Properties of the SMC “InstalledModules”

idShort:	InstalledModules		
Class:	SubmodelElementCollection		
semanticId:	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/InstalledModules		
Parent:	Submodel element collection “SoftwareNameplateInstance”		
Explanation:	Contains a list of installed modules of the software instance.		
[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	Description@en	example	
[Property] InstalledModule	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/InstalledModule	string	[1..*]
	The name of a particular module installed	main	

2.5 Properties of the SMC “ConfigurationPaths”

Figure 6 shows the UML-diagram defining the relevant properties which need to be set. Table 5 describes the details of the SMC structure combined with examples.

Table 5: Properties of the SMC “ConfigurationPaths”

idShort:	ConfigurationPaths		
Class:	SubmodelElementCollection		
semanticId:	[[IRI]] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/ConfigurationPaths		
Parent:	Submodel element collection “SoftwareNameplateInstance”		
Explanation:	Contains a list of configuration entries of the software instance.		
[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	Description@en	example	
[SubmodelElementCollection]	[[IRI]]https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/ConfigurationPath	n/a	[1..*]
	Contains a single configuration entry		

2.6 Properties of the SMC “ConfigurationPath”

Figure 6 shows the UML-diagram defining the relevant properties which need to be set. Table 6 describes the details of the SMC structure combined with examples.

Table 6: Properties of the SMC “ConfigurationPath”

idShort:	ConfigurationPath		
Class:	SubmodelElementCollection		
semanticId:	[[IRI]] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/ConfigurationPath		
Parent:	Submodel element collection “ConfigurationPaths”		
Explanation:	Contains a single configuration entry of the software instance.		
[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	Description@en	example	
[Property] ConfigurationURI	[[IRI]]https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/ConfigurationURI	anyURI	[1]
	Indicates the path to the configuration	C:\Users\mw30\Documents\ZVEI\AKITinAutomation\20210113	
[Property] ConfigurationType	[[IRI]]https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateInstance/ConfigurationType	string	[0..1]
	Indicates the type of configuration (e.g. general configuration, user configuration)	initial configuration	

2.7 Properties of the SMC “Contact”

This SMC contains information on the contact information of the person responsible for the software instance. It uses the SMC structure defined in the Submodel “ContactInformation” [10].

The property “RoleOfContactPerson” shall use the IRDI 0173-1#07-AAS931#001 (technical contact).

2.8 Display names for sections and properties with no semanticId available

For displaying property names in a user interface, the following precedence of display names shall be maintained:

Table 7: Display names for sections and properties with no semanticId available

Priority	Concept in AAS metamodel	Description
1 (highest)	SubmodelElement/ description	<p>If available, an adequate language will be chosen from the LangStringSet of description. Thus, the user explanation of the property by the SubmodelElement will overrule the definition of the ConceptDescription by the semanticId.</p> <p>Note: In a future version of the AAS metamodel, a distinctive "DisplayName" will be available and will substitute the use of description.</p> <p>Note: If a user or application requests a preferred language, then this language shall be used; default is English (en).</p>
2	ConceptDescription/ preferredName	<p>If available, an adequate language will be chosen from preferred name.</p> <p>Note: The user interface is recommended to be capable of handling presentation of at least 50 characters.</p> <p>Note: If a user or application requests a preferred language, then this language shall be used; default is English (en).</p>
3	ConceptDescription/ shortName	<p>As above, but shorter definition.</p> <p>Note: If a user or application requests a preferred language, then this language shall be used; default is English (en).</p>
4 (lowest)	SubmodelElement/ idShort	<p>If only available, the idShort shall be presented.</p> <p>Note: This is usually not language adequate and the least user orientation presentation.</p>

Note: Consequently, for editors of instances of this Submodel Template, the following procedure is recommended to be applied:

- (i) Whenever possible, include a ConceptDescription within the AAS with speaking elaboration of preferredName, shortName and definition.
- (ii) If no ConceptDescription could be provided (no semanticId, that is, no formal conceptualization of the property could be given to the user), a speaking explanation shall be provided by SubmodelElement/description.
- (iii) in any case, a short but speaking, unique idShort for the SubmodelElement shall be chosen, respecting the allowed characters (regex definition: [a-zA-Z0-9_-]+)

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 [] from 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 chosen, 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/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 Commision; 2019; Available: <https://basecamp.iec.ch/download/iec-white-paper-semantic-interoperability-challenges-in-the-digital-transformation-age-en/?>
- [8] Common terms and abbreviations according to VDI FA 7.21 Wiki; Available: <http://i40.iosb.fraunhofer.de/>
- [9] “Submodel Template of the Asset Administration Shell - Digital Nameplate for Industrial Equipment (Version 2.0)”, IDTA 02006-2-0, 2022, [Online]. Available: <https://github.com/admin-shell-io/submodel-templates/tree/main/published>
- [10] “Submodel Template of the Asset Administration Shell - Submodel for Contact Information (Version 1.0)”, IDTA-02002-1-0, 2022, [Online]. Available: <https://github.com/admin-shell-io/submodel-templates/tree/main/published>
- [11] “Submodel Template of the Asset Administration Shell - Hierarchical Structures enabling Bills of Material (Version 1.0)”, IDTA-02011-1-0, April 2023, [Online]. Available: <https://github.com/admin-shell-io/submodel-templates/tree/main/published>

www.industrialdigitaltwin.org