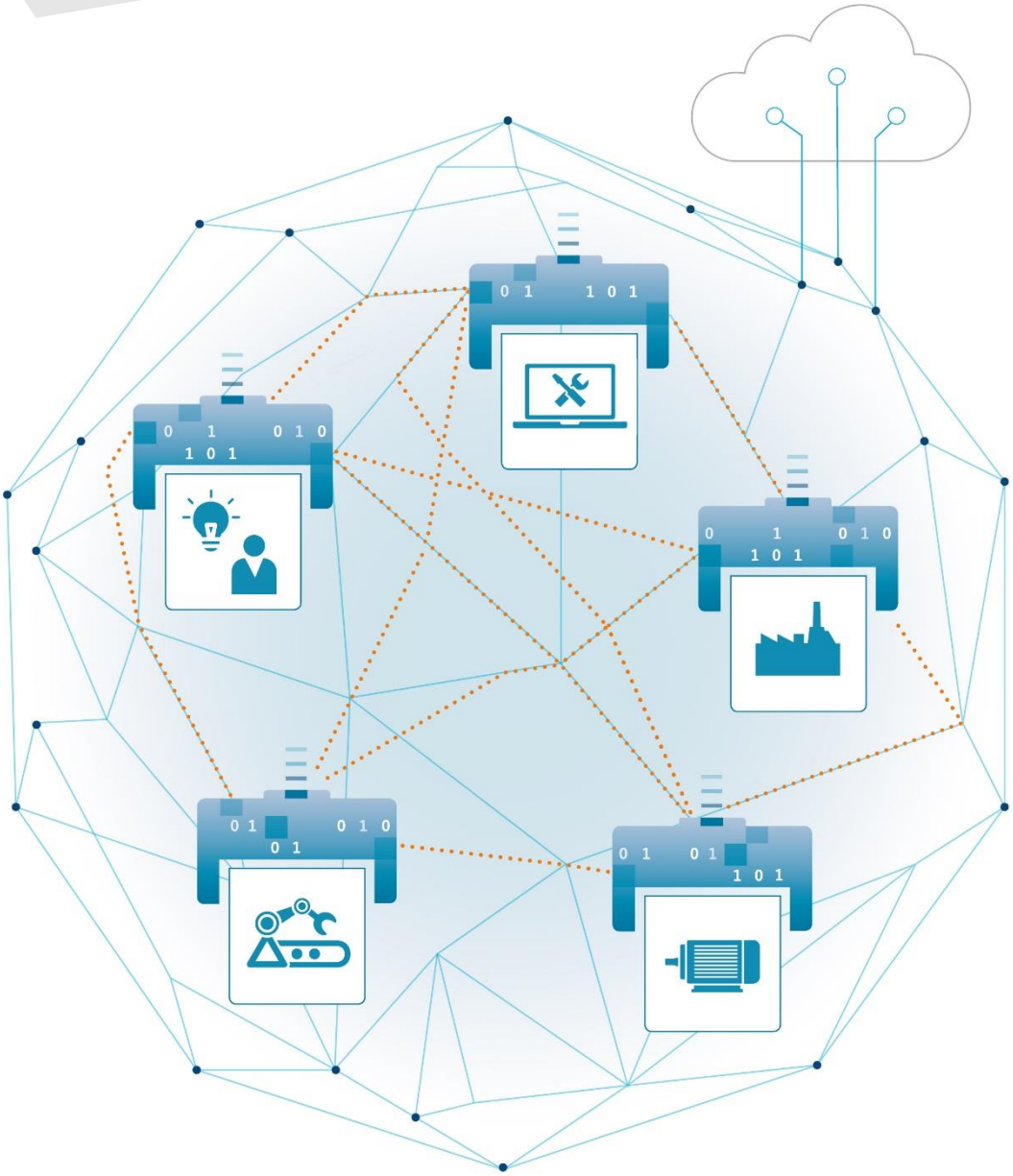
PLATTFORM INDUSTRIE 4.0 in cooperation with ZVEI: Die Elektroindustrie

SPECIFICATION

Details of the Asset Administration Shell



Part 1 - The exchange of information between partners in the value chain of Industrie 4.0 (Version 2.0)

.

Imprint

Publisher

Federal Ministry for Economic Affairs

and Energy (BMWi)

Public Relations

10119 Berlin

www.bmwi.de

Text and editing

Plattform Industrie 4.0

Bertolt-Brecht-Platz 3

10117 Berlin

Design and production

The Plattform Industrie 4.0 secretariat, Berlin

Status

November 2019

Illustrations

Plattform Industrie 4.0; Anna Salari, designed by freepik (Title)

Contents

Table of Tables

Table of Figures

1 Preamble

1.1 Editorial Notes

This document was produced August 2018 to November 2019 by the sub working group “Asset Administration Shell” of the Platform Industrie 4.0 Working Group “Reference Architectures, Standards and Norms “.

The first version of this document was produced Sep 2017 to July 2018 by a joint working group with members from ZVEI SG “Models and Standards” and Platform Industrie 4.0 Working Group “Reference Architectures, Standards and Norms “. The document was subsequently validated by the platform’s Working Group ““Reference Architectures, Standards and Norms “.

For better readability, in compound terms the abbreviation "I4.0" is consistently used for "Industrie 4.0". Used on its own "Industrie 4.0" continues to be used.

1.2 Scope of this Document

The aim of this document is to make selected specifications of the structure of the Administration Shell in such a way that information about assets and I4.0 Components can be exchanged in a meaningful way between partners in a value creation network.

This part of this document therefore focuses on the question of how such information needs to be processed and structured. In order to make these specifications, the document formally stipulates a few structural principles of the Administration Shell. This part does not describe technical interfaces of the Administration Shell or other systems to exchange information, protocols or interaction patterns.

This document focuses on:

- Exchange format for the transport of information from one partner in the value chain to the next
- Metamodel for specifying information of an Asset Administration Shell and its submodels
- Identifiers
- Access Control
- Mappings to suitable technologies to be used in different life cycle phases of a product

This document currently features the second version V2.0. It targets to be adequately complete and coherent to be used as basis for developments and as input for discussion with international standardization organisations and further cooperations.

The definitions in and the form of the document should be such that development departments in the value creation networks have enough detailed information to start work on internal systems for exchanging information and on corresponding databases.

1.3 Structure of the Document

Clause 3 summarises relevant, existing content from the standardization of Industrie 4.0. In other words, this clause provides an overview and explains the motives, and is not absolutely necessary for an understanding of the subsequent definitions.

Clause 4 stipulates sufficient structural principles of the Administration Shell in a formal manner in order to ensure an exchange of information between the Administration Shells. An excerpt of a UML diagram is drafted for this purpose. A more comprehensive UML discussion which does not set standards can be found in the annex.

Clause 5 provides detailed definitions for the exchange of information compliant to this specification in existing data formats like XML, AutomationML, OPC UA information models, JSON or RDF. An explanation is provided for each of these data formats stating how information is to be represented (metamodel), and an example of a representation is provided.

Clause 6 describes the promotion of attribute based access models for information security.

Clause 7 describes, how the information of one or more Administration Shell could be packed into a compound file format.

The Annex contains details w.r.t. the mappings done in Clause 5 plus additional background information on asset administration shell.

1.4 Principles of the Work

The work is based on the following principle: keep it simple but do not simplify if it affects interoperability.

For creating a detailed specification of the Administration Shell according to the scope of part 1 (→ 1.2), result papers published by Plattform Industrie 4.0, the Trilateral cooperation with France and Italy and international standardization results were analysed and takes as source of requirements for the specification process. As many ideas as possible from the discussion papers were considered.

The partners represented in the Plattform Industrie 4.0 and associations such as the ZVEI, the VDMA, VDI/ VDE and Bitkom, ensure that there is broad sectoral coverage, both in process, hybrid and factory automation and in terms of integrating information technology (IT) and operational technology (OT).

Design alternatives were intensively discussed within the working group. An extensive feedback process with the so called "sounding board" of this document series, with the Plattform's working groups and with associated partners were engaged about the design alternatives and the final content of the specification.

Guiding principle for the specification was to provide detailed information, which can be easily implemented also by small and medium-sized enterprises.

2 Terms, Definitions and Abbreviations

2.1 Terms & Definitions

Forward notice:

Definition of terms are only valid in a certain context. The current glossary applies to the context of this document.

access control

protection of system resources against unauthorized access; a process by which use of system resources is regulated according to a security policy and is permitted by only authorized entities (users, programs, processes, or other systems) according to that policy

→ [SOURCE: IEC TS 62443-1-1]

application

software functional element specific to the solution of a problem in industrial-process measurement and control

Note: An application can be distributed among resources and may communicate with other applications.

→ [SOURCE: IEC TR 62390:2005-01, 3.1.2]

asset

physical or logical object owned by or under the custodial duties of an organization, having either a perceived or actual value to the organization

Note: In the case of industrial automation and control systems, the physical assets that have the largest directly measurable value can be the equipment under control.

→ [SOURCE: IEC TS 62443-1-1:2009, 3.2.6]

asset administration shell (AAS)

standardized *digital representation* of the *asset*, corner stone of the interoperability between the applications managing the manufacturing systems. It identifies the Administration Shell and the assets represented by it, holds digital models of various aspects (*submodels*) and describes *technical functionality* exposed by the Administration Shell or respective assets.

Note: Asset Administration Shell and Administration Shell are used synonymously.

→ [SOURCE: Glossary Industrie 4.0]

attribute

→ data element of *a property*, a relation, or a *class* in information technology
→ [SOURCE: ISO/IEC Guide 77-2, SO/IEC 27460, IEC 61360]

class

description of a set of objects that share the same *attributes*, *operations*, methods, relationships, and semantics

→ [SOURCE: IEC TR 62390:2005-01, 3.1.4]

capability

implementation-independent potential of an Industrie 4.0 component to achieve an effect within a domain

Note 1: Capabilities can be orchestrated and hierarchically structured.
Note 2: Capabilities can be made executable via services.
Note 3: The impact manifests in a measurable effect within the physical world

→ [SOURCE: Glossary Industrie 4.0]

component

product used as a constituent in an assembled product, *system* or plant

→ [SOURCE: IEC 61666:2010, 3.6]

concept

unit of knowledge created by a unique combination of characteristics

→ [SOURCE: IEC 61360-1, ISO 22274:2013, 3.7]

digital representation

information that represents characteristics and behaviors of an entity

Note 1: Information is data that within a certain context has a particular meaning. Data is content represented in a digital and formalized manner suitable for communication, storage, interpretation or processing
Note 2: Behavior includes functionality (description and execution)

→ SOURCE: [IIC Vocabulary IIC:IIVOC:V2.2:20190903, notes added]

digital twin

digital representation, sufficient to meet the requirements of a set of use cases

Note: in this context, the entity in the definition of digital representation is typically an asset

→ [SOURCE: IIC Vocabulary IIC:IIVOC:V2.2:20190903, adapted]

identifier (ID)

identity information that unambiguously distinguishes one entity from another one in a given domain

Note: There are specific identifiers, e.g. UUID Universal unique identifier, IEC 15418 (GS1).

→ [SOURCE: Glossary Industrie 4.0]

instance

concrete, clearly identifiable component of a certain *type*

Note 1: It becomes an individual entity of a type, for example a device, by defining specific property values.
Note 2: In an object-oriented view, an instance denotes an object of a class (of a type).

→ [SOURCE: IEC 62890:2016, 3.1.16 65/617/CDV]

operation

executable realization of a function

Note 1: The term method is synonym to operation
Note 2: an operation has a name and a list of parameters [ISO 19119:2005, 4.1.3]

→ [SOURCE: Glossary Industrie 4.0]

ontology

an explicit specification of a (shared) conceptualization

→ [SOURCE: Gruber “A Translation Approach to portable ontology specifications”, Knowledge acquisition 5.2 (1993): 199-220]

property

defined characteristic suitable for the description and differentiation of products or components

Note 1: The concept of type and instance applies to properties.
Note 2: This definition applies to properties such as described in IEC 61360/ ISO 13584-42
Note 3: The property types are defined in dictionaries (like IEC component Data dictionary or eCl@ss), they do not have a value. The property type is also called data element type in some standards.
Note 4: The property instances have a value and they provided by the manufacturers. A property instance is also called property-value pair in certain standards.
Note 5: Properties include nominal value, actual value, runtime variables, measurement values, etc.
Note 6: A property describes one characteristic of a given object.
Note 7: A property can have attributes such as code, version, and revision.
Note 8: The specification of a property can include predefined choices of values.

→ [SOURCE:according ISO/IEC Guide 77-2] as well as [SOURCE:according Glossary Industrie 4.0]

qualifier

well-defined element associated with a *property* instance or *submodel element*, restricting the value statement to a certain period of time or use case

Note: qualifier can have value associated

→ [SOURCE: according to IEC 62569-1]

variable

software *entity* that may take different values, one at a time

→ [SOURCE: IEC 61499-1]

smart manufacturing

manufacturing approach, that improves its performance aspects with integrated and intelligent use of processes and resources in cyber, physical and human spheres to create and deliver products and services, which also collaborates with other domains within an enterprise's' value chains.

Note 1: Performance aspects include agility, efficiency, safety, security, sustainability or any other performance indicators identified by the enterprise.
Note 2: In addition to manufacturing, other enterprise domains can include engineering, logistics, marketing, procurement, sales or any other domains identified by the enterprise.
Note 3: this definition is, as of November 2019, under discussion within the ISO/ IEC joint working group (JWG) 21. However, it gives a good indication and a citable source.

→ [SOURCE: ISO/TMB/SMCC]

submodel

models which are technically separated from each other and which are included in the *asset administration shell*

Note 1: Each submodel refers to a well-defined domain or subject matter. Submodels can become standardized and thus become submodels templates.
Note 2: Submodels can have different life cycles.
Note 3: The concept of template and instance applies to submodels.

→ [SOURCE: Glossary Industrie 4.0]

submodel element

element suitable for the description and differentiation of *assets*

Note 1: extends the definition of *properties*
Note 2: could comprise *operations*, binary objects

→ [SOURCE: Glossary Industrie 4.0]

system

interacting, interrelated, or interdependent elements forming a complex whole

→ [SOURCE: IEC TS 62443-1-1:2009, 3.2.123]

technical functionality

functionality of the *Administration Shell* that is exposed by an application programming interface (API) and that is creating added value to the respective *assets(s).*

Note: can consist of single elements, which are also known as functions, operations, methods, skills.

→ [SOURCE: according [18]]

template

specification of the common features of an object in sufficient detail that such object can be instantiated using it

Note: object can be anything that has a type

→ [SOURCE: according ISO/IEC 10746-2]

type

hardware or software element which specifies the common *attributes* shared by all instances of the type

→ [SOURCE: IEC TR 62390:2005-01, 3.1.25]

view

projection of a model or models, which is seen from a given perspective or vantage point and omits *entities* that are not relevant to this perspective

→ [SOURCE: unified modelling language - UML]

2.2 Abbreviations

Abbreviation	Description
AAS	Asset Administration Shell
AASX	Package file format extension for the AAS
AML	AutomationML
API	Application programmers interface
BITKOM	Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V.
BLOB	Binary Large Object
CDD	Common Data Dictionary
GUID	Globally unique identifier
I4.0	Industrie 4.0
ID	Identifier
IEC	International Electrotechnical Commission
IRDI	International Registration Data Identifier
IRI	Internationalized Resource Identifier
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
MIME	Multipurpose Internet Mail Extensions
OPC	Open Packaging Conventions (ECMA-376, ISO/IEC 29500-2)
OPC UA	Unified Architecture for the Object Linking and Embedding for Process Control
PDF	Portable Document Format
RAMI4.0	Reference Architecture Model Industrie 4.0
RDF	Resource Description Framework
REST	Representational State Transfer
RFC	Request for Comment
SOA	Service Oriented Architecture
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
VDE	Verein Deutscher Ingenieure
VDE	Verband der Elektrotechnik Elektronik Informationstechnik e. V.
VDMA	Verband Deutscher Maschinen- und Anlagenbau e.V.
W3C	World Wide Web Consortium
XML	eXtensible Markup Language
ZIP	archive file format that supports lossless data compression
ZVEI	Zentralverband Elektrotechnik- und Elektronikindustrie e. V.

3 Basic Concepts and Leading Picture

3.1 Basic Concepts

Many concepts for Industrie 4.0 and smart manufacturing are already existing. The most important ones are summarised in the informative Annex A.

3.2 Leading Picture

The leading use case in this document is the exchange of an Asset Administration Shell including all its auxiliary documents and artifacts from one value chain partner to another. This is, in this document we do not deal with the use case of already deployed Asset Administration Shells running in a specific infrastructure but only with file exchange between partners.

Figure 1 Use Case File Exchange between Value Chain Partners

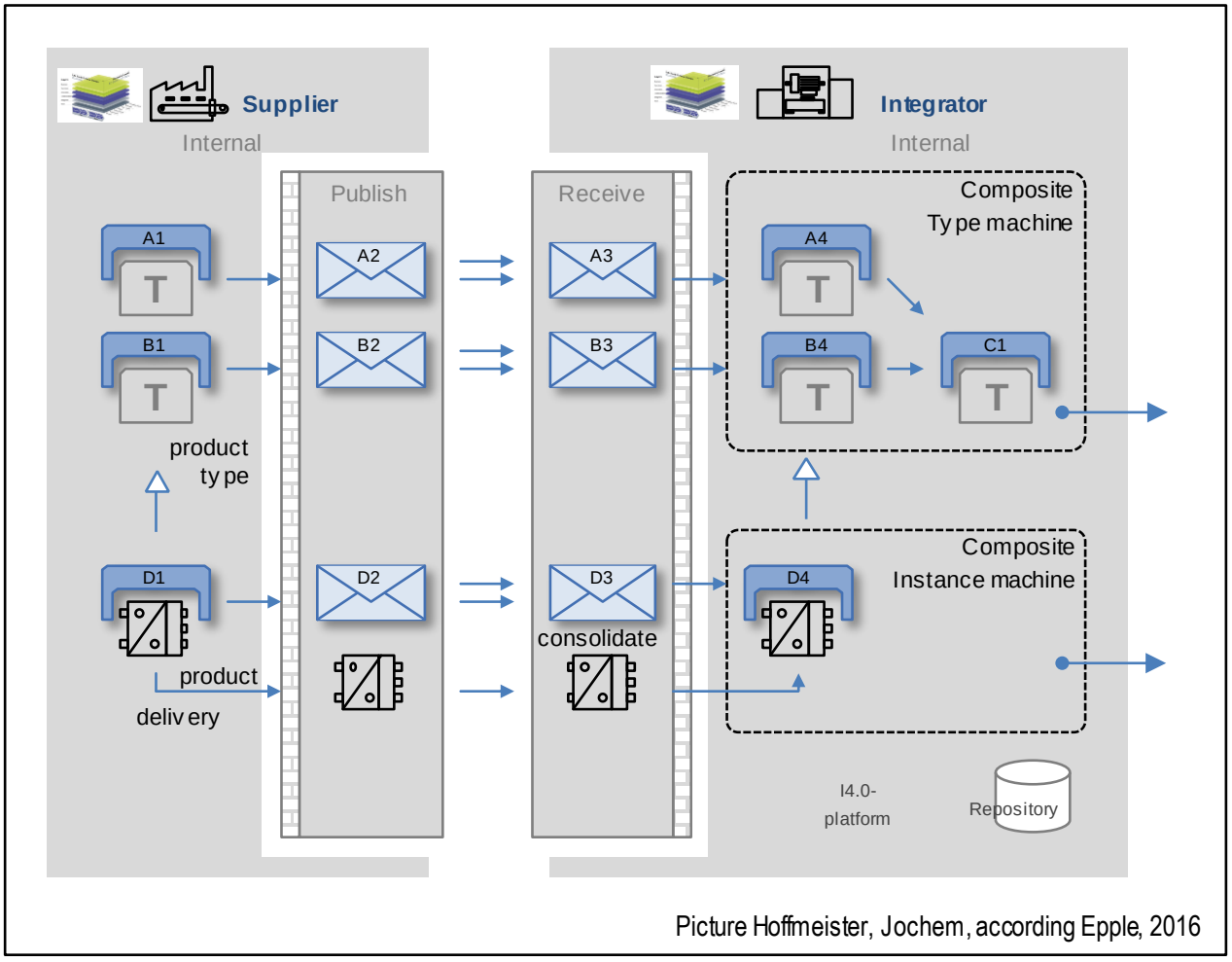
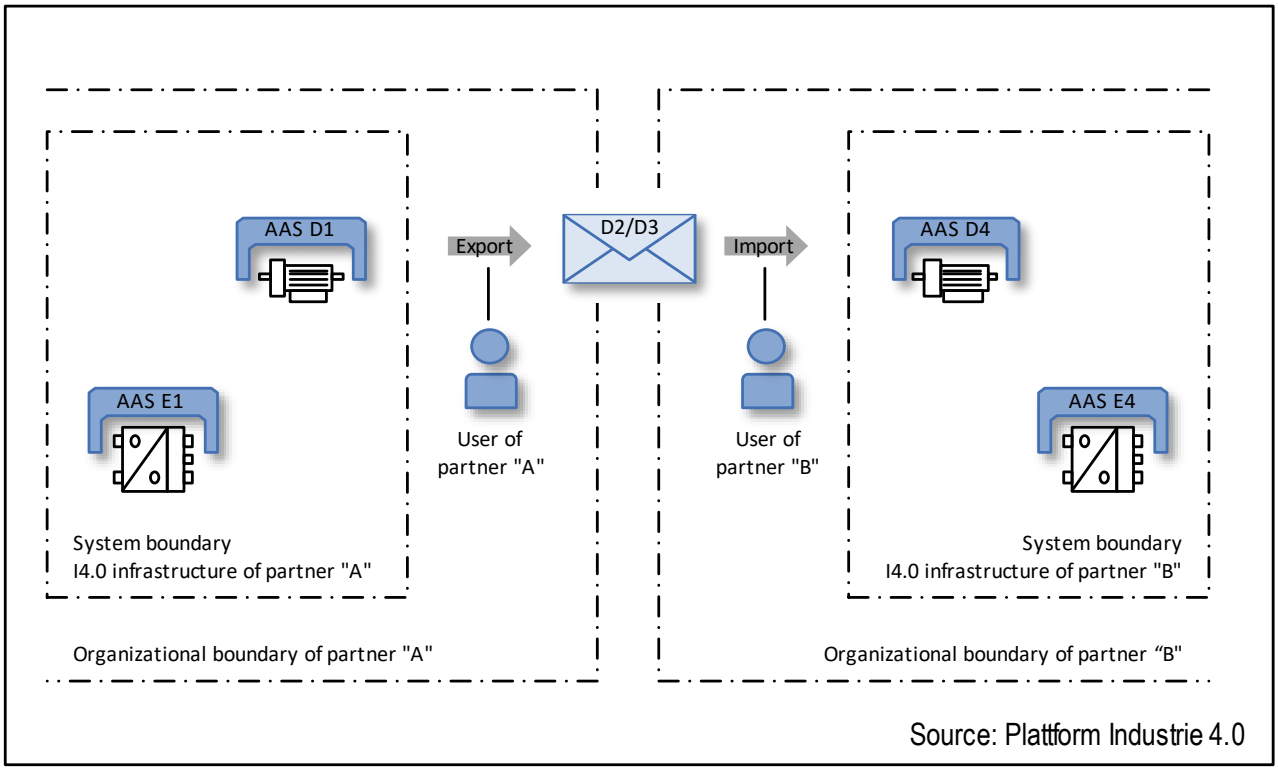


Figure 1 shows the overall picture. It depicts two value chain partners; "Supplier" is going to provide some products, "Integrator" is going to utilize this products in order to build a machine. Two kinds of Administration Shells are being provided; one for the asset being the type of a product, one for the assets being the actual product instances. "Supplier" and "Integrator" are forming two independent legal bodies (Figure 2).

Figure 2 File Exchange between two value chain partners



The exchange of files needs to fulfil some requirements with respect to usability and security. There needs to be a bilateral agreement on security constraints to be fulfilled for the transfer and usage of the files. This is explained in more detail in Clause 6.

For usability a container format for exchanging files is used and a corresponding structure is defined (see Clause 7). This predefined structure helps the consumer to understand the content of the single files. This is important because an AssetAdministration Shell specification can be spread across several files. Additionally, the container may contain auxiliary files references by the AAS or even executable code.

4 The Metamodel of the Administration Shell

4.1 Introduction

This clause specifies the information metamodel of the AssetAdministration Shell. Before doing so some general aspect of the handling of asset types and instances are described (see Subclause 4.2 Types and Instances). Another very important aspect of the AAS is the identification aspect, see Subclause 4.3. In Subclause 4.5 aspects of event handling are discussed. In Subclause 4.3 it is described which elements to use for modelling composite I4.0 Components.

An overview of the metamodel of the Asset Administration Shell is given in Subclause 4.6. In Subclause 4.7 the classes are described in detail together with all their attributes.

The metamodel for security aspects of the Administration Shell is described in Clause 6.

The legend for understanding the UML diagrams and the table specification of the classes are found in Annex B and Annex C.

4.2 Types and Instances

4.2.1 Life Cycle with Asset Types and Instances

Industrie 4.0 utilizes an extended understanding of asset, comprising elements such as factories, production systems, equipment, machines, components, produced products and raw materials, business processes and orders, immaterial assets (such as processes, software, documents, plans, intellectual property, standards), services and human personnel and more.

The RAMI4.0 model [3] features one, generalized life-cycle axis, which was derived from IEC 62890. The basic idea is to distinguish for all assets within Industrie 4.0 between possible types and instance. This makes it possible to apply the type/instance distinction for all elements such as material type/material instance, product type/product instance, machine type/ machine instance and more. Business related information will be handled on the 'Business' layer of the RAMI4.0 model, as well, covering also order details and workflows, again with types/ instances.

Table 1 Life cycle phases and roles of type and instance

Phase		Description
Type	Development	Valid from the ideation/ conceptualization to first prototypes/ test. The 'type' of an asset is defined, and distinguishing properties and functionalities are defined and implemented. All (internal) design artefacts are created, such as CAD data, schematics, embedded software, and associated with the asset type.
	Usage / Maintenance	Ramping up production capacity. The 'external' information associated to the asset is created, such as technical data sheets, marketing information. The selling process starts.
Instance	Production	Asset instances are created/ produced, based on the asset type information. Specific information about production, logistics, qualification and test are associated with the asset instances.
	Usage / Maintenance	Usage phase by the purchaser of the asset instances. Usage data is associated with the asset instance and might be shared with other value chain partners, such as the manufacturer of the asset instance. Also included: maintenance, re-design, optimization and de-commissioning of the asset instance. The full life-cycle history is associated with the asset and might be archived/ shared for documentation.

Table 1 gives an overview of the different life cycles phases and the role of type and instance in these phases: The most important relationship is between asset types and asset instance. This relationship should be maintained throughout the life of the asset instances. By this relationship, updates to the asset types can be forwarded to the asset instances, either automatically or on demand.

Note: for the distinction of 'type' and 'instance', the term 'kind' is used in this document.

The second class of relationships are feedback loops/ information within the life-cycle of the asset type and instance. For product assets, for example, information on usage and maintenance of product instances can improve the manufacturing of products as also cause design improvements for the (next) product type.

The third class of relationships are feedforward/ information exchange with assets of other asset classes. For example, sourcing information from business assets can influence design aspects of products; or, the design of the products affects the design of the manufacturing line.

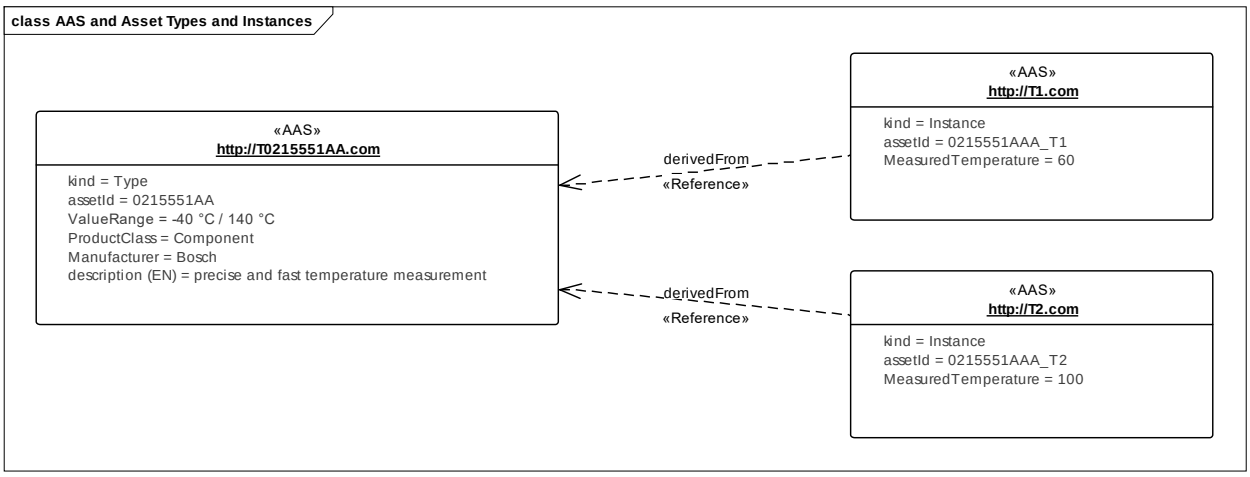
Note: For an illustration of the second/ third class of relationships confer the NIST model, as well.

A forth class of relationships are between asset of different hierarchy levels. For example, these could be the (dynamic) relationships between manufacturing stations and products being currently produced. These could be also the decompositions of production systems in physical, functional or safety hierarchies. By this class of relationships, automation equipment is explained as a complex, interrelated graph of automation devices and products, performing intelligent production and self-learning/ optimization tasks.

4.2.2 Example

The following figure gives an example for handling of asset types and asset instances, handling some exemplary information as well. Further explanation will follow in the next clauses.

Figure 3 Exemplary types and instances of assets represented by multiple AAS



Note: The example is simplified for ease of understanding and does only roughly comply to the metamodel as it is specified in Clause 4. The id handling is simplified as well: the names of the classes correspond to the unique global identifier of the AASs.

Note: In the context of Platform Industrie 4.0 types and instances typically refer to ”asset types” and “asset instances”. When referring to types or instances of an AAS this is explicitly denoted as “AAS types” and “AAS instances” to not mix up both. AAS types are synonymously used with the term “AAS template”.

Note: Please refer to Clause 2 for the IEC definition of types and instances. For the scope of this document, there is no full equivalency between these definitions and the type/ instance concepts of object oriented programming (OO).

There shall be a concrete asset type of a temperature sensor and two uniquely identifiable physical temperature sensors of this type. The intention is to provide a separate AAS for the asset type as well as for every single asset instance.

In the example the first sensor has the unique ID “0215551AA_T1” and the second sensor has the unique ID “0215551AA_T2”. The AAS for the first sensor has the unique URI “http://admin-shell.io/T1” and the AAS for the second sensor has the unique URI “http://admin-shell.io/T2”. The kind for both is “Instance”. The example shows that the measured temperature at operation time of the two sensors is different: for T1 it is 60 °C, for T2 it is 100 °C. For the

time-being we ignore the relationship “derivedFrom” of the two AAS “T1” and “T2” with AAS “http://admin-shell.io/T0215551AA”.

Note: Even though the HTTP scheme is used in the example, the URIs do not need to be valid URLs and therefore do not need to point to accessible content.

Note: The unit can be obtained by the semantic reference of the element “measuredTemperature”. For simplicity this is not shown in the example.

These two asset instances do have a lot of information they share: the information of the asset type (in this example a sensor type). For this asset type an own AAS is created. The unique ID for this AAS is “http://admin-shell.io/T0215551AA”, the unique id of the sensor type is “0215551AA”. The kind in this case is “Type” and not “Instance”. The information that is the same for all instances of this temperature sensor type is the ProductClass (=”Component”), the manufacturer (=”Bosch”) and the English Description “=’precise and fast temperature measurement’” as well as the value range “-40 °C / 140 °C”.

Now the two AAS of the two asset instances may refer to the AAS of the asset type “0215551AA” using the relationship attribute “derivedFrom”.

Note: "attribute" refers in the UML sense to the property or characteristic of a class (instance).
Note: Typically, if a specific asset type does exist, it exists in time before the respective asset instances.
Note: An AAS is used synonym to an AAS instance. An AAS may be realized based on an AAS type. AAS types are out of scope of this document.
Note: In public standardization the AAS Types might be standardized. However, it is much more important to standardize the property types (called property definitions or concept descriptions) or other submodel element typed as well as complete submodel types because those can be reused in different AAS.
Note: In the domain of internet of things (IoT), asset instances are typically denoted as “Things” whereas asset types are denoted as “Product”.

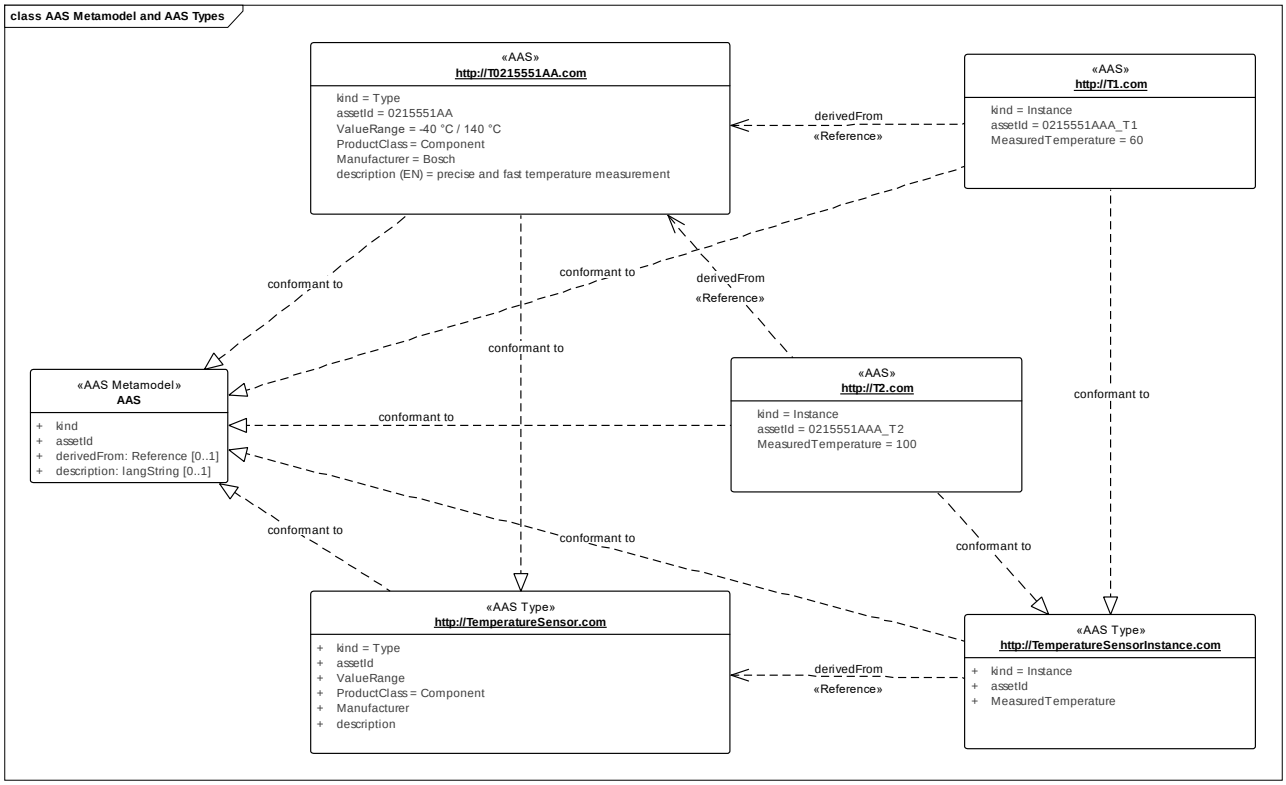
4.2.3 Asset Administration Shell Types and Instances

In the previous clause type and instances of assets were explained. Obviously the question then comes up how to harmonize AAS as well as AAS types. In our example it can be seen that the attributes “assetId” and “kind” as well as the global identifier (id, represented as name of the class) are present for all AAS. However, if there is no standard, it is not clear that the semantics of “id”, “assetId” and “kind” are the same for all AAS and it is not clear, which of the attributes are mandatory and which are specific for the asset (type or instance). This is illustrated in Figure 4.

This is the task of this document: The definition of a metamodel that defines which attributes are mandatory and which are optional for all AAS. The Platform Industrie 4.0 metamodel for AssetAdministration Shells is defined in Clause 4.

Note: This approach ensures that requirement tAAS-#19 is fulfilled. Another approach could have been to define two metamodels: one for asset types and one for asset instances. However, the large set of similarities motivated to go with one metamodel.
Note: The metamodel itself does not prescribe mandatory submodels. This is another step of standardization similar to the prescription of submodels of AAS Type level.
Note: An AAS type shall be realized based on the metamodel of an AAS as defined in this document. This Metamodel is referred to as the “AAS Metamodel”.
Note: It is not mandatory to define an AAS type before defining an AAS (instance). An AAS instance that does not realize an AAS type shall be realized based on the Metamodel of an AAS as defined in this document.

Figure 4 Exemplary relations between metamodel of AAS, AAS types and AAS instances



4.3 Composite I4.0 Components

As described in Clause 4.2.1 there is a class of relationships between assets of different hierarchy levels. By this class of relationships, automation equipment is explained as a complex, interrelated graph of automation devices and products, performing intelligent production and self-learning/ optimization tasks.

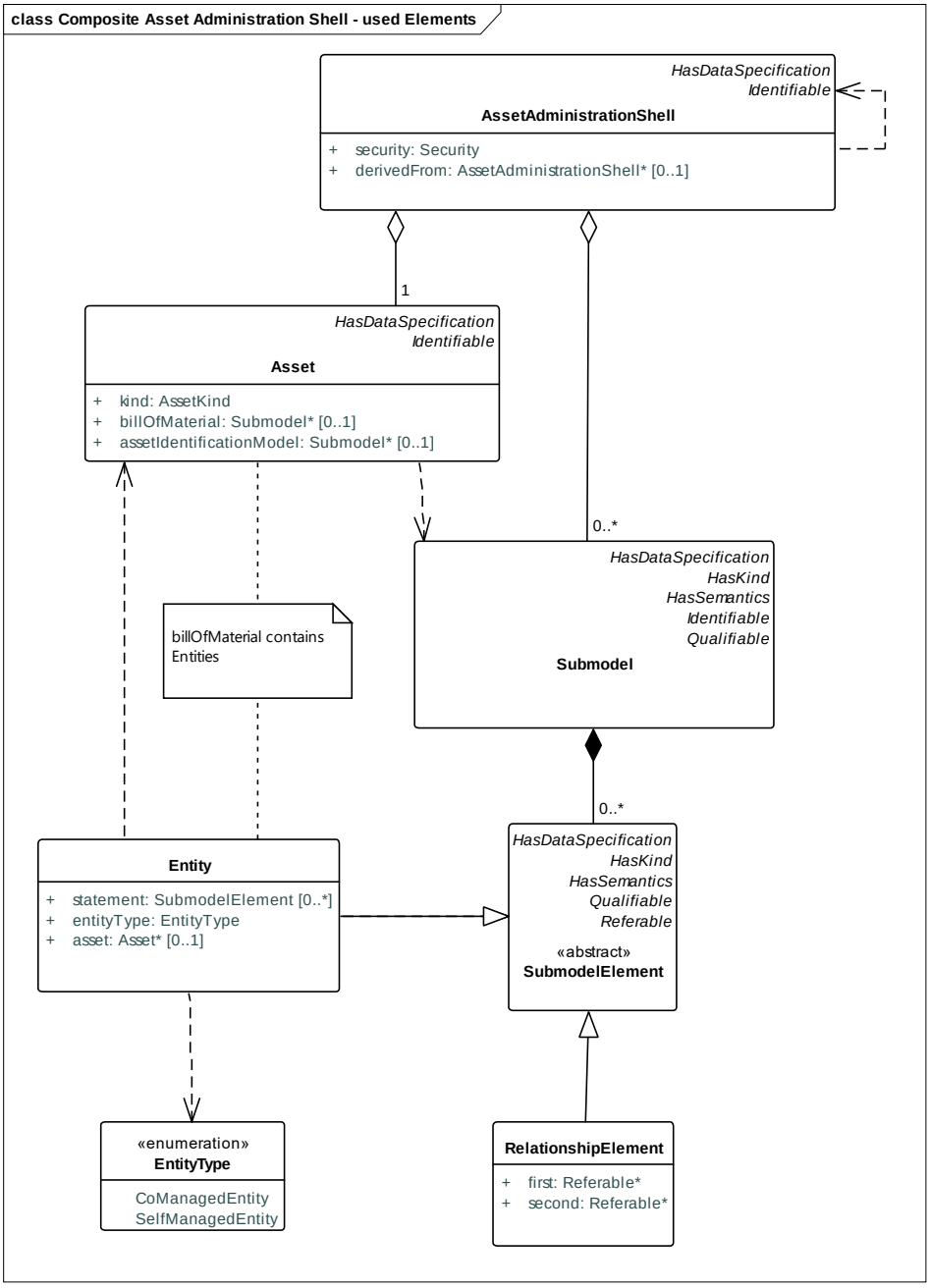
Details and examples for composite I4.0 Components can be found in [12].

The following modelling elements in the AAS metamodel can be used to realize such composite I4.0 Components:

- *RelationshipElement* – used to describe relationships between assets and other elements
- *Asset/billOfMaterial* – A complex asset is composed out of other entities and assets. These entities and assets being part of the asset are specified in the bill of material.
  Note: The submodel template defining the structure of such a bill of material is not predefined by the AAS metamodel but is assumed to contain Entity elements.
- Not every entity (*Entity*) that is part of the bill of material of an asset has necessarily its own asset administration shell. As described in [12] self-managed entities are distinguished from co-managed entities (*Entity/entityType*).
  - Self-Managed Entities have their own AAS. This is why a reference to this asset is specified as well (*Entity/asset*). Additionally, further property statements (compare to [15]) can be added to the asset that are not specified in the AAS of the asset itself because they are specified in relation to the complex I4.0 Component only.
  - For co-managed entities there is no separate AAS. The relationships and property statements of such entities are managed within the AAS of the composite I4.0 Component.

Figure 5 shows an extract of the metamodel that is introduced later containing the elements being the moste important to describe composite I4.0 Components.

Figure 5 Extract from Metamodel for Composite I4.0 Components



4.4 Identification of Elements

4.4.1 Overview

Identifiers are needed according to [4] for the unique identification of many different elements within the domain of smart manufacturing. For this reason, they are a fundamental element of a formal description of the Administration Shell. Especially, identification is at least required for:

- Asset Administration Shells,
- assets,
- submodel instances and submodel templates,
- property definitions/concept descriptions in external repositories, such as eCl@ss or IEC CDD

Identification will take place for two purposes:

(1) to uniquely distinguish all elements of an Administration Shell, and

(2) to relate elements to external definitions, such as submodel templates and property definitions, in order to bind a semantics to these data and functional elements of an Administration Shell.

4.4.2 What Identifiers Exist?

In [4], [20] two standard-conforming global identification types are defined:

(a) **IRDI** - ISO29002-5, ISO IEC 6523 and ISO IEC 11179-6 [20] as an Identifier scheme for properties and classifications. They are created in a process of consortium-wise specification or international standardization. To this end, users sit down together and feed their ideas into the consortia or standardization bodies. Properties in ISO, IEC help to safeguard key commercial interests. Repositories like eCl@ss and others make it possible to standardise a relatively large number of Identifiers in an appropriately short time.

(b) **IRI** – IRI (Rfc 3987[1]) or URI and URL according to RFC 3986[2] as identification of assets, Administration Shells and other (probably not standardized, but globally unique) properties and classifications.

The following is also permitted:

(c) **Custom** - Internal custom Identifiers such as GUIDs (globally unique Identifiers[3]), which a manufacturer can use for all sorts of in-house purposes within the Administration Shell.

This means that the URIs/URLs and internal custom Identifiers can represent and communicate manufacturer-specific information and functions in the Administration Shell and the 4.0 infrastructure just as well as standardized information and functions. One infrastructure can serve both purposes.

CLSID are URIs for GUIDs. They start with a customer specific schema. So Custom should really only be used if the custeomer specific identifier is no IRDI nor an IRI.

Besides the global Identifiers there are also Identifiers that are unique only within a defined namespace, typically its parent element. These Identifiers are also called local identifiers. Example: Properties within a submodel have local identifiers.

Besides absolute URIs there are also relative URIs.

See also DIN SPEC 91406 [43] for further information on identification of physical objects.

4.4.3 Identifiers for Assets and Administration Shells

For the domain of smart manufacturing, the assets need to be identified worldwide unique [4] [20] by the means of identifiers (IDs). The Administration Shell has a unique ID, as well.

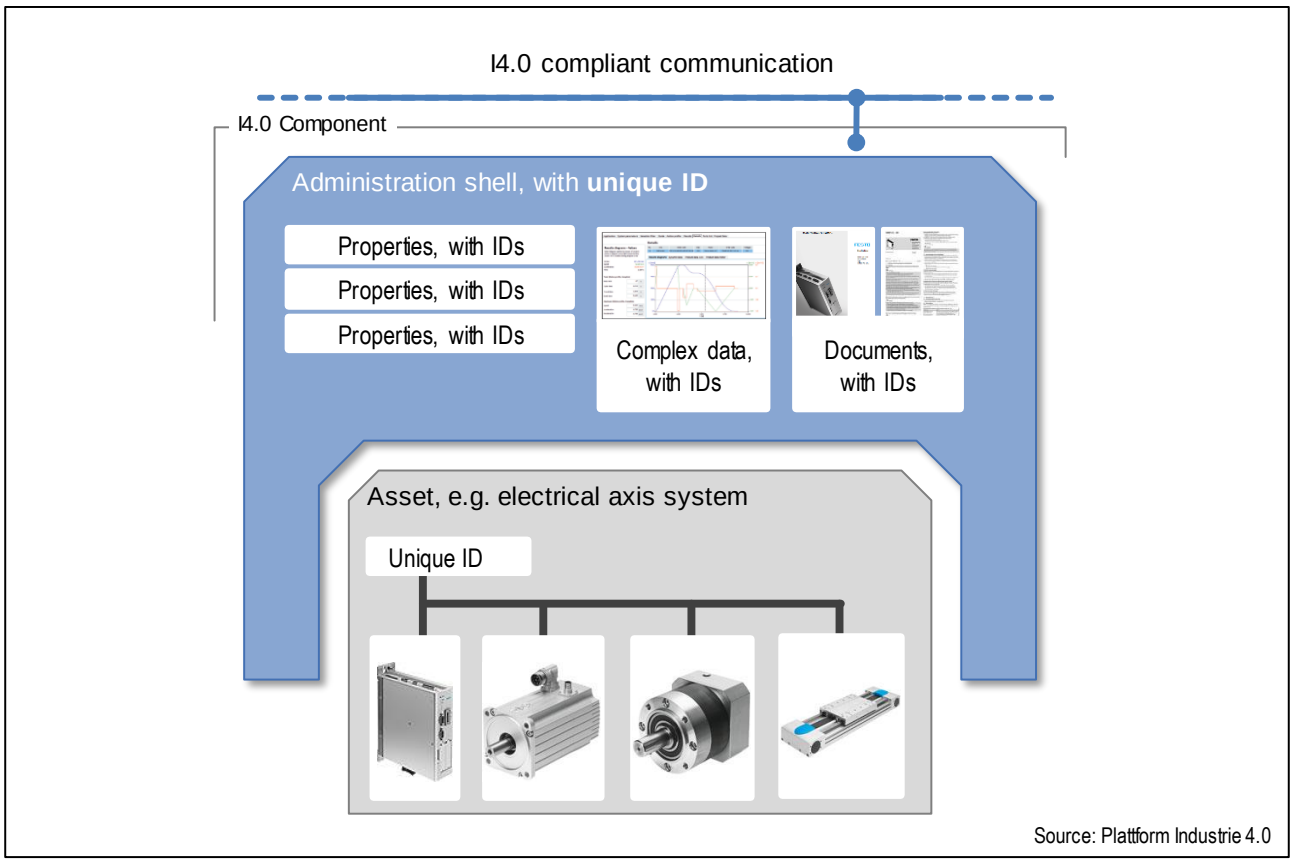
---

[1] https://tools.ietf.org/html/rfc3987

[2] https://tools.ietf.org/html/rfc3986

[3] https://en.wikipedia.org/wiki/Universally_unique_identifier

Figure 6 The Administration Shell needs a unique Identifier, as well as the asset being described (Modified figure from [4])



An Administration Shell represents exactly one asset, with a unique asset ID. In a batch based production, the batches will become the asset and will be described by a respective Administration Shell. If a set of assets shall be described by an Administration Shell, a unique ID for the composite asset needs to be created [12].

The ID of the asset needs to comply the restrictions for global Identifiers according [4][20]. If the asset is featuring further identifications, serial numbers and such, there are not to be confused with the unique global Identifiers of the asset itself[4].

4.4.4 Which Identifiers to use for which Elements

Not every Identifier is applicable for every element of the UML model; the following table therefore puts constraints on the various entities, which implement "Identifiable" or "hasSemantics". Attributes relate to the metamodel in Clause 4.6 and Clause 4.7.

Table 2 Identifiables, attributes and allowed identifiers

Identifiable	Attribute	Allowed Identifiers (recommended or typical)	Remarks
Asset AdministrationShell	id	IRI (URL)	mandatory Typically, URLs will be used
	idShort	string	mandatory[5]
Asset	id	IRI	mandatory

[4] Such additional local identifiers are contained in the submodel "*assetIdentificationModel*".

[5] Note: In version V1.0 of this specification idShort was optional for Identifiables. This changed in V2.0: now idShort is mandatory for all Referables.

Identifiable	Attribute	Allowed Identifiers (recommended or typical)	Remarks
			Typically, URLs will be used [4]
	idShort	string	mandatory
Submodel with kind = Template	id	IRDI, IRI (URI)	mandatory IRDI, if the defined submodel is standardized and an IRDI was applied for it
	idShort	string	mandatory Typically used as idShort for the submodel of kind Instance as well
	semanticId	IRDI, IRI (URI)	optional The semantic id might refer to an external information source, which explains the formulation of the submodel (for example an PDF if a standard)
Submodel with kind = Instance	id	IRI (URI), Custom	mandatory
	idShort	string	mandatory Typically, the idShort or English short name of the submodel template referenced via semanticId
	semanticId	IRDI, IRI (URI)	recommended The semantic id may be either a reference to a submodel with kind = Template (within the same or another AAS) or it can be an external reference to an external standard defining the semantics of the submodel.
SubmodelElement	idShort	string	mandatory Typically the English short name of the element referenced via semanticId
	semanticId	IRDI, IRI (URI), Custom	mandatory link to the *conceptDescription* or the concept definition in an external repository via a global id
ConceptDescription	id	Custom or IRDI	mandatory *ConceptDescription* needs to have a global id. If the concept description is a copy from an external dictionary like eCl@ss it may use the same global id as it is used in the external dictionary.
	idShort	string	mandatory e.g. same as English short name
	isCaseOf	IRDI, IRI (URI)	optional links to the concept definition in an external repository the concept description is a copy from or that it corresponds to
View	idShort	string	mandatory

Identifiable	Attribute	Allowed Identifiers (recommended or typical)	Remarks
	semanticId	IRDI, IRI (URI), Custom	recommended links to the view definition in an external repository
Qualifier	semanticId	IRDI, IRI (URI), Custom	recommended Links to the qualifier type definition in an external repository IRDI, if the defined qualifier type is standardized and an IRDI was applied for it

4.4.5 How are New Identifiers Created?

Following the different identification types from Clause 4.4.3, it can be stated:

- **(a)** IRDIs are assumed to be already existing by an external specification and standardization process, when it comes to the creation of a certain Administration Shell. For bringing such IRDI Identifiers into life, refer to Clause 4 of the document [4].
- **(b)** URIs and URLs can easily be formed by developers themselves, also on the fly when creating a certain Administration Shell. All that is needed is a valid authority, for example of the company, and to make sure that the way the domain (e.g. admin-shell.io) is organised ensures that the path behind the host name is reserved in a semantically unique way for these Identifiers. In this way, each developer can create an arbitrary URI or URL by combining the host name and some chosen path, which only needs to be unique in the developer's organisation.
- **(c)** Custom identifiers can also be easily formed by developers themselves. All that is necessary is a corresponding programmatic functionality to be retrieved. It is necessary to ensure that internal custom identifiers can be clearly distinguished from (a) or (b).
- **(d)** Local identifiers can also be created on the fly. They have to be unique within their namespace, usually defined by the *parent* relationship.

4.4.6 Best Practice for Creating URI Identifiers

The approach for semantics and interaction for I4.0 components [17] suggests the use of the following structure for URIs[6], which is slightly modified here. The idea is to always structure URIs following a scheme of different elements. However, this is just a recommendation and not mandatory to be used.

Table 3 Proposed structure for URIs

Element	Description	Syntax component
Organisation	Legal body, administrative unit or company issuing the ID	A
Organisational subunit/ Document ID/ Document subunit	Sub entity in organisation above, or released specification or publication of organisation above.	P
Submodel / Domain-ID	Submodel of functional or knowledge-wise domain of asset or Administration Shell, the Identifier belongs to.	P
Version	Version number in line with release of specification or publication of Identifier	P
Revision	Revision number in line with release of specification or publication of Identifier	P

[6] URLs are also URIs

Property / Element-ID	Property or further structural element ID of the Administration Shell	P
Instance number	Individual numbering of the instances within release of specification or publication	P

In the table, syntax component "A" refers to authority of RFC 3986 (URI) and namespace identifier of RFC 2141 (URN); "P" refers to path of RFC 3986 (URI) and namespace specific string of RFC 2141 (URN).

<AAS URI> ::= <scheme> ":" <authority> [<path>]

<scheme> ::= a valid URI scheme

<authority> ::= <Organisation>

<path> ::= <subunit> <domain> <release> <element>

<subunit> ::= [("/" | ":") <Organisational Subunit/Document ID/Document subunit>]*

<domain> ::= [("/" | ":") <Submodel / Domain-ID>

<release> ::= [("/" | ":") <Version> [("/" | ":") <Revision>]*]

<element> ::= [("/" | ":" | "#") (<Property/Element-ID> | <Instance number>)*]

Using this scheme, valid URNs and URLs can be created, both being URIs. For the use of Administration Shells, URLs are preferred, as functionality (such as REST services) can be bound to the Identifiers, as well. Examples of such Identifiers are given in Table 4.

Table 4 Example URN and URL-based Identifiers of the Administration Shell

Identifier	Description	Property class	Examples
Administration Shell ID	ID of the Administration Shell	Basis	urn:zvei:SG2:aas:1:1:demo11232322 http://www.zvei.de/SG2/aas/1/1/demo11232322
Submodel ID (Type)	Identification of type of submodel	Selected submodels are basis, others free	urn:GMA:7.20:contractnegotiation:1:1 http://www.vdi.de/gma720/contractnegotiation/1/1
Submodel ID (Instance)	Identification of the instance of the submodel	Free	urn:GMA:7.20:contractnegotiation:1:1#001 http://www.vdi.de/gma720/ contractnegotiation/1/1#001
Property/parameter/status type IDs	Identification of the property, parameter and status types	Domain-specific	urn:PROFIBUS:PROFIBUS-PA:V3-02:Parameter:1:1:MaxTemp http://www.zvei.de/SG2/aas/1/1/demo11232322/maxtemp
Property/parameter/status instance IDs **(not used by metamodel)**	Identification of the property, parameter and status instance	Domain-specific	urn:PROFIBUS:PROFIBUS-PA:V3-02:Parameter:1:1: MaxTemp#0002 http://www.zvei.de/SG2/aas/1/1/demo11232322/maxtemp#0002

Note: the last row of the table is only used for completion; the metamodel does not foresee identifiers for property/parameter/status instances.

4.4.7 Creating a Submodel Instance based on an Existing Submodel Template

In order to instantiate an existing submodel template, there should be a public specification of the submodel template, e.g. via publication by Plattform Industrie 4.0. As a special case, instantiating a submodel from a non-public submodel template, such as a manufacturer specification, is also possible.

As of November 2019, there are no finally published standardized submodel templates available, but some examples are described in [6], which provides simple tables listing properties in a predefined hierarchy.

In each submodel template, the **Identifiers of property definitions to be used as semantic references are already predefined.** An instantiation of such submodel merely has to create properties with a semantic reference to the property definitions and attach values to these properties.

In such case, the Identifier for the existing submodel template is also predefined, probably as a URL, and is to be used as semantic reference for the submodel instance.

What remains is to create an Identifier of the submodel instance itself, which is in the regular case an URI/URL.

Note: for maintaining integrity over multiple Administration Shells, appropriate referencing (*derivedFrom*) between submodel instances and submodel templates has to occur, as well as for submodel instances of interlinked asset types and instances. A possible framework could then monitor and synchronize changes to the value statements of the submodel instances according to user requirements (automatic synchronization is not always desired).

4.4.8 Can New or Proprietary Submodels be Formed?

It is in the interest of Industrie 4.0 for as many submodels as possible, including free and proprietary submodels, to be formed (→ [4], “Free property sets”). A submodel can be formed at any time for a specific Administration Shell of an asset. For this purpose, the provider of the Administration Shell can form in-house Identifiers for the type and instance of the submodel in line with Section 4.4.5. All I4.0 systems are called on to ignore submodels and properties that are not individually known, and simply to “overlook” them. For this reason, it is always possible to deposit proprietary – e.g. manufacturer-specific or user-specific – information, submodels or properties in an Administration Shell.

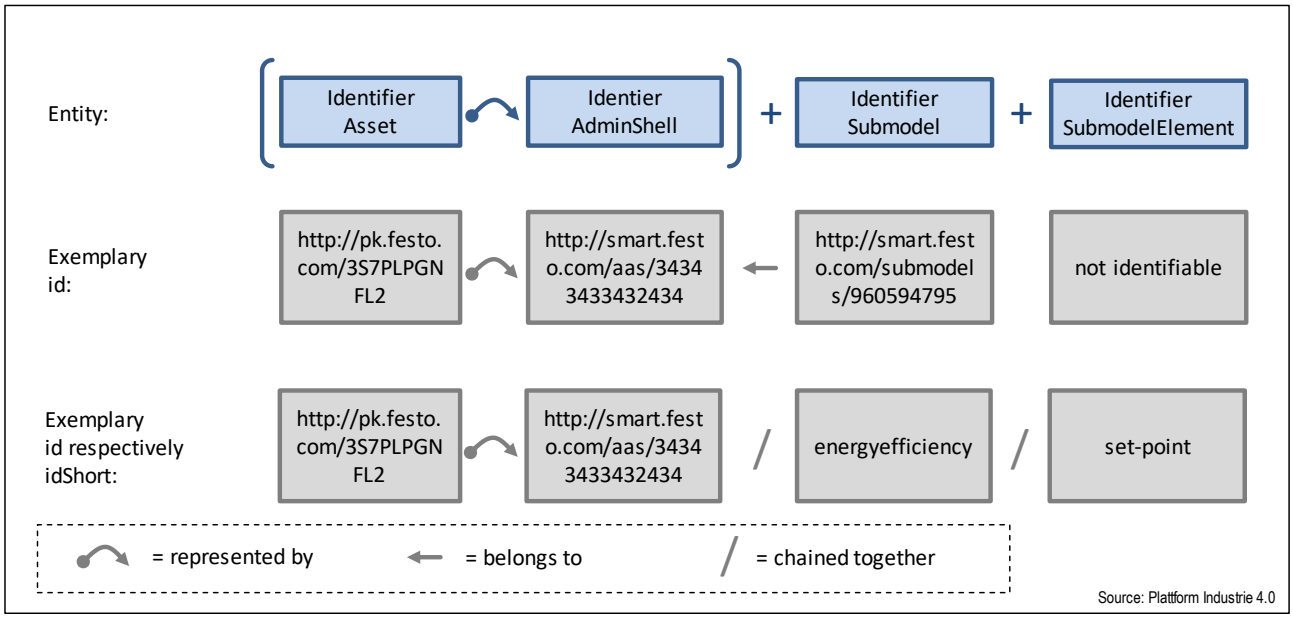
Note: it is in the intention of the Administration Shell, that proprietary information is included as well. For example to link to company-wide identification schemes or information required for company-wide data processing. By this, a single infrastructure can be used to transport standardized and proprietary information at the same time; this conveys the introduction (and later standardization) of new information elements as well.

Note: if a submodel instance is formed without a clear relation to a submodel template or semantic definition, this will be of limited use for other users/ accessing systems of the Administration Shell, as these cannot grasp the semantic context of the data contained.

4.4.9 Usage of Short ID for Identifiable Elements

The Administration Shell fosters the use of worldwide unique identifiers to a large degree. However, in some cases, this may lead to inefficiencies. An example might be referring to a property, which is part of a submodel which is part of an Administration Shell and each of these identified by global Identifiers [4]. For example, in an application featuring a resource oriented architecture (ROA), a worldwide unique resource locator (URL) might be composed of a series of segments, which in turn do not need to be worldwide unique:

Figure 7 Motivation of exemplary identifiers and idShort



In order to allow such efficient addressing of elements by an API of an Administration Shell, idShort is provided for a set of classes of the metamodel, which inherit from abstract class Referable, in order to refer to such dependent elements (→ 4.6). However, an external system addressing resources of an Administration Shell is required to check the respective semantics by asserting *semanticId* first, before accessing elements by *id* or *idShort* (→ 4.7.2).

4.5 Events

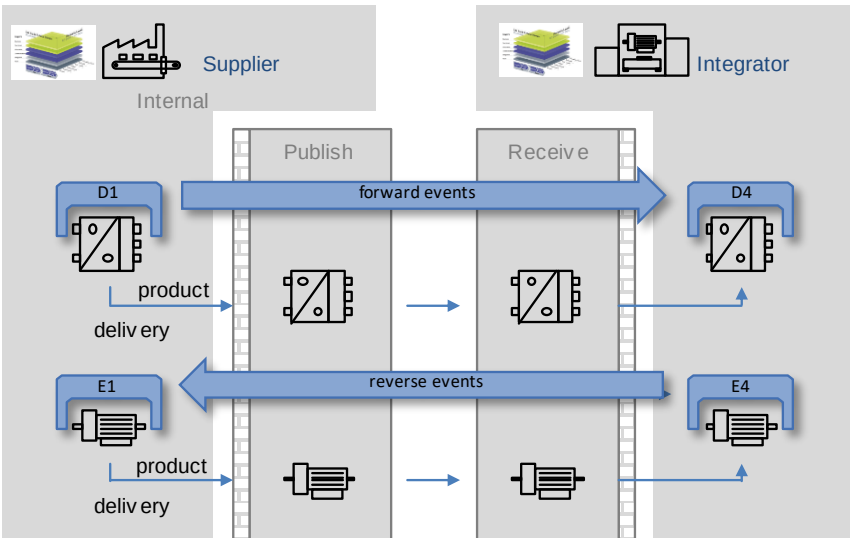
4.5.1 Overview

Events are a very versatile mechanism of the AAS. In the following sections, first some use-cases for events are described. Different types of events are summarized in order to depict requirements. A *SubmodelElement* "*Event*" is introduced, which is able to declare events of an AAS. The general format of event messages is specified.

4.5.2 Brief Use Cases for Events Used in Asset Administration Shells

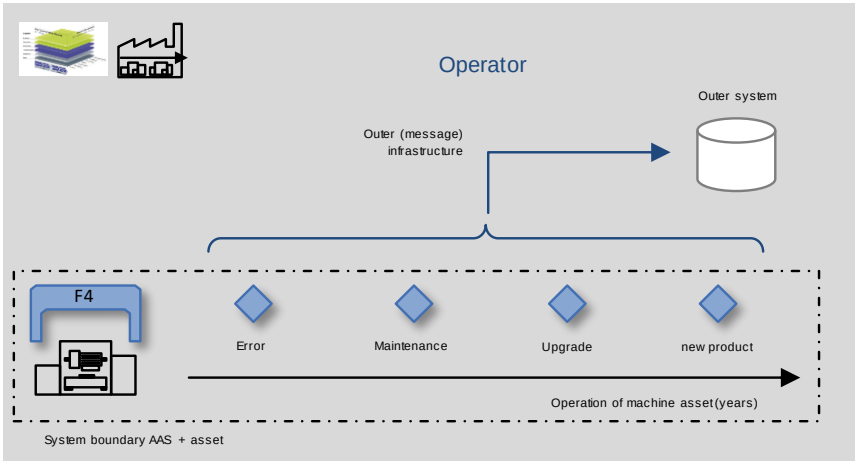
- An integrator has purchased a device. Later in time, the supplier of the device provides a new firmware. The integrator wants to detect the offer of a new firmware and wants to update the firmware after evaluating its suitability ("forward events"). The mechanism is, that a dependent AAS ("D4") detects events from a parent or type AAS ("D1"), which is described by the *derivedFrom* relation.
- An integrator/ operator operates a motor purchased from a supplier. During operation, condition monitoring incidents occur. Both parties agree on a business model providing availability. So, the supplier wants to monitor status of devices which are further in the value chain ("reverse events").

Figure 8 Forward and Revers Events



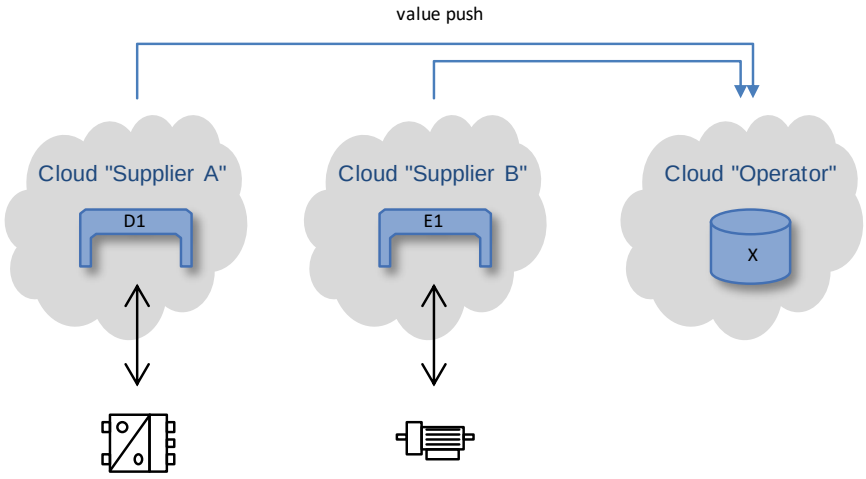
- An operator is operating a certain I4.0 component over time. Changes occasionally occur to these I4.0 components from different systems. For documentation and auditing, changes to this I4.0 component shall be tracked. This can be achieved by recording events over time.

Figure 9 Tracking of Changes via Events



- An operator is operating different I4.0 components, which are deployed to manufacturer clouds. The operator wants to integrate data from these components, according to DIN SPEC 92222. Therefore, information needs to be forwarded to the operator cloud ("value push").

Figure 10 Value Push Events across Clouds



4.5.3 Input and Output Directions of Events

It may be relevant to distinguish between input and output directions of an event with respect to the observed model, the respective Referable.

Direction	Descriptions
Output	The event is monitoring the *Referable* it is attached to. An outer message infrastructure, e.g. by OPC UA or MQTT or AMQP, will transport these events to other AASes and further outer systems and users.
Input	The software entity, which implements the respective *Referable*, can handle incoming events. These incoming events will be delivered by an outer message infrastructure, e.g. by OPC UA or MQTT or AMQP, to the software entity of the *Referable*.

4.5.4 Types of Events

According to the above use-cases, different types of events are possible. The following table gives an impression on possible event types. Each event type will be identified by a *semanticId* and will feature a specialized payload.

Group	**Direction**[7]	**Motivation / conditions**
Structural changes of the AAS	Out	• CRUD[8] of Submodels, Assets, SubmodelElements and such
	In	• Detect updates on parent/ type/ *derivedFrom* AAS
Updates of Properties and dependent attribute	Out	• update of values of SubmodelElements • timestamped updates and time series update • explicit triggering of an update event
Operation of AAS	Out	• monitoring of (long-lasting) execution of *OperationElement* and updating events while execution

[7] see below

[8] Create, Retrieve, Update, Delete

Monitoring, conditional, calculated events	Out	• e.g. when voiding some limits (e.g. stated by Qualifiers with expression semantics)
Infrastructure events	Out	• Booting, Shutdown, out of memory .. of software entity of respective Referable (AAS, Submodel)
Repository events	In/ Out	• Change of semantics of IRDIs (associated concept definition)
Security events	Out	• logging events • access violations, non-fitting roles & rights, denial of service, ..
Alarms & events	Out	• alarms and events management analog to distributed control systems (DCS)

Custom event types

In any case, it is possible to define custom event types by using a proprietary, but worldwide unique, semanticId for this event type. Such customized events can be sent or received by the software entity of the respective Referable, based on arbitrary conditions, triggers or behavior. However, the general format of the event messages needs to comply this specification, but the payload might be completely customized.

Event scopes

Events can be stated with an *observableReference* to the *Referables* of AAS, *Submodel*s, *SubmodelElementCollections* and *SubmodelElements*. These *Referables* are defining the scope of the events, which are to be received or sent.

Event attached to ..	Scope
AAS	This event is monitoring/ representing all logical elements of an Administration Shell, such as *AAS, Asset, Views, Submodels*.
Submodel	This event is monitoring/ representing all logical elements of the respective *Submodel* und all logical dependents.
SubmodelElementCollection	This event is monitoring/ representing all logical elements of the respective *SubmodelElementCollection* und all logical dependents.
SubmodelElement (others)	This event is monitoring/ representing a single atomic *SubmodelElement*, e.g. a data element which might include the contents of a *Blob* or *File*.

4.5.5 Possible Future Attributes of an Event

Up to now the metamodel offers a very simple modeling of an event. Besides the inherited attributes that are available for every *Referable* only one attribute for referencing the data or other elements being observed is added. This is expected to be a good starting point.

For future extensions other attributes like explained in the following tables.

Attributes of the event element

Class:	EventElement (non- normative, only for discussion)			
Explanation:	Defines the necessary information for sending or receiving events.			
Inherits from:	SubmodelElement			
Attribute (*=mandatory) (+=inherited)	Explanation	Type	Kind	Card.

Class:	EventElement (non- normative, only for discussion)			
idShort+	Identification of the element itself. Provides a unique identification for a possible event flow scheduling.	string	attr	1
semanticId+	Semantic identification of the event type.	Reference	attr	1
observableReference	Reference to the Referable, which defines the scope of the event. Can be AAS, Submodel, SubmodelElementCollection or SubmodelElement.	Reference	attr	1
direction	Can be { Input, Output }.	Enum	attr	1
state	Can be { On, Off }.	Enum	attr	1
messageTopic	Information for the outer message infrastructure for scheduling the event to the respective communication channel.	string	attr	0..1
messageBroker	Information, which outer message infrastructure shall handle messages for the EventElement. Refers to a Submodel, SubmodelElementCollection, which contains DataElements describing the proprietary specification for the message broker. Note: for different message infrastructure, e.g. OPC UA or MQTT or AMQP, these proprietary specification could be standardized by having respective Submodels.	Reference	attr	0..1
lastUpdate	Timestamp in UTC, when the last event was received (input direction) or sent (output direction).	xsd:dateTime	attr	0..1
minInterval	For input direction, reports on the maximum frequency, the software entity behind the respective Referable can handle input events. For output events, specifies the maximum frequency of outputting this event to an outer infrastructure. Might be not specified, that is, there is no minimum interval.	xsd:dateTime	attr	0..1
maxInterval	For input direction: not applicable. For output direction: maximum interval in time, the respective Referable shall send an update of the status of the event, even if not other trigger condition for the event was not met. Might be not specified, that is, there is no maximum interval.	xsd:dateTime	attr	0..1

Attributes of the event message

Events sent or received by AAS always comply to a general format. Exception: events exchanged in the course of an I4.0 interaction pattern.

Class:	EventMessage (non- normative, only for discussion)			
Explanation:	Defines the necessary information of an event instance sent out or received.			
Inherits from:	-			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
source	Reference to the source EventElement, including identification of AAS, Submodel, SubmodelElements.	Reference	attr	1
sourceSemanticId	semanticId of the source EventElement, if available	Reference	attr	0..1
observableReference	Reference to the Referable, which defines the scope of the event. Can be AAS, Submodel, SubmodelElementCollection or SubmodelElement.	Reference	attr	1
observableSemanticId	semanticId of the Referable, which defines the scope of the event, if available. See above.	Reference	attr	0..1
topic	Information for the outer message infrastructure for scheduling the event to the respective communication channel.	string	attr	0..1
subject	ABAC-Subject, who/ which initiated the creation	string	attr	0..1
timestamp	Timestamp in UTC, when this event was triggered.	string	attr	1
payload	Event specific payload. Detailed in annex.	string	attr	0..1

4.6 Overview Metamodel of the Administration Shell

Figure 11 Overview Metamodel of the Asset Administration Shell

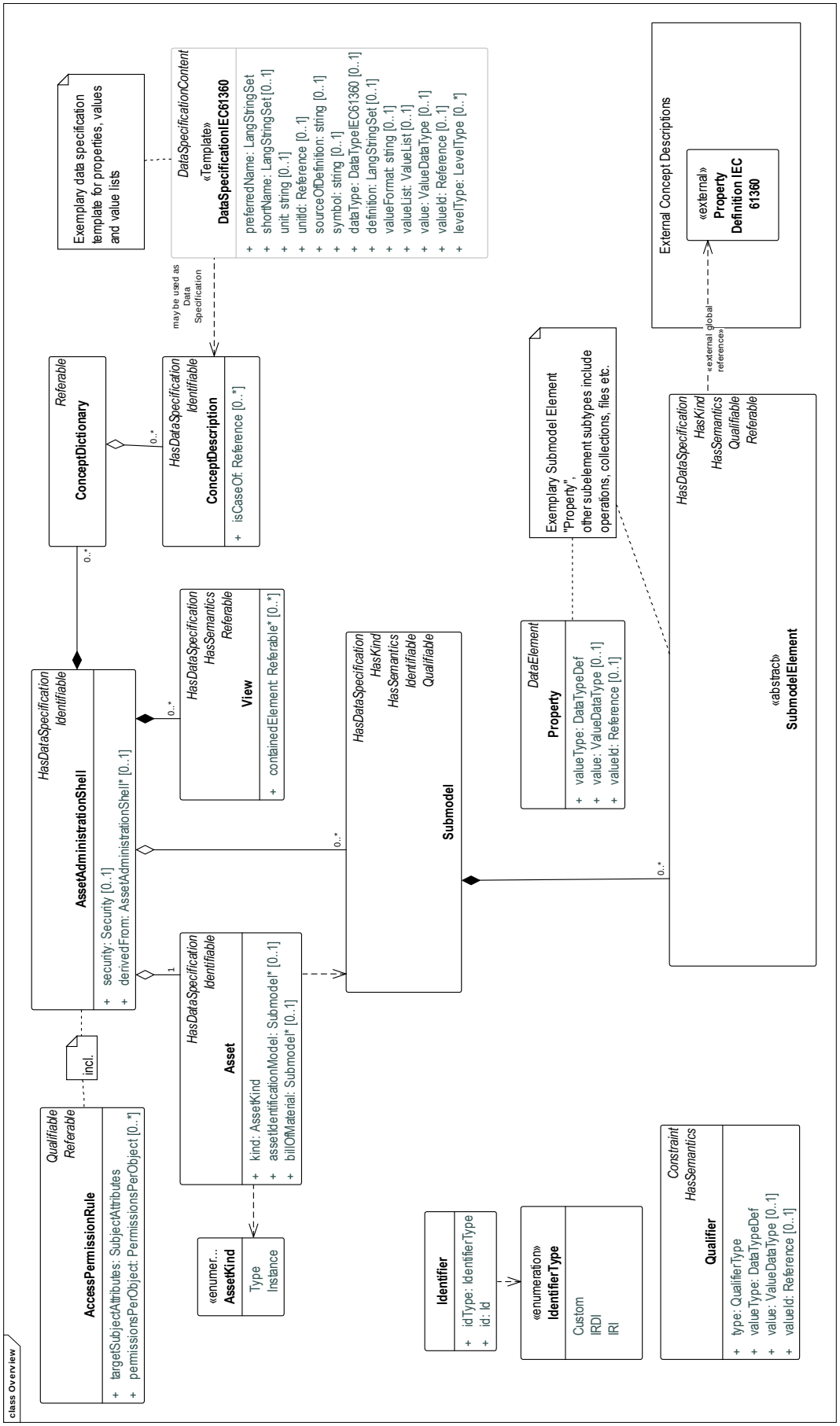
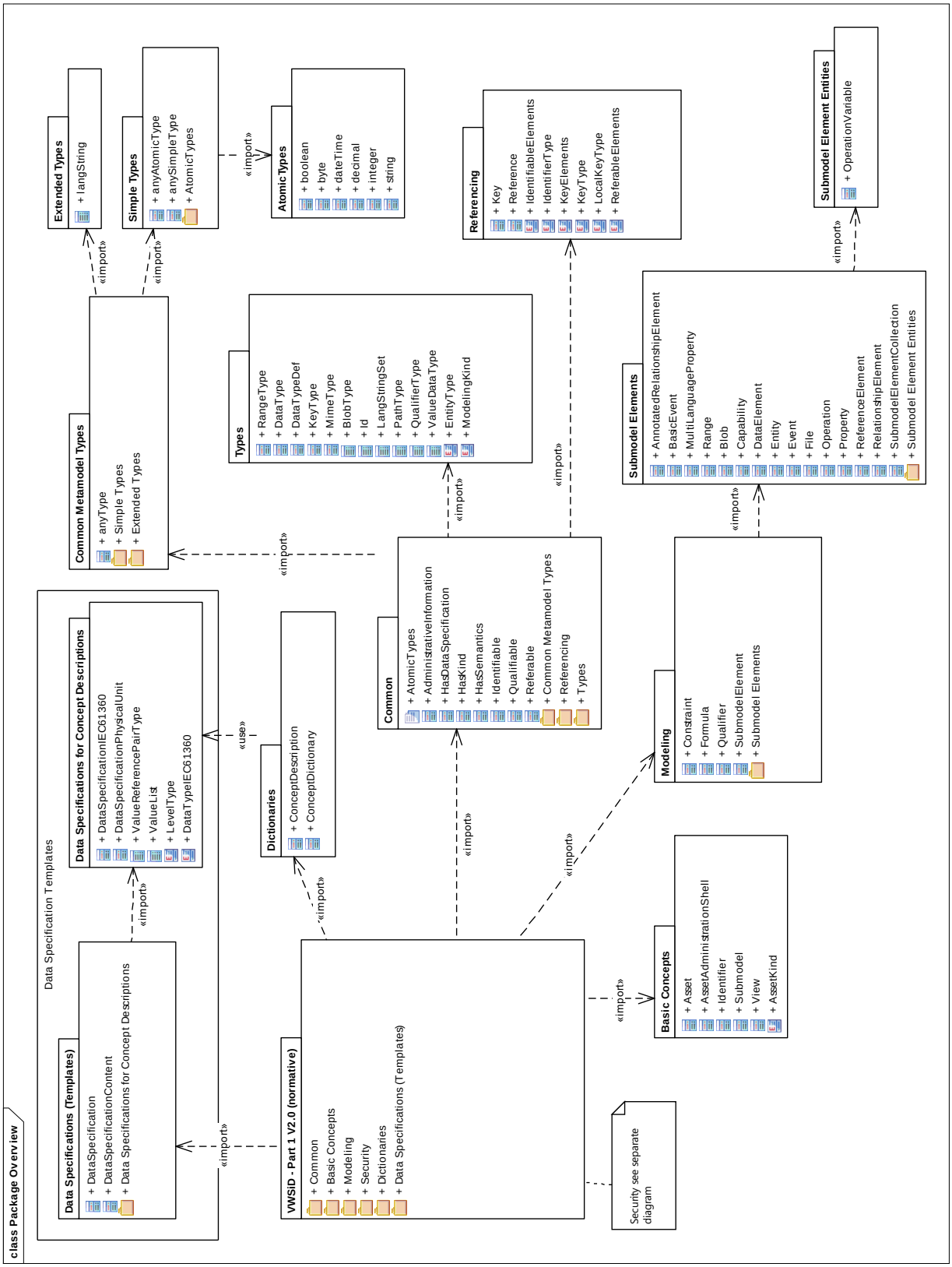


Figure 12 Metamodel package overview



In this clause an overview of the main concepts of the AssetAdministration Shell metamodel is presented.

The main parts of an Asset Administration Shell (AAS) is the asset it is representing as well as the submodels. Optionally, dictionaries and views may be part of the AAS. A dictionary contains concept descriptions used to describe the semantics of submodels. For details see Clause 4.7.3. Views define a set of elements selected for a specific stakeholder. For details see Clause 4.7.16.

An AAS represents exactly one asset. Asset types and asset instances are distinguished by setting the attribute "*kind*". For details see Clause 4.7.2.3.

Note: the UML modelling uses so-called abstract classes for denoting reused concepts like "HasSemantics", "Qualifiable" etc.

In case of an AAS of an instance asset, a reference to the AAS representing the corresponding asset type or another asset instance is was derived from may be added (*derivedFrom*). The same holds for AAS of an asset type: also types can be derived from other types.

An asset typically may be represented by several different identification properties like for example the serial number, its RFID code etc. Such local identification properties are defined in the asset identification submodel (*assetIdentificationModel*). For details see Clause 4.7.4.

AASs, assets, submodels and concept descriptions need to be globally uniquely identifiable (*Identifiable*). Other elements like for example properties, single local dictionaries just need to be referable within the model and thus only need a local identifier (*idShort* from *Referable*). For details on identification see Clause 4.3. For details on *Identifiable* and *Referable* see Clause 4.7.2.1.

*Submodel*s consist of a set of submodel elements. Submodel elements may be qualified by a so-called *Qualifier*. For details see Clause 4.7.5.

There are different subtypes of submodel elements like properties, operations, collections etc. For details see Clause 4.7.5. A typical submodel element is shown in the overview figure: a property. A property is a data submodel element that has a value of simple type like string, date etc. For details on properties see Clause 4.7.7.

Every submodel element needs a semantic definition *(semanticId* in *HasSemantics*). The submodel element might either refer directly to a corresponding semantic definition provided by an external reference (e.g. to an eCl@ss or IEC CDD property definition) or it may reference a submodel element of *kind = Template* that defines the semantics of submodel elements of *kind = Instance*. For details see Clause 4.7.2.5.

The AAS itself can also define its own dictionary that contains semantic definitions of its submodel elements. These semantic definitions are called concept descriptions (*ConceptDescription*). It is optional whether an AAS defines its own concept dictionary (*ConceptDictionary*) or not. For details see Clause 4.7.20.

The concept dictionary may contain copies of property definitions of external standards. In this case a semantic definition to the external standard shall be added (*isCaseOf). isCaseOf* is a more formal definition of *sourceOfDefinition* that is just text.

Note: in this case most of the attributes are redundant because these are defined in the external standard. It is about usability to add attributes for information like *preferredName*, *unit* etc. Consistency w.r.t. to the referenced submodel element definitions should be ensured by corresponding tooling.

The concept dictionary may also contain proprietary definitions. In this case the provider of the AAS shall be aware that no interoperability with other AAS can be ensured.

Data Specification Templates can be used (*DataSpecification*) to define which attributes (besides those predefined by the metamodel) are used to define a submodel element or a concept description. For the concept description of properties typically the Data Specification Template following IEC 61360 is used. For denoting recommended Data Specification Templates to be used the <<*template*>>-dependency is used. For details see Clause 4.7.2.6.

Some Data Specification Templates like the template for IEC 61360 property definitions (*DataSpecificationIEC61360*) are explicitly predefined and recommended to be used by the Plattform Industrie 4.0. For details see Clause 4.8.2. If proprietary templates are used, again, interoperability with other AAS cannot be ensured.

Besides submodel elements including properties and concept descriptions also other identifiable elements may use additional templates (*HasDataSpecification*). For details see Clause 4.7.2.7.

Submodel elements and the submodels themselves may have additional qualifiers (*Qualifiable*). Per *Qualifiable* there might be more than one qualifier. For details see Clause 4.7.2.6.

Additionally, Views can be defined within an AAS. Views may consist of any elements that are referable (*containedElement*). A “Safety View”, for example, contains all properties or operations that are safety relevant and need special treatment. For details see Clause 4.7.16. A View definition can also be used in different life cycle stages. For example, there could be a view for engineering and all referenced artefacts are deleted before delivering the AAS to the customer.

For every AAS security aspects need to be considered (*security*). In this document the aspect of access control is covered in more detail. So-called access permission rules are defined, that define which permission a specific authenticated subject has on which object. For details see Clause 6.

Figure 12 gives a complete picture of all elements defined in the metamodel excluding security. Information on the Security part is found in Clause 6.4.

4.7 Metamodel Specification Details: Designators

4.7.1 Introduction

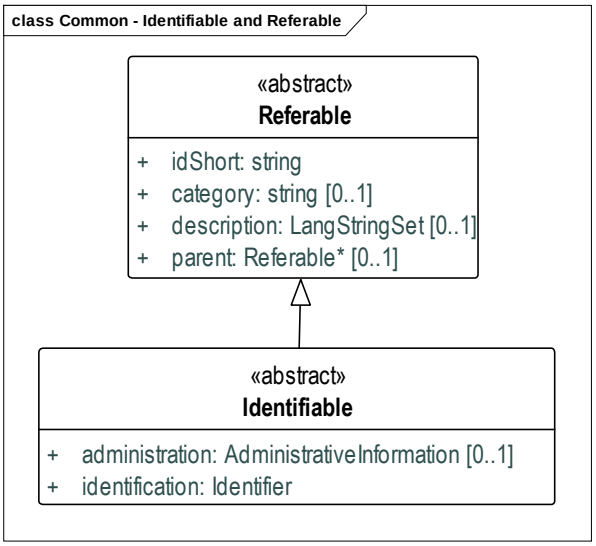
In this clause the classes of the metamodel are specified in detail. In Annex B the template used to describe the classes and relationships is explained. In Annex C some of the diagrams are shown together with all its inherited attributes to give a complete overview.

For understanding the specifications, it is crucial to understand the common attributes first (Clause 4.7.2). They are reused throughout the specifications of the other classes (“inherits from”) and define important concepts like identifiable, qualifiable etc. They are abstract, i.e. there is no object instance of such classes.

4.7.2 Common Attributes

4.7.2.1 Identifiables & Referables

Figure 13 Metamodel for Identifiables and Referables



The metamodel distinguishes between elements that are identifiable, referable or none of both. An identifiable element as a globally unique identifier (*Identifier*). Referable elements can be referenced but for doing so the context of the element is needed. A referable has a short unique identifier (*idShort*) that is unique just in its context, its name space. An identifiable is also referable but there are elements that are not referable: they are just attributes of a referable. Identifiables may have administrative information like version etc.

A name space is defined as follows in this context: The parent element an element is part of and that is either referable or identifiable is the name space of the element. Examples: A submodel is the name space for the properties contained in it. The name space of a submodel element being contained in a submodel element collection is the submodel element collection. However, for identifiables the name space is not important since identifiables per definition have a global identifier.

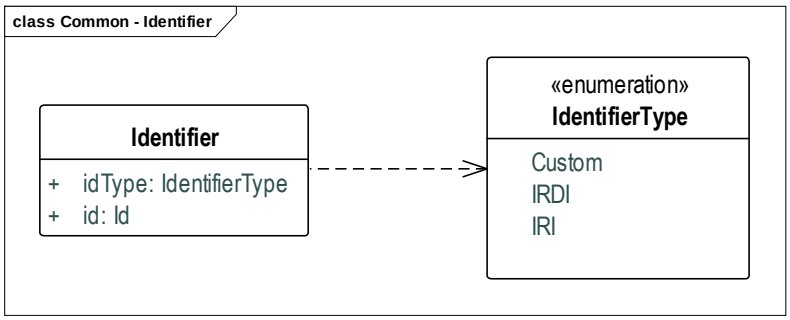
Class:	Referable <<abstract>>			
Explanation:	An element that is referable by its idShort. This id is not globally unique. This id is unique within the name space of the element.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
idShort*	Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: *idShort* shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: *idShort* shall be matched case-insensitive. Note: In case of an identifiable element *idShort* is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the “BrowserPath” in OPC UA. Note: In case the element is a property and the property has a semantic definition (*HasSemantics*) the *idShort* is typically identical to the short name in English.	string	attr	1
category	The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints. Note: The category is not identical to the semantic definition (*HasSemantics*) of an element. The category e.g. could denote that the element is a measurement value whereas the semantic definition of the element would denote that it is the measured temperature.	string	attr	0..1
description	Description or comments on the element. The description can be provided in several languages.	LangStringSet	attr	0..1
parent	Reference to the next referable parent element of the element.	Referable	ref*	0..1

Class:	Referable <<abstract>>			
	Constraint AASd-004: Add parent in case of non-identifiable elements. Note: This element is used to ease navigation in the model and thus it enables more performant implementation. In does not give any additional information.			

Class:	Identifiable <<abstract>>			
Explanation:	An element that has a globally unique identifier.			
Inherits from:	Referable			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
administration	Administrative information of an identifiable element. Note: Some of the administrative information like the version number might need to be part of the identification.	AdministrativeInformation	attr	0..1
identification*	The globally unique identification of the element.	Identifier	attr	1

4.7.2.2 Identifier

Figure 14 Metamodel for Identifier



Information about identification can be found in Clause 4.4. In Clause 4.4.4 constraints and recommendation on when to use which type of Identifier can be found.

Examples for Identifiers can be found in Clause 4.4.3 Identifiers for Assets and Administration Shells.

See Clause 4.4.4 for information which identifier types are supported.

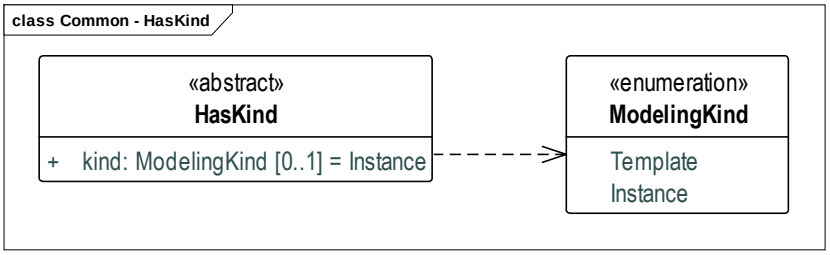
Class:	Identifier
Explanation:	Used to uniquely identify an entity by using an identifier.
Inherits from:	--

Class:	Identifier			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
idType*	Type of the Identifier, e.g. IRI, IRDI etc. The supported Identifier types are defined in the enumeration "*IdentifierType*".	IdentifierType	attr	1
id*	Identifier of the element. Its type is defined in *idType*.	Id	attr	1

Enumeration:	IdentifierType
Explanation:	Enumeration of different types of Identifiers for global identification
Literal	Explanation
IRDI	IRDI according to ISO29002-5 as an Identifier scheme for properties and classifications.
IRI	IRI according to Rfc 3987. Every URI is an IRI.
Custom	Custom identifiers like GUIDs (globally unique identifiers)

4.7.2.3 Type or Instance of Model Elements (HasKind)

Figure 15 Metamodel for HasKind



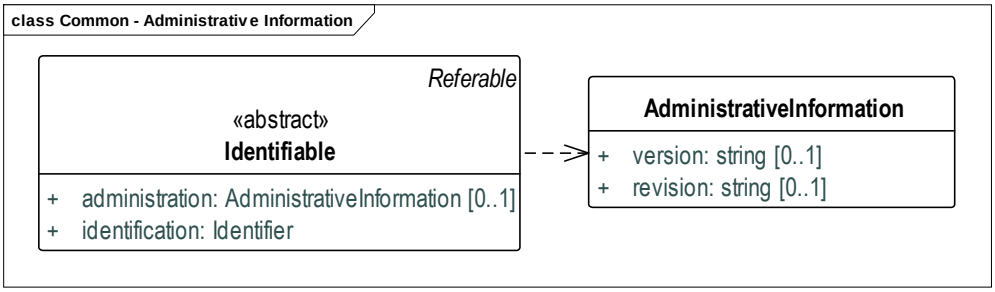
Class:	HasKind			
Explanation:	An element with a kind is an element that can either represent a template (type) or an instance. Default for an element is that it is representing an instance.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
kind	Kind of the element: either type or instance. Default Value = *Instance*	ModelingKind	attr	0..1

The kind enumeration is used to denote whether an element is of kind Template or Instance.

Enumeration:	ModelingKind
Explanation:	Enumeration for denoting whether an element is a template or an instance.
Inherits from:	--
Literal	Explanation
Template	Software element which specifies the common attributes shared by all instances of the template. [SOURCE: IEC TR 62390:2005-01, 3.1.25] modified
Instance	Concrete, clearly identifiable component of a certain template. Note: It becomes an individual entity of a template, for example a device model, by defining specific property values. Note: In an object oriented view, an instance denotes an object of a template (class). [SOURCE: IEC 62890:2016, 3.1.16 65/617/CDV] modified

4.7.2.4 Administrative Information

Figure 16 Metamodel for Administrative Information



Every *Identifiable* may have administrative information. Administrative information includes for example

- Information about the version of the element
- Information about who created or who made the last change to the element
- Information about the languages available in case the element contains text, for translating purposed also the master or default language may be defined

In the first version of the AAS metamodel only version information as defined by IEC 61360 is defined. In later versions additional attributes may be added.

Version corresponds in principle to the *version_identifier* according to IEC 62832 but is not used for concept identifiers only (IEC TS 62832-1) but for all identifiable elements. Version and revision together correspond to the version number according to IEC 62832.

AdministrativeInformation allows the usage of templates (*HasDataSpecification*) but there are no predefined templates in this version of the metamodel.

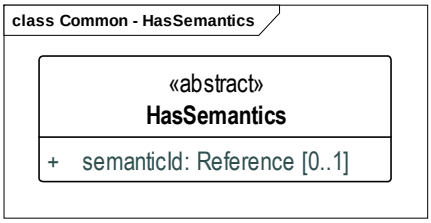
Note: Some of the administrative information like the version number might need to be part of the identification.

Class:	AdministrativeInformation
Explanation:	Administrative metainformation for an element like version information.

Class:	AdministrativeInformation			
Inherits from:	HasDataSpecification			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
version	Version of the element.	string	attr	0..1
revision	Revision of the element. Constraint AASd-005: A revision requires a version. This means, if there is no version there is no revision neither.	string	attr	0..1

4.7.2.5 Semantic References (HasSemantics)

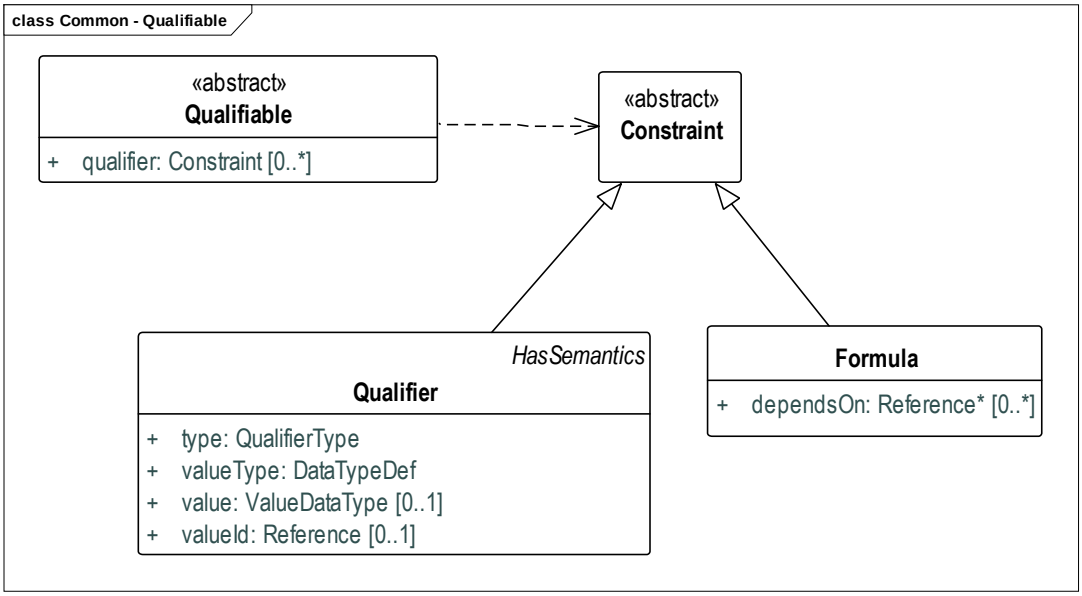
Figure 17 Metamodel for Semantic References (HasSemantics)



Class:	HasSemantics <<abstract>>			
Explanation:	Element that can have a semantic definition.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
semanticId	Identifier of the semantic definition of the element. It is called semantic id of the element. The semantic id may either reference an external global id or it may reference a referable model element of kind=Template that defines the semantics of the element. Note: In many cases the idShort is identical to the English short name within the semantic definition as referenced via its semantic id.	Reference	attr	0..1

4.7.2.6 Qualifiables and Formulas - Constraints

Figure 18 Metamodel Qualifiables, Formulas and Constraints



For qualifiable elements additional qualifiers may be defined. For details on qualifiers and for predefined standardized qualifier types see IEC 62569-1. For example, a level qualifier defining the level type minimal value, maximum value, typical value and nominal value can be found in IEC 62569-1. Additional qualifier types are planned to be defined in the ongoing work of DIN SPEC 92000 like for example expressions semantics and expression logic.

If there are no predefined qualifier types or the additional qualification is quite complex then instead of a set of qualifiers also a formula can be defined.

In Figure 19 an example for a formula depending on the property “Status” is shown. Up to now no formula language is defined for the AAS.

Figure 19 Example Formula “Machine Status not Running”

```
<aas:Formula>
 <aas:dependsOn>
   <Keys> <Key local="True" type="AssetAdministrationShell" idType="IRI">http://myShell</Key>
          <Key local="True" type="Submodel" idType="IdShort">Maschine</Key>
          <Key local="True" type="Property" idType="IdShort">Status</Key>
 </aas:dependsOn> != RUNNING
</aas:Formula>
```

Class:	Qualifiable <<abstract>>			
Explanation:	The value of a qualifiable element may be further qualified by one or more qualifiers or complex formulas.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
qualifier	Additional qualification of a qualifiable element.	Constraint	aggr	0..*

Class:	Constraint <<abstract>>			
Explanation:	A constraint is used to further qualify an element.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.

Class:	Qualifier			
Explanation:	A qualifier is a type-value-pair that makes additional statements w.r.t. the value of the element.			
Inherits from:	Constraint; HasSemantics			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
type*	The qualifier *type* describes the type of the qualifier that is applied to the element.	QualifierType	attr	1
valueType*	Data type of the qualifier value.	DataTypeDef	attr	1
value	The qualifier value is the value of the qualifier. Constraint AASd-006: if both, the value and the valueId are present then the value needs to be identical to the value of the referenced coded value in Qualifier/valueId.	ValueDataType	attr	0..1
valueId	Reference to the global unqiue id of a coded value.	Reference	attr	0..1

Class:	Formula			
Explanation:	A formula is used to describe constraints by a logical expression.			
Inherits from:	Constraint			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
dependsOn	A formula may depend on referable or even external global elements that are used in the logical expression. The value of the referenced elements needs to be accessible so that it can be evaluated in the formula to true or false in the corresponding logical expression it is used in.	Reference	aggr	0..*

--

4.7.2.7 Used Templates for Data Specification (HasDataSpecification)

Figure 20 Metamodel HasDataSpecification

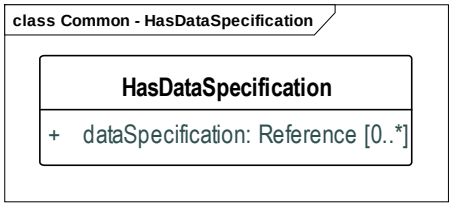
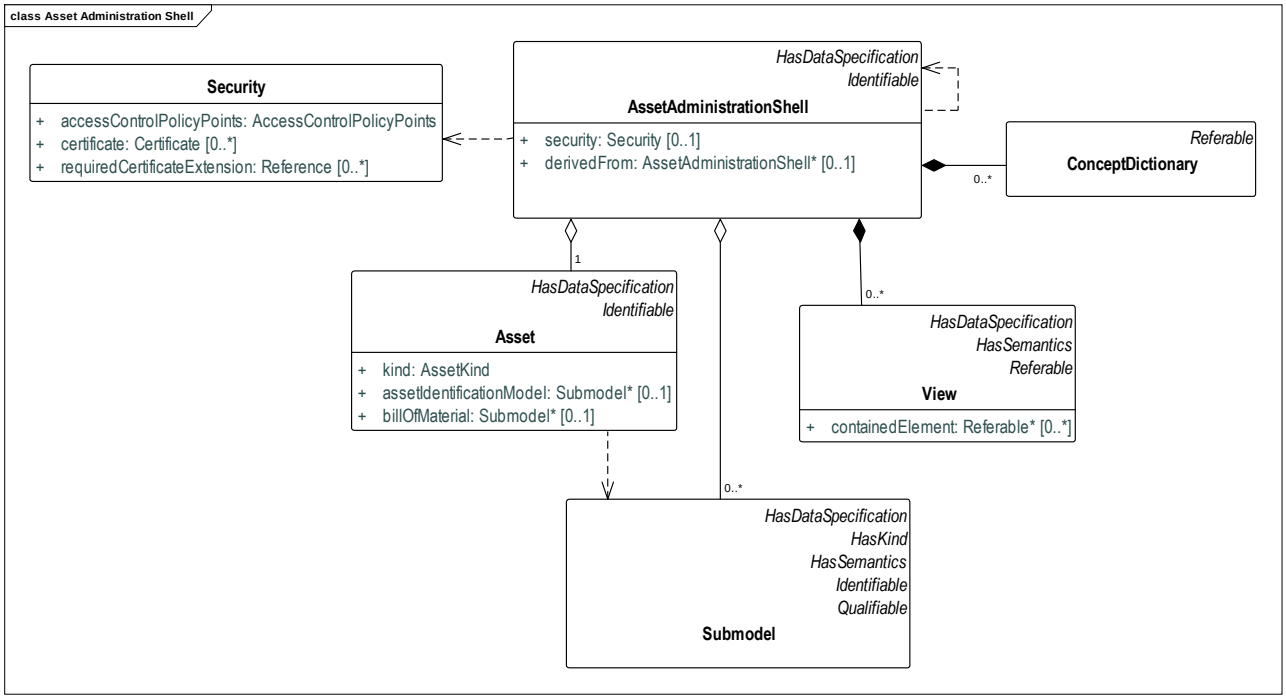


Figure 21 Metamodel for HasDataSpecification

Class:	HasDataSpecification			
Explanation:	Element that can be extended by using data specification templates. A data specification template defines the additional attributes an element may or shall have. The data specifications used are explicitly specified with their global id.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
dataSpecification	Global reference to the data specification template used by the element.	Reference	aggr	0..*

4.7.3 Asset Administration Shell Attributes

Figure 22 Metamodel AssetAdministrationShell



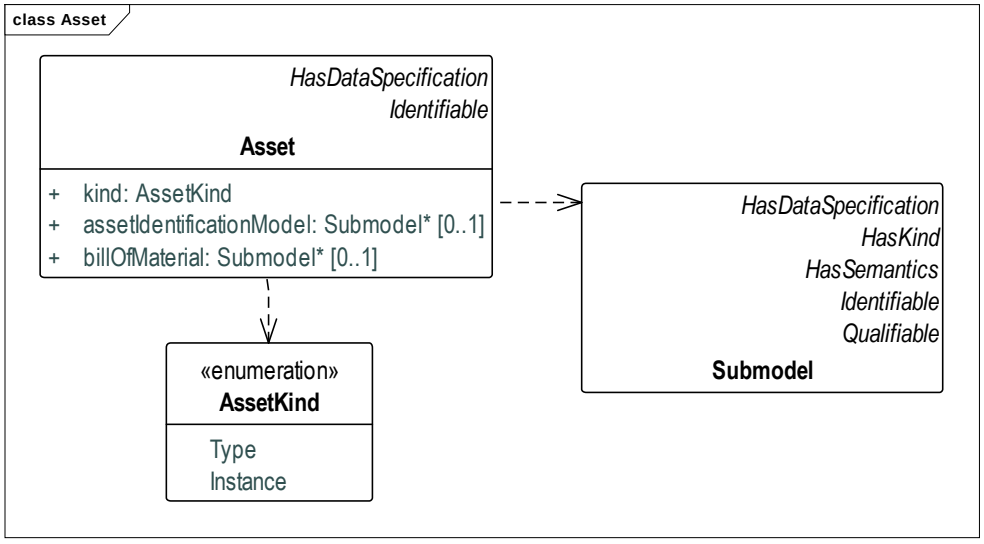
An Administration Shell is uniquely identifiable since it inherits from *Identifiable*.

The *derivedFrom* attribute is used to establish a relationship between two AssetAdministration Shells that are derived from each other. For more detailed information on the *derivedFrom* concept see Clause 4.2 Types and Instances.

Class:	AssetAdministrationShell			
Explanation:	An AssetAdministration Shell.			
Inherits from:	HasDataSpecification,; Identifiable			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
derivedFrom	The reference to the AAS the AAS was derived from.	AssetAdministrationShell	ref*	0..1
security	Definition of the security relevant aspects of the AAS.	Security	aggr	0..1
asset*	The asset the AAS is representing.	Asset	ref*	1
submodel	The asset of an AAS is typically described by one or more submodels. Temporarily no submodel might be assigned to the AAS.	Submodel	ref*	0..*
conceptDictionary	An AAS max have one or more concept dictionaries assigned to it. The concept dictionaries typically contain only descriptions for elements that are also used within the AAS (via *HasSemantics*).	ConceptDictionary	aggr	0..*
view	If needed stakeholder specific views can be defined on the elements of the AAS.	View	aggr	0..*

4.7.4 Asset Attributes

Figure 23 Metamodel of Asset



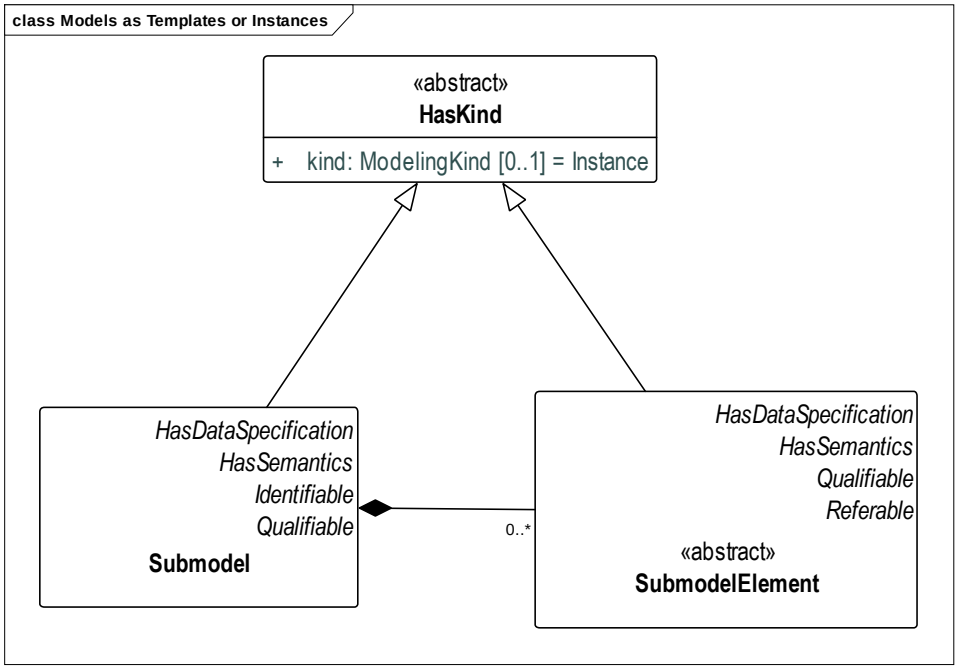
Class:	Asset			
Explanation:	An *Asset* describes meta data of an asset that is represented by an AAS. The asset may either represent an asset type or an asset instance. The asset has a globally unique identifier plus – if needed – additional domain specific (proprietary) identifiers.			
Inherits from:	HasDataSpecification; Identifiable			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
kind*	Denotes whether the Asset of of kind “Type” or “Instance”.	AssetKind	attr	1
assetIdentificationModel	A reference to a *Submodel* that defines the handling of additional domain specific (proprietary) Identifiers for the asset like e.g. serial number etc.	Submodel	ref*	0..1
billOfMaterial	Bill of material of the asaset represented by a submodel of the same AAS. This submodel contains a set of entities describing the material used to compose the composite I4.0 Component.	Submodel	ref*	0..1

Enumeration:	AssetKind
Explanation:	Enumeration for denoting whether an element is a type or an instance.
Inherits from:	--
Literal	Explanation
Type	hardware or software element which specifies the common attributes shared by all instances of the type [SOURCE: IEC TR 62390:2005-01, 3.1.25]
Instance	concrete, clearly identifiable component of a certain type Note: It becomes an individual entity of a type, for example a device, by defining specific property values. Note: In an object oriented view, an instance denotes an object of a class (of a type). [SOURCE: IEC 62890:2016, 3.1.16] 65/617/CDV

For more information on types and instances see Clause 4.2.

4.7.5 Submodel and Submodel Element Attributes

Figure 24 Metamodel for Submodel



Class:	Submodel			
Explanation:	A submodel defines a specific aspect of the asset represented by the AAS. A submodel is used to structure the digital representation and technical functionality of an Administration Shell into distinguishable parts. Each submodel refers to a well-defined domain or subject matter. Submodels can become standardized and thus become submodels types. Submodels can have different life-cycles.			
Inherits from:	HasDataSpecification; HasSemantics; Identifiable; Qualifiable; HasKind			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
submodelElement	A submodel consists of zero or more submodel elements.	SubmodelElement	aggr	0..*

A submodel instance can reference the submodel template it was derived from. Formulated in a technical way: *semanticId* of a *Submodel* with *kind=Instance* may refer to a *Submodel* of *kind=Template (kind* inherited via *HasKind)*.

A submodel can be qualified (*Qualifiable*).

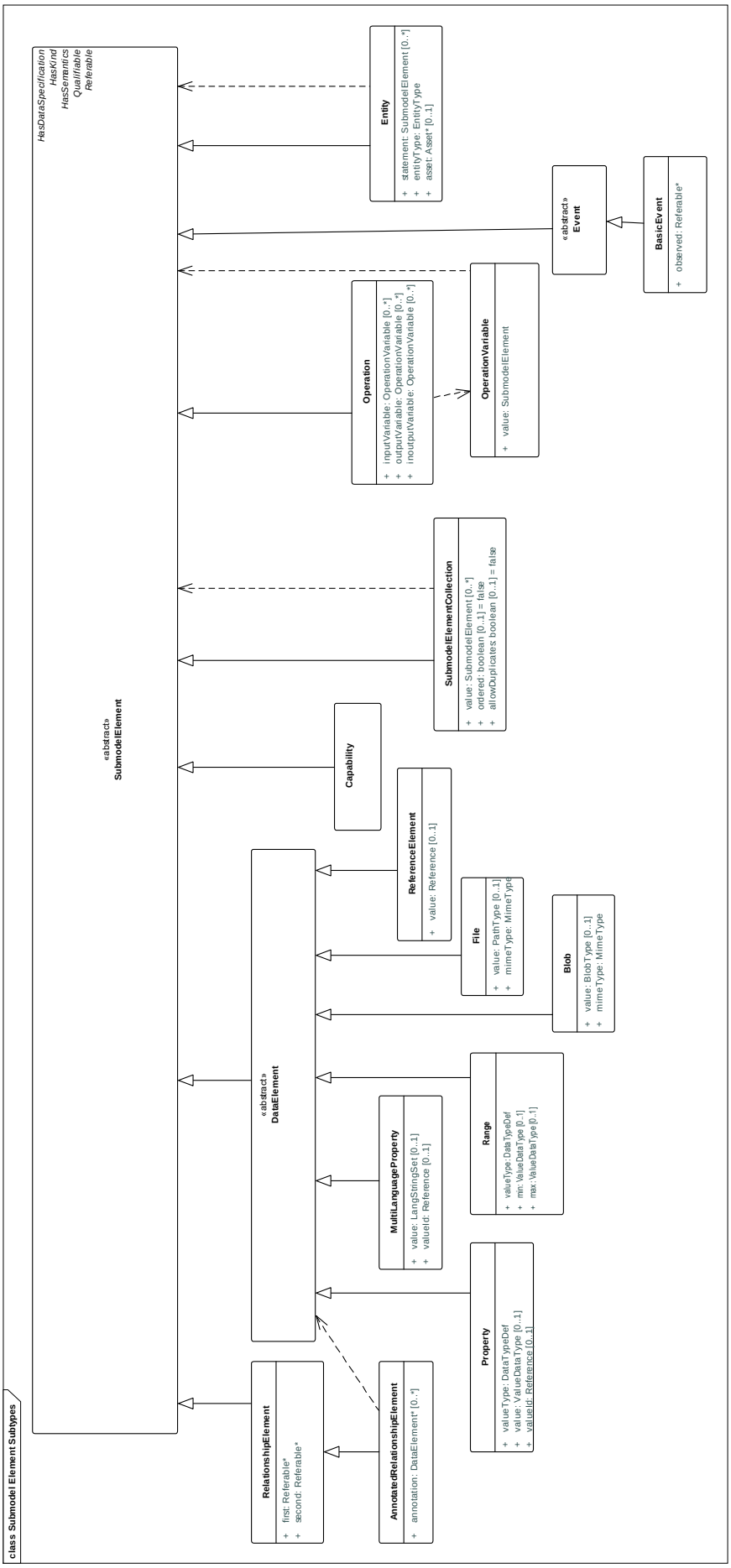
Submodel element are qualifiable elements, i.e. one or more qualifier may be defined for each of them.

Submodels and submodel elements may also have data specification templates defined for them. A template might for example be defined to mirror some of the attributes like *preferredName* and *unit* of a property definition if the AAS does not contain a corresponding concept description. Otherwise there only is the property definition referenced by *semanticId* available for the property: the lookup of the attributes has to be realized online in a different way and is not available offline.

In case the submodel is of *kind=Template* then the submodel elements within the submodel are presenting submodel element types. In case the submodel is of *kind=Instance* then its submodel elements represent submodel element instances.

Class:	SubmodelElement <<abstract>>			
Explanation:	A submodel element is an element suitable for the description and differentiation of assets. NOTE: The concept of type and instance applies to submodel elements. Properties are special submodel elements. The property types are defined in dictionaries (like the IEC Common Data Dictionary or eCl@ss), they do not have a value. The property type (kind=Template is also called data element type in some standards. The property instances (kind=Instance) typically have a value. A property instance is also called property-value pair in certain standards.			
Inherits from:	HasDataSpecification; Referable; Qualifiable; HasSemantics; HasKind			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.

