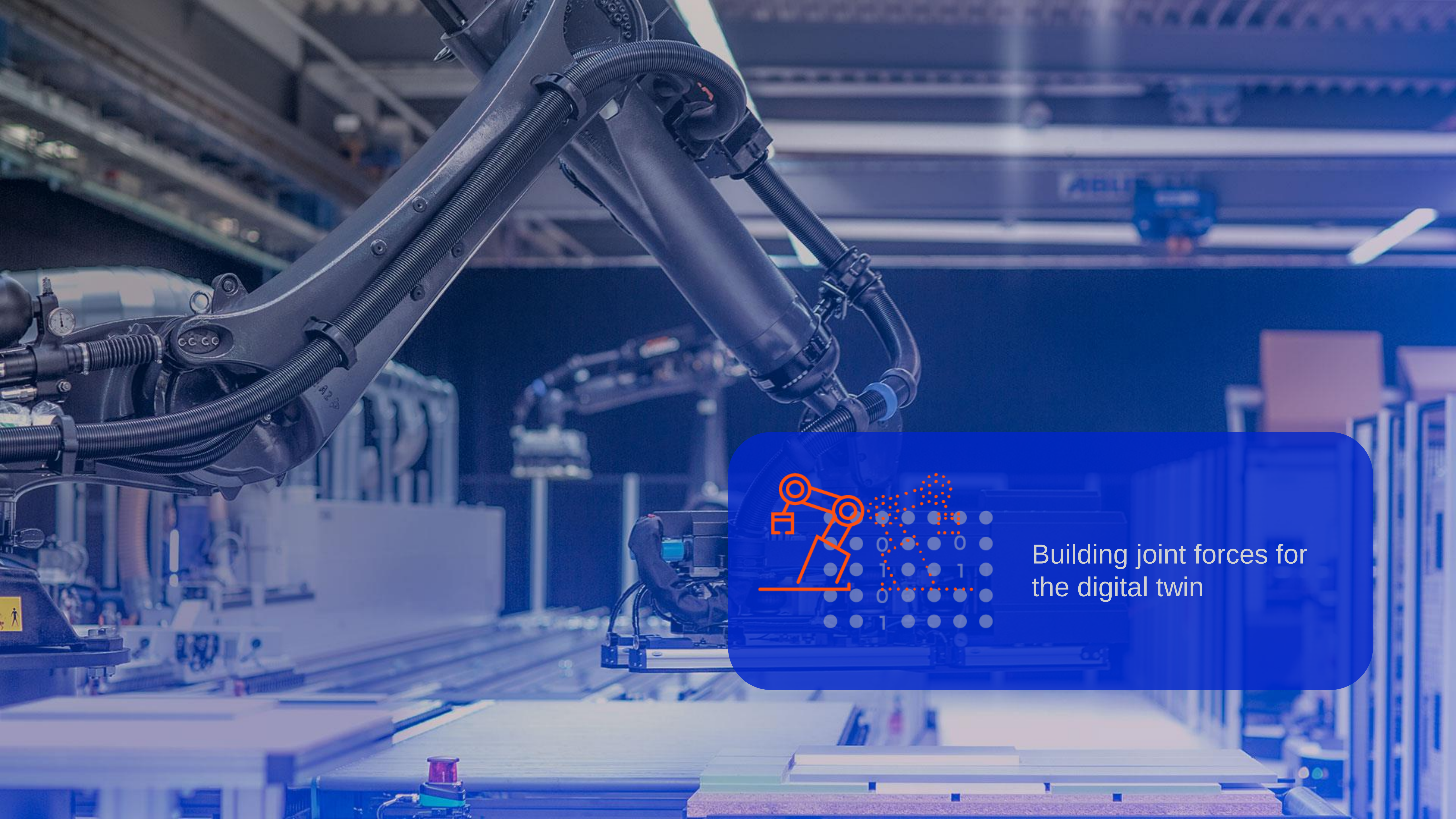
IDTA Industrial Digital Twin Association