Figure 25 Metamodel Overview for Submodel Element Subtypes

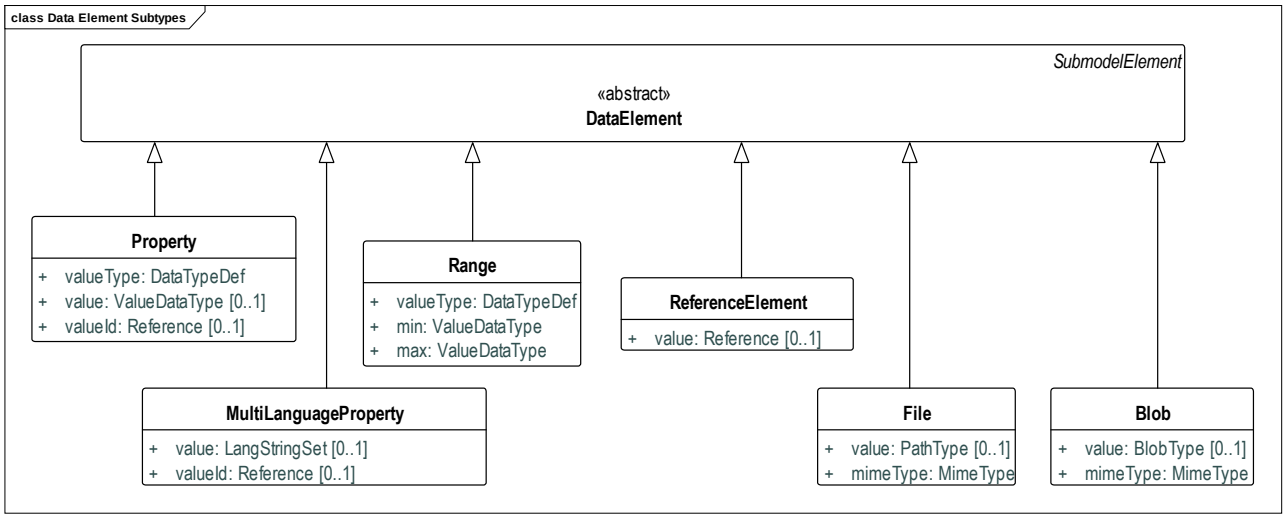


4.7.6 Overview of Submodel Element Types

Submodel elements include data properties as well as operations, events and other elements needed to describe a model for an asset (see Figure 25).

4.7.7 Data Element Attributes

Figure 26 Metamodel for Data Elements



A data element is a submodel element that is not further composed out of other submodel elements.

A data element is a submodel element that has a value or a predefined number of values like range data elements.

A controlled value is a value whose meaning is given in an external source (see “ISO/TS 29002-10:2009(E)”).

The type of value differs for different subtypes of data elements. Data Elements include properties and file handling and reference elements, see Figure 25.

The following categories are defined for data elements except for files and blobs:

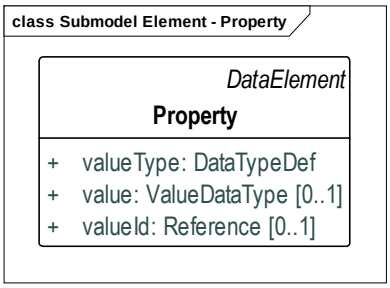
Category:	Applicable to:	Explanation:
CONSTANT	Property MultiLanguageProperty	A constant property is a property with a value that does not change over time. In eCl@ss this kind of category has the category “Coded Value”.
PARAMETER	Property MultiLanguageProperty	A parameter property is a property that is once set and then typically does not change over time. This is for example the case for configuration parameters.
VARIABLE	Property MultiLanguageProperty	A variable property is a property that is calculated during runtime, i.e. its value is a runtime value.

Class:	DataElement <<abstract>>
Explanation:	A data element is a submodel element that is not further composed out of other submodel elements. A data element is a submodel element that has a value. The type of value differs for different subtypes of data elements.
Inherits from:	SubmodelElement

Class:	DataElement <<abstract>>			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.

4.7.8 Property Attributes

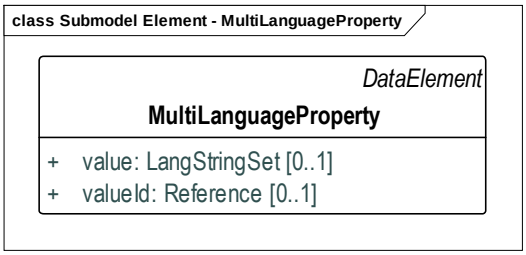
Figure 27 Metamodel for Property



Class:	Property			
Explanation:	A property is a data element that has a single value.			
Inherits from:	DataElement			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
valueType*	Data type of the value	DataTypeDef	attr	1
value	The value of the property instance.	ValueDataType	attr	0..1
valueId	Reference to the global unqiue id of a coded value. Constraint AASd-007: if both, the value and the valueId are present then the value needs to be identical to the value of the referenced coded value in valueId.	Reference	attr	0..1

4.7.9 Multi Language Property Attributes

Figure 28 Metamodel for MultiLanguageProperty

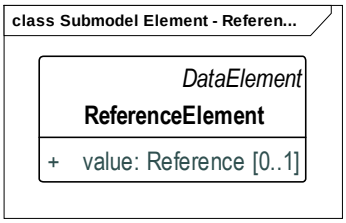


Class:	MultiLanguageProperty
Explanation:	A property is a data element that has a multi language value.

Class:	MultiLanguageProperty			
Inherits from:	DataElement			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
value	The value of the property instance.	LangStringSet	attr	0..1
valueId	Reference to the global unqiue id of a coded value. Constraint AASd-012: if both, the value and the valueId are present then for each string in a specific language the meaning must be the same as specified in valueId.	Reference	attr	0..1

4.7.10 Reference Element Attributes

Figure 29 Metamodel for ReferenceElement

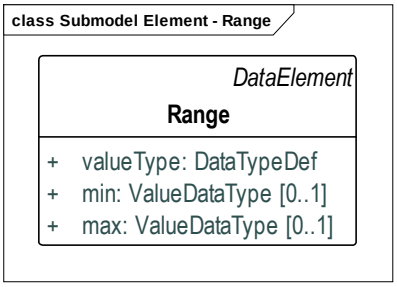


Class:	ReferenceElement			
Explanation:	A reference element is a data element that defines a logical reference to another element within the same or another AAS or a reference to an external object or entity.			
Inherits from:	DataElement			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
value	Reference to any other referable element of the same of any other AAS or a reference to an external object or entity.	Reference	aggr	0..1

For more information on references see Clause 4.7.21.

4.7.11 Range Attributes

Figure 30 Metamodel for Range

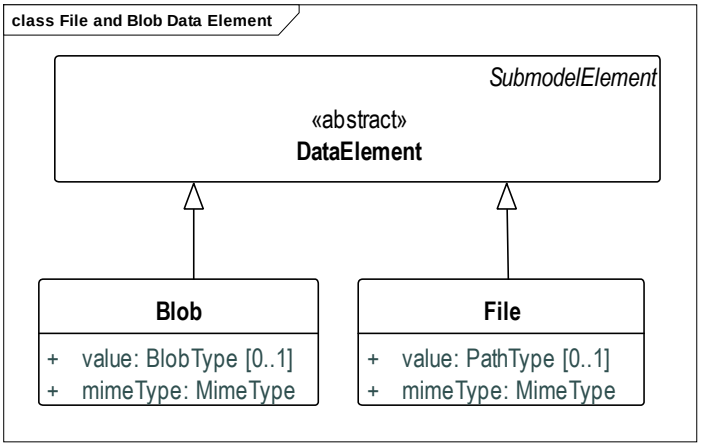


Class:	Range			
Explanation:	A range data element is a data element that defines a range with min and max.			
Inherits from:	DataElement			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
valueType*	Datat type of the min und max	DataTypeDef	attr	1
min	The minimum value of the range. If the min value is missing then the value is assumed to be negative infinite. Constraint AASd-013: In case of a range with kind=Instance either the min or the max value or both need to be defined.	ValueDataType	attr	0..1
max	The maximum value of the range. If the max value is missing then the value is assumed to be positive infinite.	ValueDataType	attr	0..1

If the *semanticId* is a reference to a concept description then it is a concept description with data specification IEC61360. The value for *levelType* is the set {Min, Max}.

4.7.12 Blob and File Attributes

Figure 31 Metamodel for Blob and File



A media type (also MIME type and content type) […] is a two-part Identifier for file formats and format contents transmitted on the Internet. The Internet Assigned Numbers Authority (IANA) is the official authority for the standardization and publication of these classifications. Media types were originally defined in Request for Comments 2045 in November 1996 as a part of MIME (Multipurpose Internet Mail Extensions) specification, for denoting type of email message content and attachments; […] hence the name /MIME type.[9]

Class:	Blob			
Explanation:	A BLOB is a data element that represents a file that is contained with its source code in the value attribute.			
Inherits from:	DataElement			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
value	The value of the BLOB instance of a blob data element. Note: In contrast to the file property the file content is stored directly as value in the Blob data element.	BlobType	attr	0..1
mimeType*	Mime type of the content of the BLOB. The mime type states which file extension the file has. Valid values are e.g. “application/json”, “application/xls”, ”image/jpg” The allowed values are defined as in RFC2046.	MimeType	attr	1

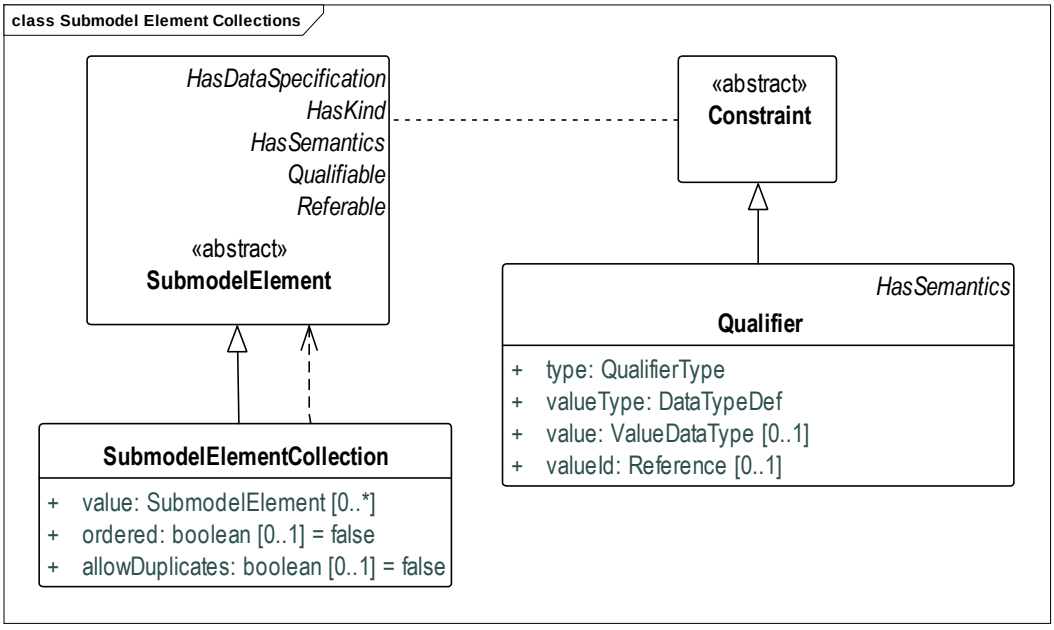
[9] Wikipedia.org, date: 2018-04-09

Class:	File			
Explanation:	A File is a data element that represents an address to a file. The value is an URI that can represent an absolute or relative path.			
Inherits from:	SubmodelElement			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
value	Path and name of the referenced file (with file extension). The path can be absolute or relative.	PathType	attr	0..1
mimeType*	Mime type of the content of the file.	MimeType	attr	1

For handling of supplementary external files in exchanging AAS specification in aasx format see also Clause 7.4 Conventions for the Asset Administration Shell package file format (AASX). An absolute path is used in the case that the file exists independently of the AAS. A relative path, relative to the package root should be used if the file is part of the serialized package of the AAS.

4.7.13 Submodel Element Collection Attributes

Figure 32 Metamodel for Submodel Element Collections

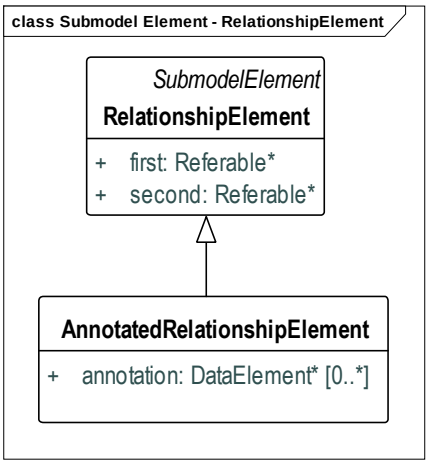


Class:	SubmodelElementCollection			
Explanation:	A submodel element collection is a set or list of submodel elements.			
Inherits from:	SubmodelElement			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
value	Submodel element contained in the collection.	SubmodelElement	aggr	0..*

Class:	SubmodelElementCollection			
ordered	If ordered=false then the elements in the property collection are not ordered. If ordered=true then the elements in the collection are ordered. Default = false Note: An ordered submodel element collection is typically implemented as an indexed array.	boolean	attr	0..1
allowDuplicates	If allowDuplicates=true then it is allowed that the collection contains the same element several times. Default = false	boolean	attr	0..1

4.7.14 Relationship and Annotated Relationship Attributes

Figure 33 Metamodel for Relationship and Annotated Relationhip Elements

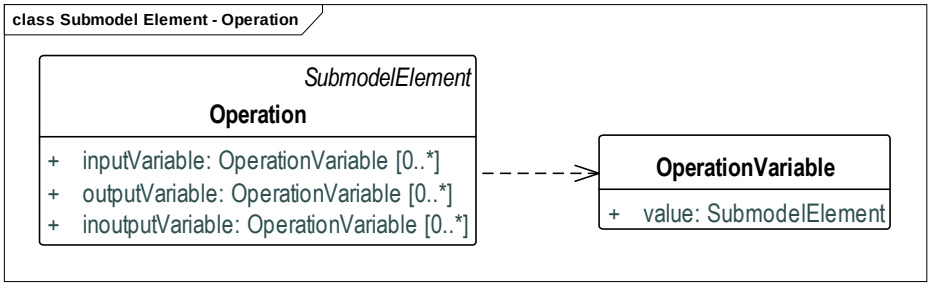


Class:	RelationshipElement			
Explanation:	A relationship element is used to define a relationship between two referable elements.			
Inherits from:	SubmodelElement			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
first*	First element in the relationship taking the role of the subject.	Referable	ref*	1
second*	Second element in the relationship taking the role of the object.	Referable	ref*	1

Class:	AnnotatedRelationshipElement			
Explanation:	An annotated relationship element is a relationship element that can be annotated with additional data elements.			
Inherits from:	RelationshipElement			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
annotation	Annotations that hold for the relationships between the two elements.	DataElement	ref*	0..*

4.7.15 Operation Attributes

Figure 34 Metamodel of Operations



Class:	Operation			
Explanation:	An operation is a submodel element with input and output variables.			
Inherits from:	SubmodelElement			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
inputVariable	Input parameter of the operation.	OperationVariable	aggr	0..*
outputVariable	Output parameter of the operation.	OperationVariable	aggr	0..*
inoutputVariable	Parameter that is input and output of the operation.	OperationVariable	aggr	0..*

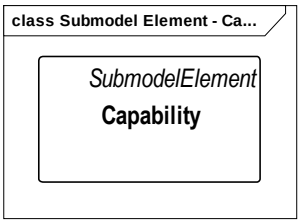
Class:	OperationVariable			
Explanation:	An operation variable is a submodel element that is used as input or output variable of an operation.			
Inherits from:				
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
value*	Describes the needed argument for an operation via a submodel element of kind=Template.	SubmodelElement	aggr	1

Class:	OperationVariable			
	Constraint AASd-008: The submodel element value of an operation variable shall be of kind=Template.			

Note: Operations typically specify the behavior of a component in terms of procedures. Hence, operations enable the specification of services with procedure-based interactions [32].

4.7.16 Capability Attributes

Figure 35 Metamodel for Capabilities



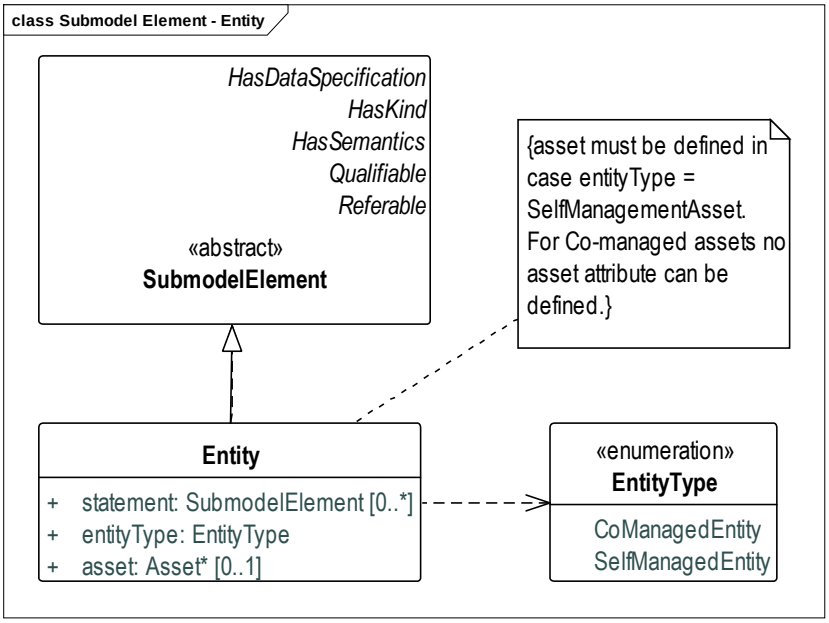
Class:	Capability			
Explanation:	A capability is the implementation-independent description of the potential of an asset to achieve a certain effect in the physical or virtual world.			
--:	SubmodelElement			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.

Note: The *semanticId* of a capability is typically an ontology. Thus reasoning on capabilities is enabled.

For information and examples how to apply the concept of capability and how to map it to one or more skills implementing the capability please refer to [36]. The mapping is done via a relationship element with the corresponding semantics. A skill is typically a property or an operation. In more complex cases the mapping can also be a collection or a complete submodel.

4.7.17 Entity Attributes

Figure 36 Metamodel of Entities



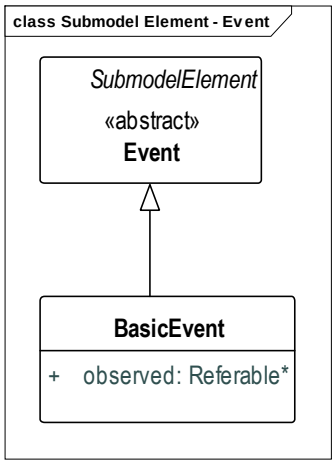
Class:	Entity			
Explanation:	An entity is a submodel element that is used to model entities.			
Inherits from:	SubmodelElement			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
statement	Describes statements applicable to the entity by a set of submodel elements, typically with a qualified value.	SubmodelElement	aggr	0..*
entityType*	Describes whether the entity is a co-managed entity or a self-managed entity.	EntityTypeEnum	attr	1
asset	Reference to the asset the entity is representing. Constraint AASd-014: The asset attribute must be set if entityType is set to "SelfManagedEntity". It is empty otherwise.	Asset	ref*	0..1

Enumeration:	EntityTypeEnum
Explanation:	Enumeration for denoting whether an entity is a self-managed entity or a co-managed entity.
Inherits from:	--
Literal	Explanation

Enumeration:	EntityTypeEnum
Explanation:	Enumeration for denoting whether an entity is a self-managed entity or a co-managed entity.
CoManagedEntity	For co-managed entities there is no separate AAS. Co-managed entities need to be part of a self-managed entity.
SelfManagedEntity	Self-Managed Entities have their own AAS but can be part of the bill of material of a composite self-managed entity. The asset of an I4.0 Component is a self-managed entity per definition.

4.7.18 Event Attributes

Figure 37 Metamodel for Event and MetaEvent

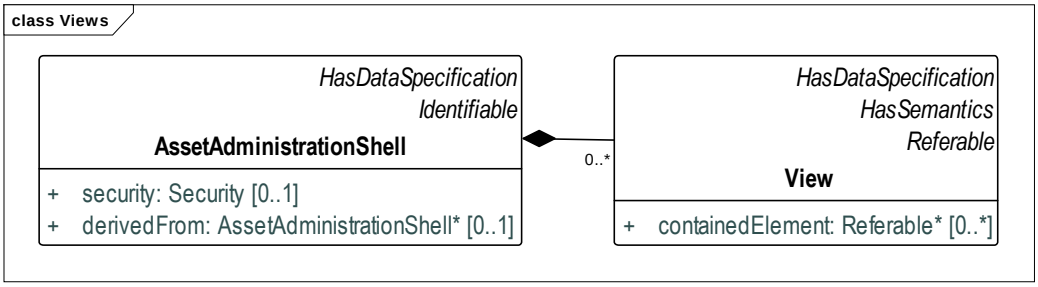


Class:	Event <<abstract>>			
Explanation:	A event			
Inherits from:	SubmodelElement			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.

Class:	BasicEvent			
Explanation:	A basic event			
Inherits from:	Event			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
observed*	Reference to the data or other elements that are being observed.	Referable	ref*	1

4.7.19 View Attributes

Figure 38 Metamodel of Views



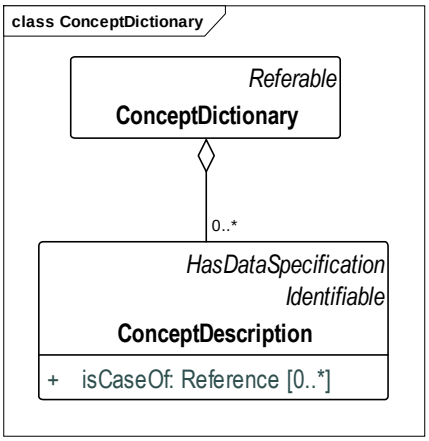
The large number of submodel elements within a submodel can be filtered by views, so that different user groups can only see relevant elements.

Note: Views are a projection of submodel elements for a given perspective. They are not equivalent to submodels.

Class:	View			
Explanation:	A view is a collection of referable elements w.r.t. to a specific viewpoint of one or more stakeholders.			
Inherits from:	HasDataSpecification; Referable; HasSemantics			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
containedElement	Referable elements that are contained in the view.	Referable	ref*	0..*

4.7.20 Concept Dictionary and Concept Description Attributes

Figure 39 Metamodel of Concept Dictionary and Concept Descriptions



Class:	ConceptDictionary
Explanation:	A dictionary contains elements that can be reused. The concept dictionary contains concept descriptions. Typically a concept description dictionary of an AAS contains only concept descriptions of elements used within submodels of the AAS.
Inherits from:	Referable

Class:	ConceptDictionary			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
conceptDescription	Concept description defines a concept.	ConceptDescription	ref	0..*

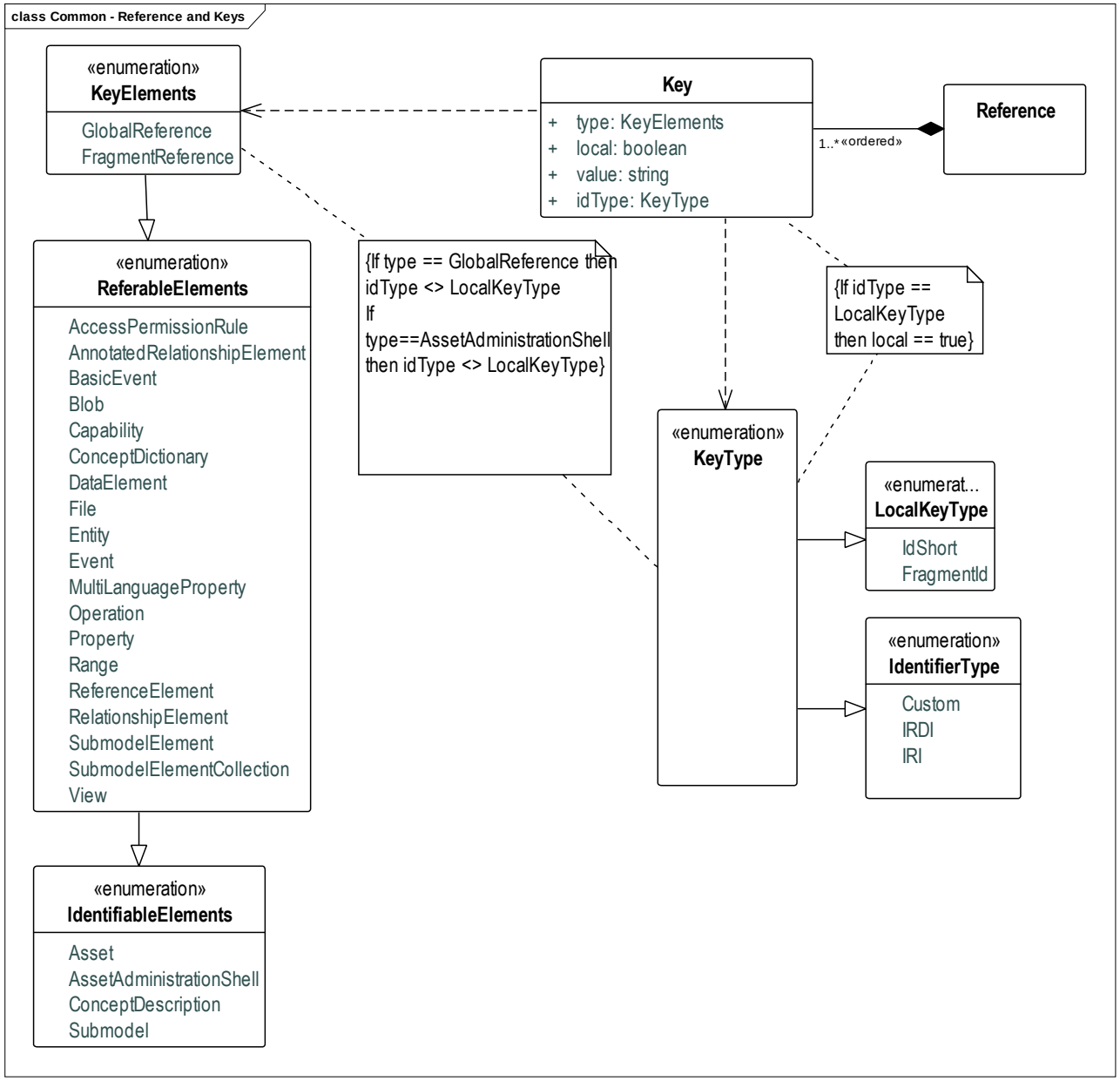
Class:	ConceptDescription			
Explanation:	The semantics of a property or other elements that may have a semantic description is defined by a concept description. The description of the concept should follow a standardized schema (realized as data specification template).			
Inherits from:	HasDataSpecification; Identifiable;			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
isCaseOf	Global reference to an external definition the concept is compatible to or was derived from. Note: Compare to is-case-of relationship in ISO 13584-32 & IEC EN 61360	Reference	aggr	0..*

Different types of submodel elements require different attributes for describing the semantics of them. This is why a concept description has at least one data specification template associated with it. Within this template the attributes needed to define the semantics are defined.

See Clause 4.8 for predefined data specification templates to be used.

4.7.21 Referencing in Asset Administration Shells

Figure 40 Metamodel for References and Keys



Note: References are used throughout the metamodel although not directly visible.

If an element is not a part of an element but just references it, this is denoted by an * behind the Type.

E.g. asset: Asset* means that asset: Reference with Key.type=Asset for the last Key in the Reference

Class:	Reference
Explanation:	Reference to either a model element of the same or another AAs or to an external entity. A reference is an ordered list of keys, each key referencing an element. The complete list of keys may for example be concatenated to a path that then gives unique access to an element or entity.
Inherits from:	--

Class:	Reference			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
key*	Unique reference in its name space.	Key	attr	1..*

Class:	Key			
Explanation:	A key is a reference to an element by its id.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
type*	Denote which kind of entity is referenced. In case *type = GlobalReference* then the element is a global unique id. In all other cases the key references a model element of the same or of another AAS. The name of the model element is explicitly listed.	KeyElements	attr	1
local*	Denotes if the key references a model element of the same AAS (=true) or not (=false). In case of local = false the key may reference a model element of another AAS or an entity outside any AAS that has a global unique id.	boolean	attr	1
value*	The key value, for example an IRDI if the idType=IRDI.	string	attr	1
idType*	Type of the key value. In case of idType = idShort local shall be true. In case type=GlobalReference idType shall not be IdShort.	KeyType	attr	1

Enumeration:	KeyElements
Explanation:	Enumeration of different key value types within a key.
Set of:	ReferableElements
Literal	Explanation
GlobalReference	reference to an element not belonging to an asset administration shell
FragmentReference	unique reference to an element within a file. The file itself is assumed to be part of an asset administration shell.

Enumeration:	ReferableElements
Explanation:	Enumeration of all referable elements within an asset administration shell
Set of:	IdentifiableElements
Literal	Explanation
AccessPermissionRule	Access Permission Rule
AnnotatedRelationshipElement	Annotated relationship element
BasicEvent	Basic Event
Blob	Blob
Capability	Capability
ConceptDictionary	Concept Dictionary
DataElement	Data Element. *Note: Data Element is abstract, i.e. if a key uses "DataElement" the reference may be a Property, a File etc.*
Entity	Entity
Event	Event *Note: Event is abstract*
File	File
MultiLanguageProperty	Property with a value that can be provided in multiple languages
Operation	Operation
Property	Property
Range	Range with min and max
ReferenceElement	Reference
RelationshipElement	Relationship
SubmodelElement	Submodel Element *Note: Submodel Element is abstract, i.e. if a key uses "SubmodelElement" the reference may be a Property, a SubmodelElementCollection, an Operation etc.*
SubmodelElementCollection	Collection of Submodel Elements
View	View

Enumeration:	IdentifiableElements
Explanation:	Enumeration of all identifiable elements within an asset administration shell that are not identifiable
Set of:	--
Literal	Explanation
Asset	Asset
AssetAdministrationShell	Asset Administration Shell
ConceptDescription	Concept Description
Submodel	Submodel

Enumeration:	KeyType
Explanation:	Enumeration of different key value types within a key.
Set of:	LocalKeyType, IdentifierType
Literal	Explanation

Enumeration:	LocalKeyType
Explanation:	Enumeration of different key value types within a key.
Literal	Explanation
IdShort	idShort of a referable element
FragementId	Identifier of a fragment within a file

IdentifierType is defined in Clause 4.7.2.2.

4.7.22 Data Types

4.7.22.1 Predefined Basic Data Types

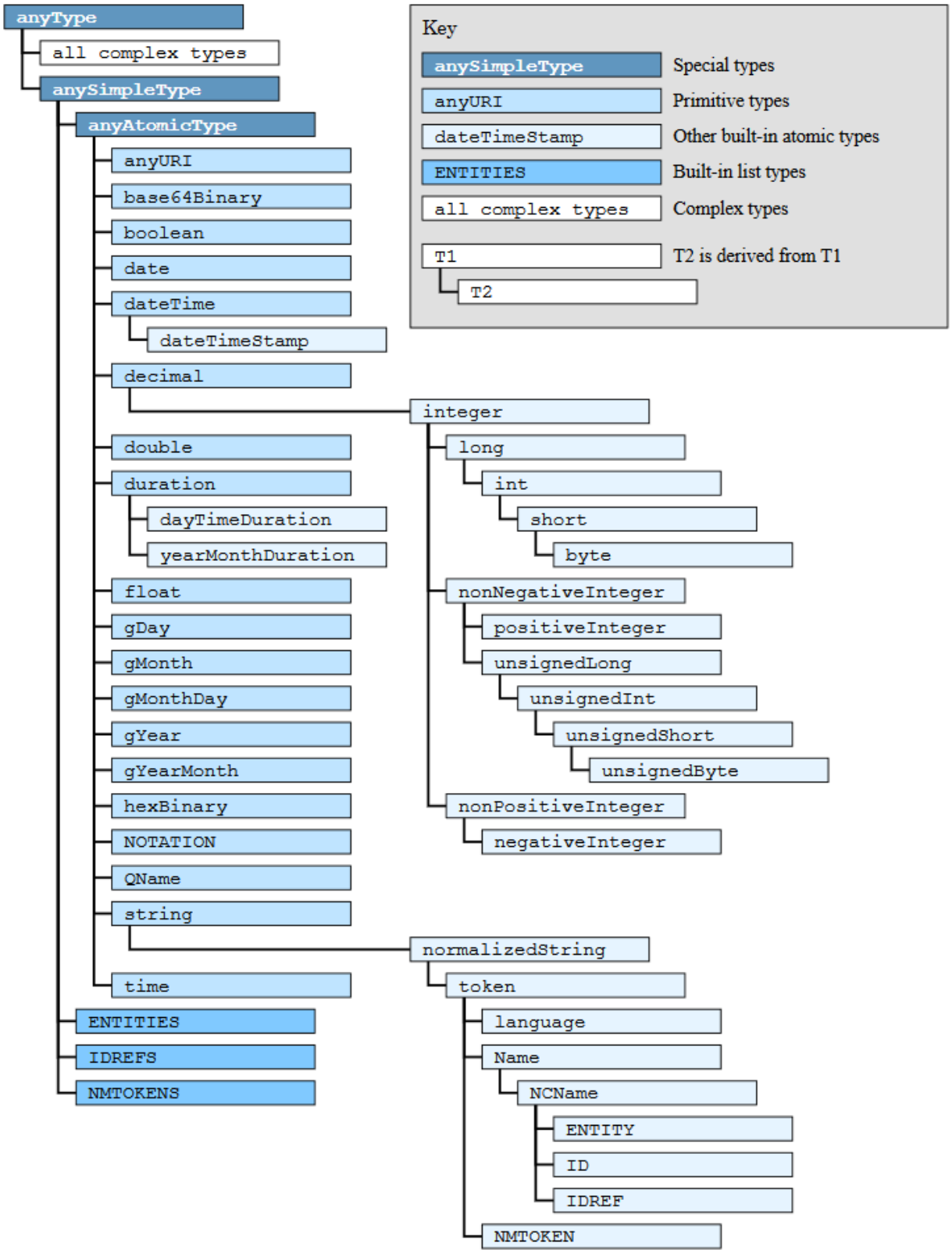
The predefined types used to define the metamodel use the names and the semantics of XML Schema Definition (XSD)[10]. Additionally the type “langString” with the semantics as defined in the Resource Description Framework (RDF)[11] is used. “langString” is a string value tagged with a language code.

These types are also used to denote the type of a value and is defined in the type anySimpleTypeDef (see Clause 4.7.22.2).

[10] see: https://www.w3.org/XML/Schema

[11] see: https://www.w3.org/TR/rdf11-concepts/

Figure 41 Built-In Types of XML Schema Definition 1.1 (XSD)



4.7.22.2 Data Types

Types that are used for specific data specification templates are listed in the corresponding clause of the data specification.

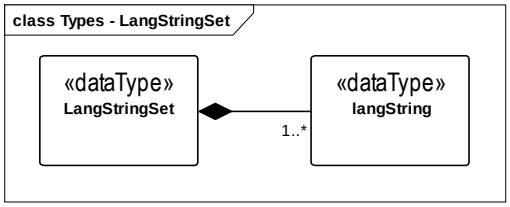
The meaning and format of xsd types is specified in https://www.w3.org/XML/Schema.

Table 5 lists additional data types used in the metamodel. Figure 42 specifies LangStringSet as set of elements of rdf type “langString”.

Table 5 Basic Types used in Metamodel

Type	Basic Type
DataTypeDef	string Note: any xsd simple type as string
BlobType	byte[0..*]
MimeType	string Note: any mimetype as in RFC2046.
PathType	string
QualifierType	string
ValueDataType	any xsd atomic type

Figure 42 Data Type LangStringSet



Class:	LangStringSet <<DataType>>			
Explanation:	A set of strings, each annotated by the language of the string. The meaning of the string in each language shall be the same.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
langString	A string in a specified language.	langString	aggr	1..*

4.7.23 Templates, Inheritance, Qualifiers and Categories

On a first glance there seem to be some overlapping between the concept of data specification templates, inheritance, qualifiers and categories. In this clause the commonalities and differences are explained and hints for good practices are given.

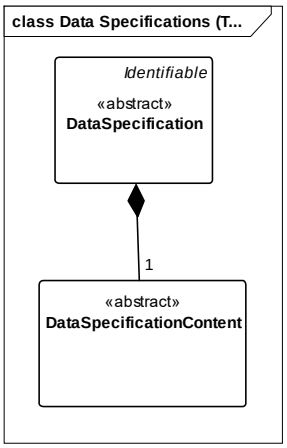
In general extension of the metamodel by inheritance is allowed. As an alternative also templates might be used.

- Templates should only be used if different instances of the class follow different schemas and the templates for the schemas are not known at design time. Templates might also be used if the overall metamodel is not yet stable enough or a tool does support templates but not (yet) the complete metamodel.
- However: when using non-standardized proprietary data specification templates interoperability cannot be ensured and thus should be avoided whenever possible.
- In case all instances of a class follow the same schema then inheritance and/or categories should be used.
- Categories can be used if all instances of a class follow the same schema but have different constraints depending on its category. Such a constraint might specify that an optional attribute is mandatory for this category (like for example the unit that is mandatory for properties representing physical values). Realizing the same via inheritance would lead to multiple inheritance what is to be omitted.
- Qualifiers are used if the semantics of the element is the same independent of its qualifiers. It is only the quality or the meaning of the value for the element that differs.

4.8 Predefined Data Specification Templates

4.8.1 Concept of Data Specification Templates

Figure 43 Concept of Data Specification Templates



Note: The Data Specification Templates do not belong to the metamodel of the Asset Administration Shell. In serializations that choose specific templates the corresponding data specification content may be directly incorporated.

It is required that a data specification template has a global unique id so that is can be referenced via *HasDataSpecification/dataSpecification*.

A template consists of the *DataSpecificationContent* containing the additional attributes to be added to the element instance that references the data specification template and meta information about the template itself (this is why *DataSpecification* inherits from *Identifiable*). In UML these are two separated classes.

4.8.2 Predefined Templates for Property and Value Descriptions

Conformant to the rules in Clause 5.2.2 the following data specification template should be referenced via the id

"http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360/2/0"

(in *hasDataSpecification/dataSpecification*).

This namespace has the qualifier "IEC:" Examples: *IEC:DataSpecificationIEC61360/preferredName* or *IEC: DataSpecificationIEC61360/levelType/Min* or *IEC:LevelType/Min*

Note: The data specification template is not identical to the data specification content as shown in Figure 44.

Figure 44 Data Specification Template for defining Property Descriptions conformant to IEC 61360

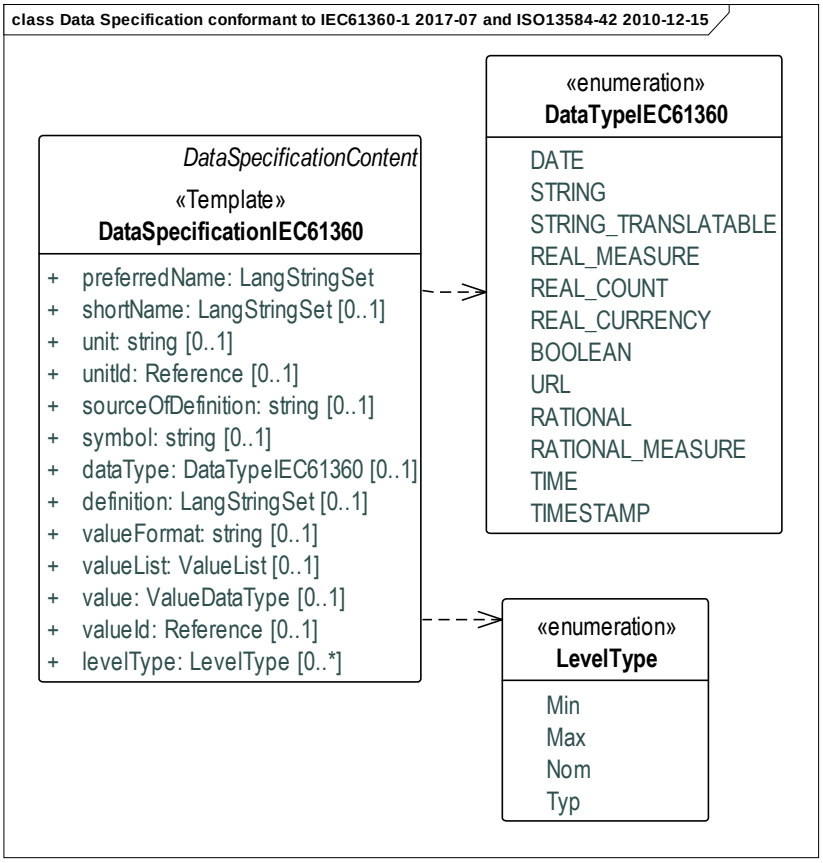


Figure 45 Example Property from eCl@ss

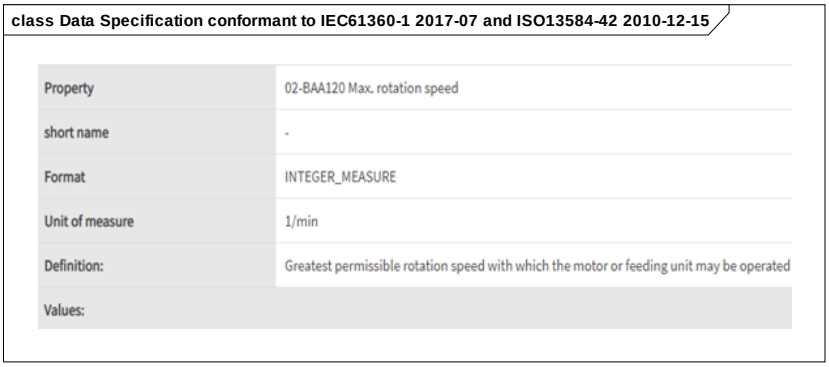


Figure 46 Example Property Description with Value List from eCl@ss

Property	02-BAE122 Cooling type
short name	-
Format	STRING
Definition:	Summary of various types of cooling, for use as search criteria that limit a selection
Values:	

0173-1#07-BAB649#001 - Air-air heat exchanger
0173-1#07-BAB650#001 - Air-water heat exchanger
0173-1#07-BAB592#001 - alien
0173-1#07-BAB611#001 - closed, external air-cooling
0173-1#07-BAB610#001 - closed, internal air-cooling
0173-1#07-BAB591#003 - free cooling
0173-1#07-BAB702#003 - Heat exchanger against other cooling medium
0173-1#07-BAB657#003 - open circuit, external cooling
0173-1#07-BAB656#003 - open circuit, internal cooling
0173-1#07-BAB535#003 - other form of cooling with primary air coolant
0173-1#07-BAB536#003 - other primary non-air coolant
0173-1#07-BAB674#003 - self

Figure 47 Example Value Description from eCl@ss

Value	0173-1#07-BAB657#003
Classification	open circuit, external cooling
short name	
Definition:	

Figure 48 Example Value Desription from eCl@ss Advanced

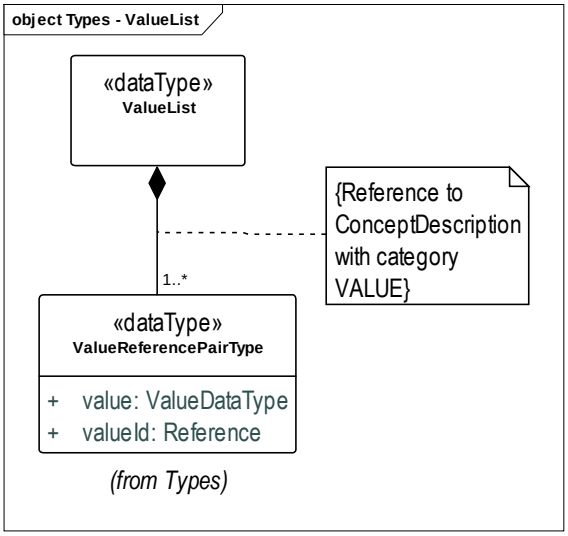
Change Text | Replace | Delete | Copy

General | Admin | Attribute | CR | History | Release

Value	BAB657
IRDI	0173-1#07-BAB657#003
eCl@ss v5 ID	BAB657001
Preferred Name	open circuit, external cooling
Short Name	
Definition	
Source of Definition	
Note to Definition	
Data Type	String
Value specification	Coded values
Exception	No

The Type "*ValueList*" lists all the allowed values for a concept description for which the allowed values are listed in an enumeration. The value list is a set of value reference pairs.

Figure 49 Type "ValueList"



Class:	ValueList <<DataType>>			
Explanation:	A set of value reference pairs.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
valueReferencePairType	A pair of a value together with its global unique id.	ValueReferencePair	aggr	1..*

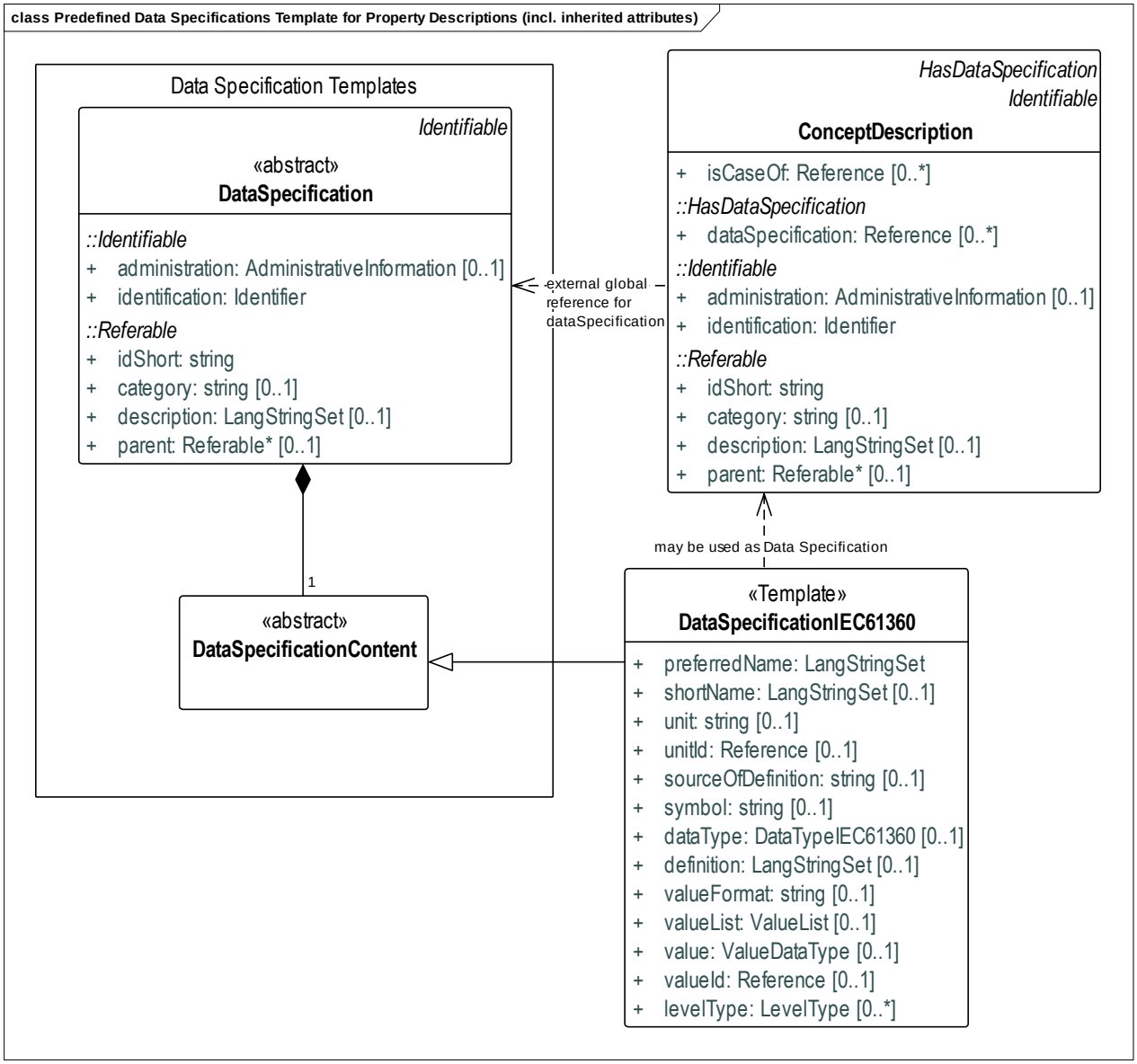
Class:	ValueReferencePair <<DataType>>			
Explanation:	A value reference pair within a value list. Each value has a global unique id defining its semantic.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
value*	the value of the referenced concept definition of the value in *valueId*.	ValueDataType	aggr	1
valueId*	Global unique id of the value.	Reference	ref	1

For the meaning of the content attributes of the IEC61360 data specification template please refer to IEC 61360 and/or eCl@ss.

The template can be used to describe properties and values, both.

See Figure 50 for how data specification templates are related to concept descriptions (showing all inherited attributes as well). In a similar way templates for other elements in the information model can be defined and used.

Figure 50 Concept Descriptions for Properties conformant to IEC61360



The following tables recommend to use specific categories to distinguish which kind of concept is described. They also give advice which attributes need to be filled for which category of concept description. These tables are not part of the specification because in a way the existing template for describing concept descriptions for properties and coded values is misused to also describe other concepts. In later version of the standards and this specification there might be concept descriptions better suited for the purpose.

Attribute[12]	Property	Property	Property	Multi-Language-Property	Range
Category of Concept Description	VALUE	PROPERTY	PROPERTY	PROPERTY	PROPERTY
Category of SubmodelElement	**CONSTANT**	**VARIABLE**	**PARAMTER**	--	--
preferredName[13]	m	m	m	m	m
shortName	(m)	(m)	(m)	(m)	(m)
unit	(m)	(m)	(m)	--	(m)
unitId	(m)	(m)	(m)	--	(m)
sourceOfDefinition	o	o	o	o	o
symbol	o	o	o	--	--
dataType	m[14]	m[15]	m[16]	STRING_-TRANSLATABLE	REAL_* or RATIONAL_*
definition	(m)	m	m	m	m
valueFormat	o	o	o	--	o
valueList	--	o	o	--	--
value	m	--	--	--	--
valueId	o	--	--	--	--
levelType	--	--	--	--	{Min, Max}

Table 6 Concept Description with IEC612360 Data Specification Template for Properties and Ranges

[12] m= mandatory, o = optional, (m) = conditionally mandatory or recommended to be added

[13] Mandatory in at least one language. Preferrable an English preferred name should allways be defined.

[14] Except STRING_TRANSLATABLE. In this case MultiLanaguageProperty shall be used.

[15] Except STRING_TRANSLATABLE. In this case MultiLanaguageProperty shall be used.

[16] Except STRING_TRANSLATABLE. In this case MultiLanaguageProperty shall be used.

Attribute[17]	Reference-Element	File[18]	Blob[18]	Capability[18]
Category of Concept Description	REFERENCE	DOCUMENT	DOCUMENT	CAPABILITY
Category of Submodel-Element	--	--	--	--
preferredName[19]	m	m	m	m
shortName	(m)	(m)	(m)	(m)
unit	--	--	--	--
unitId	--	--	--	--
sourceOf-Definition	o	o	o	o
symbol	--	--	--	--
dataType	STRING	URL	STRING	--
definition	m	m	m	m
valueFormat	--	--	--	--
valueList	--	--	--	--
value	--	--	--	--
valueId	--	--	--	--
levelType	--	--	--	--

[17] m= mandatory, o = optional, (m) = conditionally mandatory or recommended to be added

[18] Template only used until explicit template for defining the corresponding types of elements are available.

[19] Mandatory in at least one language. Preferrable an English preferred name should allways be defined.

Table 7 Concept Description with IEC612360 Data Specification Template for other Data Elements and Capabilities

Attribute	SubmodelElement-Collection[18][18]	RelationElement[18]	Operation[18]	Event	Entity
Category of Concept Description	COLLECTION	REFERENCE	FUNCTION	EVENT	ENTITY
Category of Submodel-Element	--	--	--	--	--
preferredName[20]	m	m	m	m	m
shortName	(m)	(m)	(m)	(m)	(m)
unit	--	--	--	--	--
unitId	--	--	--	--	--
sourceOfDefinition	o	o	o	o	o
symbol	--	--	--	--	--
dataType	--	--	--	--	--
definition	m	m	m	m	m
valueFormat	--	--	--	--	--
valueList	--	--	--	--	--
value	--	--	--	--	--
valueId	--	--	--	--	--
levelType	--	--	--	--	--

Table 8 Concept Description with IEC612360 Data Specification Template for other Submodel Elements Data

[20] Mandatory in at least one language. Preferrable an English preferred name should allways be defined.

Attribute[17]	Submodel	Qualifier	View
category	APPLICATION_CLASS	QUALIFIER	VIEW
preferredName	m	m	m
shortName	(m)	(m)	(m)
unit	--	--	--
unitId	--	--	--
sourceOfDefinition	o	o	o
symbol	--	--	--
dataType	--	m	--
definition	m	m	m
valueFormat	--	o	--
valueList	--	o	--
value	--	--	--
valueId	--	--	--
levelType	--	--	--

Table 9 Other Elements with semanticId

4.8.3 Predefined Templates for Unit Concept Descriptions

Conformant to the rules in Clause 5.2.2 the following data specification template should be referenced via the id "http://admin-shell.io/DataSpecificationTemplates/DataSpecificationTemplates/DataSpecificationPhysicalUnit/1/0" (in *hasDataSpecification/dataSpecification*).

The recommendation is to use "IEC:" as namespace qualifier as already discussed in Clause 4.8.2.

Examples: *IEC:DataSpecificationPhysicalUnit/unitName* or *IEC:DataSpecificationPhysicalUnit/definition*

Units are used in data specification templates for properties when defining the *unitId* (*IEC:/DataSpecificationIEC61250/unitId*, see Figure 51). The unit value corresponds then to the *unitName* as specified in the concept description referenced via *unitId*.

The data specification template for concept descriptions for units (see Figure 51) is defined conformant to IEC61360-1 and ISO13854-42 and is following the xml schema UnitML. An example unit is shown in Figure 52.

Figure 51 Data Specification Tempate for Physical Units (DataSpecificationPhysicalUnit) and its Usage

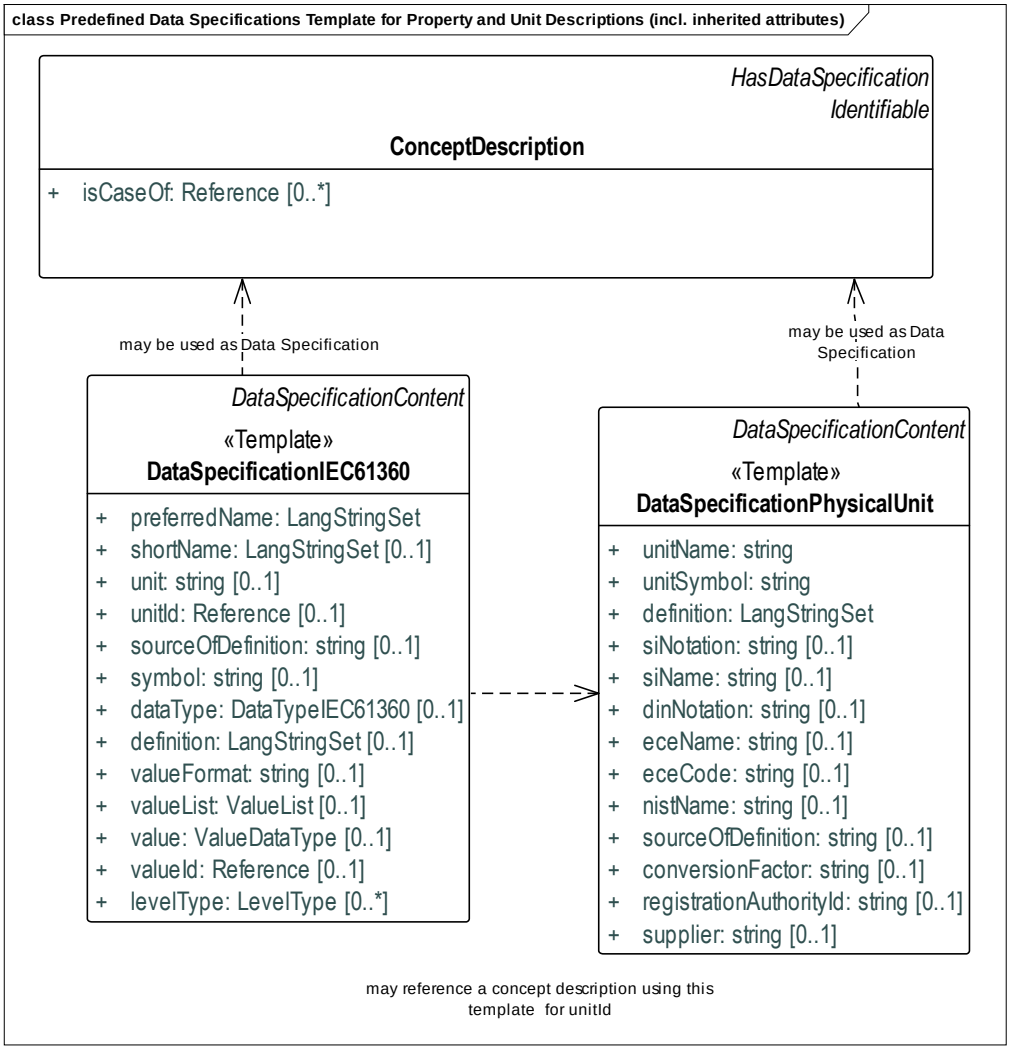


Figure 52 Example of a concept description for a unit: 1/min (from eCl@ss)

class Data Specification Units conformant to IEC61360-1 and ISO13584-42 (NEW)

ID	0173-1#05-AAA650#002
Name	1/min
Primary language	German
Structured Naming	min^{-1}
Short Name	1/min
Definition	reciprocal of the unit minute
Source	NIST Special Publication 811:1995
Comment	
SI Notation	1/min
SI-Name	reciprocal minute
DIN Notation	min^{-1}
ECE Name	reciprocal minute
ECE-Code	C94
NIST-Name	revolutions per minute
Conversion factor	1.0/60
Registration authority ID	0173-1
Supplier	ECL

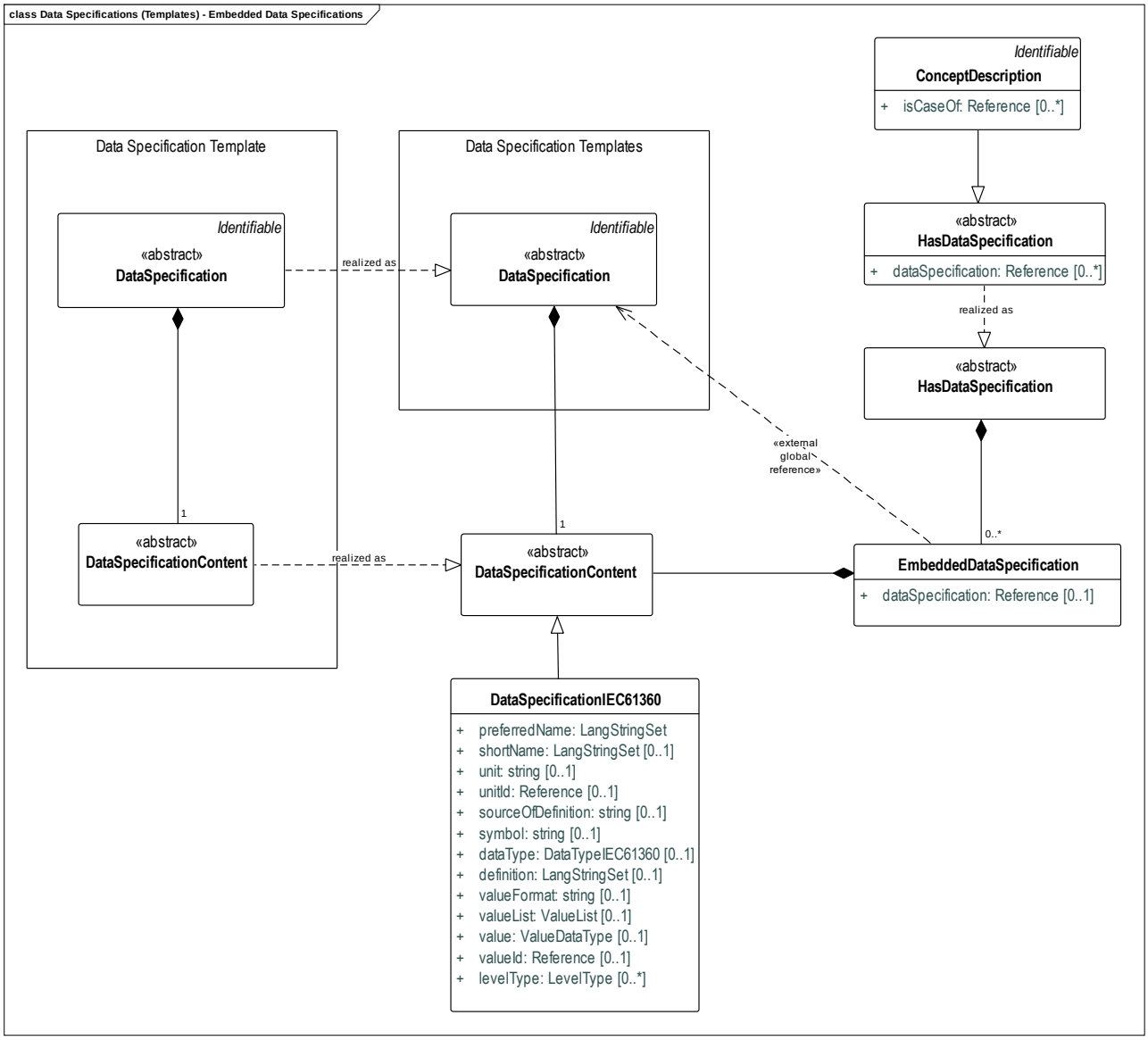
4.8.4 Embedded Data Specifications

This specification predefines data specifications that can be used within an AAS to ensure interoperability.

Thus, some serializations or mapping support exactly those data specifications defined in this specification and no others although the metamodel as such is more flexible and would also support proprietary data specifications.

In this case of restricted data specifications to be used the notation is that of “embedded data specifications”. Figure 53 the realization is explained: instead of a set of external global references to externally defined data specifications a set of pairs consisting of an external global reference to a data specification as well as the data specification content itself is directly “embedded”. In this realization the data specification content belongs to the schema etc. whereas in the general concept the data specification including its content are not part of the schema. This is similar to the concept of semanticIds: either it is an external global reference to an external concept dictionary or it is a reference to a concept descriptions within the schema. However, for semanticId we only allow exactly one reference, whereas for data specifications a set of data specifications references is allowed.

Figure 53 Realization of Embedded Data Specifications



5 Mappings to Data Formats to Share I4.0-Compliant Information

5.1 General

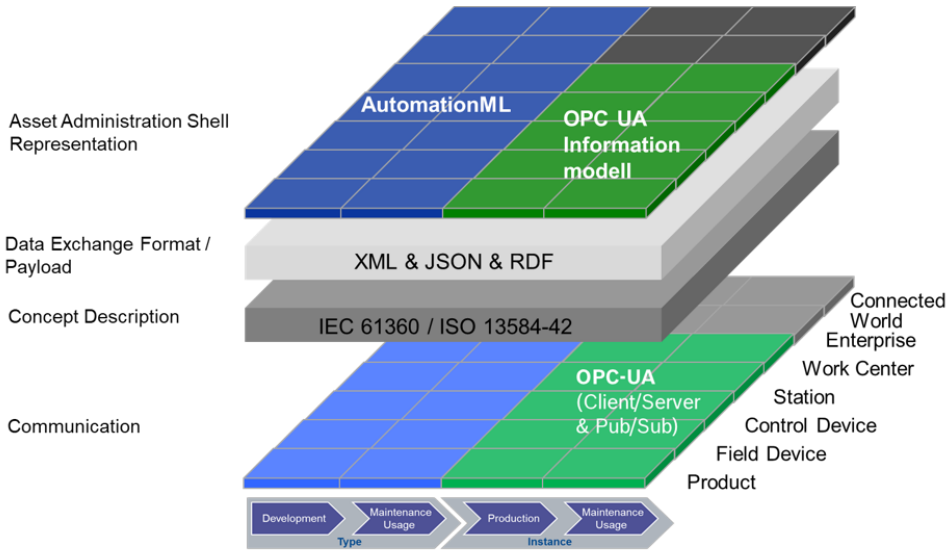
It should be possible to share information between different systems throughout the area covered by the entire RAMI4.0 model [1] [2]. OPC UA has been targeted as a format for information models in the domain of production operations, but there is a need for other formats for the other areas and the interrelationships between them.

This document describes the AssetAdministration Shell together with its submodels in different data formats[21]:

Table 10 Distinction of different data format for the AAS

Data format	Purpose / motivation
OPC UA Information models	Access to all information of the administration data and sharing of live data within production operations. Access for higher-level factory systems to this information.
AutomationML	Sharing of type and instance information about assets, particularly during engineering. Transfer of this information into the operational phase (cf. OPC UA and the corresponding mapping)
XML, JSON	Serialisation of this information for the purpose of technical communication between phases.
RDF	Mapping of this information to enable full use of the advantages of semantic technologies.

Figure 54 Graphic View on Exchange Data Formats for the Asset Administration Shell[22]



Source: Bosch Rexroth AG. Plattform Industrie 4.0

The specifications of the preceding clause are now specifically transferred to the individual data exchange formats.

5.2 General Rules

In the following we distinguish between global and model keys. They are defined as follows:

- A **global key** is a key with idType <> IdShort. A global key can be local (local = true) if it references an element within the same AAS, for example a ConceptDescription or another Submodel.
- A **model key** is a key with type <> GlobalReference, i.e. it references a model element within the same AAS (local = true) or within another AAS (local=false).
- A similar distinction is done for references:

[21]The abbreviated use of the word “data formats” includes the use of conceptual advantages such as information models, schemes, transmission protocols, etc.

[22] Only data formats considered in this document so far are mentioned in the figure.

- A **model reference** is a reference key chain in which the last key is a model key.
- A **global reference** is a reference key chain in which the last key is a global key with type = GlobalReference.
- An **external global reference** is a global reference for which the first key in the reference key chain is not local (local = false).
- A **local global reference** is a global reference for which the first key in the reference key chain is local (local = true).

The following rules hold and ensure that potential cyclical References can be serialized:

- In a Reference key chain, a key with local “true” is followed either by no key or a key with "local" is "true".
- In a Reference key chain, a key with local “false” is followed either by no key or a key with "local" is "true"

5.2.1 Serialization of Values of Type “Reference”

In some mapping or serializations the Type “Reference” is converted into a single string. In this case we recommend to use the following serilization:

<Reference> ::= <Key>{,<Key>}*
<Key> ::= (<KeyType>)(<Local>)[<KeyIdType>]<KeyValue>
<KeyType> ::= value of AAS:Key/type
<Local> ::= local | no-local
<KeyIdType> ::= value of AAS:Key/.idType
<KeyValue> ::= value of AAS:Key/value

With <Local> == local if AAS:Key/local = True and no-local if AAS:Key/local == False.

Examples:

(ConceptDescription)(local)[IRDI]0173-1#02-BAA120#008

(GlobalReference)(no-local)[IRDI]0173-1#01-AFZ615#016

(Submodel)(local)[IRI]http://customer.com/demo/aas/1/1/1234859590,(Property)(local)[IdShort]Temperature

5.2.2 Semantic Identifiers for Metamodel and Data Specifications

To enable the unique identification of concepts as used and defined in the metamodel of the asset administration shell rules for creating such identifiers are defined.

The name spaces and their qualifier are defined as follows:

<Namespace> ::= (<AAS Namespace>|<Data Specification Namespace>)

<Namespace Qualifier> ::= <AAS Namespace Qualifier>| Data Specification Qualifier>

<AAS Namespace> ::= <Shell-Namespace>”/aas/”<Version>

<Data Specification Namespace> ::=

<Shell-Namespace>“/DataSpecifications/“<idShort of Data Specification><Version>

<Shell-Namespace> ::= “http://admin-shell.io/”

<Version> ::= <Digit>/<Digit>

<Digit> ::= 1 | 2 | 3 | 4 | 5 | 6 | …

Up to now only one data specification is defined, otherwise there would be a list of possible data specification namespaces. The order is identical to the order of the data specification namespaces. I.e. IEC: maps to http://admin-shell.io/ds/IEC61360.

<AAS Namespace Qualifier> ::= "AAS:"

<Data Specification Qualifier> ::= defined per Data Specification

A concrete unique identifier is defined as follows:

<AAS Unique Concept Identifier> ::= (<Namespace> | <Namespace Qualifier>)"/"<AAS Concept Identifier>

<AAS Concept Identifier> ::= <AAS Class Name>[(<AAS Attribute>|<AAS Enumeration>)]

<AAS Attribute> ::= "/"<AAS Attribute Name>[{"/"<AAS Attribute Name>}*]

<AAS Enumeration> ::= [{"/"<AAS Attribute Name>}*]"/"<AAS Enumeration Value>

Examples for valid unique AAS Concept Identifiers:

http://admin-shell.io/aas/2/0/AssetAdministrationShell/administration/version

AAS:AssetAdministrationShell/administration/version

AAS:/Asset/kind/Instance

The application of the pattern is explained in the following:

The concept identifier of a Class follows the pattern:

<AAS Class name>

This also holds for abstract classes and types including Enumerations.

Examples: *AAS:View, AAS:Submodel, AAS:Qualifier, AAS:Reference, AAS:MimeType, AAS:IdentifierType*

Attributes of Classes are separated by "/". Also inherited attributes can be referenced like this if the concrete referable is important in the context.

Basic Pattern:

<AAS Class name>"/"<AAS Attribute Name>

Examples[23]: *AAS:Referable/idShort* or *AAS:Property/idShort* or *AAS:Asset/assetIdentificationModel* or *AAS:Qualifier/semanticId* or *AAS:Identifier/id*

This also holds for attributes of attributes if the cardinality of the attributes involved is not greater than 1:

<AAS Class Name>"/"<AAS Attribute Name>[{"/"<AAS Attribute Name>}*]

Examples: *AAS:Submodel/identification/id* or *AAS:Identifiable/administration/version*

This also holds for values of enumerations

<AAS Class Name>[{"/"<AAS Attribute Name>}*]["/"<AAS Enumeration Value>]

Examples: *AAS:Submodel/identification/idType/IRDI* or *AAS:Identifiable/identification/idType/IRDI* or *AAS:Identifier/idType/IRDI* or *AAS:IdentifierType/IRDI*

In case of an attribute with cardinalität greater than 1 no further attributes or enumeration values can be added.

[23] For simplicity most examples use the namespace qualifier and not the full path of the namespace.

Note: Although the attribute name in UML is always singular even if the cardinality is > 1 the attribute name is annotated by the plural "s".

Examples: *AAS:Operation/InputVariables* or *AAS:AssetAdministrationShell/submodels* or *AAS:SubmodelElementCollection/submodelElements*

AAS:AssetAdministrationShell/submodels/administration/version or *AAS:Submodel/Property/idShort* are no valid concept identifier.

A concrete example how these rules are applied is given in Annex H.iii: the identifiers are used as values for the *RefSemantic* attribute in AutomationML Mapping of the Asset Administration Shell. These identifiers are also used in OPC UA (Clause 5.7) to describe the semantics of the metamodel via the OPC UA *HasDictionaryEntry* reference type.

For specific serializations and mappings additional identifiers might be needed. For example for a set of asset administration shells or a set of available assets or concept descriptions etc. Here, the AAS metamodel and specification does not give any recommendations.

Data Specifiation Handling is special. Data Specification Templates do not belong to the metamodel of the AAS. However, only the predefined data specification templates as specified in this specification are supported in the serializations. For these the corresponding name space qualifier are defined individually.

Examples: *IEC:DataSpecificationIEC61360/preferredName* (see Clause 4.8.2) or *IEC:DataSpecificationIEC61360/unit (see Clause 4.8.3).*

For the data specification itself the AAS namespace is used: *AAS:DataSpecifciationIEC61360*

In xml and JSON data specifications are embedded into the schema itself using the attribute "embeddedDataSpecification". For these no concept identifier shall be used. I.e.

AAS:ConceptDescription/embeddedDataSpecification

is not a valid concept identifier. *AAS:DataSpecificationContent* is a valid concept identifier.

5.3 Unified example

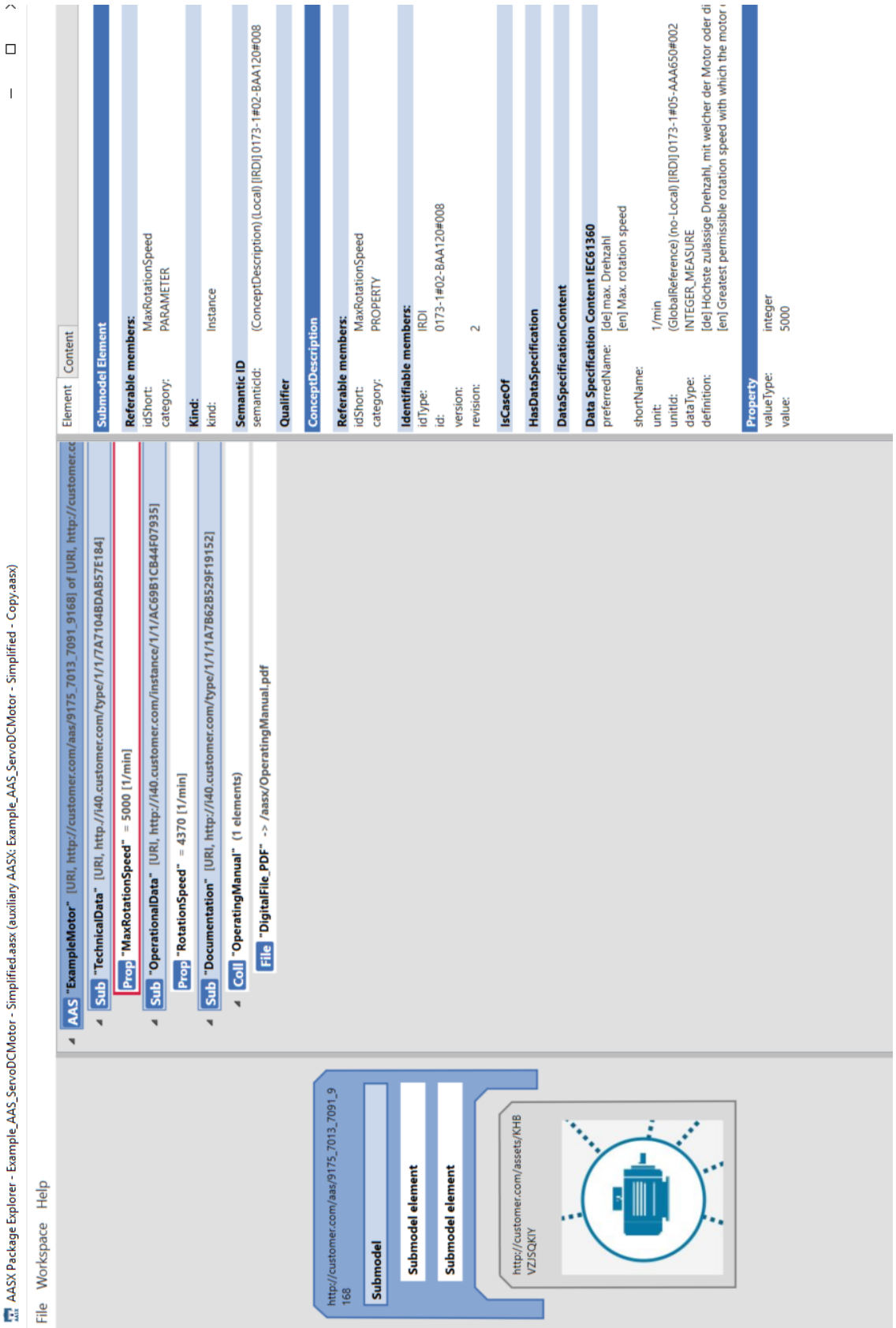
The following example is used to demonstrate the main features of the data formats as explained in the following clauses for different data formats. Intention is to motivate the equivalency of information in different representations. The examples themselves can be found in the annex.

It shows an AAS with three submodels: TechnicalData, OperationalData and Documentation. The asset, an motor, that it is representing has the global ID "http://customer.com/assets/KHBVZJSQKIY".

The TechnicalData submodel contains data that is available at engineering time: the maximum rotation speed measured in 1/min. Its semanticId is 0173-1#02-BAA120#008. It is an eCl@ss IRDI. However, in this example a copy of the eCl@ss entry values are copied to a corresponding concept description with the same IRDI. The unit "1/min" has also a unique id, "0173-1#05-AAA650#002".

The third submodel "Documentation" contains a pdf document, the operating manual.

Figure 55 Unified Example for ExampleMotor



5.4 XML

5.4.1 General

In the following clauses an overview of the main concepts of the AssetAdministration Shell XML serialization is presented. For import and export scenarios the metamodel of an AssetAdministration Shell needs to be serialized. A serialization format is XML. The information is divided in three parts. The first part discusses the rules, in the second part are examples for some specific rules and in the third part the schema and a complete example is shown in the annex.

5.4.2 Introduction

eXtensible Markup Language (XML[24]) is very well suited to deriving information from an IT system, perhaps to process it manually, and then to feed it into another IT system. It therefore meets the needs of the information sharing scenario defined in Section 0. XML provides for the possibilities of scheme definitions which can be used to syntactically validate the represented information in each step. For this reason, this document provides basic scheme definitions to permit a validation of information which is shared.

The XML schema definitions are divided into three different files:

- Core definitions for the AssetAdministration Shell and its export container: aas.xsd
  - Namespace: "http://www.admin-shell.io/aas/2/0"
- IEC61360 datatype definition: iec61360.xsd
  - Namespace: "http://www.admin-shell.io/IEC61360/2/0"
- Attributed based access control definition: aas_abac.xsd
  - Namespace: "http://www.admin-shell.io/aas/abac/2/0"

The namespace reflects the current version (2.0) of the specification.

Subsequently, an example in XML is provided.

5.4.3 Rules

The main concepts of the XML schema and the resulting XML serialization are explained by the following rules. Rules 1 through 6 are general rules, while rules 7 through 11 are specific to References.

(1) XSD global Types are used for modeling

For reusability XSD global types will be used for modeling. There will be a naming convention *<informationModelName>+'_t'*

(2) If present, names are taken from the information model.

For comprehensibility reasons the XML key names should be the same as the representing Element in the metamodel.

(3) All identifiables have an aggregation on root level.

The identifiables are AssetAdministrationShells, Assets, Submodels, ConceptDescriptions. To reduce redundancy instances, they are located exclusively in the top-level aggregation.

(4) Repeating elements and their types will get the same names of their instances in plural.

If the element has a cardinality of n>1 a parent element will be used with the name of the name of the element in plural. For example, each element in the aggregation *assets* needs to be an *asset*.

(5) Identifiables which are not in the top-level aggregations are only references to the corresponding instances in one of the top-level aggregations.

This rule completes the concept of rule 3. There should be no redundant identifiable in the serialized metamodel.

---

[24] see: https://www.w3.org/TR/2008/REC-xml-20081126/

(6) For elements with type LangStringSet an aggregation element called langStringSet_t is added. For the single element a language tag “lang” is added.

For internationalization purposes this rule is necessary.

(7) The attributes of a key in a reference except for the value itself are realized as XML attributes.

(8) Data Specification Templates are directly added to the Concept Description because up to now only property descriptions are supported.

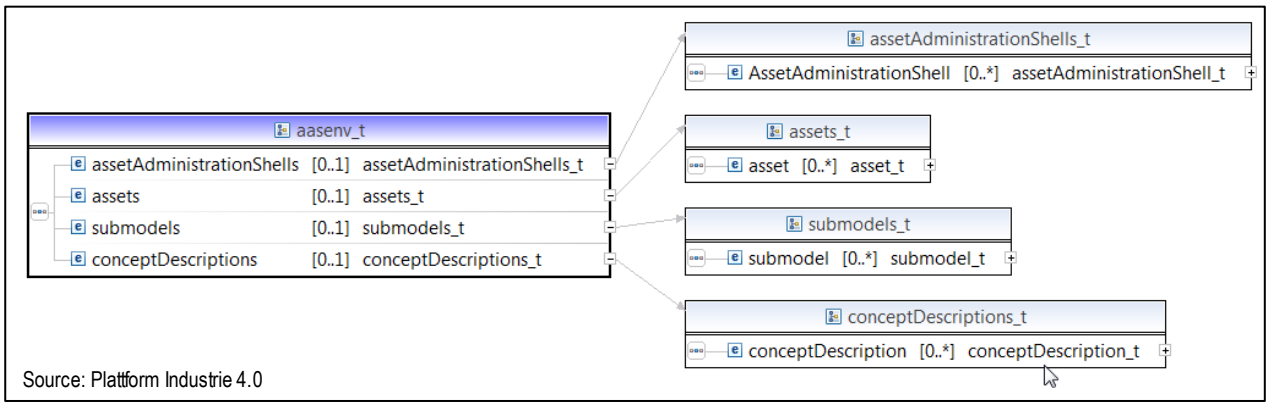
Additionally, a new element EmbeddedDataSpecification is introduced that has two attributes: one for the global reference to the data specification identifier and one for the content of the data specification.

(9) Attribute based access control is added as a separate XML schema that is linked by *AssetAdministrationShell/security*

5.4.4 Example for Top-Level Structures

One serialization describes one asset Administration Shell environment that is a collection of Administration Shells. The root element of the AssetAdministration Shell environment has 4 aggregations. For each identifiable class, one aggregation is featured, as required by rule 3.

Figure 56 Top level structure of an AssetAdministration Shell environment mapped to XML Schema



Note: XSD structuring was done with Eclipse tool chain

The resulting XML is the minimal XML:

Table 11 Minimal XML for top level structure

```
<?xml version="1.0" encoding="UTF-8"?>
<aas:aasenv xmlns:aas="http://www.admin-shell.io/aas/2/0"
xmlns:abac="http://www.admin-shell.io/aas/abac/2/0"
xmlns:aas_common="http://www.admin-shell.io/aas_common/2/0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:IEC="http://www.admin-shell.io/IEC61360/2/0"
xsi:schemaLocation="http://www.admin-shell.io/aas/2/0 AAS.xsd http://www.admin-shell.io/IEC61360/2/0 IEC61360.xsd http://www.admin-shell.io/aas/abac/2/0 AAS_ABAC.xsd">
<aas:assetAdministrationShells></aas:assetAdministrationShells>
<aas:assets></aas:assets>
<aas:submodels></aas:submodels>
```

```
<aas:conceptDescriptions></aas:conceptDescriptions>
</aas:aasenv>
```

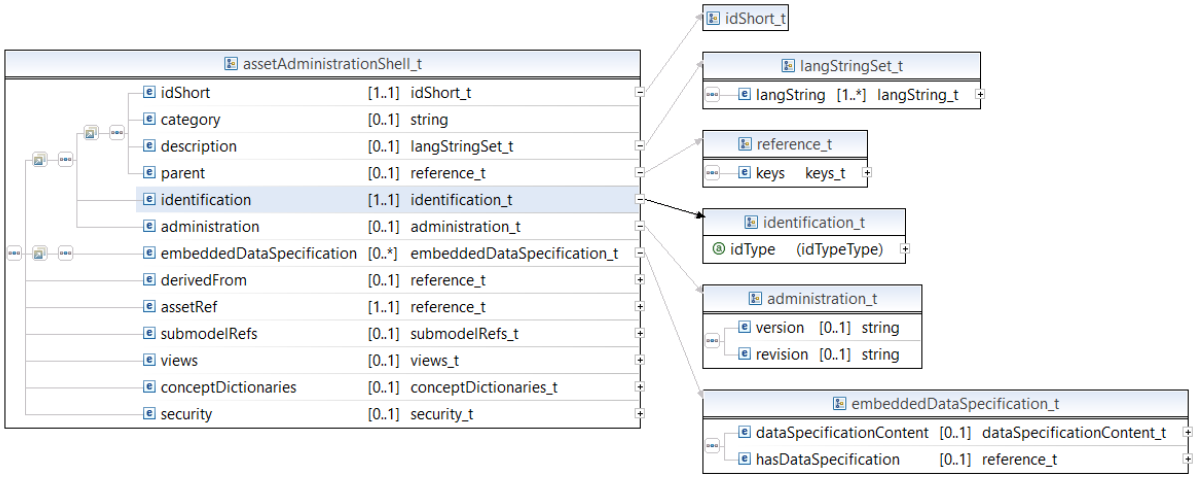
Note: ↙ designates line-wrap for purpose of layout

5.4.5 XSD Model Groups

There are a number of attribute groups in the UML model – i.e. identifiable or hasSemantics. These groups are modelled as XSD model groups so they could be reused as anonymous groups in different places.

Note: Identifier/id is not modelled as attribute but a typical identification looks like this: <aas:identification> idType="IRI">www.admin-shell.io/aas-sample/2/0</aas:identification>

Figure 57 XSD Model Groups



This is realized in the according XSD as follows:

Table 12 Using XSD Model Groups

```
<complexType name="assetAdministrationShell_t">
    <sequence>
        <group ref="aas:identifiable"></group>
        <group ref="aas:hasDataSpecifications"></group>
        <element name="derivedFrom" type="aas:reference_t" minOccurs="0" maxOccurs="1"></element>
        <element name="assetRef" type="aas:reference_t"  minOccurs="1" maxOccurs="1"></element>
        <element name="submodelRefs" type="aas:submodelRefs_t" minOccurs="0" maxOccurs="1"></element>
        <element name="views" type="aas:views_t" minOccurs="0"  maxOccurs="1"></element>
        <element name="conceptDictionaries"  type="aas:conceptDictionaries_t"  minOccurs="0"
maxOccurs="1"></element>
        <element name="security" type="abac:security_t" minOccurs="0" maxOccurs="1"></element>
    </sequence>
```

```
</complexType>
```

Note: due to XSD group mechanism, hasDataSpecification maps to an element of embeddedDataSpecification_t and identifiable maps to multiple elements in Figure 57.

5.4.6 Keys and References

Keys and References (see 4.7.21) are mapped on the same XML schema construct. Some of the attributes have enumerations defined – these are mapped on string constraints.

Figure 58 Keys and References

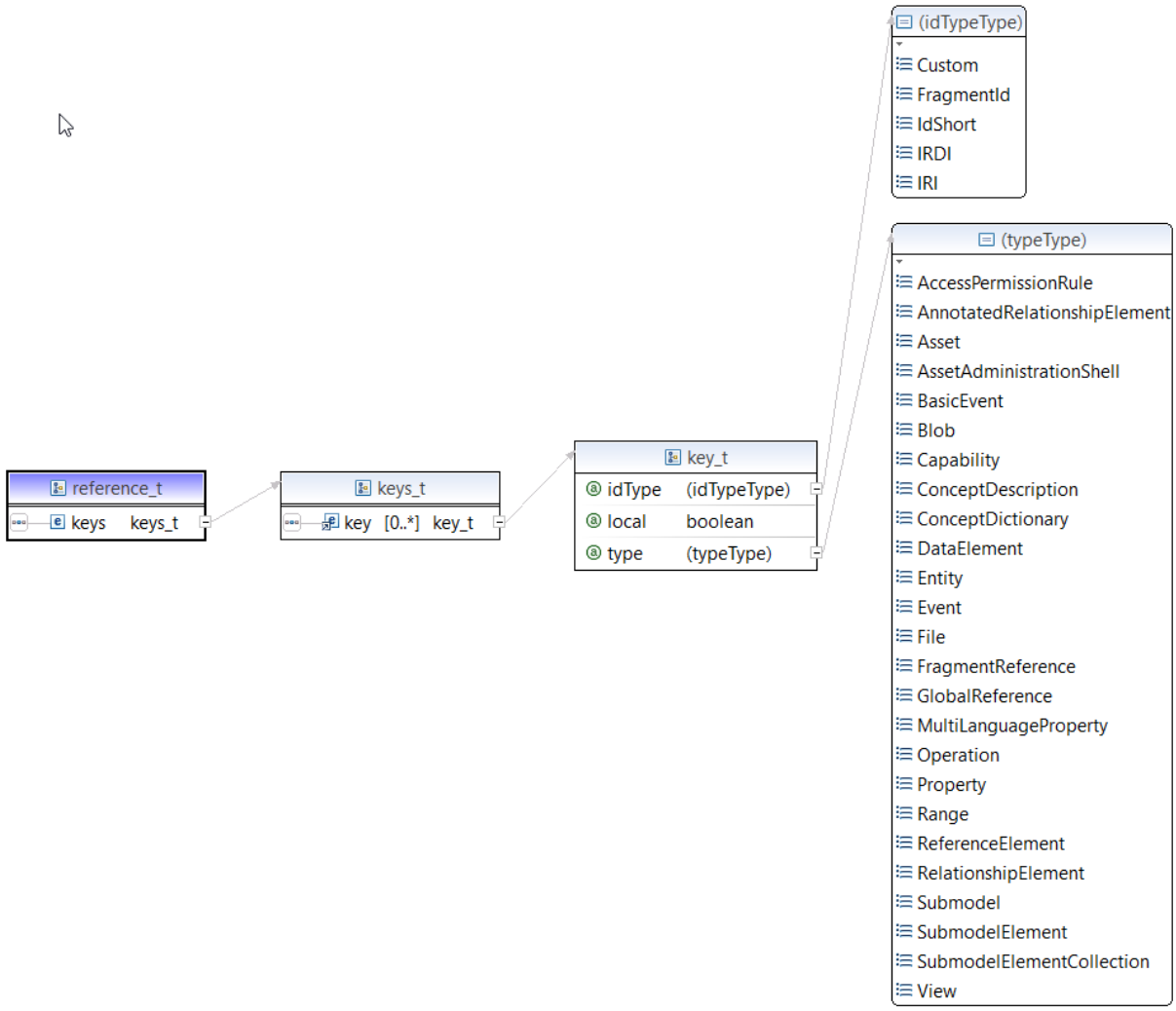
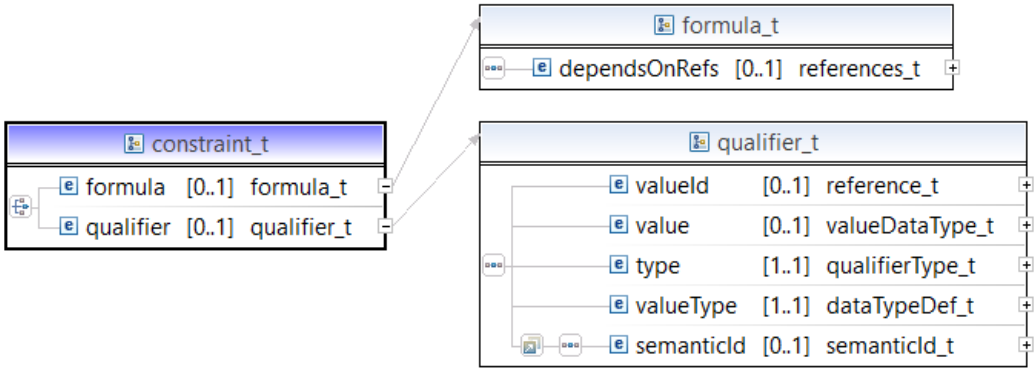


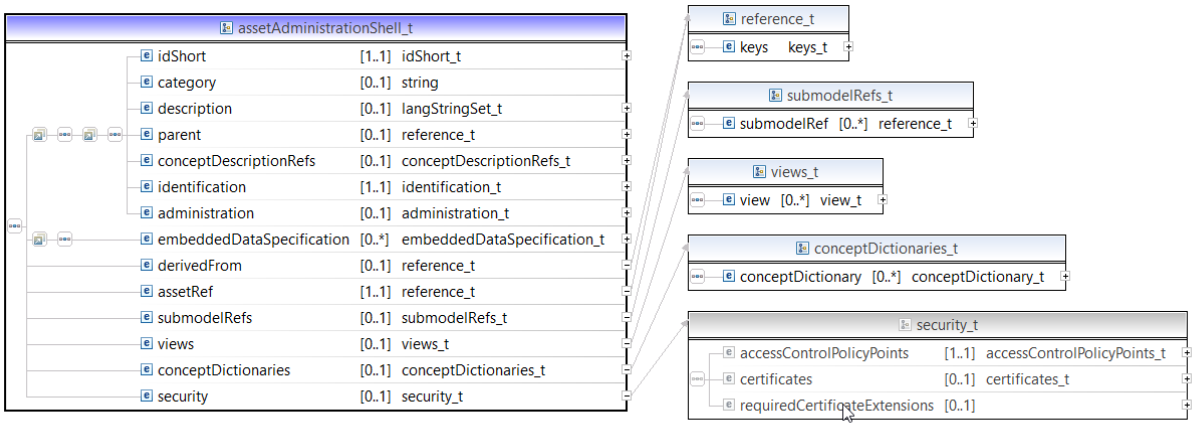
Figure 59 Constraints and Qualifiers



5.4.7 Asset Administration Shell Mapping

Asset Administration Shells are mapped using the following XML Schema construct – it consists of a set of meta data parameters and mostly links to other parts of the XML document or to external elements (based on keys and references).

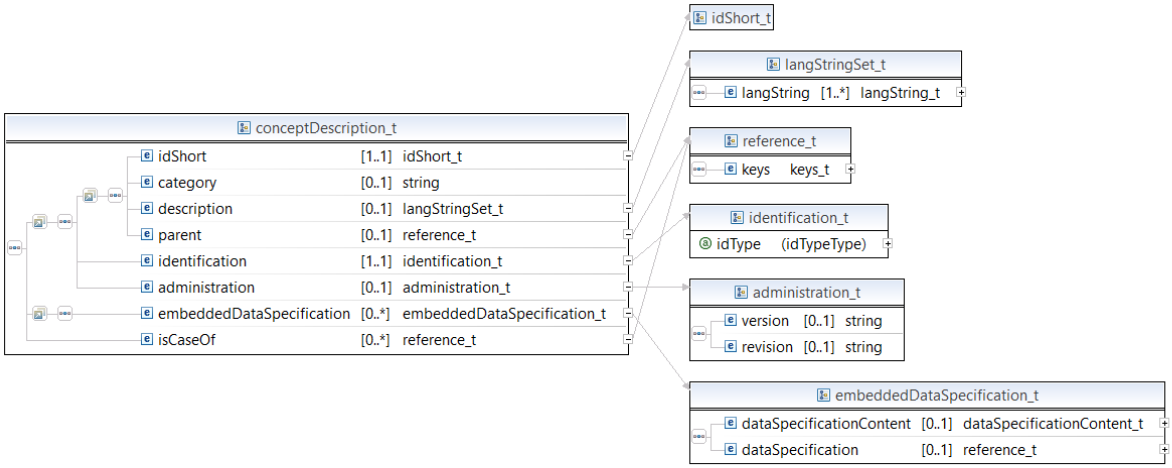
Figure 59 Overview on mapping and meta-data



5.4.8 ConceptDescriptions and EmbeddedDataSpecifications Mapping

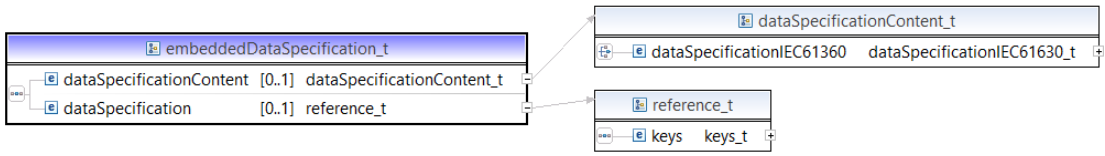
As described above, the definition of a concept comprises of an according reference and a content, which is realized by a data specification.

Figure 60 Concept description in XML in general



The data specification can be e.g. along of an IEC 61360 property:

Figure 61 Data specification via IEC 61360 property attributes



Full XSD and example XML can be found in Annex C.

5.4.9 Attribute Based Access Control Mapping

As described above, the definition of a concept comprises of an according reference and a content, which is realized by a data specification.

Figure 62 Attribute Based Access Control Model Mapping – Security Model

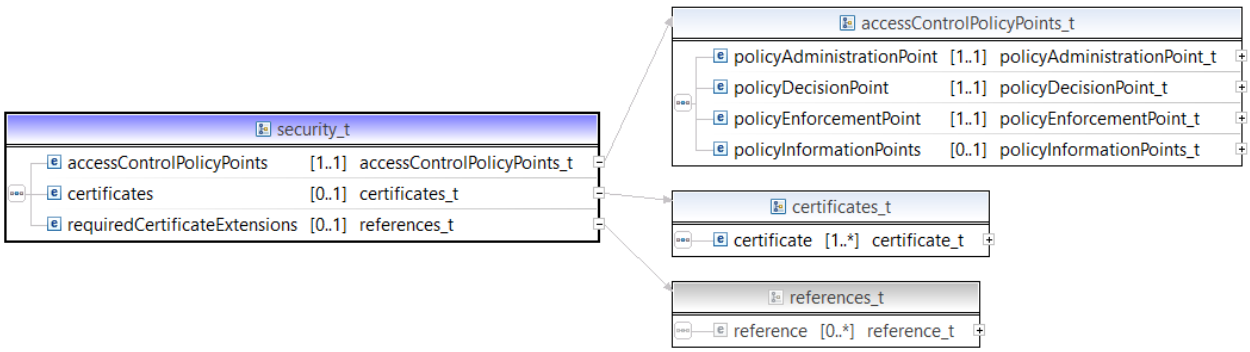


Figure 63 Attribute Based Access Control Model Mapping – Policy Points Model

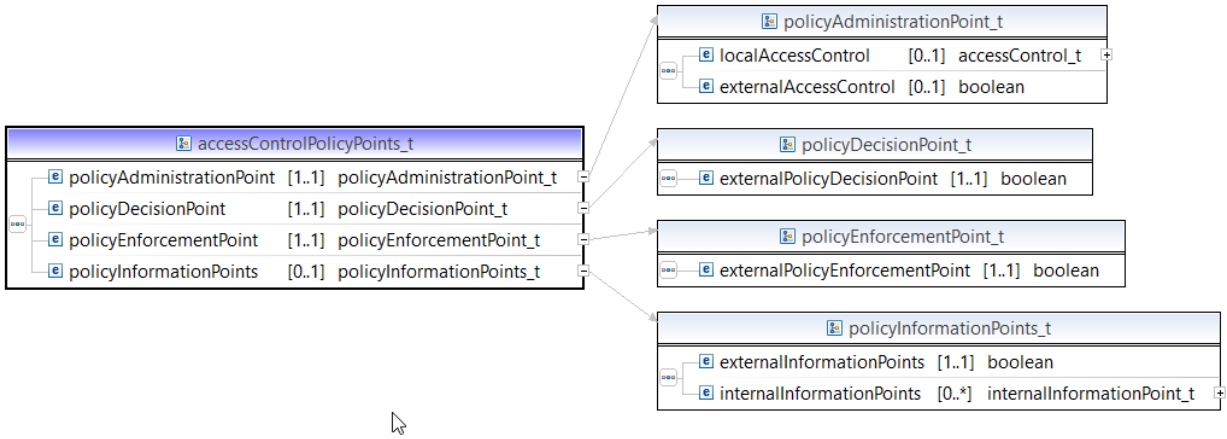
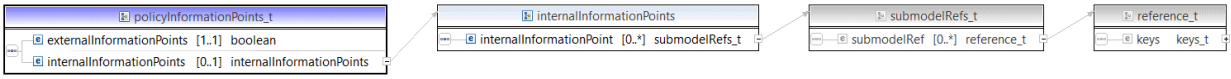


Figure 64 Attribute Based Access Control Model Mapping – Policy Information Points



5.5 JSON

5.5.1 General

In the following clauses an overview of the main concepts of the AssetAdministration Shell JSON serialization is presented. For import and export scenarios the metamodel of an AssetAdministration Shell needs to be serialized. A serialization format is JSON[25] (JavaScript Object Notation). The information is divided in three parts. The first part discusses the rules, in the second part are examples for some specific rules and in the third part the schema and a complete example is shown in the annex.

5.5.2 Rules

The main concepts of the JSON serialization are explained by the following rules.

(1) **If present, names are taken from the information model.**
For comprehensibility reasons the JSON key names should be the same as the representing Element in the metamodel.

(2) Each Referable, Qualifier and Formula have an additional attribute “modelType” with the name of the corresponding object class as value
This rule is needed for deserialization reasons.

(3) **All identifiables have an aggregation on root level**.
The identifiables are AssetAdministrationShells, Assets, Submodels and ConceptDescriptions. To reduce redundancy instances, they are located exclusively in the top-level aggregation.

(4) Identifiables which are not in the top-level aggregations are only references to the corresponding instances in one of the top-level aggregations.
This rule completes the concept of rule 3. There should be no redundant identifiable in the serialized metamodel.

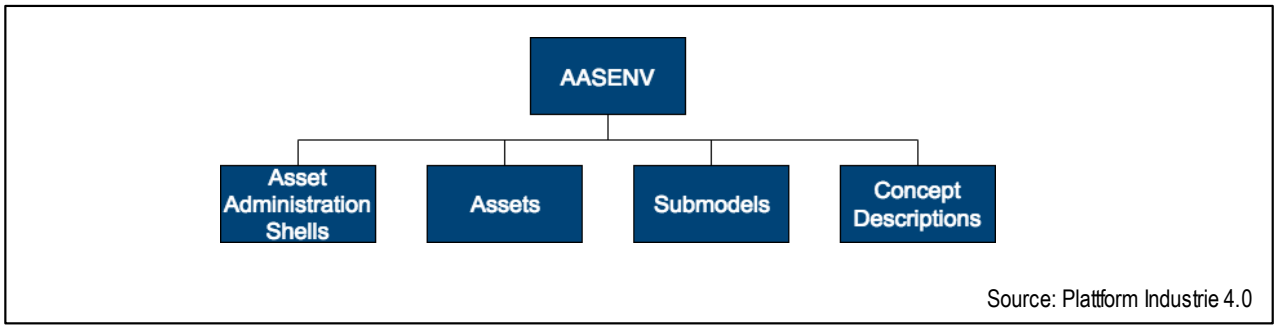
(5) **Data Specification Templates are directly added to the Concept Description.**
Additionally, a new element EmbeddedDataSpecification is introduced that has two attributes: one for the global reference to the data specification identifier and one for the content of the data specification.

[25] see: https://tools.ietf.org/html/rfc8259 or https://www.ecma-international.org/publications/standards/Ecma-404.htm

5.5.3 Example for Top-Level Structures

One serialization describes one Asset Administration Shell environment, that is, a collection of Administration Shells. The root element of the Asset Administration Shell environment has 4 aggregations. For each identifiable class, one aggregation is provided, as required by rule 3.

Figure 65 Top level structure of an AssetAdministration Shell environment mapped to JSON



The resulting JSON is the minimal valid JSON:

Table 13 Minimal JSON for top level structure

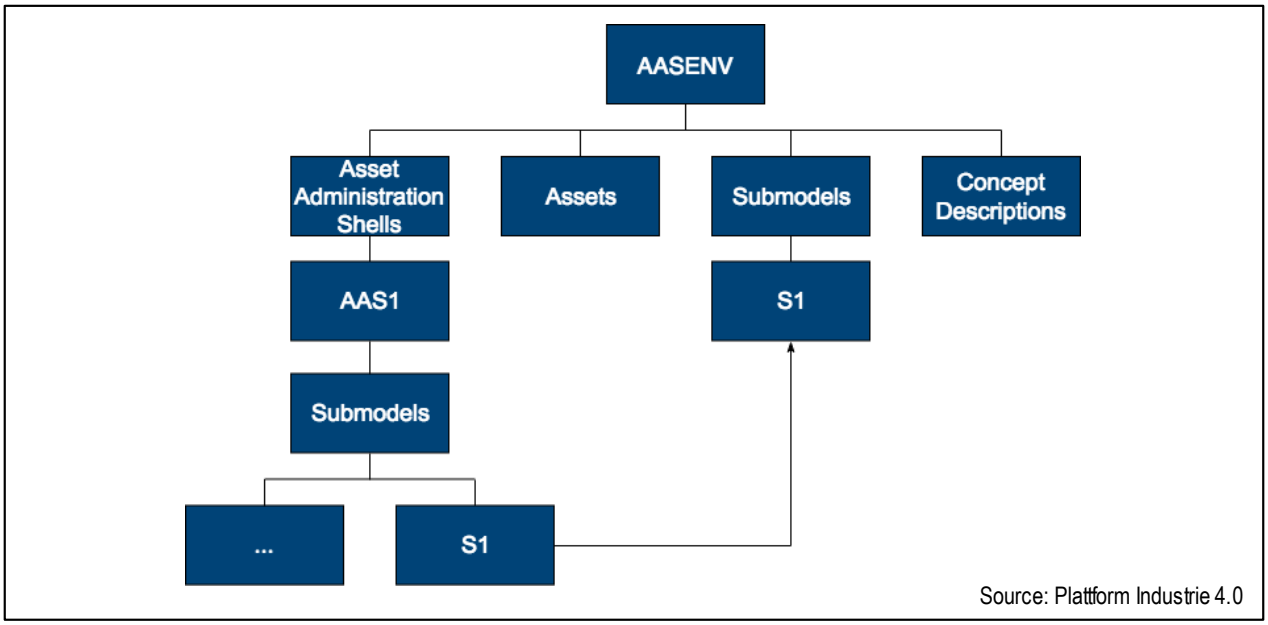
```
{
  "assetAdministrationShells":[ ],
  "assets":[ ],
  "submodels":[ ],
  "conceptDescriptions":[ ]
}
```

5.5.4 Examples for References to Identifiables

As required by rule 4, Identifiables are only allowed to be located in the top-level aggregations. In deeper parts of the structure only References to the corresponding Identifiable must be taken.

In the Asset Administration Shell AAS1, the submodel S1 is only a Reference to the Submodel S1 instance in the top level Submodels aggregation.

Figure 66 Submodel reference in AssetAdministrationShell for JSON



This results in the following exemplary JSON:

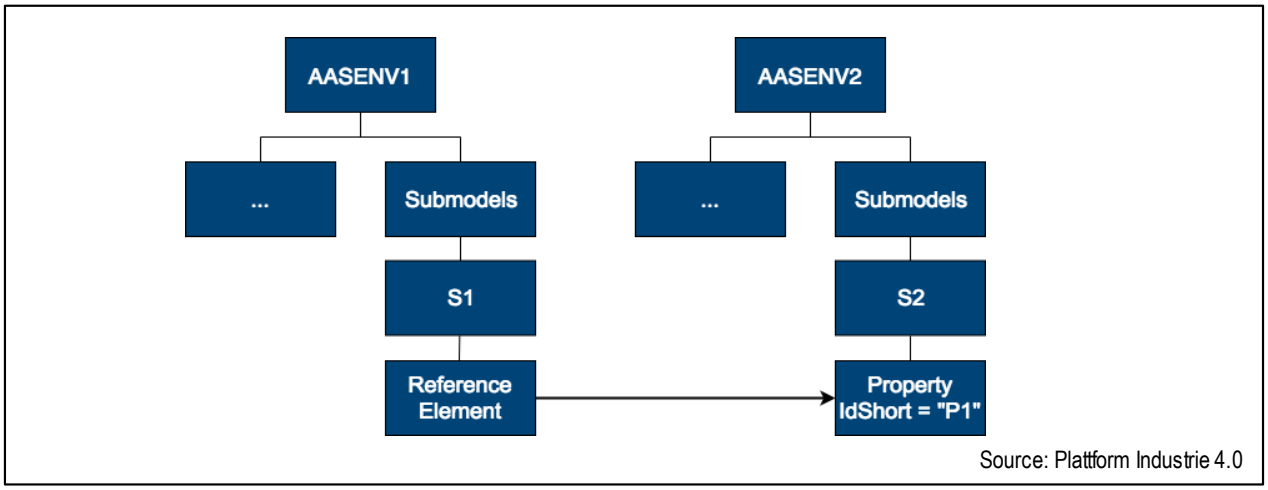
Table 14 Exemplary minimal JSON for References

```
{
  "assetAdministrationShells":[
    {
      "modelType": "AssetAdministrationShell",
      "submodels":[
        {
          "keys":[
            {
              "idType":"IRI",
              "local":true,
              "type":"Submodel",
              "value": "http://env.com/submodels/S1"
            }
          ]
        }
      ],
    …
    }
  ],
  "assets":[],
  "submodels":[
    {
      "modelType":"Submodel",
      "identification":{
        "id": "http://env.com/submodels/S1",
        "idType":"IRI"
      },
      "idShort":"S1",
      "submodelElements":[],
     …
    }
  ],
  "conceptDescriptions":[]
}
```

5.5.5 Examples for ReferenceElement

A ReferenceElement has a Reference as value. This Reference has an aggregation of keys which represents a key chain. The resolved key chain points to an element. In this example the ReferenceElement's value points to a property of another submodel in another Asset Administration Shell environment. The first key is a global key with "local"-attribute set to false, i.e. the reference is not part of the own environment. The second key is a model key which is used to define the corresponding property in the other environment by its IdShort. It is best practice to use the shortest key chain if there are multiple options.

Figure 67 Usage of ReferenceElement in JSON



This results in an exemplary JSON as such:

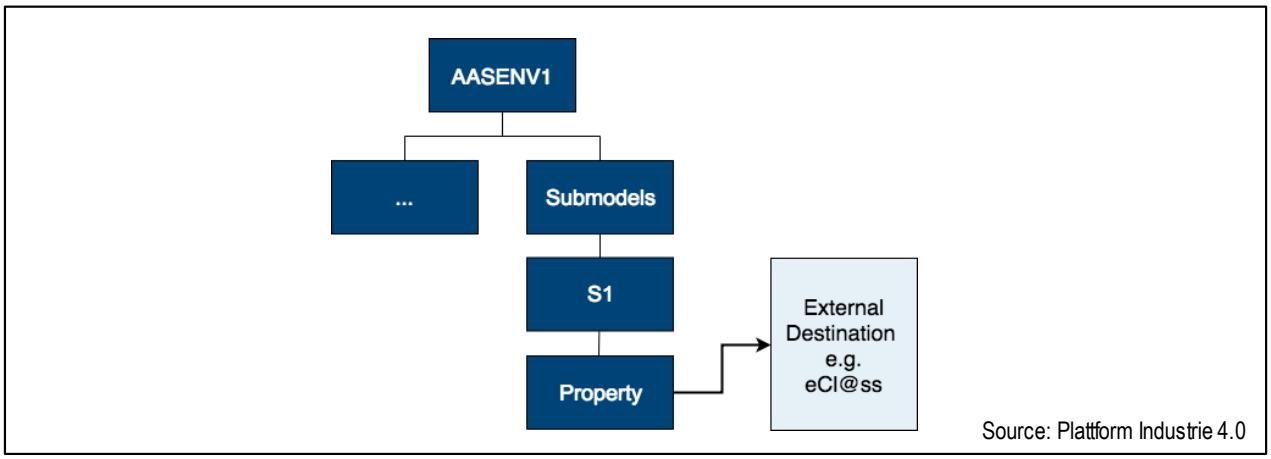
Table 15 Exemplary ReferenceElement in JSON

```
{
  "keys":[
    {
      "idType":"IRI",
      "local":false,
      "type":"Submodel",
      "value":"http://admin-shell.io/submodels/Temperature"
    },
    {
      "idType":"IdShort",
      "local":true,
      "type":"Property",
      "value":"NMax"
    }
  ]
}
```

5.5.6 Examples for GlobalReference

Sometimes it is useful to refer to another standard or something that is not provided by the own Asset Administration Shell environment. In this example the semantics of a Property refers to eCl@ss.

Figure 68 Usage of GlobalReference in JSON



This results in an exemplary JSON as such:

Table 16 Exemplary GlobalReference in JSON

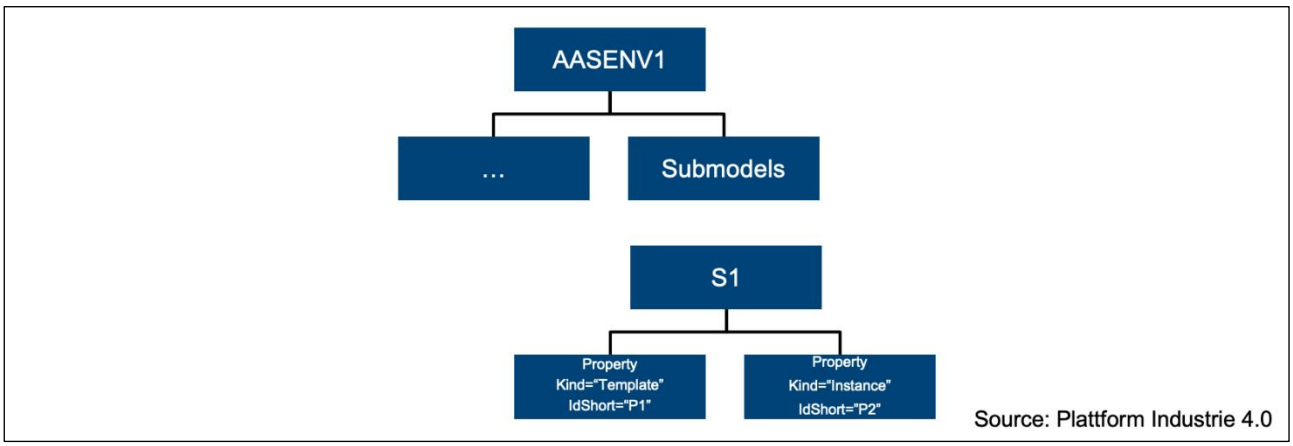
```
{
  "keys":[
    {
      "idType":"IRDI",
      "local": false,
      "type":"GlobalReference",
      "value":"0173-1#02-AAC962#006"
    }
  ]
}
```

5.5.7 Example for a kind = "Template" Reference

A semantic description can either be something external or an instance with kind =”Template”. In this example the Property P2 uses P1 as a template. P1 has kind =”Template” and P2 kind =”Instance”.

Note: typically, templates are assumed to be specified in another Asset Administration Shell as the instances. Here, the depicted situation is simplified for layout reasons.

Figure 69 Exemplary type Reference in JSON



This results in an exemplary JSON as such:

Table 17 Exemplary type Reference in JSON

```
{
  "keys":[
    {
      "idType":"IRI",
      "local":true,
      "type":"Submodel",
      "value":"http://aasenv1.com/submodel/S1"
    },
    {
      "idType":"IdShort",
      "local":true,
      "type":"Property",
      "value":"P1"
    }
  ]
}
```

5.6 RDF

5.6.1 General

The Resource Description Framework (RDF) [44] is recommended standard of the W3C to unambiguously model and present semantic data. RDF documents are structured in the form of triples, consisting of subjects, relations and objects. The resulting model is often interpreted as a graph, with the subject and object elements as the nodes and the relations as the graph edges.

RDF is closely related to Web standards, illustrated by the fact that all elements are encoded using (HTTP-)URIs. As a common practice, the provision of additional information at the referenced location of an RDF entity directly allows the interlinking of entities[26] based on the Web. This process, the following of links in order to discover related information, is called dereferencing a resource and is supported by any browser or web client. Connecting distributed data sources through the Web in the described manner is referenced by the term Linked Data. Connecting the available resources and capabilities of Linked Data with the expressiveness of the Asset Shell is one motivation for the RDF serialization.

In addition, RDF is the basis of a wide range of logical inference and reasoning techniques. Vocabularies like RDF Schema (RDFS) and the Web Ontology Language (OWL) combine the graph-based syntax of RDF with formal definitions and axioms. This allows automated reasoners to understand the relation between entities to some extent and draw conclusions.

Combining both features, the RDF mapping of the Asset Administration Shell can provide the basis for complex queries and requests. SPARQL, the standard query language for the Semantic Web, can combine reasoning features with the integration of external data sources. In order to benefit of these abilities, the AAS requires a clear scheme of its RDF representation. In the following, the necessary transformation rules are presented, followed by an illustration of relevant parts of the scheme and an example. The complete data model together with the RDF scheme are listed in Annex G.

5.6.2 Rules

(1) The default serialization format is Turtle.
Several, equivalent serializations exist for RDF. Among them, the Turtle syntax is regarded as the most appropriate compromise between readability and tool-support. Other formats (RDF/XML, JSON-LD, N3, etc.) can be used without any loss of information.

(2) Shape Graphs represent the validation rules.
The data model itself is an RDF ontology. As RDF itself is following the open-world-assumption, SHACL [38] constraints are necessary in order to enable schema validation. Similarly to XSD for XML, the SHACL format can be used to describe contraints (or shapes) of RDF graphs.

(3) Every entity is encoded as either an IRI or a Literal.
RDF uses IRIs for both entities and relations. If no IRI is predefined, a globally unique IRI is generated. Primitive values are encoded as Typed Literals.

(4) Entities are enhanced with well-known RDF attributes.
Interoperability of concepts and attributes is the main advantage of the RDF mapping. Applying common attributes (rdf:type, rdfs:label, and rdfs:comment) enables the usage of standard tools and interfaces.

(5) Repeating elements are described once and then linked using their IRI identifier.
If a distinct element appears more than one time in the original model but in a different context, for instance in more than one submodel, the RDF entity represents the combination of all attributes.

(6) Keys must have an index attribute.
Keys of a Reference have a defined order, however RDF is explicitly set-based. The index attribute encodes the position in the original sequence. Consequently, Keys belonging to one Reference must have unique numbers in the range [0..*keyCount*], ascending from 0. If only one Key is supplied, the index attribute can also be skipped, implying a default value of ‘0’.

[26] Note: entity as a generic term and entity as a specific submodel element subtype need tobe distinugished.

(7) Multilanguage Strings are split into distinct language strings.
Objects are expected to contain a singular information entity, and the currently available tools would not recognize the different pattern used.

5.6.3 Example Overview

RDF is often regarded as a graph model, as it provides the flexibility to interlink entities at any stage. In the following, the running example is originally provided in Turtle but accompanied with visualizations of the represented graph. Attributes referencing non-literal values are shown as directed links while Literal values are drawn together with the corresponding entity itself. In order to increase readability, the namespace declaring sections are omitted. The complete example with all namespaces can be found in Annex G.

The instances of the core classes, the AssetAdministrationShell, the Asset, Submodels, and ConceptDescriptions are shown in Figure 70. A short snippet of the AssetAdministrationShell is also provided in Table 18. The RDF identifiers are visualized in a condensed format in the figure but represent complete URIs, as displayed in Table 18.One can see the additionally inserted triples for `rdf:type` (1), `rdfs:label` (2), and `rdfs:comment` (3) as determined by Rule 4. The first attribute states the instance' class. The second provides its commonly used name, for instance based on the idShort attribute. rdfs:comment supplies a short description about the regarded entity, based on the description value. The generally available tools, for instance the open source tool Protégé, interpret these attributes and display the correct class hierarchy, render the elements with their labels or supply short explanations based on the comments.

Figure 70 Simplified graph of the core classes in the example

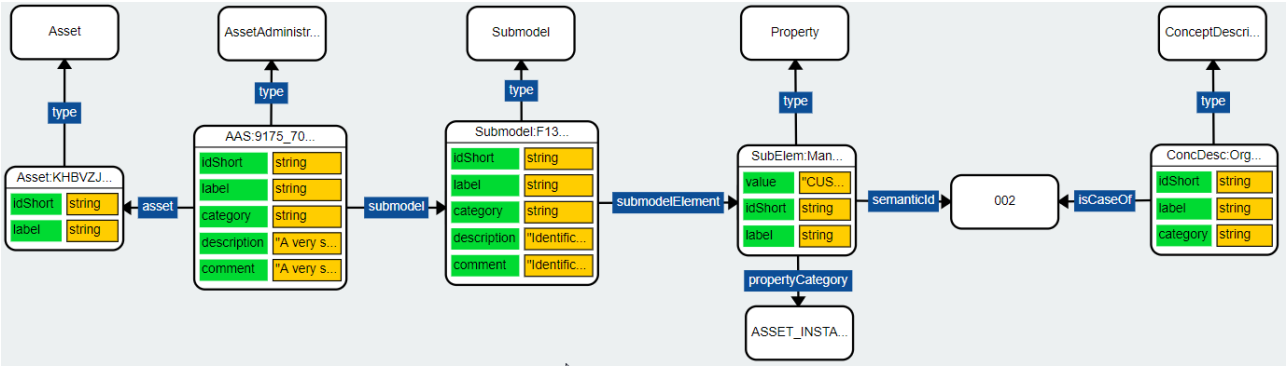


Table 18 Turtle excerpt of an AssetAdministrationShell class

```
<http://customer.com/aas/9175_7013_7091_9168>
 rdf:type aas:AssetAdministrationShell ;                    (1)
 aas:idShort "ExampleMotor"^^xsd:string ;
 rdfs:label "ExampleMotor"^^xsd:string ;                    (2)
 aas:description "A very short description of the AAS instance."@en ;
 rdfs:comment "A very short description of the AAS instance."^^xsd:string ;   (3)
 aas:asset [ … aas:value <http://customer.com/assets/KHBVZJSQKIY> ; … ] ;
 aas:submodel […  aas:value <http://i40.customer.com/type/1/1/F13E8576F6488342> ; …] ;
 aas:submodel […  aas:value <http://i40.customer.com/type/1/1/7A7104BDAB57E184> ; …] ;
 aas:submodel […  aas:value <http://i40.customer.com/instance/1/1/AC69B1CB44F07935>; …];
 aas:submodel […  aas:value <http://i40.customer.com/type/1/1/1A7B62B529F19152> ; …] ;
 aas:identification <http://customer.com/aas/9175_7013_7091_9168> ;
 aas:category "CONSTANT"^^xsd:string .
```

5.6.4 Example Schema Shape

The Shapes Constraint Language (SHACL) [38] introduces a W3C recommendation for validation mechanisms on RDF graphs. The definition of required attributes, cardinality of relations or datatype restrictions in the form of shapes (see Table 19 for an example shape) is an important aspect to enable data quality assurance in any productive system. Some tools are already created to assist the creation of SHACL shapes, e.g. a Protégé plugin and as a part of TopBraid Composer. As SHACL shapes are also defined in RDF, they share the same format and thereby reduce the required technology stack and reduces the amount of necessary libraries.

Table 19: A SHACL Shape for the AssetAdministrationShell and its asset attribute

```
aas:AssetAdministrationShellShape a sh:NodeShape ;
  sh:targetClass aas:AssetAdministrationShell ;
  rdfs:subClassOf aas:HasDataSpecificationShape ;
  rdfs:subClassOf aas:IdentifiableShape ;
  sh:property [
     a sh:PropertyShape ;
     sh:path aas:asset ;
     sh:class aas:Reference ;
     sh:minCount 1 ;
     sh:maxCount 1 ;
     sh:message "Exactly one <i>asset</i> attribute having an <i>Reference</i> entity is required."^^xsd:string ;
     sh:name "asset"^^xsd:string ;
   ] ; …
```

5.6.5 IRI Mapping

Every enitiy in RDF is either a Resource or a Literal. While Literals present data values, like strings, numbers or any sequence of characters, Resources represent the nodes and edges in the data graph. As Resources must be identified through IRIs (preferable even URIs), the creation of suitable IRIs is fundamental for the mapping. Whenever resources of an AAS are already identified through IRIs (see also Section 4.4.6), these IRIs/URIs are also utilized in the RDF model. However, in cases where no IRI is given, a defined procedure has to be followed. The following decision steps determine the necessary steps in order to create an unambiguous IRI for every element of the AAS.

These steps are only executed once per distinct element. If elements occur more than one time, always the same IRI identifier has to be used. This is especially relevant for the fallback solution (2b and 3) where the initial character sequence must be reused. Different elements must not get the same sequence under any circumstances.

1. If the element has IdentifierType = "IRI":
   Use the value of the *identification* attribute, else:
2. If the enclosing AAS has an IdentifierType = "IRI":
   a. **If the element inherits from Referable:**
      Apply the template: <AAS identification URI>/<path/to/element>/<idShort>, where the path to the element is the concatenation of the respective idShorts separated by slashes, **else:**
   b. Apply the template: <AAS identification URI>/<path/to/element>/<random characters>
3. Use randomized character sequence:
   <scheme>://<random characters>

5.6.6 Example Mapping

Several mapping languages have been created in order to transform structured data into RDF graphs. Most prominently, R2RML (relational data only) and RML (relational data, XML, JSON, etc.) are used to specify the necessary mappings.

A RML mapping can be used to transform a given Asset Administration Shell from its XML or JSON serialization to any potential RDF serialization. The snippet in Table 20 illustrates an RML TriplesMap used to convert the XML example from Annex E. The AssetAdministrationShell elements are iterated (`rml:logicalSource`) and, among others, all relations to referenced Submodels are extracted (`rr:predicateObjectMap`).

Table 20 RML TriplesMap snippet for parsing XML to RDF

```
_:AssetAdministrationShellMap   rdf:type  rr:TriplesMap ;
   rml:logicalSource [
      rml:source "[…]/customer_com_aas_9175_7013_7091_9168.aas.xml" ;
      rml:referenceFormulation ql:XPath ;
      rml:iterator "//*[local-name()='assetAdministrationShell']"
   ] ;
   rr:subjectMap [
      rml:reference "identification" ;
      rr:class aas:AssetAdministrationShell
   ] ;
   rr:predicateObjectMap [
      rr:predicate aas:submodel ;
      rr:objectMap [
         rml:reference "submodelRefs/submodelRef/keys/key" ;
         rr:termType rr:URI
      ]
   ] ; …
```

5.6.7 Example Asset Administration Shell with References

RDF entities usually do not have an explicit order. That implies that a reordering of elements is allowed. As for instance references to Submodels do not depend on their sequence, this behaviour is fine. However, especially Key elements do have a certain sequence and keeping that is important. Still, the available set of tools does not guarantee to preserve their original order. Consequently, a new attribute (`aas:index`) needs to be introduced. Figure 71 explains the regarded situation, where an Asset Administration Shell expresses its relation with an Asset. Table 21 illustrates the use of `aas:index`. If more than one Key appears, the index attribute must always be used. Furthermore, no two index values must have the same integer value as otherwise the order information is lost. The first index is denoted with 0 in ascending order.

Figure 71 Graph snippet of an AssetAdministrationShell, linked to an Asset through a Reference and Key

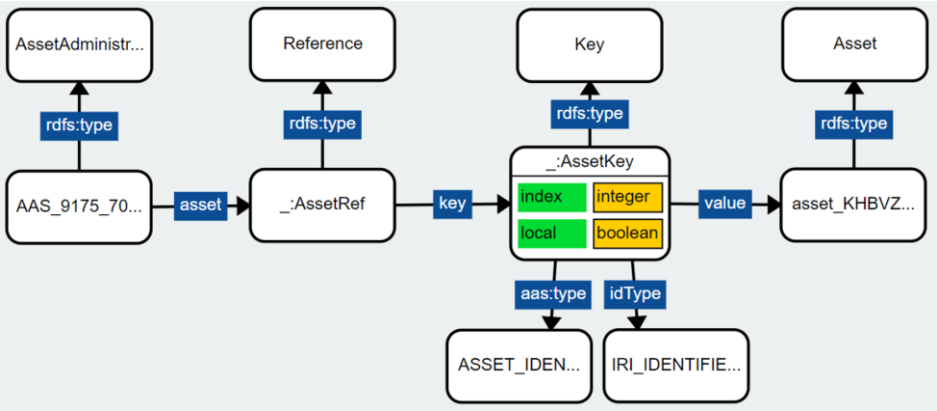


Table 21 Exemplary AssetAdministrationShell stating its identification and containing one Asset

```
<http://customer.com/aas/9175_7013_7091_9168> rdf:type aas:AssetAdministrationShell ;
   aas:identification <http://customer.com/aas/9175_7013_7091_9168> ;
   aas:asset [
      a aas:Reference ;
      aas:key [
         a aas:Key ;
         aas:index "0"^^xsd:integer ;
         aas:type aas:ASSET_IDENTIFIABLE_ELEMENT ;
         aas:local "true"^^xsd:boolean ;
         aas:value <http://customer.com/assets/KHBVZJSQKIY> ;
         aas:idType aas:IRI_IDENTIFIER_TYPE ;
      ] ;
   ] ; … .

<http://customer.com/assets/KHBVZJSQKIY> rdf:type aas:Asset ;
   aas:identification <http://customer.com/assets/KHBVZJSQKIY> ;
…
```

5.6.8 Example Asset

The Asset is the objective of the whole Asset Administration Shell. Figure 72shows the graph of the RDF encoded ServoDCMotor in Table 22. The motor, identified through its unique URI, has a Submodel providing all necessary information describing its identity.

Figure 72 Asset and its identifying Submodel

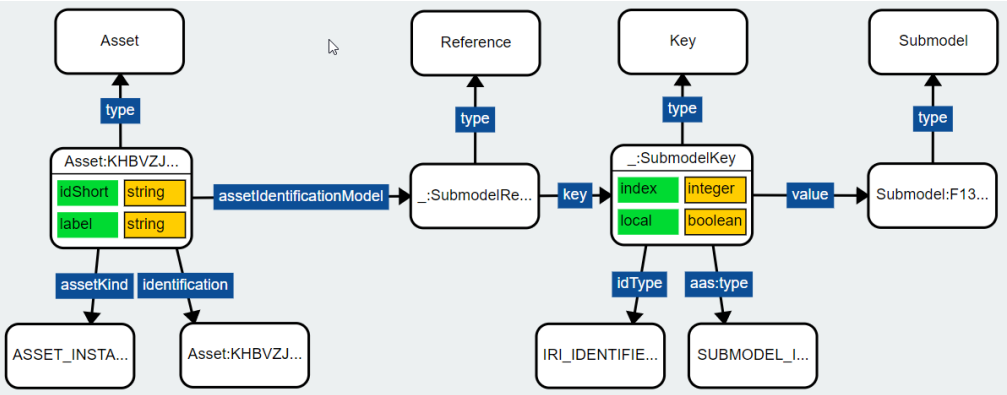


Table 22 RDF Asset in Turtle

```
<http://customer.com/assets/KHBVZJSQKIY> rdf:type aas:Asset ;
   aas:idShort "ServoDCMotor"^^xsd:string ;
   rdfs:label "ServoDCMotor"^^xsd:string ;
   aas:identification> <http://customer.com/assets/KHBVZJSQKIY>.
   aas:assetIdentificationModel [
   rdf:type aas:Reference ;
   aas:key [
      rdf:type aas:Key ;
      aas:index "0"^^xsd:integer ;
      aas:type aas:SUBMODEL_IDENTIFIABLE_ELEMENT ;
      aas:local "true"^^xsd:boolean ;
      aas:value <http://i40.customer.com/type/1/1/F13E8576F6488342> ;
      aas:idType aas:IRI_IDENTIFIER_TYPE .
   ] ;
   aas:assetKind aas:ASSET_INSTANCE .
```

5.6.9 Example Submodel with Property

Submodels contain the relevant information for a use case. Figure 73 shows the Identification Submodel with the Property containing the manufacturer of the asset. In the RDF terminology, every relation actually is called a property. In the data model of the Asset Administration Shell however, a Property is a defined sub class of the SubmodelElement. The distinction is made through the namespace: `rdf:Property` for every relation, `aas:Property` for certain SubmodelElements (see Table 23).

Figure 73 Graph of the relations between a Submodel (F13E8576F6488342) and a Property (Manufacturer)

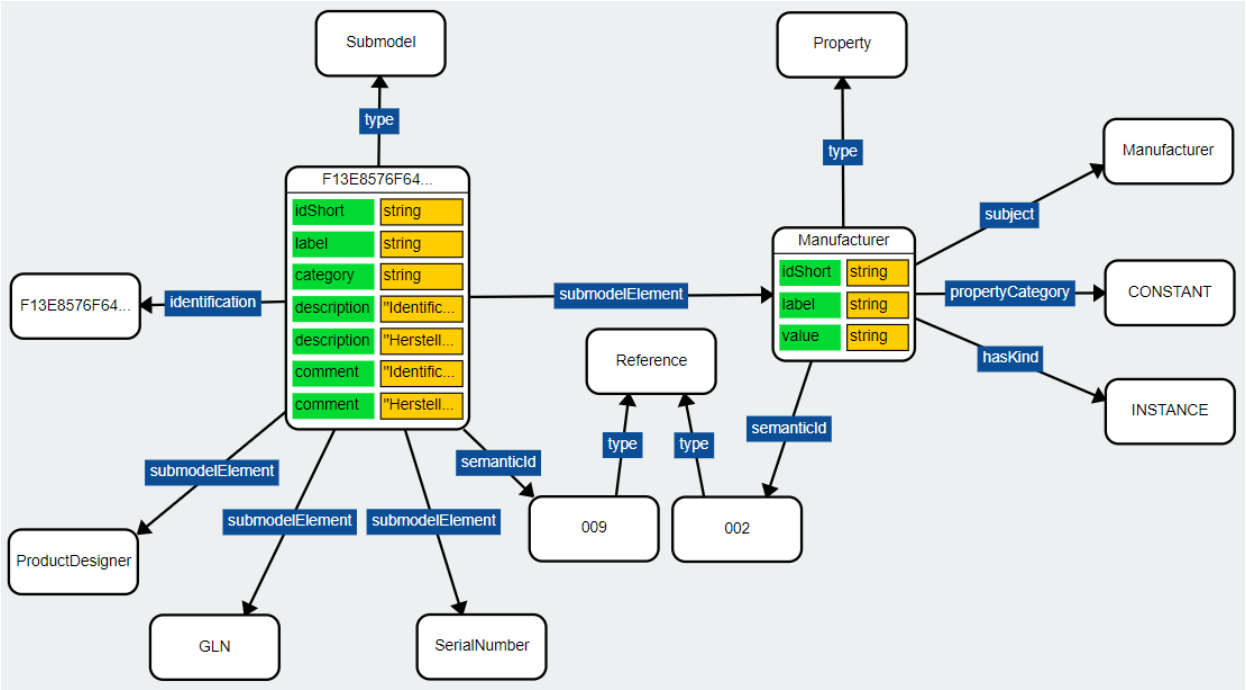


Table 23 Exemplary Submodel stating its semantic ID and containing one SubmodelElement

```
<http://i40.customer.com/type/1/1/F13E8576F6488342> rdf:type aas:Submodel ;
  aas:idShort "Identification"^^xsd:string ;
  aas:semanticId [
    a aas:Reference ;
    aas:key [
      a aas:Key ;
      aas:index "0"^^xsd:integer ;
      aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
      aas:local "false"^^xsd:boolean ;
      aas:value "0173-1#01-ADN198#009"^^xsd:string ;
      aas:idType aas:IRDI_IDENTIFIER_TYPE ;
    ] ;
  ]; …
 aas:identification <http://i40.customer.com/type/1/1/F13E8576F6488342> ;

 aas:submodelElement [
    rdf:type aas:Property ;
    rdf:subject <http://i40.customer.com/type/1/1/F13E8576F6488342/Manufacturer> ;
    aas:idShort "Manufacturer"^^xsd:string ; …
    aas:semanticId [
      a aas:Reference ;
```

```
        aas:key [
          a aas:Key ;
          aas:index "0"^^xsd:integer ;
          aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
          aas:local "false"^^xsd:boolean ;
          aas:value "0173-1#02-AAO677#002"^^xsd:string ;
          aas:idType aas:IRDI_IDENTIFIER_TYPE ;
        ] ;
    ]; …
] ; …
```

5.6.10 Example MultiLanguage String

The Identification Submodel in Table 24 has two descriptions, one in English and one in German. RDF proposes the usage of LangStrings, typed Literals with a language tag. Table 25 illustrates, how Rule 7 leads to different object values.

Table 24 Exemplary MultiLanguage description of a Submodel

```
<aas:submodel>
    <aas:idShort>Identification</aas:idShort>
    <aas:description>
        <aas:langString lang="EN">Identification from Manufacturer</aas:langString>
        <aas:langString lang="DE">Hersteller-Identifikation</aas:langString>
    </aas:description>
    …
</aas:submodel>
```

Table 25 RDF serialization of attribute values in different languages

```
<http://i40.customer.com/type/1/1/F13E8576F6488342>
            aas:description "Identification from Manufacturer"@en ;
            aas:description "Hersteller-Identifikation"@de .
```

5.6.11 Example Concept Description

A Concept Description defines the meaning of the entities used in the Asset Administration Shells, Assets, and Submodels. Table 26Table 26 illustrates a description for the organization name of a constant. The core building blocks of the example are the reference to the data specification key (1), the actual content conforming to IEC 61360 (2), and the reference to the actual identifier in the previously used elements (3).

Table 26 ConceptDescription Example in RDF

```
<http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Organization/OrganizationOfficialName>
    rdf:type aas:ConceptDescription ;
    aas:idShort "OrganizationName"^^xsd:string ;
    rdfs:label "OrganizationName"^^xsd:string ;
```

```
   aas:category "CONSTANT"^^xsd:string ;
   aas:identification <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Organization/OrganizationOfficialName> ;
   aas:hasDataSpecification [ (1)
      rdf:type aas:Reference ;
      aas:key [
         a aas:Key ;
         aas:index "0"^^xsd:integer ;
         aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
         aas:local "false"^^xsd:boolean ;
         aas:value <http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360> ;
         aas:idType aas:IRI_IDENTIFIER_TYPE ;
      ] ;
   ] ;
   aas:content [ (2)
      rdf:type aas:DataSpecificationIEC61360 ;
      aas:preferredName "offizieller Name der Organisation"@de ;
      aas:preferredName "official name of the organization"@en ;
      aas:shortName "OrganizationOfficialName"@en ;
      aas:datatype "STRING"^^xsd:string ;
      aas:definition "Der offizielle Name der Organisation."@de ;
   ] ;
   aas:isCaseOf [ (3)
      rdf:type aas:Reference ;
      aas:key [
         a aas:Key ;
         aas:index "0"^^xsd:integer ;
         aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
         aas:local "true"^^xsd:boolean ;
         aas:value "0173-1#02-AAO677#002"^^xsd:string ;
         aas:idType aas:IRDI_IDENTIFIER_TYPE ;
      ] ;
   ];
.
```

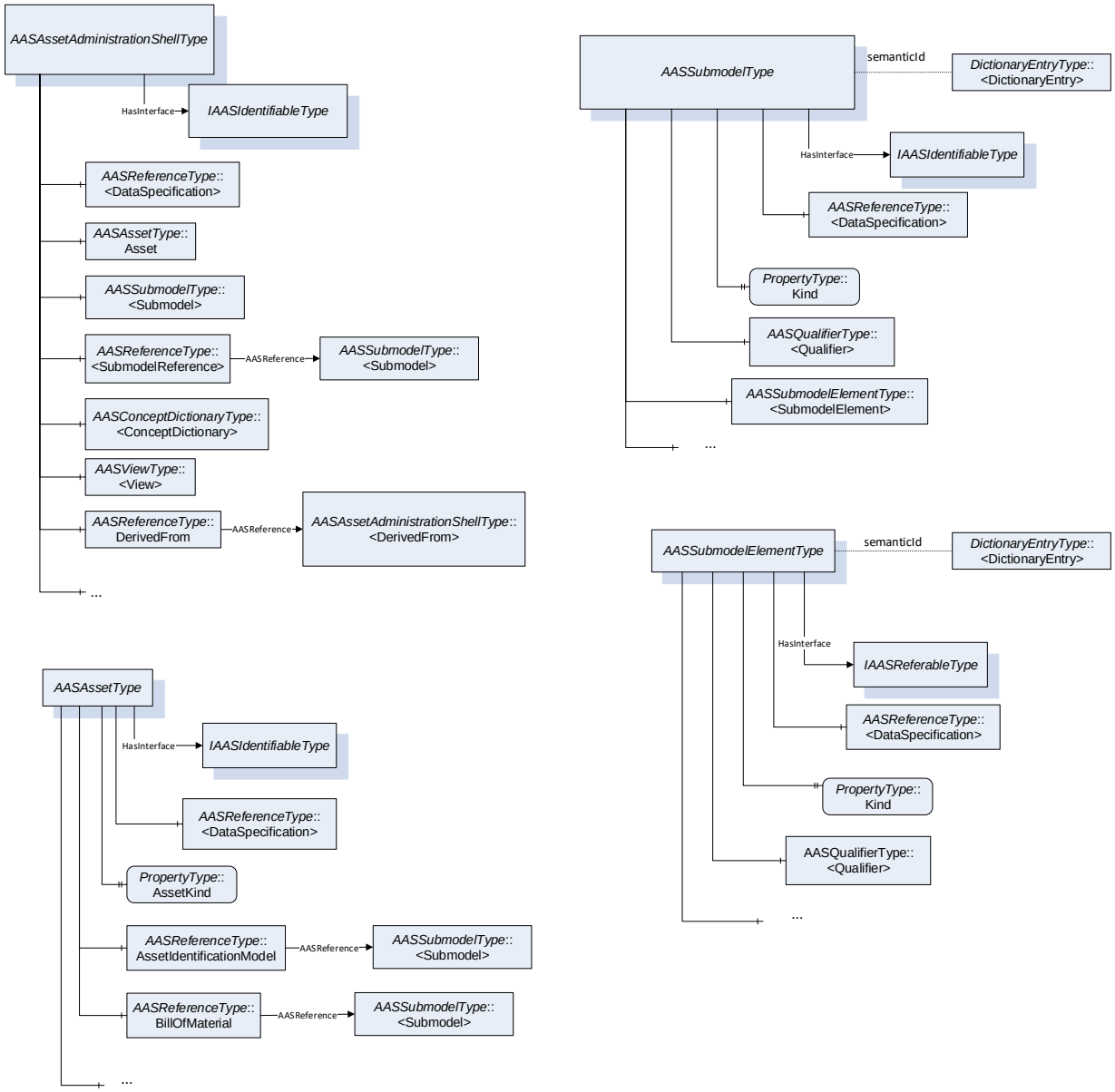
5.7 OPC UA

5.7.1 General

The works of the mapping to the OPC Unified Architecture are currently carried out in a joint working group[27] between OPC Foundation, ZVEI and VDMA. In the following the main aspects for the mapping are described. In [39] the details of the companion specification can be found.

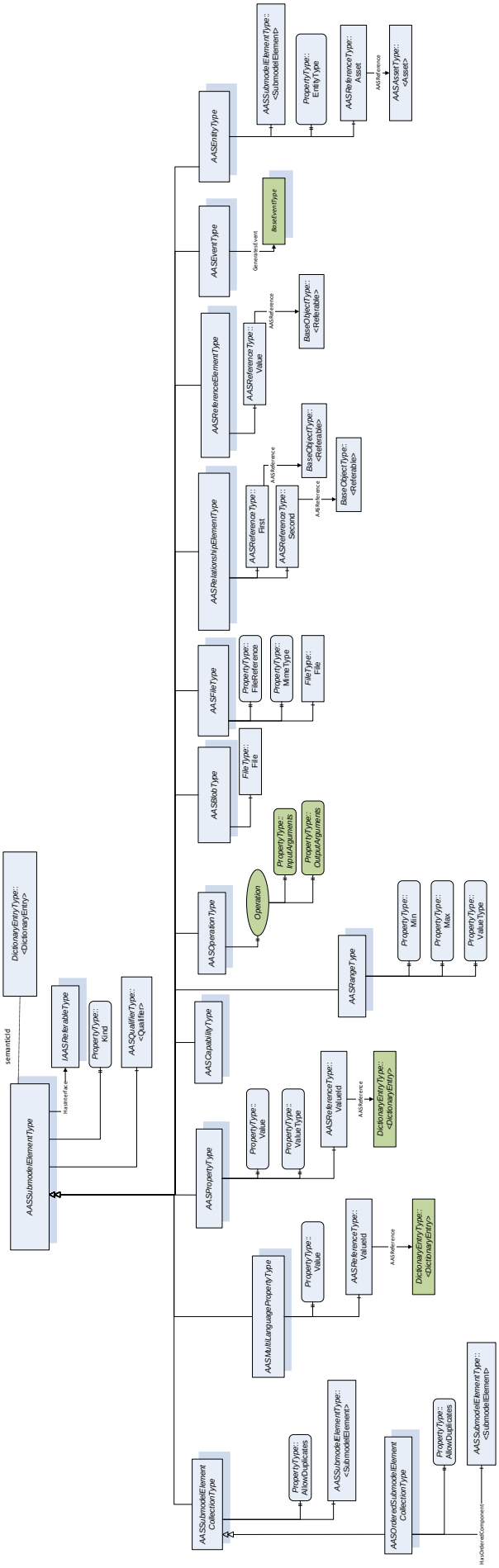
Note: Boxes in green (like "*HasDictionaryEntry*") are denoting elements (object types, reference types etc.) that are predefined in the OPC UA Specifications.

Figure 74 Overview OPC UA Information Model for AAS



[27] see: https://opcfoundation.org/collaboration/i4aas/

Figure 75 Submodel Element Subtypes in OPC UA



5.7.2 Rules

The rules for mapping of the AAS information model to OPC UA information model are given in the following. In subsequent clauses examples for the rules are given.

General Rules:

1. For all class elements in AAS an object type with the same name + suffix "Type" + prefix "AAS" is added. Example: *AASAssetType for Asset, AASSubmodelElementType for SubmodelElement and AASQualifierType for Qualifier*. These Types are derived from *BaseObjectType.* Exception: *ConceptDescription*s und *Referables* see below.
2. For all types in AAS that can not directly be mapped to OPC UA primitive types a data type is created with the same name + suffix "DataType" + prefix "AAS". Example: *AASAssetKindDataType for AssetKind*. Exception: *LangStringSet* is mapped to the predefined OPC type "*LocalizedText*".
3. Attributes of classes in AAS that have a simple data type are mapped to "HasProperty" references within the object type. The BrowseName corresponds to the name in the AAS UML model but is starting with Capital Letter. Example: *AASAdministrativeInformationType* has property *Version* with data type "String".
4. The cardinality of an association or aggregation is specified via OPC Modelling rules. The OPC modelling rule "*Optional*" is used if the cardinality is Zero or 1. The OPC modelling rule "*Mandatory*" is used if the cardinality is One. The OPC Modelling rule "*OptionalPlaceholder*" is used if the cardinality is zero, one or more than one element. The OPC Modelling rule "*MandatoryPlaceholder*" " is used if the cardinality is one or more than one element.
5. Aggregation and composition attributes of classes in AAS are mapped to "HasComponent" References within the object type.
   In case of cardinality 0 .. 1 or 1 the BrowseName corresponds to the name in the AAS UML model but is starting with Capital Letter. Example: Objects of type "*AASAssetType*" have a component with browse name "AssetIdentificationModel".
   In case of cardinality > 0 the BrowseName corresponds to the idShort in the AAS UML model. Example: *AASSubmodelType* has OPC components with OPC TypeDefinition *AASSubmodelElementType*. A submodel element may have the idShort "MaxRotationSpeed". Then the browse names of the component ist "MaxRotationSpeed" as well.
6. Since OPC UA does not support multiple inheritance abstract classes (like e.g. "*Qualifiable*" or *"Identifiable*") are not modelled via subtype reference in OPC UA. The corresponding attributes, aggregations and compositions are modelled as part of the inheriting class. For details see rules below.

Rules for SubmodelElements:

7. Specific for the *Blob* SubmodelElement type (*AASBlobType*) the predefined OPC type definition "*FileType*" is used for the value. References of type "*FileType*" are components of the object type. The browser name is not "value" but "File". The mime type is part of the OPC FileType and therefore not added. In contrast to the OPC FileType mime type is mandatory to be filled.
8. Specific for the *File* SubmodelElement type (*AASFileType*) the value attribute is mapped to an OPC property with BrowseName "FileReference". Additionally an object of type "FileType" with browse name "File" can be added similar as for the Blob. Since this is optional the mime type is modelled as OPC property. In case both are present, then the mime type needs to be the same.
9. *SubmodelElementCollection* can be either ordered or not ordered. In case of an ordered collection "SubmodelElementCollection" is realized as *AASOrderedSubmodelElementCollectionType* and the relationship between collection and submodel elements is realized via the predefined OPC UA "*HasOrderedComponent*" reference type. Otherwise a AAS*SubmodelElementCollectionType* is used.
10. For Operations first an *AASOperationType* is defined but then the OPC "Method" is used for describing the operation. The name of the method is "*Operation*".
    Hint: The OPC UA Specification Amendment 3: Method Metadata allows to add meta information to individual arguments (*HasArgumentDescription*). This is used to realize *semanticId* by using the OPC reference type "*HasDictionaryEntry*".

11. For AAS references as used in *ReferenceElement* of *RelationshipElement* see rule for referencing.
12. For AAS submodel elements "Event" the object type "*AASEventType*" references an OPC UA event via the reference type "*GenerateEvent*".

Rules for Referables and Identifiables:

13. For Referable and Identifiable separate OPC object types are defined that are referenced from the corresponding object types representing the concrete referables and identifiables via the OPC "*HasInterface*" Reference type. The naming convention for this is as follows "IAAS<AAS UML class name>". Example: *IAASIdentifiableType*
14. In case of referenced referables with modelling rule "OptionalPlaceholder" or "MandatoryPlaceholder" the attribute *idShort* of AAS Referables is represented by the browse name of an element. Since there are cases like for AssetAdministrationShell/asset where the browse name is "Asset" but the asset has an *idShort* as well, *idShort* is modelled additionally. In cases with no prefined browse name the browse name and the *idShort* should be identicial.
15. The *parent* attribute of *Referables* is not explicitly modelled because OPC UA supports native navigation.
16. In case of referenced referables with modelling rule "Optional" or "Mandatory" the browse name is identical to the AAS attribute name and the display name shall be identical to the idShort.

Rules for Qualifiables:

17. Qualifier of an element are modelled with OPC UA **HasComponent** reference type. Since qualifier are not referable they do not have a browse name that corresponds to an AAS attribute. Instead the name should be generated as follows: **qualifier:<value of AAS:Qualifier/type>=<value of Qualifier/value>**.

Rules for semanticId and Concept Descriptions:

18. A concept description is inheriting from the predefined OPC "*IrdiDictionaryEntry*" or "*UriDictionaryEntry*". This is why there are both object types: "*AASIrdiConceptDescriptionType*" or "AASIriConceptDescription-Type" and not only one like for the other AAS classes. Additionally for idType = Custom a new Type "*AASCustomConceptDescriptionType*" is created inheriting directly from "*DictionaryEntryType*".
19. Concept descriptions are added to a folder on the server side. The top-level folder shall be named "*Dictionaries*". Below additional subfolders can be created.
20. *semanticId* is modelled by using the predefined OPC reference type "*HasDictionaryEntry*" and is either referencing an object of type "AASIrdiConceptDescriptionType" or of type "AASUriConceptDescriptionType" or of type "AASCustomConceptDescriptionType".
21. Additionally a concept description has at least one Add-In to allow the usage of the IEC61360 data specification template (see rules for data specifications).

Rules for Data Specifications:

22. Concrete data specifications are inheriting from the AAS object type "*AASDataSpecificationType*".
23. There is no need in OPC to distinguish between the data specification properties and the data specification content defining the properties that shall be added to the object type that uses the data specification. The AAS attributes of *DataSpecification* are modelled as OPC UA properties or components (rules as above) of the *AASDataSpecificationType* but are not instantiated. This is always the case in OPC UA if there are no modelling rules attached to a property or component.
24. The concept of embedded data specifications is used. The element that is using the data specification uses the OPC reference type "*HasAddIn*". This Add-In uses pairs of elements: one property being the global external reference to a data specification, the other one the data specification content.

Rules for Referencing:

25. For AAS references as used in *ReferenceElement* of *RelationshipElement* a new non-hierarchial Reference Type "*AASReference*" is introduced. The OPC Reference Type "*HasComponent*" is not directly used to reference an element because OPC references can only reference to elements within its own name space. For AAS, however, also global external references are possible – to elements in other AAS on other OPC Servers or to entities completely outside the scope of AAS. The object with type "*AASReferenceType*" is holding the unique key chain to the referenced elements and optionally can reference the "real" element via "AASReference" reference. There is no special rule for the browse name in this case. The display name should be the same as the idShort of the referenced element.
26. The Keys of a references are realized as an array. Every single Key is serialized as described in Clause 5.2.1.

Rules for Semantics of Model Elements

27. The "HasDictionaryEntry" reference type of OPC UA is not only used to describe the semantics of objects but also of object types. For doing so the rules for creating identifers as defined in Clause 5.2.2 are used.

5.7.3 Example Overview

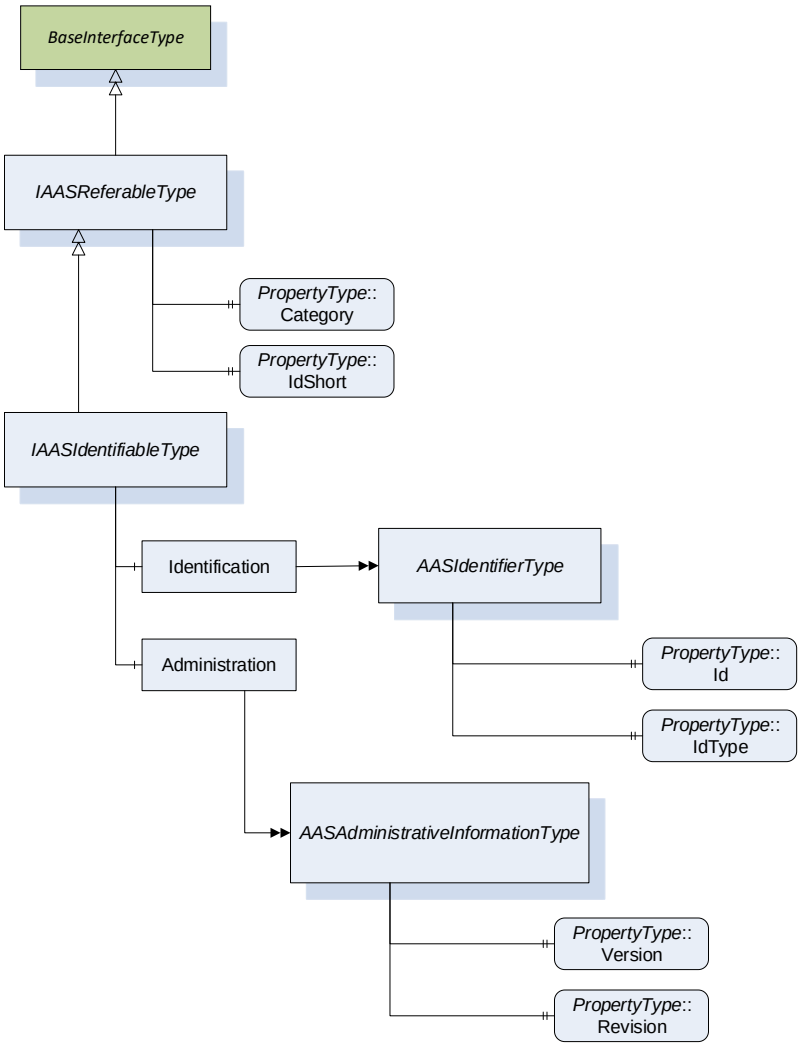
Figure 76 shows an OPC UA Server with an AAS containing several submodels: Documentation, Identification, OperationalData and TechnicalData. For TechnicalData the attributes CoolingType, Identification, Kind, etc. are visible. The property maxRotationSpeed has the OPC properties Kind, Value, Category and so on. The mapping are explained in more detail in the following subclauses.

Figure 76 Overview OPC UA Server with Max Rotation Speed

5.7.4 Identifiables and Referables

Identifiables and Referables are modelled as interfaces, see Figure 77. As exaplained in the rules idShort is not only a property but in cases with no predefined browse name also the browse name of an object.

Figure 77 Identifiables and Referables in OPC UA



5.7.5 Example Submodel with Property etc.

In Figure 78 an example for a submodel with one property “MaxRotationSpeed” is shown. The example is not complete, some attributes like kind etc. are missing. In the next Clause the same example is shown adding *semanticId*.

In Figure 79 the definition of the submodel type is shown.

In Figure 80 the same example from Figure 78 is shown in UAExpert.

Figure 78 Example Submodel TechnicalData (Extract)

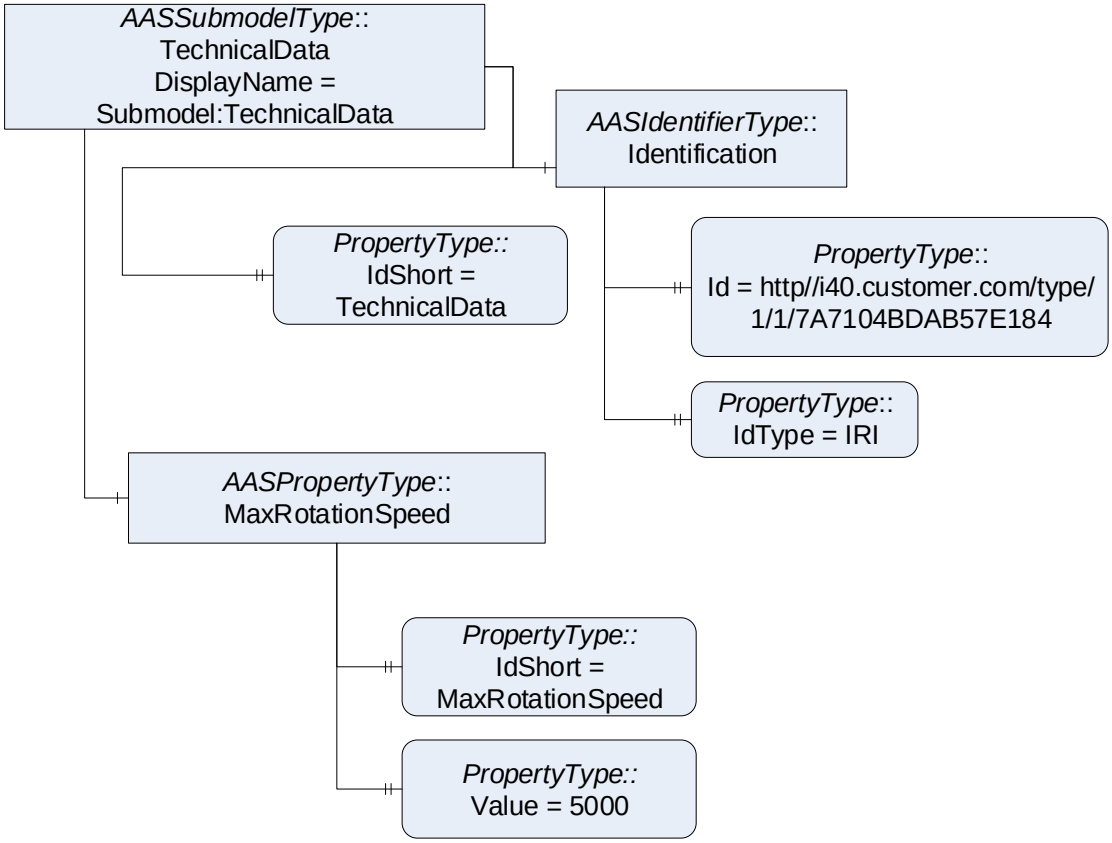


Figure 79 OPC UA Structure Submodel

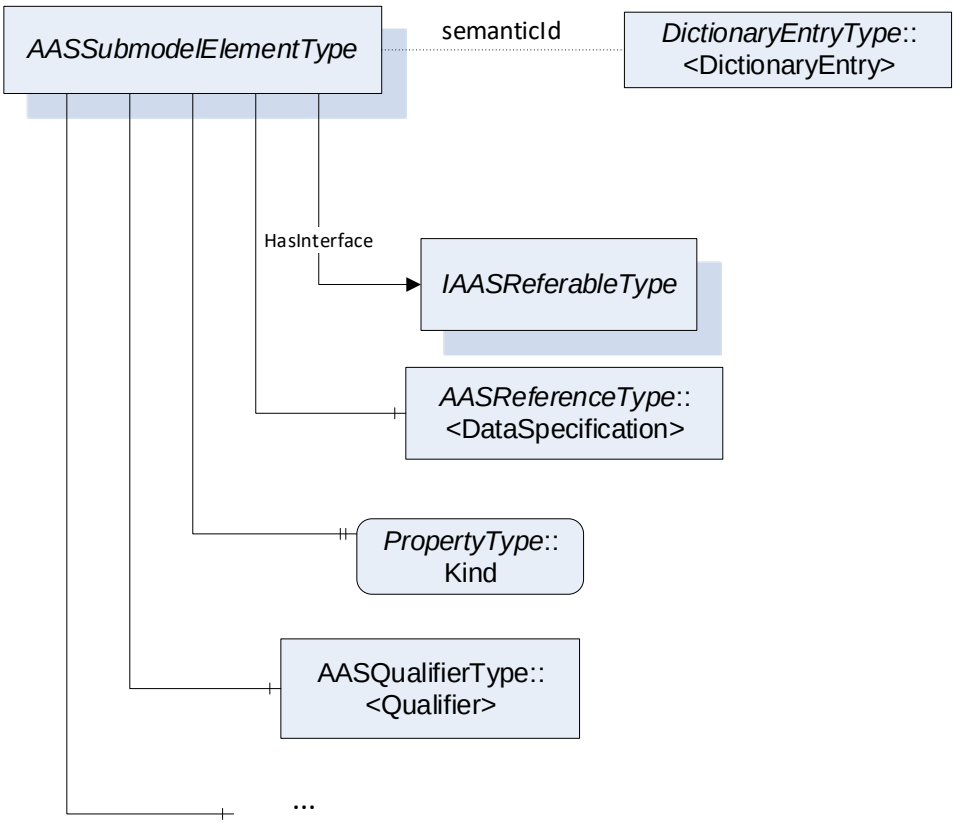


Figure 80 Example UAExpert Max. Rotatation Speed Property

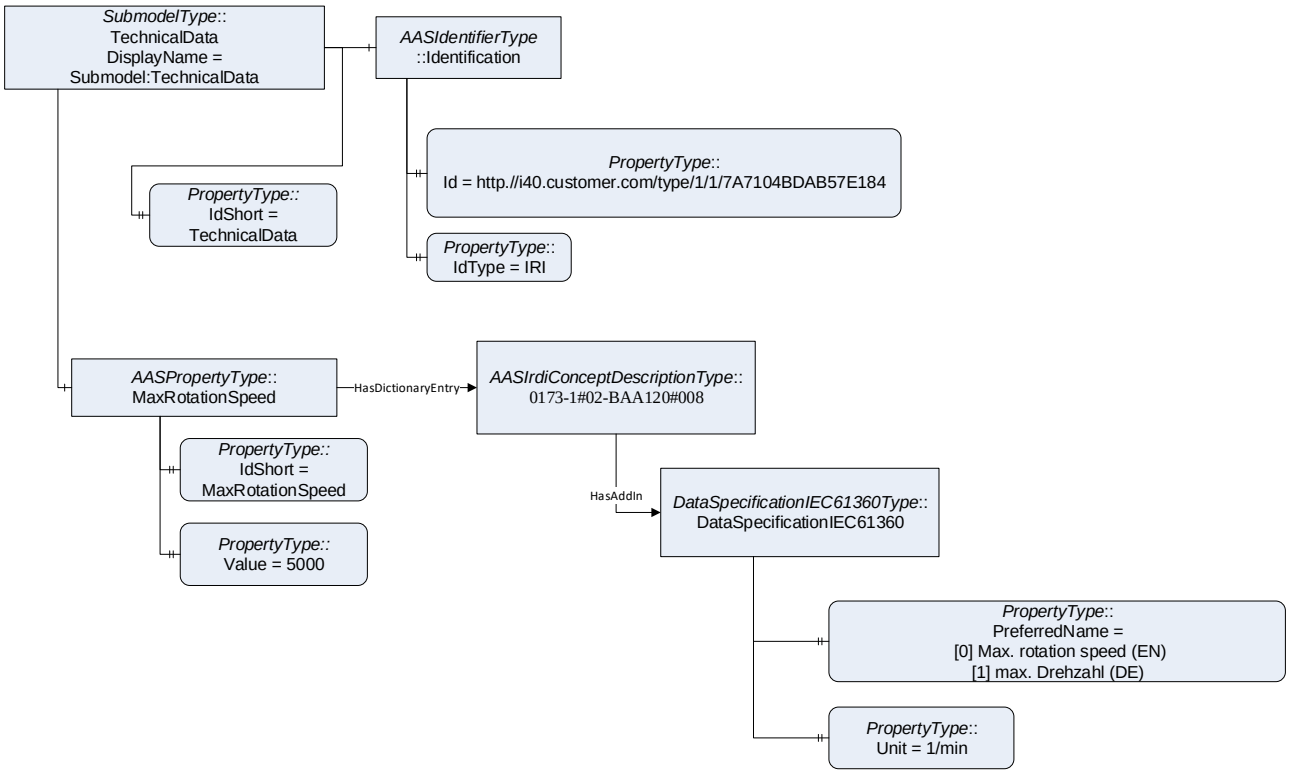
#	Display Name	Value	Datatype
1	Category	PROPERTY	String
2	Keys		Null
3	Id	0173-1#02-BAA120#008	String
4	IdType	IRDI	String
5	Keys		Null
6	DataType	INTEGER_MEASURE	String
7	Definition	Double click to display value	LocalizedText
8	PreferredName	Double click to display value	LocalizedText
9	ShortName		String
10	Unit	1/min	String
11	Revision	2	String
12	Version		String
13	Keys	{'(GlobalReference)(no-local)[IRDI]0173-1#05-AAA650#002'}	String

5.7.6 Example Property of a Submodel with SemanticId

In the following (Figure 81) the example from Clause 5.7.4 is extended and now also includes a semantic reference to a concept description ofr MaxRotationSpeed (modelled via "*HasDictionaryEntry*" reference type of OPC UA).

The property with idShort and thus browse name *MaxRotationSpeed* has a reference to a dictionary entry with IRDI "0173-1#02-BAA120#008". Since the semanticId is an IRDI the *AASIrdiConceptDescriptionType* is instantiated. *AASIrdiConceptDescriptionType* is inheriting from the OPC UA Object Type „*IrdiDictionaryEntryType*" as defined in the Amendment for Dictionary Entries.

Figure 81 Example MaxRotationSpeed Property as part of Submodel TechnicalData



5.7.7 Examples Submodel Element Collections

SubmodelElementCollection is mapped to AASOrderedSubmodelElementCollectionType if ordered=True and AASSubmodelElementCollection if ordered=False (see Figure 82).

An Example for a collection is shown in Figure 83.

Figure 82 Structure CollectionType and OrderedCollectionType

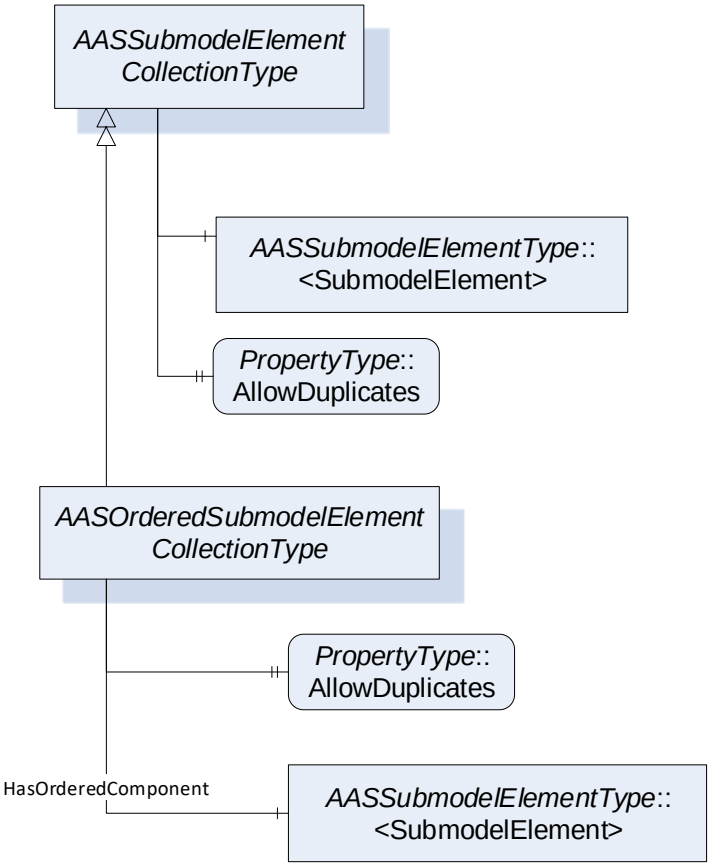
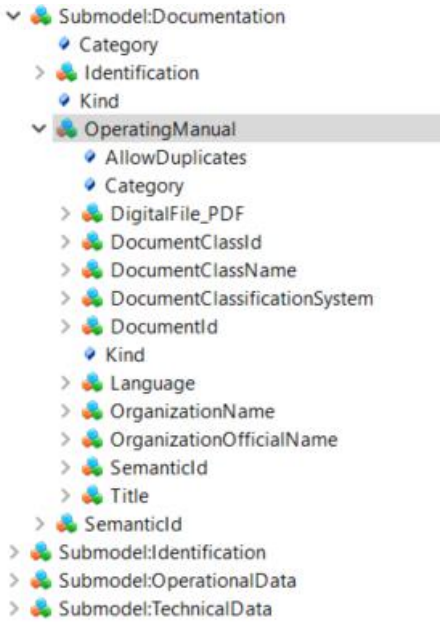


Figure 83 Example Submodel Documentation with Collection for OperatingManual



5.7.8 Example Asset

Asset as modelled as a normal component of the AAS object (see Figure 86). Figure 84 shows as an example an Asset with idShort “ServoDCMotor” with its reference to the assetIdentificationModel. Figure 85 shows the same example in UAExpert.

Figure 84 Example ServoDCMotor as Asset

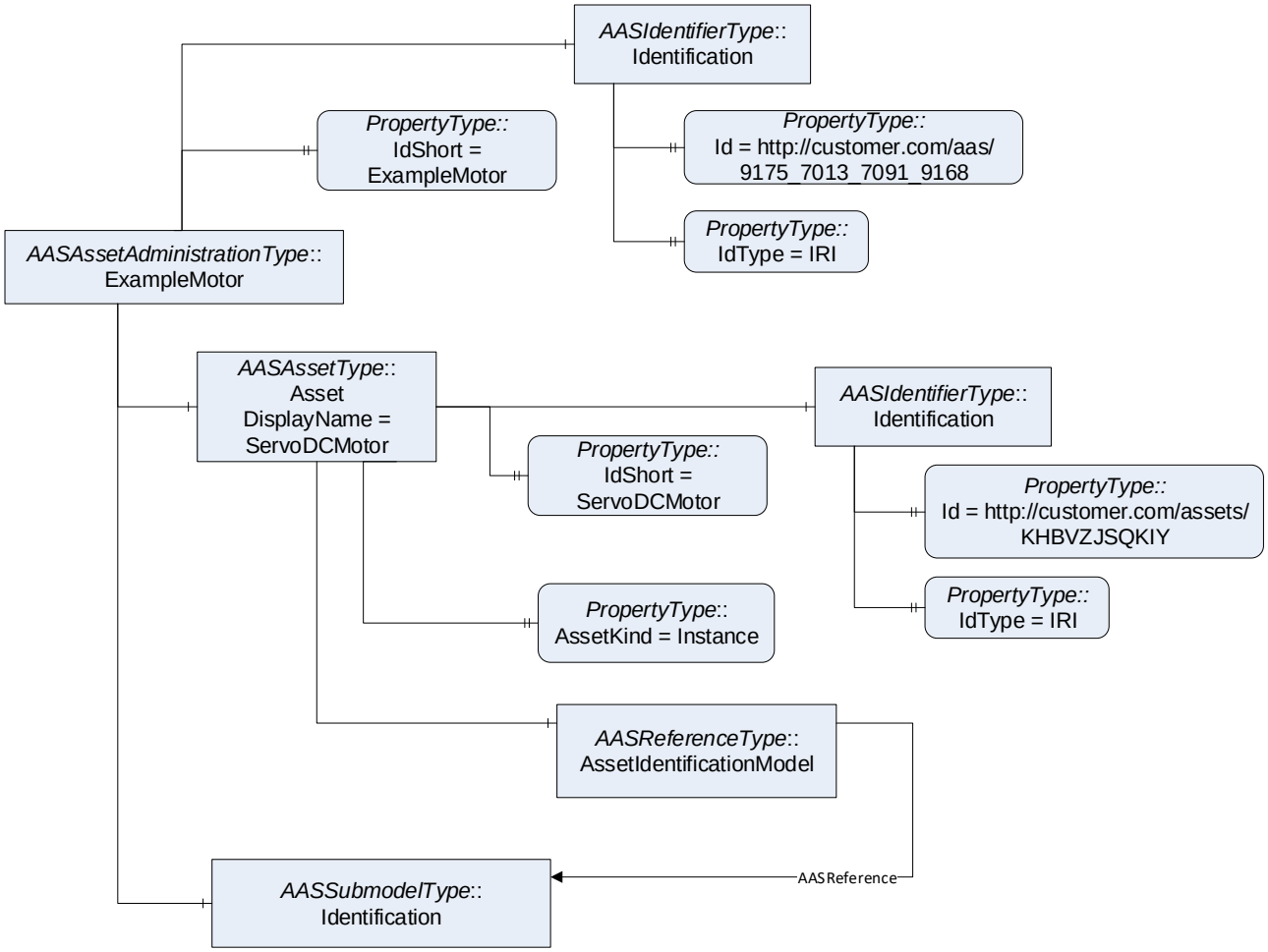


Figure 85 Example UAExpert Asset ServoMotor

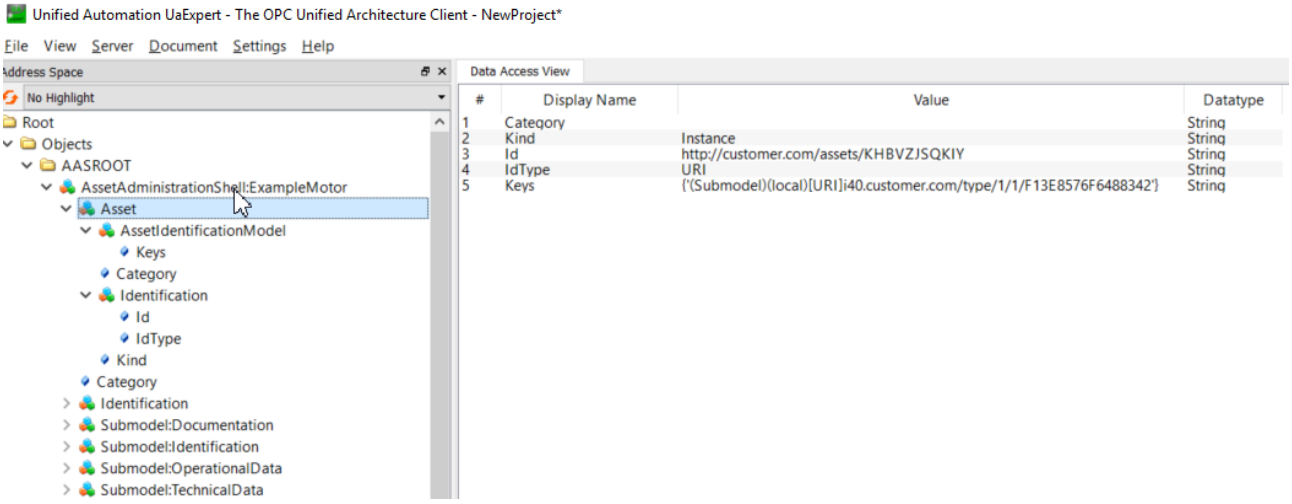
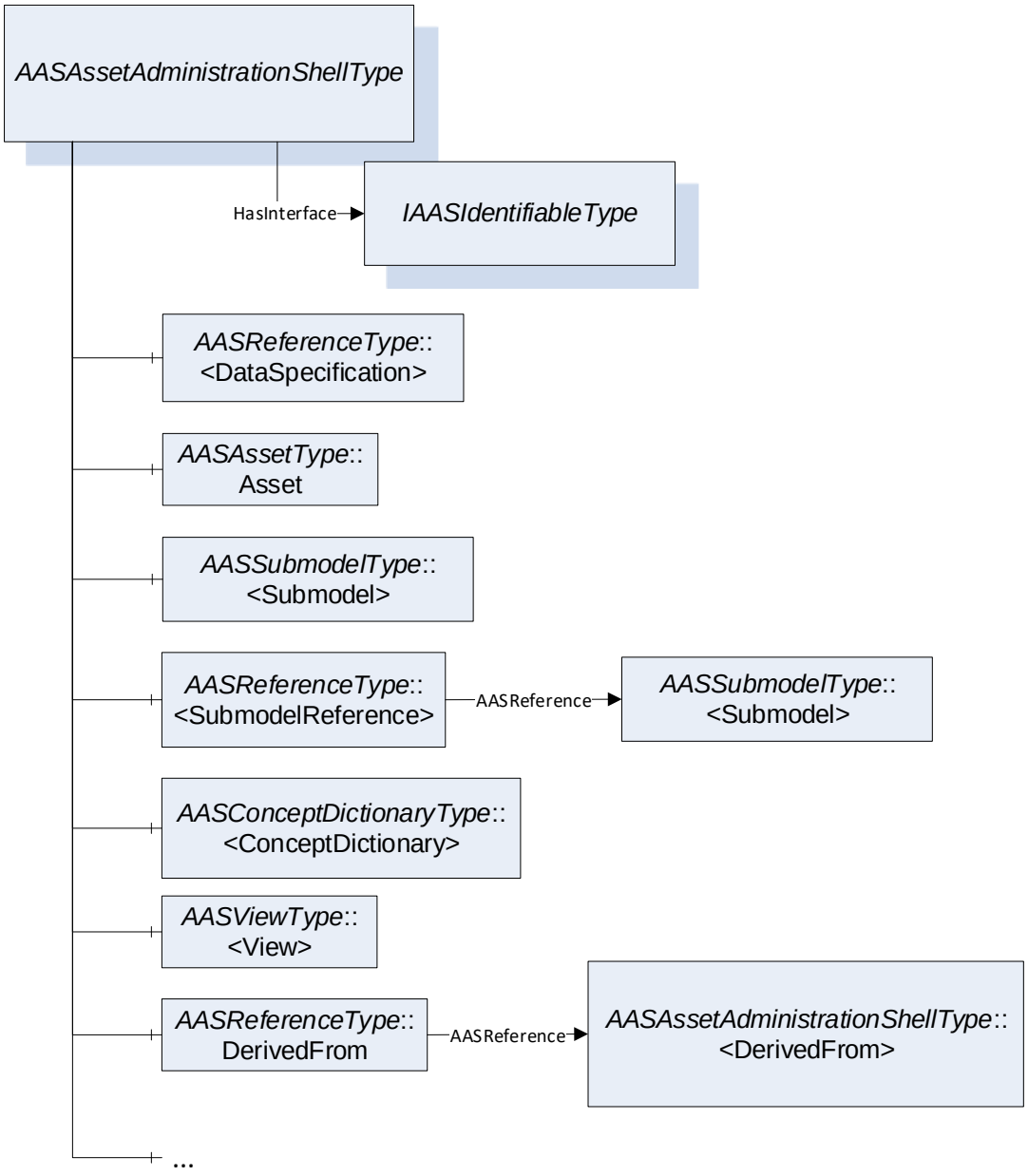


Figure 86 Structure Asset Administration Shell with Asset and Submodels



5.7.9 Example File

In Figure 87 the OPC UA types for *Blob* and *File* submodel elements are shown (inheritance from *AASSubmodelElement-Type* is not shown, see Figure 75). In Figure 88 an example how to model a submodel element "*File*" is shown for a documentation submodel conformant to VDI2770 containing an OperationManual. The OPC type "*FileType*" is used for modelling the file itself. In Figure 89 the same example is shown in UAExpert.

Figure 87 OPC UA Types for Submodel Elements File and Blob

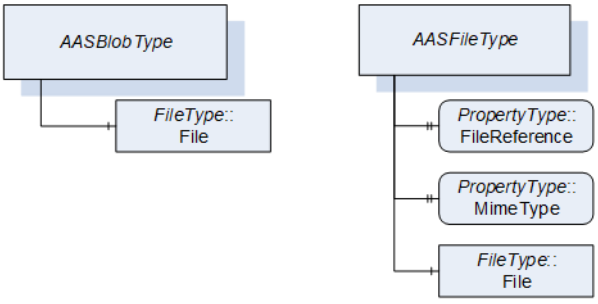


Figure 88 Example OperationManual as AASFileType

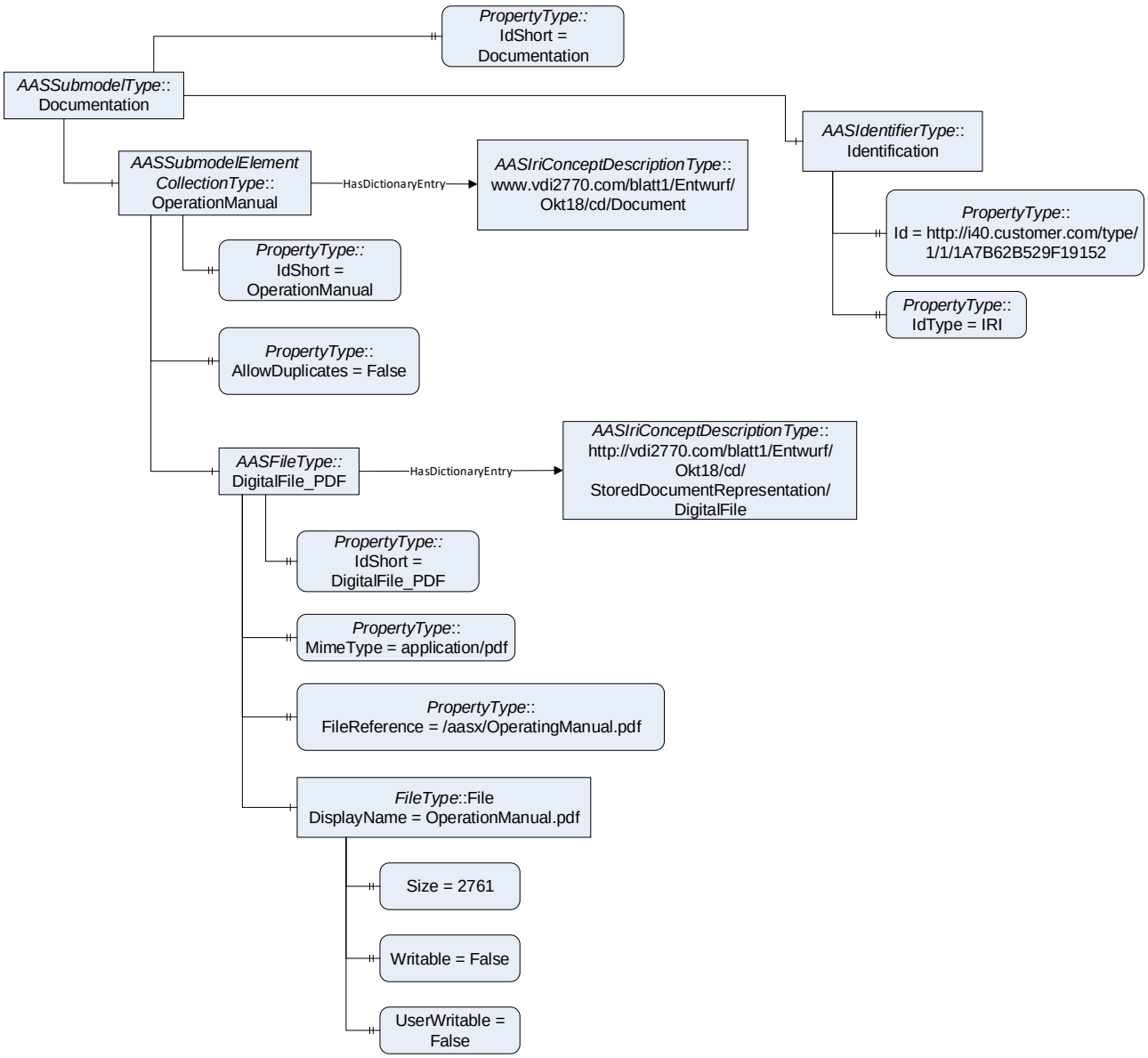
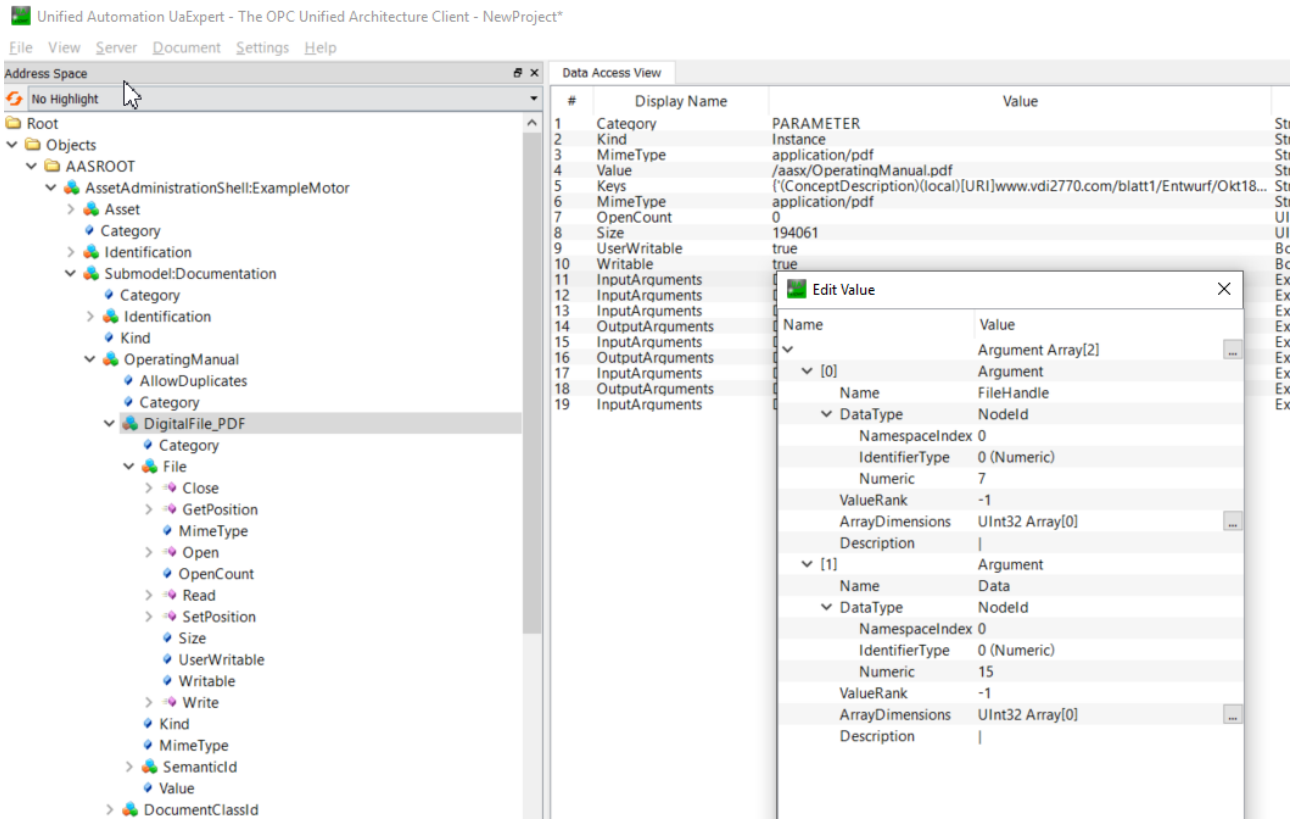


Figure 89 Example OperatingManual File in UAExpert



5.7.10 Example Operation and Capabilities

In Figure 90 an operation “Scan” is shown as defined in the OPC UA Companion Specification for AutoId.

It is visible that methods are not directy contained as part of the submodel but as part of an Instance of “OperationType” first. This is because on the one side to be consistent with the other submodel elements and on the other side otherwise no *DictionaryEntry* reference can be added: this is not forseen so far in OPC UA Amendment for Dictionary Entries.

The predefined reference type for methods “*HasArgumentDescription*” is used to describe the semantics of the input and output variables. The references variable with the description has the “HasDictionaryEntry” representing the semanticId information.

In Figure 91 a different modelling is shown, not yet using the predefined reference type “*HasArgumentDescription*” but just explicitly modelling the input and output variables as components.

Figure 90 Example Operation Scan

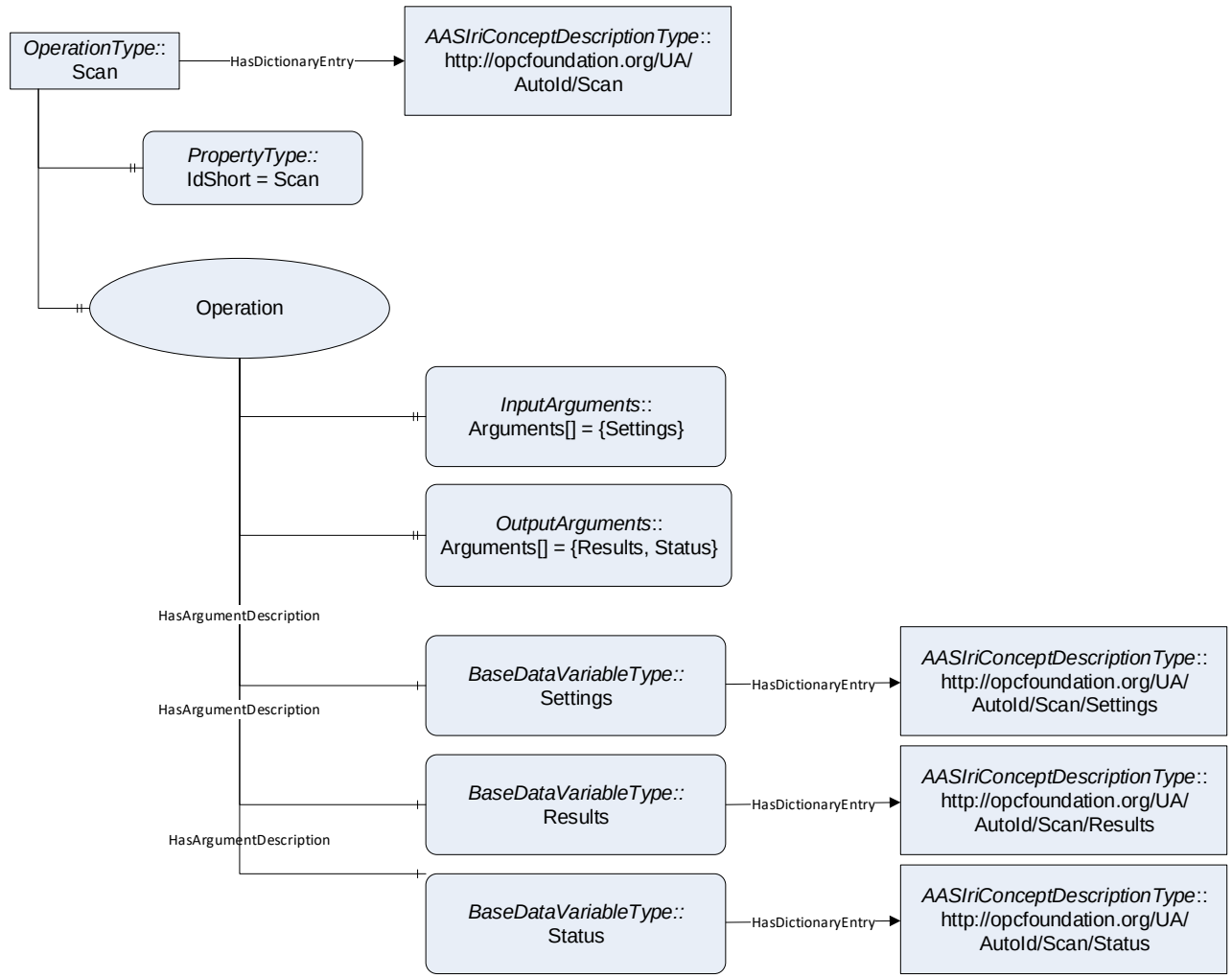
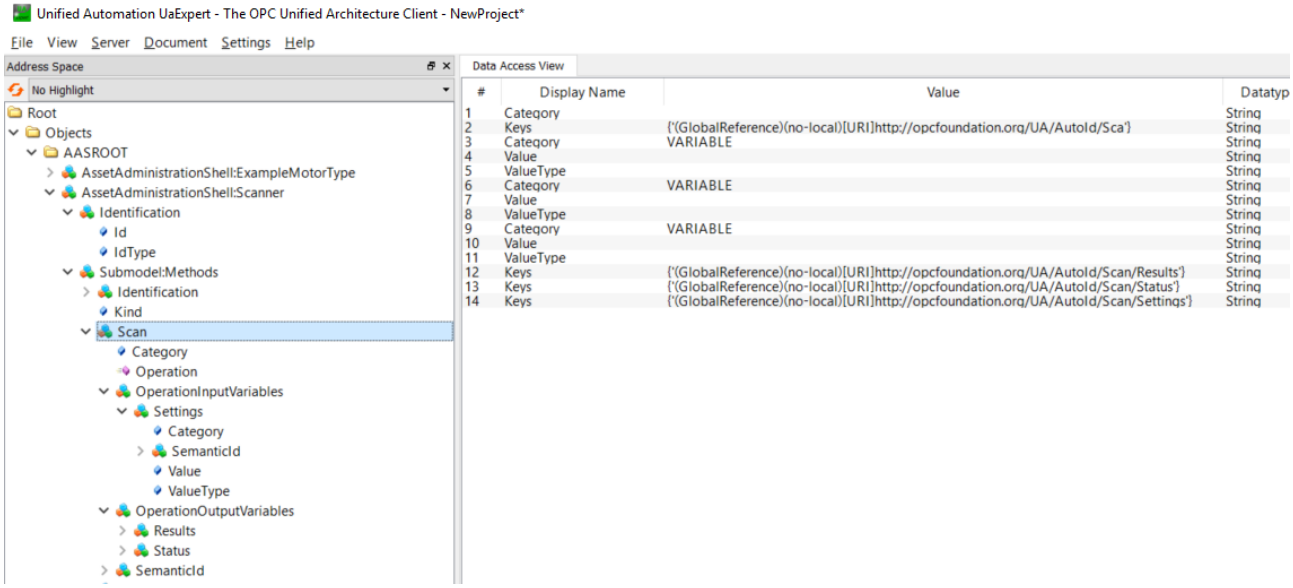


Figure 91 Example Operation Scan in UAExpert

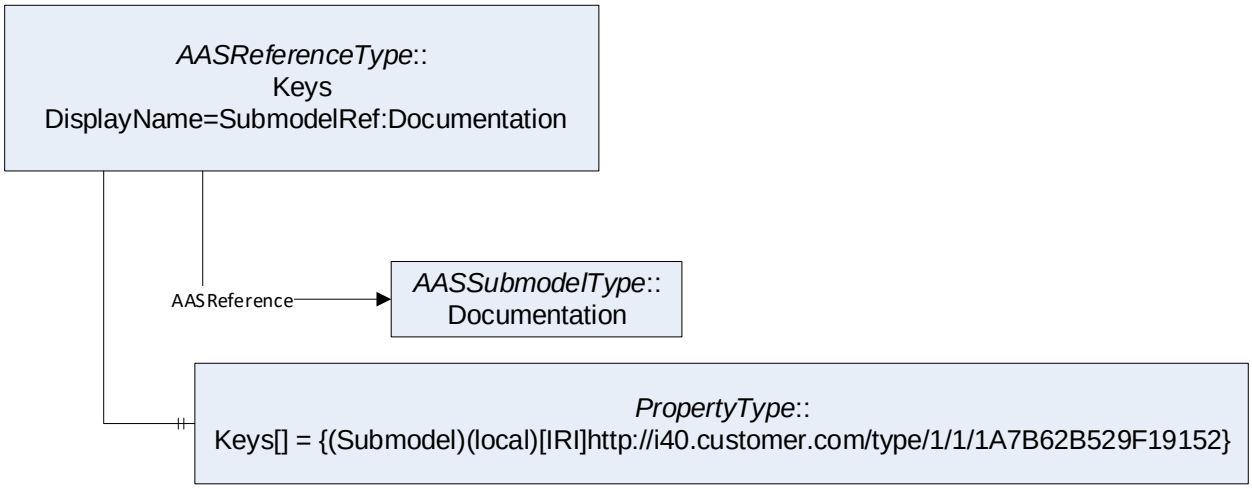


5.7.11 Example References

In Figure 92 an example is show how to use references. References are for example used in ReferenceElements and in RelationshipElements. For this there is a predefined *AASReference* reference type defined that is used for references to

local elements (*Key/local*=True). Local in this case means the referenced object is on the same OPC UA Server as the AAS. For global references (Key/local=False) no *AASReference* can be used. In this example the global reference is "http://i40.customer.com/type/1/1/1A7B62B529F19152".

Figure 92 Example References shown for the reference to a submodel



5.7.12 Example Qualifier

In Figure 93 the OPC UA Type for Qualifier is shown.

In Figure 93 a Submodel "TechnicalData" with the qualifier of type "life cycle qual" and value "SPEC" for "as specified" is shown.. Normally the name of an OPC UA element corresponds to the *idShort* within the AAS metamodel. However, qualifiers do not have an *idShort* because they are no referables. Therefore the name in OPC UA needs to be created, it is "LifeCycle=SPEC".

In Figure 95 the example is shown in UA Expert[28].

[28] In the example the name of the submodel is „TechnicalData_SPEC".

Figure 93 Qualifier Type in OPC UA

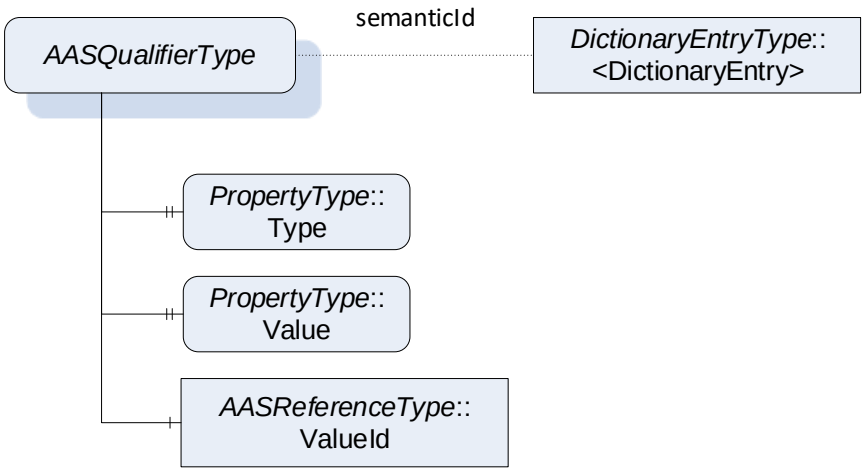


Figure 94 Example Qualifiers for Submodel

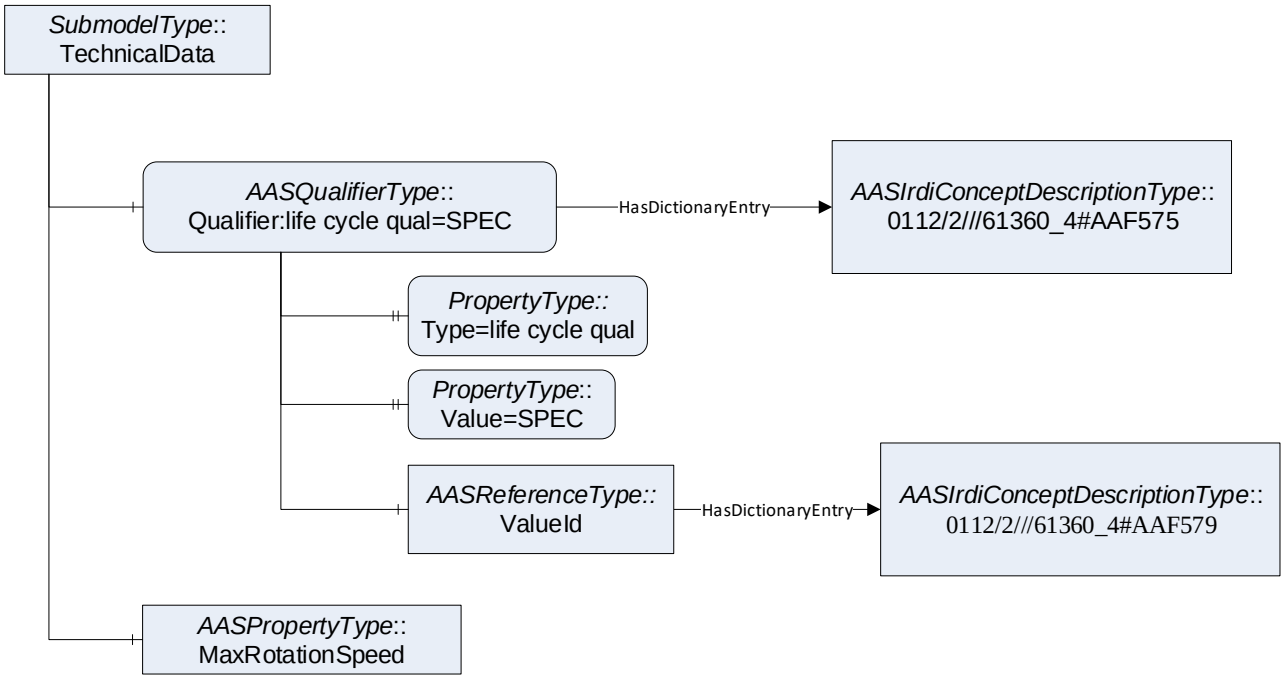
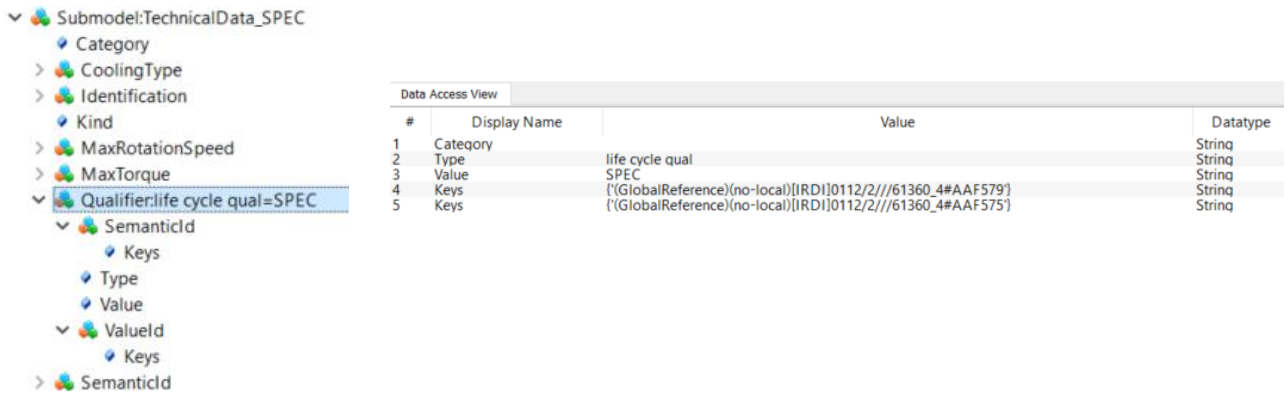


Figure 95 Example Lifecycle Qualifier for Submodel Techncial Data



5.7.13 Example Concept Description

In Figure 96 Example ConceptDescription Max. Rotation Speed (Extract) it is shown how to add the semanticId to a property. In this example the reference “HasDictionaryEntry” references the concept description. The difference between an external reference and a local reference to a concept dictionary is only visible looking at the addIns used to describe the attributes of a dictionary entry. The corresponding extract that can be referenced by any element that can have a dictionaryEntry reference is shown in Figure 96. A concrete example how it is realized in an OPC UA Server is shown in Figure 97.

Figure 96 Example ConceptDescription Max. Rotation Speed (Extract)

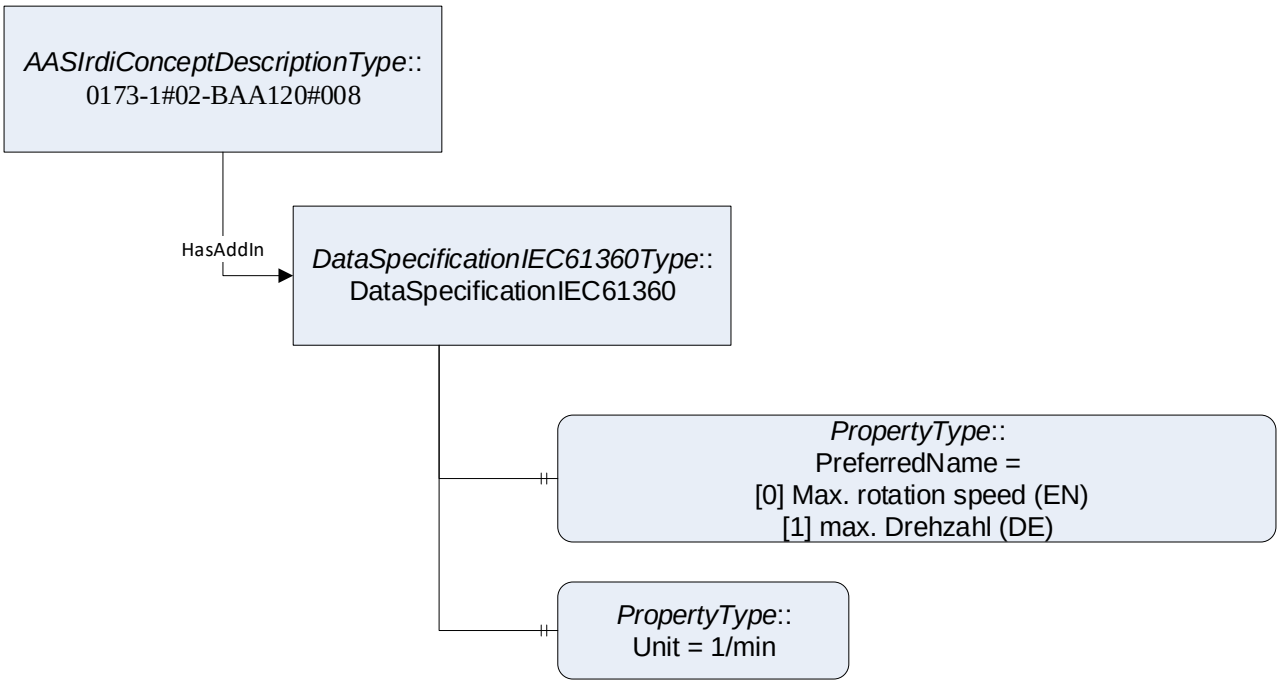
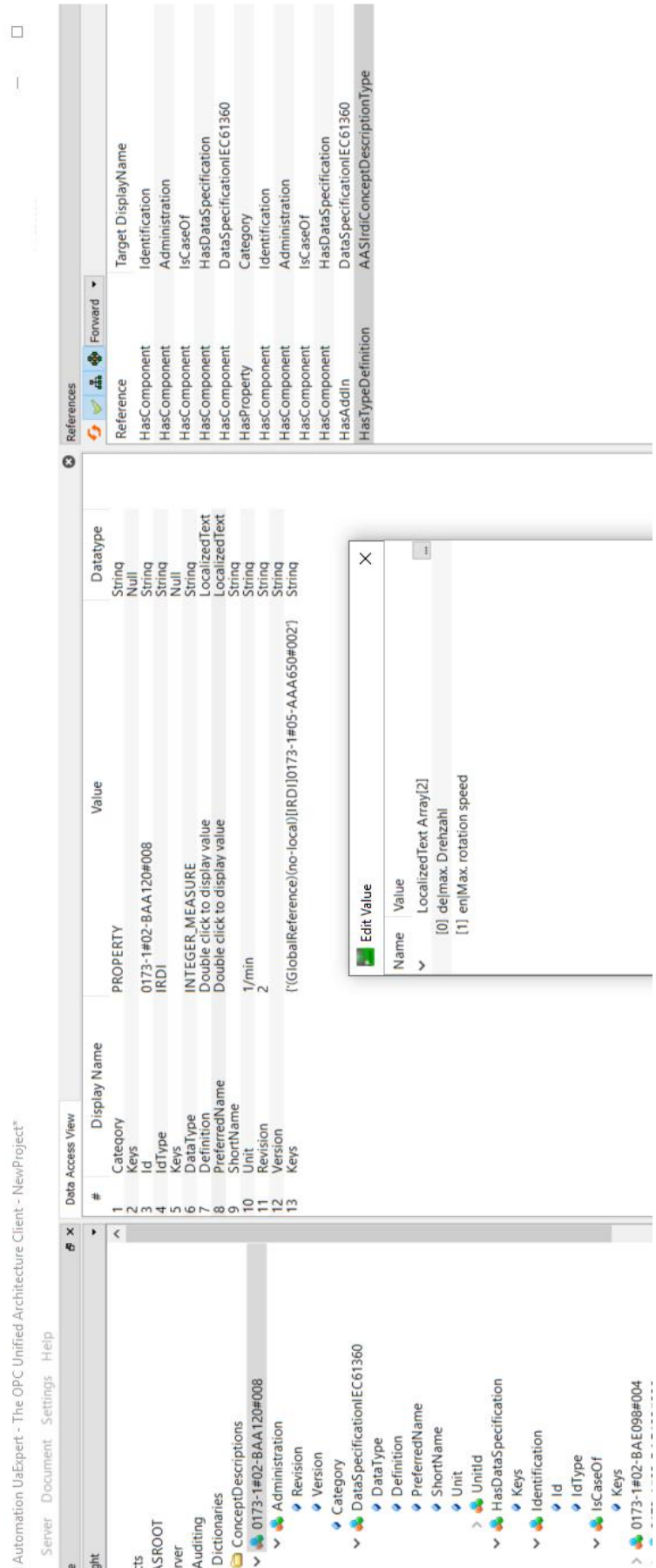


Figure 97 ConceptDescription MaxRoationSpeed in OPC UA Server

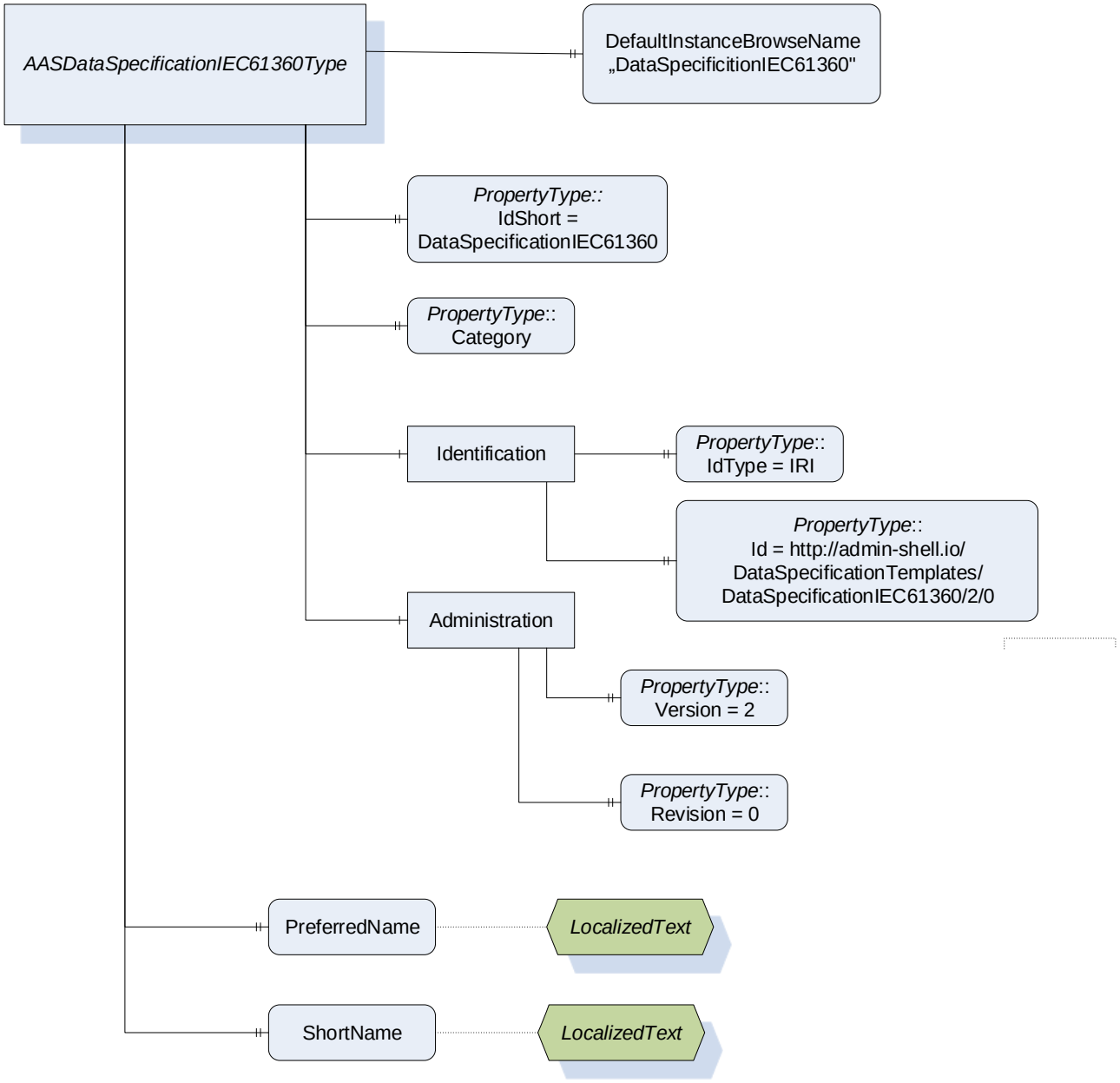


5.7.14 Example Data Specification

In Figure 98 the data specification template for IEC61360 is defined. In this case the id and idShort are fixed. It also has a version. Additionally the Browse Name is fixed since it is used as addIn (using the reference type "*HasAddIn*" from OPC UA).

The usage of the template was discussed in Clause 5.7.13.

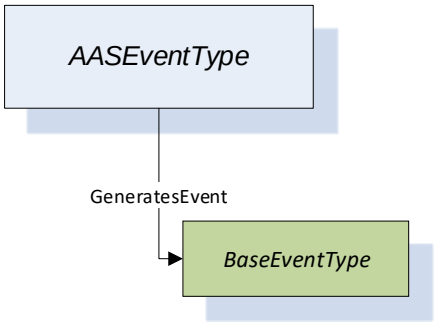
Figure 98 Example Data Specification Template IEC61360



5.7.15 Example Event

In Figure 99 the event type in OPC UA is shown. It uses the Event Mechanism of OPC UA.

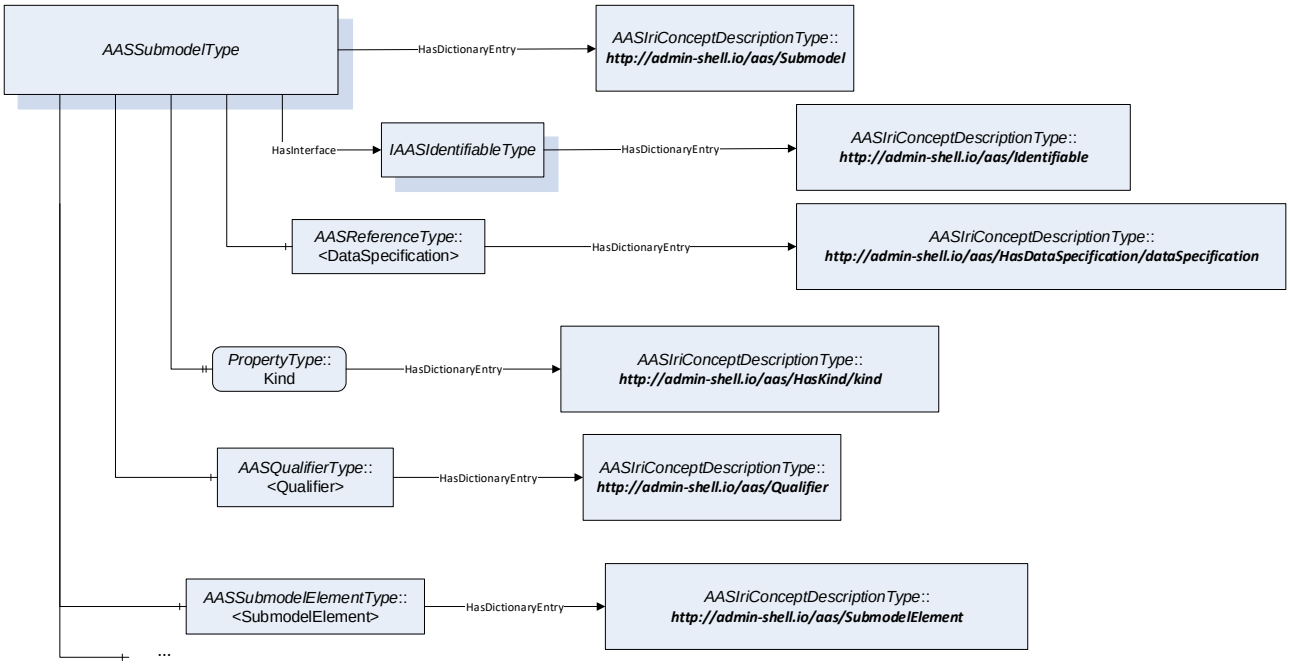
Figure 99 Event Type in OPC UA



5.7.16 Example HasDictionaryEntry for Model

In Clause 5.7.6 it was shown how to describe the semantics of an instance of an OPC UA type. Using the same reference type “HasDictionaryEntry” also the semantic of the types, variables etc. themselves can be described. This is shown for the submodel in Figure 100.

Figure 100 HasDictionaryEntry for Submodel



5.8 AutomationML

5.8.1 General

For import and export scenarios the metamodel of an Asset Administration Shell needs to be serialized. In the following clauses an overview of the main concepts of the Asset Administration Shell serialization using AutomationML (IEC 62714) are presented. As a serialization format, AutomationML is especially suitable for the engineering phase.

In general the serialization approach is to map each object of the Asset Administration Shell metamodel to an AutomationML Role Class or to an AutomationML Role Class accompanied by an AutomationML Interface Class. This Role Class and (if applied) Interface Class then also define the required attributes in AutomationML.

Asset Administration Shells itself shall be modelled as AutomationML System Unit Classes or as Internal Elements within an Instance Hierarchy depending of the kind information of type and instance.

For the Role Classes and Interface Classes that are required for the serialization an AutomationML Role Class Library resp. an Interface Class Library are defined and provided to the public.

One of the goals is to ensure that the AutomationML model of the Asset Administration Shell can be used as a standalone AutomationML model as well as in combination with existing AutomationML models such as the upcoming AutomationML Component Description. Therefore, the definition of the serialization approach defined in this clause is interleaved with the AutomationML definitions and applies the AutomationML technology definitions widely on https://www.automationml.org/o.red.c/dateien.html

The example is shown in tool "AutomationML Editor" of AutomationML e.V. 2015.

5.8.2 Rules

The rules for mapping of the AAS information model to AutomationML information model are given in the following. In subsequent clauses examples for the rules are given. For reasons of simplicity the term AML element is used for either *InternalElements* or for *SystemUnitClasses* or interfaces (or all of them) depending on the context.

Generic Rules for mapping:

(1) If present, AML role class and attribute name are taken from the AAS metamodel.

(2) If present, AML element names are the same as the value of i*dShort* information from the AAS. If not present, a sufficiently unique element name is to be generated.

(3) **Attributes of AAS Classes are modelled as attributes of AML elements.**
The morphology of the AML and AAS information is in principle the same.

(4) Semantics of AML attributes is given by the AML *RefSemantic* attribute. Semantics of AML elements are defined via AML role and interface classes.

The values of RefSemantic follow the rules as described in Clause 5.2.2.

Example for RefSemantic values: *AAS:Qualifier*, *AAS:Qualifier/type*, *AAS:Qualifier/value*
For a complete list of all RefSemantic values see Annex H.iii.
Note: RefSemantic is not identical to semanticId in AAS. The difference is explained in Clause 5.8.8.

(5) **Attributes on AML Elements are created only if required.**
Attributes, such as category, are only created for AML elements, if values are present in the AAS and vice versa.

(6) Values of Attributes in AAS which are of type "Reference" are serialized as string.

- The rules for serialization can be found in Clause 5.2.1
- Example: *(Submodel)(local)[IRI]http://www.myuri.de*
- **Every internal element that represents information of the AAS metamodel.** A Tool with AASX-Import would search only for the AAS roles in the AML file. However, an AutomationML tool could also export/write a file containing AAS roles and other roles/models.

Rules for elements other than SubmodelElements of the AAS:

(7) **Qualifiers** are mapped on instance level to a complex attribute in AML

- Name of top-most hierarchy attribute: **qualifier:<value of AAS:Qualifier/type>=<value of Qualifier/value>**. Example: "*Qualifier:PredicateRelation=GREATER_THAN_0*"
- Subordinate attributes of the qualifier are mapped to subordinate attributes of the attribute in AML, e.g. for qualifier type, qualifier value with according AML *RefSemantic*.

(8) **View as *InternalElement* with *RoleClass* View groups mirror elements**
The internal elements with the assigned role class *"View"* group one to many mirror elements as a child. The target of the mirror element is the AAS entity identified by the AAS View as a contained element. A mirror element points with it's *RefBaseSystemUnitPath* attribute to the UUID of the corresponding AML element.

Rules for subtypes of AAS SubmodelElement:

(9) All **AAS SubmodelElements** are mapped to AML InternalElements with an associated role class equal to the respective *SubmodelElement* subtype (e.g. AAS Property to AML Role Class Property).

(10) For submodel element **File** an interface **FileDataReference with it's predefined attributes refURI and MIMEType** is used for referencing the file. Attribute value of submodel element File is not needed.

(11) For submodel elements **ReferenceElement** and **RelationshipElement** the interface "**ReferableReference**" is used to reference to the corresponding objects. In case of a local reference (AAS:Key/local = True) the interface "*ReferableReference*" needs to be set and is pointing to the corresponding element within AutomationML via an InternalLink. In this case, the value is empty. In case of an external reference (*AAS:Key/local* = False) no InternalLink is set for the interface. Instead, the *value* attribute carries a serialization of the AAS Reference.

(12) For **Operation**, the Operation is mapped to an InternalElement in AML with Role Class *Operation*.

- *Input~/output~/inoutputVariable* attributes of an AAS Operation are mapped to *InternalElements* in AML with RoleClass *OperationInputVariables*, *OperationOutputVariables* resp. *OperationInoutputVariables* . These *InternalElements* contain subordinated *InternalElements* for the submodel elements.

(13) **SubmodelElementCollection** is mapped to an *InternalElement* in AML with *RoleClass SubmodelElementCollection*. It contains subordinated *InternalElements* for contained elements.

Rules for the instance hierarchy of AML:

(14) For an AASX Tool with AML Export typically an instance hierarchy needs to be created. This instance hierarchy needs a name. If there is no other naming convention or no existing instance hierarchy that shall be used a possible name for the InstanceHierarchy is "*AssetAdministrationShellInstanceHierarchy*". It contains the asset administration shells containing elements of *kind*=Instance. Elements within this hierarchy have the role "*AssetAdministrationShell*".
Note: The AML import to an AASX Tool just needs to look for the role "*AssetAdministrationShell*", not for the names of the instance hierarchies.

(15) For an AAS with concept descriptions an AASX Tool with AML Export needs to create an instance for the concept descriptions. If there is no other naming convention or no existing concept description instance hierarchy then a possible name for the *InstanceHierarchy* is"*AssetAdministrationShellConceptDescriptions*". It contains the concept descriptions used or that can be used within the asset administration shells. The role of the elements contained in this instance hierarchy is "*ConceptDescription*".
Note: The AML import to an AASX Tool just needs to look for the role "*ConceptDescription*", not for the names of the instance hierarchies.

(16) For each AAS related element within an instance with role "*AssetAdministrationShell*" the corresponding role within *AssetAdministrationShellRoleClassLib* is assigned.

(17) In case of an AAS element having a data specification additionally an instance of the corresponding template with Role **DataSpecificationConcent** within the corresponding S*ystemUnitClassLib* is assigned. Thus, all attributes defined within the template are additionally predefined for the element.

(18) The name of the element within a concept description being instantiated by a data specification template, i.e. an element with role "*DataSpecifiationConent*", is "*EmbeddedDataSpecification*". If more than one data specification template is used then the element containing the different elements has the name "***EmbeddedDataSpecifications***" (no role) and its sub-elements are names "***EmbeddedDataSpecificication_<Number>*"** because the names need to be unique.

(19) In case of a concept description AAS has a predefined data specification template (Role *DataSpecification*) called "*DataSpecificationIEC61360Template*". This template is used for a concept description by instantiating its content, i.e. element with role "*DataSpecificationContent"*.

(20) The name of the elements within a library needs to be unqiue. This is why for concept descriptions within the library for concept descriptions the name is choosen as follows:
<value of AAS:ConceptDescription/idShort>__<value of AAS:ConceptDescription/identification/idType>_<value of AAS:ConceptDescription/identification/id>

Rules for the Role Class Library of AML:

(21) There is a predefined *RoleClassLibrary* with name "*AssetAdministrationShellRoleClassLib*". It contains all roles specific for asset administration shells.

(22) All AAS referables (and thus identifiables) are mapped to specific Role Classes in AML. The name is identical to the name in AAS.

(23) A small number of role classes are required for entities that have cardinality > 1 and different names like "*OperationInputVariables*", "*OperationOutputVariables*". and "*OperationInoutputVariables* " of Operation

Rules for System Unit Class Libraries:

(24) For an AASX Tool with AML Export a system unit class library needs to be created if the AASX contains submodels or submodel elements of kind=Template. This system unit class library needs a name. If there is no other naming convention or no other system unit class lib that can be used a possible name for the library is "*AssetAdministrationShellSystemUnitClasses*". It contains the asset administration shells containing elements of *kind*=Type. Elements within this library have the role "*AssetAdministrationShell*".
Note: The AML import to an AASX Tool just needs to look for the role "*AssetAdministrationShell*", not for the names of the system unit class libraries.

(25) If an AAS contains a submodel of *kind*=Template, then a corresponding *SystemUnitClass* is created for:
- the AAS itself, and
- the submodels within the AAS with kind=Template

(26) The same roles as for AAS with submodels of kind=Instance are assigned.

(27) There is a predefined *SystemUnitClassLibrary* with name "*AssetAdministrationShellDataSpecification-Templates*".
It contains the predefined data specification templates as defined in Asset Administration Shell in Detail. These data specifications have the role "DataSpecification".

(28) A Data Specification has an internal element with role "*DataSpecificationContent*" defining all the attributes available when using the data specification.

(29) In case there is the need to assign more than one data specification template to an element a system unit class containing the needed data specification content elements needs to be defined**.**

Rules for Interface Libraries:

(1) There is a predefined InterfaceClassLibrary with name "***AssetAdministrationShellInterfaceClassLib***". It contains the predefined interfaces used within the asset administration shell.

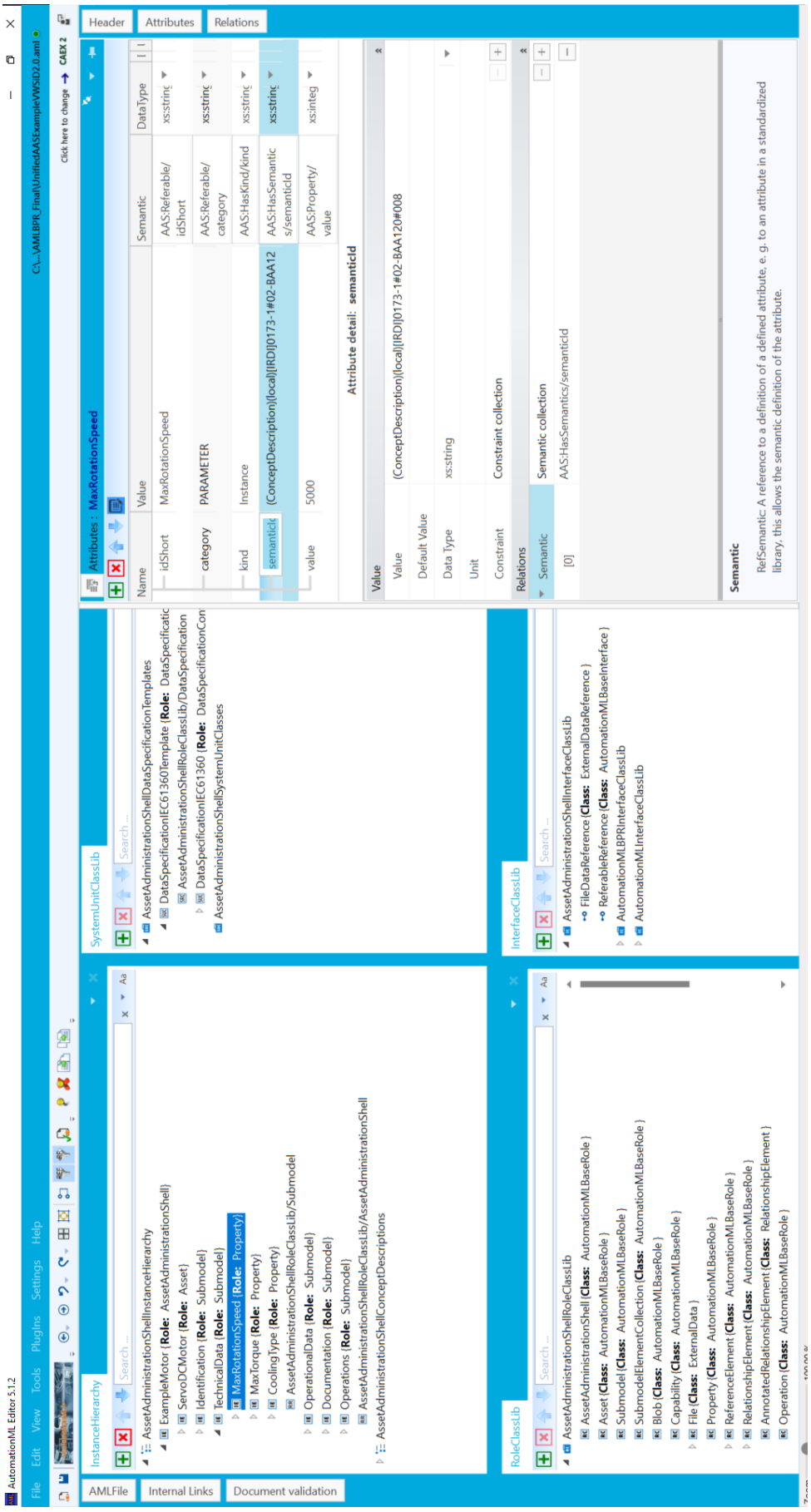
(2) *FileDataReference* is an interface derived from the AutomationML interface "***ExternalDataReference***". It is used as interface for submodel element "File".