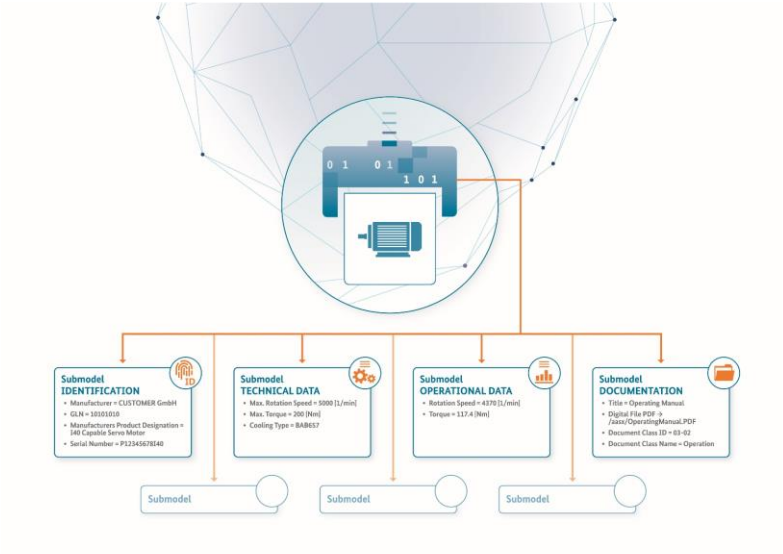
Tutorial

Details of the Asset Administration Shell Taken Literally – Learn about the underlying UML Information Model V3.0RC02



Building joint forces for the digital twin

From Concept to Specification



Submodel IDENTIFICATION
- Manufacturer = CUSTOMER GmbH
- GLN = 10101010
- Manufacturers Product Designation = I40 Capable Servo Motor
- Serial Number = P12345678I40

Submodel TECHNICAL DATA
- Max. Rotation Speed = 5000 [1/min]
- Max. Torque = 200 [Nm]
- Cooling Type = BAB657

Submodel OPERATIONAL DATA
- Rotation Speed = 4370 [1/min]
- Torque = 117.4 [Nm]

Submodel DOCUMENTATION
- Title = Operating Manual
- Digital File PDF → /aasx/OperatingManual.PDF
- Document Class ID = 03-02
- Document Class Name = Operation

Submodel

Submodel

Submodel

PLATTFORM INDUSTRIE 4.0

SPECIFICATION

Details of the Asset Administration Shell

In cooperation with:

IDTA Industrial Digital Twin Association

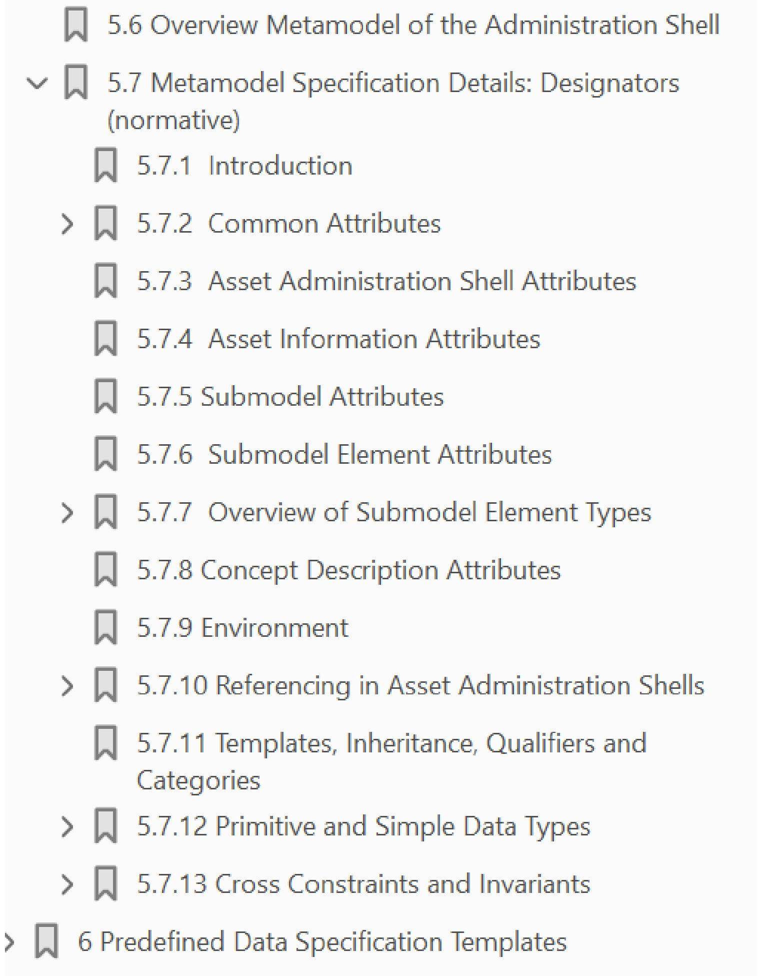
ZVEI:

Part 1 - The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC02)

May 2022

For whom is this tutorial?

- You should have an idea of the benefits of introducting digital twins to your domain
- You should know about the basic concepts of the Asset Administration Shell
- You should have basic knowledge in UML modeling
- You want to learn more about the underlying information model of the Asset Administration Shell
- Security Experts are put of to wait for a separate tutorial



5.6 Overview Metamodel of the Administration Shell

5.7 Metamodel Specification Details: Designators (normative)

5.7.1 Introduction

5.7.2 Common Attributes

5.7.3 Asset Administration Shell Attributes

5.7.4 Asset Information Attributes

5.7.5 Submodel Attributes

5.7.6 Submodel Element Attributes

5.7.7 Overview of Submodel Element Types

5.7.8 Concept Description Attributes

5.7.9 Environment

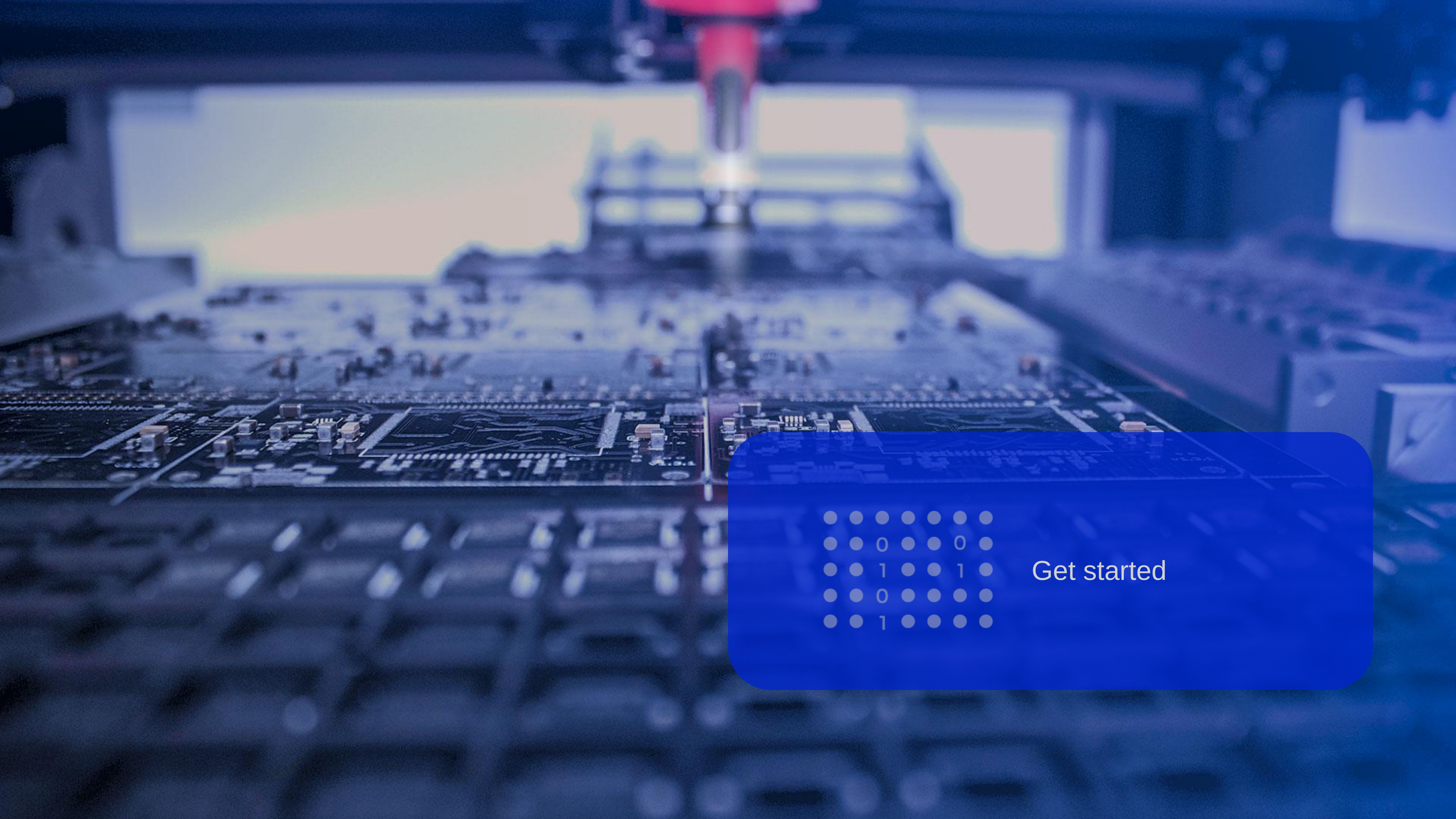
5.7.10 Referencing in Asset Administration Shells