(3) ***ReferableReference*** is an interface for realizing references as used within submodel elements "*ReferenceElement*" and "*RelationshipElement*" (and its subclasses) within asset administration shells

5.8.3 Example Overview

In Figure 101 the example is shown in the AutomationML Editor. Details are explained in the following subclauses.

Figure 101 Example in Automation ML Editor



5.8.4 Example Property and Concept Description

Figure 102 shows the property “MaxRotationSpeed”.

Please note: the Unit “1/min”could have been added to the AML field “Unit” of attribute value. However, it is available only indirectly via its semanticId (see Figure 103).

The value of attribute “semanticId” used the reference serialization as defined in Clause 5.2.1.

Figure 102 Example Property MaxRotationSpeed

Attributes : MaxRotationSpeed

Name	Value	Semantic	DataType
idShort	MaxRotationSpeed	AAS:Referable/idShort	xs:string
category	PARAMETER	AAS:Referable/category	xs:string
kind	Instance	AAS:HasKind/kind	xs:string
semanticId	(ConceptDescription)(local)[IRDI]0173-1#02-BAA120#008	AAS:HasSemantics/ semanticId	xs:string
value	5000	AAS:Property/value	xs:integer

Figure 103 Example DataSpecificationContent of Concept Description MaxRotationSpeed conformant to template DataSpecificationIEC61360Template

Attributes

Attribute-Table

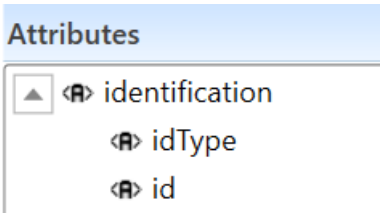
Name	Value	Unit
preferredName		
aml-lang=de	max. Drehzahl	
aml-lang=en	Max. rotation speed	
Attribute		
shortName		
unit	1/min	
unitId	(GlobalReference)(no-local)[IRDI] 0173-1#05-AAA650#002	
dataType	INTEGER_MEASURE	
definition		
aml-lang=de	Höchste zulässige Drehzahl, mit welcher der Motor oder die Speiseinheit betrieben werden darf	
aml-lang=en	Greatest permissible rotation speed with which the motor or feeding unit may be operated	

Header | Attributes | Internal Links

5.8.5 Example Attributes of Attributes

Complex Types are realized as attributes with sub-attributes (see Figure 104).

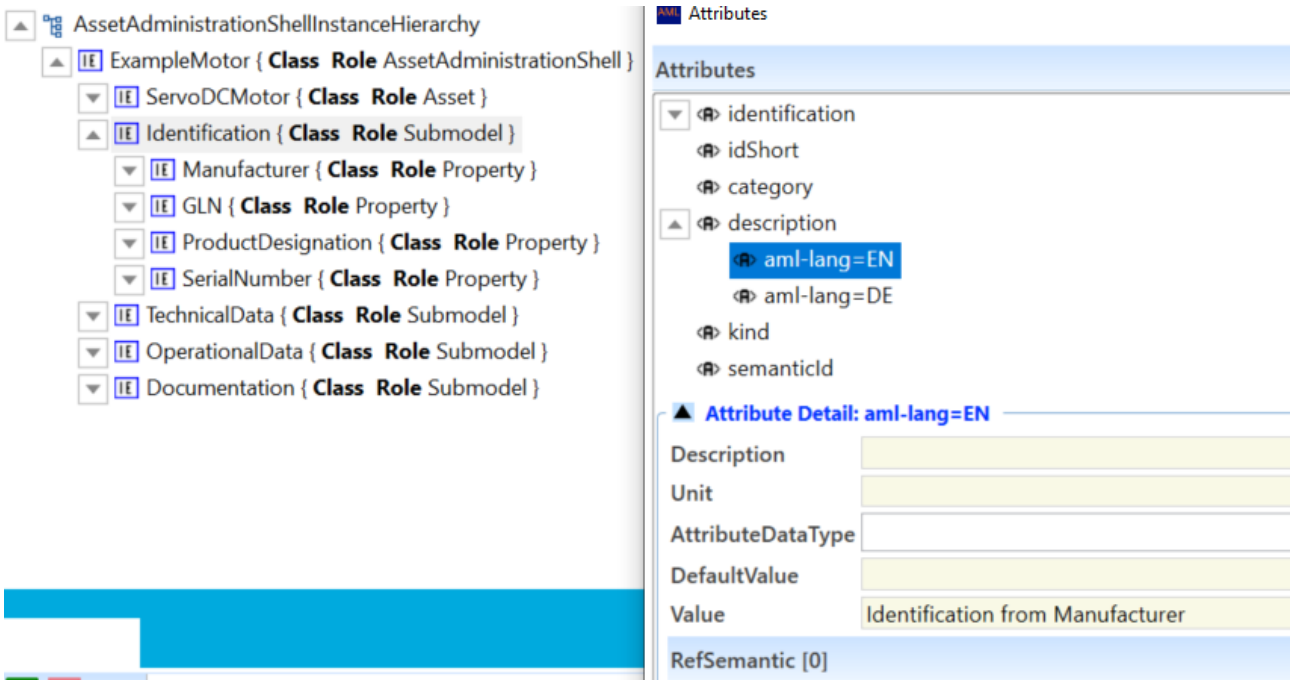
Figure 104 Example Identification with two subattributes



5.8.6 Example Language Tagged Strings

For attributes of type langString or LangStringSet the predefined AML attribute “aml-lang” is used (see Figure 105).

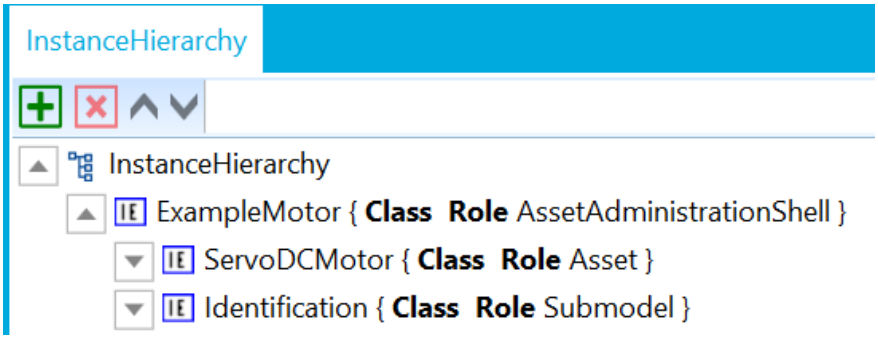
Figure 105 Example for attribute value in multiple languages



5.8.7 Example Asset

In Figure 106 the asset represented by an asset administration shell is shown: it has the role “Asset” assigned to it.

Figure 106 Example Asset in Instance Hierarchy



5.8.8 Example RefSemantic

Note: *RefSemantic* is an attribute of AutomationML whereas semanticId is an attribute of Asset Administration Shell.

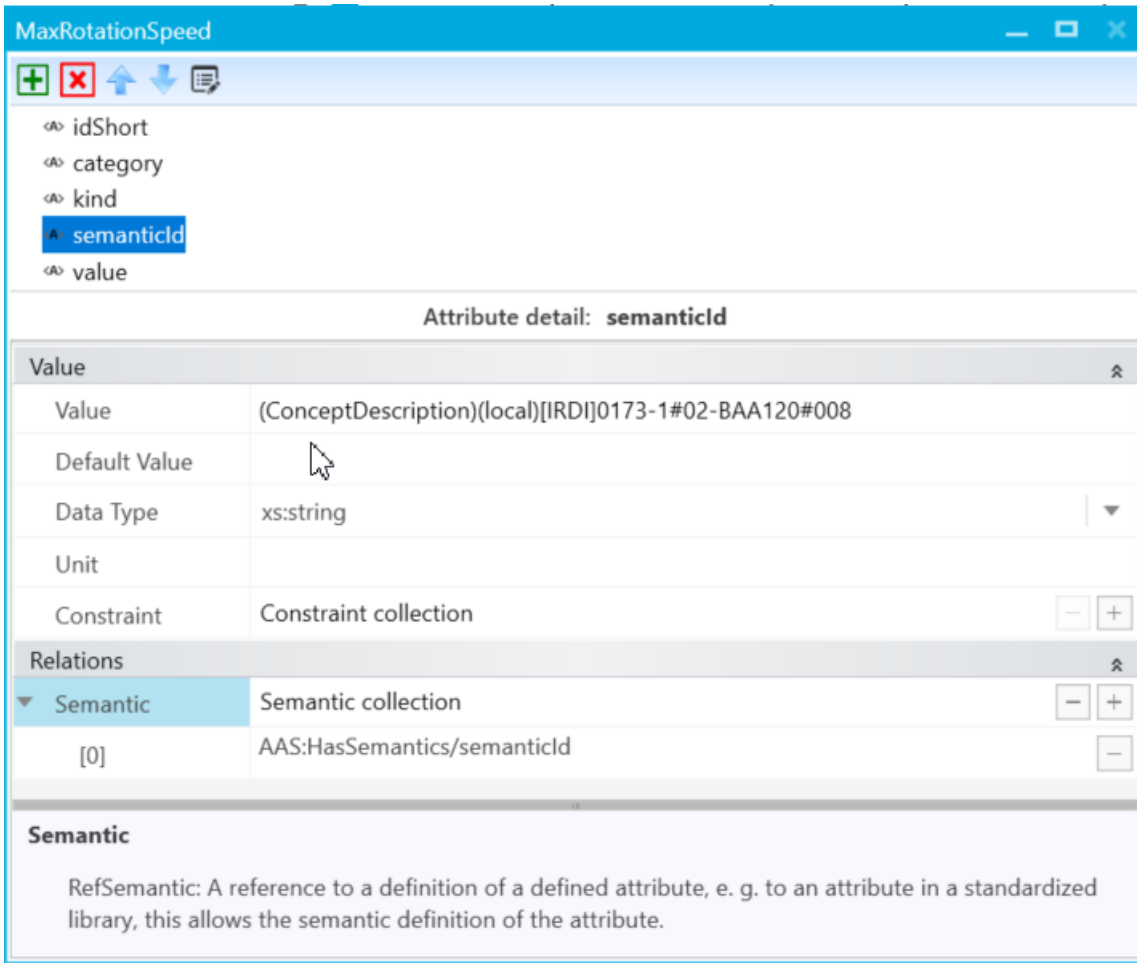
RefSemantic[29] describes the semantics of the metamodel. Thus the values reference to a description within the Asset Administration Shell Specification.

SemanticId describes the semantics of an instance.

RefSemantic for the AML attribute "semanticId" of an AML element "MaxRotationSpeed" for example says that it has the semantics as defined for the attribute "semanticId" within the AAS specification. The semanticId of the property says that it is the maximum rotation speed (see Figure 107).

For a complete list of all RefSemantic see Annex H.iii.

Figure 107 Example for RefSemantic and semanticId of the Property "MaxRotationSpeed"



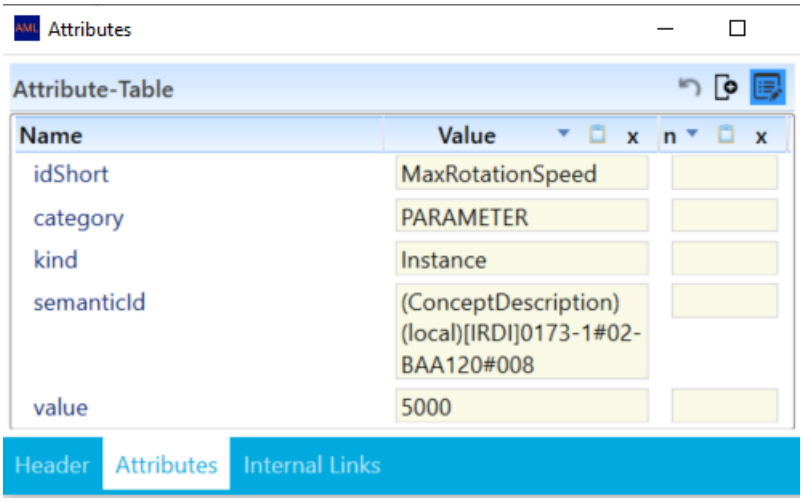
5.8.9 Example References

References are serialized into a single string. See example in Figure 108: The value of the semanticId is typed as "Reference" and is serialized as string.

Exceptions: ReferenceElement and RelationshipElement including its subtypes as well as Views.

[29] In the tooling it is sometimes just denoted as „Semantic".

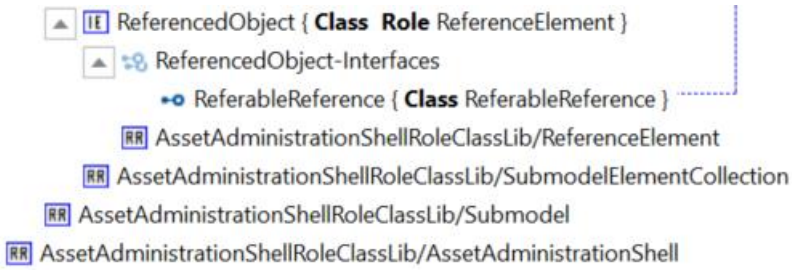
Figure 108 Example for serialized reference as value for attribute semanticId



5.8.10 Example ReferenceElement

In Figure 108 an example for a reference element is given: it references another element within the AAS by using the interface "ReferableReference". The blue dotted line represents a InternalLink to the target referenced element (not visible here).

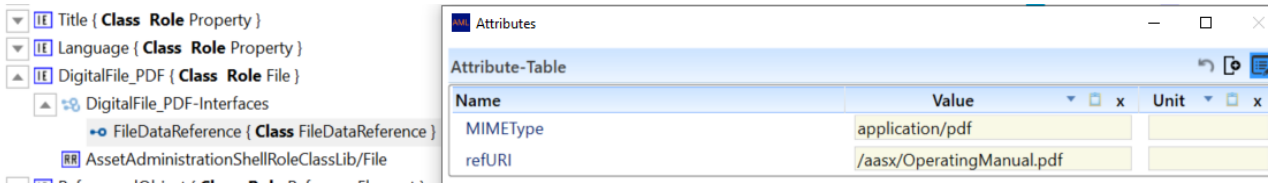
Figure 109 Example for ReferenceElement with Interface



5.8.11 Example File

In Figure 110 an AAS File submodel element is shown. It is realized with the predefined AML interface FileDataReference from Interface Class Library "AssetAdministrationShellInterfaceClassLib". It is derived from the AML interface "ExternalDataReference" from Interface Class Library "AutomationMLBPRInterfaceClassLib" that again is derived from the AML interface "ExternalDataConnector" as defined in the Interface Class Library "AutomationML-InterfaceClassLib". The interface *FileDataReference* already has a MIMEType inherited that is conformant to AAS:/File/mimeType.

Figure 110 Example File



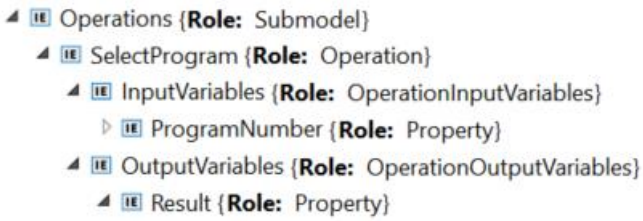
5.8.12 Example Operation

Operation is mapped to InternalElement in AML with Role Class Operation.

InoputVariables/OutputVariables/InoutputVariables are mapped to InternalElements in AML with RoleClass *Operation-InputVariables* resp. *~Output~*. These InternalElements contain subordinated *InternalElements* for the submodel elements.

The in or out element can be empty, i.e. does not need to have child elements.

Figure 111 Example Operation "SelectProgram" with input variables



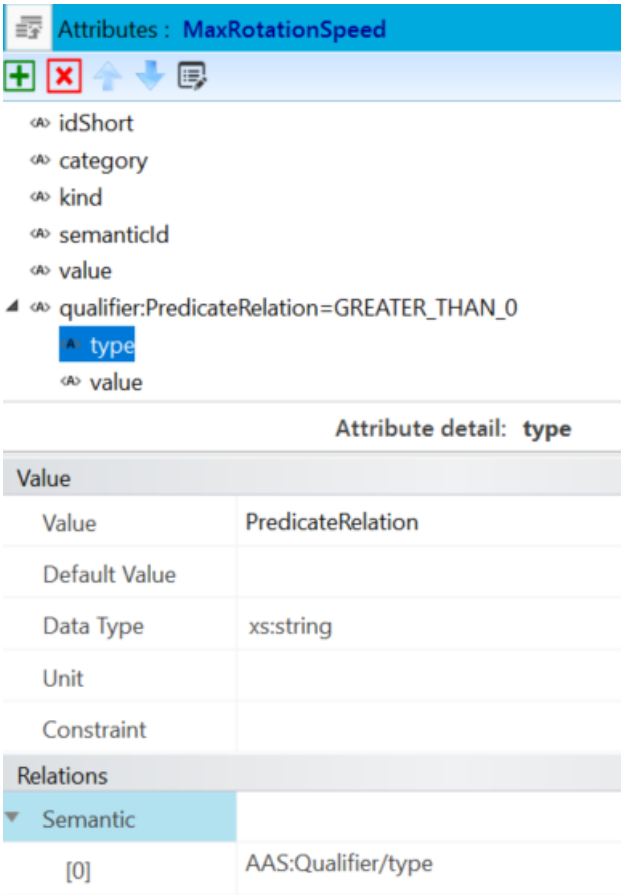
5.8.13 Example Qualifier

The example in Figure 112 shows a property with two qualifiers. One of the qualifiers is of type “ExpressionSemantic”, the other of type “PredicateRelation”. The qualifier value of the “PredicateRelation” qualifier is “GREATER_THAN_0”.

Qualifiers are not referable, i.e. they do not have an idShort attribute. This is why a name of the AML attribute was generated as follows:

qualifier:<value of AAS:Qualifier/type>=<value of Qualifier/value>.

Figure 112 Example Qualifier “PredicateRelation” with qualifier value “GREATER_THAN_0” for a Property



5.8.14 Example Concept Descriptions

Concept Descriptions are stored in an Instance Hierarchy. The default name is “*AssetAdministrationShellConcept-Descriptions*”.

The name of the elements within an instance hierarchy needs to be unique. This is why the name is choosen as follows:

<value of AAS:ConceptDescription/idShort>_<value of AAS:ConceptDescription/identification/idType>__<value of AAS:ConceptDescription/identification/id>

An example concept description for max. rotation speed using the predefined data specification template "D*ata-SpecificationIEC61360*" is shown in Figure 113 and Figure 114.

Figure 113 Example Concept Description using predefined data specification template IEC61360

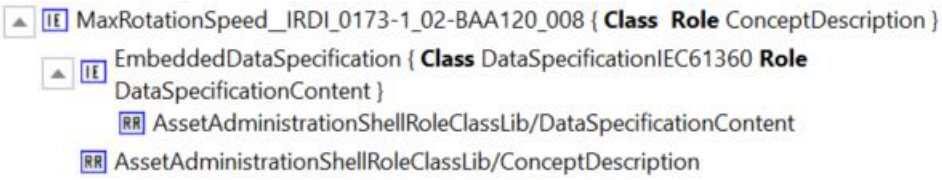
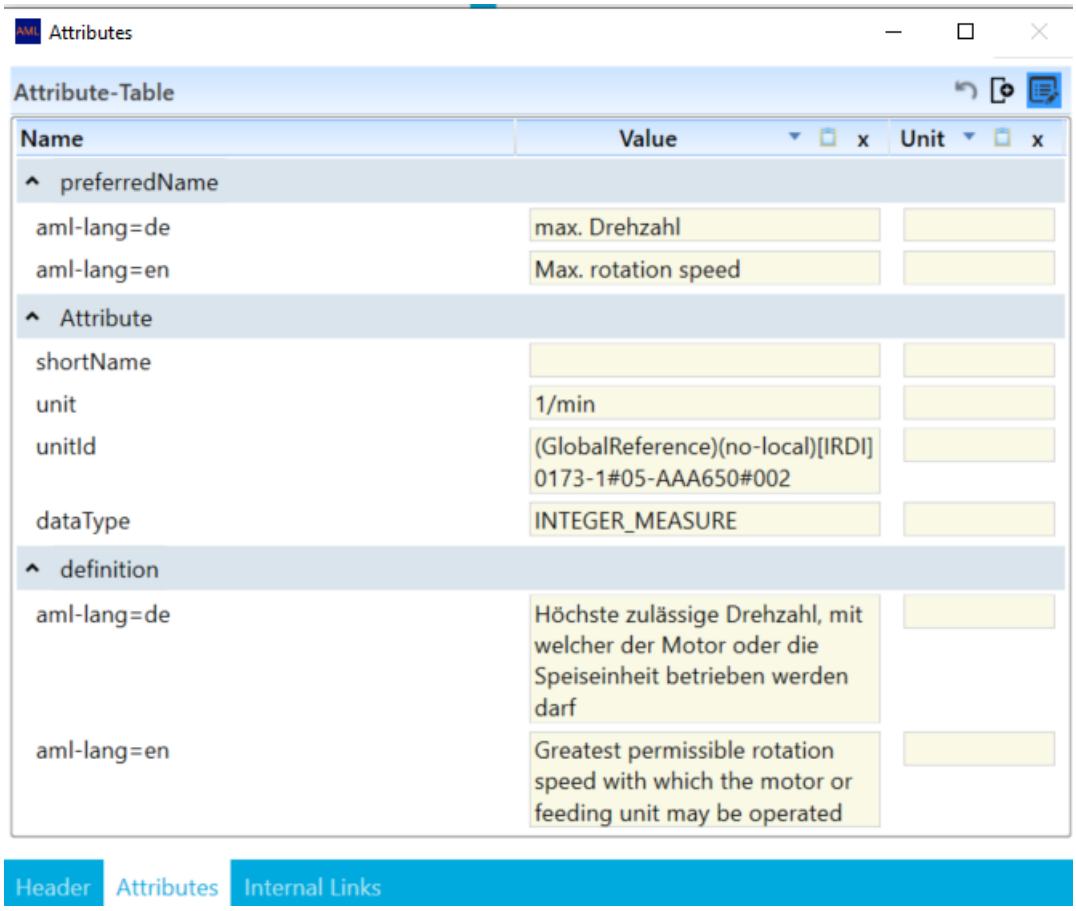


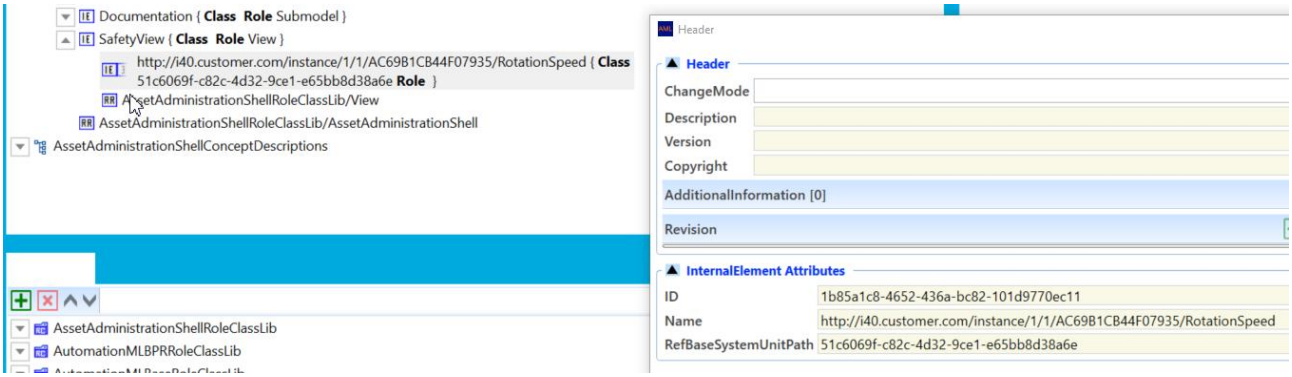
Figure 114 Example Embedded Data Specification IEC61360 of Concept Description for Property "MaxRotationSpeed"



5.8.15 Example View

In Figure 115 an example view with name "SafetyView" is shown that contains a reference to the property RotationSpeed.

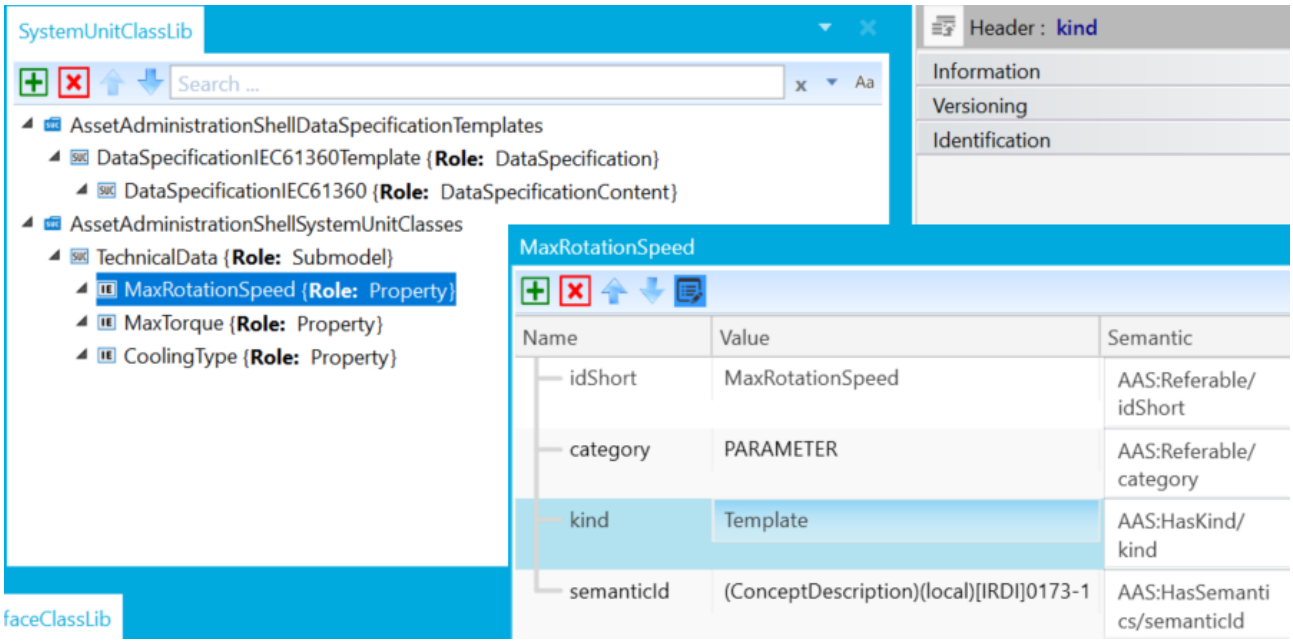
Figure 115 Example SafetyView



5.8.16 Example Submodels of kind=Template

Submodel templates (i.e. submodels and elements with *kind* = Template) are modelled as System Unit Classes. They are part of a System Unit Class Library with default name "*AssetAdministrationShellSystemUnitClasses*".

Figure 116 Example for System Unit Class with a Submodel template for Technical Data



6 Attribute Based & Role Based Access

6.1 Passing Permissions for Access

When having a look at the leading picture (Figure Figure 1 in Clause 3.2) also security aspects have to be considered when transferring information from one value chain partner to the next.

When admin shell content is passed from one partner to the next, the following typical steps need to be done, here shown for the example that the supplier passes on content to the integrator:

- Step A1-A2: The supplier makes a choice which data is to be passed on (see Clause 6.3), and thus determines the content of the AASX package (see Clause 7).
- Step A2-A3: The AASX package is transferred to the integrator.
- Step A3-A4: The integrator receives the package and imports the content into his security domain. During this step, the integrator has to establish access rights according to the requirements in his own security domain.

The admin shell supports attribute based access control (ABAC), a role can be considered as one attribute in this context; other attributes might be time-of-day, originating address and others.

There are two reasons why access permissions are passed between partners:

(a) Access permissions to information elements of an AAS must be established in each security domain.

(b) One partner must be able to pass a suggestion which access permissions should be established for the asset that is described in the AAS.

An example for the second requirement: a robot manufacturer suggests that for the robot the following roles should be established: machine setter, operator and a maintenance role. He also suggests permissions for these roles, e.g. an installer does have write-access to the program of the robot, but an operator does not.

The above example motivates, that the semantics of access permission rules and their exact definitions need to be passed from one security domain to the other.

The passing on of the semantics of attribute based access is implemented by following means:

- Definition of access permissions: The detailed access permission (e.g. read, write, delete, create, invoke method etc.) are defined in a domain specific submodel (see *defaultPermissions* and *selectablePermissions* in Clause 6.5.5).
- Definition of the access permission rules, based on the defined access permissions. These are defined as part of access control (see Clause 6.5.6).
- Association of access permission rules to each information element (object) of the AAS. This means is realized by the information structure of the AAS, itself (see *PermissionsPerObject* in Clause 6.5.6).

In [19] examples and more background information on attribute access control and access control in general can be found.

6.2 Effective Access Permissions based on Access Permission Rules

Effective access permissions are determined based on the access permission rules.

Each information element (object) in the AAS shall have rules that defines its access permissions for each subject. The subject is assumed to be already authenticated.

If an information element does not have these rules, it will automatically use the table for the element where it is included (“inheritance from above"). The most upper object is the AAS itself, i.e. the AAS is the starting point for the inheritance.

As indicated before, subject identification, rule definitions and also permissions could be different for the receiving security domain.

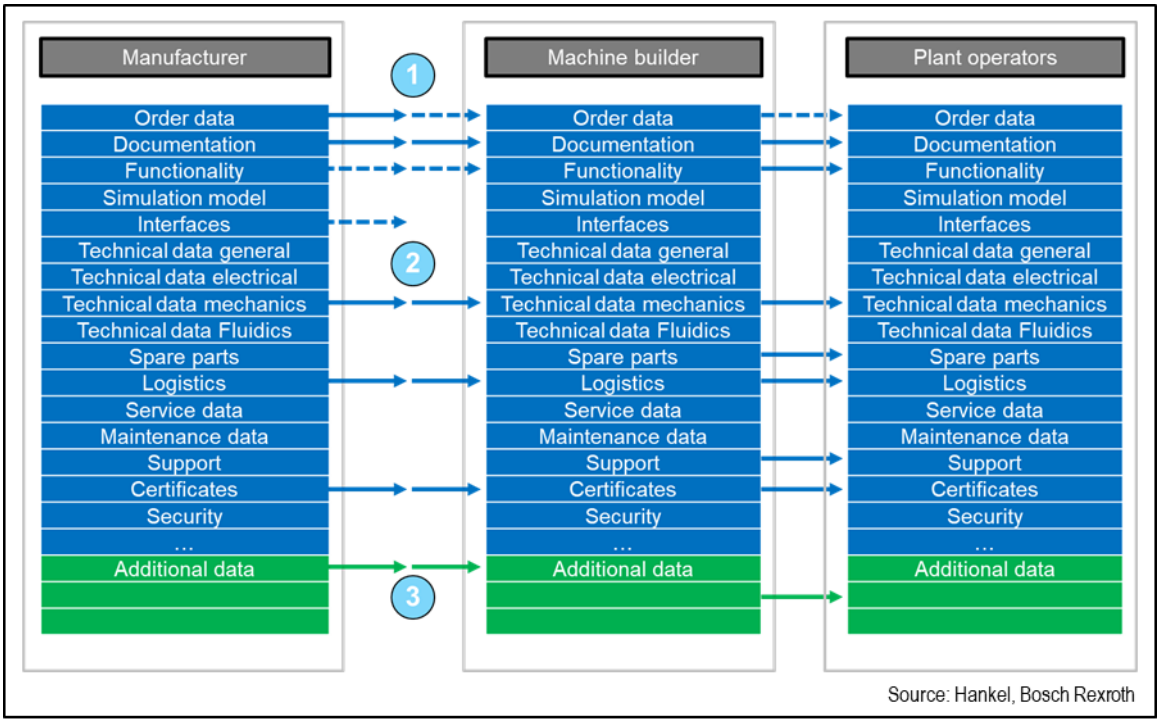
When the receiving party establishes access permissions during step A3-A4, it must merge the passed-on access definitions (permissions and access permission rules) to the existing definitions in its security domain.

6.3 Filtering of Information in Export and Import

When exchanging information from partner A to partner B there are two use cases:

- The producer of information does not want to submit the complete information but only parts of it. The information submitted might vary depending on the specific consumer the information is submitted to. I.e. a filtering mechanism is needed that allows to individually shape the information for the specific consumer.
- The consumer of information does not want to include all information provided by the producer of information in his own process, i.e. he wants to filter only the relevant information.

Figure 117 Example Filtering for Export and Import



As an example, assume that the producer is submitting the complete order data. However, the consumer (in this case the machine builders) is filtering the information (1) and is only importing the information relevant to him. For the functionality both are filtering: the producer is filtering what he submits to the consumer (2) and the consumer again is not using all functionality but is filtering again which functionality shall be used in his environment. The same is possible between machine builders and operator.

Note: In the use case considered in this document, the exchange of information via sharing of xml files etc. the information that is not intended to be submitted needs to be extracted from the corresponding xml files before delivery or before import, respectively. Role or attributes access control do not fit here. The corresponding access policies might help filtering the corresponding information but they cannot be submitted as part of the corresponding file exchanged.

Table 27 shows an example when using the defined xml format as defined in this document. In the example the German translation shall not be submitted, only English language is provided for partner B.

Table 27 Example Filtering of Information in XML

```
[…]
<property>
 <idShort>>NMax</idShort>
 <category>PARAMETER</category>
 <description lang="EN">maximum rotation
speed</description>
 <description lang="DE">maximale Drehzahl</description>
 <ref_hasSemantics>
  <keys>
   <key local="false" type="GlobalReference"
        keytype="IRDI">0173-1#02-baa120#007</key>
  </keys>
 </ref_hasSemantics>
 <value>2000</value>
</property>
[…]
```

Only Lang = "EN"

```
[…]
<property>
 <idShort>>NMax</idShort>
 <category>PARAMETER</category>
 <description lang="EN">maximum rotation
speed</description>
  <ref_hasSemantics>
   <keys>
    <key local="false" type="GlobalReference"
         keytype="IRDI">0173-1#02-baa120#007</key>
  </keys>
 </ref_hasSemantics>
 <value>2000</value>
</property>
[…]
```

6.4 Overview Metamodel Asset Administration Shell for Security

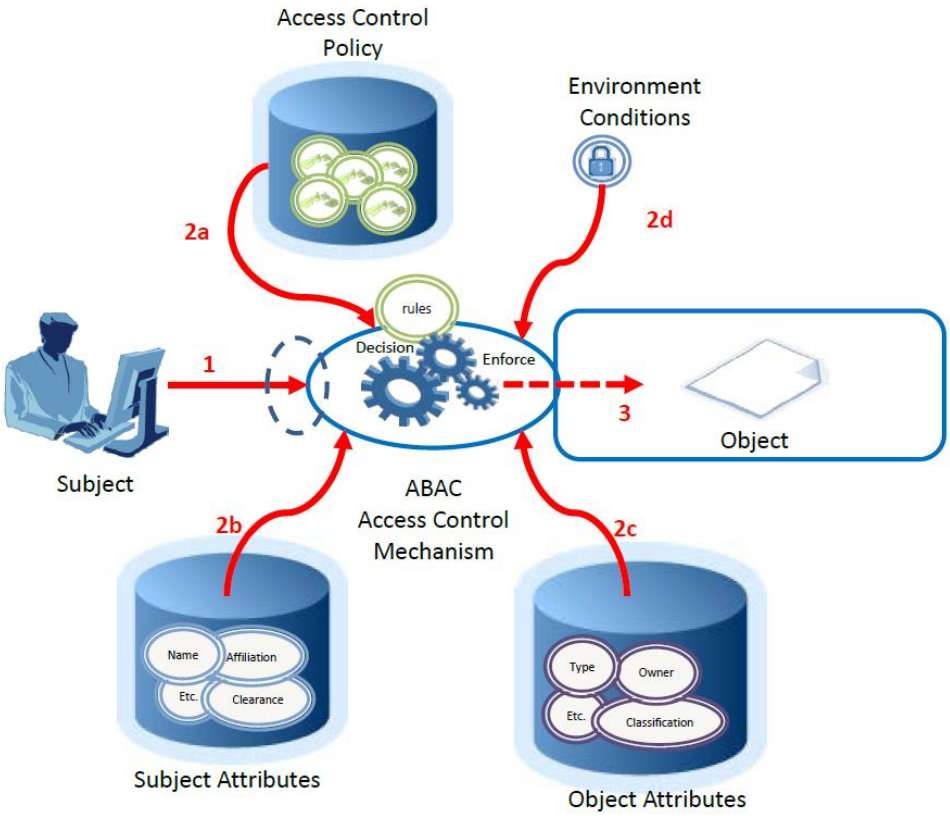
The security attributes are a mandatory part of any Administration Shell.

The security attributes describe:

- Access Control Policy Points including definition of access permission rules
- Certificates

In this document mainly the aspect of access permission is dealt with. The underlying concept is the concept of attribute based access control (ABAC) as described in [22].

Figure 118 Attribute Based Access Control [22]



Note: Attribute in the context of ABAC is different from attributes of elements as defined in the metamodel.

The overall concept is depicted in Figure 118: A subject is requesting access to an object (1). In the context of an AAS an object typically is a submodel or a property or any other submodel element connected to the asset. The implemented access control mechanism of the AAS evaluates the access permission rules (2a) that include constraints that need to be fulfilled w.r.t. the subject attributes (2b), the object attributes (2c) and the environment conditions (2d).

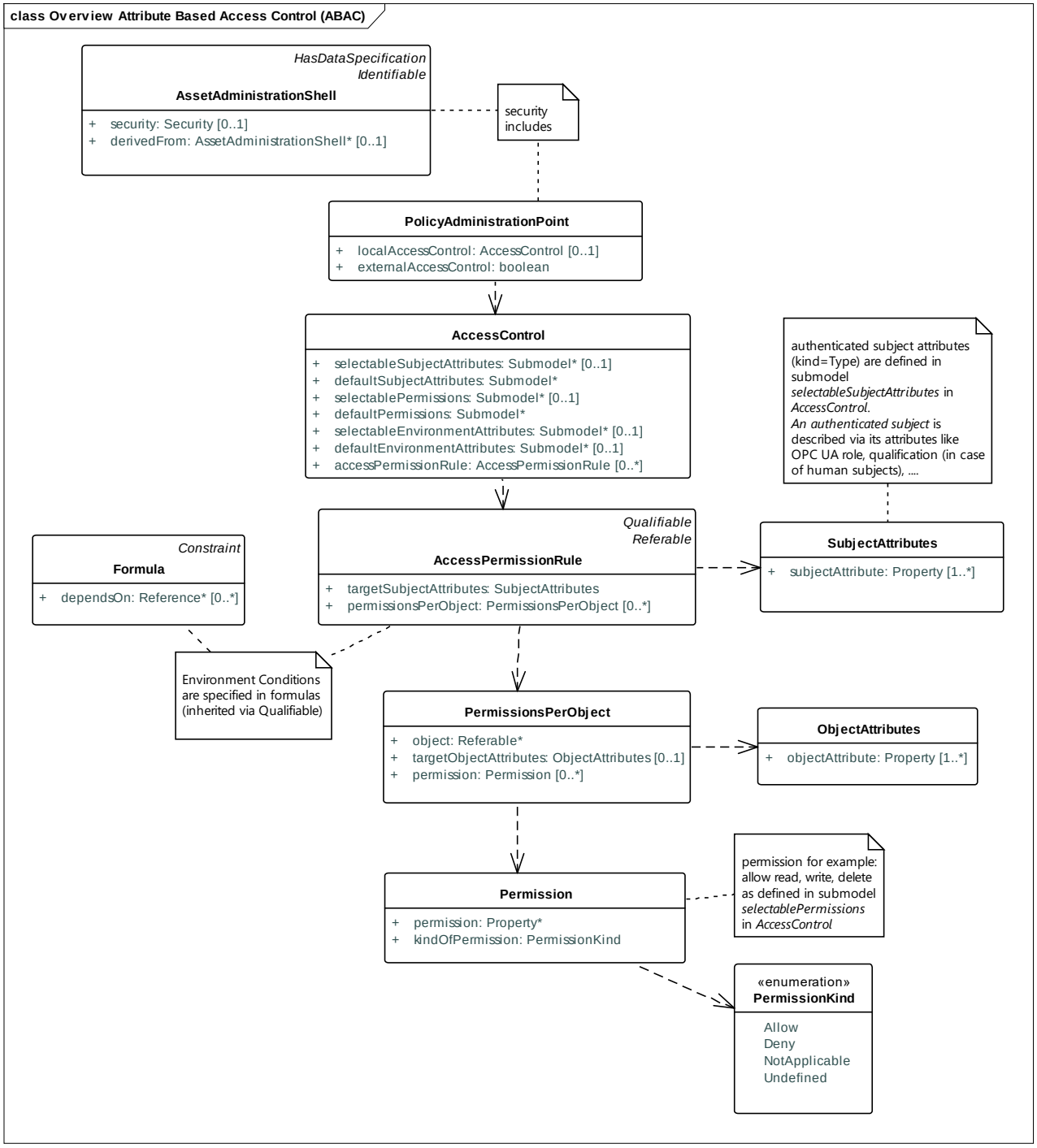
In Figure 119 an overview of the information model of the AAS w.r.t. security is given. The focus is on access control.

An object in the context of ABAC corresponds typically to a submodel or to a submodel element. The object attributes again are modelled as submodel elements.

Subject Attributes need to be accessed either via an external policy information point or they are defined as properties within a special submodel of the AAS. A typical subject attribute is its role. The role is the only subject attribute defined in case of role based access control.

Optionally, environment conditions can be defined. In role based access control no environment conditions are defined. Environment conditions can be expressed via formula constraints. To be able to do so the values needed should be defined as property or reference to data within a submodel of the AAS.

Figure 119 Metamodel Overview Access Control of AAS



Via access permission rules it is defined which subject is allowed to access which objects[30] within the AAS. It is assumed that the subject is already authenticated. Objects can be any referable elements, i.e. they include submodels, assets, concept descriptions, views etc. More general it can be specified whether an authenticated subject is allowed or denied to access an object a.s.o. “Access” might be one of the specified permissions on an element of the AAS. Which permissions are selectable is not defined by the metamodel of the AAS. The selectable permissions are defined via a submodel (*selectablePermissions*). The same holds for the subject attributes (*selectableSubjectAttributes*). The default subject attributes and default permissions are used if they are not overwritten by the owner of the AAS. As for permissions the used authenticated subject attributes are defined in submodel *selectableSubjectAttributes*.

[30] The term “object” is used because it is more generic and in future also other objects like for example attributes of classes may be included besides elements.

Via formula constraints the access rights might be further constrained. For example a formula might specify that the role “maintenance engineer” (to be more precise: an authenticated subject with subject attribute “role = ‘maintenance engineer’”) is only allowed to write configuration parameters if the machine (the asset) is not running. See Figure 19 in Clause 4.7.2.6 for a formal expression of this access rule based on the property “Status”.

Object Attributes are handled in a different way. It is assumed that any property of the object in focus can additionally take over the role of an object attribute. Therefore there is no special submodel for default or selectable object attributes.

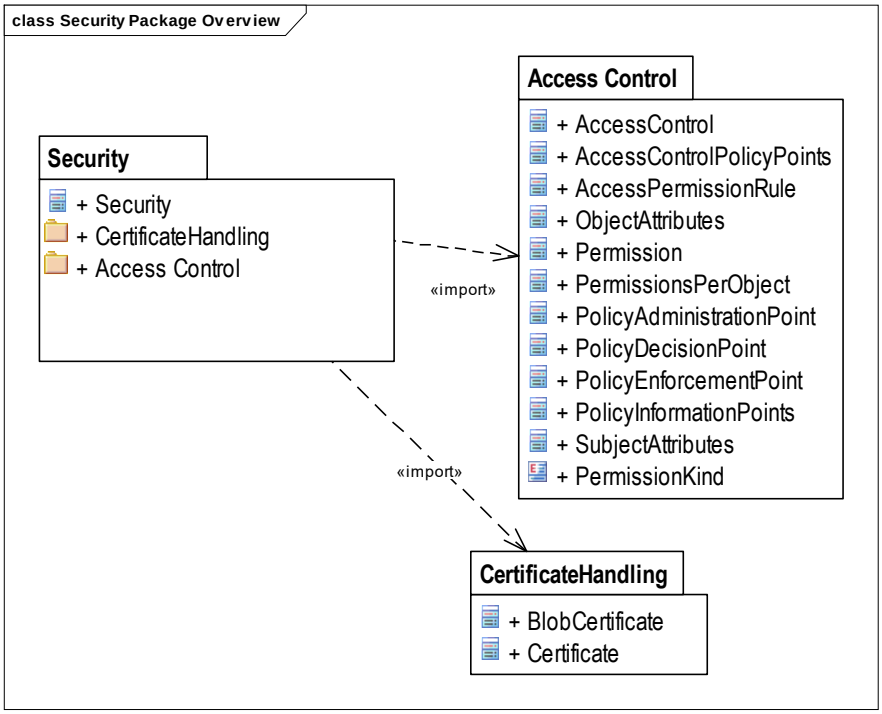
Also the more traditional role based access control can be realized for an AAS: in this case there are no constraints (= environment attributes) defined for the access control rules. For a subject only one subject attribute needs to be defined: its role. For the object no additional object attributes need to be defined.

For more details on attribute based access control including examples how to apply the metamodel as defined in this document see [19].

The classes and their attributes are defined in the following Clause 6.5.

Figure 120 gives an overview of all elements defined for security issues in the metamodel.

Figure 120 Security Overview Packages



6.5 Metamodel Specification Details: Designators

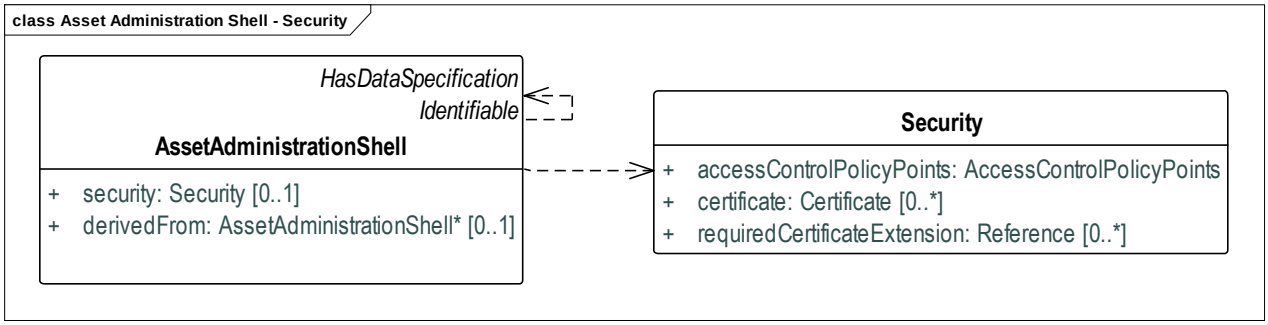
6.5.1 Introduction

In this clause the classes of the metamodel related to security are specified in detail. It is an extension of the metamodel as described in Clause 4.7.

For understanding the extension the basics and common abstract classes need to be understood (see especially Clause 4.7.2, Clause 4.7.21 and Clause 4.7.22).

6.5.2 Security Attributes

Figure 121 Metamodel for Security Attributes of AAS



Class:	Security			
Explanation:	Container for security relevant information of the AAS.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
accessControlPolicyPoints*	Access control policy points of the AAS.	AccessControlPolicy Points	aggr	1
certificate	Certificates of the AAS.	Certificate	aggr	0..*
requiredCertificateExtension	Certificate extensions as required by the AAS	Reference	ref*	0..*

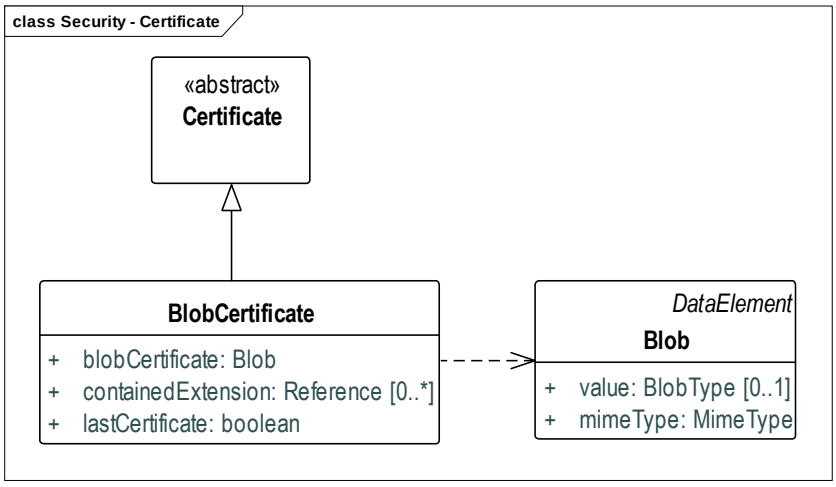
In general it has to be considered how to enable the first configuration of the AAS w.r.t. security. This would include setting the authorization provider endpoint etc.

There is not only one certificate per AAS because certificates can be overwritten if an AAS is taken over by a new owner. The new owner adds a new certificate. Nevertheless the complete set of certificates needs to be available.

Attributes of certificates are defined in X509. A required extension of an ASN1 certificate can be registered via an OID.

6.5.3 Certificate Attributes

Figure 122 Metamodel for Certificates



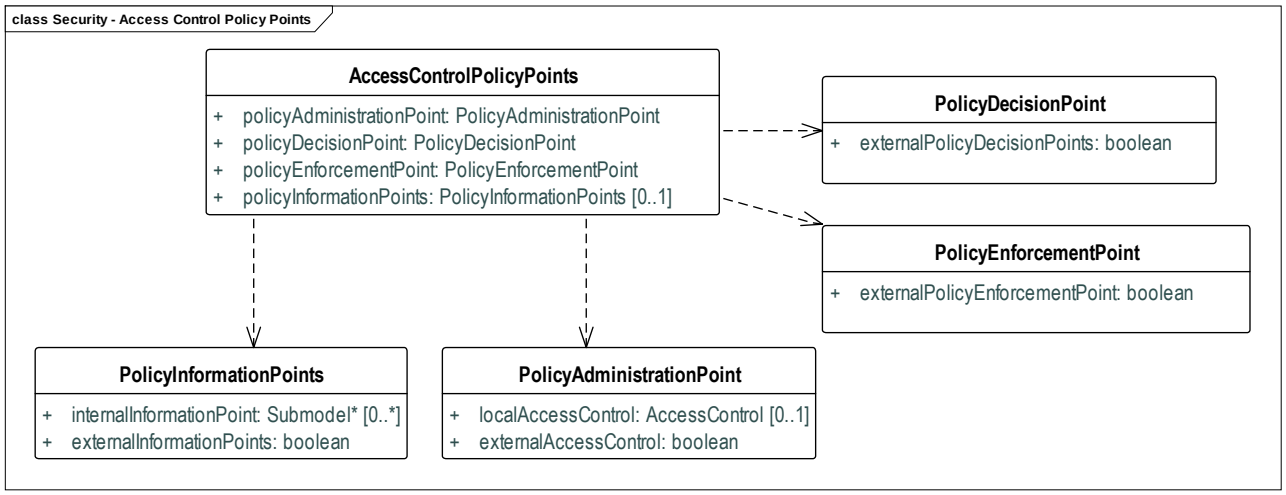
Attributes of certificates are defined in X509. A required extension of an ASN1 certificate can be registered via an OID.

Class:	Certificate <<abstract>>			
Explanation:	Certificate			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
policyAdministrationPoint*	The access control administration policy point of the AAS.	PolicyAdministrationPoint	aggr	1

Class:	BlobCertificate			
Explanation:	Certificate provided as BLOB			
Inherits from:	Certificate			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
blobCertificate*	Certificate as BLOB.	Blob	aggr	1
containedExtension	Extensions contained in the certificate.	Reference	aggr	0..*
lastCertificate*	Denotes whether this certificate is the certificated that fast added last.	boolean	attr	1

6.5.4 Access Control Policy Point Attributes

Figure 123 Metamodel for Access Control Policy Points



Class:	AccessControlPolicyPoints			
Explanation:	Container for access control policy points.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
policyAdministrationPoint*	The access control administration policy point of the AAS.	PolicyAdministrationPoint	aggr	1

Class:	AccessControlPolicyPoints			
policyDecisionPoint*	The access control policy decision point of the AAS.	PolicyDecisionPoint	aggr	1
policyEnforcementPoint*	The access control policy enforcement point of the AAS.	PolicyEnforcementPoint	aggr	1
policyInformationPoints	The access control policy information points of the AAS.	PolicyInformationPoints	aggr	0..1

The definition of policy decision point (PDP) is taken from [22]. The PDP computes access decisions by evaluating the applicable decision points and meta policies. One of the main functions of the policy decision point is to mediate or deconflict decision policies according to meta policies. Either the decision taking is done within the AAS. Then, the AAS is autonomous and independent from an external access control system. Or the decision taking is done outside the AAS. Then, the AAS needs to be able to access this external endpoint for decision taking.

Class:	PolicyAdministrationPoint			
Explanation:	Definition of a security policy administration point (PAP).			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
localAccessControl	The policy administration point of access control as realized by the AAS itself. Constraint AASd-009: Either there is an external policy administration point endpoint defined or the AAS has its own access control.	AccessControl	aggr	0..1
externalAccessControl*	If *externalAccessControl* True then an Endpoint to an external access control defining a policy administration point to be used by the AAS needs to be configured.	boolean	attr	1

Class:	PolicyInformationPoints			
Explanation:	Defines the security policy information points (PIP). Serves as the retrieval source of attributes, or the data required for policy evaluation to provide the information needed by the policy decision point to make the decisions.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
internalInformationPoint	References to submodels defining information used by security access permission rules.	Submodel	ref*	0..*
externalInformationPoints*	If *externalInformationPoints* True then at least one Endpoint to external available information needs to be configured for the AAS.	boolean	aggr	1

The definition of policy information point (PIP) is taken from [22]. The difference between external and internal information points is whether the AAS needs access via an endpoint to an externa source of information or whether the

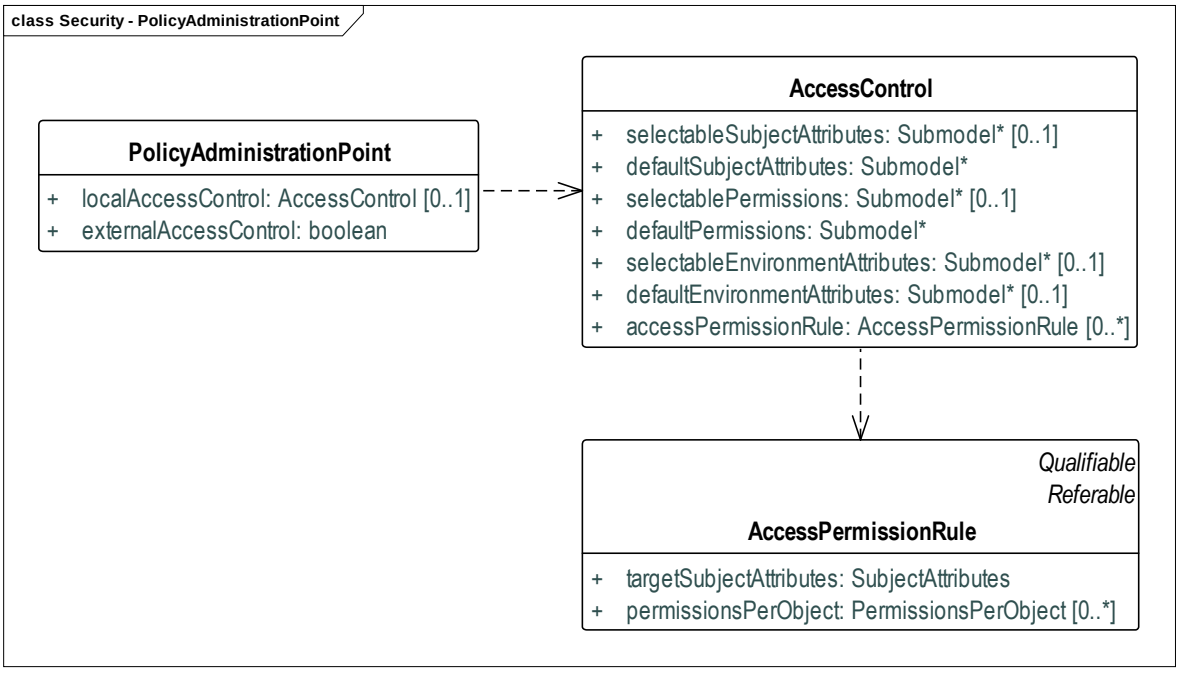
AAS stores the needed information itself. There might also be external and internal information points for an AAS to be considered for decision taking.

Class:	PolicyEnforcementPoints			
Explanation:	Defines the security policy enforcement points (PEP).			
Inherits from:				
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
externalPolicyEnforcementPoint*	If *externalPolicyEnforcementPoint* True then an Endpoint to external available enforcement point taking needs to be configured for the AAS.	boolean	aggr	1

Class:	PolicyDecisionPoint			
Explanation:	Defines the security policy decision points (PDP).			
Inherits from:				
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
externalPolicyDecisionPoints*	If *externalPolicyDecisionPoints* True then Endpoints to external available decision points taking into consideration for access control for the AAS need to be configured.	boolean	aggr	1

6.5.5 Local Access Control Attributes

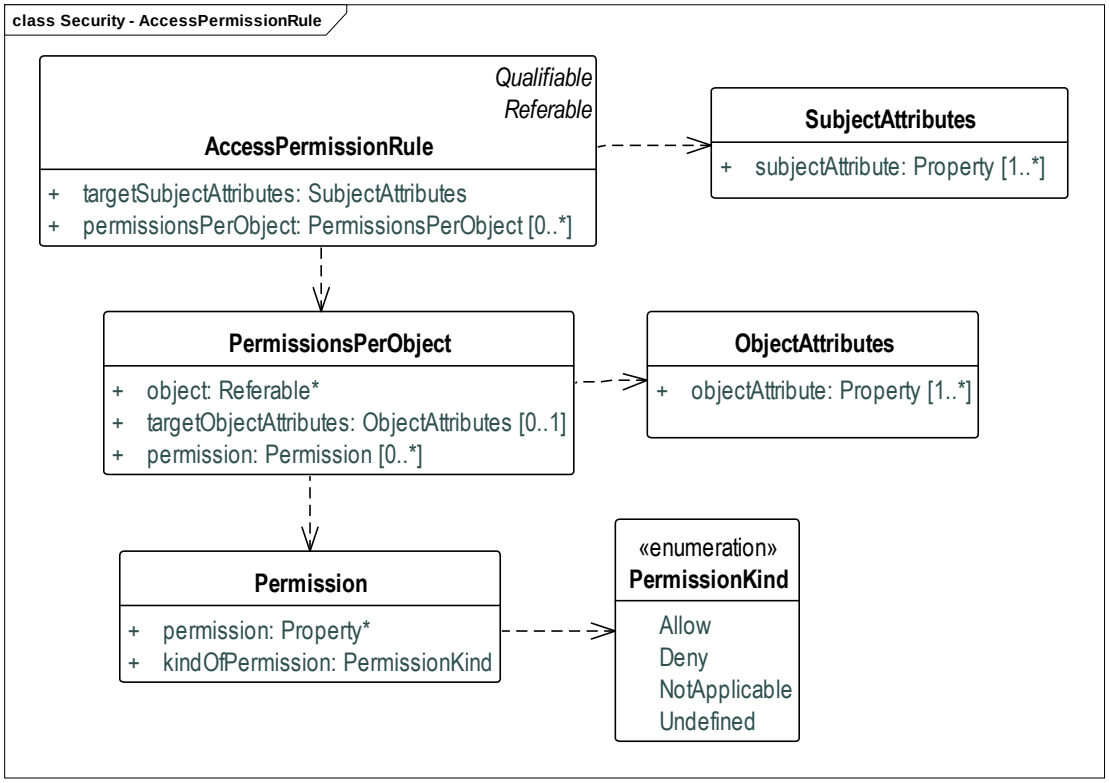
Figure 124 Metamodel for Access Control



Class:	AccessControl			
Explanation:	Access Control defines the local access control policy administration point. Access Control has the major task to define the access permission rules.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
accessPermissionRule	Access permission rules of the AAS describing the rights assigned to (already authenticated) subjects to access elements of the AAS.	AccessPermissionRule	aggr	0..*
selectableSubjectAttributes	Reference to a submodel defining the authenticated subjects that are configured for the AAS. They are selectable by the access permission rules to assign permissions to the subjects. Default: reference to the submodel referenced via *defaultSubjectAttributes*.	Submodel	ref*	0..1
defaultSubjectAttributes*	Reference to a submodel defining the default subjects attributes for the AAS that can be used to describe access permission rules. The submodel is of kind=Template.	Submodel	ref*	1
selectablePermissions*	Reference to a submodel defining which permissions can be assigned to the subjects. Default: reference to the submodel referenced via *defaultPermissions*	Submodel	ref*	0..1
defaultPermissions*	Reference to a submodel defining the default permissions for the AAS.	Submodel	ref*	1
selectableEnvironmentAttributes	Reference to a submodel defining which environment attributes can be accessed via the permission rules defined for the AAS, i.e. attributes that are not describing the asset itself. Default: reference to the submodel referenced via *defaultEnvironmentAttributes*	Submodel	ref*	0..1
defaultEnvironmentAttributes	Reference to a submodel defining default environment attributes, i.e. attributes that are not describing the asset itself. The submodel is of *kind=Template*. At the same type the values of these environment attributes need to be accessible when evaluating the access permission rules. This is realized as a policy information point.	Submodel	ref*	0..1

6.5.6 Attributes for Access Permission Rule

Figure 125 Metamodel for Access Permission Rule



Class:	AccessPermissionRule			
Explanation:	Table that defines access permissions per authenticated subject for a set of objects (referable elements).			
Inherits from:	Referable; Qualifiable			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
targetSubjectAttributes*	Target subject attributes that need to be fulfilled by the accessing subject to get the permissions defined by this rule.	SubjectAttributes	aggr	1
permissionsPerObject	Set of object-permission pairs that define the permissions per object within the access permission rule.	PermissionsPerObject	aggr	0..*

Class:	PermissionsPerObject			
Explanation:	Table that defines access permissions for a specified object. The object is any referable element in the AAS. Additionally object attributes can be defined that further specify the kind of object the permissions apply to.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.

Class:	PermissionsPerObject			
object*	Element to which permission shall be assigned.	Referable	attr	1
targetObjectAttributes	Target object attributes that need to be fulfilled so that the access permissions apply to the accessing subject.	ObjectAttributes	aggr	0..1
permission	Permissions assigned to the object. The permissions hold for all subjects as specified in the access permission rule.	Permission	attr	0..*

Class:	ObjectAttributes			
Explanation:	A set of data elements that describe object attributes. These attributes need to refer to a data element within an existing submodel.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
objectAttribute*	A data elements that further classifies an object.	DataElement	ref*	1..*

Class:	Permission			
Explanation:	Description of a single permission.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
permission*	Reference to a property that defines the semantics of the permission. Constraint AASd-010: The property has the category "CONSTANT". Constraint AASd-011: The permission property shall be part of the submodel that is referenced within the "*selectablePermissions*" attribute of "*AccessControl*".	Property	ref*	1
kindOfPermission*	Description of the kind of permission. Possible kind of permission also include the denial of the permission. Values: • Allow • Deny • NotApplicable • Undefined	PermissionKind	attr	1

Class:	SubjectAttributes			
Explanation:	A set of data elements that further classifies a specific subject.			
Inherits from:	--			
Attribute (*=mandatory)	Explanation	Type	Kind	Card.
subjectAttribute*	A data element that further classifies a specific subject. Constraint AASd-015: The data element shall be part of the submodel that is referenced within the "*selectableSubjectAttributes*" attribute of "*AccessControl*".	DataElement	ref*	1..*

Enumeration:	PermissionKind
Explanation:	Enumeration of the kind of permissions that is given to the assignment of a permission to a subject.
Literal	Explanation
Allow	Allow the permission given to the subject.
Deny	Explicitly deny the permission given to the subject.
NotApplicable	The permission is not applicable to the subject.
Undefined	It is undefined whether the permission is allowed, not applicable or denied to the subject.

7 Package File Format for the Asset Administration Shell (AASX)

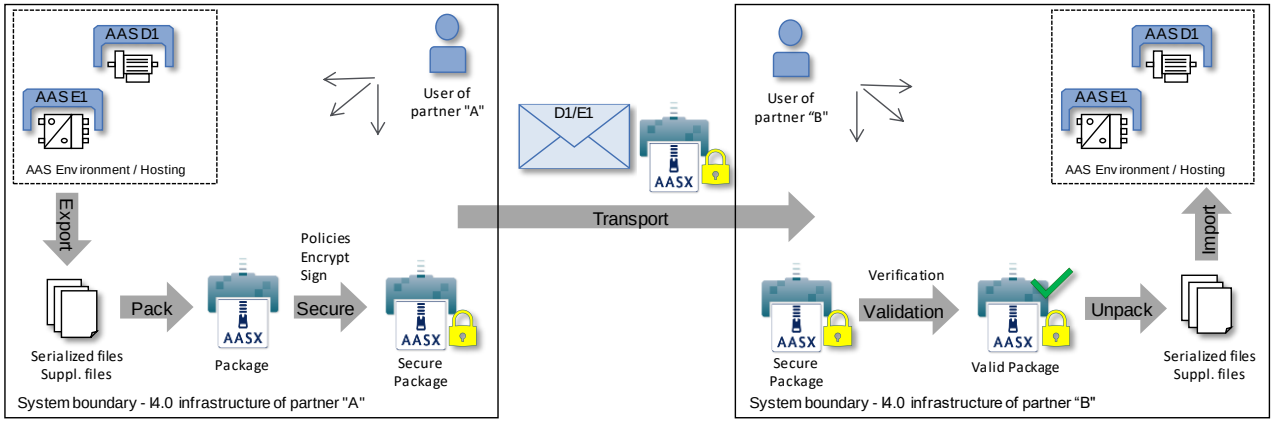
7.1 General

In some use cases it is necessary to exchange the full or partial structure of the Asset Administration Shell with or without associated values and/or make the information persistent (e.g. store it in a file server). This would mean that it is necessary to define a file format that can hold and store this information. Therefore, a package file format for the Asset Administration Shell (AASX) is defined based on the following requirements:

- Generic package file format to include the Asset Administration Shell structure, data and other related files
- Main use cases are the exchange between organizations/partners and storage/persistency of the Asset Administration Shells' information.
- Without any legal restriction and no royalties. Preferably based on an international standard with high guarantees of future maintainability of that format
- Existence of APIs to create, read and write this format
- Digital signatures & encryption capabilities must be provided
- Policies for authenticity and integration of package files[31]

The following process in Figure 126 is defined for creating and consuming AASX packages.

Figure 126 Process for generating and consuming AASX packages



The process starts by serializing the existing AAS (e.g. D1 and E1) into files (according to the serialization mechanisms described in this document), as well as exporting other supplementary files (which are files mentioned in the structure of the AAS, such as manuals, CAD files, etc.). All of those files will be packaged together into the AASX ZIP file format and will be followed by several security steps that defines the policies for modifiability, encryption and digitally signing of the files inside the AASX. The final AASX can then be transported from the AASX producer (in this case partner A) to the AASX consumer (partner B), by digital media such as e-mail, USB-Sticks, etc. The consumer needs first to validate and verify the incoming AASX, unpack the contained files and then import them to generate the new AAS in the consumer environment. The process will be explained in detail in the following sub-sections.

7.2 Selection of the Reference Format for the Asset Administration Shell Package Format

The *Führungskreis Industrie 4.0 – UAG Verwaltungsschale* has decided to use the Open Packaging Conventions (OPC)[32] format as the reference for the Asset Administration Shell package format definition, due to the following reasons:

---

[31] Role-based policies to access this package is not defined, as this is a feature of the systems that host the AASs (see section 6)

[32] Not to be confused with OPC (Open Platform Communication) of the OPC Foundation. Therefore, we will use the full term of "Open Packaging Conventions" instead of the abbreviation "OPC".

- Open Packaging Conventions is an international standard specified in ISO/IEC 29500-2:2012 and ECMA-376.
- Open Packaging Conventions is based on ZIP (as a package container) and XML (for the description of some internal files and definitions). Those two technologies are the most widely used in their respective domains and are also addressed for long-term archiving.
- Open Packaging Conventions can be used as package for non-office applications too (there are many examples available, such as NuGet, FDI packages, etc.). It provides a logical model that is independent from how the files are stored in the package. This logical model can be expanded to any sort of application.
- Open Packaging Conventions is also used in the scope of Industry (e.g. FDI packages) and currently in discussion as possible container format for some FDT® and ODVA Project xDS™ use cases.
- Open Packaging Conventions (and Open Document Format packages too) supports digital signing. It can be done for individual files inside the package. Encryption isn't specified in Open Packaging Conventions (it only mentions what shall not be done). Anyway, encryption is still possible (see later)
- There are some APIs to handle Open Packaging Conventions packages (Windows API, .NET, Java, …) without the need of much knowledge on the technical specification
- Chunking in Open Packaging Conventions is encouraged, i.e. split files into small chunks. This is better for reducing the effect of file corruption and better for data access.
- There are some international organizations that recommend using Open Document Format (ISO/IEC 26300-3) instead (e.g. EU, NATO, …), but this recommendation is related to the formats used specifically in office applications.
- The Office Open XML and Open Packaging Conventions specifications originated from Microsoft Corporation and later standardized as ISO/IEC 29500 and ECMA-376. Current and future versions of ISO/IEC 29500 and ECMA-376 are covered by Microsoft's Open Specification Promise, whereby Microsoft "irrevocably promises" not to assert any claims against those making, using, and selling conforming implementations of any specification covered by the promise (so long as those accepting the promise refrain from suing Microsoft for patent infringement in relation to Microsoft's implementation of the covered specification). [24]
- Office Open XML (including the Open Packaging Conventions format) and Open Document Format are politically conflicting formats (see details in [25]and [26]). Choosing Open Packaging Conventions as the option for storing the Asset Administration Shell information was solely a technical decision based on the arguments mentioned here.
- Open Packaging Conventions was chosen in favour of iiRDS (v1.0). The scope of iiRDS might not be aligned with the requirements of the Asset Administration Shell, i.e. iiRDS is mostly a format for storing technical documentation of industry devices based on concepts of ontology.

7.3 Basic Concepts of the Open Packaging Conventions

The packaging model specified by the Open Packaging Conventions describes **packages**, **parts**, and **relationships**. Packages hold parts, which hold content and resources, such as **files**[33]. Every file in a package has a unique URI-compliant file name along with a specified content-type expressed in the form of a MIME media type.

Relationships are defined to connect the package to files, and to connect various files in the package. The definition of the relationships is the **logical model** of the package. The resource that is a source of a relationship must be either the package itself or a data component (file) inside of the package. The target resource of a relationship can be any URI-addressable resource inside or outside of the package. It is possible to have more than one relationship that share the same target file (see example 9–6 in ISO/IEC 29500-2: 2012).

The **physical model** maps those logical concepts to a physical format. The result of this mapping is a physical package format (a ZIP archive format) in which files appear in a directory-like hierarchy. Any individual or organization can

[33] The term "file" will be used instead of "part".

design a physical package format by mapping logical package concepts to a desired physical format. Thus, package format designers can design and optimize a physical format for the specific needs of an application without compromising the logical structure of the package (adapted from [27]and [28]).

7.4 Conventions for the Asset Administration Shell package file format (AASX)

The Asset Administration Shell Package (AASX) format derives from the Open Package Conventions standards, consequently inheriting its characteristics. Nevertheless, some convention shall be defined for the AASX:

- Package format and rules according to ISO/IEC 29500-2:2012. Any derivate format from this standard (such as the AASX format) requires the definition of a logical model, physical model and a security model. Those specific conventions are described in the next subsections.
- File extension for the AASX format: **.aasx**
- MIME-type for the AASX format: application/asset-administration-shell-package[34]
- **Icon** for the AASX.
- The AASX format can be identified by the file extension and MIME type. Content-wise, it is possible to identify it when reading the first relationship file /_rels/.rels (as defined in Open Packaging Conventions) and looking for a relationship type **http://admin-shell.io/aasx/relationships/aasx-origin** (which is the entry point for the logical model of the Asset Administration Shell).
- The following paths and filenames in the package are already reserved by the Open Packaging Conventions specification and therefore shall not be used for any derivative format: /[Content_Types].xml;/_rels/.rels; /<*file_path*>/_rels/<*filename*>.rels (where <*filename*> is a file in the package that is source of relationships and <*file_path*> is the path to that file).
- It is not mandatory to open the AASX format in any existing Office Open XML / Open Packaging Conventions compatible office-application (e.g. Microsoft Office, LibreOffice), because the required relationships and files for the different office "models" may not be present (e.g. http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument for "docx" document).

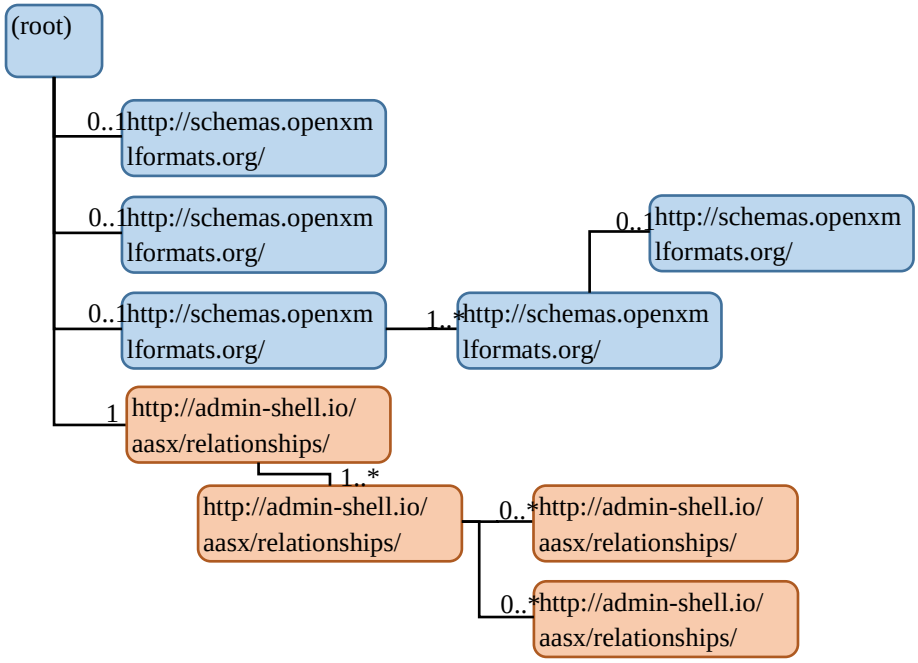
7.5 Logical Model

As mentioned before, it is necessary to define a logical model for formats on top of Open Packaging Conventions.

Figure 127 defines the logical model for the AASX format. It is made of a set of relationship types (URI), their cardinality (how many relationships of that type are possible) and the source of the relationship. In addition (not shown in Figure 127), a specific relationship instance has also a unique ID and a target resource (URI of a target file inside or outside the package).

[34] The currenty MIME-type is provisory and needs to be requested officially.

Figure 127 Logical model for the AASX format[35]



The relationships for thumbnail, core-properties, digital-signatures (origin, signature and certificate) are defined by Open Packaging Conventions, so no need to reinvent. The other relationships were specifically defined to support the Asset Administration Shell specific files. Here a short description on each relationship[36] of Figure 127:

- **thumbnail** – Optional. Required to define a thumbnail for that package (e.g. picture of the administrated device). The thumbnail picture can be shown instead of the package's icon based on the extension and/or MIME type.
- **core-properties** – Optional. There is a schema for describing the package through "core properties," which uses selected Dublin Core metadata elements in addition to some Open Packaging Conventions-specific elements. The core-properties do not describe the Administration Shell, but the package itself. Some elements of the core-properties may be similar/equal to elements of the Administration Shell. Some core-properties are: Title, Subject, Creator, Keywords, Description, LastModifiedBy, Revision, LastPrinted, Created, Modified, Category, Identifier, ContentType, Language, Version, ContentStatus.
- **digital-signature/origin, digital-signature/signature and digital-signature/certificate** – Optional. Required if you need to sign files and relationships inside the package. Their relationships basically target files that contain the data on signatures (e.g. certificate, digests, …). See the description later in this document about digital signatures.
- **aasx-origin** – Mandatory. Origin of the AASX specific relationships and files. From this origin one or more AAS can be defined. The producer should not create any content in the aasx-origin file itself, meaning that the file is empty or contains the the text "Intentionally empty".As the Open Packaging Conventions relationship model does not allow to target directories inside the ZIP, the alternative is to create an empty file that serves as the entry-point for the AASX information (this is the same approach as it is used for digital signatures).
- **aas-spec** – at least one relationship of this type is mandatory. Targets the file that contains the structure/specification of one or more AAS (aasenv), as defined in this document. Optionally, some of

[35] Note that the logical model does not state anything about the format / content of the target files of relationships. This will be addressed in the physical model.

[36] To avoid the long names of the relationships, we will use the short name along the text.

the specification can be "splitted" into separate files, but in any case, this aas-spec file is still mandatory and contains at least the non-splittable information.

- **aas-spec-split** – Optional. This relationship will target a file containing a splittable part of the AAS specification (aasenv). Some serialization formats allow that parts of the AAS sspecification can be splitted into several files. Those files are then referenced by this relationship type, so that any consumer of the AASX can "reassemble" the AAS information.
- **aas-suppl** – Optional. Targets any additional file, especially if it is referenced (not stored as blob) in the data of an AAS (via File property).

Note: not all of the references inside the specification of an AAS may target files that are also stored inside the AASX. A relative URIs mentioned in the serialized files is an indication that the file is inside the AASX.

7.6 Physical Model

The physical model defines how the different files are stored in the package, based on Open Packaging Conventions and how files are addressed in the relationships. As mentioned before, the physical package format is a ZIP file that can be open and edit in any PKWARE/ZIP compatible application.

In order to utilize the identifiers of Administration Shell and SubModels, **friendly names** are required. The friendly name of such entities is built by searching all characters of the identifier, which are not letters or digits and substituting them with an underscore "_".

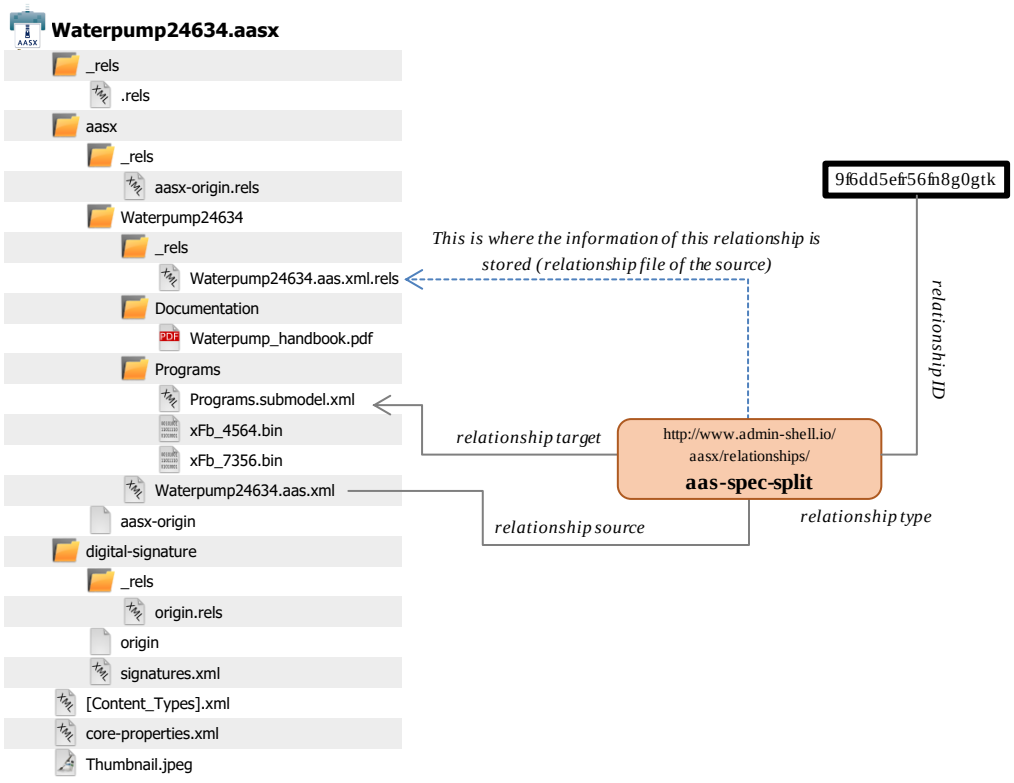
A feature of this physical and logical model is that the filename and location of those files can be customized (if associated relationships have a correct URI to those files, and therefore can be used to locate the files according to the logical structure). For example, one package producer might store an aas-spec file in /aasx/device.xml, the other one in /asset-admin-shell/productX123.xml, but both use the same relationship type for that files. To have a more consistent approach on the physical model, the following best-practice is defined for storing files inside the AASX package:

- Open Packaging Conventions related files should be stored according to the API that was used to generate/manipulate the AASX package (it is not recommended to do this manually).
- **/aasx/** shall be the root folder for the AASX package specific information.
- **/aasx/aasx-origin** shall be the target of the relationship aasx-origin without content (empty file).
- **/aasx/<*aas-friendly-name*>/** shall be the folder for storing all files for a specific AAS, where <*aas-friendly-name*> is the friendly name of the AAS.
- **/aasx/<*aas-friendly-name*>/<*aas-friendly-name*>.aas.<extension>** shall be the target of a relationship of type aas-spec, where <*extension*> is the extension based on the type of serialization (e.g. .xml, .json).
- **/aasx/<*aas-friendly-name*>/<*aas-friendly-name*>.<*view-idshort*>.view.<extension>** shall be the target of a relationship of type aas-spec-split that contains a specific view definitions of an AAS. This is only required if the view definitions aren't defined in the target file of aas-spec.
- **/aasx/<*aas-friendly-name*>/<*aas-friendly-name*>.cdic.<extension>** shall be the target of a relationship of type aas-spec-split that contains the ConceptDictionary definition of an AAS. This is only required if the ConceptDictionary isn't defined in the target file of aas-spec.
- **/aasx/<*aas-friendly-name*>/<*aas-friendly-name*>.secattrib.<extension>** shall be the target of a relationship of type aas-spec-split that contains the Security model of an AAS. This is only required if the Security models isn't defined in the target file of aas-spec.
- **/aasx/<*aas-friendly-name*>/<*submodel-friendly-name*>/** shall be the folder to store files related to a submodel of an AAS (targets of aas-suppl relationships that are referenced in that submodel and splits containing submodel information). <*submodel-friendly-name*> is the friendly name of the submodel.
- **/aasx/<*aas-friendly-name*>/<*submodel-friendly-name*>/<*submodel-friendly-name*>.submodel.<*extension*>** shall be the target of a relationship of type aas-spec-split that contains a submodel definition of an AAS. This is only required if a submodel isn't defined in the target file of aas-spec.
- Those file names must only contain characters that can be used for file names.

- The conventions defined here shall not be used for other files. E.g. any other file in a submodel folder shall not contain the substring ".submodel." in its name.
- Note that the format of the files targeted by the relationships aas-spec and aas-spec-split depend on the serialization format that was used to generate them (e.g. xml, json, …).
- It is also possible to have different serialization formats of the same Administration Shell stored in the same AASX. In this case, the different serialization formats can be stored in parallel with different extension, different MIME type and different relationships. For example, Waterpump24634.aas.xml and Waterpump24634.aas.json are stored in the same folder Waterpump24634, but are targets of different relationships (different IDs) of the same relationship type aas-spec. Both are then the entry-point of different source relationship branches (each one having its own .rels file, i.e. Waterpump24634.aas.xml.rels and Waterpump24634.aas.json.rels).
- To avoid duplication of data, it is possible to target the same file by different relationships (e.g. two different relationships of type aas-suppl may have the same target file).

An example of a physical model for an AASX based on a sample product is shown in Figure 128. It shows the content of the package listed in a tree view and one example mapping to the logical model as defined in Figure 127. The physical structure is based on the best practice mentioned before. Note that in the example there is only one AAS in the package, one submodel (programs) is stored in a separate file and the certificate is embedded into the signature file (so no need of the additional relationship). It is also assumed in this example that the AAS specification files are serialized into XML.

Figure 128 Physical model for an AASX based on a sample product (left) and an example of mapping to the logical model (right)



It is possible to classify the files in a AASX package into the following types: 1) files that are referenced in the relationships of the logical model and must match the target URI inside each relationship, 2) "Files that aren't source or target of any relationship (not allowed as they do not follow the logical model defined in this document and might impact some aspects regarding digital signatures and its verification) and 3) Open Packaging Conventions specific files that aren't associated to the logical model (relationships):

- **/[Content_Types].xml** – contains a list of extensions and MIME type of all file types inside the package. The element override can specify the MIME type for specific files independent of the extension. The MIME type of AAS-specific files depends on the type of serialization that was used to generate the content of the files (e.g. if XML was used for some files, then the MIME type "text/xml" together with the used file extensions must appear in [Content_Types].xml. If there is no specific MIME type for some files, then "application/octet-stream" shall be used.
- **/_rels/.rels** – contains all relationships coming from the source “root” (which is the package itself), binding the source with a target (the URI of an internal file or external resource). For example, for the thumbnail relationship it looks like this:
- <Relationship Type = "http://schemas.openxmlformats.org/package/2006/ relationships/metadata/thumbnail" Target = "/Thumbnail.jpeg" Id = "Rc76d59d18bd7440f" />
- This means that the target data for this thumbnail relationship is stored in /Thumbnail.jpeg.
- **/<*file_path*>/_rels/<*filename*>.rels** – non-root relationships are stored in those files. E.g. the relationship based on type aas-spec-split starting from the source file Waterpump24634.aas.xml are stored in file /aasx/Waterpump24634/_rels/Waterpump24634.aas.xml.rels.

An AASX can be generated by using different means:

- Manually by adding files (changing files) to (of) a Zip file. This requires a deep understanding of the Open Packaging Conventions format, because adding just a file to the package with an ZIP editor isn’t enough (i.e. need to edit the [Content_Types].xml and some of the .rels files too)
- Programmatically generating and changing the package format (e.g. using .NET System.IO.Packaging). This will typically avoid the errors that can be done when creating manually the package. In addition, the Open Packaging Conventions specific procedures, the logical, physical and security model defined for the derived AASX must be considered.

7.7 Digital Signatures

Essentially the digital signature of an electronic document (in this case the files and relationships inside the container) aims to fulfil the following requirements [29]:

- that the recipient can verify the identity of the sender (authenticity);
- that the sender cannot deny that he signed a document (non-repudiation);
- that the recipient is unable to invent or modify a document signed by someone else (integrity).

A digital signature does not "lock" a document or cause it to become encrypted (although it may already be encrypted). Document content remains unchanged after being signed. Digital signatures do not prevent signed content from being viewed by unintended consumers.

A digital signing feature is already provided by the Open Packaging Conventions specification [27]. This signing framework for packages uses the XML Digital Signature Standard, as defined in the W3C Recommendation XML-Signature Syntax and Processing. This recommendation specifies the XML syntax and processing rules for producing and storing digital signatures.

- The package files defined for the signing framework are the origin file, the signature file(s), and the certificate file(s).
- **digital-signature/origin file** – starting point for navigating through the signatures in a package. The origin file is targeted from the package root using the digital signature origin relationship (as shown in the logical model in Figure 127). Multiple signature files may be targeted from the origin file. If there are no signatures in the package, the origin file will not be present.
- **digital-signature/signature file(s)** – contain markup defined in the W3C Digital Signature standard as well as in the packaging namespace. The files are targeted from the origin file with the signature relationship (as shown in the logical model in Figure 127).
- **digital-signature/certificate file(s)** – The X.509 certificate required for identifying the signer, if placed in the package, may be embedded within a signature file, or stored in a separate certificate file. The

optional certificate file is targeted from a signature file with the certificate relationship. The certificate file can be shared between multiple signature files.

In the package, individual files and relationships can be independently signed[37], meaning that it is possible to select which files and relationships need a signature and which certificate to be used to sign. When the relationships file (.rels) is signed as a whole, all the relationships defined in that file are signed too. Moreover, it is possible to use more than one certificate to sign files and relationships.

The Open Packaging Conventions signing framework is quite flexible, and consequently some considerations must be taken, especially when defining policies. The Open Packaging Conventions specification does not define policies, only mentions that "designers that include digital signatures should define signature policies that are meaningful to their users". Besides guaranteeing authenticity, non-repudiation and integrity, digital signatures shall also be used to define policies that are intended by the signers[38] (typically the package producers) or in agreement with the package consumers (e.g. consumer will only accept package with signed content). The decisions taken during the signature process impact which consequent operations can be verified (e.g. allowing post-modification of a file, adding new relationships…).

There is no need of a separate file in the package about policies, because these policies information can be retrieved on how signing is performed. Signing a specific file in the package will implicitly express the intention of the signer on what is or isn't allowed with that file and related files (in case of relationship files). For instance, signing the aasx-origin relationship file will not permit adding new AAS to the package. If new AAS are added anyway, this will invalidate the original signature and nobody can blame the original signer for that change.

A package producer shall follow a digital signing policy based on the following options:

1. Sign nothing
2. Sign everything and thus following policy "No change allowed to the package" of Table 28.
3. Custom signing according to one or more policies of Table 28.

The package consumer may follow a validation process based on the policy of the signer(s) or an internal verification of the package according to its own policies. The signature policy defined by the signer(s) does not directly tell that the consumer should validate the package, but tells how it is intended by the signer(s). Nevertheless, validation might be mandatory for joint applications where several parties (package producers and consumers) need to follow the same rules. The following process for validation[39] for AASX packages is established:

1. The validation process must start by checking that the consumed package is according to the Open Packaging Conventions specification and that its implements the logical model according to the AASX definition. Optionally it may analyse if the physical model is according to the best-practice for storing files inside the AASX package.
2. Files that aren't source or target of any relationship, aren't allowed (besides the Open Packaging Conventions specific files).
3. After these steps, the existing certificates that were used to sign the content of the package must valid and trusted.
4. All signed content must then be verified and valid against the provided certificate information.
5. The signed content will also reveal a set of policies[40] defined by the signers or defined in agreement with the several parties (package producers and consumers), that must be followed by the consumer when changing the package without invalidating it (see Table 28).
6. A package is only valid, if all previous steps are performed successfully. Any change done to the package by the consumer requires a revalidation of the package.

---

[37] Individual files and relationships can be signed, but not the full package. This is a question of definition, but signing the full package could mean to sign all files inside the package (except the signature file).

[38] The policies described here are for the AASX package and what can be changed. It does not define any policy e.g. on how to use an AAS.

[39] Validation. The assurance that a product, service, or system meets the needs of the customer and other identified stakeholders. It often involves acceptance and suitability with external customers. Contrast with verification, which is often an internal process. (Adapted from The PMBOK guide, a standard adopted by IEEE, 4th edition)

[40] These policies are for the AASX package and not for the AAS itself.

Any of the steps mentioned in the validation can be performed independently without the other ones, but doing so, it is not considered as validation (e.g. internal verification process by a consumer may only require to check if the package is according to the Open Packaging Conventions and implementing the AASX logical model, without checking the signatures).

Table 28 Set of possible policies based on how package files are signed, how to enable a given policy and the consequences of a policy

	Policy	How to enable the policy	Consequence
General	No change allowed to the package	Sign all files and relationships in the package (except for [Content_Types].xml[41] and the signature file(s))	Invalidates any change in the signed files. New files that are added afterwards do not have a signature and aren't mentioned in any relationship, thus invalidating those files.
	No change allowed to the content of file X or deletion of file X	Sign file X inside the package (e.g. AAS, a submodel file, any file, …)	Invalidates tampering the content or deletion of file X.
	No change allowed to a relationship X	Sign relationship X	Invalidates tampering or deletion of the relationship entry (i.e. the relationship type, id and target URI) in the corresponding relationship file. This will not invalidate the content of source and target files of a relationship, once tampered.
	No change allowed to any relationships that have source file X	Sign relationship file X (X.rels)	Invalidates adding, changing or removing any relationship mentioned in that relationship file. This will also invalidate the addition of new files that would otherwise being target in that relationship file. For example, if there is no relationship for the thumbnail in the root relationship file before the signing of that file, a posterior addition of thumbnail relationship is then invalidated.
	Enable digital signatures	The digital-signature/origin relationship must be signed (alternatively, sign the complete root relationship file that contains this relationship)	Will enable digital signatures (but does not specify the rules for signing, e.g. if new signatures can be added).
	Enable core-properties	The metadata/core-properties relationship must be signed (alternatively, sign the	Will enable the core-properties of the package.

[41] When reading an AASX package, do not rely on the trustability of the file [Content_Types}.xml, as it was not possible to sign this file.

		complete root relationship file that contains this relationship)	
	Enable thumbnail	The metadata/thumbnail relationship must be signed (alternatively, sign the complete root relationship file that contains this relationship)	Will enable the thumbnail for the package.
	Forbid counter-signatures (adding new signatures)	Sign the signature origin relationship file	Invalidates counter-signatures.
	Forbid modifying existing file/relationship digests for signatures based on a certificate	Sign object inside the corresponding signature file that contains all the file/relationship digests	Invalidates any change in the digests and addition of new file digests.
AASX-specific	Enable AASX specification	The aasx-origin relationship must be signed (alternatively, sign the complete root relationship file that contains this relationship).	Will enable the AASX specification on top of the Open Packaging Conventions.
	Forbid adding a new AAS or removing an existing AAS.	Sign the aasx-origin relationship file	Invalidates adding or removing AAS.
	Forbid adding a new splittable parts or removing an existing one to/from an AAS	Sign the aas-spec relationship file	Invalidates adding or removing of splittable parts to/from an AAS.
	Forbid adding a new supplementary file or removing an existing one to/from an AAS	Sign the aas-spec relationship file	Invalidates adding or removing of extra files to/from an AAS.

7.8 Encryption

The Open Packaging Conventions specification (ISO/IEC 29500-2:2012) mentions that "ZIP-based packages shall not include encryption as described in the ZIP specification. Package implementers shall enforce this restriction. [M3.9]"[42]. However, an Open Packaging Conventions package may be encrypted with other means and some applications using this package format as the basis for a more specific format, may use encryption during interchange or DRM for distribution. [24]

An example is the Office Document Cryptography Structure (MS-OFFCRYPTO) used by derivate office formats. Some used technologies may be covered by Patents from Microsoft and therefore it isn't recommended for the AASX format. Digital Rights Management (DRM) can also be used to encrypt content elements in a package with specific access rights granted to authorize users (see the implementation in the system.io.packaging namespace [31]).

Regarding encryption and confidentiality, the following rules shall be followed:

1. Decide if there is a need of including confidential content in a package. If there is no reason, then the confidential content should not be included.

[42] The reason for this might be related to the transparency requirement for the package format as well as license requirements of PKWARE. For the ISO/IEC 21320-1 (Document Container File: Core) there is the following statement: "Encryption of individual files and of the central directory is prohibited. Hence this profile of ZIP_PK is more transparent than its parent format." [30]

2. If encryption is desired for a temporary communication act (e.g. e-mail exchange, …) or if a AASX needs to be stored somewhere so that it can be opened later by the same entity, then encryption methods can be used for that specific mean (e.g. use BitLocker when storing the AASX in Windows-based systems that support it, use S/MIME for exchanging encrypted e-mails between entities, etc.).
3. For all other use cases[43] where encryption is required for some or all of the content of the AASX:
    - Encryption methods can be used for individual files in the AASX package, as soon as the "encrypted" version replaces the original file in the package, the MIME type of the encryption format is known, and the MIME type must be listed in the [Content-Type].xml. The relationships as defined in this document remain the same, whenever content is encrypted or not. Note that Open Packaging Conventions related files as well as relationship files shall not be encrypted, and digital signing must be performed after encryption. One example of an encryption standard is the Secure MIME (S/MIME), where the encrypted content should be stored in application/pkcs7-mime format as defined in RFC 5652 and use the file extension *.p7m.
    - Besides encrypting the content of the package (individual files) it is possible to encrypt the full package (e.g. also using Secure MIME and saving the encrypted package in application/pkcs7-mime file format). In this case, the signature of the content of the package must be done before the encryption.

[43] A use case could be to encrypt a submodel and only provide the access to the unencrypted data after paying a fee.

8 Tools for the Asset Administration Shell

8.1 Open Source Tools

This clause gives some hints with respect to available open source tools supporting the creation and operating of an asset administration shell. It is not necessarily complete. There might be other implementations as well.

The aasx package explorer is an open source browser and editor for creating asset administration shells as .aasx packages [40]. The aasx package explorer supports the xml and JSON serialization of the asset administration shell. Additionally export formats for AutomationML or server generation for OPC UA are provided. But also additional export formats like BMEcat etc. are supported. Since it is an open source implementation new features are added continuosly. On [41] also additional information and code in the context of the asset administration shell can be found.

BaSyx, a software platform, is another open source impelementation for the asset administration shell and provides software developement kits for C++, C# and Java [42].

9 Summary and Outlook

In this document a metamodel for the structural viewpoint of the Asset Administration Shell is defined using UML. It covers security aspects as well as features for handling composite I4.0 Components. Data specification templates for defining concept descriptions for properties and physical units are provided.

Several serilializations and mappings are offered:

- XML and JSON for Exchange between partners via exchange format *.aasx*
- RDF for reasoning
- AutomationML for the engineering phase
- OPC UA for the operation phase

Additional parts of the document series cover:

- Interfaces and API for using a single AAS information model described in Part 1 (access, modify, query and execute information and active functionality)
- The infrastructure, which hosts and interconnects multiple Asset Administration Shells together. It implements registry, discovery services, endpoint handling and more.

Annex

Annex A. Concepts of the Administration Shell

i. General

In this clause, a general information is given about sources of information and relevant concepts for the Asset Administration Shell. Some of these concepts are explained in a general manner. Some concepts are update in order to reflect actual design decisions. No new concepts are introduced. Thus, the clause can be taken as a fully informative (annex) to the specification of the Administration Shell.

ii. Relevant Sources and Documents

The following documents were used to identify requirements and concepts for the Administration Shell:

- Implementation strategy of Plattform Industrie 4.0 [1][2]
- Aspects of the research roadmap in application scenarios [7]
- Continuation of the application scenarios [8]
- Structure of the Administration Shell [4] [18]
- Examples for the Administration Shell of the Industrie 4.0 Components [6]
- Technical Overview “Secure identities” [9]
- Security of the Administration Shell [14]
- Relationships between I4.0 components – Composite components and smart production [12]

Note: The global Industrie 4.0 glossary can be found at: https://www.plattform-i40.de/PI40/Navigation/EN/Industrie40/Glossary/glossary.html
Note: The online library of the Plattform Industrie 4.0 can be found at: https://www.plattform-i40.de/SiteGlobals/PI40/Forms/Listen/Downloads/EN/Downloads_Formular.html?cl2Categories_Typ_name=veroeffentlichung

iii. Basic concepts for Industrie 4.0

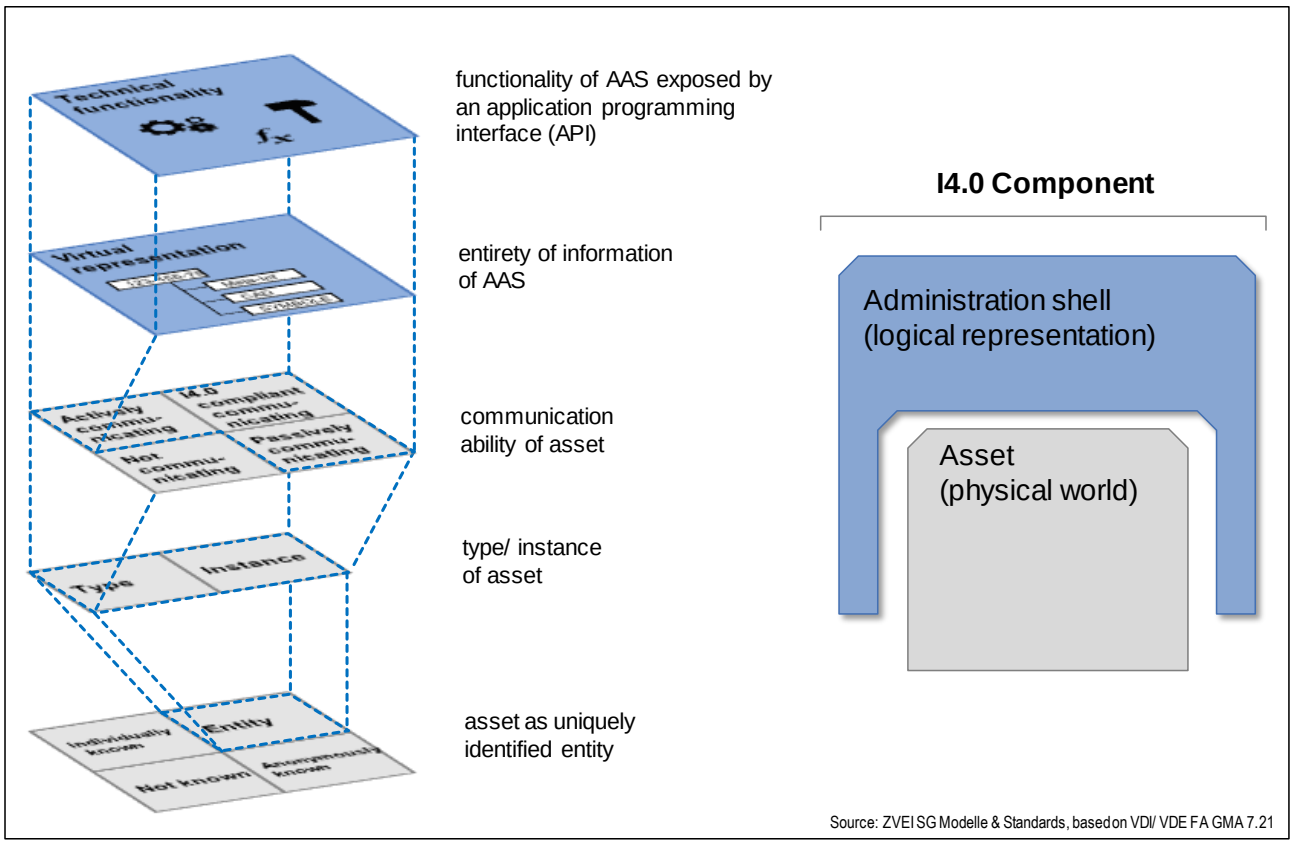
Industrie 4.0 describes concepts and definitions for the domain of smart manufacturing. For Industrie 4.0, the term asset, being any "object which has a value for an organization", is of central importance [2] [23]. Thus, assets in Industrie 4.0 can take almost any form, for example be a production system, a product, a software installation, intellectual properties or even human resources.

According [23], the "reference architecture model Industry 4.0 (RAMI4.0) provides a structured view of the main elements of an asset using a level model consisting of three axes [...]. Complex interrelationships can thus be broken down into smaller, more manageable sections by combining all three axes at each point in the asset’s life to represent each relevant aspect."

Assets shall have a logical representation in the "information world", for example shall be managed by IT-systems. Thus, an asset has to be precisely identified as an entity, shall have a "specific state within its life (at least a type or instance)", shall have communication capabilities, shall be represented by means of information and shall be able to provide technical functionality [23]. This logical representation of an asset is called Administration Shell [4]. The combination of asset and Administration Shell forms the so-called I4.0 Component. In international papers [18], the term smart manufacturing replaces the term Industrie 4.0.

For the large variety of assets in Industrie 4.0, the Administration Shell allows handling of these assets in the information world in always the same manner. This reduces complexity and allows for scalability. Additional motivation can be found in [2] [4] [7] [8].

Figure 129 Important concepts of Industrie 4.0 attached to the asset [2] [23]. I4.0 Component to be formed by Administration Shell and asset.



iv. The Concept of Properties

According [20], the "IEC 61360 series provides a framework and an information model for product dictionaries. The concept of product type is represented by 'classes' and the product characteristics are represented by 'properties'".

Such properties are standardized data elements. The definitions of such properties can be found in a range of repositories, such as IEC CDD (common data dictionary) or eCl@ss. The definition of a property (aka standardized data element type, property type) associates a worldwide unique identifier with a definition, which is a set of well-defined attributes. Relevant attributes for the Administration Shell are, amongst other, the preferred name, the symbol, the unit of measure and a human-readable textual definition of the property.

Figure 130 Exemplary definition of a property in the IEC CDD

Code:	0112/2///62683#ACE424
Version:	001
Revision:	01
IRDI:	0112/2///62683#ACE424#001
Preferred name:	rated current
Synonymous name:	
Symbol:	In
Synonymous symbol:	
Short name:	
Definition:	maximum uninterrupted current equal to the conventional free-air thermal current (Ith)
Note:	
Remark:	
Primary unit:	A
Alternative units:	
Level:	
Data type:	LEVEL(MAX) OF REAL_MEASURE_TYPE

The instantiation of such definition (just 'property', property instance) typically associates a value to the property. By this mechanism, semantically well-defined information can be conveyed by the Administration Shell.

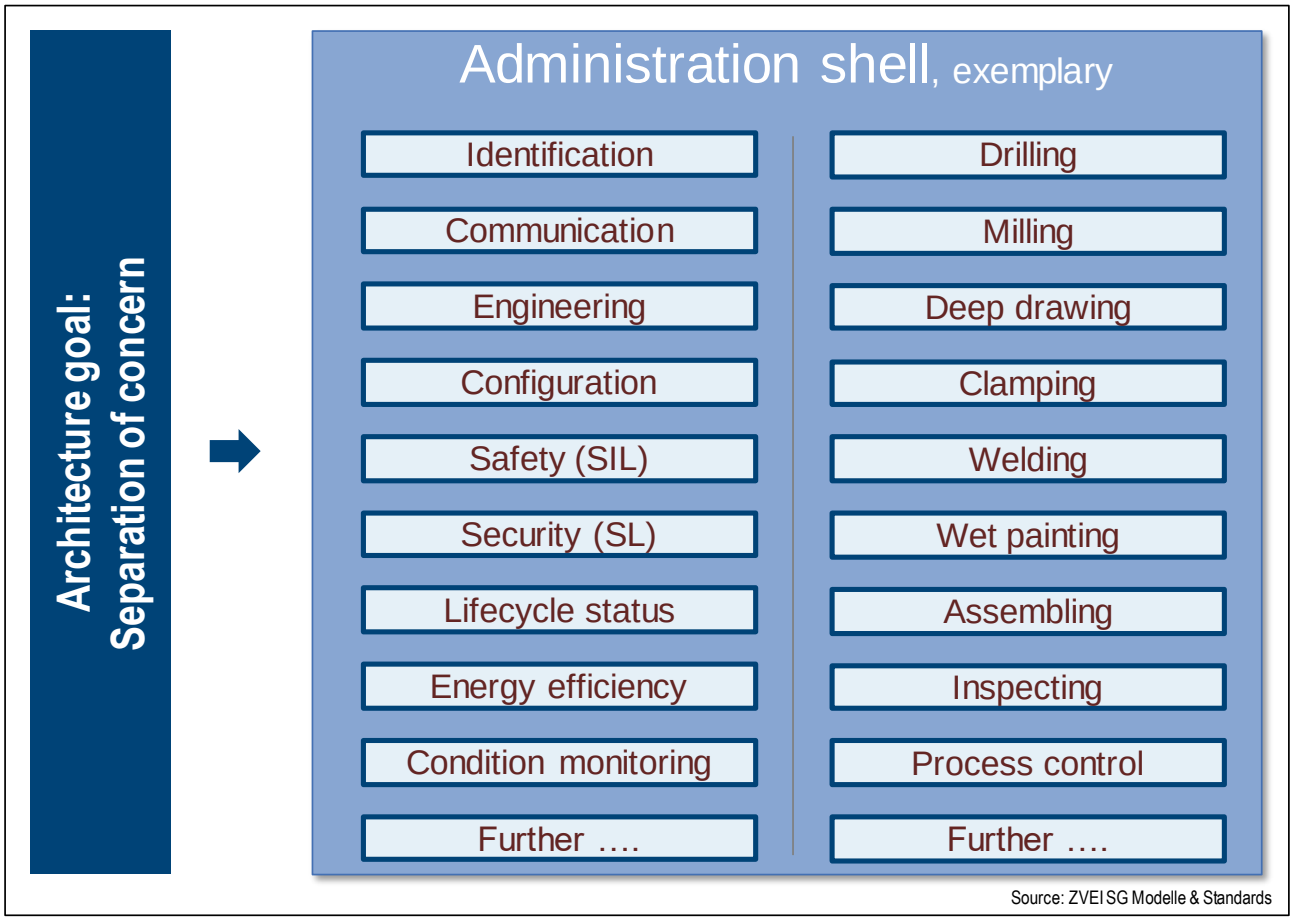
Note: Industrie 4.0 and smart manufacturing in general will require many properties which are beyond the current scope of IEC CDD, eCl@ss or other repositories. It is expected, that these sets of properties will be introduced, as more and more domains are modelled and standardized (next clause).

v. The Concept of Submodels

"The Administration Shell is the standardized digital representation of the asset, corner stone of the interoperability between the applications managing the manufacturing systems" [18]. Thus, it needs to provide a minimal but sufficient description according to the different application scenarios in Industrie 4.0 [7] [8]. Many different (international) standards, consortium specifications and manufacturer specifications can already contribute to this description [18].

As the figure shows, information from different many different technical domains could be associated with a respective asset and thus, many different properties are required to be represented in Administration Shells of future I4.0 Components. In order to manage these complex set of information, submodels provide a separation of concern.

Figure 131 Examples of different domains providing properties for submodels of the Administration Shell



The Administration Shell is thus made up of a series of submodels [4]. These represent different aspects of the asset concerned; for example, they may contain a description relating to safety or security [14] but could also outline various process capabilities such as drilling or installation [6].

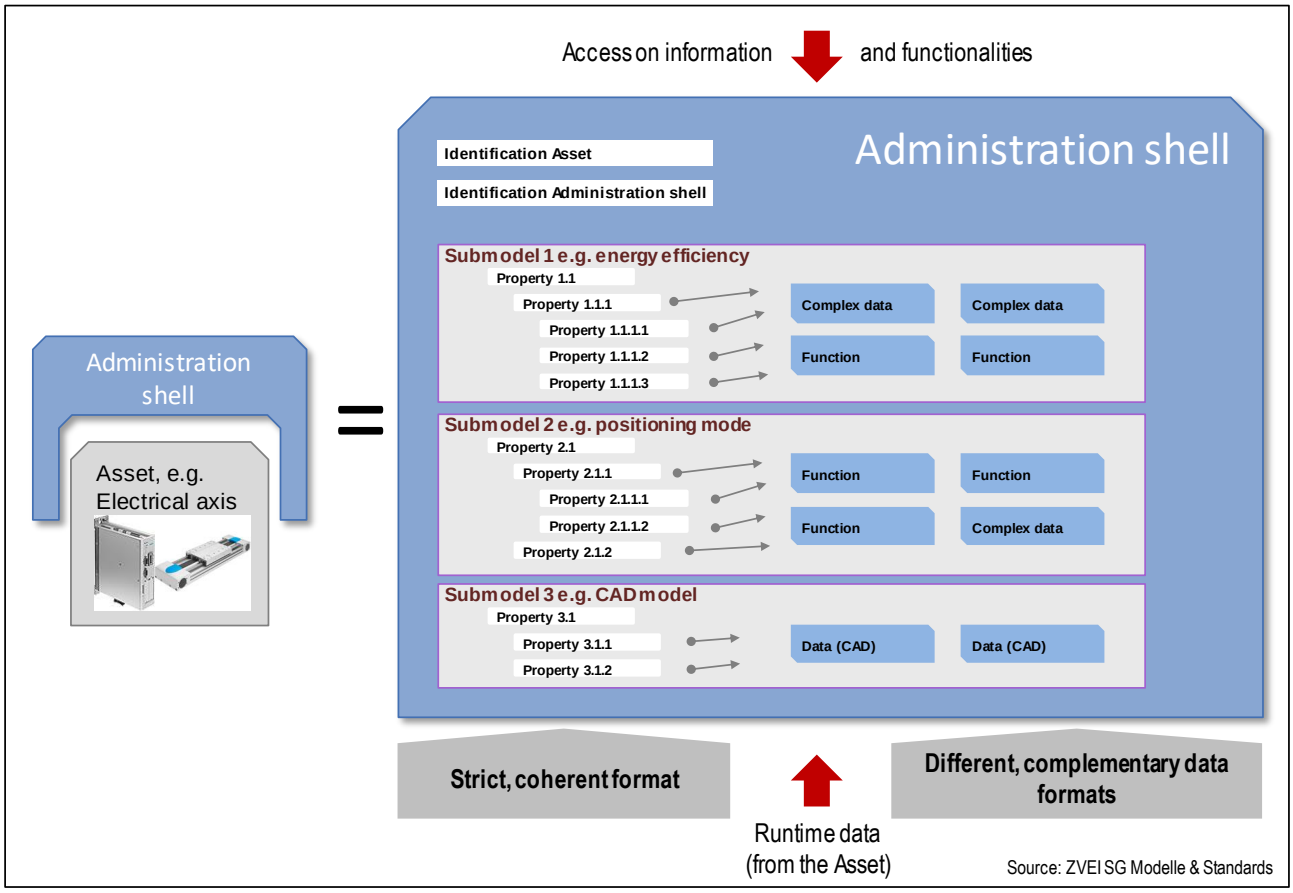
From the perspective of interoperability, the aim is to standardise only a single submodel for each aspect / technical domain. For example, it will thus be possible to find a drilling machine by searching for an Administration Shell containing a submodel “Drilling” with appropriate properties. For communication between different I4.0 components, certain properties can then be assumed to exist. In an example like this, a second submodel, “energy efficiency”, could then ensure that the drilling machine is able to cut its electricity consumption when it is not in operation.

Note: side benefit of the Administration Shell will be to simplify the update of properties from product design (and in particular system design) tools, update of properties from real data collected in the instances of assets, improve traceability of assets along life cycle and help certify assets from data.

vi. Basic Structure of the Asset Administration Shell

The document on the Structure of the Asset Administration Shell [4] [18] presented a rough, logical view of the AssetAdministration Shell’s structure. The AssetAdministration Shell – shown in blue in the following figure – comprises different sets of information. Both, the asset and the Administration Shell are identified by a globally unique identifier. It comprises a number of submodels for a characterisation of the AssetAdministration Shell.

Figure 132 Basic structure of the AssetAdministration Shell



Properties, data and functions will also contain information which not every partner within a value-added network or even within an organisational unit should be able to access or whose integrity and availability should be guaranteed. Therefore the structure of the Administration Shell shall be able to handle aspects such as access protection, visibility, identity and rights management, confidentiality and integrity. Information security needs to be respected and has to be aligned with an overall security concept. Implementation of security must go together with the implementation of other components of an overall system.

Each submodel contains a structured quantity of properties that can refer to data and functions. A standardized format based on IEC 61360-1/ ISO 13584-42 is envisaged for the properties. Thus, property value definition shall follow the same principles as also ISO 29002-10 and IEC 62832-2. Data and functions may be available in various, complementary formats.

The properties of all the submodels therefore result in a constantly readable directory of the key information of the Administration Shell and thus of the I4.0 component. To enable binding semantics, Administration Shells, assets, submodels and properties must all be clearly identified. Permitted global identifiers are IRDI (e.g. in ISO TS 29002-5, eCl@ss and IEC Common Data Dictionaries) and URIs (Unique Resource Identifiers, e.g. for ontologies).

It should be possible to filter elements of the Administration Shell or submodels according to different given views (→ Example C.4 in [18]). This facilitates different perspectives or use-cases for the application of Administration Shell's information.

vii. Requirements

This section collects the requirements from various documents that have impact on the specific structure of the Administration Shell. These requirements serve as input for the specific description of the structures of the Administration Shell.

The following requirements are taken from the document “Implementation strategy of Plattform Industrie 4.0” [2]. They are marked “STRAT”. The "Tracking" column validates the requirements by linking to features of the UML metamodel or this document in general.

ID	Requirement	Tracking
STRAT#1	A network of Industrie 4.0 components must be structured in such a way that connections between any end point (Industrie 4.0 components) are possible. The Industrie 4.0 components and their contents are to follow a common semantic model.	Network possible but not scope of this part of the document series. Common semantic model realized by domain specific submodels (*HasSemantics*/ *ConceptDescription* and by *Relations*)
STRAT#2	It must be possible to define the concept of an Industrie 4.0 component in such a way that it can meet requirements with different focal areas, i. e. “office floor” or “shop floor”.	Content-wise, many different submodels possible.
STRAT#3	Industrie 4.0 compliant communication must be performed in such a way that the data of a virtual representation of an Industrie 4.0 component can be kept either in the object itself or in a (higher level) IT system.	Metamodel and information representation independent of any deployment scenario.
STRAT#4	In the case of a virtual representation of an I4.0 component in a higher-level system, an integrity association must be ensured between the asset and its representation.	Integrity part of security approach.
STRAT#5	A suitable reference model must be established to describe how a higher level IT system can make the Administration Shell available in an Industrie 4.0 compliant manner (SOA approach, delegation principle).	Scope of upcoming part of the document series; not scope of this part.
STRAT#6	A description is required of how the Administration Shell can be “transported” from the originator (e.g. component manufacturer or electrical designer) to the higher level IT system (e.g. as an attachment to an email).	Hierarchical representation by XML/ JSON and package file format allow for different transport scenarios.
STRAT#7	Depending on the nature of the higher level systems, it may be necessary for the administration objects to allow for deployment in more than one higher level IT system.	Metamodel and information representation independent of any deployment scenario.
STRAT#8	The Industrie 4.0 component, and in particular the Administration Shell, its inherent functionality and the protocols concerned are to be “encapsulation-capable” or "separable" from any field busses in use.	Metamodel and information representation independent of any communication scenario.
STRAT#9	The aim of the Industrie 4.0 component is to detect non-Industrie 4.0 compliant communication relationships leading to or from the object’s Administration Shell and to make them accessible to end-to-end engineering.	Non-Industrie 4.0 compliant communication relationships could be modelled by submodels and therefore made available.
STRAT#10	It should be possible to logically assign other Industrie 4.0 components to one Industrie 4.0 component (e.g. an entire machine) in such a way that there is (temporary) nesting.	*References* and preparations for *Composite components* [12]
STRAT#11	Higher level systems should be able to access all Industrie 4.0 components in a purpose-driven and restrictable manner, even when these are (temporarily) logically assigned.	Scope of upcoming part of the document series; not scope of this part.

STRAT#12	Characteristics (1) Identifiability	Given by *Identifiable*
STRAT#13	Characteristics (2) I4.0-compliant communication	Not scope of part 1
STRAT#14	Characteristics (3) I4.0-compliant services and multiple status	Standardisation of submodels
STRAT#15	Characteristics (4) Virtual description	Available by digital representation (*Submodel* and *SubmodelElements*)
STRAT#16	Characteristics (5) I4.0-compliant semantics	HasSemantics
STRAT#17	Characteristics (6) Security and safety	Security by Attribute Based & Role Based Access. Safety not scope of part 1
STRAT#18	Characteristics (7) Quality of services	Metamodel and information representation independent of any communication scenario.
STRAT#19	Characteristics (8) Status	Standardisation of *submodels*
STRAT#20	Characteristics (9) Nestability	Supported by *billOfMaterial* of an Asset and RelationshipElements
STRAT#21	The minimum infrastructure must satisfy the principles of Security by Design (SbD).	Security by Attribute Based & Role Based Access.

The following requirements are taken from the document “The Structure of the Administration Shell:

Trilateral perspectives from France, Italy and Germany” [18]. They are marked “tAAS”.

Note: The term “property” was used in a very broad sense in previous publications of the Plattform Industrie 4.0. The metamodel in this document distinguishes between properties in a more classical sense as data element like “maximum temperature” and other submodel elements like operations, events etc.

Source	Requirement	Tracking
tAAS-#1	The Administration Shell shall accept properties from different technical domains in mutually distinct submodels that can be version-controlled and maintained independently of each other.	*Identifiable* *AdministrativeInformation* *Submodel* Requirements tAAS-#1 implicitly contains the requirements of versioning. Versioning is supported for all elements inheriting from *Identifiable*. Requirement tAAS-#1 is fulfilled because several submodels per AAS are possible. Every submodel is identifiable and an *Identifiable* may contain administrative information (*AdministrativeInformation*) for versioning. The reason for submodels to be identifiable is that they may be maintained independently of other submodels (Requirement tAAS-#1) and that they can be reused within different AAS. However, since submodel elements may refer to elements from other AAS dependencies have to be considered in parallel development and before reuse.
tAAS-#2	The Administration Shell should be capable of including properties from a wide range of technical domains and of [sic!] identify which domain they derive from.	*HasSemantics* Via semantic references property definitions from different dictionaries and thus different domains can be used within submodels.

		The only thing required is that the domain a property is derived from has a unique id (*semanticId*).
tAAS-#3	For finding definitions within each relevant technical domain, different procedural models should be allowed that respectively meet the requirements of standards, consortium specifications and manufacturer specifications sets.	*HasSemantics/semanticId* (see tAAS-#2) *ConceptDescription* Proprietary manufacturer specific property – or more general – concept descriptions or copies from external dictionaries are supported by defining *ConceptDescriptions*. They are referenced in *semanticId* via their global *id*. Up to now there is only a predefined data specification template for *Property* elements (*DataSpecification_IEC61360*). Usage of proprietary concept descriptions is not recommended because then interoperability cannot be ensured.
tAAS-#4	Different Administration Shells in respect of an asset must be capable of referencing each other. In particular, elements of an Administration Shell should be able to play the role of a "copy" of the corresponding components from another Administration Shell.	*AssetAdministrationShell.derivedFrom* The *derivedFrom* relationship is especially designed for supporting the relationship between an Asset Administration Shell representing an asset type and the Asset Administration Shells representing the asset instances of this asset type. See also tAAS-#16
tAAS-#5	Individual Administration Shells should, while retaining their structure, be combined into an overall Administration Shell.	*Asset/billOfMaterial* *RelationshipElement* The billOfMaterial fo an asset defined co-managed and self-managed entieis the asset is composed of. Via the submodel element "*RelationshipElement*" relations between entities can be defined.
tAAS-#6	Identification of assets, Administration Shells, properties and relationships shall be achieved using a limited set of identifiers (IRDI, URI and GUID), providing as far as possible offer global uniqueness.	*Identifiable* *Identification/idType* Requirement tAAS-#6 is fulfilled for all elements inheriting from *Identifiable*. For example, this is the case for *Asset*, *AssetAdministrationShell* and for concept descriptions. However, properties (like any other submodel element) are only referable. However, unique referencing is possible via the unique submodel id and the *Reference* via *Key*s concept. The supported id types include IRDI, URI (=*IRI*), IRI and GUID (=*Custom*) as requested.
tAAS-#7	The Administration Shell should allow retrieval of alternative identifiers such as a GS1 and GTIN identifier in return to asset ID (deferencing).	*Asset.assetIdentificationModel* Every asset has a globally unique identifier. Besides this global identifier additional local identifiers can be specified within a special submodel called "*assetIdentificationModel*". The asset identification model itself is not predefined by the metamodel. This means there is the need to define a submodel that

		can contain alternative identifiers including semantic references to know the meaning of the additional identifier.
tAAS-#8	The Administration Shell consists of header and body.	*AssetAdministrationShell* *AssetAdministrationShell/identification* *AssetAdministrationShell/administration* *AssetAdministrationShell/asset* The Asset Administration Shell does not explicitly distinguish between Header and Body. However, the Asset Administration Shell has attributes defined that belong to itself like the global unique id (*identification*), version information (*administration*), a mandatory reference to the asset (*asset*) it represents etc.
tAAS-#9	The header contains information about the identification.	*AssetAdministrationShell/asset* The Asset Administrative Shell is representing an asset with a unique id. See also tAAS-#7 See also tAAS-#13
tAAS-#10	The body contains information about the respective asset(s).	*AssetAdministrationShell/submodels* All submodels give information with respect to or related to the asset presented by the AAS. Note: An Asset Administration Shell is representing exactly one asset. In case of a Composite Asset Administration Shell it is implicitly representing several assets (see also tAAS-#5).
tAAS-#11	The information and functionality in the Administration Shell is accessible by means of a standardized application programming interface (API).	Will be covered in future parts of the document series
tAAS-#12	The Administration Shell has a unique ID.	*AssetAdministrationShell/identification/id* Since AssetAdministrationShell inherits from *Identifiable* Requirement tAAS-#12 is fulfilled.
tAAS-#13	The asset has a unique ID.	*Asset/identification/id* Since Asset inherits from *Identifiable* Requirement tAAS-#13 is fulfilled. See also Requirement tAAS-#7. Since *Asset* does not contain any specific attributes mandatory and only suitable for sensors etc. also more complex assets like industrial facilities can be modelled (Requirement tAAS-#14). The only assumption is that the industrial facility also has a globally unique id. Note: See also Composite Asset Administration Shell (see tAAS-#5) that allows the modelling of complex assets consisting of other assets that

		are represented by an AAS each by themselves.
tAAS-#14	An industrial facility is also an asset, it has an Administration Shell and is accessible by means of ID.	*Asset* *Asset/identification/id* Since *Asset* does not contain any specific attributes mandatory and only suitable for sensors etc. also more complex assets like industrial facilities can be modelled. The only assumption is that the industrial facility also has a globally unique id. Note: See also Composite AssetAdministration Shell (see tAAS-#5) that allows the modelling of complex assets consisting of other assets that are represented by an AAS each by themselves.
tAAS-#15	Types and instances must be identified as such.	Attribute *kind=Type* or *kind=Instance* for Asset *AssetAdministrationShell/derivedFrom* With attribute kind of Asset Requirement tAAS-#15 is fulfilled and asset types can be distinguished from asset instances. Additionally a *derivedFrom* relationship can be established between the AAS for an asset instance and the AAS for the asset type.
tAAS-#16	The Administration Shell can include references to other Administration Shells or Smart Manufacturing information.	*ReferenceElement* *File* *Blob* *AssetAdministrationShell/derivedFrom* The *derivedFrom* relationship between two AAS is special and is for example used to establish a relationship between asset instances and the asset type. For composite AAS (see tAAS-#5) there also is the relationship to AAS the composite AAS is composed of. The *ReferenceElement* is very generic and can reference another AAS as well as information within another AAS or even some information that is completely outside any AAS (as long as it has a global unique id). Files and BLOB can be used as submodel elements to include very generic manufacturing information that is not or cannot be modelled via properties or the other submodel elements defined for the Asset Administration Shell.
tAAS-#17	Additional properties, e. g. manufacturer specific, must be possible.	*HasDataSpecification* *ConceptDictionary*

		Via Data Specification Templates additional attributes for assets, properties and other submodel elements, submodels, views and even the AssetAdministration Shell itself can be defined and checked by tools. New proprietary property descriptions can be locally added to the local concept dictionary of the AAS and used for semantic definition in properties or other submodel elements. An extension of the metamodel by defining proprietary classes inheriting from the defined classes of this metamodel is also possible. Via API (see tAAS-#11) new properties, other submodel elements and submodels can be added – assumed the corresponding access permissions are given.
tAAS-#18	A reliable minimum number of properties must be defined for each Administration Shell.	*hasKind* for *Submodel* and *SubmodelElements* A reliable minimum number of properties is defined by the metamodel itself. They are called (class) attributes. *HasKind (with kind=Template) for* Submodel and *submodel element*s enables the definition of submodel (element) templates. These templates are referenced via *semanticId*. Note: the term property within the metamodel has a special semantics and shall not be mixed with the implicitly available attributes of the different classes. Although these attributes as well might be based on existing standards they are no properties in the sense that a semantic reference can be added that defines the semantics externally: The semantics is defined for the metamodel itself in the class tables within this document.
tAAS-#19	The properties and other elements of information in the Administration Shell must be suitable for types and instances.	*HasKind* (with *kind=Template* or *kind=Instance*) for *Submodel* and *SubmodelElement* All elements inheriting from *HasKind* can distinguish between types and instances. This is especially true for *SubmodelElement* and *Submodel*. Note: Submodels or properties of *kind=Template* do not describe an asset of *kind=Type*. This is done via properties of *kind=Instance*.
tAAS-#20	There must be a capability of hierarchical and countable structuring of the properties.	*DataElementCollection* Requirement tAAS-#20 is fulfilled by collections of data elements. The collection can be further characterized whether it is ordered and whether it may contain duplicates. Collections are built recursively and thus contain other submodel elements of the same AAS. For referencing properties or other submodel elements of other AAS a reference (*ReferenceElement*) or relationship element (*RelationshipElement*) needs to be included as part of the complex property.

tAAS-#21	Properties shall be able to reference other properties, even in other Administration Shells.	*DataElementCollection* *ReferenceElement* *RelationshipElement* *OperationVariable* in *Operation* A reference element can either reference any other element that is referable (i.e. inheriting from *Referable*) within the same or another AAS. Or it can reference entities completely outside any AAS via its global id. Note: For referencing elements within the same AAS it is not always necessary to use a reference property. Depending on the context also submodel element collections, relations etc. might be more suitable. Within *operation*s also other elements are referenced or used as input or output argument via *OperationVariable*
tAAS-#22	Properties must be able to reference information and functions of the Administration Shell.	*Operation* See also tAAS-#21 Functions in the sense of executable entities are represented as *operation*s.

The following requirements have been derived from the document "Security of the Administrative Shell" [14]. They are marked as "SecAAS"

ID	Requirement	Tracking
SecAAS-#1	Identification and authentication: It must be ensured that the correct entities (Administration Shell and users) interact with each other. This applies both in a local communication context (within a machine or plant) and in a global context (across companies). The clear identification (by authentication) of the communication partners is a basic requirement for the interaction with a management shell. Without them, further security features (confidentiality, integrity, etc.) cannot be guaranteed.	*Security/certificate* Certificates are supported.
SecAAS-#2	User and rights management: An AssetAdministration Shell can have different interaction partners. To control the possibilities of interaction with the Administration Shell, a user and rights management is necessary.	*Security/accessControlPolicyPoints/policyAdministrationPoint* *AccessControl* *AccessControl/accessPermissionRules* There is no explicit subject management in the AAS: It is assumed that the identity of the subject requesting access with a given role (via the API - see tAAS-#11) is authenticated

		outside the AAS. The AAS can check the authorization via the endpoint to the subject attributes provider. For every object in the Asset Administration Shell access permission rules can be defined.
SecAAS-#3	Secure Communication: Communication with the Administrative Shell may include sensitive information. Likewise, a change in the communication between the Administration Shell and its communication partners can cause serious and dangerous disruptions in a machine or plant. It is therefore mandatory that adequate measures be taken to ensure communication security. This must be done by using appropriate security protocols.	Not applicable
SecAAS-#4	Event logging: The traceability of interaction with the Administration Shell plays a crucial role in the detection of security incidents. This traceability is achieved through logging / event logging and auditing. The management shell must therefore provide methods that log accesses and changes in state of the management shell without modification. It is also important to be able to centrally collect and evaluate this event information.	History handling will be detailed in future parts or versions of the document (series).

Annex B. Templates for UML Tables

In this annex, the templates used for element specification are explained.

Template for document classes (elements):

Class:				
Explanation:				
Inherits from:				
Attribute (*=mandatory)	Explanation	Type	Kind	Card.

Kind is defined with semantics of UML:

- attr: attribute (Type is no Object)
- aggr: aggregation (does not exist independent of its parent)
- ref: composition (does exist independent of its parent)

Additionally, there is kind:

- ref*: reference via “Reference” class with Key/type=<Type> for the last Key in the Reference.

For more information on referencing see Clause 4.7.21.

Card. is the cardinality. “*” denotes an arbitrary number of elements of the corresponding Type. “n..m” mean minimum n-times and maximum n times. For example “0..1” means optional.

Template for enumerations:

Enumeration:	
Explanation:	
Literal	Explanation

Annex C. Legend for UML Modelling

Figure 133 Aggregation in Metamodel in UML – Legend

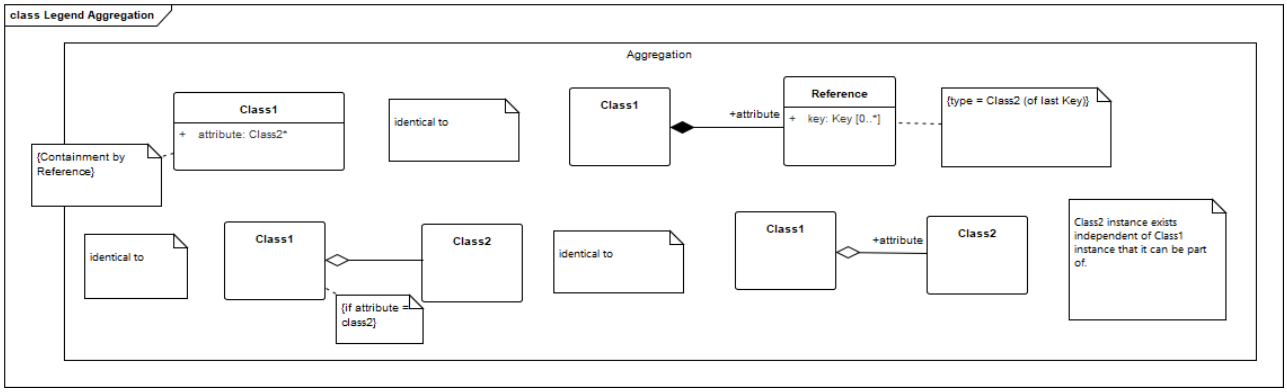


Figure 134 Association in Metamodel in UML - Legend

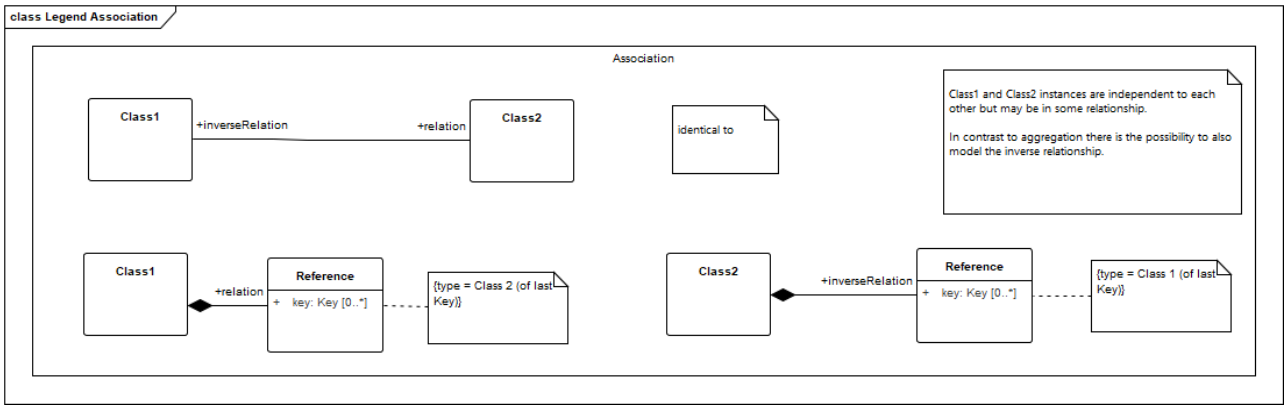


Figure 135 Composition in Metamodel in UML - Legend

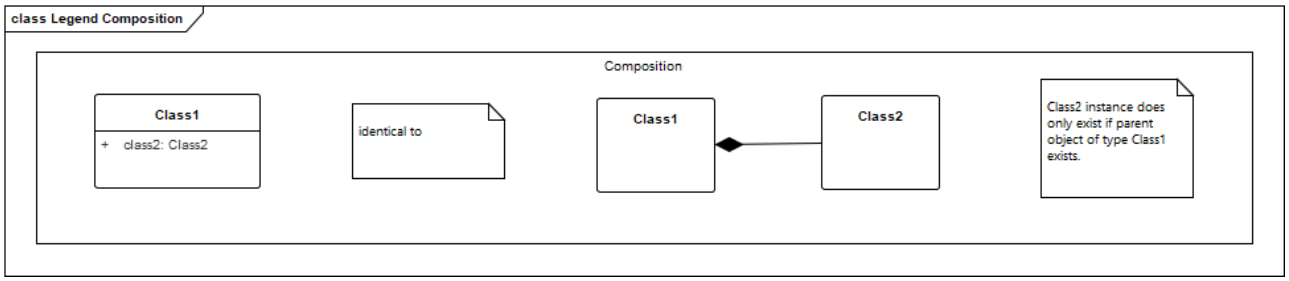


Figure 136 Identification in Metamodel in UML - Legend

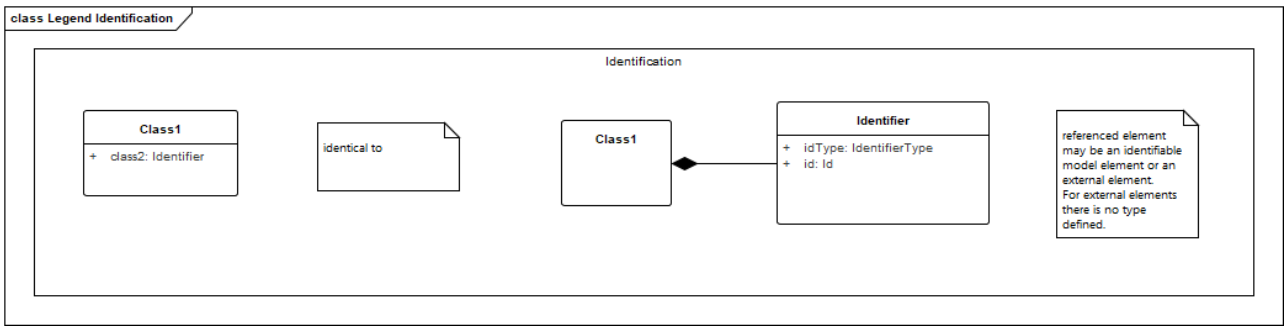


Figure 137 Inheritance Classes in Metamodel in UML - Legend

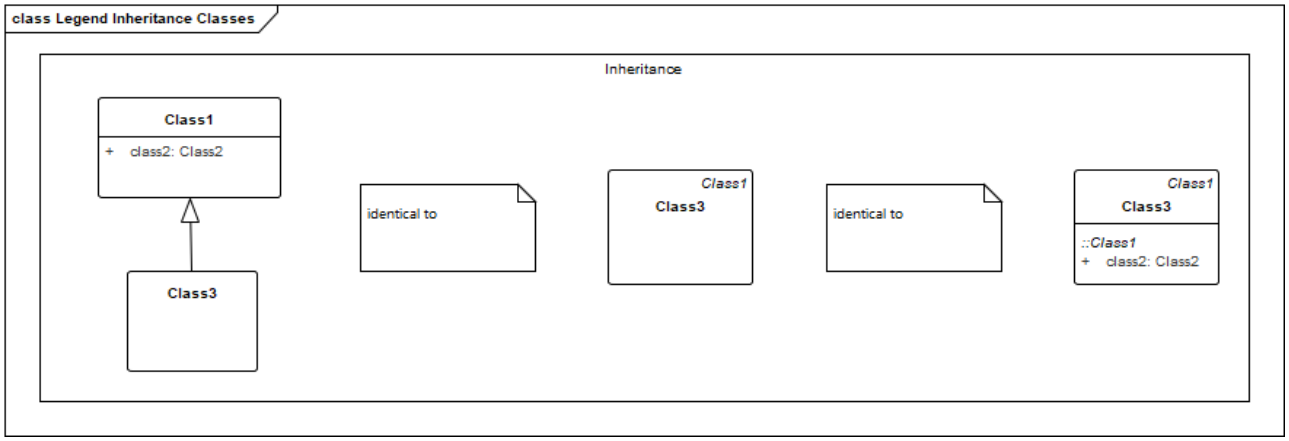
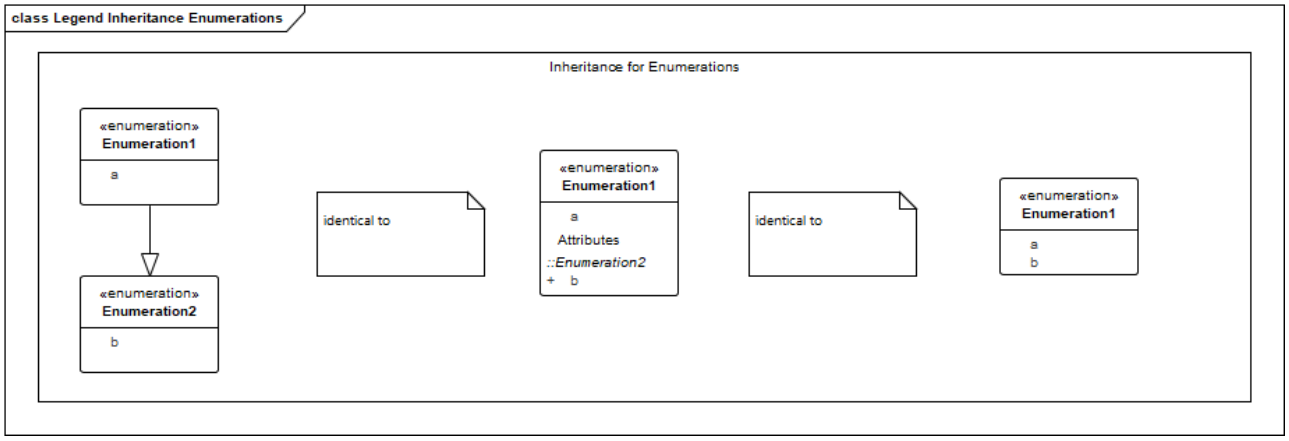


Figure 138 Inheritance Enumerations in Metamodel in UML - Legend



Annex D. Metamodel UML with inherited Attributes

In this annex some UML diagrams are shown together with all attributes inherited for better overview.

Figure 139 Core Model with inherited Attributes

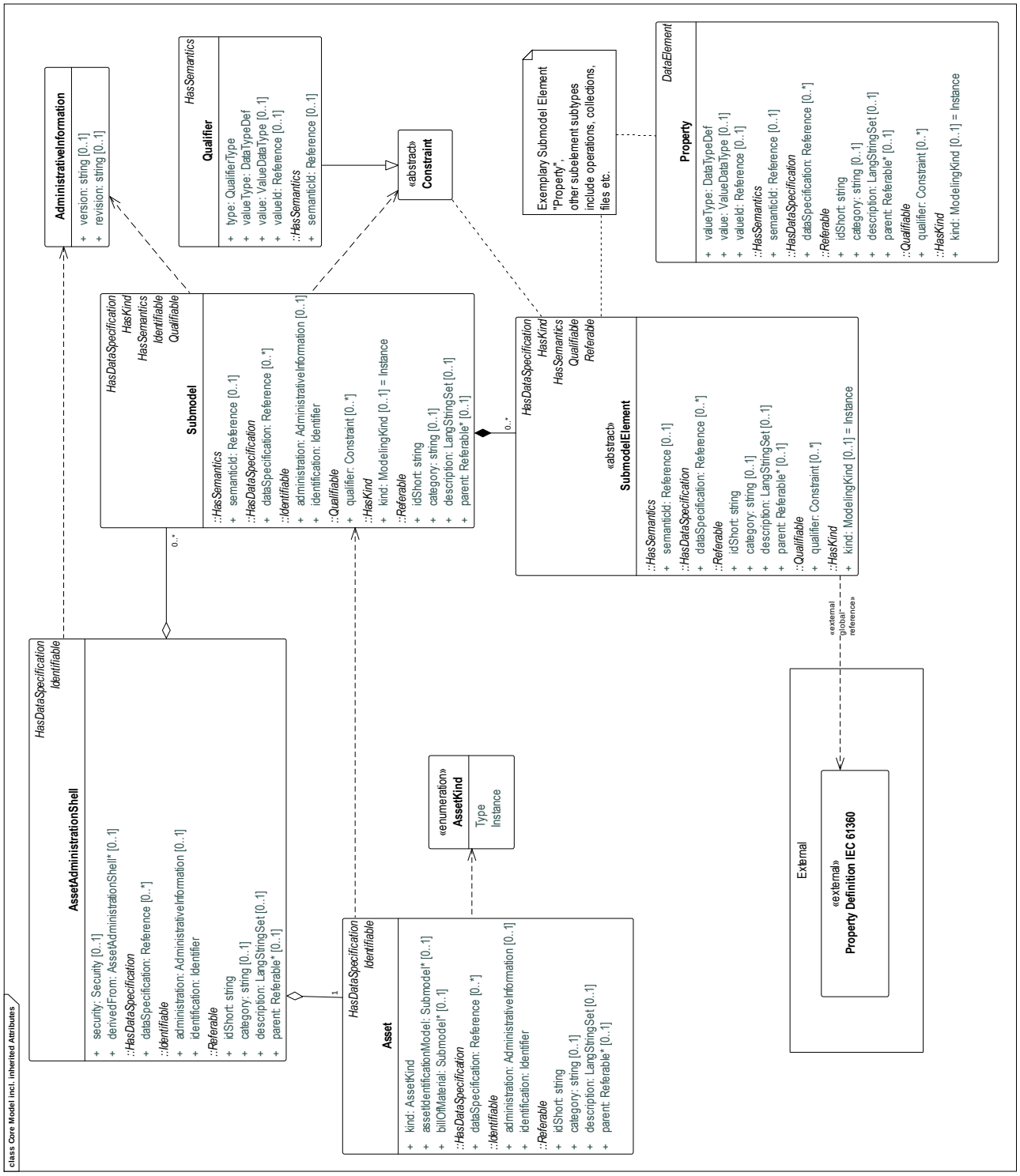


Figure 140 Operation with inherited Attributes

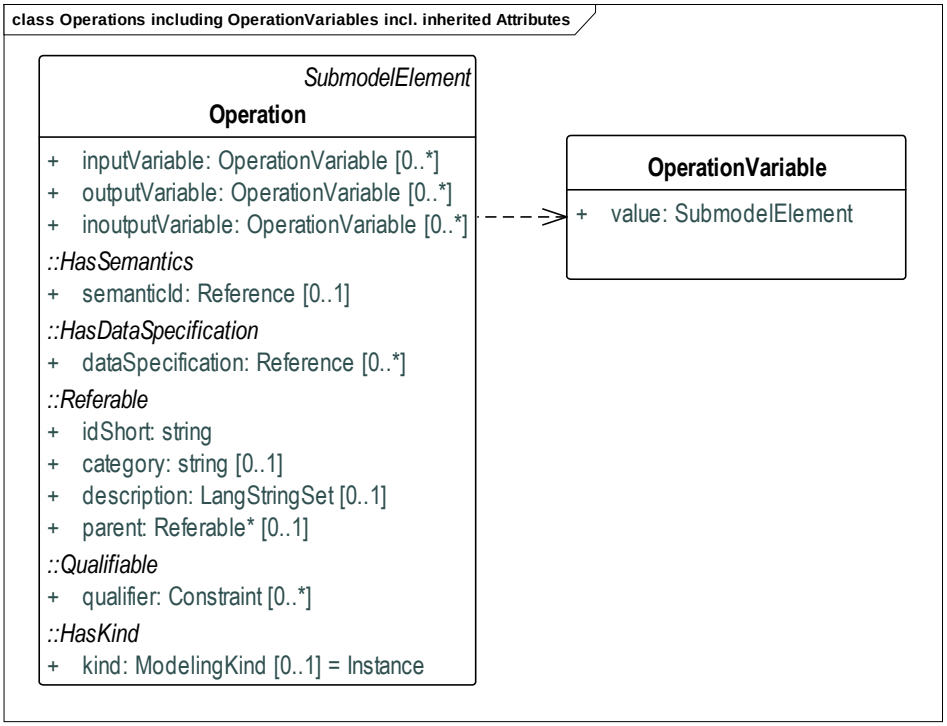


Figure 141 Access Control with inherited attributes

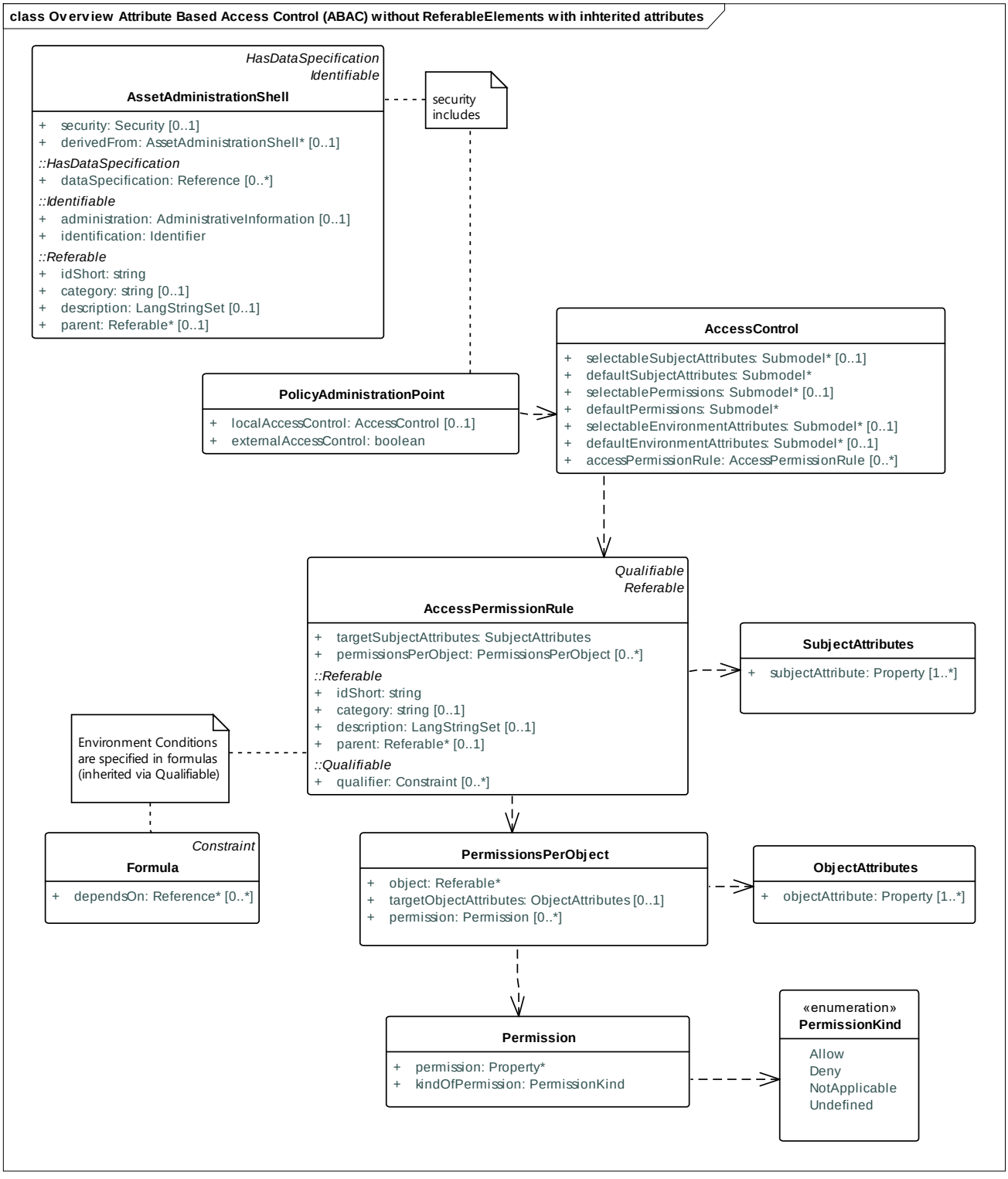
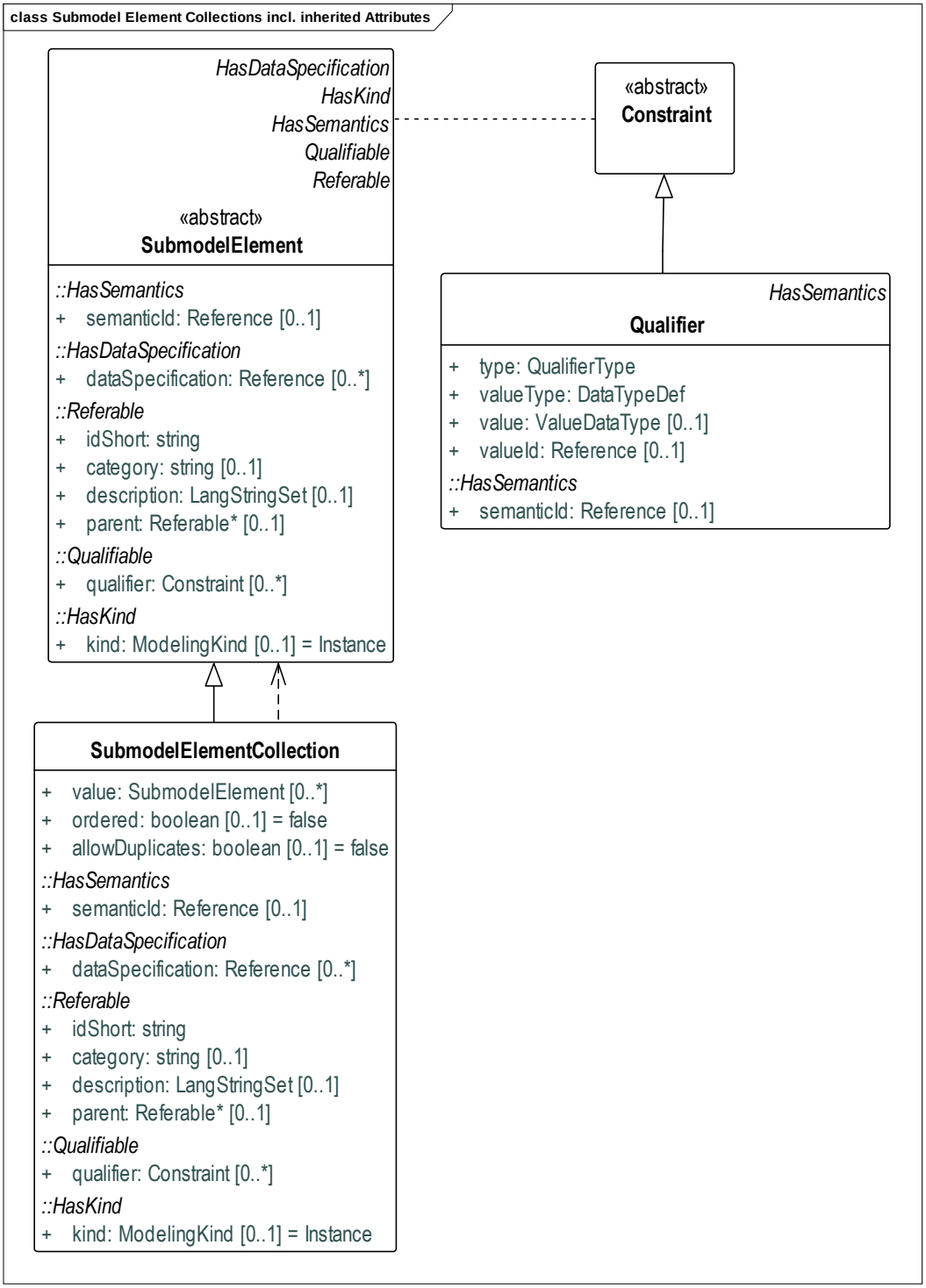


Figure 142 Submodel Element Collection with inheritance



Annex E. XML schemas and complete example

i. XML Schemas for Administration Shell

The schema is splitted into three parts:

- The main concepts of the Administration Shell (AAS.xsd)
- The Data Specification Template IEC61360 (IEC616360.xsd)
- The Asset based access control model (AAS_ABAC.x

Subsequently, an example in XML is discussed.

ii. Schema for overall Administration Shell

```
<?xml version="1.0" encoding="UTF-8"?>
<schema elementFormDefault="qualified" targetNamespace="http://www.admin-shell.io/aas/2/0" xmlns="http://www.w3.org/2001/XMLSchema" xmlns:aas="http://www.admin-shell.io/aas/2/0" xmlns:abac="http://www.admin-shell.io/aas/abac/2/0" xmlns:IEC61360="http://www.admin-shell.io/IEC61360/2/0">
    <import namespace="http://www.admin-shell.io/aas/abac/2/0" schemaLocation="AAS_ABAC.xsd"/>
    <import namespace="http://www.admin-shell.io/IEC61360/2/0" schemaLocation="IEC61360.xsd"/>
    <attributeGroup name="keyTypes">
        <attribute name="identifierType" use="optional">
            <simpleType>
                <restriction base="string">
                    <enumeration value="Custom"/>
                    <enumeration value="IRDI"/>
                    <enumeration value="IRI"/>
                </restriction>
            </simpleType>
        </attribute>
        <attribute name="localKeyType" use="optional">
            <simpleType>
                <restriction base="string">
                    <enumeration value="idShort"/>
                </restriction>
            </simpleType>
        </attribute>
    </attributeGroup>
    <complexType name="aasenv_t">
        <sequence>
            <element maxOccurs="1" minOccurs="0" name="assetAdministrationShells" type="aas:assetAdministrationShells_t" />
            <element maxOccurs="1" minOccurs="0" name="assets" type="aas:assets_t"/>
            <element maxOccurs="1" minOccurs="0" name="submodels" type="aas:submodels_t" />
            <element maxOccurs="1" minOccurs="0" name="conceptDescriptions" type="aas:conceptDescriptions_t"/>
        </sequence>
    </complexType>
```

```
    <complexType name="administration_t">
        <sequence>
            <element maxOccurs="1" minOccurs="0" name="version" type="string" />
            <element maxOccurs="1" minOccurs="0" name="revision" type="string"/>
        </sequence>
    </complexType>
    <complexType name="asset_t">
        <sequence>
            <group ref="aas:identifiable"/>
            <group ref="aas:hasDataSpecification"/>
            <element maxOccurs="1" minOccurs="0" name="assetIdentificationModelRef" type="aas:reference_t"/>
            <element maxOccurs="1" minOccurs="0" name="billOfMaterialRef" type="aas:reference_t"/>
            <element maxOccurs="1" minOccurs="0" name="kind" type="aas:assetKind_t"/>
        </sequence>
    </complexType>
    <complexType name="assetAdministrationShell_t">
        <sequence>
            <group ref="aas:identifiable"/>
            <group ref="aas:hasDataSpecification"/>
            <element maxOccurs="1" minOccurs="0" name="derivedFrom" type="aas:reference_t" />
            <element maxOccurs="1" minOccurs="1" name="assetRef" type="aas:reference_t" />
            <element maxOccurs="1" minOccurs="0" name="submodelRefs" type="aas:submodelRefs_t" />
            <element maxOccurs="1" minOccurs="0" name="views" type="aas:views_t" />
            <element maxOccurs="1" minOccurs="0" name="conceptDictionaries" type="aas:conceptDictionaries_t"/>
            <element maxOccurs="1" minOccurs="0" name="security" type="abac:security_t" />
    </sequence>
    </complexType>
    <complexType name="assetAdministrationShells_t">
        <sequence>
            <element maxOccurs="unbounded" minOccurs="0" name="assetAdministrationShell"
type="aas:assetAdministrationShell_t"/>
        </sequence>
    </complexType>
    <complexType name="assets_t">
        <sequence>
            <element maxOccurs="unbounded" minOccurs="0" name="asset" type="aas:asset_t"/>
        </sequence>
    </complexType>
    <complexType name="basicEvent_t">
        <complexContent>
            <extension base="aas:eventAbstract_t">
                <sequence>
                    <element maxOccurs="1" minOccurs="1" name="observed" type="aas:reference_t"/>
```

```
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="blob_t">
    <complexContent>
        <extension base="aas:submodelElementAbstract_t">
            <sequence>
                <element maxOccurs="1" minOccurs="0" name="value" type="aas:blobType_t"/>
                <element maxOccurs="1" minOccurs="1" name="mimeType" type="string"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="blobType_t">
    <simpleContent>
        <extension base="base64Binary"/>
    </simpleContent>
</complexType>
<complexType name="conceptDescription_t">
    <sequence>
        <group ref="aas:identifiable"/>
        <group ref="aas:hasDataSpecification"/>
        <element maxOccurs="unbounded" minOccurs="0" name="isCaseOf" type="aas:reference_t"/>
    </sequence>
</complexType>
<complexType name="conceptDescriptionRefs_t">
    <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="conceptDescriptionRef" type="aas:reference_t"/>
    </sequence>
</complexType>
<complexType name="conceptDescriptions_t">
    <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="conceptDescription" type="aas:conceptDescription_t"/>
    </sequence>
</complexType>
<complexType name="conceptDictionaries_t">
    <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="conceptDictionary" type="aas:conceptDictionary_t"/>
    </sequence>
</complexType>
<complexType name="conceptDictionary_t">
    <sequence>
```

```
            <group ref="aas:referable" />
            <element name="conceptDescriptionRefs" type="aas:conceptDescriptionRefs_t"></element>
        </sequence>
    </complexType>
    <complexType name="constraint_t">
        <choice>
            <element maxOccurs="1" minOccurs="0" name="formula" type="aas:formula_t"/>
            <element maxOccurs="1" minOccurs="0" name="qualifier" type="aas:qualifier_t"/>
        </choice>
    </complexType>
    <complexType name="constraints_t">
        <sequence>
            <element maxOccurs="unbounded" minOccurs="0" name="qualifiers" type="aas:constraint_t"/>
        </sequence>
    </complexType>
    <complexType name="containedElements_t">
        <sequence>
            <element maxOccurs="unbounded" minOccurs="0" name="containedElementRef" type="aas:reference_t"/>
        </sequence>
    </complexType>
    <complexType name="dataSpecificationContent_t">
        <choice>
            <element name="dataSpecificationIEC61360" type="IEC61360:dataSpecificationIEC61630_t"/>
        </choice>
    </complexType>
    <complexType name="dataTypeDef_t">
        <simpleContent>
            <extension base="string"/>
        </simpleContent>
    </complexType>
    <complexType name="embeddedDataSpecification_t">
        <sequence>
            <element maxOccurs="1" minOccurs="0" name="dataSpecificationContent" type="aas:dataSpecificationContent_t" />
            <element maxOccurs="1" minOccurs="0" name="dataSpecification" type="aas:reference_t"/>
    </sequence>
    </complexType>
    <complexType name="entity_t">
        <complexContent>
            <extension base="aas:submodelElementAbstract_t">
                <sequence>
                    <element name="assetRef" type="aas:reference_t"/>
                    <element name="entityType">
                        <simpleType>
```

```
                        <restriction base="aas:entityType_t">
                            <enumeration value="CoManagedEntity"/>
                            <enumeration value="SelfManagedEntity"/>
                        </restriction>
                    </simpleType>
                </element>
                <element name="statements" type="aas:submodelElement_t"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="eventAbstract_t">
    <complexContent>
        <extension base="aas:submodelElementAbstract_t"/>
    </complexContent>
</complexType>
<complexType name="file_t">
    <complexContent>
        <extension base="aas:submodelElementAbstract_t">
            <sequence>
                <element name="mimeType" type="string"/>
                <element name="value" type="aas:pathType_t"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="formula_t">
    <sequence>
        <element maxOccurs="1" minOccurs="0" name="dependsOnRefs" type="aas:references_t"/>
    </sequence>
</complexType>
<complexType name="identification_t">
    <simpleContent>
        <extension base="string">
            <attribute name="idType" use="optional">
                <simpleType>
                    <restriction base="string">
                        <enumeration value="Custom"/>
                        <enumeration value="IRDI"/>
                        <enumeration value="IRI"/>
                    </restriction>
                </simpleType>
            </attribute>
```

```
            </extension>
        </simpleContent>
    </complexType>
    <complexType name="identifier_t">
        <sequence>
            <element maxOccurs="1" minOccurs="1" name="id" type="string"/>
            <element maxOccurs="1" minOccurs="1" name="idType" type="aas:identifierType_t"/>
        </sequence>
    </complexType>
    <complexType name="idPropertyDefinition_t">
        <simpleContent>
            <extension base="string">
                <attribute name="idType" type="string"/>
            </extension>
        </simpleContent>
    </complexType>
    <complexType name="idShort_t">
        <simpleContent>
            <extension base="string"/>
        </simpleContent>
    </complexType>
    <complexType name="key_t">
        <simpleContent>
            <extension base="string">
                <attribute name="idType">
                    <simpleType>
                        <restriction base="string">
                            <enumeration value="Custom"/>
                            <enumeration value="FragmentId"/>
                            <enumeration value="IdShort"/>
                            <enumeration value="IRDI"/>
                            <enumeration value="IRI"/>
                        </restriction>
                    </simpleType>
                </attribute>
                <attribute name="local" type="boolean"/>
                <attribute name="type">
                    <simpleType>
                        <restriction base="string">
                            <enumeration value="AccessPermissionRule"/>
                            <enumeration value="AnnotatedRelationshipElement"/>
                            <enumeration value="Asset"/>
                            <enumeration value="AssetAdministrationShell"/>
```

```
                        <enumeration value="BasicEvent"/>
                        <enumeration value="Blob"/>
                        <enumeration value="Capability"/>
                        <enumeration value="ConceptDescription"/>
                        <enumeration value="ConceptDictionary"/>
                        <enumeration value="DataElement"/>
                        <enumeration value="Entity"/>
                        <enumeration value="Event"/>
                        <enumeration value="File"/>
                        <enumeration value="FragmentReference"/>
                        <enumeration value="GlobalReference"/>
                        <enumeration value="MultiLanguageProperty"/>
                        <enumeration value="Operation"/>
                        <enumeration value="Property"/>
                        <enumeration value="Range"/>
                        <enumeration value="ReferenceElement"/>
                        <enumeration value="RelationshipElement"/>
                        <enumeration value="Submodel"/>
                        <enumeration value="SubmodelElement"/>
                        <enumeration value="SubmodelElementCollection"/>
                        <enumeration value="View"/>
                    </restriction>
                </simpleType>
            </attribute>
        </extension>
    </simpleContent>
</complexType>
<complexType name="keys_t">
    <sequence>
        <element maxOccurs="unbounded" minOccurs="0" ref="aas:key"/>
    </sequence>
</complexType>
<complexType name="langString_t">
    <simpleContent>
        <extension base="string">
            <attribute name="lang" type="string"/>
        </extension>
    </simpleContent>
</complexType>
<complexType name="langStringSet_t">
    <sequence>
        <element maxOccurs="unbounded" minOccurs="1" name="langString" type="aas:langString_t"/>
    </sequence>
```

```
    </complexType>
    <complexType name="multiLanguageProperty_t">
        <complexContent>
            <extension base="aas:submodelElementAbstract_t">
                <sequence>
                    <element maxOccurs="1" minOccurs="0" name="valueId" type="aas:reference_t"/>
                    <element maxOccurs="1" minOccurs="0" name="value" type="aas:langStringSet_t"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="operation_t">
        <complexContent>
            <extension base="aas:submodelElementAbstract_t">
                <sequence>
                    <element maxOccurs="1" minOccurs="0" name="inoutputVariable" type="aas:operationsVariables_t"/>
                    <element maxOccurs="1" minOccurs="0" name="inputVariable" type="aas:operationsVariables_t"/>
                    <element maxOccurs="1" minOccurs="0" name="outputVariable" type="aas:operationsVariables_t"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="operationsVariables_t">
        <sequence>
            <element maxOccurs="unbounded" minOccurs="0" name="operationVariable" type="aas:operationVariable_t"/>
        </sequence>
    </complexType>
    <complexType name="operationVariable_t">
        <sequence>
            <element name="value" type="aas:submodelElement_t"/>
        </sequence>
    </complexType>
    <complexType name="pathType_t">
        <simpleContent>
            <extension base="string"/>
        </simpleContent>
    </complexType>
    <complexType name="property_t">
        <complexContent>
            <extension base="aas:submodelElementAbstract_t">
                <sequence>
                    <element maxOccurs="1" minOccurs="0" name="valueId" type="aas:reference_t"/>
                    <element maxOccurs="1" minOccurs="0" name="value" type="aas:valueDataType_t"/>
```

```
                    <element maxOccurs="1" minOccurs="1" name="valueType" type="aas:dataTypeDef_t"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="qualifier_t">
        <sequence>
            <element maxOccurs="1" minOccurs="0" name="valueId" type="aas:reference_t"/>
            <element maxOccurs="1" minOccurs="0" name="value" type="aas:valueDataType_t"/>
            <element maxOccurs="1" minOccurs="1" name="type" type="aas:qualifierType_t"/>
            <element maxOccurs="1" minOccurs="1" name="valueType" type="aas:dataTypeDef_t"/>
            <group ref="aas:hasSemantics"/>
        </sequence>
    </complexType>
    <complexType name="qualifierType_t">
        <simpleContent>
            <extension base="string"/>
        </simpleContent>
    </complexType>
    <complexType name="range_t">
        <complexContent>
            <extension base="aas:submodelElementAbstract_t">
                <sequence>
                    <element maxOccurs="1" minOccurs="0" name="max" type="aas:valueDataType_t"/>
                    <element maxOccurs="1" minOccurs="0" name="min" type="aas:valueDataType_t"/>
                    <element maxOccurs="1" minOccurs="1" name="valueType" type="aas:dataTypeDef_t"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="reference_t">
        <sequence>
            <element name="keys" type="aas:keys_t"/>
        </sequence>
    </complexType>
    <complexType name="referenceElement_t">
        <complexContent>
            <extension base="aas:submodelElementAbstract_t">
                <sequence>
                    <element name="value" type="aas:reference_t"/>
                </sequence>
            </extension>
        </complexContent>
```

```
    </complexType>
    <complexType name="references_t">
        <sequence>
            <element maxOccurs="unbounded" minOccurs="0" name="reference" type="aas:reference_t"/>
        </sequence>
    </complexType>
    <complexType name="relationshipElement_t">
        <complexContent>
            <extension base="aas:submodelElementAbstract_t">
                <sequence>
                    <element name="first" type="aas:reference_t"/>
                    <element name="second" type="aas:reference_t"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="semanticId_t">
        <complexContent>
            <extension base="aas:reference_t"/>
        </complexContent>
    </complexType>
    <complexType name="submodel_t">
        <sequence>
            <group ref="aas:identifiable"/>
            <group ref="aas:hasKind"></group>
            <group ref="aas:hasSemantics"></group>
            <group ref="aas:qualifiable"/>
            <group ref="aas:hasDataSpecification"/>
            <element name="submodelElements" type="aas:submodelElements_t"/>
        </sequence>
    </complexType>
    <complexType name="submodelElement_t">
        <choice>
            <element name="annotatedRelationshipElement" type="aas:relationshipElement_t"/>
            <element name="basicEvent" type="aas:basicEvent_t"/>
            <element name="blob" type="aas:blob_t"/>
            <element name="capability" type="string"/>
            <element name="entity" type="aas:entity_t"/>
            <element name="file" type="aas:file_t"/>
            <element name="multiLanguageProperty" type="aas:multiLanguageProperty_t"/>
            <element name="operation" type="aas:operation_t"/>
            <element name="property" type="aas:property_t"/>
            <element name="range" type="aas:range_t"/>
```

```
            <element name="referenceElement" type="aas:referenceElement_t"/>
            <element name="relationshipElement" type="aas:relationshipElement_t"/>
            <element name="submodelElementCollection" type="aas:submodelElementCollection_t"/>
        </choice>
    </complexType>
    <complexType name="submodelElementAbstract_t">
        <sequence>
            <group ref="aas:referable"/>
            <group ref="aas:hasKind"/>
            <group ref="aas:hasSemantics"/>
            <group ref="aas:qualifiable"/>
            <group ref="aas:hasDataSpecification"/>
        </sequence>
    </complexType>
    <complexType name="submodelElementCollection_t">
        <complexContent>
            <extension base="aas:submodelElementAbstract_t">
                <sequence>
                    <element name="allowDuplicates" type="boolean"/>
                    <element name="ordered" type="boolean"/>
                    <element name="value" type="aas:submodelElements_t"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <complexType name="submodelElements_t">
        <sequence>
            <element maxOccurs="unbounded" minOccurs="0" name="submodelElement" type="aas:submodelElement_t"/>
        </sequence>
    </complexType>
    <complexType name="submodelRefs_t">
        <sequence>
            <element maxOccurs="unbounded" minOccurs="0" name="submodelRef" type="aas:reference_t"/>
        </sequence>
    </complexType>
    <complexType name="submodels_t">
        <sequence>
            <element maxOccurs="unbounded" minOccurs="0" name="submodel" type="aas:submodel_t"/>
        </sequence>
    </complexType>
    <complexType name="valueDataType_t">
        <simpleContent>
            <extension base="anySimpleType"/>
```

```
        </simpleContent>
    </complexType>
    <complexType name="view_t">
        <sequence>
            <group ref="aas:referable"/>
            <group ref="aas:hasSemantics"/>
            <group ref="aas:hasDataSpecification"/>
            <element name="containedElements" type="aas:containedElements_t" ></element>
        </sequence>
    </complexType>
    <complexType name="views_t">
        <sequence>
            <element maxOccurs="unbounded" minOccurs="0" name="view" type="aas:view_t"/>
        </sequence>
    </complexType>
    <element name="aasenv" type="aas:aasenv_t"/>
    <element name="key" type="aas:key_t"/>
    <group name="hasDataSpecification">
        <sequence>
            <element maxOccurs="unbounded" minOccurs="0" name="embeddedDataSpecification"
type="aas:embeddedDataSpecification_t"/>
        </sequence>
    </group>
    <group name="hasKind">
        <sequence>
            <element maxOccurs="1" minOccurs="0" name="kind" type="aas:modelingKind_t"/>
        </sequence>
    </group>
    <group name="hasSemantics">
        <sequence>
            <element minOccurs="0" name="semanticId" type="aas:semanticId_t"/>
        </sequence>
    </group>
    <group name="identifiable">
        <sequence>
            <group ref="aas:referable"/>
            <element maxOccurs="1" minOccurs="1" name="identification" type="aas:identification_t" />
            <element maxOccurs="1" minOccurs="0" name="administration" type="aas:administration_t" />
        </sequence>
    </group>
    <group name="referable">
        <sequence>
            <element maxOccurs="1" minOccurs="1" name="idShort" type="aas:idShort_t" />
```

```
            <element maxOccurs="1" minOccurs="0" name="category" type="string"/>
            <element maxOccurs="1" minOccurs="0" name="description" type="aas:langStringSet_t"/>
            <element maxOccurs="1" minOccurs="0" name="parent" type="aas:reference_t"/>
        </sequence>
    </group>
    <group name="qualifiable">
        <sequence>
            <element maxOccurs="1" minOccurs="0" name="qualifier" type="aas:constraints_t"/>
        </sequence>
    </group>
    <simpleType name="assetKind_t">
        <restriction base="string">
            <enumeration value="Instance"/>
            <enumeration value="Template"/>
        </restriction>
    </simpleType>
    <simpleType name="entityType_t">
        <restriction base="string"/>
    </simpleType>
    <simpleType name="identifierType_t">
        <restriction base="string">
            <enumeration value="Custom"/>
            <enumeration value="IRDI"/>
            <enumeration value="IRI"/>
        </restriction>
    </simpleType>
    <simpleType name="modelingKind_t">
        <restriction base="string">
            <enumeration value="Instance"/>
            <enumeration value="Template"/>
        </restriction>
    </simpleType>
</schema>
```

iii. AAS IEC61360 Datatype

For IEC 61360, a data specification is made available, individually:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema elementFormDefault="qualified" targetNamespace="http://www.admin-shell.io/IEC61360/2/0" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:IEC61360="http://www.admin-shell.io/IEC61360/2/0">
  <attributeGroup name="keyTypes">
    <attribute name="identifierType" use="optional">
      <simpleType>
```

```
          <restriction base="string">
            <enumeration value="Custom"/>
            <enumeration value="IRDI"/>
            <enumeration value="IRI"/>
          </restriction>
        </simpleType>
      </attribute>
      <attribute name="localKeyType" use="optional">
        <simpleType>
          <restriction base="string">
            <enumeration value="idShort"/>
          </restriction>
        </simpleType>
      </attribute>
    </attributeGroup>
    <complexType name="code_t"/>
    <complexType name="dataSpecificationIEC61630_t">
      <choice minOccurs="1" maxOccurs="unbounded">
            <element maxOccurs="1" minOccurs="1" name="preferredName"
type="IEC61360:langStringSet_t" />
            <element maxOccurs="1" minOccurs="0" name="shortName"
type="IEC61360:langStringSet_t" />
            <element maxOccurs="1" minOccurs="0" name="unit" type="string" />
            <element maxOccurs="1" minOccurs="0" name="unitId"
type="IEC61360:reference_t" />
            <element maxOccurs="1" minOccurs="0" name="sourceOfDefinition"
type="string" />
            <element maxOccurs="1" minOccurs="0" name="symbol" type="string"
/>
            <element maxOccurs="1" minOccurs="0" name="dataType"
type="IEC61360:dataTypeIEC61360_t"/>
        <element maxOccurs="1" minOccurs="0" name="definition"
type="IEC61360:langStringSet_t"/>
        <element maxOccurs="1" minOccurs="0" name="valueFormat" type="string"/>
            <element maxOccurs="1" minOccurs="0" name="valueList"
type="IEC61360:valueList_t" />
            <element maxOccurs="1" minOccurs="0" name="value"
type="IEC61360:valueDataType_t" />
            <element maxOccurs="1" minOccurs="0" name="valueId"
type="IEC61360:reference_t"/>
        <element maxOccurs="unbounded" minOccurs="0" name="levelType"
type="IEC61360:levelType_t"/>
      </choice>
    </complexType>
    <complexType name="key_t">
      <simpleContent>
        <extension base="string">
          <attribute name="idType">
            <simpleType>
              <restriction base="string">
                <enumeration value="Custom"/>
                <enumeration value="FragementId"/>
                <enumeration value="idShort"/>
                <enumeration value="IRDI"/>
                <enumeration value="IRI"/>
              </restriction>
            </simpleType>
          </attribute>
          <attribute name="local" type="boolean"/>
          <attribute name="type">
            <simpleType>
              <restriction base="string">
                <enumeration value="AccessPermissionRule"/>
                <enumeration value="AnnotatedRelationshipElement"/>
```

```
                <enumeration value="Asset"/>
                <enumeration value="AssetAdministrationShell"/>
                <enumeration value="BasicEvent"/>
                <enumeration value="Blob"/>
                <enumeration value="Capability"/>
                <enumeration value="ConceptDescription"/>
                <enumeration value="ConceptDictionary"/>
                <enumeration value="DataElement"/>
                <enumeration value="Entity"/>
                <enumeration value="Event"/>
                <enumeration value="File"/>
                <enumeration value="FragmentReference"/>
                <enumeration value="GlobalReference"/>
                <enumeration value="MultiLanguageProperty"/>
                <enumeration value="Operation"/>
                <enumeration value="Property"/>
                <enumeration value="Range"/>
                <enumeration value="ReferenceElement"/>
                <enumeration value="RelationshipElement"/>
                <enumeration value="Submodel"/>
                <enumeration value="SubmodelElement"/>
                <enumeration value="SubmodelElementCollection"/>
                <enumeration value="View"/>
              </restriction>
            </simpleType>
          </attribute>
        </extension>
      </simpleContent>
    </complexType>
    <complexType name="keys_t">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="0" ref="IEC61360:key"/>
      </sequence>
    </complexType>
    <complexType name="langString_t">
      <simpleContent>
        <extension base="string">
          <attribute name="lang" type="string"/>
        </extension>
      </simpleContent>
    </complexType>
    <complexType name="langStringSet_t">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="1" name="langString"
type="IEC61360:langString_t"/>
      </sequence>
    </complexType>
    <complexType name="reference_t">
      <sequence>
        <element name="keys" type="IEC61360:keys_t"/>
      </sequence>
    </complexType>
    <complexType name="valueDataType_t"/>
    <complexType name="valueList_t">
      <sequence>
        <element maxOccurs="unbounded" minOccurs="1" name="valueReferencePair"
type="IEC61360:valueReferencePair_t"/>
      </sequence>
    </complexType>
    <complexType name="valueReferencePair_t">
      <sequence>
        <element name="valueId" type="IEC61360:reference_t"/>
        <element name="value" type="IEC61360:valueDataType_t"/>
      </sequence>
```

```
  </complexType>
  <element name="key" type="IEC61360:key_t"/>
  <simpleType name="dataTypeIEC61360_t">
    <restriction base="string">
      <enumeration value="BOOLEAN"/>
      <enumeration value="DATE"/>
      <enumeration value="RATIONAL"/>
      <enumeration value="RATIONAL_MEASURE"/>
      <enumeration value="REAL_COUNT"/>
      <enumeration value="REAL_CURRENCY"/>
      <enumeration value="REAL_MEASURE"/>
      <enumeration value="STRING"/>
      <enumeration value="STRING_TRANSLATABLE"/>
      <enumeration value="TIME"/>
      <enumeration value="TIME_STAMP"/>
      <enumeration value="URL"/>
    </restriction>
  </simpleType>
  <simpleType name="levelType_t">
    <restriction base="string">
      <enumeration value="Max"/>
      <enumeration value="Min"/>
      <enumeration value="Nom"/>
      <enumeration value="Typ"/>
    </restriction>
  </simpleType>
</schema>
```

iv. AAS Attribute Based Access Control Model

For Asset based Access Control a specification is made available individually:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.admin-shell.io/aas/abac/2/0"
  elementFormDefault="qualified"
  xmlns:aas="http://www.admin-shell.io/aas/2/0"
  xmlns:abac="http://www.admin-shell.io/aas/abac/2/0">

  <import schemaLocation="AAS.xsd" namespace="http://www.admin-
shell.io/aas/2/0"></import>

  <complexType name="security_t">
    <sequence>
      <element name="accessControlPolicyPoints"
        type="abac:accessControlPolicyPoints_t" minOccurs="1"
        maxOccurs="1">
      </element>
      <element name="certificates" type="abac:certificates_t"
        minOccurs="0" maxOccurs="1">
      </element>
      <element name="requiredCertificateExtensions" minOccurs="0"
maxOccurs="1" type="aas:references_t"></element>
    </sequence>
  </complexType>


  <complexType name="accessControlPolicyPoints_t">
    <sequence>
      <element name="policyAdministrationPoint" minOccurs="1" maxOccurs="1"
type="abac:policyAdministrationPoint_t"></element>
```

```
        <element name="policyDecisionPoint" type="abac:policyDecisionPoint_t"
minOccurs="1" maxOccurs="1"></element>
        <element name="policyEnforcementPoint"
type="abac:policyEnforcementPoint_t" minOccurs="1" maxOccurs="1"></element>
        <element name="policyInformationPoints"
type="abac:policyInformationPoints_t" minOccurs="0" maxOccurs="1"></element>
      </sequence>
    </complexType>

    <complexType name="policyAdministrationPoint_t">
     <sequence>
        <element name="localAccessControl"
          type="abac:accessControl_t" minOccurs="0" maxOccurs="1">
        </element>
        <element name="externalAccessControl" type="boolean" minOccurs="0"
maxOccurs="1"></element>
      </sequence>
    </complexType>

    <complexType name="policyDecisionPoint_t">
     <sequence>
        <element name="externalPolicyDecisionPoint" type="boolean"
minOccurs="1" maxOccurs="1"></element>
      </sequence>
    </complexType>

    <complexType name="policyEnforcementPoint_t">
     <sequence>
        <element name="externalPolicyEnforcementPoint" type="boolean"
minOccurs="1" maxOccurs="1"></element>
      </sequence>
    </complexType>

    <complexType name="policyInformationPoints_t">
     <sequence>
        <element name="externalInformationPoints"
          type="boolean" minOccurs="1" maxOccurs="1">
        </element>
        <element name="internalInformationPoints"
type="abac:internalInformationPoints" minOccurs="0" maxOccurs="1"></element>
      </sequence>
    </complexType>

   <complexType name="internalInformationPoints">
      <sequence>
        <element name="internalInformationPoint" type="aas:submodelRefs_t"
minOccurs="0" maxOccurs="unbounded"></element>
      </sequence>
   </complexType>

    <complexType name="accessControl_t">
     <sequence>
            <element name="selectableSubjectAttributes" type="aas:reference_t"
minOccurs="0" maxOccurs="1"></element>
            <element name="defaultSubjectAttributes" type="aas:reference_t"
minOccurs="1" maxOccurs="1"></element>
            <element name="selectablePermissions" type="aas:reference_t"
minOccurs="0" maxOccurs="1"></element>
            <element name="defaultPermissions" type="aas:reference_t"
minOccurs="1" maxOccurs="1"> </element>
            <element name="selectableEnvironmentAttributes"
type="aas:reference_t" minOccurs="0" maxOccurs="1"></element>
            <element name="defaultEnvironmentAttributes"
type="aas:reference_t" minOccurs="0" maxOccurs="1"></element>
```

```
            <element name="accessPermissionRules"
type="abac:accessPermissionRules_t" minOccurs="0" maxOccurs="1"></element>
     </sequence>
    </complexType>

    <complexType name="permissionPerObject_t">
     <sequence>
       <element name="object" type="aas:reference_t"></element>
       <element name="targetObjectAttributes" type="abac:objectAttributes_t"
minOccurs="0" maxOccurs="1"></element>
       <element name="permissions" type="abac:permissions_t" minOccurs="0"
maxOccurs="1"></element>
     </sequence>
    </complexType>

    <complexType name="objectAttributes_t">
     <sequence>
       <element name="objectAttribute" type="aas:property_t" minOccurs="1"
maxOccurs="unbounded"></element>
     </sequence>
    </complexType>

    <complexType name="permissions_t">
     <sequence>
       <element name="permission" type="aas:property_t" minOccurs="1"
maxOccurs="1"></element>
       <element name="kindOfPermission" type="abac:permissionKind"></element>
     </sequence>
    </complexType>

    <simpleType name="permissionKind">
     <restriction base="string">
       <enumeration value="Allow"></enumeration>
       <enumeration value="Deny"></enumeration>
       <enumeration value="NotApplicable"></enumeration>
       <enumeration value="Undefined"></enumeration>
     </restriction>
    </simpleType>

    <complexType name="accessPermissionRule_t">
     <sequence>
       <group ref="aas:qualifiable"></group>
       <group ref="aas:referable"></group>
       <element name="targetSubjectAttributes" type="abac:subjectAttributes_t"
minOccurs="1" maxOccurs="unbounded"></element>
       <element name="permissionsPerObject" type="abac:permissionPerObject_t"
minOccurs="0" maxOccurs="unbounded"></element>
     </sequence>
    </complexType>

    <complexType name="subjectAttributes_t">
     <sequence>
       <element name="subjectAttribute" type="aas:property_t" minOccurs="1"
maxOccurs="unbounded"></element>
     </sequence>
    </complexType>

    <complexType name="certificates_t">
     <sequence>
       <element name="certificate" type="abac:certificate_t" minOccurs="1"
maxOccurs="unbounded"></element>
     </sequence>
    </complexType>
```

```
   <complexType name="certificate_t">
     <choice>
       <element name="blobCertificate"
type="abac:blobCertificate_t"></element>
     </choice>
   </complexType>


    <element name="internalInformationPoints" type="aas:submodelRefs_t" />


    <complexType name="accessPermissionRules_t">
     <sequence>
       <element name="accessPermissionRule" type="abac:accessPermissionRule_t"
minOccurs="0" maxOccurs="unbounded"></element>
     </sequence>
    </complexType>


   <complexType name="certificateAbstract_t">
   </complexType>

   <complexType name="blobCertificate_t">
     <complexContent>
       <extension base="abac:certificateAbstract_t">
         <sequence>
           <element name="blobCertificate" type="aas:blob_t"></element>
           <element name="containedExtensions"
             type="abac:containedExtensions_t" minOccurs="0" maxOccurs="1">
           </element>
           <element name="lastCertificate" type="boolean"></element>
         </sequence>
       </extension>
     </complexContent>
   </complexType>

   <complexType name="containedExtensions_t">
     <sequence>
       <element name="containedExtension" type="aas:reference_t" minOccurs="0"
maxOccurs="unbounded"></element>
     </sequence>
   </complexType>
</schema>
```

→

v. XML Example

For cross reference, a complete self-contained example is given, which relates to the unified example in Clause 5.2.1.

```
<?xml version="1.0" encoding="utf-8"?>
<aas:aasenv xmlns:aas="http://www.admin-shell.io/aas/2/0"
xmlns:IEC61360="http://www.admin-shell.io/IEC61360/2/0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.admin-shell.io/aas/2/0 AAS.xsd
http://www.admin-shell.io/IEC61360/2/0 IEC61360.xsd">
   <aas:assetAdministrationShells>
     <aas:assetAdministrationShell>
       <aas:idShort>
         ExampleMotor
       </aas:idShort>
       <aas:category>CONSTANT</aas:category>
```

```
        <aas:identification idType="IRI">
          http://customer.com/aas/9175_7013_7091_9168
        </aas:identification>
        <aas:assetRef>
          <aas:keys>
            <aas:key idType="IRI" local="true" type="Asset">
              http://customer.com/assets/KHBVZJSQKIY
            </aas:key>
          </aas:keys>
        </aas:assetRef>

        <aas:submodelRefs>
          <aas:submodelRef>
            <aas:keys>
              <aas:key idType="IRI" local="true" type="Submodel">
                http://i40.customer.com/instance/1/1/AC69B1CB44F07935
              </aas:key>
            </aas:keys>
          </aas:submodelRef>
          <aas:submodelRef>
            <aas:keys>
              <aas:key idType="IRI" local="true" type="Submodel">
                http://i40.customer.com/type/1/1/1A7B62B529F19152
              </aas:key>
            </aas:keys>
          </aas:submodelRef>
          <aas:submodelRef>
            <aas:keys>
              <aas:key idType="IRI" local="true" type="Submodel">
                http://i40.customer.com/type/1/1/7A7104BDAB57E184
              </aas:key>
            </aas:keys>
          </aas:submodelRef>
        </aas:submodelRefs>
      </aas:assetAdministrationShell>
    </aas:assetAdministrationShells>
    <aas:assets>
      <aas:asset>
        <aas:idShort>
          ServoDCMotor
        </aas:idShort>
        <aas:identification idType="IRI">
          http://customer.com/assets/KHBVZJSQKIY
        </aas:identification>
        <aas:assetIdentificationModelRef>
          <aas:keys/>
        </aas:assetIdentificationModelRef>
        <aas:billOfMaterialRef>
          <aas:keys/>
        </aas:billOfMaterialRef>
        <aas:kind>Instance</aas:kind>
      </aas:asset>
    </aas:assets>
      <aas:submodels>
      <aas:submodel>
        <aas:idShort>
          Documentation
        </aas:idShort>
        <aas:category>
          CONSTANT
        </aas:category>
        <aas:identification idType="IRI">
          http://i40.customer.com/type/1/1/1A7B62B529F19152
        </aas:identification>
```

```
        <aas:kind>Instance</aas:kind>
        <aas:qualifier/>
        <aas:submodelElements>
          <aas:submodelElement>
            <aas:submodelElementCollection>
              <aas:idShort>
                OperatingManual
              </aas:idShort>
              <aas:kind>Instance</aas:kind>
              <aas:semanticId>
                <aas:keys>
                  <aas:key idType="IRI" local="true"
type="ConceptDescription">
                    www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Document
                  </aas:key>
                </aas:keys>
              </aas:semanticId>
              <aas:allowDuplicates>false</aas:allowDuplicates>
              <aas:ordered>false</aas:ordered>
              <aas:value>
                <aas:submodelElement>
                  <aas:file>
                    <aas:idShort>
                      DigitalFile_PDF
                    </aas:idShort>
                    <aas:category>
                      PARAMETER
                    </aas:category>
                    <aas:kind>Instance</aas:kind>
                    <aas:semanticId>
                      <aas:keys>
                        <aas:key idType="IRI" local="true"
type="ConceptDescription">

  http://vdi2770.com/blatt1/Entwurf/Okt18/cd/StoredDocumentRepresentation/Digi
talFile
                        </aas:key>
                      </aas:keys>
                    </aas:semanticId>
                    <aas:mimeType>
                      application/pdf
                    </aas:mimeType>
                    <aas:value>
                      /aasx/OperatingManual.pdf
                    </aas:value>
                  </aas:file>
                </aas:submodelElement>
                <aas:submodelElement>
                  <aas:multiLanguageProperty>
                    <aas:idShort>
                      Title
                    </aas:idShort>
                    <aas:kind>Instance</aas:kind>
                    <aas:semanticId>
                      <aas:keys>
                        <aas:key idType="IRI" local="true"
type="ConceptDescription">

  http://vdi2770.com/blatt1/Entwurf/Okt18/cd/Description/Title
                        </aas:key>
                      </aas:keys>
                    </aas:semanticId>
                    <aas:value>
                      <aas:langString lang="EN">
```

```
                        Operating Manual
                      </aas:langString>
                    </aas:value>
                  </aas:multiLanguageProperty>
                </aas:submodelElement>
              </aas:value>
            </aas:submodelElementCollection>
          </aas:submodelElement>
        </aas:submodelElements>
      </aas:submodel>
      <aas:submodel>
        <aas:idShort>
          TechnicalData
        </aas:idShort>
        <aas:category>
          CONSTANT
        </aas:category>
        <aas:identification idType="IRI">
          http://i40.customer.com/type/1/1/7A7104BDAB57E184
        </aas:identification>
        <aas:kind>Instance</aas:kind>
        <aas:semanticId>
          <aas:keys>
            <aas:key idType="IRDI" local="false" type="GlobalReference">
              0173-1#01-AFZ615#016
            </aas:key>
          </aas:keys>
        </aas:semanticId>
        <aas:submodelElements>
          <aas:submodelElement>
            <aas:property>
              <aas:idShort>
                MaxRotationSpeed
              </aas:idShort>
              <aas:category>
                PARAMETER
              </aas:category>
              <aas:kind>Instance</aas:kind>
              <aas:semanticId>
                <aas:keys>
                  <aas:key idType="IRDI" local="true"
type="ConceptDescription">
                    0173-1#02-BAA120#008
                  </aas:key>
                </aas:keys>
              </aas:semanticId>
              <aas:value>
                5000
              </aas:value>
              <aas:valueType>
                integer
              </aas:valueType>
            </aas:property>
          </aas:submodelElement>
        </aas:submodelElements>
      </aas:submodel>
      <aas:submodel>
        <aas:idShort>
          OperationalData
        </aas:idShort>
        <aas:category>
          VARIABLE
        </aas:category>
        <aas:identification idType="IRI">
```

```
          http://i40.customer.com/instance/1/1/AC69B1CB44F07935
        </aas:identification>

        <aas:kind>Instance</aas:kind>
        <aas:semanticId>
          <aas:keys/>
        </aas:semanticId>
        <aas:submodelElements>
          <aas:submodelElement>
            <aas:property>
              <aas:idShort>
                RotationSpeed
              </aas:idShort>
              <aas:category>
                VARIABLE
              </aas:category>
              <aas:kind>Instance</aas:kind>
              <aas:semanticId>
                <aas:keys>
                  <aas:key idType="IRI" local="true"
type="ConceptDescription">
                    http://customer.com/cd/1/1/18EBD56F6B43D895
                  </aas:key>
                </aas:keys>
              </aas:semanticId>
              <aas:embeddedDataSpecification>
                <aas:dataSpecification>
                  <aas:keys/>
                </aas:dataSpecification>
              </aas:embeddedDataSpecification>
              <aas:value>
                4370
              </aas:value>
              <aas:valueType>
                integer
              </aas:valueType>
            </aas:property>
          </aas:submodelElement>
        </aas:submodelElements>
      </aas:submodel>
    </aas:submodels>
      <aas:conceptDescriptions>
      <aas:conceptDescription>
        <aas:idShort>
          MaxRotationSpeed
        </aas:idShort>
        <aas:category>PROPERTY</aas:category>
        <aas:identification idType="IRDI">
          0173-1#02-BAA120#008
        </aas:identification>
        <aas:administration>
          <aas:version>2</aas:version>
          <aas:revision>0</aas:revision>
        </aas:administration>
        <aas:embeddedDataSpecification>
          <aas:dataSpecificationContent>
            <aas:dataSpecificationIEC61360>
              <IEC61360:dataType>REAL_MEASURE</IEC61360:dataType>
              <IEC61360:definition>
                <IEC61360:langString lang="de">
                  Höchste zulässige Drehzahl, mit welcher der Motor oder die
Speiseinheit betrieben werden darf
                </IEC61360:langString>
                <IEC61360:langString lang="en">
```

```
                    Greatest permissible rotation speed with which the motor or
feeding unit may be operated
                  </IEC61360:langString>
                </IEC61360:definition>
                <IEC61360:preferredName>
                  <IEC61360:langString lang="de">
                    max. Drehzahl
                  </IEC61360:langString>
                  <IEC61360:langString lang="en">
                    Max. rotation speed
                  </IEC61360:langString>
                </IEC61360:preferredName>
                <IEC61360:shortName>
                  <IEC61360:langString></IEC61360:langString>
                </IEC61360:shortName>
                <IEC61360:sourceOfDefinition/>
                <IEC61360:unit>
                  1/min
                </IEC61360:unit>
                <IEC61360:unitId>
                  <IEC61360:keys>
                    <IEC61360:key idType="IRDI" local="false"
type="GlobalReference">
                      0173-1#05-AAA650#002
                    </IEC61360:key>
                  </IEC61360:keys>
                </IEC61360:unitId>
              </aas:dataSpecificationIEC61360>
            </aas:dataSpecificationContent>
          </aas:embeddedDataSpecification>
          <aas:isCaseOf>
            <aas:keys/>
          </aas:isCaseOf>
        </aas:conceptDescription>
        <aas:conceptDescription>
          <aas:idShort>
            Title
          </aas:idShort>
          <aas:category>PROPERTY</aas:category>
          <aas:identification idType="IRI">
            http://vdi2770.com/blatt1/Entwurf/Okt18/cd/Description/Title
          </aas:identification>
          <aas:embeddedDataSpecification>
            <aas:dataSpecificationContent>
              <aas:dataSpecificationIEC61360>
                <IEC61360:dataType>STRING_TRANSLATABLE</IEC61360:dataType>
                <IEC61360:definition>
                  <IEC61360:langString lang="DE">
                    Sprachabhängiger Titel des Dokuments.
                  </IEC61360:langString>
                </IEC61360:definition>
                <IEC61360:preferredName>
                  <IEC61360:langString lang="DE">
                    Titel
                  </IEC61360:langString>
                  <IEC61360:langString lang="EN">
                    Title
                  </IEC61360:langString>
                </IEC61360:preferredName>
                <IEC61360:shortName>
                  <IEC61360:langString lang="DE">
                    Titel
                  </IEC61360:langString>
                </IEC61360:shortName>
```

```
              <IEC61360:sourceOfDefinition/>
              <IEC61360:unit/>
            </aas:dataSpecificationIEC61360>
          </aas:dataSpecificationContent>
          <aas:dataSpecification>
            <aas:keys>
              <aas:key idType="IRI" local="false" type="GlobalReference">
                http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360
              </aas:key>
            </aas:keys>
          </aas:dataSpecification>
        </aas:embeddedDataSpecification>
      </aas:conceptDescription>
      <aas:conceptDescription>
        <aas:idShort>
          RotationSpeed
        </aas:idShort>
        <aas:category>PROPERTY</aas:category>
        <aas:identification idType="IRI">
          http://customer.com/cd/1/1/18EBD56F6B43D895
        </aas:identification>

        <aas:embeddedDataSpecification>
          <aas:dataSpecificationContent>
            <aas:dataSpecificationIEC61360>
              <IEC61360:dataType>REAL_MEASURE</IEC61360:dataType>
              <IEC61360:definition>
                <IEC61360:langString lang="DE">
                  Aktuelle Drehzahl, mit welcher der Motor oder die
Speiseinheit betrieben wird
                </IEC61360:langString>
                <IEC61360:langString lang="EN">
                  Actual rotation speed with which the motor or feeding unit
is operated
                </IEC61360:langString>
              </IEC61360:definition>
              <IEC61360:preferredName>
                <IEC61360:langString lang="DE">
                  Aktuelle Drehzahl
                </IEC61360:langString>
                <IEC61360:langString lang="EN">
                  Actual rotation speed
                </IEC61360:langString>
              </IEC61360:preferredName>
              <IEC61360:shortName>
                <IEC61360:langString lang="EN">
                  RotationSpeed
                </IEC61360:langString>
              </IEC61360:shortName>
              <IEC61360:sourceOfDefinition/>
              <IEC61360:unit>
                1/min
              </IEC61360:unit>
              <IEC61360:unitId>
                <IEC61360:keys>
                  <IEC61360:key idType="IRDI" local="false"
type="GlobalReference">
                    0173-1#05-AAA650#002
                  </IEC61360:key>
                </IEC61360:keys>
              </IEC61360:unitId>
            </aas:dataSpecificationIEC61360>
          </aas:dataSpecificationContent>
```

```
          <aas:dataSpecification>
            <aas:keys>
              <aas:key idType="IRI" local="false" type="GlobalReference">
                http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360
              </aas:key>
            </aas:keys>
          </aas:dataSpecification>
        </aas:embeddedDataSpecification>
      </aas:conceptDescription>
      <aas:conceptDescription>
        <aas:idShort>
          Document
        </aas:idShort>
        <aas:category>
          COLLECTION
        </aas:category>
        <aas:description>
          <aas:langString/>
        </aas:description>
        <aas:identification idType="IRI">
          http://vdi2770.com/blatt1/Entwurf/Okt18/cd/Document
        </aas:identification>
        <aas:embeddedDataSpecification>
          <aas:dataSpecificationContent>
            <aas:dataSpecificationIEC61360>
              <IEC61360:dataType>URL</IEC61360:dataType>
              <IEC61360:definition>
                <IEC61360:langString lang="DE">
                  Feste und geordnete Menge von für die Verwendung durch
Personen bestimmte Informationen, die verwaltet und als Einheit zwischen
Benutzern und System ausgetauscht werden kann.
                </IEC61360:langString>
              </IEC61360:definition>
              <IEC61360:preferredName>
                <IEC61360:langString lang="DE">
                  Dokument
                </IEC61360:langString>
                <IEC61360:langString lang="EN">
                  Document
                </IEC61360:langString>
              </IEC61360:preferredName>
              <IEC61360:shortName>
                <IEC61360:langString lang="EN">
                  Document
                </IEC61360:langString>
              </IEC61360:shortName>
              <IEC61360:sourceOfDefinition>
                [ISO 15519-1:2010]
              </IEC61360:sourceOfDefinition>
              <IEC61360:unit/>
            </aas:dataSpecificationIEC61360>
          </aas:dataSpecificationContent>
          <aas:dataSpecification>
            <aas:keys>
              <aas:key idType="IRI" local="false" type="GlobalReference">
                http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360
              </aas:key>
            </aas:keys>
          </aas:dataSpecification>
        </aas:embeddedDataSpecification>

      </aas:conceptDescription>
```

```
    <aas:conceptDescription>
      <aas:idShort>
        DigitalFile
      </aas:idShort>
      <aas:category>
        DOCUMENT
      </aas:category>
      <aas:identification idType="IRI">

  http://vdi2770.com/blatt1/Entwurf/Okt18/cd/StoredDocumentRepresentation/Digi
talFile
      </aas:identification>
      <aas:embeddedDataSpecification>
        <aas:dataSpecificationContent>
          <aas:dataSpecificationIEC61360>
            <IEC61360:dataType>URL</IEC61360:dataType>
            <IEC61360:definition>
              <IEC61360:langString lang="DE">
                Eine Datei, die die DocumentVersion repräsentiert. Neben
der obligatorischen PDF/A Datei können weitere Dateien angegeben werden.
              </IEC61360:langString>
            </IEC61360:definition>
            <IEC61360:preferredName>
              <IEC61360:langString lang="DE">
                Digitale Datei
              </IEC61360:langString>
            </IEC61360:preferredName>
            <IEC61360:shortName>
              <IEC61360:langString lang="DE">
                digitale Datei
              </IEC61360:langString>
            </IEC61360:shortName>
            <IEC61360:sourceOfDefinition/>
            <IEC61360:unit/>
          </aas:dataSpecificationIEC61360>
        </aas:dataSpecificationContent>
        <aas:dataSpecification>
          <aas:keys>
            <aas:key idType="IRI" local="false" type="GlobalReference">
              http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360
            </aas:key>
          </aas:keys>
        </aas:dataSpecification>
      </aas:embeddedDataSpecification>
    </aas:conceptDescription>
  </aas:conceptDescriptions>
</aas:aasenv>
```

Note: ↙ designates line-wrap for purpose of layout

Annex F. JSON schema and complete example

i. JSON Schema for the Asset Administration Shell Environment

The following schema uses JSON Schema[44] in version 2019-09 to allow validation of JSON files.

Table 29 JSON schema

```
{
  "$schema": "https://json-schema.org/draft/2019-09/schema",
  "title": "AssetAdministrationShellEnvironment",
  "type": "object",
  "$id": "urn://www.admin-shell.io/schema/json/v2.0",
  "required": ["assetAdministrationShells",
  "submodels",
  "assets",
  "conceptDescriptions"],
  "properties": {
    "assetAdministrationShells": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/AssetAdministrationShell"
      }
    },
    "submodels": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/Submodel"
      }
    },
    "assets": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/Asset"
      }
    },
    "conceptDescriptions": {
    },
    "required": ["value"]
  },
  "SubmodelElement": {
    "allOf": [{
      "$ref": "#/definitions/Referable"
    },
    {
      "$ref": "#/definitions/HasDataSpecification"
    },
    {
      "$ref": "#/definitions/HasSemantics"
    },
    {
      "$ref": "#/definitions/Qualifiable"
    },
    {
      "properties": {
        "kind": {
          "$ref": "#/definitions/ModelingKind"
        }
      }
    }]
  },
  "Event": {
    "allOf": [{
      "$ref": "#/definitions/SubmodelElement"
    }]
  },
  "BasicEvent": {
```

[44] see: http://json-schema.org/

```
        "type": "array",
        "items": {
          "$ref": "#/definitions/ConceptDescription"
        }
      }
    },
    "definitions": {
      "Referable": {
        "type": "object",
        "properties": {
          "idShort": {
            "type": "string"
          },
          "category": {
            "type": "string"
          },
          "description": {
            "type": "array",
            "items": {
              "$ref": "#/definitions/LangString"
            }
          },
          "parent": {
            "$ref": "#/definitions/Reference"
          },
          "modelType": {
            "$ref": "#/definitions/ModelType"
          }
        },
        "required": ["idShort",
        "modelType"]
      },
      "Identifiable": {
        "allOf": [{
          "$ref": "#/definitions/Referable"
        },
        {
```

```
        "allOf": [{
          "$ref": "#/definitions/Event"
        },
        {
          "properties": {
            "observed": {
              "$ref": "#/definitions/Reference"
            }
          },
          "required": ["observed"]
        }]
      },
      "EntityType": {
        "type": "string",
        "enum": ["CoManagedEntity",
        "SelfManagedEntity"]
      },
      "Entity": {
        "allOf": [{
          "$ref": "#/definitions/SubmodelElement"
        },
        {
          "properties": {
            "statements": {
              "type": "array",
              "items": {
                "$ref":
"#/definitions/SubmodelElement"
              }
            },
            "entityType": {
              "$ref": "#/definitions/EntityType"
            },
            "asset": {
              "$ref": "#/definitions/Reference"
            },


          },
```

```
            "properties": {
              "identification": {
                "$ref": "#/definitions/Identifier"
              },
              "administration": {
                "$ref":
"#/definitions/AdministrativeInformation"
              },

            },
            "required": ["identification"]
          }]
        },
        "Qualifiable": {
          "type": "object",
          "properties": {
            "qualifiers": {
              "type": "array",
              "items": {
                "$ref": "#/definitions/Constraint"
              }
            },

          }
        },
        "HasSemantics": {
          "type": "object",
          "properties": {
            "semanticId": {
              "$ref": "#/definitions/Reference"
            }
          },
          "required": ["semanticId"]
        },
        "HasDataSpecification": {
          "type": "object",
          "properties": {
            "embeddedDataSpecifications": {
              "required": ["entityType"]
            }]
          },
          "View": {
            "allOf": [{
              "$ref": "#/definitions/Referable"
            },
            {
              "$ref": "#/definitions/HasDataSpecification"
            },
            {
              "$ref": "#/definitions/HasSemantics"
            },
            {
              "properties": {
                "containedElements": {
                  "type": "array",
                  "items": {
                    "$ref": "#/definitions/Reference"
                  }
                }
              }
            }]
          },
          "ConceptDictionary": {
            "allOf": [{
              "$ref": "#/definitions/Referable"
            },
            {
              "$ref": "#/definitions/HasDataSpecification"
            },
            {
              "properties": {
                "conceptDescriptions": {
                  "type": "array",
                  "items": {
                    "$ref": "#/definitions/Reference"
```

```
            "type": "array",
            "items": {
              "$ref":
"#/definitions/EmbeddedDataSpecification"
            }
          }
        }
      },
      "AssetAdministrationShell": {
        "allOf": [{
          "$ref": "#/definitions/Identifiable"
        },
        {
          "$ref": "#/definitions/HasDataSpecification"
        },
        {
          "properties": {
            "derivedFrom": {
              "$ref": "#/definitions/Reference"
            },
            "asset": {
              "$ref": "#/definitions/Reference"
            },
            "submodels": {
              "type": "array",
              "items": {
                "$ref": "#/definitions/Reference"
              }
            },
            "views": {
              "type": "array",
              "items": {
                "$ref": "#/definitions/View"
              }
            },
            "conceptDictionaries": {
              "type": "array",
              "items": {
            }
          }
        }
      }]
    },
    "ConceptDescription": {
      "allOf": [{
        "$ref": "#/definitions/Identifiable"
      },
      {
        "$ref": "#/definitions/HasDataSpecification"
      },
      {
        "properties": {
          "isCaseOf": {
            "type": "array",
            "items": {
              "$ref": "#/definitions/Reference"
            }
          }
        }
      }]
    },
    "Capability": {
      "allOf": [{
        "$ref": "#/definitions/SubmodelElement"
      }]
    },
    "Property": {
      "allOf": [{
        "$ref": "#/definitions/SubmodelElement"
      },
      {
        "$ref": "#/definitions/ValueObject"
      }]
    },
    "Range": {
```

```
                    "$ref":
"#/definitions/ConceptDictionary"
                  }
                },
                "security": {
                  "$ref": "#/definitions/Security"
                }
              },
              "required": ["asset"]
            }]
        },
        "Identifier": {
          "type": "object",
          "properties": {
            "id": {
              "type": "string"
            },
            "idType": {
              "$ref": "#/definitions/KeyType"
            }
          },
          "required": ["id",
          "idType"]
        },
        "KeyType": {
          "type": "string",
          "enum": ["Custom",
          "IRDI",
          "IRI",
          "IdShort",
          "FragmentId"]
        },
        "AdministrativeInformation": {
          "type": "object",
          "properties": {
            "version": {
              "type": "string"
            },
```

```
        "allOf": [{
          "$ref": "#/definitions/SubmodelElement"
        },
        {
          "properties": {
            "valueType": {
              "type": "string",
              "enum": ["anyUri",
              "base64Binary",
              "boolean",
              "date",
              "dateTime",
              "dateTimeStamp",
              "decimal",
              "integer",
              "long",
              "int",
              "short",
              "byte",
              "nonNegativeInteger",
              "positiveInteger",
              "unsignedLong",
              "unsignedInt",
              "unsignedShort",
              "unsignedByte",
              "nonPositiveInteger",
              "negativeInteger",
              "double",
              "duration",
              "dayTimeDuration",
              "yearMonthDuration",
              "float",
              "gDay",
              "gMonth",
              "gMonthDay",
              "gYear",
              "gYearMonth",
```

```
        "revision": {
          "type": "string"
        }
      }
    },
    "LangString": {
      "type": "object",
      "properties": {
        "language": {
          "type": "string"
        },
        "text": {
          "type": "string"
        }
      },
      "required": ["language",
      "text"]
    },
    "Reference": {
      "type": "object",
      "properties": {
        "keys": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/Key"
          }
        }
      },
      "required": ["keys"]
    },
    "Key": {
      "type": "object",
      "properties": {
        "type": {
          "$ref": "#/definitions/KeyElements"
        },
        "idType": {
          "hexBinary",
          "NOTATION",
          "QName",
          "string",
          "normalizedString",
          "token",
          "language",
          "Name",
          "NCName",
          "ENTITY",
          "ID",
          "IDREF",
          "NMTOKEN",
          "time"]
        },
        "min": {
          "type": "string"
        },
        "max": {
          "type": "string"
        }
      },
      "required": ["valueType",
      "min",
      "max"]
    }]
  },
  "MultiLanguageProperty": {
    "allOf": [{
      "$ref": "#/definitions/SubmodelElement"
    },
    {
      "properties": {
        "value": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/LangString"
```

```
            "$ref": "#/definitions/KeyType"
          },
          "value": {
            "type": "string"
          },
          "local": {
            "type": "boolean"
          }
        },
        "required": ["type",
        "idType",
        "value",
        "local"]
      },
      "KeyElements": {
        "type": "string",
        "enum": ["Asset",
        "AssetAdministrationShell",
        "ConceptDescription",
        "Submodel",
        "AccessPermissionRule",
        "AnnotatedRelationshipElement",
        "BasicEvent",
        "Blob",
        "Capability",
        "ConceptDictionary",
        "DataElement",
        "File",
        "Entity",
        "Event",
        "MultiLanguageProperty",
        "Operation",
        "Property",
        "Range",
        "ReferenceElement",
        "RelationshipElement",
        "SubmodelElement",
```

```
            }
          },
          "valueId": {
            "$ref": "#/definitions/Reference"
          }
        }
      }]
    },
    "File": {
      "allOf": [{
        "$ref": "#/definitions/SubmodelElement"
      },
      {
        "properties": {
          "value": {
            "type": "string"
          },
          "mimeType": {
            "type": "string"
          }
        },
        "required": ["mimeType",
        "value"]
      }]
    },
    "Blob": {
      "allOf": [{
        "$ref": "#/definitions/SubmodelElement"
      },
      {
        "properties": {
          "value": {
            "type": "string"
          },
          "mimeType": {
            "type": "string"
          }
```

```
        "SubmodelElementCollection",
        "View",
        "GlobalReference",
        "FragmentReference"]
    },
    "ModelTypes": {
        "type": "string",
        "enum": ["Asset",
        "AssetAdministrationShell",
        "ConceptDescription",
        "Submodel",
        "AccessPermissionRule",
        "AnnotatedRelationshipElement",
        "BasicEvent",
        "Blob",
        "Capability",
        "ConceptDictionary",
        "DataElement",
        "File",
        "Entity",
        "Event",
        "MultiLanguageProperty",
        "Operation",
        "Property",
        "Range",
        "ReferenceElement",
        "RelationshipElement",
        "SubmodelElement",
        "SubmodelElementCollection",
        "View",
        "GlobalReference",
        "FragmentReference",
        "Constraint",
        "Formula",
        "Qualifier"]
    },
    "ModelType": {
        },
        "required": ["mimeType",
        "value"]
    }]
    },
    "ReferenceElement": {
    "allOf": [{
        "$ref": "#/definitions/SubmodelElement"
    },
    {
        "properties": {
        "value": {
            "$ref": "#/definitions/Reference"
        }
        }
    }]
    },
    "SubmodelElementCollection": {
    "allOf": [{
        "$ref": "#/definitions/SubmodelElement"
    },
    {
        "properties": {
        "value": {
            "type": "array",
            "items": {
            "oneOf": [{
                "$ref": "#/definitions/Blob"
            },
            {
                "$ref": "#/definitions/File"
            },
            {
                "$ref": "#/definitions/Capability"
            },
            {
                "$ref": "#/definitions/Entity"
```

```
        "type": "object",
        "properties": {
          "name": {
            "$ref": "#/definitions/ModelTypes"
          }
        },
        "required": ["name"]
      },
      "EmbeddedDataSpecification": {
        "type": "object",
        "properties": {
          "dataSpecification": {
            "$ref": "#/definitions/Reference"
          },
          "dataSpecificationContent": {
            "$ref":
"#/definitions/DataSpecificationContent"
          }
        },
        "required": ["dataSpecification",
        "dataSpecificationContent"]
      },
      "DataSpecificationContent": {
        "oneOf": [{
          "$ref":
"#/definitions/DataSpecificationIEC61360Content"
        },
        {
          "$ref":
"#/definitions/DataSpecificationPhysicalUnitContent"
        }]
      },
      "DataSpecificationPhysicalUnitContent": {
        "type": "object",
        "properties": {
          "unitName": {
            "type": "string"
          },
                    },
                    {
                      "$ref": "#/definitions/Event"
                    },
                    {
                      "$ref": "#/definitions/BasicEvent"
                    },
                    {
                      "$ref":
"#/definitions/MultiLanguageProperty"
                    },
                    {
                      "$ref": "#/definitions/Operation"
                    },
                    {
                      "$ref": "#/definitions/Property"
                    },
                    {
                      "$ref": "#/definitions/Range"
                    },
                    {
                      "$ref":
"#/definitions/ReferenceElement"
                    },
                    {
                      "$ref":
"#/definitions/RelationshipElement"
                    },
                    {
                      "$ref":
"#/definitions/SubmodelElementCollection"
                    }]
                  }
                },
                "allowDuplicates": {
                  "type": "boolean"
                },
                "ordered": {
                  "type": "boolean"
```

```
            "unitSymbol": {
              "type": "string"
            },
            "definition": {
              "type": "array",
              "items": {
                "$ref": "#/definitions/LangString"
              }
            },
            "siNotation": {
              "type": "string"
            },
            "siName": {
              "type": "string"
            },
            "dinNotation": {
              "type": "string"
            },
            "eceName": {
              "type": "string"
            },
            "eceCode": {
              "type": "string"
            },
            "nistName": {
              "type": "string"
            },
            "sourceOfDefinition": {
              "type": "string"
            },
            "conversionFactor": {
              "type": "string"
            },
            "registrationAuthorityId": {
              "type": "string"
            },
            "supplier": {
                }
              }
            }]
          },
          "RelationshipElement": {
            "allOf": [{
              "$ref": "#/definitions/SubmodelElement"
            },
            {
              "properties": {
                "first": {
                  "$ref": "#/definitions/Reference"
                },
                "second": {
                  "$ref": "#/definitions/Reference"
                }
              },
              "required": ["first",
              "second"]
            }]
          },
          "AnnotatedRelationshipElement": {
            "allOf": [{
              "$ref": "#/definitions/RelationshipElement"
            },
            {
              "properties": {
                "annotation": {
                  "type": "array",
                  "items": {
                    "$ref": "#/definitions/Reference"
                  }
                }
              }
            }]
          },
          "Qualifier": {
```

```
            "type": "string"
          }
        },
        "required": ["unitName",
        "unitSymbol",
        "definition"]
      },
      "DataSpecificationIEC61360Content": {
        "allOf": [{
          "$ref": "#/definitions/ValueObject"
        },
        {
          "type": "object",
          "properties": {
            "dataType": {
              "enum": ["DATE",
              "STRING",
              "STRING_TRANSLATABLE",
              "REAL_MEASURE",
              "REAL_COUNT",
              "REAL_CURRENCY",
              "BOOLEAN",
              "URL",
              "RATIONAL",
              "RATIONAL_MEASURE",
              "TIME",
              "TIMESTAMP"]
            },
            "definition": {
              "type": "array",
              "items": {
                "$ref": "#/definitions/LangString"
              }
            },
            "preferredName": {
              "type": "array",
              "items": {
```

```
        "allOf": [{
          "$ref": "#/definitions/Constraint"
        },
        {
          "$ref": "#/definitions/HasSemantics"
        },
        {
          "$ref": "#/definitions/ValueObject"
        },
        {
          "properties": {
            "type": {
              "type": "string"
            }
          },
          "required": ["type"]
        }]
      },
      "Formula": {
        "allOf": [{
          "$ref": "#/definitions/Constraint"
        },
        {
          "properties": {
            "dependsOn": {
              "type": "array",
              "items": {
                "$ref": "#/definitions/Reference"
              }
            }
          }
        }]
      },
      "Security": {
        "type": "object",
        "properties": {
          "accessControlPolicyPoints": {
```

```
                    "$ref": "#/definitions/LangString"
                }
            },
            "shortName": {
                "type": "array",
                "items": {
                    "$ref": "#/definitions/LangString"
                }
            },
            "sourceOfDefinition": {
                "type": "string"
            },
            "symbol": {
                "type": "string"
            },
            "unit": {
                "type": "string"
            },
            "unitId": {
                "$ref": "#/definitions/Reference"
            },
            "valueFormat": {
                "type": "string"
            },
            "valueList": {
                "$ref": "#/definitions/ValueList"
            },
            "levelType": {
                "type": "array",
                "items": {
                    "$ref": "#/definitions/LevelType"
                }
            }
        },
        "required": ["preferredName"]
    }]
},
                    "$ref":
"#/definitions/AccessControlPolicyPoints"
            },
            "certificate": {
                "type": "array",
                "items": {
                    "oneOf": [{
                        "$ref": "#/definitions/BlobCertificate"
                    }]
                }
            },
            "requiredCertificateExtension": {
                "type": "array",
                "items": {
                    "$ref": "#/definitions/Reference"
                }
            }
        },
        "required": ["accessControlPolicyPoints"]
    },
    "Certificate": {
        "type": "object"
    },
    "BlobCertificate": {
        "allOf": [{
            "$ref": "#/definitions/Certificate"
        },
        {
            "properties": {
                "blobCertificate": {
                    "$ref": "#/definitions/Blob"
                },
                "containedExtension": {
                    "type": "array",
                    "items": {
                        "$ref": "#/definitions/Reference"
                    }
                },
```

```
    "LevelType": {
      "type": "string",
      "enum": ["Min",
      "Max",
      "Nom",
      "Typ"]
    },
    "ValueList": {
      "type": "object",
      "properties": {
        "valueReferencePairTypes": {
          "type": "array",
          "minItems": 1,
          "items": {
            "$ref":
"#/definitions/ValueReferencePairType"
          }
        }
      },
      "required": ["valueReferencePairTypes"]
    },
    "ValueReferencePairType": {
      "allOf": [{
        "$ref": "#/definitions/ValueObject"
      }]
    },
    "ValueObject": {
      "type": "object",
      "properties": {
        "value": {
          "type": "string"
        },
        "valueId": {
          "$ref": "#/definitions/Reference"
        },
        "valueType": {
          "type": "string",
          "enum": ["anyUri",
          "lastCertificate": {
            "type": "boolean"
          }
        }
      }]
    },
    "AccessControlPolicyPoints": {
      "type": "object",
      "properties": {
        "policyAdministrationPoint": {
          "$ref":
"#/definitions/PolicyAdministrationPoint"
        },
        "policyDecisionPoint": {
          "$ref": "#/definitions/PolicyDecisionPoint"
        },
        "policyEnforcementPoint": {
          "$ref":
"#/definitions/PolicyEnforcementPoint"
        },
        "policyInformationPoints": {
          "$ref":
"#/definitions/PolicyInformationPoints"
        }
      },
      "required": ["policyAdministrationPoint",
      "policyDecisionPoint",
      "policyEnforcementPoint"]
    },
    "PolicyAdministrationPoint": {
      "type": "object",
      "properties": {
        "localAccessControl": {
          "$ref": "#/definitions/AccessControl"
        },
        "externalAccessControl": {
          "type": "boolean"
        }
```

```
            "base64Binary",
            "boolean",
            "date",
            "dateTime",
            "dateTimeStamp",
            "decimal",
            "integer",
            "long",
            "int",
            "short",
            "byte",
            "nonNegativeInteger",
            "positiveInteger",
            "unsignedLong",
            "unsignedInt",
            "unsignedShort",
            "unsignedByte",
            "nonPositiveInteger",
            "negativeInteger",
            "double",
            "duration",
            "dayTimeDuration",
            "yearMonthDuration",
            "float",
            "gDay",
            "gMonth",
            "gMonthDay",
            "gYear",
            "gYearMonth",
            "hexBinary",
            "NOTATION",
            "QName",
            "string",
            "normalizedString",
            "token",
            "language",
            "Name",
```

```
        },
        "required": ["externalAccessControl"]
      },
      "PolicyInformationPoints": {
        "type": "object",
        "properties": {
          "internalInformationPoint": {
            "type": "array",
            "items": {
              "$ref": "#/definitions/Reference"
            }
          },
          "externalInformationPoints": {
            "type": "boolean"
          }
        },
        "required": ["externalInformationPoints"]
      },
      "PolicyEnforcementPoint": {
        "type": "object",
        "properties": {
          "externalPolicyEnforcementPoint": {
            "type": "boolean"
          }
        },
        "required": ["externalPolicyEnforcementPoint"]
      },
      "PolicyDecisionPoint": {
        "type": "object",
        "properties": {
          "externalPolicyDecisionPoints": {
            "type": "boolean"
          }
        },
        "required": ["externalPolicyDecisionPoints"]
      },
      "AccessControl": {
```

```
            "NCName",
            "ENTITY",
            "ID",
            "IDREF",
            "NMTOKEN",
            "time"]
        }
      }
    },
    "Asset": {
      "allOf": [{
        "$ref": "#/definitions/Identifiable"
      },
      {
        "$ref": "#/definitions/HasDataSpecification"
      },
      {
        "properties": {
          "kind": {
            "$ref": "#/definitions/AssetKind"
          },
          "assetIdentificationModel": {
            "$ref": "#/definitions/Reference"
          },
          "billOfMaterial": {
            "$ref": "#/definitions/Reference"
          }
        },
        "required": ["kind"]
      }]
    },
    "AssetKind": {
      "type": "string",
      "enum": ["Type",
      "Instance"]
    },
    "ModelingKind": {
      "type": "object",
      "properties": {
        "selectableSubjectAttributes": {
          "$ref": "#/definitions/Reference"
        },
        "defaultSubjectAttributes": {
          "$ref": "#/definitions/Reference"
        },
        "selectablePermissions": {
          "$ref": "#/definitions/Reference"
        },
        "defaultPermissions": {
          "$ref": "#/definitions/Reference"
        },
        "selectableEnvironmentAttributes": {
          "$ref": "#/definitions/Reference"
        },
        "defaultEnvironmentAttributes": {
          "$ref": "#/definitions/Reference"
        },
        "accessPermissionRule": {
          "type": "array",
          "items": {
            "$ref":
"#/definitions/AccessPermissionRule"
          }
        }
      }
    },
    "AccessPermissionRule": {
      "allOf": [{
        "$ref": "#/definitions/Referable"
      },
      {
        "$ref": "#/definitions/Qualifiable"
      },
      {
        "properties": {
```

```
        "type": "string",
        "enum": ["Template",
        "Instance"]
      },
      "Submodel": {
        "allOf": [{
          "$ref": "#/definitions/Identifiable"
        },
        {
          "$ref": "#/definitions/HasDataSpecification"
        },
        {
          "$ref": "#/definitions/Qualifiable"
        },
        {
          "$ref": "#/definitions/HasSemantics"
        },
        {
          "properties": {
            "kind": {
              "$ref": "#/definitions/ModelingKind"
            },
            "submodelElements": {
              "type": "array",
              "items": {
                "$ref":
"#/definitions/SubmodelElement"
              }
            }
          }
        }]
      },
      "Constraint": {
        "type": "object",
        "properties": {
          "modelType": {
            "$ref": "#/definitions/ModelType"
          }
          "targetSubjectAttributes": {
            "type": "array",
            "items": {
              "$ref":
"#/definitions/SubjectAttributes"
            },
            "minItems": 1
          },
          "permissionsPerObject": {
            "type": "array",
            "items": {
              "$ref":
"#/definitions/PermissionsPerObject"
            }
          }
        },
        "required": ["targetSubjectAttributes"]
      }]
    },
    "SubjectAttributes": {
      "type": "object",
      "properties": {
        "subjectAttributes": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/Reference"
          },
          "minItems": 1
        }
      }
    },
    "PermissionsPerObject": {
      "type": "object",
      "properties": {
        "object": {
          "$ref": "#/definitions/Reference"
        },
        "targetObjectAttributes": {
```

```
        },
        "required": ["modelType"]
      },
      "Operation": {
        "allOf": [{
          "$ref": "#/definitions/SubmodelElement"
        },
        {
          "properties": {
            "inputVariable": {
              "type": "array",
              "items": {
                "$ref":
"#/definitions/OperationVariable"
              }
            },
            "outputVariable": {
              "type": "array",
              "items": {
                "$ref":
"#/definitions/OperationVariable"
              }
            },
            "inoutputVariable": {
              "type": "array",
              "items": {
                "$ref":
"#/definitions/OperationVariable"
              }
            }
          }
        }]
      },
      "OperationVariable": {
        "type": "object",
        "properties": {
          "value": {
            "oneOf": [{
              "$ref": "#/definitions/ObjectAttributes"
            },
            "permission": {
              "type": "array",
              "items": {
                "$ref": "#/definitions/Permission"
              }
            }
          }
        },
      "ObjectAttributes": {
        "type": "object",
        "properties": {
          "objectAttribute": {
            "type": "array",
            "items": {
              "$ref": "#/definitions/Property"
            },
            "minItems": 1
          }
        }
      },
      "Permission": {
        "type": "object",
        "properties": {
          "permission": {
            "$ref": "#/definitions/Reference"
          },
          "kindOfPermission": {
            "type": "string",
            "enum": ["Allow",
            "Deny",
            "NotApplicable",
            "Undefined"]
          }
        },
        "required": ["permission",
```

```
                "$ref": "#/definitions/Blob"
            },
            {
                "$ref": "#/definitions/File"
            },
            {
                "$ref": "#/definitions/Capability"
            },
            {
                "$ref": "#/definitions/Entity"
            },
            {
                "$ref": "#/definitions/Event"
            },
            {
                "$ref": "#/definitions/BasicEvent"
            },
            {
                "$ref":
"#/definitions/MultiLanguageProperty"
            },
            {
                "$ref": "#/definitions/Operation"
            },
            {
                "$ref": "#/definitions/Property"
            },
            {
                "$ref": "#/definitions/Range"
            },
            {
                "$ref": "#/definitions/ReferenceElement"
            },
            {
                "$ref":
"#/definitions/RelationshipElement"
            },
            {
```

```
                "kindOfPermission"]
        }
    }
}
```

```
                "$ref":
"#/definitions/SubmodelElementCollection"
              }]
            }
```


ii. JSON Example

For cross reference, a complete self-contained example is given, which relates to the unified example in Clause 5.3.