5.7.11 Templates, Inheritance, Qualifiers and Categories

5.7.12 Primitive and Simple Data Types

5.7.13 Cross Constraints and Invariants

6 Predefined Data Specification Templates



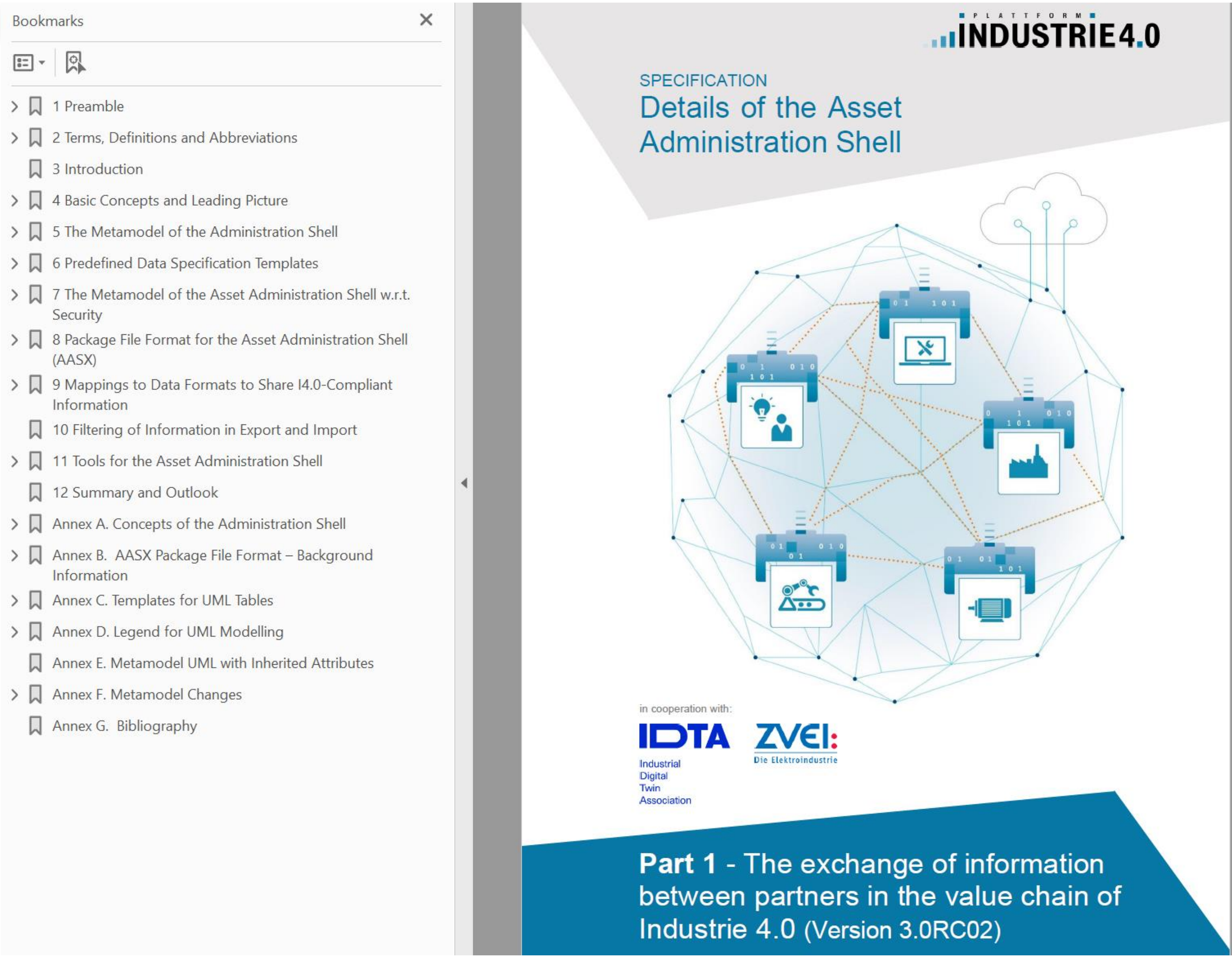
Get started

Download Specification

https://www.plattform-i40.de/IP/Redaktion/EN/Standardartikel/specification-administrationshell.html