Table 30 JSON example


```
{
  "assetAdministrationShells": [{
    "asset": {
      "keys": [{
        "type": "Asset",
        "local": true,
        "value":
"http://customer.com/assets/KHBVZJSQKIY",
        "idType": "IRI"
      }]
    },
    "submodels": [{
      "keys": [{
        "type": "Submodel",
        "local": true,
        "value":
"http.//i40.customer.com/type/1/1/7A7104BDAB57E184"
,
        "idType": "IRI"
      }]
    },
    {
      "keys": [{
        "type": "Submodel",
        "local": true,
        "value":
"http://i40.customer.com/instance/1/1/AC69B1CB44F079
35",
        "idType": "IRI"
      },
      {
        "identification": {
          "idType": "IRDI",
          "id": "0173-1#02-BAA120#008"
        },
        "administration": {
          "version": "",
          "revision": "2"
        },
        "idShort": "MaxRotationSpeed",
        "category": "PROPERTY",
        "modelType": {
          "name": "ConceptDescription"
        },
        "embeddedDataSpecifications": [{
          "dataSpecification": {
            "keys": []
          },
          "dataSpecificationContent": {
            "preferredName": [{
              "language": "de",
              "text": "max. Drehzahl"
            },
            {
              "language": "en",
              "text": "Max. rotation speed"
```

```
        }]
      },
      {
        "keys": [{
          "type": "Submodel",
          "local": true,
          "value":
"http://i40.customer.com/type/1/1/1A7B62B529F19152",
          "idType": "IRI"
        }]
      }],
      "conceptDictionaries": [],
      "identification": {
        "idType": "IRI",
        "id":
"http://customer.com/aas/9175_7013_7091_9168"
      },
      "idShort": "ExampleMotor",
      "category": "CONSTANT",
      "modelType": {
        "name": "AssetAdministrationShell"
      }
    }],
    "assets": [{
      "assetIdentificationModelRef": {
        "keys": [{
          "type": "Submodel",
          "local": true,
          "value":
"i40.customer.com/type/1/1/F13E8576F6488342",
          "idType": "IRI"
        }]
      },
      "identification": {
        "idType": "IRI",
        "id":
"http://customer.com/assets/KHBVZJSQKIY"
      },
```

```
      }],
      "shortName": [],
      "unit": "1/min",
      "unitId": {
        "keys": [{
          "type": "GlobalReference",
          "local": false,
          "value": "0173-1#05-AAA650#002",
          "idType": "IRDI"
        }]
      },
      "sourceOfDefinition": "",
      "dataType": "REAL_MEASURE",
      "definition": [{
        "language": "de",
        "text": "Höchste zulässige Drehzahl, mit
welcher der Motor oder die Speiseinheit betrieben werden
darf"
      },
      {
        "language": "en",
        "text": "Greatest permissible rotation speed
with which the motor or feeding unit may be operated"
      }]
    }],
    "isCaseOf": [{
      "keys": []
    }]
  },
  {
    "identification": {
      "idType": "IRI",
      "id":
"http://customer.com/cd/1/1/18EBD56F6B43D895"
    },
    "idShort": "RotationSpeed",
    "category": "PROPERTY",
```

```
    "idShort": "ServoDCMotor",
    "category": "",
    "modelType": {
      "name": "Asset"
    },
    "kind": "Instance"
  }],
  "submodels": [{
    "semanticId": {
      "keys": [{
        "type": "GlobalReference",
        "local": false,
        "value": "0173-1#01-AFZ615#016",
        "idType": "IRDI"
      }]
    },
    "identification": {
      "idType": "IRI",
      "id":
"http.//i40.customer.com/type/1/1/7A7104BDAB57E184"
    },
    "idShort": "TechnicalData",
    "category": "CONSTANT",
    "modelType": {
      "name": "Submodel"
    },
    "kind": "Instance",
    "submodelElements": [{
      "value": "5000",
      "semanticId": {
        "keys": [{
          "type": "ConceptDescription",
          "local": true,
          "value": "0173-1#02-BAA120#008",
          "idType": "IRDI"
        }]
      },
    "modelType": {
      "name": "ConceptDescription"
    },
    "embeddedDataSpecifications": [{
      "dataSpecification": {
        "keys": [{
          "type": "GlobalReference",
          "local": false,
          "value": "http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIE
C61360",
          "idType": "IRI"
        }]
      },
      "dataSpecificationContent": {
        "preferredName": [{
          "language": "DE",
          "text": "Aktuelle Drehzahl"
        },
        {
          "language": "EN",
          "text": "Actual rotation speed"
        }],
        "shortName": [{
          "language": "DE",
          "text": "AktuelleDrehzahl"
        },
        {
          "language": "EN",
          "text": "ActualRotationSpeed"
        }],
        "unit": "1/min",
        "unitId": {
          "keys": [{
            "type": "GlobalReference",
            "local": false,
            "value": "0173-1#05-AAA650#002",
```

```
        "constraints": [],
        "idShort": "MaxRotationSpeed",
        "category": "PARAMETER",
        "modelType": {
          "name": "Property"
        },
        "valueType": {
          "dataObjectType": {
            "name": "integer"
          }
        },
        "kind": "Instance"
      }]
    },
    {
      "semanticId": {
        "keys": []
      },
      "identification": {
        "idType": "IRI",
        "id": "http://i40.customer.com/type/1/1/1A7B62B529F19152"
      },
      "idShort": "Documentation",
      "category": "CONSTANT",
      "modelType": {
        "name": "Submodel"
      },
      "kind": "Instance",
      "submodelElements": [{
        "ordered": false,
        "allowDuplicates": false,
        "semanticId": {
          "keys": [{
            "type": "ConceptDescription",
            "local": true,
            "idType": "IRDI"
          }]
        },
        "sourceOfDefinition": "",
        "dataType": "REAL_MEASURE",
        "definition": [{
          "language": "DE",
          "text": "Aktuelle Drehzahl, mit welcher der Motor oder die Speiseinheit betrieben wird"
        },
        {
          "language": "EN",
          "text": "Actual rotation speed with which the motor or feeding unit is operated"
        }]
      }],
      "isCaseOf": []
    },
    {
      "identification": {
        "idType": "IRI",
        "id": "www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Document"
      },
      "idShort": "Document",
      "modelType": {
        "name": "ConceptDescription"
      },
      "embeddedDataSpecifications": [{
        "dataSpecification": {
          "keys": [{
            "type": "GlobalReference",
            "local": false,
            "value": "http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360",
            "idType": "IRI"
```

```
            "value":
"www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Document",
            "idType": "IRI"
          }]
        },
        "constraints": [],
        "idShort": "OperatingManual",
        "category": "",
        "modelType": {
          "name": "SubmodelElementCollection"
        },
        "value": [{
          "value": "Operating Manual",
          "semanticId": {
            "keys": [{
              "type": "ConceptDescription",
              "local": true,
              "value":
"www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Description/
Title",
              "idType": "IRI"
            }]
          },
          "constraints": [],
          "idShort": "Title",
          "modelType": {
            "name": "Property"
          },
          "valueType": {
            "dataObjectType": {
              "name": "langString"
            }
          },
          "kind": "Instance"
        },
        {
          "mimeType": "application/pdf",
          "value": "/aasx/OperatingManual.pdf",
          }]
        },
        "dataSpecificationContent": {
          "preferredName": [],
          "shortName": [{
            "language": "EN",
            "text": "Document"
          },
          {
            "language": "DE",
            "text": "Dokument"
          }],
          "unit": "",
          "sourceOfDefinition": "[ISO 15519-1:2010]",
          "dataType": "STRING",
          "definition": [{
            "language": "DE",
            "text": "Feste und geordnete Menge von für
die Verwendung durch Personen bestimmte Informationen,
die verwaltet und als Einheit zwischen Benutzern und
System ausgetauscht werden kann."
          }]
        }
      }],
      "isCaseOf": [],
      "descriptions": []
    }]
}
```

```
            "semanticId": {
               "keys": [{
                  "type": "ConceptDescription",
                  "local": true,
                  "value":
"www.vdi2770.com/blatt1/Entwurf/Okt18/cd/StoredDocumentRepresentation/DigitalFile",
                  "idType": "IRI"
               }]
            },
            "constraints": [],
            "idShort": "DigitalFile_PDF",
            "category": "PARAMETER",
            "modelType": {
               "name": "File"
            },
            "kind": "Instance"
         }],
         "kind": "Instance"
      }]
   },
   {
      "semanticId": {
         "keys": []
      },
      "qualifiers": [],
      "identification": {
         "idType": "IRI",
         "id":
"http://i40.customer.com/instance/1/1/AC69B1CB44F07935"
      },
      "idShort": "OperationalData",
      "category": "VARIABLE",
      "modelType": {
         "name": "Submodel"
      },
      "kind": "Instance",
```

```
      "submodelElements": [{
        "value": "4370",
        "hasDataSpecification": {
          "reference": []
        },
        "semanticId": {
          "keys": [{
            "type": "ConceptDescription",
            "local": true,
            "value":
"http://customer.com/cd/1/1/18EBD56F6B43D895",
            "idType": "IRI"
          }]
        },
        "constraints": [],
        "idShort": "RotationSpeed",
        "category": "VARIABLE",
        "modelType": {
          "name": "Property"
        },
        "valueType": {
          "dataObjectType": {
            "name": "integer"
          }
        },
        "kind": "Instance"
      }]
    }],
    "conceptDescriptions": [{
      "identification": {
        "idType": "IRI",
        "id":
"www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Description/
Title"
      },
      "idShort": "Title",
      "category": "CONSTANT",
      "modelType": {
```

```
      "name": "ConceptDescription"
    },
    "embeddedDataSpecifications": [{
      "dataSpecification": {
        "keys": [{
          "type": "GlobalReference",
          "local": false,
          "value": "http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360",
          "idType": "IRI"
        }]
      },
      "dataSpecificationContent": {
        "preferredName": [{
          "language": "EN",
          "text": "Title"
        },
        {
          "language": "DE",
          "text": "Titel"
        }],
        "shortName": [{
          "language": "EN",
          "text": "Title"
        },
        {
          "language": "DE",
          "text": "Titel"
        }],
        "unit": "",
        "sourceOfDefinition": "",
        "dataType": "STRING_TRANSLATABLE",
        "definition": [{
          "language": "DE",
          "text": "Sprachabhängiger Titel des Dokuments."
        }]
```

```
        }
      }],
      "isCaseOf": []
    },
    {
      "identification": {
        "idType": "IRI",
        "id":
"www.vdi2770.com/blatt1/Entwurf/Okt18/cd/StoredDocumentRepresentation/DigitalFile"
      },
      "idShort": "DigitalFile",
      "modelType": {
        "name": "ConceptDescription"
      },
      "embeddedDataSpecifications": [{
        "dataSpecification": {
          "keys": [{
            "type": "GlobalReference",
            "local": false,
            "value": "http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360",
            "idType": "IRI"
          }]
        },
        "dataSpecificationContent": {
          "preferredName": [{
            "language": "EN",
            "text": "Digital File"
          },
          {
            "language": "DE",
            "text": "Digitale Datei"
          }],
          "shortName": [{
            "language": "EN",
            "text": "DigitalFile"
```

```
        },
        {
          "language": "DE",
          "text": "DigitaleDatei"
        }],
        "unit": "",
        "sourceOfDefinition": "",
        "dataType": "STRING",
        "definition": [{
          "language": "DE",
          "text": "Eine Datei, die die DocumentVersion repräsentiert. Neben der obligatorischen PDF/A Datei können weitere Dateien angegeben werden."
        }]
      }
    }],
    "isCaseOf": []
```

Annex G. RDF schema and complete example

i. RDF Data Model for the Administration Shell

```
@prefix aas: <http://admin-shell.io/aas/2/0#> .
@prefix dash: <http://datashapes.org/dash#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix vann: <http://purl.org/vocab/vann/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@base <http://admin-shell.io/aas/2/0#> .

<http://admin-shell.io/aas/2/0#> rdf:type owl:Ontology ;
   vann:preferredNamespaceUri "http://admin-shell.io/aas/2/0#"^^xsd:string ;
   owl:versionInfo 2.0 ;
   rdfs:comment "An Ontology to represents the data model for the Asset Administration Shell accoridng to the specification 'Details of the Asset Administration Shell - Part 1 - Version 2.0'."@en ;
   skos:prefLabel "aas"@en ;
   vann:preferredNamespacePrefix "aas"^^xsd:string ;
   rdfs:isDefinedBy <http://admin-shell.io/aas/2/0#> ;
.

#################################################################
#    Object Properties
#################################################################

###  https://admin-shell.io/aas/2/0/rdf#annotation
aas:annotation rdf:type owl:ObjectProperty ;
    rdfs:comment "Annotations that hold for the relationships between the two elements."@en ;
    rdfs:label "has annotation"@en ;
    rdfs:domain aas:AnnotatedRelationshipElement ;
.

###  https://admin-shell.io/aas/2/0/rdf#administration
```

```
aas:administration rdf:type owl:ObjectProperty ;
    rdfs:comment "Administrative information of an identifiable element."@en ;
    skos:note "Some of the administrative information like the version number might need to be part of the identification."@en ;
    rdfs:label "has administration"@en ;
    rdfs:domain aas:Identifiable ;
    rdfs:range aas:AdministrativeInformation ;
    .

### https://admin-shell.io/aas/2/0/rdf#asset
aas:asset rdf:type owl:ObjectProperty ;
     #rdfs:domain aas:AssetAdministrationShell;
     rdfs:comment "The asset the AAS is representing."@en ;
     rdfs:label "has asset"@en .

### https://admin-shell.io/aas/2/0/rdf#assetKind
aas:assetKind rdf:type owl:ObjectProperty ;
    rdfs:domain aas:Asset ;
    rdfs:range aas:AssetKind ;
    rdfs:label "has asset kind"^^xsd:string ;
    rdfs:comment "Denotes whether the Asset of of kind 'Type' or 'Instance'."@en ;
.

### https://admin-shell.io/aas/2/0/rdf#assetIdentificationModel
aas:assetIdentificationModel rdf:type owl:ObjectProperty ;
    rdfs:domain aas:Asset ;
    rdfs:range aas:Reference ;
    rdfs:comment "A reference to a Submodel that defines the handling of additional domain specific (proprietary) Identifiers for the asset like e.g. serial number etc."@en ;
    rdfs:label "has Asset Identification Model"^^xsd:string .

### https://admin-shell.io/aas/2/0/rdf#billOfMaterial
aas:billOfMaterial rdf:type owl:ObjectProperty ;
    rdfs:domain aas:Asset ;
    rdfs:range aas:Reference ;
    rdfs:comment "Bill of material of the asaset represented by a submodel of the same AAS. This submodel contains a set of entities describing the material used to compose the composite I4.0 Component."@en ;
    rdfs:label "has Bill of Material"^^xsd:string ;
```

```
.


###  https://admin-shell.io/aas/2/0/rdf#blobCertificate

aas:blobCertificate rdf:type owl:ObjectProperty ;

    rdfs:comment "Certificate as BLOB."@en ;

    rdfs:domain aas:BlobCertificate ;

    rdfs:range aas:Blob ;

.


###  https://admin-shell.io/aas/2/0/rdf#certificate

aas:certificate rdf:type owl:ObjectProperty ;

   rdfs:comment "Certificates of the AAS."@en ;

   rdfs:label "has certificate"^^xsd:string ;

   rdfs:domain aas:Security ;

   rdfs:range aas:Certificate ;

.


###  https://admin-shell.io/aas/2/0/rdf#conceptDescription

aas:conceptDescription rdf:type owl:ObjectProperty ;

    rdfs:domain aas:ConceptDictionary;

    rdfs:range aas:ConceptDescription ;

    rdfs:comment "The semantics of a property or other elements that may have a semantic description is defined by a concept description. The description of the concept should follow a standardized schema (realized as data specification template)."@en ;

    rdfs:label "has concept description" .


###  https://admin-shell.io/aas/2/0/rdf#conceptDictionary

aas:conceptDictionary rdf:type owl:ObjectProperty ;

    rdfs:domain aas:AssetAdministrationShell;

    rdfs:range aas:ConceptDictionary ;

    rdfs:comment "An AAS max have one or more concept dictionaries assigned to it. The concept dictionaries typically contain only descriptions for elements that are also used within the AAS (via HasSemantics)."@en ;

    rdfs:label "has concept dictionary" .


###  https://admin-shell.io/aas/2/0/rdf#containedElement

aas:containedElement rdf:type owl:ObjectProperty ;

    rdfs:comment "Referable elements that are contained in the view."@en ;

    rdfs:label "contains element"^^xsd:string ;
```

```
    rdfs:domain aas:View ;
.

### https://admin-shell.io/aas/2/0/rdf#containedExtension
aas:containedExtension rdf:type owl:ObjectProperty ;
    rdfs:comment "Extensions contained in the certificate."@en ;
    rdfs:label "contains extension"^^xsd:string ;
    rdfs:domain aas:BlobCertificate ;
    rdfs:range aas:Reference ;
.

### https://admin-shell.io/aas/2/0/rdf#content
aas:content rdf:type owl:ObjectProperty ;
   rdfs:label "has content"^^xsd:string ;
   rdfs:comment ""@en ;
   rdfs:domain aas:DataSpecification ;
   rdfs:range aas:DataSpecificationContent ;
.

### https://admin-shell.io/aas/2/0/rdf#dataSpecification
aas:dataSpecification rdf:type owl:ObjectProperty ;
    rdfs:comment "Global reference to the data specification template used by the element."@en ;
    rdfs:label "has Data Specification"@en ;
    rdfs:domain aas:HasDataSpecification ;
    rdfs:range aas:Reference ;
.

### https://admin-shell.io/aas/2/0/rdf#defaultEnvironmentAttributes
aas:defaultEnvironmentAttributes rdf:type owl:ObjectProperty ;
   rdfs:comment "Reference to a submodel defining default environment attributes, i.e. attributes that are not describing the asset itself. The submodel is of kind=Type. At the same type the values of these environment attributes need to be accessible when evaluating the access permission rules. This is realized as a policy information point."@en ;
    rdfs:label "has default environment attributes"^^xsd:string .

### https://admin-shell.io/aas/2/0/rdf#defaultPermissions
aas:defaultPermissions rdf:type owl:ObjectProperty ;
     rdfs:comment "Reference to a submodel defining the default permissions for the AAS."@en ;
```

```
    rdfs:label "has default permissions"^^xsd:string .


###  https://admin-shell.io/aas/2/0/rdf#defaultSubjectAttributes

aas:defaultSubjectAttributes rdf:type owl:ObjectProperty ;

  rdfs:comment "Reference to a submodel defining the default subjects attributes for the AAS that can be used to describe access permission rules. The submodel is of kind=Type."@en ;

  rdfs:label "has default subject attributes"^^xsd:string .


###  https://admin-shell.io/aas/2/0/rdf#dependsOn

aas:dependsOn rdf:type owl:ObjectProperty ;

    rdfs:domain aas:Formula ;

    rdfs:range aas:Reference ;

    rdfs:comment "A formula may depend on referable or even external global elements - assumed that can be referenced and their value may be evaluated - that are used in the logical expression."@en ;

    rdfs:label "depends on"@en .


###  https://admin-shell.io/aas/2/0/rdf#derivedFrom

aas:derivedFrom rdf:type owl:ObjectProperty ;

  rdfs:subPropertyOf <http://www.w3.org/TR/2013/REC-prov-o-20130430/#wasDerivedFrom> ;

  rdfs:domain aas:AssetAdministrationShell ;

  rdfs:range aas:Reference ;

  rdfs:comment "This relation connects instances of AAS with their respective types. Refer to Asset Kind for further information of instance and type kinds."@en ;

  rdfs:label "was derived from"^^xsd:string .


###  https://admin-shell.io/aas/2/0/rdf#describes

aas:describes rdf:type owl:ObjectProperty ;

    rdfs:domain aas:AssetAdministrationShell;

    rdfs:range aas:Asset ;

    rdfs:comment "Connect the Asset with the Administration Shell"@en ;

    rdfs:label "describes"@en .


###  https://admin-shell.io/aas/2/0/rdf#embeddedDataSpecification

aas:embeddedDataSpecification rdf:type owl:ObjectProperty ;

  rdfs:label "has embedded data specification"^^xsd:string ;

  rdfs:comment ""@en ;

.
```

```
### https://admin-shell.io/aas/2/0/rdf#entityType
aas:entityType rdf:type owl:ObjectProperty ;
    rdfs:label "has entity type"^^xsd:string ;
    rdfs:domain aas:Entity ;
    rdfs:range rdfs:EntityType ;
    rdfs:comment "Describes whether the entity is a co-managed entity or a self-managed entity."@en ;
.

### https://admin-shell.io/aas/2/0/rdf#idType
aas:idType rdf:type owl:ObjectProperty ;
    rdfs:comment "Type of the Identifier, e.g. IRI, IRDI etc. The supported Identifier types are defined in the enumeration 'IdentifierType'."@en ;
    rdfs:domain aas:Identifier ;
    rdfs:range aas:IdentifierType ;
    rdfs:label "has idType"@en
.

### https://admin-shell.io/aas/2/0/rdf#identification
aas:identification rdf:type owl:ObjectProperty ;
    rdfs:domain aas:Identifiable ;
    rdfs:label "has identification"@en ;
    rdfs:comment "The globally unique identification of the element."@en ;
.

### https://admin-shell.io/aas/2/0/rdf#inputVariable
aas:inputVariable rdf:type owl:ObjectProperty ;
    rdfs:comment "Input parameter of the operation."@en ;
    rdfs:label "has input variable"^^xsd:string ;
    rdfs:domain aas:Operation ;
    rdfs:range aas:OperationVariable ;
    .

### https://admin-shell.io/aas/2/0/rdf#inoutputVariable
aas:inoutputVariable rdf:type owl:ObjectProperty ;
    rdfs:comment "Parameter that is input and output of the operation."@en ;
    rdfs:label "has input/output variable"^^xsd:string ;
    rdfs:domain aas:Operation ;
    rdfs:range aas:OperationVariable ;
```

```
    .

###  https://admin-shell.io/aas/2/0/rdf#inAccordanceWith
aas:inAccordanceWith rdf:type owl:ObjectProperty ;
   rdfs:domain aas:Submodel ;
   rdfs:comment "The Submodel is in accordance with a given standard specified in the STO ontology"@en ;
   rdfs:label "is in accordance With"@en .

###  https://admin-shell.io/aas/2/0/rdf#isCaseOf
aas:isCaseOf rdf:type owl:ObjectProperty ;
   rdfs:subPropertyOf dcterms:identifier ;
    rdfs:comment "Global reference to an external definition the concept is compatible to or was derived from."@en ;
    skos:note "Compare to is-case-of relationship in ISO 13584-32 & IEC EN 61360."@en ;
    rdfs:label "is case of"^^xsd:string ;
    rdfs:domain aas:ConceptDescription ;
.

###  https://admin-shell.io/aas/2/0/rdf#keyElement
aas:keyElement rdf:type owl:ObjectProperty ;
    rdfs:comment ""@en ;
    rdfs:label "has key element"^^xsd:string ;
    rdfs:domain aas:Key ;
    rdfs:range aas:KeyElement ;
.

###  https://admin-shell.io/aas/2/0/rdf#key
aas:key rdf:type owl:ObjectProperty ;
    rdfs:comment "Unique reference in its name space."@en ;
    rdfs:label "has key"^^xsd:string ;
    rdfs:range aas:Key ;
    .

###  https://admin-shell.io/aas/2/0/rdf#kind
aas:kind rdf:type owl:ObjectProperty ;
    rdfs:domain aas:HasKind ;
    rdfs:range aas:ModelingKind ;
    rdfs:label "has kind"^^xsd:string ;
```

```
    rdfs:comment "ModelingKind of the element: either type or instance."@en ;
.


###  https://admin-shell.io/aas/2/0/rdf#max

aas:max rdf:type owl:ObjectProperty ;

    rdfs:domain aas:Range ;

    rdfs:range rdfs:Literal ;

    rdfs:label "has maximum value"@en ;

    rdfs:comment "The maximum value of the range."@en ;

    skos:note "Constraint AASd-012-3: In case of a range with kind=Instance either the min or the max value or both need to be defined."@en ;

.


###  https://admin-shell.io/aas/2/0/rdf#mimeType

aas:mimeType rdf:type owl:ObjectProperty ;

    rdfs:domain [ rdf:type owl:Class ; owl:unionOf ( aas:File aas:Blob) ;] ;

   rdfs:range xsd:string ;

   rdfs:comment "Mime type of the content of the BLOB. The mime type states which file extension the file has. Valid values are e.g. 'application/json', 'application/xls', 'image/jpg' The allowed values are defined as in RFC2046."@en ;

   rdfs:comment "Mime type of the content of the File."@en ;

   rdfs:label "has mimetype"^^xsd:string ;

   rdfs:seeAlso "http://uri4uri.net/vocab.html/#MimetypeDatatype"^^xsd:string ;

   .


###  https://admin-shell.io/aas/2/0/rdf#min

aas:min rdf:type owl:ObjectProperty ;

    rdfs:domain aas:Range ;

    rdfs:range rdfs:Literal ;

    rdfs:label "has minimum value"@en ;

    rdfs:comment "The minimum value of the range."@en ;

    skos:note "Constraint AASd-012-3: In case of a range with kind=Instance either the min or the max value or both need to be defined."@en ;

.


###  https://admin-shell.io/aas/2/0/rdf#object

aas:object rdf:type owl:ObjectProperty ;

    rdfs:comment "Element to which permission shall be assigned."@en ;

    rdfs:label "has object"^^xsd:string .
```

```
###  https://admin-shell.io/aas/2/0/rdf#observed
aas:observed rdf:type owl:ObjectProperty ;
    rdfs:comment "Reference to the data or other elements that are being observed."@en ;
    rdfs:label "observed by"^^xsd:string ;
    rdfs:domain aas:BasicEvent ;
    rdfs:range aas:Reference ;
.

###  https://admin-shell.io/aas/2/0/rdf#objectAttribute
aas:objectAttribute rdf:type owl:ObjectProperty ;
     rdfs:comment "A data element that further classifies an object."@en ;
     rdfs:label "has object attribute"^^xsd:string .

###  https://admin-shell.io/aas/2/0/rdf#outputVariable
aas:outputVariable rdf:type owl:ObjectProperty ;
   rdfs:comment "Output parameter of the operation."@en ;
   rdfs:label "has output variable"@en ;
    rdfs:domain aas:Operation ;
    rdfs:range aas:OperationVariable ;
.

###  https://admin-shell.io/aas/2/0/rdf#parent
aas:parent rdf:type owl:ObjectProperty ;
    rdfs:comment "Reference to the next referable parent element of the element. Constraint AASd-004: Add parent in
case of non-identifiable elements."@en ;
    skos:note "This element is used to ease navigation in the model and thus it enables more performant"@en ;
    rdfs:label "has parent"@en ;
    rdfs:domain aas:Referable ;
    rdfs:range aas:Referable ;
    .

###  https://admin-shell.io/aas/2/0/rdf#preferredName
aas:preferredName rdf:type owl:ObjectProperty ;
     rdfs:comment "This is the preferredName of a ConceptDescription object"@en ;
     rdfs:label "has preferred name"^^xsd:string .

###  https://admin-shell.io/aas/2/0/rdf#propertyCategory
```

```
aas:propertyCategory rdf:type owl:ObjectProperty ;
   rdfs:label "has property category"^^xsd:string ;
   rdfs:comment "The following categories are defined for properties and multi-language properties: aas:CONSTANT, aas:PARAMETER, and aas:VARIABLE."@en ;
.

###  https://admin-shell.io/aas/2/0/rdf#qualifier
aas:qualifier rdf:type owl:ObjectProperty ;
    rdfs:comment "Additional qualification of a qualifiable element."@en ;
    rdfs:label "has qualifier"^^xsd:string ;
    rdfs:domain aas:Qualifiable ;
    rdfs:range aas:Constraint ;
.

###  https://admin-shell.io/aas/2/0/rdf#qualifierType
aas:qualifierType rdf:type owl:ObjectProperty ;
    rdfs:comment "The qualifier type describes the type of the qualifier that is applied to the element."@en ;
    rdfs:label "has qualifier type"^^xsd:string ;
    rdfs:domain aas:Qualifier ;
    rdfs:range xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#qualifierValue
aas:qualifierValue rdf:type owl:ObjectProperty ;
    rdfs:comment "The qualifier value is the value of the qualifier."@en ;
    rdf:label "has qualifier value"^^xsd:string ;
    skos:note "Constraint AASd-006: if both, the value and the valueId are present then the value needs to be identical to the short name of the referenced coded value in qualifierValueId."@en ;
    rdfs:domain aas:Qualifier ;
    rdfs:range rdfs:Literal ;
.

###  https://admin-shell.io/aas/2/0/rdf#qualifierValueId
aas:qualifierValueId rdf:type owl:ObjectProperty ;
    rdfs:comment "Reference to the global unqiue id of a coded value."@en ;
    rdf:label "has qualifier value id"^^xsd:string ;
    rdfs:domain aas:Qualifier ;
    rdfs:range aas:Reference ;
```

```
    .


###  https://admin-shell.io/aas/2/0/rdf#relationshipFirst
aas:relationshipFirst rdf:type owl:ObjectProperty ;
     rdfs:comment "First element in the relationship taking the role of the subject."@en ;
     rdfs:label "has first relationship"^^xsd:string ;
    rdfs:domain aas:RelationshipElement ;
    rdfs:range aas:Referable ;
.


###  https://admin-shell.io/aas/2/0/rdf#relationshipSecond
aas:relationshipSecond rdf:type owl:ObjectProperty ;
    rdfs:comment "Second element in the relationship taking the role of the object."@en ;
    rdfs:label "has second relationship"^^xsd:string ;
    rdfs:domain aas:RelationshipElement ;
    rdfs:range aas:Referable ;
.


###  https://admin-shell.io/aas/2/0/rdf#security
aas:security rdf:type owl:ObjectProperty ;
    rdfs:comment "Definition of the security relevant aspects of the AAS."@en ;
    rdfs:label "has security"^^xsd:string ;
    rdfs:domain aas:AssetAdministrationShell ;
    .


###  https://admin-shell.io/aas/2/0/rdf#selectableEnvironmentAttributes
aas:selectableEnvironmentAttributes rdf:type owl:ObjectProperty ;
     rdfs:comment "Reference to a submodel defining which environment attributes can be accessed via the permission rules defined for the AAS, i.e. attributes that are not describing the asset itself. Default: reference to the submodel referenced via defaultEnvironmentAttributes"@en ;
     rdfs:label "has selectable environment attributes"^^xsd:string ;
     .


###  https://admin-shell.io/aas/2/0/rdf#semanticId
aas:semanticId rdf:type owl:ObjectProperty ;
   rdfs:subPropertyOf rdfs:seeAlso ;
   rdfs:label "has semantic ID"^^xsd:string ;
   skos:altLabel "has Semantic Expression"@en ;
```

```
    rdfs:comment "Points to the Expression Semantic of the Submodels"@en ;
    rdfs:comment "The semantic id might refer to an external information source, which explains the formulation of the submodel (for example an PDF if a standard)."@en ;
    rdfs:domain aas:HasSemantics ;
.

### https://admin-shell.io/aas/2/0/rdf#statement
aas:statement rdf:type owl:DatatypeProperty ;
     rdfs:label "has statement"^^xsd:string ;
     rdfs:comment "Describes statements applicable to the entity by a set of submodel elements, typically with a qualified value."@en ;
     rdfs:domain aas:Entity ;
     rdfs:range rdf:SubmodelElement ;
.

### https://admin-shell.io/aas/2/0/rdf#subjectAttribute
aas:subjectAttribute rdf:type owl:ObjectProperty ;
    rdfs:comment "A data element that further classifies a specific subject."@en ;
    rdfs:label "has subject attribute"^^xsd:string .

### https://admin-shell.io/aas/2/0/rdf#submodel
aas:submodel rdf:type owl:ObjectProperty ;
    rdfs:domain aas:AssetAdministrationShell ;
    rdfs:range aas:Submodel ;
    rdfs:comment "Points from the Admin Shell to the Submodels that describe the Admin Shell of a given Asset"@en ;
    rdfs:label "has Submodel"@en ;
    .

### https://admin-shell.io/aas/2/0/rdf#submodelElement
aas:submodelElement rdf:type owl:ObjectProperty ;
     rdfs:domain aas:Submodel ;
      rdfs:range aas:SubmodelElement ;
      rdfs:comment "A submodel consists of zero or more submodel elements."@en ;
      rdfs:label "has Submodel Element"@en .

### https://admin-shell.io/aas/2/0/rdf#targetObjectAttributes
aas:targetObjectAttributes rdf:type owl:ObjectProperty ;
```

```
    rdfs:comment "Target object attributes that need to be fulfilled so that the access permissions apply to the accessing subject."@en ;
    rdfs:label "has target object attributes"^^xsd:string .


###  https://admin-shell.io/aas/2/0/rdf#targetSubjectAttributes
aas:targetSubjectAttributes rdf:type owl:ObjectProperty ;
    rdfs:comment "Target subject attributes that need to be fulfilled by the accessing subject to get the permissions defined by this rule."@en ;
    rdfs:label "has target subject attributes"^^xsd:string .


###  https://admin-shell.io/aas/2/0/rdf#value
aas:value rdf:type rdf:Property ;
    rdfs:comment "The value, for example an IRDI if the idType=IRDI, a literal, or anything else."@en ;
    rdfs:label "has value"^^xsd:string ;
    .


###  https://admin-shell.io/aas/2/0/rdf#view
aas:view rdf:type owl:ObjectProperty ;
    rdfs:domain aas:AssetAdministrationShell;
    rdfs:range aas:View ;
    rdfs:comment "Points to the differents views associated to the Administration Shell via the Submodels."@en ;
    rdfs:label "has View"@en ;
    skos:prefLabel "view"@en .


#################################################################
#    Data properties
#################################################################


###  https://admin-shell.io/aas/2/0/rdf#allowDuplicates
aas:allowDuplicates rdf:type owl:DatatypeProperty ;
    rdfs:comment "If allowDuplicates=true then it is allowed that the collection contains the same element several times. Default = false"@en ;
    skos:altLabel "allow duplicates"^^xsd:string ;
   rdfs:domain aas:SubmodelElementCollection ;
   rdfs:range xsd:boolean ;
.


###  https://admin-shell.io/aas/2/0/rdf#category
```

```
aas:category rdf:type owl:DatatypeProperty ;
   rdfs:label "has category"^^xsd:string ;
   rdfs:comment "The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints."@en ;
   rdfs:domain aas:Referable ;
   rdfs:range xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#conversionFactor
aas:conversionFactor rdf:type owl:DatatypeProperty ;
     rdfs:comment ""@en ;
     rdfs:label "has conversion factor"^^xsd:string ;
     rdfs:domain aas:DataSpecificationPhysicalUnit ;
     rdfs:range xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#datatype
aas:datatype rdf:type owl:ObjectProperty ;
    rdfs:label "has datatype"^^xsd:string ;
    rdfs:comment ""@en ;
    rdfs:domain aas:DataSpecificationIEC61360 ;
    rdfs:range aas:DataTypeIEC61360 ;
.

###  https://admin-shell.io/aas/2/0/rdf#definition
aas:definition rdf:type owl:DatatypeProperty ;
    rdfs:label "has definition"^^xsd:string ;
    rdfs:comment ""@en ;
    rdfs:domain aas:DataSpecificationContent ;
    rdfs:range rdf:langString ;
.

###  https://admin-shell.io/aas/2/0/rdf#description
aas:description rdf:type owl:DatatypeProperty ;
     rdfs:subPropertyOf rdfs:comment ;
     rdfs:label "has description"^^xsd:string ;
     rdfs:domain aas:Referable ;
     rdfs:range rdfs:Literal ;
```

```
    rdfs:comment "Description or comments on the element. The description can be provided in several languages."@en ;
.

###  https://admin-shell.io/aas/2/0/rdf#dinNotation
aas:dinNotation rdf:type owl:DatatypeProperty ;
    rdfs:comment ""@en ;
    rdfs:label "has DIN notation"^^xsd:string ;
    rdfs:domain aas:DataSpecificationPhysicalUnit ;
    rdfs:range xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#eceCode
aas:eceCode rdf:type owl:DatatypeProperty ;
    rdfs:comment ""@en ;
    rdfs:label "has ECE code"^^xsd:string ;
    rdfs:domain aas:DataSpecificationPhysicalUnit ;
    rdfs:range xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#eceName
aas:eceName rdf:type owl:DatatypeProperty ;
    rdfs:comment ""@en ;
    rdfs:label "has ECE name"^^xsd:string ;
    rdfs:domain aas:DataSpecificationPhysicalUnit ;
    rdfs:range xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#externalPolicyDecisionPoint
aas:externalPolicyDecisionPoint rdf:type owl:DatatypeProperty ;
    rdfs:comment "If externalPolicyDecisionPoints True then Endpoints to external available decision points taking into consideration for access control for the AAS need to be configured."@en ;
    rdfs:label "is external policy decision point defined"^^xsd:string ;
    rdfs:domain aas:PolicyEnforcementPoint ;
    rdfs:range xsd:boolean ;
.

###  https://admin-shell.io/aas/2/0/rdf#externalPolicyEnforcementPoint
```

```
aas:externalPolicyEnforcementPoint rdf:type owl:DatatypeProperty ;
    rdfs:comment "If externalPolicyEnforcementPoint True then an Endpoint to external available enforcement point taking needs to be configured for the AAS."@en ;
    rdfs:label "is external policy enforcement point defined"^^xsd:string ;
    rdfs:domain aas:PolicyEnforcementPoint ;
    rdfs:range xsd:boolean ;
.

###  https://admin-shell.io/aas/2/0/rdf#id
aas:id rdf:type owl:DatatypeProperty ;
    rdfs:domain aas:Identifier ;
    rdfs:range rdfs:Literal ;
    rdfs:comment "A globally unique identifier which might not be a URI. Its type is defined in idType."@en ;
    rdfs:label "has identification"@en .

###  https://admin-shell.io/aas/2/0/rdf#identification
aas:identification rdf:type owl:DatatypeProperty ;
   rdfs:domain aas:Identifiable ;
   rdfs:label "has identification"^^xsd:string ;
   rdfs:comment "The globally unique identification of the element."@en ;
.

###  https://admin-shell.io/aas/2/0/rdf#idShort
aas:idShort rdf:type owl:DatatypeProperty ;
    rdfs:subPropertyOf dcterms:identifier ;
    rdfs:label "has short id"^^xsd:string ;
    rdfs:comment "Identifying string of the element within its name space."@en ;
    skos:note "Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space."@en ;
    skos:note "Constraint AASd-002: idShort shall only feature letters, digits, underscore ('_'); starting mandatory with a letter."@en ;
    skos:note "Constraint AASd-003: idShort shall be matched case-insensitive."@en ;
    skos:note "Note: In case the element is a property and the property has a semantic definition (HasSemantics) the idShort is typically identical to the short name in English. "@en ;
    skos:note "Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the 'BrowserPath' in OPC UA."@en ;
    rdfs:domain aas:Referable ;
    rdfs:range xsd:string ;
```

```
.


###  https://admin-shell.io/aas/2/0/rdf#index

aas:index rdf:type owl:DatatypeProperty ;

    rdfs:label "has index"^^xsd:string ;

    rdfs:comment "The index attribute states the position of each key instance in the sequence of all other related keys."@en ;

    rdfs:domain aas:Key ;

    rdfs:range xsd:integer ;

.


###  https://admin-shell.io/aas/2/0/rdf#lastCertificate

aas:lastCertificate rdf:type owl:DatatypeProperty ;

   rdfs:comment "Denotes whether this certificate is the certificated that fast added last."@en ;

   rdfs:label "is last certificate"^^xsd:string ;

   rdfs:domain aas:BlobCertificate ;

   rdfs:range xsd:boolean ;

.


###  https://admin-shell.io/aas/2/0/rdf#local

aas:local rdf:type owl:DatatypeProperty ;

   rdfs:comment "Denotes if the key references a model element of the same AAS (=true) or not (=false). In case of local = false the key may reference a model element of another AAS or an entity outside any AAS that has a global unique id."@en ;

   rdfs:label "is local"^^xsd:string ;

   rdfs:domain aas:Key ;

   rdfs:range xsd:boolean ;

.


###  https://admin-shell.io/aas/2/0/rdf#nistName

aas:nistName rdf:type owl:DatatypeProperty ;

    rdfs:comment ""@en ;

    rdfs:label "has NIST name"^^xsd:string ;

    rdfs:domain aas:DataSpecificationPhysicalUnit ;

    rdfs:range xsd:string ;

.


###  https://admin-shell.io/aas/2/0/rdf#siName
```

```
aas:siName rdf:type owl:DatatypeProperty ;
    rdfs:comment ""@en ;
    rdfs:label "has SI name"^^xsd:string ;
    rdfs:domain aas:DataSpecificationPhysicalUnit ;
    rdfs:range xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#siNotation
aas:siNotation rdf:type owl:DatatypeProperty ;
    rdfs:comment ""@en ;
    rdfs:label "has SI notation"^^xsd:string ;
    rdfs:domain aas:DataSpecificationPhysicalUnit ;
    rdfs:range xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#ordered
aas:ordered rdf:type owl:DatatypeProperty ;
    rdfs:comment "If ordered=false then the elements in the property collection are not ordered. If ordered=true then the elements in the collection are ordered. Default = false"@en ;
    rdfs:label "ordered"^^xsd:string ;
   rdfs:domain aas:SubmodelElementCollection ;
   rdfs:range xsd:boolean ;
.

###  https://admin-shell.io/aas/2/0/rdf#preferedName
aas:preferedName rdf:type owl:DatatypeProperty ;
   rdfs:label "has prefered name"^^xsd:string ;
   rdfs:comment ""@en ;
   rdfs:domain aas:DataSpecificationIEC61360 ;
   rdfs:range rdf:langString ;
.

###  https://admin-shell.io/aas/2/0/rdf#registrationAuthority
aas:registrationAuthority rdf:type owl:DatatypeProperty ;
    rdfs:comment ""@en ;
    rdfs:label "has registration authority"^^xsd:string ;
    rdfs:domain aas:DataSpecificationPhysicalUnit ;
    rdfs:range xsd:string ;
```

```
.


###  https://admin-shell.io/aas/2/0/rdf#revision

aas:revision rdf:type owl:DatatypeProperty ;

   rdfs:subPropertyOf dcterms:hasVersion ;

   rdfs:comment "Revision of the element. Constraint AASd-005: A revision requires a version. This means, if there is no version there is no revision neither."@en ;

   rdfs:label "has revision"@en ;

   rdfs:domain aas:AdministrativeInformation ;

   rdfs:range xsd:string ;

.


###  https://admin-shell.io/aas/2/0/rdf#shortName

aas:shortName rdf:type owl:DatatypeProperty ;

    rdfs:label "has short name"^^xsd:string ;

    rdfs:comment ""@en ;

    rdfs:domain aas:DataSpecificationIEC61360 ;

    rdfs:range rdf:langString ;

.


###  https://admin-shell.io/aas/2/0/rdf#sourceOfDefinition

aas:sourceOfDefinition rdf:type owl:DatatypeProperty ;

    rdfs:label "has source of definition"^^xsd:string ;

    rdfs:comment ""@en ;

    rdfs:domain aas:DataSpecificationContent ;

    rdfs:range rdf:langString ;

.


###  https://admin-shell.io/aas/2/0/rdf#supplier

aas:supplier rdf:type owl:DatatypeProperty ;

     rdfs:comment ""@en ;

     rdfs:label "has supplier"^^xsd:string ;

     rdfs:domain aas:DataSpecificationPhysicalUnit ;

     rdfs:range xsd:string ;

.


###  https://admin-shell.io/aas/2/0/rdf#symbol

aas:symbol rdf:type owl:DatatypeProperty ;
```

```
    rdfs:label "has symbol"^^xsd:string ;
    rdfs:comment ""@en ;
    rdfs:domain aas:DataSpecificationIEC61360 ;
    rdfs:range xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#unit
aas:unit rdf:type owl:DatatypeProperty ;
    rdfs:label "has unit"^^xsd:string ;
    rdfs:comment ""@en ;
    rdfs:domain aas:DataSpecificationIEC61360 ;
    rdfs:range xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#unitId
aas:unitId rdf:type owl:DatatypeProperty ;
    rdfs:label "has unit id"^^xsd:string ;
    rdfs:comment ""@en ;
    rdfs:domain aas:DataSpecificationIEC61360 ;
    rdfs:range aas:Reference ;
.

###  https://admin-shell.io/aas/2/0/rdf#unitName
aas:unitName rdf:type owl:DatatypeProperty ;
     rdfs:comment ""@en ;
     rdfs:label "unit has name"^^xsd:string ;
     rdfs:domain aas:DataSpecificationPhysicalUnit ;
     rdfs:range xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#unitSymbol
aas:unitSymbol rdf:type owl:ObjectProperty ;
     rdfs:comment ""@en ;
     rdfs:label "unit has symbol"^^xsd:string ;
     rdfs:domain aas:DataSpecificationPhysicalUnit ;
     rdfs:range xsd:string ;
.
```

```
###  https://admin-shell.io/aas/2/0/rdf#valueFormat
aas:valueFormat rdf:type owl:DatatypeProperty ;
    rdfs:label "has value format"^^xsd:string ;
    rdfs:comment ""@en ;
    rdfs:domain aas:DataSpecificationIEC61360 ;
    rdfs:range xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#valueId
aas:valueId rdf:type owl:ObjectProperty ;
    rdfs:label "has value list"^^xsd:string ;
    rdfs:comment "The Type 'ValueList' lists all the allowed values for a concept description for which the allowed values are listed in an enumeration. The value list is a set of value reference pairs."@en ;
    rdfs:domain aas:DataSpecificationIEC61360 ;
    rdfs:range xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#version
aas:version rdf:type owl:DatatypeProperty ;
     rdfs:subPropertyOf dcterms:hasVersion ;
    rdfs:domain aas:AdministrativeInformation ;
    rdfs:range xsd:string ;
     rdfs:comment "Version of the element."@en ;
     rdfs:label "has version"@en ;
.

#################################################################
#    Classes
#################################################################

###  https://admin-shell.io/aas/2/0/rdf#AssetKind
aas:AssetKind rdf:type owl:Class ;
   rdfs:comment "Enumeration for denoting whether an element is a type or an instance."@en ;
   rdfs:label "Asset Kind"^^xsd:string  ;
   owl:oneOf (aas:ASSET_INSTANCE aas:ASSET_TYPE) ;
   .
```

```
### https://admin-shell.io/aas/2/0/rdf#AnnotatedRelationshipElement

aas:AnnotatedRelationshipElement rdf:type owl:Class ;

    rdfs:subClassOf aas:RelationshipElement ;

   rdfs:comment "An annotated relationship element is an relationship element that can be annotated with additional data elements."@en ;

   rdfs:label "Annotated Relationship Element"^^xsd:string ;

   .


### https://admin-shell.io/aas/2/0/rdf#AccessControl

aas:AccessControl rdf:type owl:Class ;

    rdfs:comment "Access Control defines the local access control policy administration point. Access Control has the major task to define the access permission rules."@en ;

    rdfs:label "Access Control"^^xsd:string ;

.


### https://admin-shell.io/aas/2/0/rdf#AccessControlPolicyPoints

aas:AccessControlPolicyPoints rdf:type owl:Class ;

   rdfs:comment "Container for access control policy points."@en ;

   rdfs:label "Access ControlPolicy Points"^^xsd:string .


### https://admin-shell.io/aas/2/0/rdf#AccessPermissionRule

aas:AccessPermissionRule rdf:type owl:Class ;

    rdfs:subClassOf aas:Referable ;

    rdfs:subClassOf aas:Qualifiable ;

   rdfs:comment "Table that defines access permissions per authenticated subject for a set of objects (referable elements)."@en ;

    rdfs:label "Access Permission Rule"^^xsd:string ;

.


### https://admin-shell.io/aas/2/0/rdf#AdministrativeInformation

aas:AdministrativeInformation rdf:type owl:Class ;

   rdfs:subClassOf aas:HasDataSpecification ;

   rdfs:comment "Every Identifiable may have administrative information. Administrative information includes for example  Information about the version of the element  Information about who created or who made the last change to the element  Information about the languages available in case the element contains text, for translating purposed also themmaster or default language may be definedIn the first version of the AAS metamodel only version information as defined by IEC 61360 is defined. In later versions additional attributes may be added."@en ;

   rdfs:label "Administrative Information"^^xsd:string .


### https://admin-shell.io/aas/2/0/rdf#AssetAdministrationShell
```

```
aas:AssetAdministrationShell rdf:type owl:Class ;
    rdfs:subClassOf aas:HasDataSpecification ,
    aas:Identifiable ;
  owl:disjointWith aas:Asset ;
  owl:disjointWith aas:ConceptDescription ;
  owl:disjointWith aas:Formula ;
  owl:disjointWith aas:Qualifier ;
  owl:disjointWith aas:Submodel ;
  owl:disjointWith aas:SubmodelElement ;
  owl:disjointWith aas:View ;
    rdfs:label "Asset Administration Shell"@en ;
    skos:altLabel "Administration Shell"@en , "Verwaltungsschale"@de ;
    skos:definition "Describes the Administration Shell for Assets, Products, Components, e.g. Machines"@en ;
    rdfs:comment "Describes the Administration Shell for Assets, Products, Components, e.g. Machines"@en ;
    skos:prefLabel "Asset Administration Shell"^^xsd:string ;
    .


###  https://admin-shell.io/aas/2/0/rdf#Asset
aas:Asset rdf:type owl:Class ;
    rdfs:subClassOf aas:HasDataSpecification ,
    aas:Identifiable ;
    skos:altLabel "Object"@en ;
    skos:definition "Clearly identifiable asset for the Administration Shell"@en ;
    skos:prefLabel "Asset"@en ;
    rdfs:label "Asset"@en ;
    skos:definition "Eindeutig identifizierbarer Gegenstand, der aufgrund seiner Bedeutung in der Informationswelt
verwaltet wird"@de ;
    rdfs:comment "An Asset describes meta data of an asset that is represented by an AAS. The asset may either
represent an asset type or an asset instance. The asset has a globally unique identifier plus – if needed – additional
domain specific (proprietary) identifiers."@en ;
    skos:note "Objects may be known in the form of a type or of an instance. An object in the planning phase is known
as a type"@en ;
  owl:disjointWith aas:AdministrativeInformation ;
  owl:disjointWith aas:AssetAdministrationShell ;
  owl:disjointWith aas:ConceptDescription ;
  owl:disjointWith aas:Constraint ;
  owl:disjointWith aas:Submodel ;
  owl:disjointWith aas:SubmodelElement ;
  owl:disjointWith aas:View ;
```

```
    .


###  https://admin-shell.io/aas/2/0/rdf#BasicEvent

aas:BasicEvent rdf:type owl:Class ;

    rdfs:subClassOf aas:Event ;

    rdfs:label "Basic Event"^^xsd:string ;

    rdfs:comment ""@en ;

.


###  https://admin-shell.io/aas/2/0/rdf#Blob

aas:Blob rdf:type owl:Class ;

   rdfs:subClassOf aas:DataElement ;

   rdfs:comment "A BLOB is a data element that represents a file that is contained with its source code in the value attribute."@en ;

   rdfs:label "Blob Data Element"@en .


###  https://admin-shell.io/aas/2/0/rdf#BlobCertificate

aas:BlobCertificate rdf:type owl:Class ;

    rdfs:comment "Certificate provided as BLOB."@en ;

    rdfs:label "Blob Certificate"^^xsd:string ;

    .


###  https://admin-shell.io/aas/2/0/rdf#Capability

aas:Capability rdf:type owl:Class ;

    rdfs:subClassOf aas:SubmodelElement ;

    rdfs:comment "A capability is the implementation-independent description of the potential of an asset to achieve a certain effect in the physical or virtual world."@en ;

    rdfs:label "Capability"^^xsd:string ;

    .


###  https://admin-shell.io/aas/2/0/rdf#Category

aas:Category rdf:type owl:Class ;

    rdfs:comment "A Category specifies the nature of a Property or a MultiLanguageProperty."@en ;

    rdfs:label "Category"^^xsd:string ;

    owl:oneOf (aas:CONSTANT aas:PARAMETER aas:VARIABLE) ;

    .


###  https://admin-shell.io/aas/2/0/rdf#Certificate
```

```
aas:Certificate rdf:type owl:Class ;
    rdfs:comment "A technical certificate proofing the identity through cryptographic measures."@en ;
    rdfs:label "Certificate"^^xsd:string ;
    .


###  https://admin-shell.io/aas/2/0/rdf#ConceptDescription
aas:ConceptDescription rdf:type owl:Class ;
    rdfs:subClassOf aas:HasDataSpecification ,
    aas:Identifiable ;
   owl:disjointWith aas:AdministrativeInformation ;
   owl:disjointWith aas:Asset ;
   owl:disjointWith aas:AssetAdministrationShell ;
   owl:disjointWith aas:Formula ;
   owl:disjointWith aas:Qualifier ;
   owl:disjointWith aas:Submodel ;
   owl:disjointWith aas:SubmodelElement ;
   owl:disjointWith aas:View ;
   rdfs:label "Concept Description"^^xsd:string ;
    rdfs:comment "The AAS itself can also define its own dictionary that contains semantic definitions of its submodel elements. These semantic definitions are called concept descriptions (ConceptDescription). It is optional whether an AAS defines its own concept dictionary (ConceptDictionary) or not."@en ;
    rdfs:comment "The semantics of a property or other elements that may have a semantic description is defined by a concept description. The description of the concept should follow a standardized schema (realized as data specification template)."@en ;
.


###  https://admin-shell.io/aas/2/0/rdf#ConceptDictionary
aas:ConceptDictionary rdf:type owl:Class ;
    rdfs:subClassOf aas:Referable ;
    rdfs:label "Concept Dictionary"^^xsd:string ;
    rdfs:comment "A dictionary contains elements that can be reused. The concept dictionary contains concept descriptions. Typically a concept description dictionary of an AAS contains only concept descriptions of elements used within submodels of the AAS."@en ;
    .


###  https://admin-shell.io/aas/2/0/rdf#Constraint
aas:Constraint  rdf:type owl:Class ;
   dash:abstract true ;
   rdfs:comment "A constraint is used to further qualify an element."@en ;
```

```
    rdfs:label "Constraint"@en ;
    skos:prefLabel "Constraint"@en ;
    owl:disjointWith aas:AdministrativeInformation ;
    owl:disjointWith aas:Asset ;
    owl:disjointWith aas:AssetAdministrationShell ;
    owl:disjointWith aas:DataSpecificationContent ;
    owl:disjointWith aas:Identifier ;
    owl:disjointWith aas:ReferableElement ;
    owl:disjointWith aas:HasKind ;
    owl:disjointWith aas:Qualifiable ;
    owl:disjointWith aas:Reference ;
    owl:disjointWith aas:Submodel ;
 .


###  https://admin-shell.io/aas/2/0/rdf#DataElement
aas:DataElement rdf:type owl:Class ;
   rdfs:subClassOf aas:SubmodelElement ;
    owl:disjointWith aas:Event ;
    owl:disjointWith aas:Operation ;
    owl:disjointWith aas:OperationVariable ;
    owl:disjointWith aas:RelationshipElement ;
    owl:disjointWith aas:SubmodelElementCollection ;
     dash:abstract true ;
   rdfs:comment "A data element is a submodel element that is not further composed out of other submodel elements. A data element is a submodel element that has a value. The type of value differs for different subtypes of data elements."@en ;
   rdfs:label "Data Element"^^xsd:string ;
   .


### https://admin-shell.io/aas/2/0/rdf#DataTypeIEC61360
aas:DataTypeIEC61360 rdf:type owl:Class ;
     rdfs:label "Data Type IEC61360"^^xsd:string ;
     rdfs:comment "Enumeration of all IEC 61360 defined data types."@en ;
     owl:oneOf (
       aas:DATE_IEC6360_DATATYPE
       aas:STRING_IEC6360_DATATYPE
       aas:STRING_TRANSLATABLE_IEC6360_DATATYPE
       aas:REAL_MEASURE_IEC6360_DATATYPE
```

```
        aas:REAL_COUNT_IEC6360_DATATYPE
        aas:REAL_CURRENCY_IEC6360_DATATYPE
        aas:BOOLEAN_IEC6360_DATATYPE
        aas:URL_IEC6360_DATATYPE
        aas:RATIONAL_IEC6360_DATATYPE
        aas:RATIONAL_MEASURE_IEC6360_DATATYPE
        aas:TIME_IEC6360_DATATYPE
        aas:TIMESTAMP_IEC6360_DATATYPE
    ) ;
.

###  https://admin-shell.io/aas/2/0/rdf#DataSpecification
aas:DataSpecification rdf:type owl:Class ;
    rdfs:subClassOf aas:Identifiable ;
    dash:abstract true ;
    owl:disjointWith aas:AdministrativeInformation ;
    owl:disjointWith aas:Asset ;
    owl:disjointWith aas:AssetAdministrationShell ;
    owl:disjointWith aas:ConceptDescription ;
    owl:disjointWith aas:Submodel ;
    owl:disjointWith aas:SubmodelElement ;
    owl:disjointWith aas:View ;
    rdfs:comment "Element that can have data specification templates. A template defines the additional attributes an element may or shall have."@en ;
    rdfs:label "DataSpecification"@en ;
.

###  https://admin-shell.io/aas/2/0/rdf#DataSpecificationContent
aas:DataSpecificationContent rdf:type owl:Class ;
    owl:disjointWith aas:Constraint ;
    owl:disjointWith aas:HasDataSpecification ;
    owl:disjointWith aas:Identifier ;
    owl:disjointWith aas:Key ;
    owl:disjointWith aas:KeyElement ;
    owl:disjointWith aas:HasKind ;
    owl:disjointWith aas:Qualifiable ;
    owl:disjointWith aas:Reference ;
    rdfs:label "Data Specification Content"^^xsd:string ;
```

```
    rdfs:comment ""@en ;
    dash:abstract true ;
.

###  https://admin-shell.io/aas/2/0/rdf#DataSpecificationIEC61360
aas:DataSpecificationIEC61360 rdf:type owl:Class ;
    rdfs:subClassOf aas:DataSpecificationContent ;
    rdfs:label "Data Specification IEC 61360"^^xsd:string ;
    rdfs:comment "Data Specification Template for defining Property Descriptions conformant to IEC 61360."@en ;
.

###  https://admin-shell.io/aas/2/0/rdf#DataSpecificationPhysicalUnit
aas:DataSpecificationPhysicalUnit rdf:type owl:Class ;
    rdfs:subClassOf aas:DataSpecificationContent ;
    rdfs:label "Data Specification Physical Unit"^^xsd:string ;
    rdfs:comment "Data Specification Tempate for Physical Units."@en ;
.

###  https://admin-shell.io/aas/2/0/rdf#Entity
aas:Entity rdf:type owl:Class ;
    rdfs:subClassOf aas:SubmodelElement ;
    rdfs:label "Entity"^^xsd:string ;
    rdfs:comment "An entity is a submodel element that is used to model entities."@en ;
.

###  https://admin-shell.io/aas/2/0/rdf#EntityType
aas:EntityType rdf:type owl:Class ;
    rdfs:label "Entity Type"^^xsd:string ;
    rdfs:comment "Enumeration for denoting whether an entity is a self-managed entity or a co-managed entity."@en
;
    owl:oneOf (aas:CO_MANAGED_ENTITY aas:SELF_MANAGED_ENTITY ) ;
.

###  https://admin-shell.io/aas/2/0/rdf#Event
aas:Event rdf:type owl:Class ;
    rdfs:subClassOf aas:SubmodelElement ;
    rdfs:label "Event"^^xsd:string ;
    rdfs:comment ""@en ;
```

```
.


###  https://admin-shell.io/aas/2/0/rdf#EventElement
aas:Event rdf:type owl:Class ;
     rdfs:subClassOf aas:SubmodelElement ;
     rdfs:label "Event Element"^^xsd:string ;
     rdfs:comment "Defines the necessary information for sending or receiving events."@en ;

###  https://admin-shell.io/aas/2/0/rdf#EventMessage
aas:EventMessage rdf:type owl:Class ;
     rdfs:subClassOf aas:SubmodelElement ;
     rdfs:label "Event Message"^^xsd:string ;
     rdfs:comment "Defines the necessary information of an event instance sent out or received."@en ;
     skos:note "non- normative, just only for discussion (as of November 2019)."@en ;
.

###  https://admin-shell.io/aas/2/0/rdf#File
aas:File rdf:type owl:Class ;
    rdfs:subClassOf aas:SubmodelElement ;
    rdfs:comment "A File is a data element that represents a file via its path description."@en ;
    rdfs:label "File Submodel Element"@en ;
    .

###  https://admin-shell.io/aas/2/0/rdf#Formula
aas:Formula rdf:type owl:Class ;
    rdfs:subClassOf aas:Constraint ;
     dc:description "A formula is used to describe constraints by a logical expression."@en ;
     rdfs:label "Formula"@en ;
.

###  https://admin-shell.io/aas/2/0/rdf#HasDataSpecification
aas:HasDataSpecification rdf:type owl:Class ;
     dash:abstract true ;
    rdfs:comment "Element that can have be extended by using data specification templates. A data specification template defines the additional attributes an element may or shall have. The data specifications used are explicitly specified with their id."@en ;
    rdfs:label "Has Data Specification"@en ;
    .
```

```
###  https://admin-shell.io/aas/2/0/rdf#HasKind

aas:HasKind rdf:type owl:Class ;

    rdfs:comment "An element with a kind is an element that can either represent a type or an instance. Default for an element is that it is representing an instance."@en ;

    rdfs:label "Has Kind"@en ;

    .


###  https://admin-shell.io/aas/2/0/rdf#HasSemantics

aas:HasSemantics rdf:type owl:Class ;

    dash:abstract true ;

    rdfs:label "Has Semantics"^^xsd:string ;

    rdfs:comment "Element that can have a semantic definition. Identifier of the semantic definition of the element. It is called semantic id of the element. The semantic id may either reference an external global id or it may reference a referable model element of kind=Type that defines the semantics of the element."@en ;

    skos:note "In many cases the idShort is identical to the English short name within the semantic definition as referenced vi aits semantic id."@en ;

    .


###  https://admin-shell.io/aas/2/0/rdf#Identifiable

aas:Identifiable rdf:type owl:Class ;

    dash:abstract true ;

   rdfs:subClassOf aas:Referable ;

   rdfs:comment "An element that has a globally unique identifier."@en ;

   rdfs:label "Identifiable"@en ;

   .


### https://admin-shell.io/aas/2/0/rdf#IdentifiableElement

aas:IdentifiableElement rdf:type owl:Class ;

    rdfs:subClassOf aas:ReferableElement ;

    rdfs:label "Identifiable Element"^^xsd:string ;

    rdfs:comment "Enumeration of all identifiable elements within an asset administration shell that are not identifiable"@en ;

    owl:oneOf (

      aas:ASSET_IDENTIFIABLE_ELEMENT

      aas:ASSET_ADMINISTRATION_SHELL_IDENTIFIABLE_ELEMENT

      aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT

      aas:SUBMODEL_IDENTIFIABLE_ELEMENT

    ) ;
```

```
.


###  https://admin-shell.io/aas/2/0/rdf#Identifier
aas:Identifier rdf:type owl:Class ;
    rdfs:comment "Used to uniquely identify an entity by using an identifier."@en ;
    rdfs:label "Identifier"@en ;
    .


### https://admin-shell.io/aas/2/0/rdf#IdentifierType
aas:IdentifierType rdf:type owl:Class ;
     rdfs:subClassOf aas:KeyType ;
     rdfs:label "Identifier Type"^^xsd:string ;
     rdfs:comment "Enumeration of different types of Identifiers for global identification"@en ;
     owl:oneOf (
       aas:IRDI_IDENTIFIER_TYPE
       aas:IRI_IDENTIFIER_TYPE
       aas:CUSTOM_IDENTIFIER_TYPE ) ;
   .


###  https://admin-shell.io/aas/2/0/rdf#Key
aas:Key rdf:type owl:Class ;
   rdfs:comment "A key is a reference to an element by its id."@en ;
   rdfs:label "Key"^^xsd:string .


###  https://admin-shell.io/aas/2/0/rdf#KeyElement
aas:KeyElement rdf:type owl:Class ;
     rdfs:label "Key Elements"^^xsd:string ;
     rdfs:comment "Enumeration of different key value types within a key. Contains KeyElements, ReferableElements,
and IdentifiableElements."@en ;
     owl:oneOf (
       aas:ACCESS_PERMISSION_RULE_REFERABLE_ELEMENT
       aas:ANNOTATED_RELATIONSHIP_ELEMENT_REFERABLE_ELEMENT
       aas:BASIC_EVENT_REFERABLE_ELEMENT
       aas:BLOB_REFERABLE_ELEMENT
       aas:CAPABILITY_REFERABLE_ELEMENT
       aas:CONCEPT_DICTIONARY_REFERABLE_ELEMENT
       aas:GLOBAL_REFERENCE_KEY_ELEMENT
       aas:DATA_ELEMENT_REFERABLE_ELEMENT
```

```
      aas:FILE_REFERABLE_ELEMENT
      aas:ENTITY_REFERABLE_ELEMENT
      aas:EVENT_REFERABLE_ELEMENT
      aas:MULTI_LANGUAGE_PROPERTY_REFERABLE_ELEMENT
      aas:OPERATION_REFERABLE_ELEMENT
      aas:PROPERTY_REFERABLE_ELEMENT
      aas:RANGE_REFERABLE_ELEMENT #
      aas:REFERENCE_ELEMENT_REFERABLE_ELEMENT
      aas:RELATIONSHIPT_ELEMENT_REFERABLE_ELEMENT
      aas:SUBMODEL_ELEMENT_REFERABLE_ELEMENT
      aas:SUBMODEL_ELEMENT_COLLECTION_REFERABLE_ELEMENT
      aas:VIEW_REFERABLE_ELEMENT

      aas:ASSET_IDENTIFIABLE_ELEMENT
      aas:ASSET_ADMINISTRATION_SHELL_IDENTIFIABLE_ELEMENT
      aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT
      aas:SUBMODEL_IDENTIFIABLE_ELEMENT

      aas:GLOBAL_REFERENCE_KEY_ELEMENT
      aas:FRAGMENT_REFERENCE_KEY_ELEMENT
    ) ;
.


### https://admin-shell.io/aas/2/0/rdf#KeyType
aas:KeyType rdf:type owl:Class ;
    rdfs:label "Key Type"^^xsd:string ;
    rdfs:comment "Enumeration of different key value types within a key. Contains IdentifierType and
LocalKeyType."@en ;
    owl:oneOf (
   aas:IRDI_IDENTIFIER_TYPE
   aas:IRI_IDENTIFIER_TYPE
   aas:CUSTOM_IDENTIFIER_TYPE
   aas:IDSHORT_LOCAL_KEY_TYPE
   aas:FRAGMENT_ID_LOCAL_KEY_TYPE
    ) ;
.


### https://admin-shell.io/aas/2/0/rdf#LocalKeyType
```

```
aas:LocalKeyType rdf:type owl:Class ;
     rdfs:subClassOf aas:KeyType ;
     rdfs:label "Local Key Type"^^xsd:string ;
     rdfs:comment "Enumeration of different key value types within a key."@en ;
     owl:oneOf (
    aas:IDSHORT_LOCAL_KEY_TYPE
    aas:FRAGMENT_ID_LOCAL_KEY_TYPE
     ) ;
.

###  https://admin-shell.io/aas/2/0/rdf#ModelingKind
aas:ModelingKind rdf:type owl:Class ;
   rdfs:comment "Enumeration for denoting whether an element is a type or an instance."@en ;
   rdfs:label "Kind"^^xsd:string  ;
   owl:oneOf (aas:INSTANCE aas:TEMPLATE) ;
   .

###  https://admin-shell.io/aas/2/0/rdf#MultiLanguageProperty
aas:MultiLanguageProperty rdf:type owl:Class ;
   rdfs:subClassOf aas:DataElement ;
   rdfs:comment "A property is a data element that has a multi language value."@en ;
   rdfs:label "Multi Language Property"@en ;
   .

###  https://admin-shell.io/aas/2/0/rdf#ObjectAttributes
aas:ObjectAttributes rdf:type owl:Class ;
   rdfs:comment "A set of data elements that describe object attributes. These attributes need to refer to a data element
within an existing submodel."@en ;
   rdfs:label "Object Attributes"^^xsd:string .

###  https://admin-shell.io/aas/2/0/rdf#Operation
aas:Operation rdf:type owl:Class ;
     rdfs:subClassOf aas:SubmodelElement ;
     rdfs:comment "An operation is a submodel element with input and output variables."@en ;
     rdfs:label "Operation"@en ;
   owl:disjointWith aas:DataElement ;
   owl:disjointWith aas:Event ;
   owl:disjointWith aas:OperationVariable ;
```

```
    owl:disjointWith aas:RelationshipElement ;
    owl:disjointWith aas:SubmodelElementCollection ;
    .

### https://admin-shell.io/aas/2/0/rdf#OperationVariable
aas:OperationVariable rdf:type owl:Class ;
     dc:description "An operation variable is a submodel element that is used as input or output variable of an operation."@en ;
      rdfs:label "Operation Variable"^^xsd:string ;
     owl:disjointWith aas:DataElement ;
     owl:disjointWith aas:Event ;
     owl:disjointWith aas:Operation ;
     owl:disjointWith aas:RelationshipElement ;
     owl:disjointWith aas:SubmodelElementCollection ;
      .

### https://admin-shell.io/aas/2/0/rdf#Permission
aas:Permission rdf:type owl:Class ;
      rdfs:comment "Description of a single permission."@en ;
      rdfs:label "Permission"^^xsd:string ;
.

### https://admin-shell.io/aas/2/0/rdf#PermissionKind
aas:PermissionKind rdf:type owl:Class ;
     rdfs:comment "Enumeration of the kind of permissions that is given to the assignment of a permission to a subject."@en ;
      rdfs:label "Permission Kind"^^xsd:string ;
      owl:oneOf (
     aas:ALLOW
     aas:DENY
     aas:NOT_APPLICABLE
     aas:UNDEFINED
      ) ;
.

### https://admin-shell.io/aas/2/0/rdf#PermissionsPerObject
aas:PermissionsPerObject rdf:type owl:Class ;
```

```
    rdfs:comment "Table that defines access permissions for a specified object. The object is any referable element in the AAS. Additionally object attributes can be defined that further specify the kind of object the permissions apply to."@en ;
    rdfs:label "Permission Per Object"^^xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#PolicyAdministrationPoint
aas:PolicyAdministrationPoint rdf:type owl:Class ;
    rdfs:comment "Definition of a security administration point (PDP)."@en ;
    rdfs:label "Policy Administration Point"^^xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#PolicyEnforcementPoint
aas:PolicyEnforcementPoint rdf:type owl:Class ;
    rdfs:comment "Defines the security policy enforcement points (PEP)."@en ;
    rdfs:label "Policy Enforcement Point"^^xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#PolicyDecisionPoint
aas:PolicyDecisionPoint rdf:type owl:Class ;
    rdfs:comment "Defines the security policy decision points (PDP). "@en ;
    rdfs:label "Policy Decision Point"^^xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#PolicyInformationPoints
aas:PolicyInformationPoints rdf:type owl:Class ;
    rdfs:comment "Defines the security policy information points (PIP). Serves as the retrieval source of attributes, or the data required for policy evaluation to provide the information needed by the policy decision point to make the decisions."@en ;
    rdfs:label "Policy Information Points"^^xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#Property
aas:Property rdf:type owl:Class ;
  rdfs:subClassOf aas:DataElement ;
  rdfs:comment "A property is a data element that has a single value."@en ;
  rdfs:label "Property"@en ;
  .
```

```
###  https://admin-shell.io/aas/2/0/rdf#Qualifiable

aas:Qualifiable rdf:type owl:Class ;

    dash:abstract true ;

    rdfs:comment "Additional qualification of a qualifiable element."@en ;

    rdfs:label "Qualifiable"@en .


###  https://admin-shell.io/aas/2/0/rdf#Qualifier

aas:Qualifier rdf:type owl:Class ;

   rdfs:subClassOf aas:Constraint ;

   rdfs:subClassOf aas:HasSemantics ;

   owl:disjointWith aas:AdministrativeInformation ;

   owl:disjointWith aas:Asset ;

   owl:disjointWith aas:AssetAdministrationShell ;

   owl:disjointWith aas:ConceptDescription ;

   owl:disjointWith aas:Formula ;

   owl:disjointWith aas:Submodel ;

   owl:disjointWith aas:SubmodelElement ;

   owl:disjointWith aas:View ;

    rdfs:comment "A qualifier is a type-value pair that makes additional statements w.r.t. the value of the element."@en
;

    rdfs:label "Qualifier"@en .


###  https://admin-shell.io/aas/2/0/rdf#Range

aas:Range rdf:type owl:Class ;

    rdfs:subClassOf aas:DataElement ;

    rdfs:comment "An element that is referable by its idShort. This id is not globally unique. This id is unique within
the name space of the element."@en ;

    rdfs:label "Range"@en ;

    .


###  https://admin-shell.io/aas/2/0/rdf#Referable

aas:Referable rdf:type owl:Class ;

    dash:abstract true ;

    rdfs:comment "An element that is referable by its idShort. This id is not globally unique. This id is unique within
the name space of the element."@en ;

    rdfs:label "Referable"@en ;

    .
```

```
### https://admin-shell.io/aas/2/0/rdf#ReferableElement

aas:ReferableElement rdf:type owl:Class ;

    rdfs:subClassOf aas:KeyElement ;

    rdfs:label "Referable Elements"^^xsd:string ;

    rdfs:comment "Enumeration of all referable elements within an asset administration shell. Contains IdentifiableElements"@en ;

    owl:oneOf (

      aas:ACCESS_PERMISSION_RULE_REFERABLE_ELEMENT

      aas:ANNOTATED_RELATIONSHIP_ELEMENT_REFERABLE_ELEMENT

      aas:BASIC_EVENT_REFERABLE_ELEMENT

      aas:BLOB_REFERABLE_ELEMENT

      aas:CAPABILITY_REFERABLE_ELEMENT

      aas:CONCEPT_DICTIONARY_REFERABLE_ELEMENT

      aas:GLOBAL_REFERENCE_KEY_ELEMENT

      aas:DATA_ELEMENT_REFERABLE_ELEMENT

      aas:FILE_REFERABLE_ELEMENT

      aas:ENTITY_REFERABLE_ELEMENT

      aas:EVENT_REFERABLE_ELEMENT

      aas:MULTI_LANGUAGE_PROPERTY_REFERABLE_ELEMENT

      aas:OPERATION_REFERABLE_ELEMENT

      aas:PROPERTY_REFERABLE_ELEMENT

      aas:RANGE_REFERABLE_ELEMENT #

      aas:REFERENCE_ELEMENT_REFERABLE_ELEMENT

      aas:RELATIONSHIPT_ELEMENT_REFERABLE_ELEMENT

      aas:SUBMODEL_ELEMENT_REFERABLE_ELEMENT

      aas:SUBMODEL_ELEMENT_COLLECTION_REFERABLE_ELEMENT

      aas:VIEW_REFERABLE_ELEMENT


      aas:ASSET_IDENTIFIABLE_ELEMENT

      aas:ASSET_ADMINISTRATION_SHELL_IDENTIFIABLE_ELEMENT

      aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT

      aas:SUBMODEL_IDENTIFIABLE_ELEMENT

    ) ;

.


### https://admin-shell.io/aas/2/0/rdf#Reference

aas:Reference rdf:type owl:Class ;
```

```
    rdfs:comment "Reference to either a model element of the same or another AAs or to an external entity. A reference is an ordered list of keys, each key referencing an element. The complete list of keys may for example be concatenated to a path that then gives unique access to an element or entity."@en ;
    rdfs:label "Reference"^^xsd:string ;
    .

###  https://admin-shell.io/aas/2/0/rdf#ReferenceElement
aas:ReferenceElement rdf:type owl:Class ;
    rdfs:subClassOf aas:DataElement ;
    rdfs:comment "A reference element is a data element that defines a logical reference to another element within the same or another AAS or a reference to an external object or entity."@en ;
    rdfs:label "Reference Element"^^xsd:string ;
    .

###  https://admin-shell.io/aas/2/0/rdf#RelationshipElement
aas:RelationshipElement rdf:type owl:Class ;
   rdfs:subClassOf aas:SubmodelElement ;
   dc:description "A relationship element is used to define a relationship between two referable elements."@en ;
   rdfs:label "Relationship Element"@en ;
  owl:disjointWith aas:DataElement ;
  owl:disjointWith aas:Event ;
  owl:disjointWith aas:Operation ;
  owl:disjointWith aas:OperationVariable ;
  owl:disjointWith aas:SubmodelElementCollection ;
  .

###  https://admin-shell.io/aas/2/0/rdf#Security
aas:Security rdf:type owl:Class ;
   rdfs:comment "Container for security relevant information of the AAS."@en ;
   rdfs:label "Security"^^xsd:string .

###  https://admin-shell.io/aas/2/0/rdf#SubjectAttributes
aas:SubjectAttributes  rdf:type owl:Class ;
     rdfs:comment "A set of data elements that further classifies a specific subject."@en ;
     rdfs:label "Subject Attributes"^^xsd:string ;
.

###  https://admin-shell.io/aas/2/0/rdf#Submodel
```

```
aas:Submodel rdf:type owl:Class ;
    rdfs:subClassOf aas:HasDataSpecification ;
    rdfs:subClassOf aas:HasKind ;
    rdfs:subClassOf aas:HasSemantics ;
    rdfs:subClassOf aas:Identifiable ;
    rdfs:subClassOf aas:Qualifiable ;
    rdfs:comment "A Submodel defines a specific aspect of the asset represented by the AAS. A submodel is used to structure the virtual representation and technical functionality of an Administration Shell into distinguishable parts. Each submodel refers to a well-defined domain or subject matter. Submodels can become standardized and thus become submodels types. Submodels can have different life-cycles."@en ,
      "Describe the different types of Data related to the I4.0 Asset"@en ;
    rdfs:label "Submodel"@en ;
    owl:disjointWith aas:AdministrativeInformation ;
    owl:disjointWith aas:Asset ;
    owl:disjointWith aas:AssetAdministrationShell ;
    owl:disjointWith aas:ConceptDescription ;
    owl:disjointWith aas:Formula ;
    owl:disjointWith aas:Qualifier ;
    owl:disjointWith aas:SubmodelElement ;
    owl:disjointWith aas:View ;
  .


### https://admin-shell.io/aas/2/0/rdf#SubmodelElement
aas:SubmodelElement rdf:type owl:Class ;
    rdfs:subClassOf aas:HasDataSpecification ;
    rdfs:subClassOf aas:HasKind ;
    rdfs:subClassOf aas:HasSemantics ;
    rdfs:subClassOf aas:Qualifiable ;
    rdfs:subClassOf aas:Referable ;
    dash:abstract true ;
    rdfs:comment "A submodel element is an element suitable for the description and differentiation of assets."@en ;
    rdfs:label "Submodel Element"^^xsd:string ;
    skos:note "The concept of type and instance applies to submodel elements. Properties are special submodel elements. The property types are defined in dictionaries (like the IEC Common Data Dictionary or eCl@ss), they do not have a value. The property type (kind=Type) is also called data element type in some standards. The property instances (kind=Instance) typically have a value. A property instance is also called property-value pair in certain standards."@en ;
    owl:disjointWith aas:AdministrativeInformation ;
    owl:disjointWith aas:Asset ;
    owl:disjointWith aas:AssetAdministrationShell ;
```

```
    owl:disjointWith aas:ConceptDescription ;
    owl:disjointWith aas:Formula ;
    owl:disjointWith aas:Qualifier ;
    owl:disjointWith aas:Submodel ;
    owl:disjointWith aas:View ;
  .


###  https://admin-shell.io/aas/2/0/rdf#SubmodelElementCollection
aas:SubmodelElementCollection rdf:type owl:Class ;
    rdfs:subClassOf aas:SubmodelElement ;
    owl:disjointWith aas:DataElement ;
    owl:disjointWith aas:Event ;
    owl:disjointWith aas:Operation ;
    owl:disjointWith aas:OperationVariable ;
    owl:disjointWith aas:RelationshipElement ;
    rdfs:comment "A submodel element collection is a set or list of submodel elements."@en ;
    rdfs:label "Submodel Element Collection"@en ;
    .


###  https://admin-shell.io/aas/2/0/rdf#View
aas:View rdf:type owl:Class ;
  rdfs:subClassOf aas:HasDataSpecification ,
  aas:Referable ,
  aas:HasSemantics ;
  owl:disjointWith aas:AdministrativeInformation ;
  owl:disjointWith aas:Asset ;
  owl:disjointWith aas:AssetAdministrationShell ;
  owl:disjointWith aas:ConceptDescription ;
  owl:disjointWith aas:Formula ;
  owl:disjointWith aas:Qualifier ;
  owl:disjointWith aas:Submodel ;
  owl:disjointWith aas:SubmodelElement ;
  rdfs:comment "Different views associated to the Administration Shell via the Submodels"@en ;
  rdfs:comment "A view is a collection of referable elements w.r.t. to a specific viewpoint of one or more
stakeholders."@en ;
  rdfs:isDefinedBy  "https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/hm-2018-trilaterale-
coop.html"@de ;
  rdfs:label "Sicht"@de ,
```

```
    "View"@en .



#################################################################
#    Individuals
#################################################################


###  https://admin-shell.io/aas/2/0/rdf#INSTANCE
aas:INSTANCE rdf:type owl:NamedIndividual ,
    aas:ModelingKind ;
    rdfs:comment "Concrete, clearly identifiable component of a certain template."@en ;
    skos:note "It becomes an individual entity of a template, for example a device model, by defining specific property values."@en ;
    skos:note "In an object oriented view, an instance denotes an object (of a template) (class)."@en ;
    rdfs:label "Instance"@en ;
    owl:differentFrom aas:ASSET_INSTANCE ;
    owl:differentFrom aas:TEMPLATE ;
.


###  https://admin-shell.io/aas/2/0/rdf#TYPE
aas:TEMPLATE rdf:type owl:NamedIndividual ,
     aas:ModelingKind ;
    rdfs:comment "Software element which specifies the common attributes shared by all instances of the template."@en ;
    rdfs:label "Template"@en ;
    owl:differentFrom aas:ASSET_TYPE ;
    owl:differentFrom aas:INSTANCE ;
.


###  https://admin-shell.io/aas/2/0/rdf#ASSET_INSTANCE
aas:ASSET_INSTANCE rdf:type owl:NamedIndividual ,
    aas:AssetKind ;
    rdfs:comment "Concrete, clearly identifiable component of a certain type."@en ;
    rdfs:label "Asset Instance"@en ;
    owl:differentFrom aas:INSTANCE ;
    owl:differentFrom aas:ASSET_TYPE ;
.
```

```
### https://admin-shell.io/aas/2/0/rdf#ASSET_TYPE

aas:ASSET_TYPE rdf:type owl:NamedIndividual ,

    aas:AssetKind ;

    rdfs:comment "hardware or software element which specifies the common attributes shared by all instances of the type."@en ;

    rdfs:label "Asset Type"@en ;

    owl:differentFrom aas:INSTANCE ;

    owl:differentFrom aas:ASSET_INSTANCE ;

.



### https://admin-shell.io/aas/2/0/rdf#CONSTANT

aas:CONSTANT rdf:type owl:NamedIndividual ,

     aas:Category ;

    rdfs:comment "A constant property is a property with a value that does not change over time. In eCl@ss this kind of category has the category 'Coded Value'."@en ;

    rdfs:label "Constant"@en ;

.


### https://admin-shell.io/aas/2/0/rdf#PARAMETER

aas:PARAMETER rdf:type owl:NamedIndividual ,

    aas:Category ;

    rdfs:comment "A parameter property is a property that is once set and then typically does not change over time. This is for example the case for configuration parameters."@en ;

    rdfs:label "Parameter"@en ;

.


### https://admin-shell.io/aas/2/0/rdf#VARIABLE

aas:VARIABLE rdf:type owl:NamedIndividual ,

    aas:Category ;

    rdfs:comment "A variable property is a property that is calculated during runtime, i.e. its value is a runtime value."@en ;

    rdfs:label "Variable"@en ;

.



### https://admin-shell.io/aas/2/0/rdf#CO_MANAGED_ENTITY

aas:CO_MANAGED_ENTITY rdf:type owl:NamedIndividual ,

    aas:EntityType ;
```

```
    rdfs:comment "For co-managed entities there is no separate AAS. Co-managed entities need to be part of a self-managed entity."@en ;
    rdfs:label "Co-managed Entity"@en ;
.


###  https://admin-shell.io/aas/2/0/rdf#SELF_MANAGED_ENTITY
aas:SELF_MANAGED_ENTITY rdf:type owl:NamedIndividual ,
    aas:EntityType ;
    rdfs:comment "Self-Managed Entities have their own AAS but can be part of the bill of material of a composite self-managed entity. The asset of an I4.0 Component is a self-managed entity per definition."@en ;
    rdfs:label "Self-managed Entity"@en ;
.


###  https://admin-shell.io/aas/2/0/rdf#GLOBAL_REFERENCE_KEY_ELEMENT
aas:GLOBAL_REFERENCE_KEY_ELEMENT rdf:type aas:KeyElement ;
    rdfs:label "Gobal Reference"@en ;
    rdfs:comment "reference to an element not belonging to an asset administration shell"@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#FRAGMENT_REFERENCE_KEY_ELEMENT
aas:FRAGMENT_REFERENCE_KEY_ELEMENT rdf:type aas:KeyElement ;
    rdfs:label "Fragement Reference"@en ;
    rdfs:comment ""@en ;
.


###  https://admin-shell.io/aas/2/0/rdf#ACCESS_PERMISSION_RULE_REFERABLE_ELEMENT
aas:ACCESS_PERMISSION_RULE_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "Access Permission Rule"@en ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#ANNOTATED_RELATIONSHIP_ELEMENT_REFERABLE_ELEMENT
aas:ANNOTATED_RELATIONSHIP_ELEMENT_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "Annotated relationship element"@en ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#BASIC_EVENT_REFERABLE_ELEMENT
aas:BASIC_EVENT_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "Basic Event"@en ;
    rdfs:comment ""@en ;
```

```
.
###  https://admin-shell.io/aas/2/0/rdf#BLOB_REFERABLE_ELEMENT
aas:BLOB_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "Blob"@en ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#CAPABILITY_REFERABLE_ELEMENT
aas:CAPABILITY_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "Capability"@en ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#CONCEPT_DICTIONARY_REFERABLE_ELEMENT
aas:CONCEPT_DICTIONARY_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "Concept Dictionary"@en ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#DATA_ELEMENT_REFERABLE_ELEMENT
aas:DATA_ELEMENT_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "Data Element"@en ;
    rdfs:comment ""@en ;
    skos:note "Data Element is abstract, i.e. if a key uses 'DataElement' the reference may be a Property, a File etc."@en
;
.
###  https://admin-shell.io/aas/2/0/rdf#ENTITY_REFERABLE_ELEMENT
aas:ENTITY_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "Entity"@en ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#EVENT_REFERABLE_ELEMENT
aas:EVENT_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "Event"@en ;
    rdfs:comment ""@en ;
    skos:note "Event is abstract"@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#MULTI_LANGUAGE_PROPERTY_REFERABLE_ELEMENT
aas:MULTI_LANGUAGE_PROPERTY_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "Multi-language Property"@en ;
    rdfs:comment "Property with a value that can be provided in multiple languages."@en ;
```

```
.
###  https://admin-shell.io/aas/2/0/rdf#OPERATION_REFERABLE_ELEMENT
aas:OPERATION_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "Operation"@en ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#PROPERTY_REFERABLE_ELEMENT
aas:PROPERTY_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "Property"@en ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#RANGE_REFERABLE_ELEMENT
aas:RANGE_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "Range"@en ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#REFERENCE_ELEMENT_REFERABLE_ELEMENT
aas:REFERENCE_ELEMENT_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "Reference Element"@en ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#RELATIONSHIPT_ELEMENT_REFERABLE_ELEMENT
aas:RELATIONSHIPT_ELEMENT_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "Relationship Element"@en ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#SUBMODEL_ELEMENT_REFERABLE_ELEMENT
aas:SUBMODEL_ELEMENT_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "SubmodelElement"@en ;
    rdfs:comment ""@en ;
    skos:note "Submodel Element is abstract, i.e. if a key uses “SubmodelElement” the reference may be a Property, a
SubmodelElementCollection, an Operation etc."@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#SUBMODEL_ELEMENT_COLLECTION_REFERABLE_ELEMENT
aas:SUBMODEL_ELEMENT_COLLECTION_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "Submodel Element Collection"@en ;
    rdfs:comment "Collection of Submodel Elements"@en ;
.
```

```
###  https://admin-shell.io/aas/2/0/rdf#VIEW_REFERABLE_ELEMENT
aas:VIEW_REFERABLE_ELEMENT rdf:type aas:ReferableElement ;
    rdfs:label "View"@en ;
    rdfs:comment ""@en ;
.

###  https://admin-shell.io/aas/2/0/rdf#ASSET_IDENTIFIABLE_ELEMENT
aas:ASSET_IDENTIFIABLE_ELEMENT rdf:type aas:IdentifiableElement ;
    rdfs:label "Asset"@en ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#ASSET_ADMINISTRATION_SHELL_IDENTIFIABLE_ELEMENT
aas:ASSET_ADMINISTRATION_SHELL_IDENTIFIABLE_ELEMENT rdf:type aas:IdentifiableElement ;
    rdfs:label "Asset Administration Shell"@en ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT
aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT rdf:type aas:IdentifiableElement ;
    rdfs:label "Concept Description"@en ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#SUBMODEL_IDENTIFIABLE_ELEMENT
aas:SUBMODEL_IDENTIFIABLE_ELEMENT rdf:type aas:IdentifiableElement ;
    rdfs:label "Submodel"@en ;
    rdfs:comment ""@en ;
.

###  https://admin-shell.io/aas/2/0/rdf#IRDI_IDENTIFIER_TYPE
aas:IRDI_IDENTIFIER_TYPE rdf:type aas:IdentifierType ;
    rdfs:label "IRDI"@en ;
    rdfs:comment "IRDI according to ISO29002-5 as an Identifier scheme for properties and classifications."@en ;
    rdfs:subClassOf [
    rdf:type owl:Restriction ;
     owl:allValuesFrom xsd:string ;
     owl:onProperty aas:value ;
    ] ;
.
```

```
### https://admin-shell.io/aas/2/0/rdf#IRI_IDENTIFIER_TYPE
aas:IRI_IDENTIFIER_TYPE rdf:type aas:IdentifierType ;
    rdfs:label "IRI"@en ;
    rdfs:comment "IRI. Should only be used if unicode symbols are used that are not allowed in URI."@en ;
.
### https://admin-shell.io/aas/2/0/rdf#CUSTOM_IDENTIFIER_TYPE
aas:CUSTOM_IDENTIFIER_TYPE rdf:Type aas:IdentifierType ;
    rdfs:label "Custom"@en ;
    rdfs:comment "Custom identifiers like GUIDs (globally unique Identifiers)"@en ;
.
### https://admin-shell.io/aas/2/0/rdf#IDSHORT_LOCAL_KEY_TYPE
aas:IDSHORT_LOCAL_KEY_TYPE rdf:type aas:LocalKeyType ;
    rdfs:label "IdShort"@en ;
    rdfs:comment "idShort of a referable element"@en ;
.
### https://admin-shell.io/aas/2/0/rdf#FRAGMENT_ID_LOCAL_KEY_TYPE
aas:FRAGMENT_ID_LOCAL_KEY_TYPE rdf:Type aas:LocalKeyType ;
    rdfs:label "FragementId"@en ;
    rdfs:comment "Identifier of a fragment within a file"@en ;
.

### https://admin-shell.io/aas/2/0/rdf#ALLOW
aas:ALLOW rdf:type aas:PermissionKind ;
     rdfs:label "allow"^^xsd:string ;
     rdfs:comment "Allow the permission given to the subject."@en ;
.
### https://admin-shell.io/aas/2/0/rdf#DENY
aas:DENY rdf:type aas:PermissionKind ;
     rdfs:label "deny"^^xsd:string ;
     rdfs:comment "Explicitly deny the permission given to the subject."@en ;
.
### https://admin-shell.io/aas/2/0/rdf#NOT_APPLICABLE
aas:NOT_APPLICABLE rdf:type aas:PermissionKind ;
     rdfs:label "not applicable"^^xsd:string ;
     rdfs:comment "The permission is not applicable to the subject."@en ;
.
### https://admin-shell.io/aas/2/0/rdf#UNDEFINED
```

```
aas:UNDEFINED rdf:type aas:PermissionKind ;
    rdfs:label "undefined"^^xsd:string ;
    rdfs:comment "It is undefined whether the permission is allowed, not applicable or denied to the subject."@en ;
.

### https://admin-shell.io/aas/2/0/rdf#DATE_IEC6360_DATATYPE
aas:DATE_IEC6360_DATATYPE rdf:type aas:DataTypeIEC61360 ;
    rdfs:label "date according to IEC61360"^^xsd:string ;
    rdfs:seeAlso xsd:date ;
    rdfs:comment ""@en ;
.
### https://admin-shell.io/aas/2/0/rdf#BOOLEAN_IEC6360_DATATYPE
aas:BOOLEAN_IEC6360_DATATYPE rdf:type aas:DataTypeIEC61360 ;
    rdfs:label "boolean according to IEC61360"^^xsd:string ;
    rdfs:seeAlso xsd:boolean ;
    rdfs:comment ""@en ;
.
### https://admin-shell.io/aas/2/0/rdf#REAL_CURRENCY_IEC6360_DATATYPE
aas:REAL_CURRENCY_IEC6360_DATATYPE rdf:type aas:DataTypeIEC61360 ;
    rdfs:label "real currency according to IEC61360"^^xsd:string ;
    rdfs:comment ""@en ;
.
### https://admin-shell.io/aas/2/0/rdf#REAL_COUNT_IEC6360_DATATYPE
aas:REAL_COUNT_IEC6360_DATATYPE rdf:type aas:DataTypeIEC61360 ;
    rdfs:label "real count according to IEC61360"^^xsd:string ;
    rdfs:comment ""@en ;
.
### https://admin-shell.io/aas/2/0/rdf#REAL_MEASURE_IEC6360_DATATYPE
aas:REAL_MEASURE_IEC6360_DATATYPE rdf:type aas:DataTypeIEC61360 ;
    rdfs:label "real measure according to IEC61360"^^xsd:string ;
    rdfs:comment ""@en ;
.
### https://admin-shell.io/aas/2/0/rdf#STRING_IEC6360_DATATYPE
aas:STRING_IEC6360_DATATYPE rdf:type aas:DataTypeIEC61360 ;
    rdfs:label "string according to IEC61360"^^xsd:string ;
    rdfs:seeAlso xsd:string ;
    rdfs:comment ""@en ;
```

```
.
###  https://admin-shell.io/aas/2/0/rdf#STRING_TRANSLATABLE_IEC6360_DATATYPE
aas:STRING_TRANSLATABLE_IEC6360_DATATYPE rdf:type aas:DataTypeIEC61360 ;
    rdfs:label "translatable string according to IEC61360"^^xsd:string ;
    rdfs:seeAlso xsd:langString ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#RATIONAL_MEASURE_IEC6360_DATATYPE
aas:RATIONAL_MEASURE_IEC6360_DATATYPE rdf:type aas:DataTypeIEC61360 ;
    rdfs:label "retional measure according to IEC61360"^^xsd:string ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#RATIONAL_IEC6360_DATATYPE
aas:RATIONAL_IEC6360_DATATYPE rdf:type aas:DataTypeIEC61360 ;
    rdfs:label "retional according to IEC61360"^^xsd:string ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#TIME_IEC6360_DATATYPE
aas:TIME_IEC6360_DATATYPE rdf:type aas:DataTypeIEC61360 ;
    rdfs:label "time according to IEC61360"^^xsd:string ;
    rdfs:comment ""@en ;
.
###  https://admin-shell.io/aas/2/0/rdf#TIMESTAMP_IEC6360_DATATYPE
aas:TIMESTAMP_IEC6360_DATATYPE rdf:type aas:DataTypeIEC61360 ;
    rdfs:label "time stamp according to IEC61360"^^xsd:string ;
    rdfs:comment ""@en ;
    rdfs:seeAlso xsd:dateTime ;
.
###  https://admin-shell.io/aas/2/0/rdf#URL_IEC6360_DATATYPE
aas:URL_IEC6360_DATATYPE rdf:type aas:DataTypeIEC61360 ;
    rdfs:label "url according to IEC61360"^^xsd:string ;
    rdfs:comment ""@en ;
.



###  https://admin-shell.io/aas/2/0/rdf#accessControlPolicyPoints
aas:accessControlPolicyPoints rdf:type owl:ObjectProperty ;
```

```
    rdfs:comment "Access control policy points of the AAS."@en ;
    rdfs:label "has access control policy points"^^xsd:string ;
    rdfs:domain aas:Security ;
    rdfs:range aas:PolicyAdministrationPoint ;
.

###  https://admin-shell.io/aas/2/0/rdf#policyAdministrationPoint
aas:policyAdministrationPoint rdf:type owl:ObjectProperty ;
    rdfs:comment "The access control administration policy point of the AAS."@en ;
    rdfs:label "has policy administration point"^^xsd:string ;
    rdfs:domain aas:AccessControlPolicyPoints ;
    rdfs:range aas:PolicyAdministrationPoint ;
.

###  https://admin-shell.io/aas/2/0/rdf#policyDecisionPoint
aas:policyDecisionPoint rdf:type owl:ObjectProperty ;
    rdfs:comment "The access control policy decision point of the AAS."@en ;
    rdfs:label "has policy decision point"^^xsd:string ;
    rdfs:domain aas:AccessControlPolicyPoints ;
    rdfs:range aas:PolicyDecisionPoint ;
.

###  https://admin-shell.io/aas/2/0/rdf#policyEnforcementPoint
aas:policyEnforcementPoint rdf:type owl:ObjectProperty ;
    rdfs:comment "The access control policy enforcement point of the AAS."@en ;
    rdfs:label "has policy enforcement point"^^xsd:string ;
    rdfs:domain aas:AccessControlPolicyPoints ;
    rdfs:range aas:PolicyEnforcementPoint ;
.

###  https://admin-shell.io/aas/2/0/rdf#policyInformationPoints
aas:policyInformationPoints rdf:type owl:ObjectProperty ;
    rdfs:comment "The access control policy information points of the AAS."@en ;
    rdfs:label "has policy information points"^^xsd:string ;
    rdfs:domain aas:AccessControlPolicyPoints ;
    rdfs:range aas:PolicyInformationPoints ;
.
```

```
###  https://admin-shell.io/aas/2/0/rdf#requiredCertificateExtension

aas:requiredCertificateExtension rdf:type owl:ObjectProperty ;

    rdfs:comment "Certificate extensions as required by the AAS."@en ;

    rdfs:label "has required certificate extension"^^xsd:string ;

    rdfs:domain aas:Security ;

    rdfs:range aas:Reference ;

.


###  https://admin-shell.io/aas/2/0/rdf#localAccessControl

aas:localAccessControl rdf:type owl:ObjectProperty ;

    rdfs:comment "The policy administration point of access control as realized by the AAS itself."@en ;

    skos:note "Constraint AASd-009: Either there is an external policy administration point endpoint defined or the AAS has its own access control."@en ;

    rdfs:label "has local access control"^^xsd:string ;

    rdfs:domain aas:PolicyAdministrationPoint ;

    rdfs:range aas:AccessControl ;

.


###  https://admin-shell.io/aas/2/0/rdf#externalAccessControl

aas:externalAccessControl rdf:type owl:ObjectProperty ;

    rdfs:comment "Endpoint to an external access control defining a policy administration point to be used by the AAS."@en ;

    rdfs:label "has external access control"^^xsd:string ;

    rdfs:domain aas:PolicyAdministrationPoint ;

    rdfs:range aas:Endpoint ;

.


###  https://admin-shell.io/aas/2/0/rdf#externalInformationPoints

aas:externalInformationPoints rdf:type owl:ObjectProperty ;

    rdfs:comment "If externalInformationPoints True then at least one Endpoint to external available information needs to be configured for the AAS."@en ;

    rdfs:label "has external information point"^^xsd:string ;

    rdfs:domain aas:PolicyInformationPoints ;

    rdfs:range aas:Endpoint ;

.


###  https://admin-shell.io/aas/2/0/rdf#internalInformationPoint
```

```
aas:internalInformationPoint rdf:type owl:ObjectProperty ;
    rdfs:comment "References to submodels defining information used by security access permission rules."@en ;
    rdfs:label "has internal information point"^^xsd:string ;
    rdfs:domain aas:PolicyInformationPoints ;
    rdfs:range aas:Submodel ;
.

### https://admin-shell.io/aas/2/0/rdf#accessPermissionRule
aas:accessPermissionRule rdf:type owl:ObjectProperty ;
    rdfs:comment "Access permission rules of the AAS describing the rights assigned to (already authenticated) subjects to access elements of the AAS."@en ;
    rdfs:label "has access permission rule"^^xsd:string ;
    rdfs:domain aas:AccessControl ;
    rdfs:range aas:AccessPermissionRule ;
.

### https://admin-shell.io/aas/2/0/rdf#selectableSubjectAttributes
aas:selectableSubjectAttributes rdf:type owl:ObjectProperty ;
    rdfs:comment "Reference to a submodel defining the authenticated subjects that are configured for the AAS. They are selectable by the access permission rules to assign permissions to the subjects."@en ;
    rdfs:label "has selectable subject attributes"^^xsd:string ;
    skos:note "Default: reference to the submodel referenced via defaultSubjectAttributes."@en ;
    rdfs:domain aas:AccessControl ;
    rdfs:range aas:Submodel ;
.

### https://admin-shell.io/aas/2/0/rdf#defaultSubjectAttributes
aas:defaultSubjectAttributes rdf:type owl:ObjectProperty ;
    rdfs:comment "Reference to a submodel defining the default subjects attributes for the AAS that can be used to describe access permission rules."@en ;
    rdfs:label "has default subject attributes"^^xsd:string ;
    skos:note "The submodel is of kind=Type."@en ;
    rdfs:domain aas:AccessControl ;
    rdfs:range aas:Submodel ;
.

### https://admin-shell.io/aas/2/0/rdf#selectablePermissions
aas:selectablePermissions rdf:type owl:ObjectProperty ;
```

```
    rdfs:comment "Reference to a submodel defining which permissions can be assigned to the subjects."@en ;
    rdfs:label "has selectable permissions"^^xsd:string ;
    skos:note "Default: reference to the submodel referenced via defaultPermissions"@en ;
    rdfs:domain aas:AccessControl ;
    rdfs:range aas:Submodel ;
.

###  https://admin-shell.io/aas/2/0/rdf#defaultPermissions
aas:defaultPermissions rdf:type owl:ObjectProperty ;
    rdfs:comment "Reference to a submodel defining the default permissions for the AAS."@en ;
    rdfs:label "has default permissions"^^xsd:string ;
    rdfs:domain aas:AccessControl ;
    rdfs:range aas:Submodel ;
.

###  https://admin-shell.io/aas/2/0/rdf#selectableEnvironmentAttributes
aas:selectableEnvironmentAttributes rdf:type owl:ObjectProperty ;
    rdfs:comment "Reference to a submodel defining which environment attributes can be accessed via the permission rules."@en ;
    rdfs:label "has selectable environment attributes"^^xsd:string ;
    rdfs:domain aas:AccessControl ;
    rdfs:range aas:Submodel ;
.

###  https://admin-shell.io/aas/2/0/rdf#defaultEnvironmentAttributes
aas:defaultEnvironmentAttributes rdf:type owl:ObjectProperty ;
    rdfs:comment "Reference to a submodel defining default environment attributes, i.e. attributes that are not describing the asset itself. The submodel is of kind=Type. At the same type the values of these environment attributes need to be accessible when evaluating the access permission rules. This is realized as a policy information point."@en ;
    rdfs:label "has default environment attributes"^^xsd:string ;
    rdfs:domain aas:AccessControl ;
    rdfs:range aas:Submodel ;
.

###  https://admin-shell.io/aas/2/0/rdf#targetSubjectAttributes
aas:targetSubjectAttributes rdf:type owl:ObjectProperty ;
    rdfs:comment "Target subject attributes that need to be fulfilled by the accessing subject to get the permissions defined by this rule."@en ;
```

```
    rdfs:label "has target subject attributes"^^xsd:string ;
    rdfs:domain aas:AccessPermissionRule ;
    rdfs:range aas:SubjectAttributes ;
.

###  https://admin-shell.io/aas/2/0/rdf#permissionsPerObject
aas:permissionsPerObject rdf:type owl:ObjectProperty ;
    rdfs:comment "Set of object-permission pairs that define the permissions per object within the access permission rule."@en ;
    rdfs:label "has permissions per object"^^xsd:string ;
    rdfs:domain aas:AccessPermissionRule ;
    rdfs:range aas:PermissionsPerObject ;
.

###  https://admin-shell.io/aas/2/0/rdf#object
aas:object rdf:type owl:ObjectProperty ;
    rdfs:comment "Element to which permission shall be assigned."@en ;
    rdfs:label "has object"^^xsd:string ;
    rdfs:domain aas:PermissionsPerObject ;
    rdfs:range aas:Referable ;
.

###  https://admin-shell.io/aas/2/0/rdf#targetObjectAttributes
aas:targetObjectAttributes rdf:type owl:ObjectProperty ;
    rdfs:comment "Target object attributes that need to be fulfilled so that the access permissions apply to the accessing subject."@en ;
    rdfs:label "has target object attributes"^^xsd:string ;
    rdfs:domain aas:PermissionsPerObject ;
    rdfs:range aas:ObjectAttributes ;
.

###  https://admin-shell.io/aas/2/0/rdf#objectAttribute
aas:objectAttribute rdf:type owl:ObjectProperty ;
    rdfs:comment "A data elements that further classifies an object."@en ;
    rdfs:label "has object attribute"^^xsd:string ;
    rdfs:domain aas:ObjectAttributes ;
    rdfs:range aas:DataElement ;
.
```

```
###  https://admin-shell.io/aas/2/0/rdf#objectPermission

aas:objectPermission rdf:type owl:ObjectProperty ;

    rdfs:comment "Permissions assigned to the object. The permissions hold for all subjects as specified in the access permission rule."@en ;

    rdfs:label "has object permission"^^xsd:string ;

    rdfs:domain aas:PermissionsPerObject ;

    rdfs:range aas:Permission ;

.


###  https://admin-shell.io/aas/2/0/rdf#permission

aas:permission rdf:type owl:ObjectProperty ;

    rdfs:comment "Reference to a property that defines the semantics of the permission."@en ;

    skos:note "Constraint AASd-010: The property has the category CONSTANT."@en ;

    skos:note "Constraint AASd-011: The permission property shall be part of the submodel that is referenced within the selectablePermissions attribute of AccessControl."@en ;

    rdfs:label "has permission"^^xsd:string ;

    rdfs:domain aas:Permission ;

    rdfs:range aas:Property ;

.


###  https://admin-shell.io/aas/2/0/rdf#propertyValueId

aas:propertyValueId rdf:type owl:ObjectProperty ;

    rdfs:comment "Reference to the global unqiue id of a coded value."@en ;

    rdfs:label "has property value id"^^xsd:string ;

    skos:note "Constraint AASd-007: if both, the value and the valueId are present then the value needs to be identical to the value of the referenced coded value in valueId."@en ;

    rdfs:domain aas:Property ;

    rdfs:range aas:Reference ;

.


###  https://admin-shell.io/aas/2/0/rdf#kindOfPermission

aas:kindOfPermission rdf:type owl:ObjectProperty ;

    rdfs:comment "Description of the kind of permission. Possible kind of permission also include the denial of the permission."@en ;

    rdfs:label "has kind of permission"^^xsd:string ;

    rdfs:domain aas:Permission ;

    rdfs:range aas:PermissionKind ;

.
```

```
### https://admin-shell.io/aas/2/0/rdf#subjectAttribute

aas:subjectAttribute rdf:type owl:ObjectProperty ;

    rdfs:comment "A data element that further classifies a specific subject. "@en ;

    skos:note "Constraint AASd-025: The data element shall be part of the submodel that is referenced within the selectableSubjectAttributes attribute of AccessControl."@en ;

    rdfs:label "has subject attribute"^^xsd:string ;

    rdfs:domain aas:SubjectAttributes ;

    rdfs:range aas:DataElement ;

.
```

ii. RDF Schema for the Administration Shell

```
@prefix aas: <http://admin-shell.io/aas/2/0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

# Metadata
<http://admin-shell.io/aas/2/0#>  a owl:Ontology ;
  owl:imports <http://datashapes.org/dash> ;
  owl:imports sh: ;
  sh:declare [
    a sh:PrefixDeclaration ;
    sh:namespace "http://admin-shell.io/aas/2/0#"^^xsd:anyURI ;
    sh:prefix "aas"^^xsd:string ;
  ] ;
.



aas:AccessControlPolicyPointsShape a sh:NodeShape ;
  sh:targetClass aas:AccessControlPolicyPoints ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:policyAdministrationPoint ;
      sh:class aas:PolicyAdministrationPoint ;
      sh:maxCount 1 ;
      sh:minCount 1 ;
      sh:message "Exactly one <i>policyAdministrationPoint</b> pointing to a <i>PolicyAdministrationPoint</i> is required."^^xsd:string ;
      sh:name "AccessControlPolicyPoints.policyAdministrationPoint"^^xsd:string ;
    ] ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:policyDecisionPoint ;
      sh:class aas:PolicyDecisionPoint ;
      sh:maxCount 1 ;
```

```
    sh:minCount 1 ;
    sh:message "Exactly one <i>policyDecisionPoint</b> pointing to a <i>PolicyDecisionPoint</i> is required."^^xsd:string ;
    sh:name "AccessControlPolicyPoints.policyDecisionPoint"^^xsd:string ;
   ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:policyEnforcementPoint ;
    sh:class aas:PolicyEnforcementPoint ;
    sh:maxCount 1 ;
    sh:minCount 1 ;
    sh:message "Exactly one <i>policyEnforcementPoint</b> pointing to a <i>PolicyEnforcementPoint</i> is required."^^xsd:string ;
    sh:name "AccessControlPolicyPoints.policyEnforcementPoint"^^xsd:string ;
   ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:policyInformationPoints ;
    sh:class aas:PolicyInformationPoints ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "Only one <i>policyInformationPoints</b> pointing to a <i>PolicyInformationPoints</i> is allowed."^^xsd:string ;
    sh:name "AccessControlPolicyPoints.policyInformationPoints"^^xsd:string ;
   ] ;
.


aas:AccessControlShape a sh:NodeShape ;
  sh:targetClass aas:AccessControl ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:accessPermissionRule ;
    sh:class aas:AccessPermissionRule ;
    sh:minCount 0 ;
    sh:message "A <i>accessPermissionRule</b> must point to a <i>AccessPermissionRule</i>."^^xsd:string ;
    sh:name "AccessControl.accessPermissionRule"^^xsd:string ;
   ] ;
  sh:property [
```

```
    a sh:PropertyShape ;
    sh:path aas:selectableSubjectAttributes ;
    sh:class aas:Submodel ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "Only one <i>selectableSubjectAttributes</b> pointing to a <i>Submodel</i> is allowed."^^xsd:string ;
    sh:name "AccessControl.selectableSubjectAttributes"^^xsd:string ;
  ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:defaultSubjectAttributes ;
    sh:class aas:Submodel ;
    sh:maxCount 1 ;
    sh:minCount 1 ;
    sh:message "Exactly one <i>defaultSubjectAttributes</b> pointing to a <i>Submodel</i> is required."^^xsd:string ;
    sh:name "AccessControl.defaultSubjectAttributes"^^xsd:string ;
  ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:selectablePermissions ;
    sh:class aas:Submodel ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "Only one <i>selectablePermissions</b> pointing to a <i>Submodel</i> is allowed."^^xsd:string ;
    sh:name "AccessControl.selectablePermissions"^^xsd:string ;
  ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:defaultPermissions ;
    sh:class aas:Submodel ;
    sh:maxCount 1 ;
    sh:minCount 1 ;
    sh:message "Exactly one <i>defaultPermissions</b> pointing to a <i>Submodel</i> is required."^^xsd:string ;
    sh:name "AccessControl.defaultPermissions"^^xsd:string ;
  ] ;
 sh:property [
```

```
    a sh:PropertyShape ;
    sh:path aas:selectableEnvironmentAttributes ;
    sh:class aas:Submodel ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "Only one <i>selectableEnvironmentAttributes</b> pointing to a <i>Submodel</i> is allowed."^^xsd:string ;
    sh:name "AccessControl.selectableEnvironmentAttributes"^^xsd:string ;
  ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:defaultEnvironmentAttributes ;
    sh:class aas:Submodel ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "Only one <i>defaultEnvironmentAttributes</b> pointing to a <i>Submodel</i> is allowed."^^xsd:string ;
    sh:name "AccessControl.defaultEnvironmentAttributes"^^xsd:string ;
  ] ;
.


aas:AccessPermissionRuleShape a sh:NodeShape ;
 sh:targetClass aas:AccessPermissionRule ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:targetSubjectAttributes ;
    sh:class aas:SubjectAttributes ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
    sh:message "Exactly one <i>targetSubjectAttributes</b> pointing to a <i>SubjectAttributes</i> is required."^^xsd:string ;
    sh:name "AccessPermissionRule.targetSubjectAttributes"^^xsd:string ;
  ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:permissionsPerObject ;
    sh:class aas:PermissionsPerObject ;
    sh:minCount 0 ;
```

```
    sh:message "A <i>permissionsPerObject</b> must point to a <i>PermissionsPerObject</i>."^^xsd:string ;
    sh:name "AccessPermissionRule.permissionsPerObject"^^xsd:string ;
  ] ;
.

aas:AdministrativeInformationShape  a sh:NodeShape ;
  sh:targetClass aas:AdministrativeInformation ;
  rdfs:subClassOf aas:HasDataSpecificationShape ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:revision ;
    sh:datatype xsd:string ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "<b>AdministrativeInformationShape</b>: Only one value for <i>revision</i> is allowed."^^xsd:string ;
    sh:name "revision"^^xsd:string ;
    sh:nodeKind sh:Literal ;
  ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:version ;
    sh:datatype xsd:string ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "Only one value for <i>version</i> is allowed."^^xsd:string ;
    sh:name "version"^^xsd:string ;
    sh:nodeKind sh:Literal ;
  ] ;
.

aas:AnnotatedRelationshipElementShape a sh:NodeShape ;
  sh:targetClass aas:AnnotatedRelationshipElement ;
  rdfs:subClassOf aas:RelationshipElementShape ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:annotation ;
    sh:class aas:Reference ;
```

```
    sh:minCount 0 ;
    sh:message "An <i>annotation</i> must point to a Reference."^^xsd:string ;
    sh:name "annotation"^^xsd:string ;
   ] ;
.

aas:AssetShape a sh:NodeShape ;
 sh:targetClass aas:Asset ;
 rdfs:subClassOf aas:HasDataSpecificationShape ;
 rdfs:subClassOf aas:IdentifiableShape ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:assetKind ;
    sh:class aas:AssetKind ;
    sh:maxCount 1 ;
    sh:minCount 1 ;
    sh:message "Exactly one <i>assetKind</i> attribute having an <i>AssetKind</i> entity is required."^^xsd:string
;
    sh:name "Asset.kind"^^xsd:string ;
   ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:billOfMaterial ;
    sh:class aas:Reference ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "Only one <i>billOfMaterial</i> attribute having an <i>reference</i> entity is allowed."^^xsd:string
;
    sh:name "Asset.billOfMaterial"^^xsd:string ;
   ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:assetIdentificationModel ;
    sh:class aas:Reference ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "Only one <i>assetIdentificationModel</i> is allowed."^^xsd:string ;
    sh:name "Asset.assetIdentificationModel"^^xsd:string ;
```

```
    ] ;
.


aas:AssetAdministrationShellShape a sh:NodeShape ;
  sh:targetClass aas:AssetAdministrationShell ;
  rdfs:subClassOf aas:HasDataSpecificationShape ;
  rdfs:subClassOf aas:IdentifiableShape ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:asset ;
       sh:class aas:Reference ;
      sh:maxCount 1 ;
      sh:message "Exactly one <i>asset</i> attribute having an <i>reference</i> entity is required."^^xsd:string ;
      sh:minCount 1 ;
      sh:name "asset"^^xsd:string ;
    ] ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:derivedFrom ;
      sh:class aas:Reference ;
      sh:maxCount 1 ;
      sh:minCount 0 ;
      sh:message "Only one <i>derivedFrom</i> attribute having an <i>reference</i> entity is allowed."^^xsd:string
;
      sh:name "derivedFrom"^^xsd:string ;
    ] ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:security ;
      sh:maxCount 1 ;
      sh:minCount 0 ;
      sh:class aas:Security ;
      sh:message "Exactly one <i>security</i> attribute to a <i>security</i> entity is required."^^xsd:string ;
      sh:name "security"^^xsd:string ;
    ] ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:hasSubmodel ;
```

```
    sh:minCount 0 ;
    sh:class aas:Reference ;
    sh:message "<i>hasSubmodel</i> must have a <i>reference</i> entity pointing to a Submodel."^^xsd:string ;
    sh:name "submodel"^^xsd:string ;
   ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:conceptDictionary ;
    sh:minCount 0 ;
    sh:class aas:ConceptDictionary ;
     sh:message "<i>conceptDictionary</i> must point to a ConceptDictionary"^^xsd:string ;
    sh:name "conceptDictionary"^^xsd:string ;
   ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:view ;
    sh:minCount 0 ;
    sh:class aas:View ;
     sh:message "<i>view</i> must point to a View"^^xsd:string ;
    sh:name "view"^^xsd:string ;
   ] ;
.

aas:BasicEventShape a sh:NodeShape ;
  rdfs:subClassOf aas:EventShape ;
  sh:targetClass aas:BasicEvent ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:observed ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
    sh:class aas:Reference ;
     sh:message "Exactly one <i>observed</i> attribute linking to a <i>Reference</i> is required."^^xsd:string ;
    sh:name "observed"^^xsd:string ;
   ] ;
.
```

```
aas:BlobCertificateShape a sh:NodeShape ;
  rdfs:subClassOf aas:CertificateShape ;
  sh:targetClass aas:BlobCertificate ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:blobCertificate ;
      sh:minCount 1 ;
      sh:maxCount 1 ;
      sh:class aas:Blob ;
      sh:message "Exactly one <i>blobCertificate</i> pointing to a  <i>Blob</i> is required."^^xsd:string ;
      sh:name "BlobCertificate.blobCertificate"^^xsd:string ;
    ] ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:containedExtension ;
      sh:minCount 0 ;
      sh:class aas:Reference ;
      sh:message "A <i>containedExtension</i> must point to a <i>Reference</i>."^^xsd:string ;
      sh:name "BlobCertificate.containedExtension"^^xsd:string ;
    ] ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:lastCertificate ;
      sh:minCount 1 ;
      sh:maxCount 1 ;
      sh:datatype xsd:boolean ;
      sh:message "Exactly one <i>lastCertificate</i> pointing to a  <i>boolean</i> is required."^^xsd:string ;
      sh:name "BlobCertificate.lastCertificate"^^xsd:string ;
    ] ;
  .


aas:BlobShape a sh:NodeShape ;
  rdfs:subClassOf aas:DataElementShape ;
  sh:targetClass aas:Blob ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:mimeType ;
```

```
    sh:maxCount 1 ;
    sh:message "A <i>mimeType</i> is required"^^xsd:string ;
    sh:minCount 1 ;
    sh:name "mime type"^^xsd:string ;
   ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:value ;
    sh:maxCount 1 ;
    sh:message "Only one <i>value</i> is allowed"^^xsd:string ;
    sh:name "value"^^xsd:string ;
   ] ;
.

aas:CapabilityShape  a sh:NodeShape ;
  rdfs:subClassOf aas:SubmodelElementShape ;
  sh:targetClass aas:Capability ;
  .

aas:CertificateShape  a sh:NodeShape ;
  sh:targetClass aas:Certificate ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:policyAdministrationPoint ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
    sh:class aas:PolicyAdministrationPoint ;
    sh:message "Exactly one <i>policyAdministrationPoint</i> pointing to a  <i>PolicyAdministrationPoint</i> is
required."^^xsd:string ;
    sh:name "Certificate.policyAdministrationPoint"^^xsd:string ;
   ] ;
  .

aas:ConceptDescriptionShape a sh:NodeShape ;
  sh:targetClass aas:ConceptDescription ;
  rdfs:subClassOf aas:HasDataSpecificationShape ;
  rdfs:subClassOf aas:IdentifiableShape ;
  sh:property [
```

```
      a sh:PropertyShape ;
      sh:path aas:isCaseOf ;
      sh:minCount 0 ;
      sh:class aas:Reference ;
      sh:message "<i>isCaseOf</i> must have a <i>reference</i> entity."^^xsd:string ;
      sh:name "ConceptDescription.isCaseOf"^^xsd:string ;
    ] ;
.

aas:ConceptDictionaryShape a sh:NodeShape ;
  rdfs:subClassOf aas:ReferableShape ;
  sh:targetClass aas:ConceptDictionary ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:conceptDescription ;
      sh:class aas:Reference ;
      sh:minCount 0 ;
      sh:message "<i>conceptDescription</i> must have a <i>reference</i> entity pointing to a ConceptDescription."^^xsd:string ;
      sh:name "concept description"^^xsd:string ;
    ] ;
.

aas:DataElementShape a sh:NodeShape ;
  rdfs:subClassOf aas:SubmodelElementShape ;
  sh:targetClass aas:DataElement ;
.

aas:DataSpecificationShape  a sh:NodeShape ;
  rdfs:subClassOf aas:IdentifiableShape ;
  sh:targetClass aas:DataSpecification ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:content ;
      sh:class aas:DataSpecificationContent ;
      sh:maxCount 1 ;
      sh:message "A value for <i>content</i> is required"^^xsd:string ;
      sh:minCount 1 ;
```

```
    sh:name "content"^^xsd:string ;
  ] ;
.

aas:DataSpecificationContentShape  a sh:NodeShape ;
.

aas:DataSpecificationIEC61360Shape  a sh:NodeShape ;
 rdfs:subClassOf aas:DataSpecificationContentShape ;
 sh:targetClass aas:DataSpecificationIEC61360 ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:preferredName ;
    sh:message "Exactly one <i>preferredName</i> is required"^^xsd:string ;
    sh:minCount 1 ;
    sh:datatype rdf:langString ;
    sh:name "preferredName"^^xsd:string ;
  ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:shortName ;
    sh:maxCount 1 ;
    sh:message "A <i>shortName</i> must have a LangString"^^xsd:string ;
    sh:minCount 0 ;
    sh:datatype rdf:langString ;
    sh:name "shortName"^^xsd:string ;
  ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:unit ;
    sh:maxCount 1 ;
    sh:message "A <i>unit</i> must have a String"^^xsd:string ;
    sh:minCount 0 ;
    sh:datatype xsd:string ;
    sh:name "unit"^^xsd:string ;
  ] ;
 sh:property [
```

```
    a sh:PropertyShape ;
    sh:path aas:unitId ;
    sh:class aas:Reference ;
    sh:maxCount 1 ;
    sh:message "A <i>unitId</i> must have a Reference."^^xsd:string ;
    sh:minCount 0 ;
    sh:name "unitId"^^xsd:string ;
  ] ;
sh:property [
    a sh:PropertyShape ;
    sh:path aas:sourceOfDefinition ;
    sh:maxCount 1 ;
    sh:message "A <i>sourceOfDefinition</i> must have a String"^^xsd:string ;
    sh:minCount 0 ;
    sh:datatype xsd:string ;
    sh:name "sourceOfDefinition"^^xsd:string ;
  ] ;
sh:property [
    a sh:PropertyShape ;
    sh:path aas:symbol ;
    sh:maxCount 1 ;
    sh:message "A <i>symbol</i> must have a string"^^xsd:string ;
    sh:minCount 0 ;
    sh:datatype xsd:string ;
    sh:name "symbol"^^xsd:string ;
  ] ;
sh:property [
    a sh:PropertyShape ;
    sh:path aas:datatype ;
    sh:maxCount 1 ;
    sh:message "Exactly one <i>datatype</i> is required."^^xsd:string ;
    sh:minCount 1 ;
    sh:name "datatype"^^xsd:string ;
  ] ;
sh:property [
    a sh:PropertyShape ;
    sh:path aas:definition ;
```

```
    sh:message "A <i>definition</i> must have a LangString"^^xsd:string ;
    sh:minCount 0 ;
    sh:datatype rdf:langString ;
    sh:name "definition"^^xsd:string ;
  ] ;
sh:property [
    a sh:PropertyShape ;
    sh:path aas:valueFormat ;
    sh:maxCount 1 ;
    sh:message "A <i>valueFormat</i> must have a string"^^xsd:string ;
    sh:minCount 0 ;
    sh:datatype xsd:string ;
    sh:name "valueFormat"^^xsd:string ;
  ] ;
sh:property [
    a sh:PropertyShape ;
    sh:path aas:valueList ;
    sh:maxCount 1 ;
    sh:message "A <i>valueList</i> must have a ValueList"^^xsd:string ;
    sh:minCount 0 ;
    sh:name "valueList"^^xsd:string ;
  ] ;
sh:property [
    a sh:PropertyShape ;
    sh:path aas:value ;
    sh:maxCount 1 ;
    sh:message "A <i>value</i> must have a Literal"^^xsd:string ;
    sh:minCount 0 ;
    sh:nodeKind sh:Literal ;
    sh:name "value"^^xsd:string ;
  ] ;
sh:property [
    a sh:PropertyShape ;
    sh:path aas:valueId ;
    sh:class aas:Reference ;
    sh:maxCount 1 ;
    sh:message "A <i>valueId</i> must have a Reference"^^xsd:string ;
```

```
    sh:minCount 0 ;
    sh:name "valueId"^^xsd:string ;
   ] ;
.


aas:DataSpecificationPhysicalUnitShape a sh:NodeShape ;
 sh:targetClass aas:DataSpecificationPhysicalUnit ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:unitName ;
    sh:class aas:SubmodelElement ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
    sh:message "Exactly one <i>unitName</i> which links to a <i>string</i> is required."^^xsd:string ;
    sh:name "DataSpecificationPhysicalUnit.unitName"^^xsd:string ;
   ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:unitSymbol ;
    sh:datatype xsd:string ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
    sh:message "Exactly one <i>unitSymbol</i> which links to a <i>string</i> is required."^^xsd:string ;
    sh:name "DataSpecificationPhysicalUnit.unitSymbol"^^xsd:string ;
   ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:definition ;
    sh:class rdf:langString ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
    sh:message "Exactly one <i>unitSymbol</i> which links to a <i>langstring</i> is required."^^xsd:string ;
    sh:name "DataSpecificationPhysicalUnit.definition"^^xsd:string ;
   ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:siNotation ;
```

```
    sh:datatype xsd:string ;
    sh:minCount 0 ;
    sh:maxCount 1 ;
    sh:message "Only one <i>siNotation</i> which links to a <i>string</i> is allowed."^^xsd:string ;
    sh:name "DataSpecificationPhysicalUnit.siNotation"^^xsd:string ;
  ] ;
sh:property [
    a sh:PropertyShape ;
    sh:path aas:siName ;
    sh:datatype xsd:string ;
    sh:minCount 0 ;
    sh:maxCount 1 ;
    sh:message "Only one <i>siName</i> which links to a <i>string</i> is allowed."^^xsd:string ;
    sh:name "DataSpecificationPhysicalUnit.siName"^^xsd:string ;
  ] ;
sh:property [
    a sh:PropertyShape ;
    sh:path aas:dinNotation ;
    sh:datatype xsd:string ;
    sh:minCount 0 ;
    sh:maxCount 1 ;
    sh:message "Only one <i>dinNotation</i> which links to a <i>string</i> is allowed."^^xsd:string ;
    sh:name "DataSpecificationPhysicalUnit.dinNotation"^^xsd:string ;
  ] ;
sh:property [
    a sh:PropertyShape ;
    sh:path aas:eceName ;
    sh:datatype xsd:string ;
    sh:minCount 0 ;
    sh:maxCount 1 ;
    sh:message "Only one <i>eceName</i> which links to a <i>string</i> is allowed."^^xsd:string ;
    sh:name "DataSpecificationPhysicalUnit.eceName"^^xsd:string ;
  ] ;
sh:property [
    a sh:PropertyShape ;
    sh:path aas:eceCode ;
    sh:datatype xsd:string ;
```

```
    sh:minCount 0 ;
    sh:maxCount 1 ;
    sh:message "Only one <i>eceCode</i> which links to a <i>string</i> is allowed."^^xsd:string ;
    sh:name "DataSpecificationPhysicalUnit.eceCode"^^xsd:string ;
  ] ;
sh:property [
    a sh:PropertyShape ;
    sh:path aas:nistName ;
    sh:datatype xsd:string ;
    sh:minCount 0 ;
    sh:maxCount 1 ;
    sh:message "Only one <i>nistName</i> which links to a <i>string</i> is allowed."^^xsd:string ;
    sh:name "DataSpecificationPhysicalUnit.nistName"^^xsd:string ;
  ] ;
sh:property [
    a sh:PropertyShape ;
    sh:path aas:sourceOfDefinition ;
    sh:datatype xsd:string ;
    sh:minCount 0 ;
    sh:maxCount 1 ;
    sh:message "Only one <i>sourceOfDefinition</i> which links to a <i>string</i> is allowed."^^xsd:string ;
    sh:name "DataSpecificationPhysicalUnit.sourceOfDefinition"^^xsd:string ;
  ] ;
sh:property [
    a sh:PropertyShape ;
    sh:path aas:conversionFactor ;
    sh:datatype xsd:string ;
    sh:minCount 0 ;
    sh:maxCount 1 ;
    sh:message "Only one <i>conversionFactor</i> which links to a <i>string</i> is allowed."^^xsd:string ;
    sh:name "DataSpecificationPhysicalUnit.conversionFactor"^^xsd:string ;
  ] ;
sh:property [
    a sh:PropertyShape ;
    sh:path aas:registrationAuthorityId ;
    sh:datatype xsd:string ;
    sh:minCount 0 ;
```

```
    sh:maxCount 1 ;
    sh:message "Only one <i>registrationAuthorityId</i> which links to a <i>string</i> is allowed."^^xsd:string ;
    sh:name "DataSpecificationPhysicalUnit.registrationAuthorityId"^^xsd:string ;
   ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:supplier ;
    sh:datatype xsd:string ;
    sh:minCount 0 ;
    sh:maxCount 1 ;
    sh:message "Only one <i>supplier</i> which links to a <i>string</i> is allowed."^^xsd:string ;
    sh:name "DataSpecificationPhysicalUnit.supplier"^^xsd:string ;
   ] ;
.


aas:EntityShape a sh:NodeShape ;
  rdfs:subClassOf aas:SubmodelElementShape ;
  sh:targetClass aas:Entity ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:statement ;
    sh:class aas:SubmodelElement ;
    sh:minCount 0 ;
    sh:message "A <i>statement</i> must link to a <i>SubmodelElement</i>."^^xsd:string ;
    sh:name "statement"^^xsd:string ;
   ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:entityType ;
    sh:class aas:EntityType ;
    sh:maxCount 1 ;
    sh:minCount 1 ;
    sh:message "Exactly one <i>entityType</i> linking to a <i>EntityType</i> is required."^^xsd:string ;
    sh:name "statement"^^xsd:string ;
   ] ;
  sh:property [
    a sh:PropertyShape ;
```

```
    sh:path aas:asset ;
    sh:class aas:Asset ;
    sh:maxCount 1 ;
    sh:message "Only one <i>asset</i> attribute linking to an <i>Asset</i> is allowed."^^xsd:string ;
    sh:minCount 0 ;
    sh:name "Entity.asset"^^xsd:string ;
   ] ;
.

aas:EventShape a sh:NodeShape ;
 rdfs:subClassOf aas:SubmodelElementShape ;
 sh:targetClass aas:Event ;
 skos:note "As of November 2019, Event is not mandatory. This shape severs as a stump for further definitions."@en
;
.

aas:EventElementShape a sh:NodeShape ;
 rdfs:subClassOf aas:SubmodelElementShape ;
 sh:targetClass aas:EventElement ;
 skos:note "As of November 2019, EventElement is not mandatory. This shape severs as a stump for further
definitions."@en ;
.

aas:EventMessageShape a sh:NodeShape ;
 rdfs:subClassOf aas:SubmodelElementShape ;
 sh:targetClass aas:EventMessage ;
 skos:note "As of November 2019, EventMessage is not mandatory. This shape severs as a stump for further
definitions."@en ;
.

aas:FileShape a sh:NodeShape ;
 rdfs:subClassOf aas:DataElementShape ;
 sh:targetClass aas:File ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:mimeType ;
    sh:maxCount 1 ;
    sh:datatype xsd:string ;
```

```
    sh:message "A <i>mimeType</i> is required"^^xsd:string ;
    sh:minCount 1 ;
    sh:name "mime type"^^xsd:string ;
  ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:value ;
    sh:datatype xsd:string ;
    sh:minCount 0 ;
    sh:maxCount 1 ;
    sh:message "Only one <i>value</i> is allowed"^^xsd:string ;
    sh:name "value"^^xsd:string ;
  ] ;
.

aas:FormulaShape a sh:NodeShape ;
  sh:targetClass aas:Formula ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:dependsOn ;
    sh:class aas:Reference ;
    sh:minCount 0 ;
    sh:message "Only References are allowed for <i>dependsOn</i>."^^xsd:string ;
    sh:name "depends on"^^xsd:string ;
  ] ;
.

aas:HasDataSpecificationShape a sh:NodeShape ;
 sh:targetClass aas:HasDataSpecification ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:dataSpecification ;
    sh:class aas:Reference ;
    sh:minCount 0 ;
    sh:message "Only References are allowed for <i>dataSpecification</i>."^^xsd:string ;
    sh:name "data specification"^^xsd:string ;
  ] ;
```

```
.

aas:HasKindShape a sh:NodeShape ;
    sh:targetClass aas:HasKind ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:kind ;
      sh:class aas:ModelingKind ;
      sh:defaultValue aas:INSTANCE ;
      sh:maxCount 1 ;
      sh:message "Only one value for <i>kind</i> is allowed."^^xsd:string ;
      sh:minCount 0 ;
      sh:name "kind"^^xsd:string ;
    ] ;
.

aas:HasSemanticsShape a sh:NodeShape ;
  sh:targetClass aas:HasSemantics ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:semanticId ;
      sh:maxCount 1 ;
      sh:minCount 0 ;
      sh:class aas:Reference ;
      sh:message "Only one value for <i>semanticId</i> is allowed."^^xsd:string ;
      sh:name "semantic id"^^xsd:string ;
    ] ;
.

aas:IdentifiableShape  a sh:NodeShape ;
  sh:targetClass aas:Identifiable ;
  rdfs:subClassOf aas:ReferableShape ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:administration ;
      sh:class aas:AdministrativeInformation ;
      sh:maxCount 1 ;
```

```
    sh:message "Only one value for <i>administration</i> is allowed."^^xsd:string ;
    sh:minCount 0 ;
    sh:name "administration"^^xsd:string ;
   ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:identification ;
    #sh:class aas:Identifier ;
    sh:maxCount 1 ;
    sh:message "An <i>identifier</i> is required."^^xsd:string ;
    sh:minCount 1 ;
    sh:name "identification"^^xsd:string ;
   ] ;
.


aas:IdentifierShape  a sh:NodeShape ;
  sh:targetClass aas:Identifier ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:idType ;
    sh:class aas:IdentifierType ;
    sh:description "identifier id type"^^xsd:string ;
    sh:maxCount 1 ;
    sh:message "An <i>idType</i> is required"^^xsd:string ;
    sh:minCount 1 ;
    sh:name "id type"^^xsd:string ;
   ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:id ;
    sh:nodeKind sh:Literal ;
    sh:description "identifier id"^^xsd:string ;
    sh:maxCount 1 ;
    sh:message "An <i>id</i> is required"^^xsd:string ;
    sh:minCount 1 ;
    sh:name "id"^^xsd:string ;
   ] ;
```

```
.


aas:KeyShape  a sh:NodeShape ;
  sh:targetClass aas:Key ;
  sh:property [
      a sh:PropertyShape ;
      sh:class aas:KeyElement ;
      sh:path aas:type ;
      sh:maxCount 1 ;
      sh:message "Exactly 1 <i>type</i> must be stated"^^xsd:string ;
      sh:minCount 1 ;
      sh:name "type"^^xsd:string ;
    ] ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:local ;
      sh:datatype xsd:boolean ;
      sh:maxCount 1 ;
      sh:message "Exactly 1 <i>local</i> must be stated"^^xsd:string ;
      sh:minCount 1 ;
      sh:name "local"^^xsd:string ;
    ] ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:value ;
      #sh:datatype xsd:string ;
      sh:maxCount 1 ;
      sh:message "A <i>value</i> is required"^^xsd:string ;
      sh:minCount 1 ;
      sh:name "value"^^xsd:string ;
    ] ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:idType ;
      sh:class aas:KeyType ;
      sh:maxCount 1 ;
      sh:message "Exactly 1 <i>idType</i> must be stated"^^xsd:string ;
```

```
    sh:minCount 1 ;
    sh:name "idType"^^xsd:string ;
   ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:index ;
    sh:datatype xsd:integer ;
    sh:defaultValue "0"^^xsd:integer ;
    sh:maxCount 1 ;
    sh:message "Only 1 <i>index</i> can be stated"^^xsd:string ;
    sh:minCount 0 ;
    sh:name "value"^^xsd:string ;
   ] ;
.


aas:MultiLanguagePropertyShape  a sh:NodeShape ;
  sh:targetClass aas:MultiLanguageProperty ;
  rdfs:subClassOf aas:SubmodelElementShape ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:propertyCategory ;
    sh:class aas:Category ;
    sh:maxCount 1 ;
    sh:name "MultiLanguageProperty.propertyCategory"^^xsd:string ;
    sh:message "Only one <i>propertyCategory</i> is allowed"^^xsd:string ;
   ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:value ;
    sh:datatype rdf:langString ;
    sh:minCount 0 ;
    sh:message "A language string <i>value</i> must have a language tag"^^xsd:string ;
    sh:name "MultiLanguageProperty.value"^^xsd:string ;
   ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:valueId ;
```

```
      sh:maxCount 1 ;
      sh:minCount 0 ;
      sh:class aas:Reference ;
      sh:message "Only one <i>valueId</i> attribute having a <i>Reference</i> is allowed"^^xsd:string ;
      sh:name "Property.valueId"^^xsd:string ;
    ] ;
.

aas:ObjectAttributesShape a sh:NodeShape ;
  sh:targetClass aas:ObjectAttributes ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:objectAttribute ;
      sh:class aas:DataElement ;
      sh:minCount 1 ;
      sh:message "At least one <i>objectAttribute</i> pointing to a <i>DataElement</i> is required."^^xsd:string ;
      sh:name "ObjectAttributes.objectAttribute"^^xsd:string ;
    ] ;
.

aas:OperationShape a sh:NodeShape ;
  rdfs:subClassOf aas:SubmodelElementShape ;
  sh:targetClass aas:Operation ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:inputVariable ;
      sh:class aas:OperationVariable ;
      sh:minCount 0 ;
      sh:message "Only OperationVariables can be accepted as <i>inputVariable</i>"^^xsd:string ;
      sh:name "input variable"^^xsd:string ;
    ] ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:outputVariable ;
      sh:class aas:OperationVariable ;
      sh:minCount 0 ;
      sh:message "Only OperationVariables can be accepted as <i>outputVariable</i>"^^xsd:string ;
```

```
    sh:name "output variable"^^xsd:string ;
   ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:inoutputVariable ;
    sh:class aas:OperationVariable ;
    sh:minCount 0 ;
    sh:message "Only OperationVariables can be accepted as <i>inoutputVariable</i>"^^xsd:string ;
    sh:name "inoutput variable"^^xsd:string ;
   ] ;
.

aas:OperationVariableShape a sh:NodeShape ;
 sh:targetClass aas:OperationVariable ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:value ;
    sh:class aas:SubmodelElement ;
    sh:maxCount 1 ;
    sh:message "OperationVariable's <i>value</i> must point to a <i>SubmodelElement</i>."^^xsd:string ;
    sh:minCount 1 ;
    sh:name "operation variable value"^^xsd:string ;
   ] ;
.

aas:PolicyAdministrationPointShape a sh:NodeShape ;
 sh:targetClass aas:PolicyAdministrationPoint ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:localAccessControl ;
    sh:class aas:AccessControl ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "Only one <i>localAccessControl</i> attribute having a <i>AccessControl</i> is
allowed."^^xsd:string ;
    sh:name "PolicyAdministrationPoint.localAccessControl"^^xsd:string ;
   ] ;
 sh:property [
```

```
    a sh:PropertyShape ;
    sh:path aas:externalAccessControl ;
    sh:datatype xsd:boolean ;
    sh:maxCount 1 ;
    sh:minCount 1 ;
    sh:message "Exactly one <i>externalAccessControl</i> attribute having a <i>boolean</i> is required."^^xsd:string ;
    sh:name "PolicyAdministrationPoint.externalAccessControl"^^xsd:string ;
   ] ;
.

aas:PermissionShape a sh:NodeShape ;
 sh:targetClass aas:Permission ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:permission ;
    sh:class aas:Property ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
    sh:message "Exactly one <i>permission</b> pointing to a <i>Property</i> is required."^^xsd:string ;
    sh:name "Permission.permission"^^xsd:string ;
   ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:kindOfPermission ;
    sh:class aas:PermissionKind ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
    sh:message "Exactly one <i>kindOfPermission</b> pointing to a <i>PermissionKind</i> is required."^^xsd:string ;
    sh:name "Permission.kindOfPermission"^^xsd:string ;
   ] ;
.

aas:PermissionsPerObjectShape a sh:NodeShape ;
 sh:targetClass aas:PermissionsPerObject ;
 sh:property [
    a sh:PropertyShape ;
```

```
    sh:path aas:object ;
    sh:class aas:Referable ;
    sh:minCount 1 ;
    sh:maxCount 1 ;
    sh:message "Exactly one <i>object</b> pointing to a <i>Referable</i> is required."^^xsd:string ;
    sh:name "PermissionsPerObject.object"^^xsd:string ;
  ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:targetObjectAttributes ;
    sh:class aas:ObjectAttributes ;
    sh:minCount 0 ;
    sh:maxCount 1 ;
    sh:message "Only one <i>targetObjectAttributes</b> pointing to a <i>ObjectAttributes</i> is allowed."^^xsd:string ;
    sh:name "PermissionsPerObject.targetObjectAttributes"^^xsd:string ;
  ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:permission ;
    sh:class aas:Permission ;
    sh:minCount 0 ;
    sh:message "A <i>permission</b> must point to a <i>Permission</i>."^^xsd:string ;
    sh:name "PermissionsPerObject.permission"^^xsd:string ;
  ] ;
.


aas:PolicyInformationPointsShape a sh:NodeShape ;
 sh:targetClass aas:PolicyInformationPoints ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:internalInformationPoint ;
    sh:class aas:Submodel ;
    sh:minCount 0 ;
    sh:message "A <i>internalInformationPoint</i> attribute must point to a <i>Submodel</i>."^^xsd:string ;
    sh:name "PolicyInformationPoints.internalInformationPoint"^^xsd:string ;
  ] ;
 sh:property [
```

```
    a sh:PropertyShape ;
    sh:path aas:externalInformationPoints ;
    sh:datatype xsd:boolean ;
    sh:maxCount 1 ;
    sh:minCount 1 ;
    sh:message "Exactly one <i>externalInformationPoints</i> attribute having a <i>boolean</i> is required."^^xsd:string ;
    sh:name "PolicyInformationPoints.externalInformationPoints"^^xsd:string ;
   ] ;
.

aas:PolicyEnforcementPointShape a sh:NodeShape ;
 sh:targetClass aas:PolicyEnforcementPoint ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:externalPolicyDecisionPoint ;
    sh:datatype xsd:boolean ;
    sh:maxCount 1 ;
    sh:minCount 1 ;
    sh:message "Exactly one <i>externalPolicyDecisionPoint</i> attribute having a <i>boolean</i> is required."^^xsd:string ;
    sh:name "PolicyEnforcementPoint.externalPolicyDecisionPoint"^^xsd:string ;
   ] ;
.

aas:PropertyShape  a sh:NodeShape ;
 sh:targetClass aas:Property ;
 rdfs:subClassOf aas:DataElementShape ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:propertyCategory ;
    sh:class aas:Category ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "Only one <i>propertyCategory</i> attribute having a <i>Category</i> is allowed"^^xsd:string ;
    sh:name "propertyCategory"^^xsd:string ;
   ] ;
 sh:property [
```

```
    a sh:PropertyShape ;
    sh:path aas:value ;
    sh:nodeKind sh:Literal ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "Only one <i>value</i> is allowed."^^xsd:string ;
    sh:name "Property.value"^^xsd:string ;
  ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:valueId ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:class aas:Reference ;
    sh:message "Only one <i>valueId</i> is allowed"^^xsd:string ;
    sh:name "Property.valueId"^^xsd:string ;
  ] ;
.

aas:QualifiableShape  a sh:NodeShape ;
 sh:targetClass aas:Qualifiable ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:qualifier ;
    sh:class aas:Constraint ;
    sh:name "qualifier"^^xsd:string ;
  ] ;
.

aas:QualifierShape  a sh:NodeShape ;
 sh:targetClass aas:Qualifier ;
 rdfs:subClassOf aas:HasSemanticsShape ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:qualifierType ;
    sh:class aas:QualifierType ;
    sh:maxCount 1 ;
```

```
    sh:minCount 1 ;
    sh:message "Exactly one <i>qualifierType</i> linking to a <i>QualifierType</i> is required."^^xsd:string ;
    sh:name "qualifier type"^^xsd:string ;
   ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:qualifierValue ;
    sh:nodeKind sh:Literal ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "Only one <i>qualifierValue</i> is allowed."^^xsd:string ;
    sh:name "qualifier value"^^xsd:string ;
   ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:qualifierValueId ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:class aas:Reference ;
    sh:message "Only one <i>qualifierValueId</i> having a <i>Reference</i> entity is allowed."^^xsd:string ;
    sh:name "qualifier value id"^^xsd:string ;
   ] ;
.


aas:RangeShape a sh:NodeShape ;
  sh:targetClass aas:Range ;
  rdfs:subClassOf aas:DataElementShape ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:min ;
    sh:nodeKind sh:Literal ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "Only one <i>min</i> attribute having a <i>Literal</i> is allowed"^^xsd:string ;
    sh:name "min"^^xsd:string ;
   ] ;
  sh:property [
```

```
    a sh:PropertyShape ;
    sh:path aas:max ;
    sh:nodeKind sh:Literal ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "Only one <i>max</i> attribute having a <i>Literal</i> is allowed"^^xsd:string ;
    sh:name "max"^^xsd:string ;
  ] ;
.


aas:ReferableShape  a sh:NodeShape ;
 sh:targetClass aas:Referable ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:category ; # todo
    sh:datatype xsd:string ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "Only one value for <i>category</i> is allowed"^^xsd:string ;
    sh:name "Referable.category"^^xsd:string ;
  ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:description ;
    sh:datatype rdf:langString ;
    sh:name "Referable.description"^^xsd:string ;
    sh:minCount 0 ;
  ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:idShort ;
    sh:datatype xsd:string ;
    sh:maxCount 1 ;
    sh:minCount 1 ;
    sh:message "Exactly one <id>idShort</id> is required."^^xsd:string ;
    sh:name "Referable.idShort"^^xsd:string ;
    sh:pattern "^[a-zA-Z]\\w*$"^^xsd:string ;
```

```
    ] ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:parent ;
      sh:class aas:Referable ;
      sh:maxCount 1 ;
      sh:minCount 0 ;
      sh:message "Only one value for <i>parent</i> is allowed"^^xsd:string ;
      sh:name "Referable.parent"^^xsd:string ;
      sh:nodeKind sh:IRI ;
    ] ;
.

aas:ReferenceShape a sh:NodeShape ;
  sh:targetClass aas:Reference ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:key ;
      sh:class aas:Key ;
      sh:name "key"^^xsd:string ;
    ] ;
.

aas:ReferenceElementShape a sh:NodeShape ;
  sh:targetClass aas:ReferenceElement ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:value ;
      sh:minCount 0 ;
      sh:maxCount 1 ;
      sh:class aas:Reference ;
      sh:message "Only one value for <i>value</i> is allowed"^^xsd:string ;
      sh:name "ReferenceElement.value"^^xsd:string ;
    ] ;
.

aas:RelationshipElementShape  a sh:NodeShape ;
```

```
  rdfs:subClassOf aas:SubmodelElementShape ;
  sh:targetClass aas:RelationshipElement ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:relationshipFirst ;
      sh:class aas:Referable ;
      sh:maxCount 1 ;
      sh:message "A value for <i>relationshipFirst</i> is required"^^xsd:string ;
      sh:minCount 1 ;
      sh:name "first"^^xsd:string ;
    ] ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:relationshipSecond ;
      sh:class aas:Referable ;
      sh:maxCount 1 ;
      sh:message "A value for <i>relationshipSecond</i> is required"^^xsd:string ;
      sh:minCount 1 ;
      sh:name "second"^^xsd:string ;
    ] ;
.


aas:SecurityShape  a sh:NodeShape ;
  sh:targetClass aas:Security ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:accessControlPolicyPoints ;
      sh:class aas:AccessControlPolicyPoints ;
      sh:maxCount 1 ;
      sh:message "A value for <i>relationshipSecond</i> is required"^^xsd:string ;
      sh:minCount 1 ;
      sh:name "accessControlPolicyPoints"^^xsd:string ;
    ] ;
  sh:property [
      a sh:PropertyShape ;
      sh:path aas:certificate ;
      sh:class aas:Certificate ;
```

```
    sh:message "A <i>certificate</i> must point to a Certificate"^^xsd:string ;
    sh:minCount 0 ;
    sh:name "certificate"^^xsd:string ;
   ] ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:requiredCertificateExtension ;
    sh:class aas:Reference ;
    sh:message "A <i>requiredCertificateExtension</i> must point to a Reference."^^xsd:string ;
    sh:minCount 0 ;
    sh:name "requiredCertificateExtension"^^xsd:string ;
   ] ;
.

aas:SubjectAttributesShape a sh:NodeShape ;
  sh:targetClass aas:SubjectAttributes ;
  sh:property [
    a sh:PropertyShape ;
    sh:path aas:subjectAttribute ;
    sh:class aas:DataElement ;
    sh:minCount 1 ;
    sh:message "At least one <i>subjectAttribute</i> pointing to a <i>DataElement</i> is required."^^xsd:string ;
    sh:name "SubjectAttributes.subjectAttribute"^^xsd:string ;
   ] ;
.

aas:SubmodelShape  a sh:NodeShape ;
  sh:targetClass aas:Submodel ;
  rdfs:subClassOf aas:HasDataSpecificationShape ;
  rdfs:subClassOf aas:HasKindShape ;
  rdfs:subClassOf aas:HasSemanticsShape ;
  rdfs:subClassOf aas:IdentifiableShape ;
  rdfs:subClassOf aas:QualifiableShape ;
  sh:message "Invalid Identification Type (For Submodels with Kind = Template only Ids of Type IRDI or IRI are
allowed) /  (For Submodels with Kind = Instance only Ids of Type Custom or IRI are allowed)"^^xsd:string ;
  sh:or (
    [
      sh:path aas:kind ;
```

```
      sh:hasValue aas:TYPE ;
    ]
    [
      sh:path aas:identification ;
    ]
  ) ;
 sh:or (
      sh:not [
        sh:path aas:kind ;
        sh:hasValue aas:TYPE ;
       ]
  ) ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:submodelElement ;
    sh:class aas:SubmodelElement ;
    sh:minCount 0 ;
    sh:message "all submodel elements must be instances of type SubmodelElement"^^xsd:string ;
    sh:name "sub model element"^^xsd:string ;
  ] ;
.

aas:SubmodelElementShape  a sh:NodeShape ;
 sh:targetClass aas:SubmodelElement ;
 rdfs:subClassOf aas:HasDataSpecificationShape ;
 rdfs:subClassOf aas:HasKindShape ;
 rdfs:subClassOf aas:HasSemanticsShape ;
 rdfs:subClassOf aas:QualifiableShape ;
 rdfs:subClassOf aas:ReferableShape  ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:idShort ;
    sh:datatype xsd:string ;
    sh:description "SubmodelElement idShort is mendatory"^^xsd:string ;
    sh:maxCount 1 ;
    sh:message "An <i>idShort</i> is required"^^xsd:string ;
    sh:minCount 1 ;
```

```
    sh:name "id short"^^xsd:string ;
  ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:semanticId ;
    sh:class aas:Reference ;
    sh:description "SubmodelElement semantic id is mendatory"^^xsd:string ;
    sh:maxCount 1 ;
    sh:message "A <i>semanticId</i> is required"^^xsd:string ;
    sh:minCount 1 ;
    sh:name "semantic id"^^xsd:string ;
  ] ;
.

aas:SubmodelElementCollectionShape  a sh:NodeShape ;
 rdfs:subClassOf aas:SubmodelElementShape ;
 sh:targetClass aas:SubmodelElementCollection ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:allowDuplicates ;
    sh:datatype xsd:boolean ;
    sh:defaultValue false ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "Only one boolean value for <i>allowDuplicates</i> is allowed"^^xsd:string ;
    sh:name "allow duplicates"^^xsd:string ;
  ] ;
 sh:property [
    a sh:PropertyShape ;
    sh:path aas:ordered ;
    sh:datatype xsd:boolean ;
    sh:defaultValue false ;
    sh:maxCount 1 ;
    sh:minCount 0 ;
    sh:message "Only one boolean value for <i>ordered</i> is allowed"^^xsd:string ;
    sh:name "ordered"^^xsd:string ;
  ] ;
```

```
  sh:property [
     a sh:PropertyShape ;
     sh:path aas:value ;
     sh:class aas:SubmodelElement ;
     sh:minCount 0 ;
     sh:message "SubmodelElementCollection can contain only SubmodelElements for <i>value</i>."^^xsd:string ;
     sh:name "value"^^xsd:string ;
    ] ;
.

aas:ViewShape a sh:NodeShape ;
  rdfs:subClassOf aas:HasDataSpecificationShape ;
  rdfs:subClassOf aas:HasSemanticsShape ;
  rdfs:subClassOf aas:ReferableShape ;
  sh:targetClass aas:View ;
  sh:property [
     a sh:PropertyShape ;
     sh:path aas:containedElement ;
     sh:class aas:Reference ;
     sh:name "view contained element"^^xsd:string ;
      sh:message "View can contain only References for <i>containedElement</i>."^^xsd:string ;
    ] ;
  sh:property [
     a sh:PropertyShape ;
     sh:path aas:idShort ;
     sh:datatype xsd:string ;
     sh:maxCount 1 ;
     sh:message "An <i>idShort</i> is required"^^xsd:string ;
     sh:minCount 1 ;
     sh:name "id short"^^xsd:string ;
    ] ;
  sh:property [
     sh:path (aas:semanticId aas:key ) ;
     sh:message "semanticId can contain IRDI or IRI keys only"^^xsd:string ;
    ] ;
.
```

iii. RDF Example

```
@prefix aas: <http://admin-shell.io/aas/2/0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .



# Asset Administration Shell
<http://customer.com/aas/9175_7013_7091_9168> rdf:type aas:AssetAdministrationShell ;
  aas:idShort "ExampleMotor"^^xsd:string ;
  rdfs:label "ExampleMotor"^^xsd:string ;
  aas:description "A very short description of the AAS instance."@en ;
  rdfs:comment "A very short description of the AAS instance."^^xsd:string ;
  aas:asset [
    a aas:Reference ;
    aas:key [
      a aas:Key ;
      aas:index "0"^^xsd:integer ;
      aas:type aas:ASSET_IDENTIFIABLE_ELEMENT ;
      aas:local "true"^^xsd:boolean ;
      aas:value <http://customer.com/assets/KHBVZJSQKIY> ;
      aas:idType aas:IRI_IDENTIFIER_TYPE ;
    ] ;
  ] ;
  aas:submodel [
    a aas:Reference ;
    aas:key [
      a aas:Key ;
      aas:index "0"^^xsd:integer ;
      aas:type aas:SUBMODEL_IDENTIFIABLE_ELEMENT ;
      aas:local "true"^^xsd:boolean ;
      aas:value <http://i40.customer.com/type/1/1/F13E8576F6488342> ;
      aas:idType aas:IRI_IDENTIFIER_TYPE ;
    ] ;
  ] ;
```

```
    aas:submodel [
        a aas:Reference ;
        aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
            aas:type aas:SUBMODEL_IDENTIFIABLE_ELEMENT ;
            aas:local "true"^^xsd:boolean ;
            aas:value <http://i40.customer.com/type/1/1/7A7104BDAB57E184> ;
            aas:idType aas:IRI_IDENTIFIER_TYPE ;
        ] ;
    ] ;
    aas:submodel [
        a aas:Reference ;
        aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
            aas:type aas:SUBMODEL_IDENTIFIABLE_ELEMENT ;
            aas:local "true"^^xsd:boolean ;
            aas:value <http://i40.customer.com/instance/1/1/AC69B1CB44F07935>;
            aas:idType aas:IRI_IDENTIFIER_TYPE ;
        ];
    ] ;
    aas:submodel [
        a aas:Reference ;
        aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
            aas:type aas:SUBMODEL_IDENTIFIABLE_ELEMENT ;
            aas:local "true"^^xsd:boolean ;
            aas:value <http://i40.customer.com/type/1/1/1A7B62B529F19152> ;
                aas:idType aas:IRI_IDENTIFIER_TYPE ;
        ] ;
    ] ;
    aas:identification <http://customer.com/aas/9175_7013_7091_9168> ;
    aas:category "CONSTANT"^^xsd:string ;
.
```

```
# Asset
<http://customer.com/assets/KHBVZJSQKIY> rdf:type aas:Asset ;
   aas:idShort "ServoDCMotor"^^xsd:string ;
   rdfs:label "ServoDCMotor"^^xsd:string ;
   aas:identification <http://customer.com/assets/KHBVZJSQKIY> ;
   aas:assetIdentificationModel [
      rdf:type aas:Reference ;
      aas:key [
         rdf:type aas:Key ;
         aas:index "0"^^xsd:integer ;
         aas:type aas:SUBMODEL_IDENTIFIABLE_ELEMENT ;
         aas:local "true"^^xsd:boolean ;
         aas:value <http://i40.customer.com/type/1/1/F13E8576F6488342> ;
         aas:idType aas:IRI_IDENTIFIER_TYPE ;
      ] ;
   ] ;
   aas:assetKind aas:ASSET_INSTANCE ;
.

# Submodel
<http://i40.customer.com/type/1/1/F13E8576F6488342> rdf:type aas:Submodel ;
   aas:idShort "Identification"^^xsd:string ;
   rdfs:label "Identification"^^xsd:string ;
   aas:category "CONSTANT"^^xsd:string ;
   aas:description "Identification from Manufacturer"@en ;
   aas:description "Hersteller-Identifikation"@de ;
   rdfs:comment "Identification from Manufacturer"@en ;
   rdfs:comment "Hersteller-Identifikation"@de ;
   aas:identification <http://i40.customer.com/type/1/1/F13E8576F6488342> ;
   aas:semanticId [
      a aas:Reference ;
      aas:key [
         a aas:Key ;
         aas:index "0"^^xsd:integer ;
         aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
            aas:local "false"^^xsd:boolean ;
```

```
        aas:value "0173-1#01-ADN198#009"^^xsd:string ;
        aas:idType aas:IRDI_IDENTIFIER_TYPE ;
    ] ;
];
aas:kind aas:INSTANCE ;
aas:submodelElement [
    rdf:type aas:Property ;
    rdf:subject <http://i40.customer.com/type/1/1/F13E8576F6488342/Manufacturer> ;
    aas:idShort "Manufacturer"^^xsd:string ;
    rdfs:label "Manufacturer"^^xsd:string ;
    aas:propertyCategory aas:CONSTANT ;
    aas:kind aas:INSTANCE ;
    aas:semanticId [
        a aas:Reference ;
        aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
            aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
            aas:local "false"^^xsd:boolean ;
            aas:value "0173-1#02-AAO677#002"^^xsd:string ;
            aas:idType aas:IRDI_IDENTIFIER_TYPE ;
        ] ;
    ];
    aas:value "CUSTOMER GmbH"^^xsd:string ;
] ;
aas:submodelElement [
    rdf:type aas:Property ;
    rdf:subject <http://i40.customer.com/type/1/1/F13E8576F6488342/GLN> ;
    aas:idShort "GLN"^^xsd:string ;
    rdfs:label "GLN"^^xsd:string ;
    aas:propertyCategory aas:CONSTANT ;
    aas:kind aas:INSTANCE ;
    aas:semanticId [
        a aas:Reference ;
        aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
```

```
            aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
          aas:local "false"^^xsd:boolean ;
          aas:value "0173-1#02-AAY812#001"^^xsd:string ;
          aas:idType aas:IRDI_IDENTIFIER_TYPE ;
        ] ;
    ];
    aas:value "10101010"^^xsd:integer ;
] ;
aas:submodelElement [
    rdf:type aas:MultiLanguageProperty ;
    rdf:subject <http://i40.customer.com/type/1/1/F13E8576F6488342/ProductDesignation> ;
    aas:idShort "ProductDesignation"^^xsd:string ;
    rdfs:label "ProductDesignation"^^xsd:string ;
    aas:propertyCategory aas:CONSTANT ;
    aas:semanticId [
        a aas:Reference ;
        aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
                aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
            aas:local "false"^^xsd:boolean ;
            aas:value "0173-1#02-AAW338#001"^^xsd:string ;
            aas:idType aas:IRDI_IDENTIFIER_TYPE ;
        ] ;
    ];
    aas:kind aas:INSTANCE ;
    aas:value "I40 Capable Servo Motor"@en ;
] ;
aas:submodelElement [
    rdf:type aas:Property ;
    rdf:subject <http://i40.customer.com/type/1/1/F13E8576F6488342/SerialNumber> ;
    aas:idShort "SerialNumber"^^xsd:string ;
    rdfs:label "SerialNumber"^^xsd:string ;
    aas:propertyCategory aas:CONSTANT ;
    aas:semanticId [
        a aas:Reference ;
        aas:key [
```

```
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
                aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
            aas:local "false"^^xsd:boolean ;
            aas:value "0173-1#02-AAM556#002"^^xsd:string ;
            aas:idType aas:IRDI_IDENTIFIER_TYPE ;
          ] ;
      ];
      aas:kind aas:INSTANCE ;
      aas:value "P12345678I40"@en ;
   ] ;
.

# Submodel
<http://i40.customer.com/type/1/1/7A7104BDAB57E184> rdf:type aas:Submodel ;
   aas:idShort "TechnicalData"^^xsd:string ;
   rdfs:label "TechnicalData"^^xsd:string ;
   aas:category "CONSTANT"^^xsd:string ;
   aas:identification <http://i40.customer.com/type/1/1/7A7104BDAB57E184> ;
   aas:semanticId [
      a aas:Reference ;
      aas:key [
         a aas:Key ;
         aas:index "0"^^xsd:integer ;
             aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
         aas:local "false"^^xsd:boolean ;
         aas:value "0173-1#01-AFZ615#016"^^xsd:string ;
         aas:idType aas:IRDI_IDENTIFIER_TYPE ;
      ] ;
   ];
   aas:kind aas:INSTANCE ;
   aas:submodelElement [
      rdf:type aas:Property ;
      rdf:subject <http://i40.customer.com/type/1/1/7A7104BDAB57E184/MaxRotationSpeed> ;
      aas:idShort "MaxRotationSpeed"^^xsd:string ;
      rdfs:label "MaxRotationSpeed"^^xsd:string ;
      aas:propertyCategory aas:PARAMETER ;
```

```
    aas:semanticId [
      a aas:Reference ;
      aas:key [
        a aas:Key ;
        aas:index "0"^^xsd:integer ;
        aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
        aas:local "true"^^xsd:boolean ;
        aas:value "0173-1#02-BAA120#008"^^xsd:string ;
        aas:idType aas:IRDI_IDENTIFIER_TYPE ;
      ] ;
    ];
    aas:kind aas:INSTANCE ;
    aas:value "5000"^^xsd:integer ;
  ] ;
  aas:submodelElement [
    rdf:type aas:Property ;
    rdf:subject <http://i40.customer.com/type/1/1/7A7104BDAB57E184/MaxTorque> ;
    aas:idShort "MaxTorque"^^xsd:string ;
    rdfs:label "MaxTorque"^^xsd:string ;
    aas:propertyCategory aas:PARAMETER ;
    aas:semanticId [
      a aas:Reference ;
      aas:key [
        a aas:Key ;
        aas:index "0"^^xsd:integer ;
        aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
        aas:local "true"^^xsd:boolean ;
        aas:value "0173-1#02-BAE098#004"^^xsd:string ;
        aas:idType aas:IRDI_IDENTIFIER_TYPE ;
      ] ;
    ];
    aas:kind aas:INSTANCE ;
    aas:value "200"^^xsd:float ;
  ] ;
  aas:submodelElement [
    rdf:type aas:Property ;
    rdf:subject <http://i40.customer.com/type/1/1/7A7104BDAB57E184/CoolingType> ;
```

```
      aas:idShort "CoolingType"^^xsd:string ;
      rdfs:label "CoolingType"^^xsd:string ;
      aas:propertyCategory aas:PARAMETER ;
      aas:description "open circuit, external cooling"@en ;
      rdfs:comment "open circuit, external cooling"@en ;
      aas:semanticId [
         a aas:Reference ;
         aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
            aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
            aas:local "true"^^xsd:boolean ;
            aas:value "0173-1#02-BAE122#006"^^xsd:string ;
            aas:idType aas:IRDI_IDENTIFIER_TYPE ;
         ] ;
      ];
      aas:kind aas:INSTANCE ;
      aas:value "BAB657"^^xsd:string ;
      aas:valueId [
         a aas:Reference ;
         aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
            aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
            aas:local "true"^^xsd:boolean ;
            aas:value "0173-1#07-BAB657#003"^^xsd:string ;
            aas:idType aas:IRDI_IDENTIFIER_TYPE ;
         ] ;
      ];
   ] ;
.

# Submodel
<http://i40.customer.com/type/1/1/1A7B62B529F19152> rdf:type aas:Submodel ;
   aas:idShort "Documentation"^^xsd:string ;
   rdfs:label "Documentation"^^xsd:string ;
   aas:category "CONSTANT"^^xsd:string ;
```

```
aas:identification <http://i40.customer.com/type/1/1/1A7B62B529F19152> ;
aas:kind aas:INSTANCE ;
aas:submodelElementCollection [
  rdf:type aas:SubmodelElementCollection ;
  rdf:subject <http://i40.customer.com/type/1/1/7A7104BDAB57E184/OperatingManual> ;
  aas:idShort "OperatingManual"^^xsd:string ;
  rdfs:label "OperatingManual"^^xsd:string ;
  aas:semanticId [
    a aas:Reference ;
    aas:key [
      a aas:Key ;
      aas:index "0"^^xsd:integer ;
      aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
      aas:local "true"^^xsd:boolean ;
      aas:value <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Document> ;
      aas:idType aas:IRI_IDENTIFIER_TYPE ;
    ] ;
  ];
  aas:kind aas:INSTANCE ;
  aas:submodelElement [
    rdf:type aas:Property ;
    rdf:subject <http://i40.customer.com/type/1/1/7A7104BDAB57E184/OperatingManual/DocumentId> ;
    aas:idShort "DocumentId"^^xsd:string ;
    rdfs:label "DocumentId"^^xsd:string ;
    aas:propertyCategory aas:CONSTANT ;
    aas:semanticId [
      a aas:Reference ;
      aas:key [
        a aas:Key ;
        aas:index "0"^^xsd:integer ;
        aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
        aas:local "true"^^xsd:boolean ;
        aas:value <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentId/Val> ;
        aas:idType aas:IRI_IDENTIFIER_TYPE ;
      ] ;
    ];
    aas:kind aas:INSTANCE ;
```

```
        aas:value "3 608 870 A47"^^xsd:string ;
      ] ;
      aas:submodelElement [
        rdf:type aas:Property ;
        rdf:subject <http://i40.customer.com/type/1/1/7A7104BDAB57E184/OperatingManual/DocumentClassId> ;
        aas:idShort "DocumentClassId"^^xsd:string ;
        rdfs:label "DocumentClassId"^^xsd:string ;
        aas:propertyCategory aas:CONSTANT ;
        aas:semanticId [
          a aas:Reference ;
          aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
            aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
            aas:local "true"^^xsd:boolean ;
            aas:value <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentClassification/ClassId> ;
            aas:idType aas:IRI_IDENTIFIER_TYPE ;
          ] ;
        ];
        aas:kind aas:INSTANCE ;
        aas:value "03-02"^^xsd:string ;
      ] ;
      aas:submodelElement [
        rdf:type aas:MultiLanguageProperty ;
        rdf:subject
<http://i40.customer.com/type/1/1/7A7104BDAB57E184/OperatingManual/DocumentClassName> ;
        aas:idShort "DocumentClassName"^^xsd:string ;
        rdfs:label "DocumentClassName"^^xsd:string ;
        aas:propertyCategory aas:CONSTANT ;
        aas:semanticId [
          a aas:Reference ;
          aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
            aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
            aas:local "true"^^xsd:boolean ;
            aas:value <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentClassification/ClassName> ;
            aas:idType aas:IRI_IDENTIFIER_TYPE ;
```

```
            ] ;
         ];
         aas:kind aas:INSTANCE ;
         aas:value "Operation"@en ;
         aas:value "Bedienung"@de ;
      ] ;
      aas:submodelElement [
         rdf:type aas:Property ;
         rdf:subject
<http://i40.customer.com/type/1/1/7A7104BDAB57E184/OperatingManual/DocumentClassificationSystem> ;
         aas:idShort "DocumentClassificationSystem"^^xsd:string ;
         rdfs:label "DocumentClassificationSystem"^^xsd:string ;
         aas:semanticId [
            a aas:Reference ;
            aas:key [
               a aas:Key ;
               aas:index "0"^^xsd:integer ;
               aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
               aas:local "true"^^xsd:boolean ;
               aas:value
<http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentClassification/ClassificationSystem> ;
               aas:idType aas:IRI_IDENTIFIER_TYPE ;
            ] ;
         ];
         aas:kind aas:INSTANCE ;
         aas:value "VDI2770:2018"^^xsd:string ;
      ] ;
      aas:submodelElement [
         rdf:type aas:Property ;
         rdf:subject <http://i40.customer.com/type/1/1/7A7104BDAB57E184/OperatingManual/OrganizationName> ;
         aas:idShort "OrganizationName"^^xsd:string ;
         rdfs:label "OrganizationName"^^xsd:string ;
         aas:semanticId [
            a aas:Reference ;
            aas:key [
               a aas:Key ;
               aas:index "0"^^xsd:integer ;
               aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
```

```
            aas:local "true"^^xsd:boolean ;
            aas:value <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Organization/OrganizationName> ;
            aas:idType aas:IRI_IDENTIFIER_TYPE ;
          ] ;
        ];
        aas:kind aas:INSTANCE ;
        aas:value "CUSTOMER"^^xsd:string ;
      ] ;
      aas:submodelElement [
        rdf:type aas:Property ;
        rdf:subject
<http://i40.customer.com/type/1/1/7A7104BDAB57E184/OperatingManual/OrganizationOfficialName> ;
        aas:idShort "OrganizationOfficialName"^^xsd:string ;
        rdfs:label "OrganizationOfficialName"^^xsd:string ;
        aas:semanticId [
          a aas:Reference ;
          aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
            aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
            aas:local "true"^^xsd:boolean ;
            aas:value  <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Organization/OrganizationOfficialName>
;
            aas:idType aas:IRI_IDENTIFIER_TYPE ;
          ] ;
        ];
        aas:kind aas:INSTANCE ;
        aas:value "CUSTOMER GmbH"^^xsd:string ;
      ] ;
      aas:submodelElement [
        rdf:type aas:Property ;
        rdf:subject <http://i40.customer.com/type/1/1/7A7104BDAB57E184/OperatingManual/Title> ;
        aas:idShort "Title"^^xsd:string ;
        rdfs:label "Title"^^xsd:string ;
        aas:semanticId [
          a aas:Reference ;
          aas:key [
            a aas:Key ;
```

```
            aas:index "0"^^xsd:integer ;
            aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
            aas:local "true"^^xsd:boolean ;
            aas:value <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Description/Title> ;
            aas:idType aas:IRI_IDENTIFIER_TYPE ;
          ] ;
      ];
      aas:kind aas:INSTANCE ;
      aas:value "Operating Manual"^^xsd:string ;
    ] ;
    aas:submodelElement [
      rdf:type aas:Property ;
      rdf:subject <http://i40.customer.com/type/1/1/7A7104BDAB57E184/OperatingManual/Language> ;
      aas:idShort "Language"^^xsd:string ;
      rdfs:label "Language"^^xsd:string ;
      aas:semanticId [
        a aas:Reference ;
        aas:key [
          a aas:Key ;
          aas:index "0"^^xsd:integer ;
          aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
          aas:local "true"^^xsd:boolean ;
          aas:value <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentVersion/Language> ;
          aas:idType aas:IRI_IDENTIFIER_TYPE ;
        ] ;
      ];
      aas:kind aas:INSTANCE ;
      aas:value "en-US"^^xsd:string ;
    ] ;
    aas:submodelElement [
      rdf:type aas:File ;
      rdf:subject <http://i40.customer.com/type/1/1/7A7104BDAB57E184/OperatingManual/DigitalFile_PDF> ;
      aas:idShort "DigitalFile_PDF"^^xsd:string ;
      rdfs:label "DigitalFile_PDF"^^xsd:string ;
      aas:category "PARAMETER"^^xsd:string ;
      aas:semanticId [
        a aas:Reference ;
```

```
            aas:key [
               a aas:Key ;
               aas:index "0"^^xsd:integer ;
               aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
               aas:local "true"^^xsd:boolean ;
               aas:value <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Description/Title> ;
               aas:idType aas:IRI_IDENTIFIER_TYPE ;
            ] ;
         ];
         aas:kind aas:INSTANCE ;
         aas:mimeType "application/pdf"^^xsd:string ;
         aas:value "/aasx/OperatingManual.pdf"^^xsd:string ;
      ] ;
      aas:ordered "false"^^xsd:boolean ;
      aas:allowDuplicates "false"^^xsd:boolean ;
   ];
.

# Submodel
<http://i40.customer.com/instance/1/1/AC69B1CB44F07935> rdf:type aas:Submodel ;
   aas:idShort "OperationalData"^^xsd:string ;
   rdfs:label "OperationalData"^^xsd:string ;
   aas:category "VARIABLE"^^xsd:string ;
   aas:identification <http://i40.customer.com/instance/1/1/AC69B1CB44F07935> ;
   aas:semanticId [
      a aas:Reference ;
      aas:key [
         a aas:Key ;
         aas:index "0"^^xsd:integer ;
            aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
         aas:local "false"^^xsd:boolean ;
         aas:value "0173-1#01-AFZ615#016"^^xsd:string ;
         aas:idType aas:IRDI_IDENTIFIER_TYPE ;
      ] ;
   ];
   aas:kind aas:INSTANCE ;
   aas:submodelElement [
```

```
    rdf:type aas:Property ;
    rdf:subject <http://i40.customer.com/instance/1/1/AC69B1CB44F07935/RotationSpeed> ;
    aas:idShort "RotationSpeed"^^xsd:string ;
    rdfs:label "RotationSpeed"^^xsd:string ;
    aas:propertyCategory aas:VARIABLE ;
    aas:kind aas:INSTANCE ;
    aas:semanticId [
      a aas:Reference ;
      aas:key [
        a aas:Key ;
        aas:index "0"^^xsd:integer ;
        aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
        aas:local "true"^^xsd:boolean ;
        aas:value <http://customer.com/cd/1/1/18EBD56F6B43D895> ;
        aas:idType aas:IRI_IDENTIFIER_TYPE ;
      ] ;
    ];
    aas:kind aas:INSTANCE ;
    aas:value "4370"^^xsd:integer ;
  ] ;
  aas:submodelElement [
    rdf:type aas:Property ;
    rdf:subject <http://i40.customer.com/instance/1/1/AC69B1CB44F07935/Torque> ;
    aas:idShort "Torque"^^xsd:string ;
    rdfs:label "Torque"^^xsd:string ;
    aas:propertyCategory aas:VARIABLE ;
    aas:kind aas:INSTANCE ;
    aas:semanticId [
      a aas:Reference ;
      aas:key [
        a aas:Key ;
        aas:index "0"^^xsd:integer ;
        aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
        aas:local "true"^^xsd:boolean ;
        aas:value <http://customer.com/cd/1/1/18EBD56F6B43D896> ;
        aas:idType aas:IRI_IDENTIFIER_TYPE ;
      ] ;
```

```
        ];
        aas:kind aas:INSTANCE ;
        aas:value "117.4"^^xsd:float ;
    ] ;
.

# ConceptDescription
<http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Document> rdf:type aas:ConceptDescription ;
    aas:idShort "Document"^^xsd:string ;
    rdfs:label "Document"^^xsd:string ;
    aas:identification <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Document> ;
    aas:hasDataSpecification [
        rdf:type aas:Reference ;
        aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
                aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
            aas:local "false"^^xsd:boolean ;
            aas:value <http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360> ;
            aas:idType aas:IRI_IDENTIFIER_TYPE ;
        ] ;
    ] ;
    aas:content [
        rdf:type aas:DataSpecificationIEC61360 ;
            aas:preferredName "Document"@en ;
            aas:preferredName "Dokument"@de ;
            aas:shortName "Document"@en ;
            aas:sourceOfDefinition "[ISO 15519-1:2010]"^^xsd:string ;
        aas:datatype "ENTITY"^^xsd:string ;
        aas:definition "Feste und geordnete Menge von für die Verwendung durch Personen bestimmte Informationen,
die verwaltet und als Einheit zwischen Benutzern und System ausgetauscht werden kann."@de ;
    ] ;
.

# ConceptDescription
<http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentId/Val> rdf:type aas:ConceptDescription ;
    aas:idShort "DocumentIdValue"^^xsd:string ;
    rdfs:label "DocumentIdValue"^^xsd:string ;
```

```
    aas:category "CONSTANT"^^xsd:string ;
    aas:identification <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentId/Val> ;
    aas:hasDataSpecification [
        rdf:type aas:Reference ;
        aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
                aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
            aas:local "false"^^xsd:boolean ;
            aas:value <http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360> ;
            aas:idType aas:IRI_IDENTIFIER_TYPE ;
        ] ;
    ] ;
    aas:content [
        rdf:type aas:DataSpecificationIEC61360 ;
        aas:preferredName "Document Id"@en ;
            aas:preferredName "Dokumenten ID"@de ;
            aas:shortName "DocumentId"@en ;
        aas:datatype "STRING"^^xsd:string ;
        aas:definition "die eigentliche Identifikationsnummer"@de ;
    ] ;
.


# ConceptDescription
<http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentClassification/ClassId>                    rdf:type
aas:ConceptDescription ;
    aas:idShort "DocumentClassId"^^xsd:string ;
    rdfs:label "DocumentClassId"^^xsd:string ;
    aas:category "CONSTANT"^^xsd:string ;
    aas:identification <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentClassification/ClassId> ;
    aas:hasDataSpecification [
        rdf:type aas:Reference ;
        aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
                aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
            aas:local "false"^^xsd:boolean ;
            aas:value <http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360> ;
```

```
        aas:idType aas:IRI_IDENTIFIER_TYPE ;
      ] ;
  ] ;
  aas:content [
    rdf:type aas:DataSpecificationIEC61360 ;
      aas:preferredName "Document Class Id"@en ;
      aas:preferredName "Dokumentenklassen ID"@de ;
      aas:shortName "DocumentClassId"@en ;
    aas:datatype "STRING"^^xsd:string ;
    aas:definition "Eindeutige ID der Klasse in einer Klassifikation."@de ;
  ] ;
.


# ConceptDescription
<http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentClassification/ClassName>                rdf:type
aas:ConceptDescription ;
  aas:idShort "DocumentClassName"^^xsd:string ;
  rdfs:label "DocumentClassName"^^xsd:string ;
  aas:category "CONSTANT"^^xsd:string ;
  aas:identification <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentClassification/ClassName> ;
  aas:hasDataSpecification [
    rdf:type aas:Reference ;
    aas:key [
      a aas:Key ;
      aas:index "0"^^xsd:integer ;
        aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
      aas:local "false"^^xsd:boolean ;
      aas:value <http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360> ;
      aas:idType aas:IRI_IDENTIFIER_TYPE ;
    ] ;
  ] ;
  aas:content [
    rdf:type aas:DataSpecificationIEC61360 ;
      aas:preferredName "Document Class Name"@en ;
      aas:shortName "DocumentClassName"@en ;
    aas:datatype "STRING_TRANSLATABLE"^^xsd:string ;
    aas:definition "Liste von sprachabhängigen Namen zur ClassId."@de ;
  ] ;
```

```
.


# ConceptDescription

<http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentClassification/ClassificationSystem>  rdf:type aas:ConceptDescription ;

   aas:idShort "DocumentClassificationSystem"^^xsd:string ;

   rdfs:label "DocumentClassificationSystem"^^xsd:string ;

   aas:category "CONSTANT"^^xsd:string ;

   aas:identification <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentClassification/ClassificationSystem> ;

   aas:hasDataSpecification [

      rdf:type aas:Reference ;

      aas:key [

         a aas:Key ;

         aas:index "0"^^xsd:integer ;

            aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;

         aas:local "false"^^xsd:boolean ;

         aas:value <http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360> ;

         aas:idType aas:IRI_IDENTIFIER_TYPE ;

      ] ;

   ] ;

   aas:content [

      rdf:type aas:DataSpecificationIEC61360 ;

      aas:preferredName "Classification System"@en ;

      aas:preferredName "Klassifikationssystem"@de ;

         aas:shortName "DocumentClassificationSystem"@en ;

         aas:datatype "STRING"^^xsd:string ;

         aas:definition "Eindeutige Kennung für ein Klassifikationssystem. Für Klassifikationen nach VDI 2770 muss 'VDI2770:2018' verwenden werden."@de ;

   ] ;

.


# ConceptDescription

<http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Organization/OrganizationName>  rdf:type aas:ConceptDescription ;

   aas:idShort "OrganizationName"^^xsd:string ;

   rdfs:label "OrganizationName"^^xsd:string ;

   aas:category "CONSTANT"^^xsd:string ;

   aas:identification <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Organization/OrganizationName> ;
```

```
    aas:hasDataSpecification [
        rdf:type aas:Reference ;
        aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
                aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
            aas:local "false"^^xsd:boolean ;
            aas:value <http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360> ;
            aas:idType aas:IRI_IDENTIFIER_TYPE ;
        ] ;
    ] ;
    aas:content [
        rdf:type aas:DataSpecificationIEC61360 ;
        aas:preferredName "gebräuchliche Bezeichnung für Organisation"@de ;
        aas:preferredName "organization name"@en ;
          aas:shortName "OrganizationName"@en ;
        aas:datatype "STRING"^^xsd:string ;
        aas:definition "Die gebräuchliche Bezeichnung für die Organisation."@de ;
    ] ;
.


# ConceptDescription
<http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Organization/OrganizationOfficialName>                rdf:type
aas:ConceptDescription ;
    aas:idShort "OrganizationName"^^xsd:string ;
    rdfs:label "OrganizationName"^^xsd:string ;
    aas:category "CONSTANT"^^xsd:string ;
    aas:identification <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Organization/OrganizationOfficialName> ;
    aas:hasDataSpecification [
        rdf:type aas:Reference ;
        aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
                aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
            aas:local "false"^^xsd:boolean ;
            aas:value <http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360> ;
            aas:idType aas:IRI_IDENTIFIER_TYPE ;
        ] ;
```

```
    ] ;
    aas:content [
        rdf:type aas:DataSpecificationIEC61360 ;
        aas:preferredName "offizieller Name der Organisation"@de ;
        aas:preferredName "official name of the organization"@en ;
            aas:shortName "OrganizationOfficialName"@en ;
        aas:datatype "STRING"^^xsd:string ;
        aas:definition "Der offizielle Namen der Organisation."@de ;
    ] ;
    aas:isCaseOf [
        rdf:type aas:Reference ;
        aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
            aas:type aas:CONCEPT_DESCRIPTION_IDENTIFIABLE_ELEMENT ;
            aas:local "true"^^xsd:boolean ;
            aas:value "0173-1#02-AAO677#002"^^xsd:string ;
            aas:idType aas:IRDI_IDENTIFIER_TYPE ;
        ] ;
    ];
.

# ConceptDescription
<http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentVersion> rdf:type aas:ConceptDescription ;
    aas:idShort "DocumentVersion"^^xsd:string ;
    rdfs:label "DocumentVersion"^^xsd:string ;
    aas:identification <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentVersion> ;
    aas:hasDataSpecification [
        rdf:type aas:Reference ;
        aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
                aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
            aas:local "false"^^xsd:boolean ;
            aas:value <http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360> ;
            aas:idType aas:IRI_IDENTIFIER_TYPE ;
        ] ;
```

```
  ] ;
  aas:content [
    rdf:type aas:DataSpecificationIEC61360 ;
      aas:preferredName "Document Version"@en ;
    aas:preferredName "Version des Dokuments"@de ;
      aas:shortName "DocumentVersion"@en ;
      aas:datatype "STRING"^^xsd:string ;
    aas:definition "Zu jedem Dokument muss eine Menge von mindestens einer Dokumentenversion existieren. Es können auch mehrere Dokumentenversionen ausgeliefert werden."@de ;
  ] ;
.


# ConceptDescription
<http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentVersion/Language> rdf:type aas:ConceptDescription ;
  aas:idShort "Language"^^xsd:string ;
  rdfs:label "Language"^^xsd:string ;
  aas:category "CONSTANT"^^xsd:string ;
  aas:identification <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentVersion/Language> ;
  aas:hasDataSpecification [
    rdf:type aas:Reference ;
    aas:key [
      a aas:Key ;
      aas:index "0"^^xsd:integer ;
        aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
      aas:local "false"^^xsd:boolean ;
      aas:value <http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360> ;
      aas:idType aas:IRI_IDENTIFIER_TYPE ;
    ] ;
  ] ;
  aas:content [
    rdf:type aas:DataSpecificationIEC61360 ;
    aas:preferredName "Sprache"@de ;
    aas:preferredName "Language"@en ;
      aas:shortName "Language"@en ;
    aas:datatype "STRING"^^xsd:string ;
    aas:definition "Eine Liste der im Dokument verwendeten Sprachen."@de ;
  ] ;
.
```

```
# ConceptDescription
<http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Description/Title> rdf:type aas:ConceptDescription ;
   aas:idShort "Title"^^xsd:string ;
   rdfs:label "Title"^^xsd:string ;
   aas:category "CONSTANT"^^xsd:string ;
   aas:identification <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Description/Title> ;
   aas:hasDataSpecification [
      rdf:type aas:Reference ;
      aas:key [
         a aas:Key ;
         aas:index "0"^^xsd:integer ;
            aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
         aas:local "false"^^xsd:boolean ;
         aas:value <http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360> ;
         aas:idType aas:IRI_IDENTIFIER_TYPE ;
      ] ;
   ] ;
   aas:content [
      rdf:type aas:DataSpecificationIEC61360 ;
      aas:preferredName "Titel"@de ;
      aas:preferredName "Title"@en ;
        aas:shortName "Title"@en ;
      aas:datatype "STRING_TRANSLATABLE"^^xsd:string ;
      aas:definition "Sprachabhängiger Titel des Dokuments."@de ;
   ] ;
.

# ConceptDescription
<http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/LifeCycleStatus/SetDate> rdf:type aas:ConceptDescription ;
   aas:idShort "Date"^^xsd:string ;
   rdfs:label "Date"^^xsd:string ;
   aas:identification <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/LifeCycleStatus/SetDate> ;
   aas:hasDataSpecification [
      rdf:type aas:Reference ;
      aas:key [
         a aas:Key ;
```

```
        aas:index "0"^^xsd:integer ;
            aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
        aas:local "false"^^xsd:boolean ;
        aas:value <http://admin-shell.io/DataSpecificationTtemplates/DataSpecificationIEC61360> ;
        aas:idType aas:IRI_IDENTIFIER_TYPE ;
      ] ;
   ] ;
   aas:content [
      rdf:type aas:DataSpecificationIEC61360 ;
        aas:preferredName "Set Date"@en ;
        aas:shortName "SetDate"@en ;
      aas:datatype "DATE"^^xsd:string ;
      aas:definition "Datum und Uhrzeit, an dem der Status festgelegt wurde. Es muss das Datumsformat „YYYY-MM-dd“ verwendet werden (Y = Jahr, M = Monat, d = Tag, siehe ISO 8601)."@de ;
   ] ;
.


# ConceptDescription
<http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentVersionId/Val> rdf:type aas:ConceptDescription ;
   aas:idShort "DocumentVersionIdValue"^^xsd:string ;
   rdfs:label "DocumentVersionIdValue"^^xsd:string ;
   aas:identification <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentVersionId/Val> ;
   aas:hasDataSpecification [
      rdf:type aas:Reference ;
      aas:key [
         a aas:Key ;
         aas:index "0"^^xsd:integer ;
            aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
         aas:local "false"^^xsd:boolean ;
         aas:value <http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360> ;
         aas:idType aas:IRI_IDENTIFIER_TYPE ;
      ] ;
   ] ;
   aas:content [
      rdf:type aas:DataSpecificationIEC61360 ;
        aas:preferredName "DocumentVersionId"@en ;
        aas:shortName "DocumentVersionId"@en ;
      aas:datatype "STRING"^^xsd:string ;
```

```
    aas:definition "Verschiedene Versionen eines Dokuments müssen eindeutig identifizierbar sein. Die DocumentVersionId stellt eine innerhalb einer Domäne eindeutige Versionsidentifikationsnummer dar."@de ;
  ] ;
.

# ConceptDescription
<http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/StoredDocumentRepresentation/DigitalFile> rdf:type aas:ConceptDescription ;
  aas:idShort "DigitalFile"^^xsd:string ;
  rdfs:label "DigitalFile"^^xsd:string ;
  aas:identification <http://www.vdi2770.com/blatt1/Entwurf/Okt18/cd/StoredDocumentRepresentation/DigitalFile> ;
  aas:hasDataSpecification [
    rdf:type aas:Reference ;
    aas:key [
      a aas:Key ;
      aas:index "0"^^xsd:integer ;
        aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
      aas:local "false"^^xsd:boolean ;
      aas:value <http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360> ;
      aas:idType aas:IRI_IDENTIFIER_TYPE ;
    ] ;
  ] ;
  aas:content [
    rdf:type aas:DataSpecificationIEC61360 ;
      aas:preferredName "Digital File"@en ;
      aas:shortName "DigitalFile"@en ;
    aas:datatype "FILE"^^xsd:string ;
    aas:definition "Eine Datei, die die DocumentVersion repräsentiert. Neben der obligatorischen PDF/A Datei können weitere Dateien angegeben werden."@de ;
  ] ;
.

# ConceptDescription
<http://i40.customer.com/type/1/1/F13E8576F6488342/MaxRotationSpeed> rdf:type aas:ConceptDescription ;
  aas:idShort "MaxRotationSpeed"^^xsd:string ;
  rdfs:label "MaxRotationSpeed"^^xsd:string ;
  aas:category "PROPERTY"^^xsd:string ;
  aas:identification [
```

```
      aas:id "0173-1#02-BAA120#008"^^xsd:string ;
      aas:idType aas:IRDI_IDENTIFIER_TYPE ;
   ] ;
   aas:administration [
      rdf:type aas:AdministrativeInformation ;
      aas:revision ""^^xsd:string ;
   ] ;
   aas:content [
      rdf:type aas:DataSpecificationIEC61360 ;
      aas:preferredName "max. Drehzahl"@de ;
      aas:preferredName "Max. rotation speed"@en ;
      aas:unit "1/min"^^xsd:string ;
      aas:unitId [
         rdf:type aas:Reference ;
         aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
                  aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
            aas:local "false"^^xsd:boolean ;
            aas:value "0173-1#05-AAA650#002"^^xsd:string ;
            aas:idType aas:IRDI_IDENTIFIER_TYPE ;
         ] ;
      ] ;
      aas:datatype "INTEGER_MEASURE"^^xsd:string ;
      aas:definition "Höchste zulässige Drehzahl, mit welcher der Motor oder die Speiseinheit betrieben werden
darf"@de ;
      aas:definition "Greatest permissible rotation speed with which the motor or feeding unit may be operated"@en ;
   ] ;
.


# ConceptDescription
<http://i40.customer.com/type/1/1/F13E8576F6488342/MaxTorque> rdf:type aas:ConceptDescription ;
   aas:idShort "MaxTorque"^^xsd:string ;
   rdfs:label "MaxTorque"^^xsd:string ;
   aas:category "PROPERTY"^^xsd:string ;
   aas:identification [
      aas:id "0173-1#02-BAE098#004"^^xsd:string ;
      aas:idType aas:IRDI_IDENTIFIER_TYPE ;
```

```
    ] ;
    aas:content [
        rdf:type aas:DataSpecificationIEC61360 ;
        aas:preferredName "Max. torque"@en ;
          aas:preferredName "max. Drehmoment"@de ;
        aas:unit "Nm"^^xsd:string ;
        aas:unitId [
            rdf:type aas:Reference ;
            aas:key [
                a aas:Key ;
                aas:index "0"^^xsd:integer ;
                     aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
                aas:local "false"^^xsd:boolean ;
                aas:value "0173-1#05-AAA212#003"^^xsd:string ;
                aas:idType aas:IRDI_IDENTIFIER_TYPE ;
            ] ;
        ] ;
        aas:datatype "REAL_MEASURE"^^xsd:string ;
        aas:definition "Größtes mechanisch zulässiges Drehmoment, welches der Motor an der Abtriebswelle abgeben
kann"@de ;
        aas:definition "Greatest permissible mechanical torque which the motor can pass on at the drive shaft"@en ;
    ] ;
.


# ConceptDescription
<http://customer.com/cd/1/1/18EBD56F6B43D895> rdf:type aas:ConceptDescription ;
    aas:idShort "RotationSpeed"^^xsd:string ;
    rdfs:label "RotationSpeed"^^xsd:string ;
    aas:category "PROPERTY"^^xsd:string ;
    aas:identification <http://customer.com/cd/1/1/18EBD56F6B43D895> ;
    aas:hasDataSpecification [
        rdf:type aas:Reference ;
        aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
                 aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
            aas:local "false"^^xsd:boolean ;
            aas:value <http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360> ;
```

```
        aas:idType aas:IRI_IDENTIFIER_TYPE ;
      ] ;
   ] ;
   aas:content [
      rdf:type aas:DataSpecificationIEC61360 ;
      aas:preferredName "Aktuelle Drehzahl"@de ;
      aas:preferredName "Actual rotation speed"@en ;
        aas:shortName "RotationSpeed"@en ;
      aas:unit "1/min"^^xsd:string ;
      aas:unitId [
         rdf:type aas:Reference ;
         aas:key [
            a aas:Key ;
            aas:index "0"^^xsd:integer ;
                aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
            aas:local "false"^^xsd:boolean ;
            aas:value "0173-1#05-AAA650#002"^^xsd:string ;
            aas:idType aas:IRDI_IDENTIFIER_TYPE ;
         ] ;
      ] ;
      aas:datatype "INTEGER_MEASURE"^^xsd:string ;
      aas:definition "Aktuelle Drehzahl, mit welcher der Motor oder die Speiseinheit betrieben wird"@de ;
      aas:definition "Actual rotation speed with which the motor or feeding unit is operated"@en ;
   ] ;
.

# ConceptDescription
<http://customer.com/cd/1/1/18EBD56F6B43D896> rdf:type aas:ConceptDescription ;
   aas:idShort "Torque"^^xsd:string ;
   rdfs:label "Torque"^^xsd:string ;
   aas:category "PROPERTY"^^xsd:string ;
   aas:identification <http://customer.com/cd/1/1/18EBD56F6B43D896> ;
   aas:hasDataSpecification [
      rdf:type aas:Reference ;
      aas:key [
         a aas:Key ;
         aas:index "0"^^xsd:integer ;
```

```
            aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
        aas:local "false"^^xsd:boolean ;
        aas:value <http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360> ;
        aas:idType aas:IRI_IDENTIFIER_TYPE ;
      ] ;
   ] ;
   aas:content [
      rdf:type aas:DataSpecificationIEC61360 ;
      aas:preferredName "Drehmoment"@de ;
      aas:preferredName "Torque"@en ;
      aas:shortName "Torque"@en ;
        aas:unit "Nm"^^xsd:string ;
      aas:unitId [
        rdf:type aas:Reference ;
        aas:key [
          a aas:Key ;
          aas:index "0"^^xsd:integer ;
          aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
              aas:local "false"^^xsd:boolean ;
          aas:value "0173-1#05-AAA212#003"^^xsd:string ;
          aas:idType aas:IRDI_IDENTIFIER_TYPE ;
        ] ;
      ] ;
      aas:datatype "REAL_MEASURE"^^xsd:string ;
      aas:definition "Aktuelles Drehmoment, welches der Motor an der Abtriebswelle abgibt"@de ;
      aas:definition "Actual mechanical torque which the motor passes on at the drive shaft"@en ;
   ] ;
.


# ConceptDescription
<http://i40.customer.com/type/1/1/F13E8576F6488342/0173-1#0173-1#02-BAE122#006>                rdf:type
aas:ConceptDescription ;
   aas:idShort "CoolingType"^^xsd:string ;
   rdfs:label "CoolingType"^^xsd:string ;
   aas:category "PROPERTY"^^xsd:string ;
   aas:identification [
      aas:id "0173-1#02-BAE122#006"^^xsd:string ;
      aas:idType aas:IRDI_IDENTIFIER_TYPE ;
```

```
  ] ;
  aas:content [
    rdf:type aas:DataSpecificationIEC61360 ;
    aas:preferredName "Art der Kühlung"@de ;
    aas:preferredName "Cooling type"@en ;
    aas:datatype "STRING"^^xsd:string ;
    aas:definition "Zusammenfassung verschiedener Kühlarten, um für Suchmerkmale zu einer begrenzten Auswahl zu kommen"@de ;
    aas:definition "Summary of various types of cooling, for use as search criteria that limit a selection"@en ;
  ] ;
.

# ConceptDescription
<http://i40.customer.com/type/1/1/F13E8576F6488342/BAB657> rdf:type aas:ConceptDescription ;
  aas:idShort "BAB657"^^xsd:string ;
  rdfs:label "BAB657"^^xsd:string ;
  aas:category "VALUE"^^xsd:string ;
  aas:identification [
    aas:id "0173-1#07-BAB657#003"^^xsd:string ;
    aas:idType aas:IRDI_IDENTIFIER_TYPE ;
  ] ;
  aas:hasDataSpecification [
    rdf:type aas:Reference ;
    aas:key [
      a aas:Key ;
      aas:index "0"^^xsd:integer ;
      aas:type aas:GLOBAL_REFERENCE_KEY_ELEMENT ;
        aas:local "false"^^xsd:boolean ;
      aas:value <http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360> ;
      aas:idType aas:IRI_IDENTIFIER_TYPE ;
    ] ;
  ] ;
  aas:content [
    rdf:type aas:DataSpecificationIEC61360 ;
    aas:preferredName "offener Kreis, Fremdkühlung"@de ;
    aas:preferredName "open circuit, external cooling"@en ;
    aas:datatype "STRING"^^xsd:string ;
  ] ;
```

.

Annex H. AutomationML and complete example

i. Introduction

[37] is the AutomationML application recommendation for the asset administration shell (AR AAS). This annex is just for information.

ii. AutomationML Libraries for Asset Administration Shell

[37] is the AutomationML application recommendation for the asset administration shell (AR AAS) and contains the libraries for the asset administration shell roles, interface classes and system unit classes. This annex is just for information.

```
<CAEXFile FileName="AssetAdministrationShellLib.aml" SchemaVersion="2.15" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="CAEX_ClassModel_V2.15.xsd">
 <AdditionalInformation AutomationMLVersion="2.0" />
 <AdditionalInformation DocumentVersions="Recommendations">
  <Document DocumentIdentifier="AR AAS" Version="1.0.0" />
 </AdditionalInformation>
 <AdditionalInformation DocumentVersions="Recommendations">
  <Document DocumentIdentifier="BPR EDRef" Version="1.0.0" />
 </AdditionalInformation>
 <AdditionalInformation>
  <WriterHeader>
   <WriterName>AutomationML Editor</WriterName>
   <WriterID>916578CA-FE0D-474E-A4FC-9E1719892369</WriterID>
   <WriterVendor>AutomationML e.V.</WriterVendor>
   <WriterVendorURL>www.AutomationML.org</WriterVendorURL>
   <WriterVersion>5.2.7.0</WriterVersion>
   <WriterRelease>5.2.7.0</WriterRelease>
   <LastWritingDateTime>2019-11-20T12:12:02.7871841</LastWritingDateTime>
   <WriterProjectTitle>Application Recommendation Asset Administration Shell</WriterProjectTitle>
   <WriterProjectID>AR AAS</WriterProjectID>
  </WriterHeader>
 </AdditionalInformation>
 <InterfaceClassLib Name="AssetAdministrationShellInterfaceClassLib">
  <Description>Interface Class Library according to Details of the Asset Administration Shell V2.0.</Description>
  <Version>1.0.0</Version>
  <InterfaceClass Name="FileDataReference" RefBaseClassPath="AutomationMLBPRInterfaceClassLib/ExternalDataReference">
   <Description>A FileDataReference represents the address to a File. FileDataReference is derived from the AutomationML Interface Class ExternalDataReference that is defined in AutomationML BPR_005E_ExternalDataReference_v1.0.0_2:The interface class “ExternalDataReference” shall be used in order to reference external documents out of the scope of AutomationML.</Description>
  </InterfaceClass>
  <InterfaceClass Name="ReferableReference" RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface">
   <Description>Reference to any other referable element of the same of any other AAS or a reference to an external object or entity. For local references inside the same Asset Administration Shell an InternalLink between two objects with this interface "ReferableReference" shall be set. In this case the attribute value has to be empty. For references between different Asset Administration Shells or external objects or entities the attribute value shall be used and no InternalLink shall be set.</Description>
```

```
<Attribute Name="value" AttributeDataType="xs:string">
<Description>Reference to any other referable element of any other AAS or a reference to an external object or entity. Note: For references to any other referable element of the same AAS InternalLinks are used and this attribute value shall be empty.</Description>
<RefSemantic CorrespondingAttributePath="AAS:ReferenceElement/value" />
</Attribute>
</InterfaceClass>
</InterfaceClassLib>
<RoleClassLib Name="AssetAdministrationShellRoleClassLib">
<Description>Role Class Library according to Details of the Asset Administration Shell V2.0.</Description>
<Version>1.0.0</Version>
<RoleClass Name="AssetAdministrationShell" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
<Description>An Asset Administration Shell.</Description>
<Attribute Name="idShort" AttributeDataType="xs:string">
<Description>Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the “BrowserPath” in OPC UA.</Description>
<RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
</Attribute>
<Attribute Name="category" AttributeDataType="xs:string">
<Description>The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints.</Description>
<Value></Value>
<RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
</Attribute>
<Attribute Name="description" AttributeDataType="xs:string">
<Description>Description or comments on the element. The description can be provided in several languages.</Description>
<Value></Value>
<RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
<Attribute Name="aml-lang=EN" AttributeDataType="xs:string">
<Value></Value>
</Attribute>
<Attribute Name="aml-lang=DE" AttributeDataType="xs:string">
<Value></Value>
</Attribute>
</Attribute>
<Attribute Name="identification">
<Description>Abstract attribute class for identification. Has the subattributes id and idType.</Description>
<RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
<Attribute Name="id" AttributeDataType="xs:string">
<Description>Identifier of the element. Its type is defined in idType. Id is a subproperty of identification.</Description>
<Value></Value>
<RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
```

```
    </Attribute>
    <Attribute Name="idType" AttributeDataType="xs:string">
     <Description>Type of the Identifier, e.g. IRI, IRDI etc. The supported Identifier types are defined in the enumeration “IdentifierType”. IdType is a subproperty of identification.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
    </Attribute>
   </Attribute>
   <Attribute Name="administration">
    <Description>Abstract attribute for administration. Has the subattributes revision and version.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Identifiable/administration" />
    <Attribute Name="revision" AttributeDataType="xs:string">
     <Description>Revision of the element. Constraint AASd-005: A revision requires a version. This means, if there is no version there is no revision neither. Revision is a subproperty of administration.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:AdministrativeInformation/revision" />
    </Attribute>
    <Attribute Name="version" AttributeDataType="xs:string">
     <Description>Version of the element. Version is a subproperty of administration.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:AdministrativeInformation/version" />
    </Attribute>
   </Attribute>
   <Attribute Name="dataSpecification" AttributeDataType="xs:string">
    <Description>Global reference to the data specification template used by the element.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasDataSpecification/dataSpecification" />
   </Attribute>
   <Attribute Name="derivedFrom" AttributeDataType="xs:string">
    <Description>The derivedFrom attribute is used to establish a relationship between two Asset Administration Shells that are derived from each other. </Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:AssetAdministrationShell/derivedFrom" />
   </Attribute>
  </RoleClass>
  <RoleClass Name="Asset" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
   <Description>An Asset describes meta data of an asset that is represented by an AAS. The asset may either represent an asset type or an asset instance. The asset has a globally unique identifier plus – if needed – additional domain specific (proprietary) identifiers.</Description>
   <Attribute Name="idShort" AttributeDataType="xs:string">
    <Description>Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the “BrowserPath” in OPC UA.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
```

```
    </Attribute>
    <Attribute Name="category" AttributeDataType="xs:string">
     <Description>The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
    </Attribute>
    <Attribute Name="description" AttributeDataType="xs:string">
     <Description>Description or comments on the element. The description can be provided in several languages.</Description>
     <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
     <Attribute Name="aml-lang=EN" />
     <Attribute Name="aml-lang=DE" />
    </Attribute>
    <Attribute Name="identification">
     <Description>Abstract attribute class for identification. Has the subattributes id and idType.</Description>
     <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
     <Attribute Name="id" AttributeDataType="xs:string">
      <Description>Identifier of the element. Its type is defined in idType.</Description>
      <Value></Value>
      <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
     </Attribute>
     <Attribute Name="idType" AttributeDataType="xs:string">
      <Description>Type of the Identifier, e.g. IRI, IRDI etc. The supported Identifier types are defined in the enumeration “IdentifierType”. IdType is a subproperty of identification.</Description>
      <Value></Value>
      <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
     </Attribute>
    </Attribute>
    <Attribute Name="administration">
     <Description>Abstract attribute for administration. Has the subattributes revision and version.</Description>
     <RefSemantic CorrespondingAttributePath="AAS:Identifiable/administration" />
     <Attribute Name="revision" AttributeDataType="xs:string">
      <Description>Revision of the element. Constraint AASd-005: A revision requires a version. This means, if there is no version there is no revision neither. Revision is a subproperty of administration.</Description>
      <Value></Value>
      <RefSemantic CorrespondingAttributePath="AAS:AdministrativeInformation/revision" />
     </Attribute>
     <Attribute Name="version" AttributeDataType="xs:string">
      <Description>Version of the element. Version is a subproperty of administration.</Description>
      <Value></Value>
      <RefSemantic CorrespondingAttributePath="AAS:AdministrativeInformation/version" />
     </Attribute>
    </Attribute>
    <Attribute Name="dataSpecification" AttributeDataType="xs:string">
```

```
    <Description>Global reference to the data specification template used by the element.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasDataSpecification/dataSpecification" />
   </Attribute>
   <Attribute Name="kind" AttributeDataType="xs:string">
    <Description>Kind of the asset: either type or instance.</Description>
    <DefaultValue>Instance</DefaultValue>
    <Value>Instance</Value>
    <RefSemantic CorrespondingAttributePath="AAS:Asset/kind" />
    <Constraint Name="">
     <NominalScaledType>
      <RequiredValue>Instance</RequiredValue>
      <RequiredValue>Type</RequiredValue>
     </NominalScaledType>
    </Constraint>
   </Attribute>
  </RoleClass>
  <RoleClass Name="Submodel" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
   <Description>A Submodel defines a specific aspect of the asset represented by the AAS. A submodel is used to structure the virtual representation and technical functionality of an Administration Shell into distinguishable parts. Each submodel refers to a well-defined domain or subject matter. Submodels can become standardized and thus become submodels types. Submodels can have different life-cycles.</Description>
   <Attribute Name="idShort" AttributeDataType="xs:string">
    <Description>Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the “BrowserPath” in OPC UA.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
   </Attribute>
   <Attribute Name="category" AttributeDataType="xs:string">
    <Description>The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
   </Attribute>
   <Attribute Name="description" AttributeDataType="xs:string">
    <Description>Description or comments on the element. The description can be provided in several languages.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
    <Attribute Name="aml-lang=EN" />
    <Attribute Name="aml-lang=DE" />
   </Attribute>
   <Attribute Name="identification">
    <Description>Abstract attribute class for identification. Has the subattributes id and idType.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
```

```
    <Attribute Name="id" AttributeDataType="xs:string">
     <Description>Identifier of the element. Its type is defined in idType. Id is a subproperty of identification.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
    </Attribute>
    <Attribute Name="idType">
     <Description>Type of the Identifier, e.g. IRI, IRDI etc. The supported Identifier types are defined in the enumeration “IdentifierType”. IdType is a subproperty of identification.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
    </Attribute>
   </Attribute>
   <Attribute Name="administration">
    <Description>Abstract attribute for administration. Has the subattributes revision and version.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Identifiable/administration" />
    <Attribute Name="revision" AttributeDataType="xs:string">
     <Description>Revision of the element. Constraint AASd-005: A revision requires a version. This means, if there is no version there is no revision neither. Revision is a subproperty of administration.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:AdministrativeInformation/revision" />
    </Attribute>
    <Attribute Name="version" AttributeDataType="xs:string">
     <Description>Version of the element. Version of the element. Version is a subproperty of administration.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:AdministrativeInformation/version" />
    </Attribute>
   </Attribute>
   <Attribute Name="dataSpecification" AttributeDataType="xs:string">
    <Description>Global reference to the data specification template used by the element.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasDataSpecification/dataSpecification" />
   </Attribute>
   <Attribute Name="kind" AttributeDataType="xs:string">
    <Description>Kind of the element: either template or instance.</Description>
    <DefaultValue>Instance</DefaultValue>
    <Value>Instance</Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
    <Constraint Name="">
     <NominalScaledType>
      <RequiredValue>Instance</RequiredValue>
      <RequiredValue>Type</RequiredValue>
     </NominalScaledType>
    </Constraint>
```

```
    </Attribute>
    <Attribute Name="semanticId" AttributeDataType="xs:string">
     <Description>Description or comments on the element. The description can be provided in several languages. This attribute has the name of the label and has a value with the label written in the default language. The individual languages are modelled as child attributes. The names of the child attributes are the prefix “aml-lang=” with the expression of the language in compliance with RFC5646. At it, the values of the child attributes are the labels within the respective language.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
    </Attribute>
    <Attribute Name="qualifier:[TYPE]=[VALUE]" AttributeDataType="xs:string">
     <Description>A qualifier is a type-value-pair that makes additional statements w.r.t. the value of the element. [TYPE] is the value of the attribute type and [VALUE] is the value of the attribute value.</Description>
     <RefSemantic CorrespondingAttributePath="AAS:Qualifiable/qualifier" />
     <Attribute Name="type" AttributeDataType="xs:string">
      <Description>The type describes the type of the qualifier that is applied to the element.</Description>
      <Value></Value>
      <RefSemantic CorrespondingAttributePath="AAS:Qualifier/type" />
     </Attribute>
     <Attribute Name="value" AttributeDataType="xs:string">
      <Description>The qualifier value is the value of the qualifier. Constraint AASd-006: if both, the value and the valueId are present then the value needs to be identical to the value of the referenced coded value in valueId.</Description>
      <Value></Value>
      <RefSemantic CorrespondingAttributePath="AAS:Qualifier/value" />
     </Attribute>
     <Attribute Name="valueId" AttributeDataType="xs:string">
      <Description>Reference to the global unqiue id of a coded value.</Description>
      <Value></Value>
      <RefSemantic CorrespondingAttributePath="AAS:Qualifier/valueId" />
     </Attribute>
    </Attribute>
   </RoleClass>
   <RoleClass Name="SubmodelElementCollection" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
    <Description>A submodel element collection is a set or list of submodel elements. </Description>
    <Attribute Name="idShort" AttributeDataType="xs:string">
     <Description>Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the “BrowserPath” in OPC UA.</Description>
     <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
    </Attribute>
    <Attribute Name="category" AttributeDataType="xs:string">
     <Description>The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints.</Description>
     <Value></Value>
```

```
    <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
   </Attribute>
   <Attribute Name="description" AttributeDataType="xs:string">
    <Description>Description or comments on the element. The description can be provided in several languages.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
    <Attribute Name="aml-lang=EN" />
    <Attribute Name="aml-lang=DE" />
   </Attribute>
   <Attribute Name="dataSpecification" AttributeDataType="xs:string">
    <Description>Global reference to the data specification template used by the element.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasDataSpecification/dataSpecification" />
   </Attribute>
   <Attribute Name="kind" AttributeDataType="xs:string">
    <Description>Kind of the element: either template or instance.</Description>
    <DefaultValue>Instance</DefaultValue>
    <Value>Instance</Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
    <Constraint Name="">
     <NominalScaledType>
      <RequiredValue>Instance</RequiredValue>
      <RequiredValue>Type</RequiredValue>
     </NominalScaledType>
    </Constraint>
   </Attribute>
   <Attribute Name="semanticId" AttributeDataType="xs:string">
    <Description>Description or comments on the element. The description can be provided in several languages. This attribute has the name of the label and has a value with the label written in the default language. The individual languages are modelled as child attributes. The names of the child attributes are the prefix “aml-lang=” with the expression of the language in compliance with RFC5646. At it, the values of the child attributes are the labels within the respective language.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
   </Attribute>
   <Attribute Name="qualifier:[TYPE]=[VALUE]" AttributeDataType="xs:string">
    <Description>A qualifier is a type-value-pair that makes additional statements w.r.t. the value of the element. [TYPE] is the value of the attribute type and [VALUE] is the value of the attribute value.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Qualifiable/qualifier" />
    <Attribute Name="type" AttributeDataType="xs:string">
     <Description>The type describes the type of the qualifier that is applied to the element.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Qualifier/type" />
    </Attribute>
    <Attribute Name="value" AttributeDataType="xs:string">
```

```
    <Description>The qualifier value is the value of the qualifier. Constraint AASd-006: if both, the value and the valueId are present then the value needs to be identical to the value of the referenced coded value in valueId.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:Qualifier/value" />
   </Attribute>
   <Attribute Name="valueId" AttributeDataType="xs:string">
    <Description>Reference to the global unqiue id of a coded value.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:Qualifier/valueId" />
   </Attribute>
  </Attribute>
  <Attribute Name="allowDuplicates" AttributeDataType="xs:boolean">
   <Description>If allowDuplicates=true, then it is allowed that the collection contains the same element several times.</Description>
   <DefaultValue>false</DefaultValue>
   <RefSemantic CorrespondingAttributePath="AAS:SubmodulElementCollection/allowDuplicates" />
  </Attribute>
  <Attribute Name="ordered" AttributeDataType="xs:boolean">
   <Description>If ordered=false, then the elements in the property collection are not ordered. If ordered=true then the elements in the collection are ordered. Default = false. Note: An ordered submodel element collection is typically implemented as an indexed array.</Description>
   <DefaultValue>false</DefaultValue>
   <RefSemantic CorrespondingAttributePath="AAS:SubmodulElementCollection/ordered" />
  </Attribute>
  <Attribute Name="value" AttributeDataType="xs:string">
   <Description>Submodel element contained in the collection.</Description>
   <RefSemantic CorrespondingAttributePath="AAS:SubmodulElementCollection/value" />
  </Attribute>
 </RoleClass>
 <RoleClass Name="Blob" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
  <Description>A BLOB is a data element that represents a file that is contained with its source code in the value attribute.</Description>
  <Attribute Name="idShort" AttributeDataType="xs:string">
   <Description>Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the “BrowserPath” in OPC UA.</Description>
   <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
  </Attribute>
  <Attribute Name="category" AttributeDataType="xs:string">
   <Description>The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints.</Description>
   <Value></Value>
   <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
  </Attribute>
  <Attribute Name="description" AttributeDataType="xs:string">
```

```
    <Description>Description or comments on the element. The description can be provided in several languages.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
    <Attribute Name="aml-lang=EN" />
    <Attribute Name="aml-lang=DE" />
   </Attribute>
   <Attribute Name="dataSpecification" AttributeDataType="xs:string">
    <Description>Global reference to the data specification template used by the element.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasDataSpecification/dataSpecification" />
   </Attribute>
   <Attribute Name="kind" AttributeDataType="xs:string">
    <Description>Kind of the element: either template or instance.</Description>
    <DefaultValue>Instance</DefaultValue>
    <Value>Instance</Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
    <Constraint Name="">
     <NominalScaledType>
      <RequiredValue>Instance</RequiredValue>
      <RequiredValue>Type</RequiredValue>
     </NominalScaledType>
    </Constraint>
   </Attribute>
   <Attribute Name="semanticId" AttributeDataType="xs:string">
    <Description>Description or comments on the element. The description can be provided in several languages. This attribute has the name of the label and has a value with the label written in the default language. The individual languages are modelled as child attributes. The names of the child attributes are the prefix “aml-lang=” with the expression of the language in compliance with RFC5646. At it, the values of the child attributes are the labels within the respective language.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
   </Attribute>
   <Attribute Name="qualifier:[TYPE]=[VALUE]" AttributeDataType="xs:string">
    <Description>A qualifier is a type-value-pair that makes additional statements w.r.t. the value of the element. [TYPE] is the value of the attribute type and [VALUE] is the value of the attribute value.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Qualifiable/qualifier" />
    <Attribute Name="type" AttributeDataType="xs:string">
     <Description>The type describes the type of the qualifier that is applied to the element.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Qualifier/type" />
    </Attribute>
    <Attribute Name="value" AttributeDataType="xs:string">
     <Description>The qualifier value is the value of the qualifier. Constraint AASd-006: if both, the value and the valueId are present then the value needs to be identical to the value of the referenced coded value in valueId.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Qualifier/value" />
```

```
    </Attribute>
    <Attribute Name="valueId" AttributeDataType="xs:string">
     <Description>Reference to the global unqiue id of a coded value.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Qualifier/valueId" />
    </Attribute>
   </Attribute>
   <Attribute Name="mimeType" AttributeDataType="xs:string">
    <Description>Mime type of the content of the BLOB. The mime type states which file extension the file has. Valid values are e.g. “application/json”, “application/xls”, ”image/jpg”. The allowed values are defined as in RFC2046.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:Blob/mimeType" />
   </Attribute>
   <Attribute Name="value" AttributeDataType="xs:string">
    <Description>The value of the BLOB instance of a blob data element. Note: In contrast to the file property the file content is stored directly as value in the Blob data element.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:Blob/value" />
   </Attribute>
  </RoleClass>
  <RoleClass Name="Capability" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
   <Description>A capability is the implementation-independent description of the potential of an asset to achieve a certain effect in the physical or virtual world. </Description>
   <Attribute Name="idShort" AttributeDataType="xs:string">
    <Description>Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the “BrowserPath” in OPC UA.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
   </Attribute>
   <Attribute Name="category" AttributeDataType="xs:string">
    <Description>The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
   </Attribute>
   <Attribute Name="description" AttributeDataType="xs:string">
    <Description>Description or comments on the element. The description can be provided in several languages.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
    <Attribute Name="aml-lang=EN" />
    <Attribute Name="aml-lang=DE" />
   </Attribute>
   <Attribute Name="dataSpecification" AttributeDataType="xs:string">
    <Description>Global reference to the data specification template used by the element.</Description>
```

```
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasDataSpecification/dataSpecification" />
   </Attribute>
   <Attribute Name="kind" AttributeDataType="xs:string">
    <Description>Kind of the element: either template or instance.</Description>
    <DefaultValue>Instance</DefaultValue>
    <Value>Instance</Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
    <Constraint Name="">
     <NominalScaledType>
      <RequiredValue>Instance</RequiredValue>
      <RequiredValue>Type</RequiredValue>
     </NominalScaledType>
    </Constraint>
   </Attribute>
   <Attribute Name="semanticId" AttributeDataType="xs:string">
    <Description>Description or comments on the element. The description can be provided in several languages. This attribute has the name of the label and has a value with the label written in the default language. The individual languages are modelled as child attributes. The names of the child attributes are the prefix “aml-lang=” with the expression of the language in compliance with RFC5646. At it, the values of the child attributes are the labels within the respective language.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
   </Attribute>
   <Attribute Name="qualifier:[TYPE]=[VALUE]" AttributeDataType="xs:string">
    <Description>A qualifier is a type-value-pair that makes additional statements w.r.t. the value of the element. [TYPE] is the value of the attribute type and [VALUE] is the value of the attribute value.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Qualifiable/qualifier" />
    <Attribute Name="type" AttributeDataType="xs:string">
     <Description>The type describes the type of the qualifier that is applied to the element.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Qualifier/type" />
    </Attribute>
    <Attribute Name="value" AttributeDataType="xs:string">
     <Description>The qualifier value is the value of the qualifier. Constraint AASd-006: if both, the value and the valueId are present then the value needs to be identical to the value of the referenced coded value in valueId.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Qualifier/value" />
    </Attribute>
    <Attribute Name="valueId" AttributeDataType="xs:string">
     <Description>Reference to the global unqiue id of a coded value.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Qualifier/valueId" />
    </Attribute>
   </Attribute>
```

```
</RoleClass>
<RoleClass Name="File" RefBaseClassPath="AutomationMLBPRRoleClassLib/ExternalData">
<Description>A role class for a File that a data element that represents an address to a file. It is derived from the AutomationML role class ExternalData that is an role type for a document type and the base class for all document type roles. It describes different document types. ExternalData is defined in AutomationML BPR_005E_ExternalDataReference_v1.0.0_2.</Description>
<Attribute Name="idShort" AttributeDataType="xs:string">
<Description>Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the “BrowserPath” in OPC UA.</Description>
<RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
</Attribute>
<Attribute Name="category" AttributeDataType="xs:string">
<Description>The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints.</Description>
<Value></Value>
<RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
</Attribute>
<Attribute Name="description" AttributeDataType="xs:string">
<Description>Description or comments on the element. The description can be provided in several languages.</Description>
<RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
<Attribute Name="aml-lang=EN" />
<Attribute Name="aml-lang=DE" />
</Attribute>
<Attribute Name="dataSpecification" AttributeDataType="xs:string">
<Description>Global reference to the data specification template used by the element.</Description>
<Value></Value>
<RefSemantic CorrespondingAttributePath="AAS:HasDataSpecification/dataSpecification" />
</Attribute>
<Attribute Name="kind" AttributeDataType="xs:string">
<Description>Kind of the element: either template or instance.</Description>
<DefaultValue>Instance</DefaultValue>
<Value>Instance</Value>
<RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
<Constraint Name="">
<NominalScaledType>
<RequiredValue>Instance</RequiredValue>
<RequiredValue>Type</RequiredValue>
</NominalScaledType>
</Constraint>
</Attribute>
<Attribute Name="semanticId" AttributeDataType="xs:string">
<Description>Description or comments on the element. The description can be provided in several languages. This attribute has the name of the label and has a value with the label written in the default language. The individual languages are modelled as child attributes. The
```

names of the child attributes are the prefix “aml-lang=” with the expression of the language in compliance with RFC5646. At it, the values of the child attributes are the labels within the respective language.</Description>

<Value></Value>

<RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />

</Attribute>

<Attribute Name="qualifier:[TYPE]=[VALUE]" AttributeDataType="xs:string">

<Description>A qualifier is a type-value-pair that makes additional statements w.r.t. the value of the element. [TYPE] is the value of the attribute type and [VALUE] is the value of the attribute value.</Description>

<RefSemantic CorrespondingAttributePath="AAS:Qualifiable/qualifier" />

<Attribute Name="type" AttributeDataType="xs:string">

<Description>The type describes the type of the qualifier that is applied to the element.</Description>

<Value></Value>

<RefSemantic CorrespondingAttributePath="AAS:Qualifier/type" />

</Attribute>

<Attribute Name="value" AttributeDataType="xs:string">

<Description>The qualifier value is the value of the qualifier. Constraint AASd-006: if both, the value and the valueId are present then the value needs to be identical to the value of the referenced coded value in valueId.</Description>

<Value></Value>

<RefSemantic CorrespondingAttributePath="AAS:Qualifier/value" />

</Attribute>

<Attribute Name="valueId" AttributeDataType="xs:string">

<Description>Reference to the global unqiue id of a coded value.</Description>

<Value></Value>

<RefSemantic CorrespondingAttributePath="AAS:Qualifier/valueId" />

</Attribute>

</Attribute>

<ExternalInterface Name="FileDataReference" ID="c7a2932e-7f90-46d3-be64-0408a927c89d" RefBaseClassPath="AssetAdministrationShellInterfaceClassLib/FileDataReference">

<Attribute Name="refURI" AttributeDataType="xs:anyURI">

<Description>The attribute refURI is an IRI that can represent an absolute or relative path to an L document. An added fragment (with #) references inside the document</Description>

</Attribute>

<Attribute Name="MIMEType" AttributeDataType="xs:string">

<Description>Mime type of the content of the File.</Description>

</Attribute>

</ExternalInterface>

</RoleClass>

<RoleClass Name="Property" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">

<Description>A property is a data element that has a single value.</Description>

<Attribute Name="idShort" AttributeDataType="xs:string">

<Description>Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the “BrowserPath” in OPC UA.</Description>

```
    <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
   </Attribute>
   <Attribute Name="category" AttributeDataType="xs:string">
    <Description>The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
   </Attribute>
   <Attribute Name="description" AttributeDataType="xs:string">
    <Description>Description or comments on the element. The description can be provided in several languages.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
    <Attribute Name="aml-lang=EN" />
    <Attribute Name="aml-lang=DE" />
   </Attribute>
   <Attribute Name="dataSpecification" AttributeDataType="xs:string">
    <Description>Global reference to the data specification template used by the element.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasDataSpecification/dataSpecification" />
   </Attribute>
   <Attribute Name="kind" AttributeDataType="xs:string">
    <Description>Kind of the element: either template or instance.</Description>
    <DefaultValue>Instance</DefaultValue>
    <Value>Instance</Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
    <Constraint Name="">
     <NominalScaledType>
      <RequiredValue>Instance</RequiredValue>
      <RequiredValue>Type</RequiredValue>
     </NominalScaledType>
    </Constraint>
   </Attribute>
   <Attribute Name="semanticId" AttributeDataType="xs:string">
    <Description>Description or comments on the element. The description can be provided in several languages. This attribute has the name of the label and has a value with the label written in the default language. The individual languages are modelled as child attributes. The names of the child attributes are the prefix “aml-lang=” with the expression of the language in compliance with RFC5646. At it, the values of the child attributes are the labels within the respective language.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
   </Attribute>
   <Attribute Name="qualifier:[TYPE]=[VALUE]" AttributeDataType="xs:string">
    <Description>A qualifier is a type-value-pair that makes additional statements w.r.t. the value of the element. [TYPE] is the value of the attribute type and [VALUE] is the value of the attribute value.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Qualifiable/qualifier" />
    <Attribute Name="type" AttributeDataType="xs:string">
```

```
    <Description>The type describes the type of the qualifier that is applied to the element.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:Qualifier/type" />
   </Attribute>
   <Attribute Name="value" AttributeDataType="xs:string">
    <Description>The qualifier value is the value of the qualifier. Constraint AASd-006: if both, the value and the valueId are present then the value needs to be identical to the value of the referenced coded value in valueId.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:Qualifier/value" />
   </Attribute>
   <Attribute Name="valueId" AttributeDataType="xs:string">
    <Description>Reference to the global unqiue id of a coded value.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:Qualifier/valueId" />
   </Attribute>
  </Attribute>
  <Attribute Name="value" AttributeDataType="xs:string">
   <Description>The value of the property instance. </Description>
   <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
  </Attribute>
  <Attribute Name="valueId" AttributeDataType="xs:string">
   <Description>Reference to the global unique id if a coded value.</Description>
   <RefSemantic CorrespondingAttributePath="AAS:Property/valueId" />
  </Attribute>
 </RoleClass>
 <RoleClass Name="ReferenceElement" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
  <Description>A reference element is a data element that defines a logical reference to another element within the same or another AAS or a reference to an external object or entity.</Description>
  <Attribute Name="idShort" AttributeDataType="xs:string">
   <Description>Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the “BrowserPath” in OPC UA.</Description>
   <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
  </Attribute>
  <Attribute Name="category" AttributeDataType="xs:string">
   <Description>The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints.</Description>
   <Value></Value>
   <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
  </Attribute>
  <Attribute Name="description" AttributeDataType="xs:string">
   <Description>Description or comments on the element. The description can be provided in several languages.</Description>
   <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
```

```
    <Attribute Name="aml-lang=EN" />
    <Attribute Name="aml-lang=DE" />
   </Attribute>
   <Attribute Name="dataSpecification" AttributeDataType="xs:string">
    <Description>Global reference to the data specification template used by the element.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasDataSpecification/dataSpecification" />
   </Attribute>
   <Attribute Name="kind" AttributeDataType="xs:string">
    <Description>Kind of the element: either template or instance.</Description>
    <DefaultValue>Instance</DefaultValue>
    <Value>Instance</Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
    <Constraint Name="">
     <NominalScaledType>
      <RequiredValue>Instance</RequiredValue>
      <RequiredValue>Type</RequiredValue>
     </NominalScaledType>
    </Constraint>
   </Attribute>
   <Attribute Name="semanticId" AttributeDataType="xs:string">
    <Description>Description or comments on the element. The description can be provided in several languages. This attribute has the name of the label and has a value with the label written in the default language. The individual languages are modelled as child attributes. The names of the child attributes are the prefix “aml-lang=” with the expression of the language in compliance with RFC5646. At it, the values of the child attributes are the labels within the respective language.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
   </Attribute>
   <Attribute Name="qualifier:[TYPE]=[VALUE]" AttributeDataType="xs:string">
    <Description>A qualifier is a type-value-pair that makes additional statements w.r.t. the value of the element. [TYPE] is the value of the attribute type and [VALUE] is the value of the attribute value.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Qualifiable/qualifier" />
    <Attribute Name="type" AttributeDataType="xs:string">
     <Description>The type describes the type of the qualifier that is applied to the element.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Qualifier/type" />
    </Attribute>
    <Attribute Name="value" AttributeDataType="xs:string">
     <Description>The qualifier value is the value of the qualifier. Constraint AASd-006: if both, the value and the valueId are present then the value needs to be identical to the value of the referenced coded value in valueId.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Qualifier/value" />
    </Attribute>
    <Attribute Name="valueId" AttributeDataType="xs:string">
```

```
    <Description>Reference to the global unqiue id of a coded value.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:Qualifier/valueId" />
   </Attribute>
  </Attribute>
  <ExternalInterface Name="ReferableReference" ID="cedd4784-8dd9-4b0d-bb16-59a82ab845da" RefBaseClassPath="AssetAdministrationShellInterfaceClassLib/ReferableReference">
   <Attribute Name="value" AttributeDataType="xs:string">
    <Description>Reference to any other referable element of any other AAS or a reference to an external object or entity. Note: For references to any other referable elment of the same AAS InternalLinks are used and this attribute value shall be empty.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:ReferenceElement/value" />
   </Attribute>
  </ExternalInterface>
 </RoleClass>
 <RoleClass Name="RelationshipElement" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
  <Description>A relationship element is used to define a relationship between two referable elements.</Description>
  <Attribute Name="idShort" AttributeDataType="xs:string">
   <Description>Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the “BrowserPath” in OPC UA.</Description>
   <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
  </Attribute>
  <Attribute Name="category" AttributeDataType="xs:string">
   <Description>The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints.</Description>
   <Value></Value>
   <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
  </Attribute>
  <Attribute Name="description" AttributeDataType="xs:string">
   <Description>Description or comments on the element. The description can be provided in several languages.</Description>
   <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
   <Attribute Name="aml-lang=EN" />
   <Attribute Name="aml-lang=DE" />
  </Attribute>
  <Attribute Name="dataSpecification" AttributeDataType="xs:string">
   <Description>Global reference to the data specification template used by the element.</Description>
   <Value></Value>
   <RefSemantic CorrespondingAttributePath="AAS:HasDataSpecification/dataSpecification" />
  </Attribute>
  <Attribute Name="kind" AttributeDataType="xs:string">
   <Description>Kind of the element: either template or instance.</Description>
   <DefaultValue>Instance</DefaultValue>
   <Value>Instance</Value>
```

```
    <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
    <Constraint Name="">
     <NominalScaledType>
      <RequiredValue>Instance</RequiredValue>
      <RequiredValue>Type</RequiredValue>
     </NominalScaledType>
    </Constraint>
   </Attribute>
   <Attribute Name="semanticId" AttributeDataType="xs:string">
    <Description>Description or comments on the element. The description can be provided in several languages. This attribute has the name of the label and has a value with the label written in the default language. The individual languages are modelled as child attributes. The names of the child attributes are the prefix “aml-lang=” with the expression of the language in compliance with RFC5646. At it, the values of the child attributes are the labels within the respective language.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
   </Attribute>
   <Attribute Name="qualifier:[TYPE]=[VALUE]" AttributeDataType="xs:string">
    <Description>A qualifier is a type-value-pair that makes additional statements w.r.t. the value of the element. [TYPE] is the value of the attribute type and [VALUE] is the value of the attribute value.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Qualifiable/qualifier" />
    <Attribute Name="type" AttributeDataType="xs:string">
     <Description>The type describes the type of the qualifier that is applied to the element.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Qualifier/type" />
    </Attribute>
    <Attribute Name="value" AttributeDataType="xs:string">
     <Description>The qualifier value is the value of the qualifier. Constraint AASd-006: if both, the value and the valueId are present then the value needs to be identical to the value of the referenced coded value in valueId.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Qualifier/value" />
    </Attribute>
    <Attribute Name="valueId" AttributeDataType="xs:string">
     <Description>Reference to the global unqiue id of a coded value.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Qualifier/valueId" />
    </Attribute>
   </Attribute>
   <ExternalInterface Name="first" ID="6382c562-f165-488c-94c6-52fe0b27bb6a" RefBaseClassPath="AssetAdministrationShellInterfaceClassLib/ReferableReference">
    <Attribute Name="value" AttributeDataType="xs:string">
     <Description>Reference to any other referable element of any other AAS or a reference to an external object or entity. Note: For references to any other referable elment of the same AAS InternalLinks are used and this attribute value shall be empty.</Description>
    </Attribute>
   </ExternalInterface>
   <ExternalInterface Name="second" ID="f05eccaf-f1f7-4c0c-8502-ba5372640814" RefBaseClassPath="AssetAdministrationShellInterfaceClassLib/ReferableReference">
```

```
    <Attribute Name="value" AttributeDataType="xs:string">
     <Description>Reference to any other referable element of any other AAS or a reference to an external object or entity. Note: For references to any other referable elment of the same AAS InternalLinks are used and this attribute value shall be empty.</Description>
    </Attribute>
   </ExternalInterface>
  </RoleClass>
  <RoleClass Name="AnnotatedRelationshipElement" RefBaseClassPath="AssetAdministrationShellRoleClassLib/RelationshipElement">
   <Description>An annotated relationship element is an relationship element that can be annotated with additional data elements.</Description>
   <Attribute Name="annotation" AttributeDataType="xs:string">
    <Description>Annotations that hold for the relationships between the two elements.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:AnnotatedRelationshipElement/annotation" />
   </Attribute>
  </RoleClass>
  <RoleClass Name="Operation" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
   <Description>An operation is a submodel element with input and output variables.</Description>
   <Attribute Name="idShort" AttributeDataType="xs:string">
    <Description>Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the “BrowserPath” in OPC UA.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
   </Attribute>
   <Attribute Name="category" AttributeDataType="xs:string">
    <Description>The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
   </Attribute>
   <Attribute Name="description" AttributeDataType="xs:string">
    <Description>Description or comments on the element. The description can be provided in several languages.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
    <Attribute Name="aml-lang=EN" />
    <Attribute Name="aml-lang=DE" />
   </Attribute>
   <Attribute Name="dataSpecification" AttributeDataType="xs:string">
    <Description>Global reference to the data specification template used by the element.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasDataSpecification/dataSpecification" />
   </Attribute>
   <Attribute Name="kind" AttributeDataType="xs:string">
    <Description>Kind of the element: either template or instance.</Description>
    <DefaultValue>Instance</DefaultValue>
```

```
    <Value>Instance</Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
    <Constraint Name="">
     <NominalScaledType>
      <RequiredValue>Instance</RequiredValue>
      <RequiredValue>Type</RequiredValue>
     </NominalScaledType>
    </Constraint>
   </Attribute>
   <Attribute Name="semanticId" AttributeDataType="xs:string">
    <Description>Description or comments on the element. The description can be provided in several languages. This attribute has the name of the label and has a value with the label written in the default language. The individual languages are modelled as child attributes. The names of the child attributes are the prefix “aml-lang=” with the expression of the language in compliance with RFC5646. At it, the values of the child attributes are the labels within the respective language.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
   </Attribute>
   <Attribute Name="qualifier:[TYPE]=[VALUE]" AttributeDataType="xs:string">
    <Description>A qualifier is a type-value-pair that makes additional statements w.r.t. the value of the element. [TYPE] is the value of the attribute type and [VALUE] is the value of the attribute value.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Qualifiable/qualifier" />
    <Attribute Name="type" AttributeDataType="xs:string">
     <Description>The type describes the type of the qualifier that is applied to the element.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Qualifier/type" />
    </Attribute>
    <Attribute Name="value" AttributeDataType="xs:string">
     <Description>The qualifier value is the value of the qualifier. Constraint AASd-006: if both, the value and the valueId are present then the value needs to be identical to the value of the referenced coded value in valueId.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Qualifier/value" />
    </Attribute>
    <Attribute Name="valueId" AttributeDataType="xs:string">
     <Description>Reference to the global unqiue id of a coded value.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Qualifier/valueId" />
    </Attribute>
   </Attribute>
  </RoleClass>
  <RoleClass Name="OperationInputVariables" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
   <Description>The list of AAS OperationVariableIn entities. In AML, the corresponding InternalElement with this role is a child of the InternalElement with the Operation role.</Description>
  </RoleClass>
  <RoleClass Name="OperationOutputVariables" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
```

```
    <Description>The list of AAS OperationVariableOut entities. In AML, the corresponding InternalElement with this role is a child of the InternalElement with the Operation role.</Description>
  </RoleClass>
  <RoleClass Name="OperationInoutputVariables" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
    <Description>The list of AAS OperationVariableOut entities. In AML, the corresponding InternalElement with this role is a child of the InternalElement with the Operation role.</Description>
  </RoleClass>
  <RoleClass Name="View" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Group">
    <Description>A view is a collection of referable elements w.r.t. to a specific viewpoint of one or more stakeholders.</Description>
    <Attribute Name="idShort" AttributeDataType="xs:string">
      <Description>Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the “BrowserPath” in OPC UA.</Description>
      <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
    </Attribute>
    <Attribute Name="category" AttributeDataType="xs:string">
      <Description>The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints.</Description>
      <Value></Value>
      <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
    </Attribute>
    <Attribute Name="description" AttributeDataType="xs:string">
      <Description>Description or comments on the element. The description can be provided in several languages.</Description>
      <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
      <Attribute Name="aml-lang=EN" />
      <Attribute Name="aml-lang=DE" />
    </Attribute>
    <Attribute Name="dataSpecification" AttributeDataType="xs:string">
      <Description>Global reference to the data specification template used by the element.</Description>
      <Value></Value>
      <RefSemantic CorrespondingAttributePath="AAS:HasDataSpecification/dataSpecification" />
    </Attribute>
    <Attribute Name="semanticId" AttributeDataType="xs:string">
      <Description>Description or comments on the element. The description can be provided in several languages. This attribute has the name of the label and has a value with the label written in the default language. The individual languages are modelled as child attributes. The names of the child attributes are the prefix “aml-lang=” with the expression of the language in compliance with RFC5646. At it, the values of the child attributes are the labels within the respective language.</Description>
      <Value></Value>
      <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
    </Attribute>
  </RoleClass>
  <RoleClass Name="ConceptDictionary" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
    <Description>A dictionary contains elements that can be reused.</Description>
    <Attribute Name="idShort" AttributeDataType="xs:string">
```

```
    <Description>Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the “BrowserPath” in OPC UA.</Description>

    <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />

   </Attribute>

   <Attribute Name="category" AttributeDataType="xs:string">

    <Description>The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints.</Description>

    <Value></Value>

    <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />

   </Attribute>

   <Attribute Name="description" AttributeDataType="xs:string">

    <Description>Description or comments on the element. The description can be provided in several languages.</Description>

    <Value></Value>

    <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />

    <Attribute Name="aml-lang=EN" AttributeDataType="xs:string">

     <Value></Value>

    </Attribute>

    <Attribute Name="aml-lang=DE" AttributeDataType="xs:string">

     <Value></Value>

    </Attribute>

   </Attribute>

  </RoleClass>

  <RoleClass Name="ConceptDescription" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">

   <Description>Explanation: The semantics of a property or other elements that may have a semantic description is defined by a concept description. The description of the concept should follow a standardized schema (realized as data specification template).</Description>

   <Attribute Name="idShort" AttributeDataType="xs:string">

    <Description>Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the “BrowserPath” in OPC UA.</Description>

    <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />

   </Attribute>

   <Attribute Name="category" AttributeDataType="xs:string">

    <Description>The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints.</Description>

    <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />

   </Attribute>

   <Attribute Name="description" AttributeDataType="xs:string">

    <Description>Description or comments on the element. The description can be provided in several languages.</Description>

    <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />

    <Attribute Name="aml-lang=EN" AttributeDataType="xs:string" />
```

```
    <Attribute Name="aml-lang=DE" AttributeDataType="xs:string" />
   </Attribute>
   <Attribute Name="identification">
    <Description>Abstract attribute class for identification. Has the subattributes id and idType.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
    <Attribute Name="id" AttributeDataType="xs:string">
     <Description>Identifier of the element. Its type is defined in idType. Id is a subproperty of identification.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
    </Attribute>
    <Attribute Name="idType" AttributeDataType="xs:string">
     <Description>Type of the Identifier, e.g. IRI, IRDI etc. The supported Identifier types are defined in the enumeration "IdentifierType". IdType is a subproperty of identification.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
    </Attribute>
   </Attribute>
   <Attribute Name="administration">
    <Description>Abstract attribute for administration. Has the subattributes revision and version.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:Identifiable/administration" />
    <Attribute Name="revision" AttributeDataType="xs:string">
     <Description>Revision of the element. Constraint AASd-005: A revision requires a version. This means, if there is no version there is no revision neither. Revision is a subproperty of administration.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:AdministrativeInformation/revision" />
    </Attribute>
    <Attribute Name="version" AttributeDataType="xs:string">
     <Description>Version of the element. Version is a subproperty of administration.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:AdministrativeInformation/version" />
    </Attribute>
   </Attribute>
   <Attribute Name="dataSpecification" AttributeDataType="xs:string">
    <Description>Global reference to the data specification template used by the element.</Description>
    <RefSemantic CorrespondingAttributePath="AAS:HasDataSpecification/dataSpecification" />
   </Attribute>
   <Attribute Name="isCaseOf" AttributeDataType="xs:string">
    <Description>Global reference to an external definition the concept is compatible to or was derived from.</Description>
    <Value></Value>
    <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/isCaseOf" />
   </Attribute>
  </RoleClass>
  <RoleClass Name="DataSpecification" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
```

```
<Description>Description Role class of an element that has a data specification template. A template defines the additional attributes an element may or shall have.</Description>
<Attribute Name="idShort" AttributeDataType="xs:string">
<Description>Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the “BrowserPath” in OPC UA.</Description>
<RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
</Attribute>
<Attribute Name="category" AttributeDataType="xs:string">
<Description>The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints.</Description>
<Value></Value>
<RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
</Attribute>
<Attribute Name="description" AttributeDataType="xs:string">
<Description>Description or comments on the element. The description can be provided in several languages.</Description>
<Value></Value>
<RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
<Attribute Name="aml-lang=EN" AttributeDataType="xs:string">
<Value></Value>
</Attribute>
<Attribute Name="aml-lang=DE" AttributeDataType="xs:string">
<Value></Value>
</Attribute>
</Attribute>
<Attribute Name="identification">
<Description>Abstract attribute class for identification. Has the subattributes id and idType.</Description>
<RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
<Attribute Name="id" AttributeDataType="xs:string">
<Description>Identifier of the element. Its type is defined in idType. Id is a subproperty of identification.</Description>
<Value></Value>
<RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
</Attribute>
<Attribute Name="idType" AttributeDataType="xs:string">
<Description>Type of the Identifier, e.g. IRI, IRDI etc. The supported Identifier types are defined in the enumeration “IdentifierType”. IdType is a subproperty of identification.</Description>
<Value></Value>
<RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
</Attribute>
</Attribute>
<Attribute Name="administration">
<Description>Abstract attribute for administration. Has the subattributes revision and version.</Description>
<RefSemantic CorrespondingAttributePath="AAS:Identifiable/administration" />
```

```
    <Attribute Name="revision" AttributeDataType="xs:string">
     <Description>Revision of the element. Constraint AASd-005: A revision requires a version. This means, if there is no version there is no revision neither. Revision is a subproperty of administration.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:AdministrativeInformation/revision" />
    </Attribute>
    <Attribute Name="version" AttributeDataType="xs:string">
     <Description>Version of the element. Version is a subproperty of administration.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:AdministrativeInformation/version" />
    </Attribute>
   </Attribute>
  </RoleClass>
  <RoleClass Name="DataSpecificationContent" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
   <Description>Content of the data specification template.</Description>
  </RoleClass>
 </RoleClassLib>
 <SystemUnitClassLib Name="AssetAdministrationShellDataSpecificationTemplates">
  <Version>0</Version>
  <SystemUnitClass Name="DataSpecificationIEC61360Template" ID="572c0568-4019-40ec-bfc4-a3a82dc6eed4">
   <Description>An AAS Data Specification template for IEC61369. A template consists of the DataSpecificationContent containing the additional attributes to be added to the element instance that references the data specification template and meta information about the template itself (this is why DataSpecification inherits from Identifiable). In UML these are two separated classes. </Description>
    <Attribute Name="idShort" AttributeDataType="xs:string">
     <Description>Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the “BrowserPath” in OPC UA.</Description>
     <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
    </Attribute>
    <Attribute Name="category" AttributeDataType="xs:string">
     <Description>The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
    </Attribute>
    <Attribute Name="description" AttributeDataType="xs:string">
     <Description>Description or comments on the element. The description can be provided in several languages.</Description>
     <Value></Value>
     <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
     <Attribute Name="aml-lang=EN" AttributeDataType="xs:string">
      <Value></Value>
     </Attribute>
     <Attribute Name="aml-lang=DE" AttributeDataType="xs:string">
```

```
      <Value></Value>
     </Attribute>
    </Attribute>
    <Attribute Name="identification">
     <Description>Abstract attribute class for identification. Has the subattributes id and idType.</Description>
     <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
     <Attribute Name="id" AttributeDataType="xs:string">
      <Description>Identifier of the element. Its type is defined in idType. Id is a subproperty of identification.</Description>
      <Value>http://admin-shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>
      <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
     </Attribute>
     <Attribute Name="idType" AttributeDataType="xs:string">
      <Description>Type of the Identifier, e.g. IRI, IRDI etc. The supported Identifier types are defined in the enumeration "IdentifierType". IdType is a subproperty of identification.</Description>
      <Value>IRI</Value>
      <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
     </Attribute>
    </Attribute>
    <Attribute Name="administration">
     <Description>Abstract attribute for administration. Has the subattributes revision and version.</Description>
     <RefSemantic CorrespondingAttributePath="AAS:Identifiable/administration" />
     <Attribute Name="revision" AttributeDataType="xs:string">
      <Description>Revision of the element. Constraint AASd-005: A revision requires a version. This means, if there is no version there is no revision neither. Revision is a subproperty of administration.</Description>
      <Value></Value>
      <RefSemantic CorrespondingAttributePath="AAS:AdministrativeInformation/revision" />
     </Attribute>
     <Attribute Name="version" AttributeDataType="xs:string">
      <Description>Version of the element. Version is a subproperty of administration.</Description>
      <Value></Value>
      <RefSemantic CorrespondingAttributePath="AAS:AdministrativeInformation/version" />
     </Attribute>
    </Attribute>
    <SupportedRoleClass RefRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecification" />
    <SystemUnitClass Name="DataSpecificationIEC61360" ID="8c019d8e-7ddd-4283-aa63-1747021c3d1b">
     <Description>The content of an AAS Data Specification template for IEC61360.</Description>
     <Attribute Name="preferredName" AttributeDataType="xs:string">
      <Description>Identifies the attribute hierarchy for preferredName in above attribute hierachy. Subordinate attributes are designated by the country code information (see aml-lang literal).</Description>
      <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/preferredName" />
     </Attribute>
     <Attribute Name="shortName" AttributeDataType="xs:string">
      <Description>Identifies the attribute for shortName in above attribute hierarchy.</Description>
      <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
```

```
    </Attribute>
    <Attribute Name="unit" AttributeDataType="xs:string">
     <Description>Identifies the attribute for unit in above attribute hierarchy.</Description>
     <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
    </Attribute>
    <Attribute Name="unitId" AttributeDataType="xs:string">
     <Description>Identifies the attribute for unitId in above attribute hierarchy in its string serialization.</Description>
     <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unitId" />
    </Attribute>
    <Attribute Name="sourceOfDefinition" AttributeDataType="xs:string">
     <Description>Identifies the attribute for sourceOfDefinition in above attribute hierachy. Subordinate attributes are designated by the
country code information (see aml-lang literal).</Description>
     <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/sourceOfDefinition" />
    </Attribute>
    <Attribute Name="symbol" AttributeDataType="xs:string">
     <Description>Identifies the attribute for symbol in above attribute hierarchy.</Description>
     <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/symbol" />
    </Attribute>
    <Attribute Name="dataType" AttributeDataType="xs:string">
     <Description>Identifies the attribute for dataType in above attribute hierarchy.</Description>
     <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
    </Attribute>
    <Attribute Name="definition" AttributeDataType="xs:string">
     <Description>Identifies the attribute for definition in above attribute hierachy. Subordinate attributes are designated by the country code
information (see aml-lang literal).</Description>
     <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
    </Attribute>
    <Attribute Name="valueFormat" AttributeDataType="xs:string">
     <Description>Identifies the attribute for valueFormat in above attribute hierarchy.</Description>
     <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/valueFormat" />
    </Attribute>
    <Attribute Name="valueList" AttributeDataType="xs:string">
     <Description>Identifies the attribute for valueList in above attribute hierarchy.</Description>
     <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/valueList" />
    </Attribute>
    <Attribute Name="value" AttributeDataType="xs:string">
     <Description>The attribute value.</Description>
     <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/value" />
    </Attribute>
    <Attribute Name="valueId" AttributeDataType="xs:string">
     <Description>The id for the value.</Description>
     <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/valueId" />
    </Attribute>
```

```
    <SupportedRoleClass RefRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
   </SystemUnitClass>
  </SystemUnitClass>
 </SystemUnitClassLib>
</CAEXFile>
```

iii. RefSemantic Values and Roles for AAS

The following symbolic identifications (literals) are used by the AutomationML mapping. Without loss of generality, these literals are recommended to be used also by other functionalities, unless not otherwise stated by this specification.

Literals to be used in names in namespace “AAS:”	Description
aml-lang=	When serializing AAS *langString* types to AML, this literal with respective country code (e.g. "*aml-lang*=EN") is used to identify the language specific value of the *langString*.
qualifier:	For the serialization of qualifiers, this literal is used to give name to the root of the hierarchy of attributes for the qualifier (e.g. "q*ualifier:PredicateRelation=GREATER_THAN_0*”)

Literals to be used in names in namespace “AAS:”	Description
AssetAdministrationShellRoleClassLib	Name of the root node element for the AAS role classes in the AML *RoleClassLib*.
AssetAdministrationShellInterfaceClassLib	Name of the root node element for the AAS information in the AML InterfaceClassLib.
AssetAdministrationShellDataSpecifications	Name of the root node element for the predefined AAS data specification templates in the AML *SystemUnitClassLib*.

The following symbolic identifications (literals) are used by the AutomationML mapping in case no other names or existing libraries are selected for export.

Literals to be used in names in namespace “AAS:”	Description
AssetAdministrationShell-InstanceHierarchy	Default name of the root node element for the AAS information in the AML InstanceHierarchy.
AssetAdministrationShell-SystemUnitClasses	Default name of the root node element for the AAS information in the AML SystemUnitClassLib.
AssetAdministrationShell-ConceptDescriptions	Default name of the root node element for the AAS concept descriptions in the AML InstanceHierarchy.

The following literals are used to identify attributes of AAS metamodel elements. In AutomationML, this is done via the "RefSemantic" mechanism and the literal is preceeded by the string "AAS:" (e.g. "AAS:Asset"). The rules for creating these values are described in Clause 5.2.2.

Literals to identify attributes[45] in namespace “AAS:” (value of *RefSemantic*)	Description
AAS:Referable/ idShort	Identifies the *idShort* attribute within an AAS Referable, such as Property.
AAS:Referable/ category	As above for *category*
AAS:Referable/ description	As above for *description*
AAS:HasKind/ kind	Identifies the *kind* attribute of submodel and submodel element entities.
AAS:HasSemantics/ semanticId	Identifies the *semanticId* attribute within various AAS entities.
AAS:Identifiable/ identification	Identitifies the attribute hierachy for *identification* information within AAS Identifiables.
AAS:Identifier/ idType	Identifies the attribute for *idType* in above attribute hierachy.
AAS:Identifier/ id	Identifies the attribute for *id* in above attribute hierachy.
AAS:Identifiable/ administration	Identitifies the attribute hierachy for *administrative information* within AAS Identifiables.
AAS:AdministrativeInformation/ version	Identifies the attribute for *version* in above attribute hierachy.
AAS:AdministrativeInformation/ revision	Identifies the attribute for *revision* in above attribute hierachy.
AAS:HasDataSpecification/ dataSpecification	Identifies an attribute containing the *dataSpecification* reference in its string serialization.
AAS:Qualifiable/ qualifier	Identifies the attribute hierachy for an AAS *Qualifiable* in total.
AAS:Qualifier/ type	Identifies the attribute for *qualifier type* in above attribute hierachy.
AAS:Qualifier/ value	Identifies the attribute for *qualifier value* in above attribute hierachy.
AAS:Qualifier/ valueId	Identifies the attribute for qualifier *valueId* in above attribute hierachy.
AAS:AssetAdministrationShell/ derivedFrom	Identifies the attribute containing the *derivedFrom* reference in its string serialization
AAS:Asset/ kind	Identifies the attribute containg the information about the kind of the asset
AAS:Asset/ assetIdentificationModel	Identifies the attribute containing the *assetIdentificationModel* reference for AAS Asset in its string serialization

[45] Note: the blank within the path is to be ignored, it is just used for better formatting of the table. I.e. *AAS:Referable/ idShort* needs to be *AAS:Referable/idShort* as value of the RefSemantics attribute.

AAS:Property/ value AAS:MultiLanguageProperty/ value AAS:Blob/ value AAS:File/ value AAS:ReferenceElement/ value	Identifies the *value* attribute in various AAS SubmodelElements.
AAS:Property/ valueId	Identifies the *valueId* attribute in the property element.
AAS:Blob/ mimeType	Identifies the *mimeType* attribute in various AAS SubmodelElements. For SubmodelElement File the *MIMEType* attribute of the predefined *FileDataReference* interface is used. That has the same semantics as AAS:File/ mimeType
AAS:AnnotationRelationshipElement/ annotations	Annotation of the AAS SubmodelElement Annotated Relationship.
AAS:ConceptDescription/ isCaseOf	Identifies the attribute containing the *isCaseOf* reference within AAS ConceptDescription in its string serialization.
AAS:ConceptDescription/ dataSpecification	Identifies the attribute containing the *dataSpecification* reference within AAS EmbeddedDataSpecification in its string serialization.

The following literals are used to identify the predefined templates as defined in Clause 4.8. So far only the IEC61360 Template for concept descriptions is supported.

Literals to identify system unit classes in AssetAdministrationShellDataSpecificationTemplates	Description
DataSpecificationIEC61360Template	System Unit class for the predefined data specification template representing IEC61360 attributes for properties etc. It has the role *DataSpecification*.
DataSpecificationIEC61360	System Unit class for the content of the predefined data specification template DataSpecificationIEC61360Template. It has the role *DataSpecificationContent*.

The AutomationML mapping supports the template *"DataSpecificationIEC616360"*. The attribute literals are defined in the following table. The name space qualifier in this case is "IEC". Example: *"IEC:DataSpecificationIEC61360/preferredName"*

Literals to identify attributes n namespace "IEC:" (value of RefSemantic)	Description

IEC:DataSpecificationIEC61360/ preferredName	Identifies the attribute hierarchy for *preferredName* in above attribute hierachy. Subordinate attributes are designated by the country code information (see aml-lang literal).
IEC:DataSpecificationIEC61360/ shortName	Identifies the attribute for *shortName* in above attribute hierachy.
IEC:DataSpecificationIEC61360/ unit	Identifies the attribute for *unit* in above attribute hierachy.
IEC:DataSpecificationIEC61360/ unitId	Identifies the attribute for *unitId* in above attribute hierachy in its string serialization.
IEC:DataSpecificationIEC61360/ sourceOfDefinition	Identifies the attribute for *sourceOfDefinition* in above attribute hierachy. Subordinate attributes are designated by the country code information (see aml-lang literal).
IEC:DataSpecificationIEC61360/ symbol	Identifies the attribute for *symbol* in above attribute hierachy.
IEC:DataSpecificationIEC61360/ valueFormat	Identifies the attribute for *valueFormat* in above attribute hierachy.
IEC:DataSpecificationIEC61360/ dataType	Identifies the attribute for *dataType* in above attribute hierachy.
IEC:DataSpecificationIEC61360/ definition	Identifies the attribute for *definition* in above attribute hierachy. Subordinate attributes are designated by the country code information (see aml-lang literal).
IEC:DataSpecificationIEC61360/ valueFormat	Identifies the attribute for *valueFormat* in above attribute hierachy.

The following literals are used to identify AAS entities. In AutomationML, this is done via the "Role" mechanism and the role name is preceeded by " AssetAdministrationShellRoleClassLib/" in order to identify the accoding role class lib (e.g. "AssetAdministrationShellRoleClassLib/ Asset").

Literals to identify AAS entities	Inherits from	Description
Asset	AutomationMLBaseRole	The AAS Asset entity. In AML, the corresponding InternalElement with this role is a child of the InternalElement with the *AssetAdministrationShell* role.
View	Group	The AAS View entity. In AML, the corresponding InternalElement with this role is a child of the InternalElement with the *AssetAdministrationShell* role.
AssetAdministrationShell	AutomationMLBaseRole	The AssetAdministrationShell entity. In AML, the corresponding InternalElement is child of the instance hierarchy named *AssetAdministrationShellInstanceHierarchy*.
Submodel	AutomationMLBaseRole	The AAS Submodel entity. In AML, the corresponding InternalElement with this role is a child of the InternalElement with the *AssetAdministrationShell* role.

SubmodelElementCollection, Operation	AutomationMLBaseRole	One of the AAS SubmodelElement entities. In AML, the corresponding element is always an InternalElement with this role and is a child of the InternalElement with the *Submodel* role.
Property, Blob, Capability	AutomationMLBaseRole	One of the AAS SubmodelElement entities. In AML, the corresponding element is always an InternalElement with this role and is a child of the InternalElement with the *Submodel* role.
ReferenceElement	AutomationMLBaseRole	The value of an ReferenceElement is realized as an interface *ReferableReference*.
RelationshipElement, AnnotatedRelationshipElement	AutomationMLBaseRole	The attributes *first* and *second* of a relationship element are realized with interface *ReferableReference*.
File	ExternalData	The AAS File SubmodelElement is realized as ExternalData with interface *FileDataReference*.
OperationInputVariables, OperationOutputVariables, OperationInoutputVariables	AutomationMLBaseRole	The list of AAS OperationVariable entities. In AML, the corresponding InternalElement with this role is a child of the InternalElement with the *Operation* role.
ConceptDescription	AutomationMLBaseRole	The AAS ConceptDescription entity. In AML, the corresponding InternalElement is child of the instance hierarchy library named *Asset-AdministrationShellConceptDescriptions*.

iv. AutomationML Example

The following AutomationML Example is a realization of the example as explained in Clause 5.3. The used role, interfaces and system unit classes are not included, They can be found in previous Clauses.

```
<?xml version="1.0" encoding="utf-8"?>
<CAEXFile FileName="UnifiedAASExampleVWSiD2.0 Boss.aml" SchemaVersion="2.15"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="CAEX_ClassModel_V2.15.xsd">
  <AdditionalInformation AutomationMLVersion="2.0" />
  <InstanceHierarchy Name="AssetAdministrationShellInstanceHierarchy">
    <Version>0</Version>
    <InternalElement Name="ExampleMotor" ID="6b5bb2c7-ebea-4f5b-a0ea-e70536f44b24">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">
          <Value>http://customer.com/aas/9175_7013_7091_9168</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
```

```
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
        <Value>ExampleMotor</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="category">
        <Value>CONSTANT</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute>
      <InternalElement Name="ServoDCMotor" ID="d2c490c9-20a0-4022-9e81-7f2cd1c0ebf3">
        <Attribute Name="identification">
          <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
          <Attribute Name="idType">
            <Value>IRI</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
          </Attribute>
          <Attribute Name="id">
            <Value>http://customer.com/assets/KHBVZJSQKIY</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
          </Attribute>
        </Attribute>
        <Attribute Name="idShort">
          <Value>ServoDCMotor</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
        </Attribute>
        <Attribute Name="category">
          <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
        </Attribute>
        <Attribute Name="kind">
          <Value>Instance</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Asset/kind" />
        </Attribute>
        <Attribute Name="assetIdentificationModelRef">
          <Value>(Submodel)(local)[IRI]i40.customer.com/type/1/1/F13E8576F6488342</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Asset/assetIdentificationModel" />
        </Attribute>
        <RoleRequirements RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Asset" />
      </InternalElement>
      <InternalElement Name="Identification" ID="45595b09-5c84-4a61-94fd-e63b01d48153">
        <Attribute Name="identification">
          <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
          <Attribute Name="idType">
            <Value>IRI</Value>
```

```
            <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
          </Attribute>
          <Attribute Name="id">
            <Value>http://i40.customer.com/type/1/1/F13E8576F6488342</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
          </Attribute>
        </Attribute>
        <Attribute Name="idShort">
          <Value>Identification</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
        </Attribute>
        <Attribute Name="category">
          <Value>CONSTANT</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
        </Attribute>
        <Attribute Name="description">
          <Value>Identification from Manufacturer</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
          <Attribute Name="aml-lang=EN">
            <Value>Identification from Manufacturer</Value>
          </Attribute>
          <Attribute Name="aml-lang=DE">
            <Value>Hersteller-Identifikation</Value>
          </Attribute>
        </Attribute>
        <Attribute Name="kind">
          <Value>Instance</Value>
          <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
        </Attribute>
        <Attribute Name="semanticId">
          <Value>(GlobalReference)(no-local)[IRDI]0173-1#01-ADN198#009</Value>
          <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
        </Attribute>
        <InternalElement Name="Manufacturer" ID="d6964099-e1fa-4157-b9b6-68068d4024c4">
          <Attribute Name="idShort">
            <Value>Manufacturer</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
          </Attribute>
          <Attribute Name="category">
            <Value>CONSTANT</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
          </Attribute>
          <Attribute Name="kind">
            <Value>Instance</Value>
```

```
            <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
          </Attribute>
          <Attribute Name="semanticId">
            <Value>(GlobalReference)(no-local)[IRDI]0173-1#02-AAO677#002</Value>
            <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
          </Attribute>
          <Attribute Name="value" AttributeDataType="xs:string">
            <Value>CUSTOMER GmbH</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
          </Attribute>
          <RoleRequirements RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property" />
        </InternalElement>
        <InternalElement Name="GLN" ID="66e4c721-d4fa-4e94-827b-dc4fd631ead9">
          <Attribute Name="idShort">
            <Value>GLN</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
          </Attribute>
          <Attribute Name="category">
            <Value>CONSTANT</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
          </Attribute>
          <Attribute Name="kind">
            <Value>Instance</Value>
            <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
          </Attribute>
          <Attribute Name="semanticId">
            <Value>(GlobalReference)(no-local)[IRDI]0173-1#02-AAY812#001</Value>
            <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
          </Attribute>
          <Attribute Name="value" AttributeDataType="xs:integer">
            <Value>10101010</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
          </Attribute>
          <RoleRequirements RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property" />
        </InternalElement>
        <InternalElement Name="ProductDesignation" ID="e32a4692-dbe6-4b13-858b-0a4e7071fd9d">
          <Attribute Name="idShort">
            <Value>ProductDesignation</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
          </Attribute>
          <Attribute Name="category">
            <Value>CONSTANT</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
          </Attribute>
```

```
          <Attribute Name="kind">
            <Value>Instance</Value>
            <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
          </Attribute>
          <Attribute Name="semanticId">
            <Value>(GlobalReference)(no-local)[IRDI]0173-1#02-AAW338#001</Value>
            <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
          </Attribute>
          <Attribute Name="value" AttributeDataType="xs:langString">
            <Value>I40 Capable Servo Motor (EN)</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
          </Attribute>
          <RoleRequirements RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property" />
        </InternalElement>
        <InternalElement Name="SerialNumber" ID="dd9ba565-175f-41df-b98a-903e80f7212b">
          <Attribute Name="idShort">
            <Value>SerialNumber</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
          </Attribute>
          <Attribute Name="category">
            <Value>CONSTANT</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
          </Attribute>
          <Attribute Name="kind">
            <Value>Instance</Value>
            <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
          </Attribute>
          <Attribute Name="semanticId">
            <Value>(GlobalReference)(no-local)[IRDI]0173-1#02-AAM556#002</Value>
            <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
          </Attribute>
          <Attribute Name="value" AttributeDataType="xs:string">
            <Value>P12345678I40</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
          </Attribute>
          <RoleRequirements RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property" />
        </InternalElement>
        <RoleRequirements RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Submodel" />
      </InternalElement>
      <InternalElement Name="TechnicalData" ID="7c75c54e-f414-4282-863c-798a8858349b">
        <Attribute Name="identification">
          <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
          <Attribute Name="idType">
            <Value>IRI</Value>
```

```
            <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
          </Attribute>
          <Attribute Name="id">
            <Value>http.//i40.customer.com/type/1/1/7A7104BDAB57E184</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
          </Attribute>
        </Attribute>
        <Attribute Name="idShort">
          <Value>TechnicalData</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
        </Attribute>
        <Attribute Name="category">
          <Value>CONSTANT</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
        </Attribute>
        <Attribute Name="kind">
          <Value>Instance</Value>
          <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
        </Attribute>
        <Attribute Name="semanticId">
          <Value>(GlobalReference)(no-local)[IRDI]0173-1#01-AFZ615#016</Value>
          <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
        </Attribute>
        <InternalElement Name="MaxRotationSpeed" ID="59aad534-0d4b-469d-83c3-4fbdb09b55f9">
          <Attribute Name="idShort">
            <Value>MaxRotationSpeed</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
          </Attribute>
          <Attribute Name="category">
            <Value>PARAMETER</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
          </Attribute>
          <Attribute Name="kind">
            <Value>Instance</Value>
            <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
          </Attribute>
          <Attribute Name="semanticId">
            <Value>(ConceptDescription)(local)[IRDI]0173-1#02-BAA120#008</Value>
            <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
          </Attribute>
          <Attribute Name="value" AttributeDataType="xs:integer">
            <Value>5000</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
          </Attribute>
```

```
        <RoleRequirements RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property" />
      </InternalElement>
      <InternalElement Name="MaxTorque" ID="cdb9f75b-da92-484d-a6dd-c024c46d58c8">
        <Attribute Name="idShort">
          <Value>MaxTorque</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
        </Attribute>
        <Attribute Name="category">
          <Value>PARAMETER</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
        </Attribute>
        <Attribute Name="kind">
          <Value>Instance</Value>
          <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
        </Attribute>
        <Attribute Name="semanticId">
          <Value>(ConceptDescription)(local)[IRDI]0173-1#02-BAE098#004</Value>
          <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
        </Attribute>
        <Attribute Name="value" AttributeDataType="xs:float">
          <Value>200</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
        </Attribute>
        <RoleRequirements RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property" />
      </InternalElement>
      <InternalElement Name="CoolingType" ID="92153ec4-89db-4f31-ae67-57dd1ce9633c">
        <Attribute Name="idShort">
          <Value>CoolingType</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
        </Attribute>
        <Attribute Name="category">
          <Value>PARAMETER</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
        </Attribute>
        <Attribute Name="description">
          <Value>open circuit, external cooling</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
          <Attribute Name="aml-lang=EN">
            <Value>open circuit, external cooling</Value>
          </Attribute>
        </Attribute>
        <Attribute Name="kind">
          <Value>Instance</Value>
          <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
```

```
        </Attribute>
        <Attribute Name="semanticId">
          <Value>(ConceptDescription)(local)[IRDI]0173-1#02-BAE122#006</Value>
          <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
        </Attribute>
        <Attribute Name="value" AttributeDataType="xs:string">
          <Value>BAB657</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
        </Attribute>
        <Attribute Name="valueId">
          <Value>(ConceptDescription)(local)[IRDI]0173-1#07-BAB657#003 </Value>
          <RefSemantic CorrespondingAttributePath="AAS:Property/valueId" />
        </Attribute>
        <RoleRequirements RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property" />
      </InternalElement>
      <RoleRequirements RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Submodel" />
    </InternalElement>
    <InternalElement Name="OperationalData" ID="e2bcafde-ccac-4c73-8df3-9afef6b6b328">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">
          <Value>http://i40.customer.com/instance/1/1/AC69B1CB44F07935</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
        <Value>OperationalData</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="category">
        <Value>VARIABLE</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute>
      <Attribute Name="kind">
        <Value>Instance</Value>
        <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
      </Attribute>
      <Attribute Name="semanticId">
        <Value>(GlobalReference)(no-local)[IRDI]0173-1#01-AFZ615#016</Value>
        <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
```

```
        </Attribute>
        <InternalElement Name="RotationSpeed" ID="5baaac73-517c-46ab-949d-02111e1322f8">
          <Attribute Name="idShort">
            <Value>RotationSpeed</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
          </Attribute>
          <Attribute Name="category">
            <Value>VARIABLE</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
          </Attribute>
          <Attribute Name="kind">
            <Value>Instance</Value>
            <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
          </Attribute>
          <Attribute Name="semanticId">

<Value>(ConceptDescription)(local)[IRI]http://customer.com/cd/1/1/18EBD56F6B43D895</Value>
            <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
          </Attribute>
          <Attribute Name="value" AttributeDataType="xs:integer">
            <Value>4370</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
          </Attribute>
          <RoleRequirements RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property" />
        </InternalElement>
        <InternalElement Name="Torque" ID="3d5d996e-e90b-4e7c-8ade-906f91daad87">
          <Attribute Name="idShort">
            <Value>Torque</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
          </Attribute>
          <Attribute Name="category">
            <Value>VARIABLE</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
          </Attribute>
          <Attribute Name="kind">
            <Value>Instance</Value>
            <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
          </Attribute>
          <Attribute Name="semanticId">

<Value>(ConceptDescription)(local)[IRI]http://customer.com/cd/1/1/18EBD56F6B43D896</Value>
            <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
          </Attribute>
          <Attribute Name="value" AttributeDataType="xs:float">
            <Value>117.4</Value>
```

```
            <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
          </Attribute>
          <RoleRequirements RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property" />
        </InternalElement>
        <RoleRequirements RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Submodel" />
      </InternalElement>
      <InternalElement Name="Documentation" ID="68fff696-8788-4704-aa0a-1170458fff61">
        <Attribute Name="identification">
          <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
          <Attribute Name="idType">
            <Value>IRI</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
          </Attribute>
          <Attribute Name="id">
            <Value>http://i40.customer.com/type/1/1/1A7B62B529F19152</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
          </Attribute>
        </Attribute>
        <Attribute Name="idShort">
          <Value>Documentation</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
        </Attribute>
        <Attribute Name="category">
          <Value>CONSTANT</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
        </Attribute>
        <Attribute Name="kind">
          <Value>Instance</Value>
          <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
        </Attribute>
        <InternalElement Name="OperatingManual" ID="cf717765-bbad-44e3-ac5f-d16415c8609d">
          <Attribute Name="idShort">
            <Value>OperatingManual</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
          </Attribute>
          <Attribute Name="category">
            <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
          </Attribute>
          <Attribute Name="kind">
            <Value>Instance</Value>
            <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
          </Attribute>
          <Attribute Name="semanticId">

<Value>(ConceptDescription)(local)[IRI]www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Document</Value>
```

```
            <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
          </Attribute>
          <Attribute Name="ordered" AttributeDataType="xs:boolean">
            <Value>false</Value>
            <RefSemantic CorrespondingAttributePath="AAS:SubmodelElementCollection/ordered" />
          </Attribute>
          <Attribute Name="allowDuplicates" AttributeDataType="xs:boolean">
            <Value>false</Value>
            <RefSemantic  CorrespondingAttributePath="AAS:SubmodelElementCollection/allowDuplicates"
/>
          </Attribute>
          <InternalElement Name="DocumentId" ID="f152af3c-893d-427e-b498-94b1324c1316">
            <Attribute Name="idShort">
              <Value>DocumentId</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
            </Attribute>
            <Attribute Name="category">
              <Value>CONSTANT</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
            </Attribute>
            <Attribute Name="kind">
              <Value>Instance</Value>
              <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
            </Attribute>
            <Attribute Name="semanticId">

<Value>(ConceptDescription)(local)[IRI]www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentId/Val</Value
>
              <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
            </Attribute>
            <Attribute Name="value" AttributeDataType="xs:string">
              <Value>3 608 870 A47</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
            </Attribute>
            <RoleRequirements   RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property"
/>
          </InternalElement>
          <InternalElement Name="DocumentClassId" ID="011501d8-8691-405d-9c1d-f2e012ce4fb2">
            <Attribute Name="idShort">
              <Value>DocumentClassId</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
            </Attribute>
            <Attribute Name="kind">
              <Value>Instance</Value>
              <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
            </Attribute>
```

```
            <Attribute Name="semanticId">

<Value>(ConceptDescription)(local)[IRI]www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentClassificatio
n/ClassId</Value>
              <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
            </Attribute>
            <Attribute Name="value" AttributeDataType="xs:string">
              <Value>03-02</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
            </Attribute>
            <RoleRequirements   RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property"
/>
          </InternalElement>
          <InternalElement Name="DocumentClassName" ID="a40f15a7-3412-459d-a431-61873e763349">
            <Attribute Name="idShort">
              <Value>DocumentClassName</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
            </Attribute>
            <Attribute Name="kind">
              <Value>Instance</Value>
              <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
            </Attribute>
            <Attribute Name="semanticId">

<Value>(ConceptDescription)(local)[IRI]www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentClassificatio
n/ClassName</Value>
              <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
            </Attribute>
            <Attribute Name="value" AttributeDataType="xs:langString">
              <Value>Operation (EN) Bedienung (DE)</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
            </Attribute>
            <RoleRequirements   RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property"
/>
          </InternalElement>
          <InternalElement     Name="DocumentClassificationSystem"     ID="4f5eda0f-e5a7-4f6f-b2ac-
319a0a5c3623">
            <Attribute Name="idShort">
              <Value>DocumentClassificationSystem</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
            </Attribute>
            <Attribute Name="kind">
              <Value>Instance</Value>
              <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
            </Attribute>
            <Attribute Name="semanticId">
```

```
<Value>(ConceptDescription)(local)[IRI]www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentClassificatio
n/ClassificationSystem</Value>
              <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
            </Attribute>
            <Attribute Name="value" AttributeDataType="xs:string">
              <Value>VDI2770:2018</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
            </Attribute>
            <RoleRequirements   RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property"
/>
          </InternalElement>
          <InternalElement Name="OrganizationName" ID="7739277a-339c-4181-aaf8-c614d4a7aab4">
            <Attribute Name="idShort">
              <Value>OrganizationName</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
            </Attribute>
            <Attribute Name="kind">
              <Value>Instance</Value>
              <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
            </Attribute>
            <Attribute Name="semanticId">

<Value>(ConceptDescription)(local)[IRI]www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Organization/Organiza
tionName</Value>
              <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
            </Attribute>
            <Attribute Name="value" AttributeDataType="xs:string">
              <Value>CUSTOMER</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
            </Attribute>
            <RoleRequirements   RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property"
/>
          </InternalElement>
          <InternalElement       Name="OrganizationOfficialName"       ID="352d1bf8-0b38-4da3-9c43-
31811f5e2543">
            <Attribute Name="idShort">
              <Value>OrganizationOfficialName</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
            </Attribute>
            <Attribute Name="kind">
              <Value>Instance</Value>
              <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
            </Attribute>
            <Attribute Name="semanticId">

<Value>(ConceptDescription)(local)[IRI]www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Organization/Organiza
tionOfficialName</Value>
```

```
              <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
            </Attribute>
            <Attribute Name="value" AttributeDataType="xs:string">
              <Value>CUSTOMER GmbH</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
            </Attribute>
            <RoleRequirements   RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property"
/>
          </InternalElement>
          <InternalElement Name="Title" ID="7e80d2c5-06db-48bb-84e7-748204dac24f">
            <Attribute Name="idShort">
              <Value>Title</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
            </Attribute>
            <Attribute Name="kind">
              <Value>Instance</Value>
              <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
            </Attribute>
            <Attribute Name="semanticId">

<Value>(ConceptDescription)(local)[IRI]www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Description/Title</Va
lue>
              <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
            </Attribute>
            <Attribute Name="value" AttributeDataType="xs:langString">
              <Value>Operating Manual</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
            </Attribute>
            <RoleRequirements   RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property"
/>
          </InternalElement>
          <InternalElement Name="Language" ID="2e6dbc49-9f76-4ef2-baed-02963f8fd669">
            <Attribute Name="idShort">
              <Value>Language</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
            </Attribute>
            <Attribute Name="kind">
              <Value>Instance</Value>
              <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
            </Attribute>
            <Attribute Name="semanticId">

<Value>(ConceptDescription)(local)[IRI]www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentVersion/Langu
age</Value>
              <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
            </Attribute>
            <Attribute Name="value" AttributeDataType="xs:string">
```

```
              <Value>en-US</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
            </Attribute>
            <RoleRequirements    RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property"
/>
          </InternalElement>
          <InternalElement Name="DigitalFile_PDF" ID="8c5c0be3-d34f-4e57-85cd-875a9af7c9c0">
            <Attribute Name="idShort">
              <Value>DigitalFile_PDF</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
            </Attribute>
            <Attribute Name="category">
              <Value>PARAMETER</Value>
              <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
            </Attribute>
            <Attribute Name="kind">
              <Value>Instance</Value>
              <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
            </Attribute>
            <Attribute Name="semanticId">

<Value>(ConceptDescription)(local)[IRI]www.vdi2770.com/blatt1/Entwurf/Okt18/cd/StoredDocumentRepresentation/DigitalFile</Value>
              <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
            </Attribute>
            <ExternalInterface   Name="FileDataReference"   ID="6d99169c-c2e8-4666-8355-ef8720cecb0e"
RefBaseClassPath="AssetAdministrationShellInterfaceClassLib/FileDataReference">
              <Attribute Name="MIMEType" AttributeDataType="xs:string">
                <Value>application/pdf</Value>
              </Attribute>
              <Attribute Name="refURI" AttributeDataType="xs:anyURI">
                <Value>/aasx/OperatingManual.pdf</Value>
              </Attribute>
            </ExternalInterface>
            <RoleRequirements RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/File" />
          </InternalElement>
          <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/SubmodelElementCollection" />
        </InternalElement>
        <RoleRequirements RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Submodel" />
      </InternalElement><InternalElement        Name="Operations"        ID="226d41de-b71f-48d4-bab2-
0642fab400ff">
        <Attribute Name="identification">
          <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
          <Attribute Name="idType">
            <Value>IRI</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
```

```
          </Attribute>
          <Attribute Name="id">
            <Value>http://i40.customer.com/instance/1/1/AC69B1CB44F07935</Value>
            <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
          </Attribute>
        </Attribute>
        <Attribute Name="idShort">
          <Value>OperationalData</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
        </Attribute>
        <Attribute Name="category">
          <Value>VARIABLE</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
        </Attribute>
        <Attribute Name="kind">
          <Value>Instance</Value>
          <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
        </Attribute>
        <Attribute Name="semanticId">
          <Value>(GlobalReference)(no-local)[IRDI]0173-1#01-AFZ615#016</Value>
          <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
        </Attribute><InternalElement Name="SelectProgram" ID="0bd6f74c-4f27-4c00-983e-
3ce5913a44f3"><InternalElement Name="InputVariables" ID="2a1d9e77-08c3-476f-942e-
66992e9acc8a"><InternalElement Name="ProgramNumber" ID="3e7bffbe-9804-466b-9c40-
4de61fe66c11"><RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property"><Attribute Name="idShort"
AttributeDataType="xs:string">
        <Description>Identifying string of the element within its name space. Constraint AASd-001: In
case of a referable element not being an identifiable element this id is mandatory and used for
referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters,
digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be
matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended
to be defined. It can be used for unique reference in its name space and thus allows better usability
and a more performant implementation. In this case it is similar to the "BrowserPath" in OPC
UA.</Description>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute><Attribute Name="category" AttributeDataType="xs:string">
        <Description>The category is a value that gives further meta information w.r.t. to the class
of the element. It affects the expected existence of attributes and the applicability of
constraints.</Description>
        <Value></Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute><Attribute Name="description" AttributeDataType="xs:string">
        <Description>Description or comments on the element. The description can be provided in several
languages.</Description>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
        <Attribute Name="aml-lang=EN" />
        <Attribute Name="aml-lang=DE" />
      </Attribute><Attribute Name="dataSpecification" AttributeDataType="xs:string">
```

```
        <Description>Global reference to the data specification template used by the element.</Description>
        <Value></Value>
        <RefSemantic CorrespondingAttributePath="AAS:HasDataSpecification/dataSpecification" />
      </Attribute><Attribute Name="kind" AttributeDataType="xs:string">
        <Description>Kind of the element: either template or instance.</Description>
        <DefaultValue>Instance</DefaultValue>
        <Value>Instance</Value>
        <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
        <Constraint Name="">
          <NominalScaledType>
            <RequiredValue>Instance</RequiredValue>
            <RequiredValue>Type</RequiredValue>
          </NominalScaledType>
        </Constraint>
      </Attribute><Attribute Name="semanticId" AttributeDataType="xs:string">
        <Description>Description or comments on the element. The description can be provided in several languages. This attribute has the name of the label and has a value with the label written in the default language. The individual languages are modelled as child attributes. The names of the child attributes are the prefix "aml-lang=" with the expression of the language in compliance with RFC5646. At it, the values of the child attributes are the labels within the respective language.</Description>
        <Value></Value>
        <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
      </Attribute><Attribute Name="qualifier:[TYPE]=[VALUE]" AttributeDataType="xs:string">
        <Description>A qualifier is a type-value-pair that makes additional statements w.r.t. the value of the element. [TYPE] is the value of the attribute type and [VALUE] is the value of the attribute value.</Description>
        <RefSemantic CorrespondingAttributePath="AAS:Qualifiable/qualifier" />
        <Attribute Name="type" AttributeDataType="xs:string">
          <Description>The type describes the type of the qualifier that is applied to the element.</Description>
          <Value></Value>
          <RefSemantic CorrespondingAttributePath="AAS:Qualifier/type" />
        </Attribute>
        <Attribute Name="value" AttributeDataType="xs:string">
          <Description>The qualifier value is the value of the qualifier. Constraint AASd-006: if both, the value and the valueId are present then the value needs to be identical to the value of the referenced coded value in valueId.</Description>
          <Value></Value>
          <RefSemantic CorrespondingAttributePath="AAS:Qualifier/value" />
        </Attribute>
        <Attribute Name="valueId" AttributeDataType="xs:string">
          <Description>Reference to the global unqiue id of a coded value.</Description>
          <Value></Value>
          <RefSemantic CorrespondingAttributePath="AAS:Qualifier/valueId" />
        </Attribute>
      </Attribute><Attribute Name="value" AttributeDataType="xs:string">
        <Description>The value of the property instance. </Description>
```

```
        <RefSemantic CorrespondingAttributePath="AAS:Property/value" />
      </Attribute><Attribute Name="valueId" AttributeDataType="xs:string">
        <Description>Reference to the global unique id if a coded value.</Description>
        <RefSemantic CorrespondingAttributePath="AAS:Property/valueId" />
      </Attribute></RoleRequirements></InternalElement><RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/OperationInputVariables"
/></InternalElement><InternalElement Name="OutputVariables" ID="c10d327e-c393-4a76-8738-
658b70ae883a"><InternalElement Name="Result" ID="64eaaaa1-0682-48ef-a028-
02bbbe4c1083"><RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Property"><Attribute Name="idShort"
AttributeDataType="xs:string">
        <Description>Identifying string of the element within its name space. Constraint AASd-001: In
case of a referable element not being an identifiable element this id is mandatory and used for
referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters,
digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be
matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended
to be defined. It can be used for unique reference in its name space and thus allows better usability
and a more performant implementation. In this case it is similar to the "BrowserPath" in OPC
UA.</Description>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute><Attribute Name="category" AttributeDataType="xs:string">
        <Description>The category is a value that gives further meta information w.r.t. to the class
of the element. It affects the expected existence of attributes and the applicability of
constraints.</Description>
        <Value></Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute><Attribute Name="description" AttributeDataType="xs:string">
        <Description>Description or comments on the element. The description can be provided in several
languages.</Description>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
        <Attribute Name="aml-lang=EN" />
        <Attribute Name="aml-lang=DE" />
      </Attribute><Attribute Name="dataSpecification" AttributeDataType="xs:string">
        <Description>Global reference to the data specification template used by the
element.</Description>
        <Value></Value>
        <RefSemantic CorrespondingAttributePath="AAS:HasDataSpecification/dataSpecification" />
      </Attribute><Attribute Name="kind" AttributeDataType="xs:string">
        <Description>Kind of the element: either template or instance.</Description>
        <DefaultValue>Instance</DefaultValue>
        <Value>Instance</Value>
        <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
        <Constraint Name="">
          <NominalScaledType>
            <RequiredValue>Instance</RequiredValue>
            <RequiredValue>Type</RequiredValue>
          </NominalScaledType>
        </Constraint>
      </Attribute><Attribute Name="semanticId" AttributeDataType="xs:string">
        <Description>Description or comments on the element. The description can be provided in several
languages. This attribute has the name of the label and has a value with the label written in the
```

```
default language. The individual languages are modelled as child attributes. The names of the child attributes are the prefix "aml-lang=" with the expression of the language in compliance with RFC5646. At it, the values of the child attributes are the labels within the respective language.</Description>

        <Value></Value>

        <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />

      </Attribute><Attribute Name="qualifier:[TYPE]=[VALUE]" AttributeDataType="xs:string">

        <Description>A qualifier is a type-value-pair that makes additional statements w.r.t. the value of the element. [TYPE] is the value of the attribute type and [VALUE] is the value of the attribute value.</Description>

        <RefSemantic CorrespondingAttributePath="AAS:Qualifiable/qualifier" />

        <Attribute Name="type" AttributeDataType="xs:string">

          <Description>The type describes the type of the qualifier that is applied to the element.</Description>

          <Value></Value>

          <RefSemantic CorrespondingAttributePath="AAS:Qualifier/type" />

        </Attribute>

        <Attribute Name="value" AttributeDataType="xs:string">

          <Description>The qualifier value is the value of the qualifier. Constraint AASd-006: if both, the value and the valueId are present then the value needs to be identical to the value of the referenced coded value in valueId.</Description>

          <Value></Value>

          <RefSemantic CorrespondingAttributePath="AAS:Qualifier/value" />

        </Attribute>

        <Attribute Name="valueId" AttributeDataType="xs:string">

          <Description>Reference to the global unqiue id of a coded value.</Description>

          <Value></Value>

          <RefSemantic CorrespondingAttributePath="AAS:Qualifier/valueId" />

        </Attribute>

      </Attribute><Attribute Name="value" AttributeDataType="xs:string">

        <Description>The value of the property instance. </Description>

        <RefSemantic CorrespondingAttributePath="AAS:Property/value" />

      </Attribute><Attribute Name="valueId" AttributeDataType="xs:string">

        <Description>Reference to the global unique id if a coded value.</Description>

        <RefSemantic CorrespondingAttributePath="AAS:Property/valueId" />

      </Attribute></RoleRequirements></InternalElement><SupportedRoleClass
RefRoleClassPath="AssetAdministrationShellRoleClassLib/OperationOutputVariables"
/></InternalElement><RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Operation"><Attribute      Name="idShort"
AttributeDataType="xs:string">

        <Description>Identifying string of the element within its name space. Constraint AASd-001: In case of a referable element not being an identifiable element this id is mandatory and used for referring to the element in its name space. Constraint AASd-002: idShort shall only feature letters, digits, underscore ("_"); starting mandatory with a letter. Constraint AASd-003: idShort shall be matched case-insensitive. Note: In case of an identifiable element idShort is optional but recommended to be defined. It can be used for unique reference in its name space and thus allows better usability and a more performant implementation. In this case it is similar to the "BrowserPath" in OPC UA.</Description>

        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />

      </Attribute><Attribute Name="category" AttributeDataType="xs:string">

        <Description>The category is a value that gives further meta information w.r.t. to the class of the element. It affects the expected existence of attributes and the applicability of constraints.</Description>
```

```
        <Value></Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute><Attribute Name="description" AttributeDataType="xs:string">
        <Description>Description or comments on the element. The description can be provided in several languages.</Description>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/description" />
        <Attribute Name="aml-lang=EN" />
        <Attribute Name="aml-lang=DE" />
      </Attribute><Attribute Name="dataSpecification" AttributeDataType="xs:string">
        <Description>Global reference to the data specification template used by the element.</Description>
        <Value></Value>
        <RefSemantic CorrespondingAttributePath="AAS:HasDataSpecification/dataSpecification" />
      </Attribute><Attribute Name="kind" AttributeDataType="xs:string">
        <Description>Kind of the element: either template or instance.</Description>
        <DefaultValue>Instance</DefaultValue>
        <Value>Instance</Value>
        <RefSemantic CorrespondingAttributePath="AAS:HasKind/kind" />
        <Constraint Name="">
          <NominalScaledType>
            <RequiredValue>Instance</RequiredValue>
            <RequiredValue>Type</RequiredValue>
          </NominalScaledType>
        </Constraint>
      </Attribute><Attribute Name="semanticId" AttributeDataType="xs:string">
        <Description>Description or comments on the element. The description can be provided in several languages. This attribute has the name of the label and has a value with the label written in the default language. The individual languages are modelled as child attributes. The names of the child attributes are the prefix "aml-lang=" with the expression of the language in compliance with RFC5646. At it, the values of the child attributes are the labels within the respective language.</Description>
        <Value></Value>
        <RefSemantic CorrespondingAttributePath="AAS:HasSemantics/semanticId" />
      </Attribute><Attribute Name="qualifier:[TYPE]=[VALUE]" AttributeDataType="xs:string">
        <Description>A qualifier is a type-value-pair that makes additional statements w.r.t. the value of the element. [TYPE] is the value of the attribute type and [VALUE] is the value of the attribute value.</Description>
        <RefSemantic CorrespondingAttributePath="AAS:Qualifiable/qualifier" />
        <Attribute Name="type" AttributeDataType="xs:string">
          <Description>The type describes the type of the qualifier that is applied to the element.</Description>
          <Value></Value>
          <RefSemantic CorrespondingAttributePath="AAS:Qualifier/type" />
        </Attribute>
        <Attribute Name="value" AttributeDataType="xs:string">
          <Description>The qualifier value is the value of the qualifier. Constraint AASd-006: if both, the value and the valueId are present then the value needs to be identical to the value of the referenced coded value in valueId.</Description>
          <Value></Value>
```

```
          <RefSemantic CorrespondingAttributePath="AAS:Qualifier/value" />
        </Attribute>
        <Attribute Name="valueId" AttributeDataType="xs:string">
          <Description>Reference to the global unqiue id of a coded value.</Description>
          <Value></Value>
          <RefSemantic CorrespondingAttributePath="AAS:Qualifier/valueId" />
        </Attribute>
      </Attribute></RoleRequirements></InternalElement>



        <RoleRequirements RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/Submodel" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/AssetAdministrationShell" />
    </InternalElement>
  </InstanceHierarchy>
  <InstanceHierarchy Name="AssetAdministrationShellConceptDescriptions">
    <Version>0</Version>
    <InternalElement           Name="Document__URI_www_vdi2770_com_blatt1_Entwurf_Okt18_cd_Document"
ID="06b4113b-6679-4abd-8d05-997ac5b4b715">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">
          <Value>www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Document</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
        <Value>Document</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
        <Value>(GlobalReference)(no-local)[IRI]http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement   Name="EmbeddedDataSpecification"   ID="46c5b4ad-a529-4925-aad4-3153bb615f9a"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">
        <Attribute Name="shortName">
          <Value>Document</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
```

```
        </Attribute>
        <Attribute Name="unit">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="sourceOfDefinition">
          <Value>[ISO 15519-1:2010]</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/sourceOfDefinition"
/>
          <Attribute Name="aml-lang=DE">
            <Value>[ISO 15519-1:2010]</Value>
          </Attribute>
        </Attribute>
        <Attribute Name="dataType">
          <Value>ENTITY</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
        <Attribute Name="definition">
          <Value>Feste und geordnete Menge von für die Verwendung durch Personen bestimmte
Informationen, die verwaltet und als Einheit zwischen Benutzern und System ausgetauscht werden
kann.</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
          <Attribute Name="aml-lang=DE">
            <Value>Feste und geordnete Menge von für die Verwendung durch Personen bestimmte
Informationen, die verwaltet und als Einheit zwischen Benutzern und System ausgetauscht werden
kann.</Value>
          </Attribute>
        </Attribute>
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
    <InternalElement   Name="DocumentId__URI_www_vdi2770_com_blatt1_Entwurf_Okt18_cd_DocumentId_Val"
ID="86ee9067-e344-431c-a02e-888ed75611d8">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">
          <Value>www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentId/Val</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
```

```
        <Value>DocumentIdValue</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="category">
        <Value>CONSTANT</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
        <Value>(GlobalReference)(no-local)[IRI]http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement  Name="EmbeddedDataSpecification"  ID="6f887cac-fd62-4580-86fc-d01d61d5359f"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">
        <Attribute Name="shortName">
          <Value>DocumentId</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
        </Attribute>
        <Attribute Name="unit">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="sourceOfDefinition">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/sourceOfDefinition"
/>
          <Attribute Name="aml-lang=" />
        </Attribute>
        <Attribute Name="dataType">
          <Value>STRING</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
        <Attribute Name="definition">
          <Value>die eigentliche Identifikationsnummer</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
          <Attribute Name="aml-lang=DE">
            <Value>die eigentliche Identifikationsnummer</Value>
          </Attribute>
        </Attribute>
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
    <InternalElement
Name="DocumentClassId__URI_www_vdi2770_com_blatt1_Entwurf_Okt18_cd_DocumentClassification_ClassId"
ID="bdb52663-52b7-4442-bb55-a6232b849b2e">
```

```
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">
          <Value>www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentClassification/ClassId</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
        <Value>DocumentClassId</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="category">
        <Value>CONSTANT</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
        <Value>(GlobalReference)(no-local)[IRI]http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement  Name="EmbeddedDataSpecification"  ID="489218e7-4cb6-4ef3-916d-d6ee3dd75a78"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">
        <Attribute Name="shortName">
          <Value>DocumentClassId</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
        </Attribute>
        <Attribute Name="unit">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="dataType">
          <Value>STRING</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
        <Attribute Name="definition">
          <Value>Eindeutige ID der Klasse in einer Klassifikation.</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
          <Attribute Name="aml-lang=DE">
            <Value>Eindeutige ID der Klasse in einer Klassifikation.</Value>
          </Attribute>
        </Attribute>
```

```
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
    <InternalElement
Name="DocumentClassName__URI_www_vdi2770_com_blatt1_Entwurf_Okt18_cd_DocumentClassification_ClassNam
e" ID="f67bfa5a-0864-408b-820f-8516445aabb1">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">
          <Value>www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentClassification/ClassName</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
        <Value>DocumentClassName</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="category">
        <Value>CONSTANT</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
        <Value>(GlobalReference)(no-local)[IRI]http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement  Name="EmbeddedDataSpecification"  ID="2faa37a4-f185-40af-9784-cc709d08435b"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">
        <Attribute Name="shortName">
          <Value>DocumentClassName</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
        </Attribute>
        <Attribute Name="unit">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="dataType">
          <Value>STRING_TRANSLATABLE</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
```

```
        <Attribute Name="definition">
          <Value>Liste von sprachabhängigen Namen zur ClassId. </Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
          <Attribute Name="aml-lang=DE">
            <Value>Liste von sprachabhängigen Namen zur ClassId. </Value>
          </Attribute>
        </Attribute>
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
    <InternalElement
Name="DocumentClassificationSystem__URI_www_vdi2770_com_blatt1_Entwurf_Okt18_cd_DocumentClassificati
on_ClassificationSystem" ID="ef224659-8fef-485c-b549-c0638d2e69b4">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">

<Value>www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentClassification/ClassificationSystem</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
        <Value>DocumentClassificationSystem</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="category">
        <Value>CONSTANT</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
        <Value>(GlobalReference)(no-local)[IRI]http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement  Name="EmbeddedDataSpecification"  ID="d304accd-8658-4433-a59e-b3a138635142"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">
        <Attribute Name="preferredName">
          <Value>Classification System</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/preferredName" />
```

```
          <Attribute Name="aml-lang=EN">
            <Value>Classification System</Value>
          </Attribute>
          <Attribute Name="aml-lang=DE">
            <Value>Klassifikationssystem</Value>
          </Attribute>
        </Attribute>
        <Attribute Name="shortName">
          <Value>DocumentClassificationSystem</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
        </Attribute>
        <Attribute Name="unit">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="dataType">
          <Value>STRING</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
        <Attribute Name="definition">
          <Value>Eindeutige Kennung für ein Klassifikationssystem. Für Klassifikationen nach VDI 2770
muss "VDI2770:2018" verwenden werden.</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
          <Attribute Name="aml-lang=DE">
            <Value>Eindeutige Kennung für ein Klassifikationssystem. Für Klassifikationen nach VDI
2770 muss "VDI2770:2018" verwenden werden.</Value>
          </Attribute>
        </Attribute>
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
    <InternalElement
Name="OrganizationName__URI_www_vdi2770_com_blatt1_Entwurf_Okt18_cd_Organization_OrganizationName"
ID="b1462079-4d9a-4c21-9433-857e388abc90">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">
          <Value>www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Organization/OrganizationName</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
        </Attribute>
      </Attribute>
```

```
      <Attribute Name="idShort">
        <Value>OrganizationName</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="category">
        <Value>CONSTANT</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
        <Value>(GlobalReference)(no-local)[IRI]http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement  Name="EmbeddedDataSpecification"  ID="ae724f57-cf34-42c6-ae7b-247e44dcff16"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">
        <Attribute Name="preferredName">
          <Value>gebräuchliche Bezeichnung für Organisation</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/preferredName" />
          <Attribute Name="aml-lang=DE">
            <Value>gebräuchliche Bezeichnung für Organisation</Value>
          </Attribute>
          <Attribute Name="aml-lang=EN">
            <Value>organization name</Value>
          </Attribute>
        </Attribute>
        <Attribute Name="shortName">
          <Value>OrganizationName</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
        </Attribute>
        <Attribute Name="unit">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="dataType">
          <Value>STRING</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
        <Attribute Name="definition">
          <Value>Die gebräuchliche Bezeichnung für die Organisation.</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
          <Attribute Name="aml-lang=DE">
            <Value>Die gebräuchliche Bezeichnung für die Organisation.</Value>
          </Attribute>
        </Attribute>
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
```

```
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
    <InternalElement
Name="OrganizationOfficialName__URI_www_vdi2770_com_blatt1_Entwurf_Okt18_cd_Organization_OrganizationOfficialName" ID="3c1163bc-6b30-41e1-beeb-8882d8716513">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">

<Value>www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Organization/OrganizationOfficialName</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
        <Value>OrganizationOfficialName</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="category">
        <Value>CONSTANT</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute>
      <Attribute Name="isCaseOf">
        <Value>(ConceptDescription)(local)[IRDI]0173-1#02-AAO677#002</Value>
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/isCaseOf" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
        <Value>(GlobalReference)(no-local)[IRI]http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement  Name="EmbeddedDataSpecification"  ID="3a4abc63-f4fe-4a68-87fc-c4ffac7eef76"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360Template/DataSpecificationIEC61360">
        <Attribute Name="preferredName">
          <Value>offizieller Name der Organisation</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/preferredName" />
          <Attribute Name="aml-lang=DE">
            <Value>offizieller Name der Organisation</Value>
          </Attribute>
          <Attribute Name="aml-lang=EN">
            <Value>official name of the organization</Value>
```

```
          </Attribute>
        </Attribute>
        <Attribute Name="shortName">
          <Value>OrganizationOfficialName</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
        </Attribute>
        <Attribute Name="unit">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="dataType">
          <Value>STRING</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
        <Attribute Name="definition">
          <Value>Der offizielle Namen der Organisation.</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
          <Attribute Name="aml-lang=DE">
            <Value>Der offizielle Namen der Organisation.</Value>
          </Attribute>
        </Attribute>
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
    <InternalElement
Name="DocumentVersion__URI_www_vdi2770_com_blatt1_Entwurf_Okt18_cd_DocumentVersion"    ID="8d30d85c-
c439-4825-84ff-4b09ddf8154e">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">
          <Value>www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentVersion</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
        <Value>DocumentVersion</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
```

```
        <Value>(GlobalReference)(no-local)[IRI]www.admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement  Name="EmbeddedDataSpecification"  ID="dc856867-37a5-4830-9499-9e7790c4f443"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">
        <Attribute Name="preferredName">
          <Value>Version des Dokuments</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/preferredName" />
          <Attribute Name="aml-lang=DE">
            <Value>Version des Dokuments</Value>
          </Attribute>
        </Attribute>
        <Attribute Name="shortName">
          <Value>DocumentVersion</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
        </Attribute>
        <Attribute Name="unit">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="dataType">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
        <Attribute Name="definition">
          <Value>Zu jedem Dokument muss eine Menge von mindestens einer Dokumentenversion existieren.
Es können auch mehrere Dokumentenversionen ausgeliefert werden.</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
          <Attribute Name="aml-lang=DE">
            <Value>Zu jedem Dokument muss eine Menge von mindestens einer Dokumentenversion
existieren. Es können auch mehrere Dokumentenversionen ausgeliefert werden.</Value>
          </Attribute>
        </Attribute>
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
    <InternalElement
Name="Language__URI_www_vdi2770_com_blatt1_Entwurf_Okt18_cd_DocumentVersion_Language"  ID="4e28a160-
322e-47df-b1b4-aed84e30b0be">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
```

```
        <Attribute Name="id">
          <Value>www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentVersion/Language</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
        <Value>Language</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="category">
        <Value>CONSTANT</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
        <Value>(GlobalReference)(no-local)[IRI]http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement  Name="EmbeddedDataSpecification"  ID="d7cbf133-5315-42a1-aaa4-7570b26550fc"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">
        <Attribute Name="preferredName">
          <Value>Language</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/preferredName" />
          <Attribute Name="aml-lang=EN">
            <Value>Language</Value>
          </Attribute>
          <Attribute Name="aml-lang=DE">
            <Value>Sprache</Value>
          </Attribute>
        </Attribute>
        <Attribute Name="shortName">
          <Value>Language</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
        </Attribute>
        <Attribute Name="unit">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="dataType">
          <Value>STRING</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
        <Attribute Name="definition">
          <Value>Eine Liste der im Dokument verwendeten Sprachen.</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
```

```
          <Attribute Name="aml-lang=DE">
            <Value>Eine Liste der im Dokument verwendeten Sprachen.</Value>
          </Attribute>
        </Attribute>
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
    <InternalElement     Name="Title__URI_www_vdi2770_com_blatt1_Entwurf_Okt18_cd_Description_Title"
ID="afa076b5-48d5-4587-a034-a6d424e6a609">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">
          <Value>www.vdi2770.com/blatt1/Entwurf/Okt18/cd/Description/Title</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
        <Value>Title</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="category">
        <Value>CONSTANT</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
        <Value>(GlobalReference)(no-local)[IRI]http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement   Name="EmbeddedDataSpecification"   ID="3d17b735-6697-407a-8f7c-37008a92cceb"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">
        <Attribute Name="preferredName">
          <Value>Title</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/preferredName" />
          <Attribute Name="aml-lang=EN">
            <Value>Title</Value>
          </Attribute>
          <Attribute Name="aml-lang=DE">
            <Value>Titel</Value>
```

```
          </Attribute>
        </Attribute>
        <Attribute Name="shortName">
          <Value>Title</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
        </Attribute>
        <Attribute Name="unit">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="dataType">
          <Value>STRING_TRANSLATABLE</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
        <Attribute Name="definition">
          <Value>Sprachabhängiger Titel des Dokuments.</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
          <Attribute Name="aml-lang=DE">
            <Value>Sprachabhängiger Titel des Dokuments.</Value>
          </Attribute>
        </Attribute>
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
    <InternalElement
Name="SetDate__URI_www_vdi2770_com_blatt1_Entwurf_Okt18_cd_LifeCycleStatus_SetDate"     ID="cb80e082-
a8a3-461b-99db-b99d9f441560">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">
          <Value>www.vdi2770.com/blatt1/Entwurf/Okt18/cd/LifeCycleStatus/SetDate</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
        <Value>Date</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
```

```
        <Value>(GlobalReference)(no-local)[IRI]www.admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement  Name="EmbeddedDataSpecification"  ID="ab0806e9-3c12-4f81-b60e-0892f922cd74"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">
        <Attribute Name="shortName">
          <Value>SetDate</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
        </Attribute>
        <Attribute Name="unit">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="dataType">
          <Value>DATE</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
        <Attribute Name="definition">
          <Value>Datum und Uhrzeit, an dem der Status festgelegt wurde. Es muss das Datumsformat
„YYYY-MM-dd" verwendet werden (Y = Jahr, M = Monat, d = Tag, siehe ISO 8601).</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
          <Attribute Name="aml-lang=DE">
            <Value>Datum und Uhrzeit, an dem der Status festgelegt wurde. Es muss das Datumsformat
„YYYY-MM-dd" verwendet werden (Y = Jahr, M = Monat, d = Tag, siehe ISO 8601).</Value>
          </Attribute>
        </Attribute>
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
    <InternalElement
Name="DocumentVersionId__URI_www_vdi2770_com_blatt1_Entwurf_Okt18_cd_DocumentVersionId_Val"
ID="6aa33760-97d0-453f-8e16-082326fe4b70">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">
          <Value>www.vdi2770.com/blatt1/Entwurf/Okt18/cd/DocumentVersionId/Val</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
```

```
        <Value>DocumentVersionIdValue</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
        <Value>(GlobalReference)(no-local)[IRI]www.admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement  Name="EmbeddedDataSpecification"  ID="dc76425b-6b2e-48da-a5e9-0afdaf3e7a08"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">
        <Attribute Name="shortName">
          <Value>DocumentVersionId</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
        </Attribute>
        <Attribute Name="unit">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="dataType">
          <Value>STRING</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
        <Attribute Name="definition">
          <Value>Verschiedene Versionen eines Dokuments müssen eindeutig identifizierbar sein. Die
DocumentVersionId stellt eine innerhalb einer Domäne eindeutige Versionsidentifikationsnummer
dar.</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
          <Attribute Name="aml-lang=DE">
            <Value>Verschiedene Versionen eines Dokuments müssen eindeutig identifizierbar sein. Die
DocumentVersionId stellt eine innerhalb einer Domäne eindeutige Versionsidentifikationsnummer
dar.</Value>
          </Attribute>
        </Attribute>
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
    <InternalElement
Name="DigitalFile__URI_www_vdi2770_com_blatt1_Entwurf_Okt18_cd_StoredDocumentRepresentation_DigitalF
ile" ID="d27235b5-a1bd-4832-8abe-84f3cb21d962">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
```

```
        <Attribute Name="id">

<Value>www.vdi2770.com/blatt1/Entwurf/Okt18/cd/StoredDocumentRepresentation/DigitalFile</Value>

          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />

        </Attribute>

      </Attribute>

      <Attribute Name="idShort">

        <Value>DigitalFile</Value>

        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />

      </Attribute>

      <Attribute Name="hasDataSpecification">

        <Value>(GlobalReference)(no-local)[IRI]http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>

        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />

      </Attribute>

      <InternalElement  Name="EmbeddedDataSpecification"  ID="4e500b50-bdaa-450f-81f6-0ba77b8b199a"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">

        <Attribute Name="shortName">

          <Value>DigitalFile</Value>

          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />

        </Attribute>

        <Attribute Name="unit">

          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />

        </Attribute>

        <Attribute Name="dataType">

          <Value>FILE</Value>

          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />

        </Attribute>

        <Attribute Name="definition">

          <Value>Eine Datei, die die DocumentVersion repräsentiert. Neben der obligatorischen PDF/A
Datei können weitere Dateien angegeben werden.</Value>

          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />

          <Attribute Name="aml-lang=DE">

            <Value>Eine Datei, die die DocumentVersion repräsentiert. Neben der obligatorischen PDF/A
Datei können weitere Dateien angegeben werden.</Value>

          </Attribute>

        </Attribute>

        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />

      </InternalElement>

      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />

    </InternalElement>

    <InternalElement Name="MaxRotationSpeed__IRDI_0173-1_02-BAA120_008" ID="cbdc9f25-7e81-4d23-8561-
5358b35fd45e">

      <Attribute Name="identification">

        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
```

```
        <Attribute Name="idType">
          <Value>IRDI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">
          <Value>0173-1#02-BAA120#008</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
        </Attribute>
      </Attribute>
      <Attribute Name="administration">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/administration" />
        <Attribute Name="version">
          <RefSemantic CorrespondingAttributePath="AAS:AdministrativeInformation/version" />
        </Attribute>
        <Attribute Name="revision">
          <Value>2</Value>
          <RefSemantic CorrespondingAttributePath="AAS:AdministrativeInformation/revision" />
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
        <Value>MaxRotationSpeed</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="category">
        <Value>PROPERTY</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute>
      <Attribute Name="isCaseOf">
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/isCaseOf" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement  Name="EmbeddedDataSpecification"  ID="81b88f57-8b23-4529-ad12-ffe481881c28"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">
        <Attribute Name="preferredName">
          <Value>max. Drehzahl</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/preferredName" />
          <Attribute Name="aml-lang=de">
            <Value>max. Drehzahl</Value>
          </Attribute>
          <Attribute Name="aml-lang=en">
            <Value>Max. rotation speed</Value>
          </Attribute>
```

```
        </Attribute>
        <Attribute Name="shortName">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
        </Attribute>
        <Attribute Name="unit">
          <Value>1/min</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="unitId">
          <Value>(GlobalReference)(no-local)[IRDI]0173-1#05-AAA650#002</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unitId" />
        </Attribute>
        <Attribute Name="dataType">
          <Value>INTEGER_MEASURE</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
        <Attribute Name="definition">
          <Value>Höchste zulässige Drehzahl, mit welcher der Motor oder die Speiseinheit betrieben
werden darf</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
          <Attribute Name="aml-lang=de">
            <Value>Höchste zulässige Drehzahl, mit welcher der Motor oder die Speiseinheit betrieben
werden darf</Value>
          </Attribute>
          <Attribute Name="aml-lang=en">
            <Value>Greatest permissible rotation speed with which the motor or feeding unit may be
operated</Value>
          </Attribute>
        </Attribute>
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
    <InternalElement     Name="MaxTorque__IRDI_0173-1_02-BAE098_004"     ID="7b8a5bf5-09ce-4228-9079-
49586d7a81c2">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRDI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">
          <Value>0173-1#02-BAE098#004</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
        </Attribute>
```

```
      </Attribute>
      <Attribute Name="idShort">
        <Value>MaxTorque</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="category">
        <Value>PROPERTY</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement  Name="EmbeddedDataSpecification"  ID="1cf999e8-8b86-41a6-8aae-256ea1b6612b"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">
        <Attribute Name="preferredName">
          <Value>Max. torque</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/preferredName" />
          <Attribute Name="aml-lang=EN">
            <Value>Max. torque</Value>
          </Attribute>
        </Attribute>
        <Attribute Name="shortName">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
        </Attribute>
        <Attribute Name="unit">
          <Value>Nm</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="unitId">
          <Value>(GlobalReference)(no-local)[IRDI]0173-1#05-AAA212#003</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unitId" />
        </Attribute>
        <Attribute Name="dataType">
          <Value>REAL_MEASURE</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
        <Attribute Name="definition">
          <Value>Greatest permissible mechanical torque which the motor can pass on at the drive
shaft</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
          <Attribute Name="aml-lang=EN">
            <Value>Greatest permissible mechanical torque which the motor can pass on at the drive
shaft</Value>
          </Attribute>
          <Attribute Name="aml-lang=DE">
```

```
            <Value>Größtes mechanisch zulässiges Drehmoment, welches der Motor an der Abtriebswelle
abgeben kann</Value>
          </Attribute>
        </Attribute>
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
    <InternalElement             Name="RotationSpeed__URI_http_customer_com_cd_1_1_18EBD56F6B43D895"
ID="8e67b9bf-52eb-49da-8efc-3f6c50f14db6">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">
          <Value>http://customer.com/cd/1/1/18EBD56F6B43D895</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
        <Value>RotationSpeed</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="category">
        <Value>PROPERTY</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
        <Value>(GlobalReference)(no-local)[IRI]http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement   Name="EmbeddedDataSpecification"   ID="03b21d8d-34c0-46e5-9a56-1d1baf988c3f"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">
        <Attribute Name="preferredName">
          <Value>Aktuelle Drehzahl</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/preferredName" />
          <Attribute Name="aml-lang=DE">
            <Value>Aktuelle Drehzahl</Value>
          </Attribute>
          <Attribute Name="aml-lang=EN">
            <Value>Actual rotation speed</Value>
```

```
          </Attribute>
        </Attribute>
        <Attribute Name="shortName">
          <Value>RotationSpeed</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
        </Attribute>
        <Attribute Name="unit">
          <Value>1/min</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="unitId">
          <Value>(GlobalReference)(no-local)[IRDI]0173-1#05-AAA650#002</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unitId" />
        </Attribute>
        <Attribute Name="dataType">
          <Value>INTEGER_MEASURE</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
        <Attribute Name="definition">
          <Value>Aktuelle Drehzahl, mit welcher der Motor oder die Speiseinheit betrieben wird</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
          <Attribute Name="aml-lang=DE">
            <Value>Aktuelle Drehzahl, mit welcher der Motor oder die Speiseinheit betrieben
wird</Value>
          </Attribute>
          <Attribute Name="aml-lang=EN">
            <Value>Actual rotation speed with which the motor or feeding unit is operated</Value>
          </Attribute>
        </Attribute>
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
    <InternalElement Name="Torque__URI_http_customer_com_cd_1_1_18EBD56F6B43D896" ID="f72c1826-0b8e-
45be-9afe-ff3c043bb602">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">
          <Value>http://customer.com/cd/1/1/18EBD56F6B43D896</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
```

```
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
        <Value>Torque</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="category">
        <Value>PROPERTY</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
        <Value>(GlobalReference)(no-local)[IRI]http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement  Name="EmbeddedDataSpecification"  ID="e9e5fdf4-ccdf-4542-8934-aee56fbe76cc"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">
        <Attribute Name="preferredName">
          <Value>Drehmoment</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/preferredName" />
          <Attribute Name="aml-lang=DE">
            <Value>Drehmoment</Value>
          </Attribute>
          <Attribute Name="aml-lang=EN">
            <Value>Torque</Value>
          </Attribute>
        </Attribute>
        <Attribute Name="shortName">
          <Value>Torque</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
        </Attribute>
        <Attribute Name="unit">
          <Value>Nm</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="unitId">
          <Value>(GlobalReference)(no-local)[IRDI]0173-1#05-AAA212#003</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unitId" />
        </Attribute>
        <Attribute Name="dataType">
          <Value>REAL_MEASURE</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
        <Attribute Name="definition">
```

```
          <Value>Actual mechanical torque which the motor passes on at the drive shaft</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
          <Attribute Name="aml-lang=EN">
            <Value>Actual mechanical torque which the motor passes on at the drive shaft</Value>
          </Attribute>
          <Attribute Name="aml-lang=DE">
            <Value>Aktuelles Drehmoment, welches der Motor an der Abtriebswelle abgibt</Value>
          </Attribute>
        </Attribute>
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
    <InternalElement    Name="CoolingType__IRDI_0173-1_02-BAE122_006"    ID="67c17a20-b270-4e2f-859f-
c9a1a917f8ad">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRDI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">
          <Value>0173-1#02-BAE122#006</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
        <Value>CoolingType</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="category">
        <Value>PROPERTY</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
        <Value>(GlobalReference)(no-local)[IRI]http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement   Name="EmbeddedDataSpecification"   ID="eb447103-035a-4045-8023-c6dc9e61dd61"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">
        <Attribute Name="preferredName">
          <Value>Art der Kühlung</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/preferredName" />
```

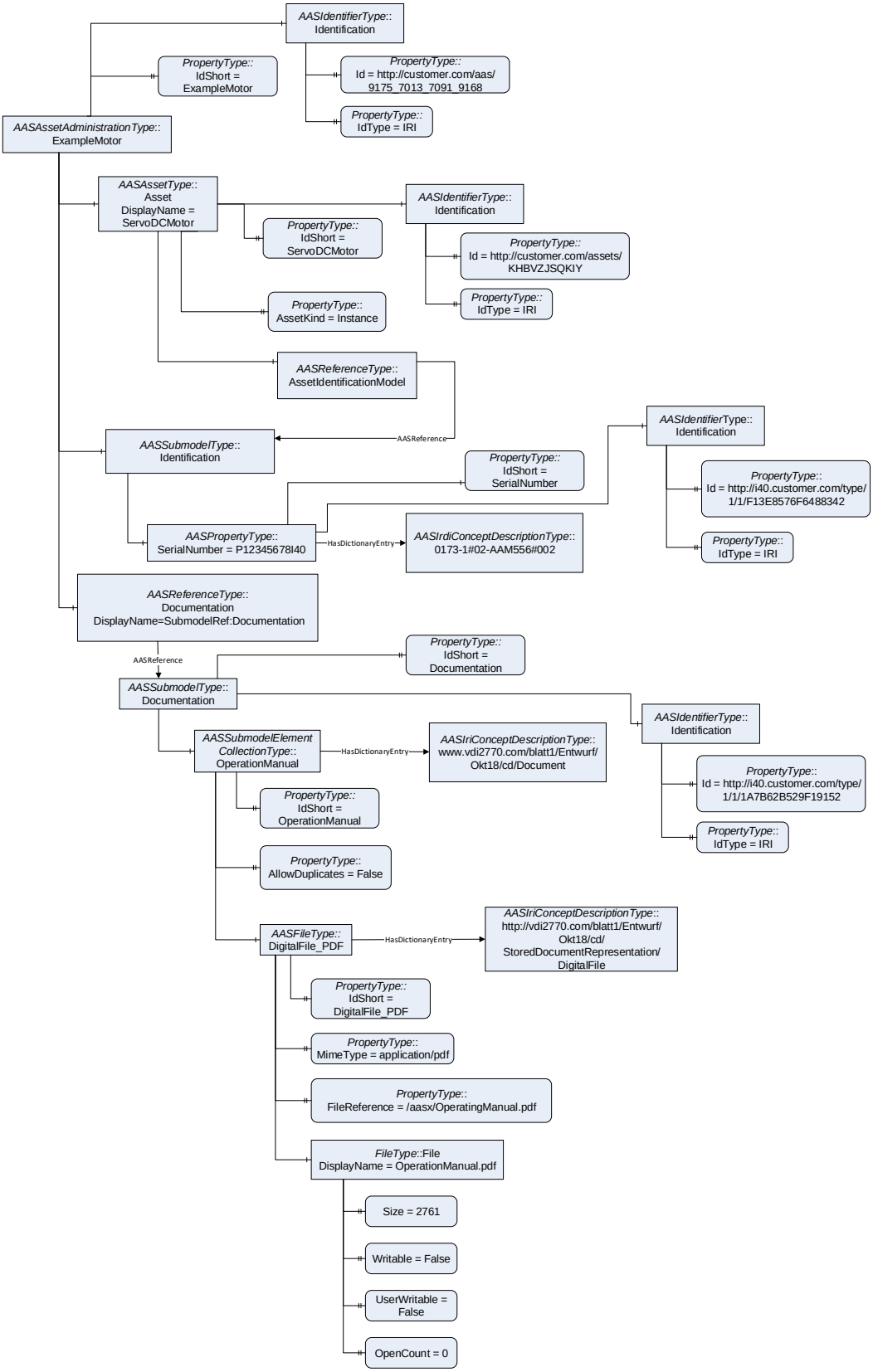
```
          <Attribute Name="aml-lang=DE">
            <Value>Art der Kühlung</Value>
          </Attribute>
          <Attribute Name="aml-lang=EN">
            <Value>Cooling type</Value>
          </Attribute>
        </Attribute>
        <Attribute Name="shortName">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
        </Attribute>
        <Attribute Name="unit">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="dataType">
          <Value>STRING</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
        <Attribute Name="definition">
          <Value>Zusammenfassung verschiedener Kühlarten, um für Suchmerkmale zu einer begrenzten
Auswahl zu kommen</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/definition" />
          <Attribute Name="aml-lang=DE">
            <Value>Zusammenfassung verschiedener Kühlarten, um für Suchmerkmale zu einer begrenzten
Auswahl zu kommen</Value>
          </Attribute>
          <Attribute Name="aml-lang=EN">
            <Value>Summary of various types of cooling, for use as search criteria that limit a
selection</Value>
          </Attribute>
        </Attribute>
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
    <InternalElement      Name="BAB657__IRDI_0173-1_07-BAB657_003"      ID="1319f7a2-eef9-4d93-8563-
3158d1d5b159">
      <Attribute Name="identification">
        <RefSemantic CorrespondingAttributePath="AAS:Identifiable/identification" />
        <Attribute Name="idType">
          <Value>IRDI</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/idType" />
        </Attribute>
        <Attribute Name="id">
          <Value>0173-1#07-BAB657#003</Value>
          <RefSemantic CorrespondingAttributePath="AAS:Identifier/id" />
```

```
        </Attribute>
      </Attribute>
      <Attribute Name="idShort">
        <Value>BAB657</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/idShort" />
      </Attribute>
      <Attribute Name="category">
        <Value>VALUE</Value>
        <RefSemantic CorrespondingAttributePath="AAS:Referable/category" />
      </Attribute>
      <Attribute Name="hasDataSpecification">
        <Value>(GlobalReference)(no-local)[IRI]http://admin-
shell.io/DataSpecificationTemplates/DataSpecificationIEC61360</Value>
        <RefSemantic CorrespondingAttributePath="AAS:ConceptDescription/dataSpecification" />
      </Attribute>
      <InternalElement  Name="EmbeddedDataSpecification"  ID="f5a3ccfb-2f4b-4450-acc3-af5bac47af66"
RefBaseSystemUnitPath="AssetAdministrationShellDataSpecificationTemplates/DataSpecificationIEC61360T
emplate/DataSpecificationIEC61360">
        <Attribute Name="preferredName">
          <Value>open circuit, external cooling</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/preferredName" />
          <Attribute Name="aml-lang=EN">
            <Value>open circuit, external cooling</Value>
          </Attribute>
          <Attribute Name="aml-lang=DE">
            <Value>offener Kreis, Fremdkühlung </Value>
          </Attribute>
        </Attribute>
        <Attribute Name="shortName">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/shortName" />
        </Attribute>
        <Attribute Name="unit">
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/unit" />
        </Attribute>
        <Attribute Name="dataType">
          <Value>STRING</Value>
          <RefSemantic CorrespondingAttributePath="IEC:DataSpecificationIEC61360/dataType" />
        </Attribute>
        <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/DataSpecificationContent" />
      </InternalElement>
      <RoleRequirements
RefBaseRoleClassPath="AssetAdministrationShellRoleClassLib/ConceptDescription" />
    </InternalElement>
  </InstanceHierarchy>
</CAEXFile>
```

Annex I. OPC UA Companion Specification

The OPC UA nodeset for the OPC UA Companion Specification of the Asset Administration Shell will be published as an OPC UA Companion Specification. In Figure 143 an extract of the Servo DC Motor example is shown.

Figure 143 Example Servo Motor in OPC UA (Extract)



Annex J. Metamodel Changes

i. Metamodel Changes w/o Security Part

Table 31 Changes w.r.t. V1.0 w/o Security

nc[46]	V2.0 Change w.r.t. V1.0	Comment
(x) [47]	anySimpleTypeDef	Type starts now with capital letter: AnySimpleTypeDef Type changed from string to values representing xsd-type anySimpleType
	Asset	Does not inherit from HasKind any longer (but attribute kind remains)
	Asset/kind	Now of type "AssetKind" instead of "Kind". Instead of value Type and Instance now value Template and Instance
	AssetAdministrationShell/security	Now optional to support passive AAS of type 1
	Code	Data type removed, not used any longer
x	DataSpecificationIEC61360/shortName	Type changed from string to LangStringSet Cardinality changed from mandatory to optional
x	DataSpecificationIEC61360/sourceOfDefinition	Type changed from langString to string
(x)[48]	DataSpecificationIEC61360/dataType	Type changed from string to Enumeration Cardinality changed from mandatory to optional
x	DataSpecificationIEC61360/code	Attribute code removed
	DataSpecificationIEC61360/definition	Cardinality changed from mandatory to optional
	HasDataSpecification	Was abstract before
	HasDataSpecification/hasDataSpecification	Renamed to HasDataSpecification/dataSpecification
x	HasKind/kind	Now of type "ModelingKind" instead of "Kind". Values changed: Type now Template; Instance remains
x	File/value	File name not without but with extension
x	Identifiable/description	Type changed from langString to LangStringSet
x	IdentifierType/URI	URI renamed to IRI
	Kind	Type Kind removed and substituted by types AssetKind and ModelingKind
x	OperationVariable	Does not inherit from SubmodelElement any longer

[46] "nc=x" means not backward compatible

[47] Implicitly there was a constraint restricting the values to the values in the enumeration. This is now formalized.

[48] Implicitly there was the constraints that only IEC61360 data types are allowed to be used. This is now formalized.

nc[46]	V2.0 Change w.r.t. V1.0	Comment
	Property/value	Type changed from anySimpleTypeDef to ValueDataType
x	Qualifier/qualifierType	Renamed to Qualifier/type
x	Qualifier/qualifierValue	Renamed to Qualifier/value Type changed from AnySimpleTypeDef to ValueDataType
x	Qualifier/qualifierValueId	Renamed to Qualifier/valueId
x	Referable/idShort	Not mandatory, was optional (but with constaints for defined elements)
x	Reference/key	Cardinality changed from 0..* to 1..*

Table 32 New Elements in Metamodel V1.0 w/o Security

V2.0	Comment
AnnotatedRelationshipElement	New submodel element, inheriting from RelationshipElement
Asset/billOfMaterial	New attribute
AssetKind	New enumeration type
BasicEvent	New submodel element, inherits from Event
Capability	New submodel element
DataSpecificationIEC61360/valueList	For value lists (string)
DataSpecificationIEC61360/value	For coded and explicit values
DataSpecificationIEC61360/valueId	For coded values
DataSpecificationIEC61360/levelType	For Ranges
DataSpecificationPhysicalUnit	New data specification template
DataTypeIEC61360	New enumeration type
Entity	New submodel Element
EntityType	New enumeration type
IdentifierType	Is a subset of KeyType Enumeration
KeyElements/FragmentReference	New value FragmentReference as part of KeyElements Enumeration
LocalKeyType	Is a subset of KeyType Enumeration
LocalKeyType/FragmentId	New value for KeyType Enumeration (via subset LocalKeyType)
LangStringSet	New type, used for example in MultiLanguageProperty
levelType	New enumeration type
ModelingKind	New enumeration type
MultiLanguageProperty	New submodel element
Qualifier/valueType	New attribute to be consistent with valueType of Property etc.
Range	New submodel element

V2.0	Comment
ReferableElements/BasicEvent	New enumeration value
ReferableElements/Capability	New enumeration value
ReferableElements/Event	New enumeration value
ReferableElements/MultiLanguageProperty	New enumeration value
ReferableElements/Range	New enumeration value
ValueDataType	New type, used for example for Property value
ValueList	New class
ValueReferencePairType	New class

ii. Metamodel Changes – Security Part

Table 33 Changes Metamodel w.r.t. V1.0 Security

nc[49]	V2.0 Change w.r.t. V1.0	Comment
x	AccessControl/selectableEnvironmentAttributes	Type changed from Submodel to Submodel*
	AccessPermissionRule/permissionsPerObject	Cardinality now consistent for figure and table: 0..*
x	AccessPermissionRule/targetSubjectAttributes	Cardinality changed from 1..* to 1
	Certificate	Was abstract, now not abstract and contains attributes (see in table New)
x	PermissionKind/allow	Now PermissionKind/Allow start with capital letter for enumeration values
x	PermissionKind/deny	Now PermissionKind/Deny start with capital letter for enumeration values
x	PermissionKind/not applicable	Now PermissionKind/NotApplicable start with capital letter for enumeration values
x	PermissionKind/Undefined	Now PermissionKind/Undefined start with capital letter for enumeration values
	PermissionsPerObject	Name now consistent for figure and table (in table PermissionPerObject, needs to be PermissionsPerObject)
x	PolicyAdministrationPoint/externalAccessControl	Type changed from Endpoint to Boolean, cardinality 1
x	PolicyInformationPoints/externalInformationPoint	Type changed from Endpoint to Boolean, cardinality 1 externalInformationPoint renamed to externalInformationPoints
x	Security/trustAnchor	Renamed to Security/certificate

[49] "nc=x" means not backward compatible

Table 34 New Elements in Metamodel w.r.t. Security

V2.0	**Comment**
BlobCertificate	New class inheriting from Certificate
Certificate	Abstract class: was foreseen in V1.0 but not yet modelled
Security/requiredCertificateExtension	New attribute
PolicyEnforcementPoint	Was foreseen in V1.0 but not yet modelled
PolicyEnforcementPoint/externalPolicyEnforcementPoint	
PolicyDecisionPoint	Was foreseen in V1.0 but not yet modelled
PolicyDecisionPoint/externalPolicyDecisionPoint	

Annex K. 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] "Referenzarchitekturmodell Industrie 4.0 (RAMI4.0) / Reference Architecture Model Industrie 4.0 (RAMI4.0) / Modèle de reference de l'architecture de l'industrie 4.0 (RAMI4.0)", DIN SPEC 91345:2016-04 ICS 03.100.01; 25.040.01; 35.240.50, April 2016. [Online]. Available: https://www.beuth.de/en/technical-rule/din-spec-91345-en/250940128

[4] "Structure of the Administration Shell, continuation of the development of the reference model for the Industrie 4.0 component", Plattform Industrie 4.0, Working Paper, April 2016. [Online]. Available: https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/structure-of-the-administration-shell.html

[5] "Which criteria do Industrie 4.0 products need to fulfil? Guideline 2019", Federal Ministry for Economic Affairs and Energy (BMWi), April 2019. [Online]. Available: https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/criteria-industrie-40-products.html/

[5b] « Produktkriterien 2020. Welche Kriterien müssen Industrie-4.0-Produkte 2020 erfüllen ? » Leitfaden. November 2019, ZVEI. [Online]. Available: https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/-2019/November/Produktkriterien_2020_-_Welche_Kritierien_muessen_Industrie-4.0-Produkte_2020_erfuellen_/ZVEI_Leitfaden_I40_Produktkriterien_2020.pdf

[6] "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/

[7] "Aspects of the research roadmap in application scenarios", Plattform Industrie 4.0, working paper, April 2016. [Online]. Available: http://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/aspects-of-the-research-roadmap.html

[8] "Fortschreibung der Anwendungsszenarien der Plattfrom Industrie 4.0 (German)"; Plattform Industrie 4.0, Ergebnispapier, October 2016. [Online]. Available: https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/fortschreibung-anwendungsszenarien.html

[9] "Security in RAMI4.0", Plattform Industrie 4.0, Berlin, technical overview, April 2016. [Online]. Available: http://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/security-rami40-en.html

[10] "Die Deutsche Normungs-Roadmap Industrie 4.0 / The German standardization roadmap Industrie 4.0", DKE Deutsche Kommission Elektrotechnik, Elektronik Informationstechnik im DIN und VDE, Version 2.0, 2015. [Online]. Available: http://www.din.de/de/forschung-und-innovation/industrie4-0/roadmap-industrie40-62178

[10a] "Weiterentwicklung des Interaktionsmodells für Industrie 4.0-Komponenten", Plattform Industrie 4.0, discussion paper, November 2016. [Online]. Available: https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/interaktionsmodell-i40-komponenten-it-gipfel.html

[11] "Definition of terms relating to Industrie 4.0", Fraunhofer IOSB and VDI GMA Fachausschuss 7.21. Accessed: 12.2.2017. [Online]. Available: http://i40.iosb.fraunhofer.de/FA7.21%20Begriffe;

[12] "Relationships between I4.0 Components – Composite Components and Smart Production", Plattform Industrie 4.0, Berlin, working paper, June 2017. [Online]. Available: https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/hm-2018-relationship.htmll

[13] “Industrie 4.0 Plug-and-Produce for Adaptable Factories”; Plattform Industrie 4.0, Berlin, working paper, June 2017. [Online]. Available: http://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/Industrie-40-%20Plug-and-Produce.html

[14] “Security der Verwaltungsschale / Security of the Administration Shell”, Plattform Industrie 4.0, Berlin, working paper, April 2017. [Online]. Available: http://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/security-der-verwaltungsschale.html

[15] “Data Exchange on the Base of Property Value Statements (PVSX)”, DIN SPEC 92000:2019-09, 2019 September.

[16] “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

[17] “I4.0-Sprache. Vokabular, Nachrichtenstruktur und semantische Interaktionsprotokolle der I4.0-Sprache (German)”, Plattform Industrie 4.0 in Kooperation mit VDI/VDE GMA Fachausschuss 7.20, April 2018. [Online]. Available: https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/hm-2018-sprache.html

[18] “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

[19] „Zugriffssteuerung für Industrie 4.0-Komponenten zur Anwendung von Herstellern, Betreibern und Integratoren (German)“, Plattform Industrie 4.0, Diskussionspapier, Nov. 2018, [Online]. Available: https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/zugriffssteuerung-industrie40-komponenten.html

[20] “Industrial automation systems and integration — Exchange of characteristic data — Part 10: Characteristic data exchange format”, Technical Specification ISO/TS 29002-10:2009(E), 2009

[21] “Reference Architecture Model for the Industrial Data Space”, Fraunhofer in cooperation with Industrial Data Space Association, 2017. [Online]. Available: https://www.fit.fraunhofer.de/content/dam/fit/en/documents/Industrial-Data-Space_Reference-Architecture-Model-2017.pdf

[22] Vincent Hu, David Ferraiolo, Rick Kuhn, Adam Schnitzer, Kenneth Sandlin, Robert Miller and Karen Scarfone, “ Guide to Attribute Based Access Control (ABAC) Definition and Considerations”, NIST Special Publication 800-162, Jan. 2014. [Online]. Available: http://dx.doi.org/10.6028/NIST.SP.800-162

[23] “Smart Manufacturing - Reference Architecture Model Industry 4.0 (RAMI4.0)”, IEC PAS 63088, International Electrotechnical Commission (IEC), 2017

[24] “Sustainability of Digital Formats: Planning for Library of Congress Collections. Open Packaging Conventions (Office Open XML)”, ISO 29500-2:2008-2012, 2012. [Online]. Available: https://www.loc.gov/preservation/digital/formats/fdd/fdd000363.shtml

[25] “Standardization of Office Open XML”, Wikipedia. Accessed: 2019-01-26 [Online]. Available: https://en.wikipedia.org/wiki/Standardization_of_Office_Open_XML

[26] “OpenDocument standardization”, Wikipedia. Accessed: 2019-01-26 [Online]. Available: https://en.wikipedia.org/wiki/OpenDocument_standardization

[27] “The Digital Signing Framework of the Open Packaging Conventions”. Accessed: 2019-01-26. [Online]. Available: https://msdn.microsoft.com/en-us/library/aa905326.aspx

[28] “Open Packaging Conventions Fundamentals”. Accessed: 2019-01-26 [Online]. Available: https://msdn.microsoft.com/en-us/library/windows/desktop/dd742818(v=vs.85).aspx

[29] “What is a digital signature? Fundamental principles”. Accessed: 2019-01-26. [Online]. Available: http://securityaffairs.co/wordpress/5223/digital-id/what-is-a-digital-signature-fundamental-principles.html

[30] “Sustainability of Digital Formats: Planning for Library of Congress Collections. Document Container File: Core (based on ZIP 6.3.3)”. Accessed: 2019-01-26. [Online]. Available: https://www.loc.gov/preservation/digital/formats/fdd/fdd000361.shtml

[31] “System.IO.Packaging Namespace”, MSDN, Accessed: 2019-01-26 [Online]. Available: https://msdn.microsoft.com/en-us/library/system.io.packaging(v=vs.110).aspx

[32] “Reference Model for Industrie 4.0 Service Architectures – Part 1: Basic Concepts of an Interaction-based Architecture”, DIN SPEC 16593-1, Beuth-Verlag: Berlin, Germany, 2018. [Online]. Available: https://www.beuth.de/en/technical-rule/din-spec-16593-1/287632675

[33] ISO 13854-42 “Standard data element types with associated classification scheme – Part 1: Definitions – Principles and methods” Edition 4.0, 2017-07

[34] IEC 61360-1 “Standard data element types with associated classification scheme – Part 1: Definitions – Principles and methods”, Edition 4.0, 2017-07

[35] ISO/TS 29002-10:2009(E) “Industrial automation systems and integration — Exchange of characteristic data — Part 10: Characteristic data exchange format”, First edition 2009-12-01

[36] A. Bayha, J. Bock, B. Boss, C. Diedrich, S. Malakuti “Describing Capabilities of Industrie 4.0 Components”. (in progress)

[37] AutomationML Association: “Application Recommendations: Asset Administration Shell Representation (AR 004E) “, Version 1.0.0, 20.11.2019, [Online]. Available: https://www.automationml.org/o.red.c/dateien.html.

[38] H. Knublauch, D. Knotokostas “Shapes Constraint Language (SHACL) “ W3C Recommendation, 2017, [Online]. Available: https://www.w3.org/TR/shacl/ (accessed at 17. November 2019)

[39] “OPC UA Companion Specification ‘Asset Administration Shell” (in progress)

[40] “AASX Package Explorer. Software” Download: https://github.com/admin-shell/aasx-package-explorer

[41] “AAS Repository. Repository for Information and Code for the Asset Administration Shell” ttps://github.com/admin-shell

[42] “Eclipse BaSyx”. [Online]. Available: https://www.eclipse.org/basyx/

[43] DIN SPEC 91406 “Automatische Identifikation im Typenschild” (in progress)

[44] F. Manola, E. Miller “RDF 1.1 Primer“ W3C Recommendation, 2014, [Online]. Available: https://www.w3.org/TR/rdf11-primer/ (accessed at 17. November 2019)

[45] Gruber, Thomas R. "A translation approach to portable ontology specifications." Knowledge acquisition 5.2 (1993): 199-220. [Online]. Available: https://tomgruber.org/writing/ontolingua-kaj-1993.htm

AUTHORS of Version 1.0 and Version 2.0

Sebastian Bader, Fraunhofer IAIS

Erich Barnstedt, Microsoft Deutschland GmbH

Dr. Heinz Bedenbender, VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik (GMA)

Meik Billmann, ZVEI – Zentralverband Elektrotechnik- und Elektronikindustrie

Dr. Birgit Boss, Robert Bosch GmbH

Dr. André Braunmandl, BSI

Erich Clauer, SAP SE

Torben Deppe, RWTH Aachen

Professor Dr. Christian Diedrich, ifak - Institut f. Automation und Kommunikation e.V. Magdeburg

Björn Flubacher, BSI

Wolfgang Fritsche, IABG mbH

Kai Garrels, ABB STOTZ-KONTAKT GmbH

Dr. Andreas Graf Gatterburg, Hilscher Gesellschaft für Systemautomation mbH

Martin Hankel, Bosch Rexroth AG

Oliver Hillermeier, SAP SE

Dr. Michael Hoffmeister, Festo AG & Co. KG

Michael Jochem, Robert Bosch GmbH

Dr. Yevgen Kogan, KUKA Deutschland GmbH

Dr. Heiko Koziolek, ABB AG

Dr. Christoph Legat, Hekuma GmbH

Professor Arndt Lüder, Otto-v.-Guericke Universität

Dr. Marco Mendes, Schneider Electric Automation GmbH

Dr. Jörg Neidig, Siemens AG

Andreas Orzelski, PHOENIX CONTACT GmbH & Co. KG

Florian Pethig, Fraunhofer IOSB-INA

Manuel Sauer, SAP SE

Marc Schier, Microsoft Deutschland GmbH

Dr. Miriam Schleipen, Siemens AG

Michael Schmitt, SAP SE

Tizian Schröder, Otto-von-Guericke-Universität Magdeburg

André Uhl, Schneider Electric Automation GmbH

Dr. Ljilijana Stojanovic, Fraunhofer IOSB

Dr. Thomas Usländer, Fraunhofer IOSB

➔ Continued next page

AUTHORS of Version 1.0 and Version 2.0 - continued

Thomas Walloschke, Fujitsu Technology Solutions GmbH

Bernd Waser, Murrelektronik GmbH

Jörg Wende, IBM Deutschland GmbH

Mathias Wiegand, Festo AG & Co. KG

Constantin Ziesche, Robert Bosch GmbH

This working paper has been elaborated in the Working Subgroup “Asset Administration Shell” of the Working Group on “Reference Architectures, Standards and Norms “ of Plattform Industrie 4.0 and the working group “Models and Standards” of the ZVEI in cooperation with the Working Groups “Security of networked Systems” (Plattform Industrie 4.0) and “Security” (ZVEI). The OPC UA Mapping has been elaborated in the joint working group “I4AAS” of the OPC Foundation, ZVEI, VDMA and the Plattform Industrie 4.0. The AutomationML Mapping has been elaborated in the joint working group of AutomationML e.V. and the Plattform Industrie 4.0.

www.plattform-i40.de