https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.html

https://industrialdigitaltwin.org/content-hub/downloads



Bookmarks

- 1 Preamble
- 2 Terms, Definitions and Abbreviations
- 3 Introduction
- 4 Basic Concepts and Leading Picture
- 5 The Metamodel of the Administration Shell
- 6 Predefined Data Specification Templates
- 7 The Metamodel of the Asset Administration Shell w.r.t. Security
- 8 Package File Format for the Asset Administration Shell (AASX)
- 9 Mappings to Data Formats to Share I4.0-Compliant Information
- 10 Filtering of Information in Export and Import
- 11 Tools for the Asset Administration Shell
- 12 Summary and Outlook
- Annex A. Concepts of the Administration Shell
- Annex B. AASX Package File Format – Background Information
- Annex C. Templates for UML Tables
- Annex D. Legend for UML Modelling
- Annex E. Metamodel UML with Inherited Attributes
- Annex F. Metamodel Changes
- Annex G. Bibliography

PLATTFORM INDUSTRIE4.0

SPECIFICATION

Details of the Asset Administration Shell

in cooperation with:

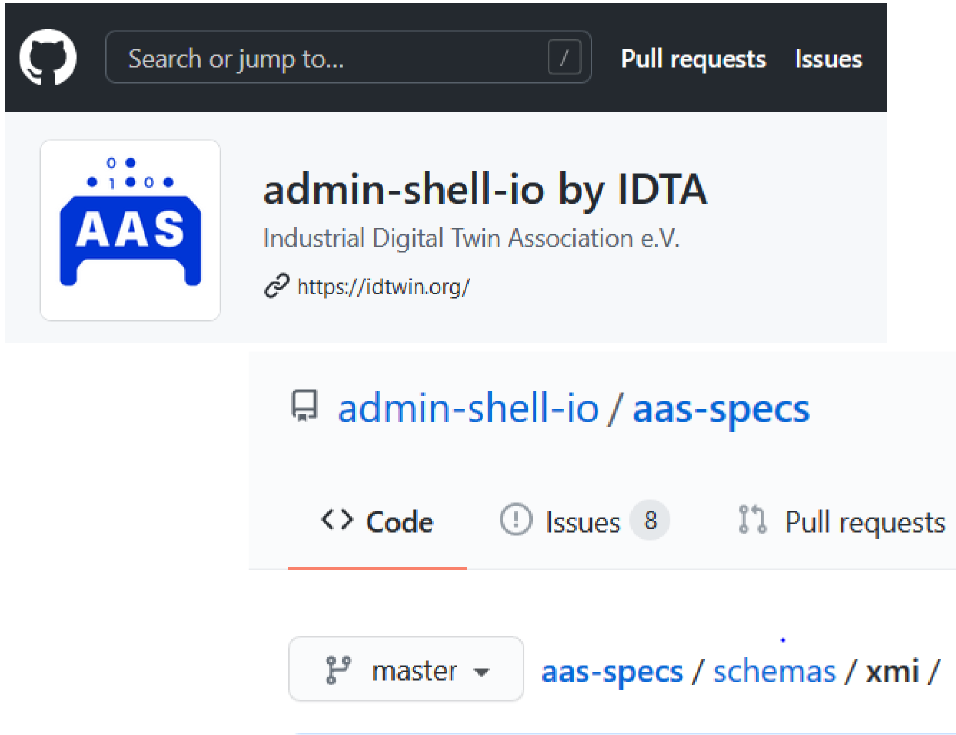
IDTA Industrial Digital Twin Association

ZVEI: Die Elektroindustrie

Part 1 - The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC02)

Import XMI to your UML tooling

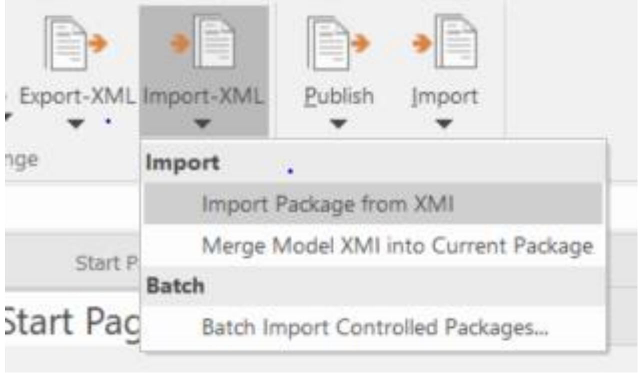
1.



https://github.com/admin-shell-io/aas-specs/tree/master/schemas/xmi

https://github.com/admin-shell-io/aas-specs/releases

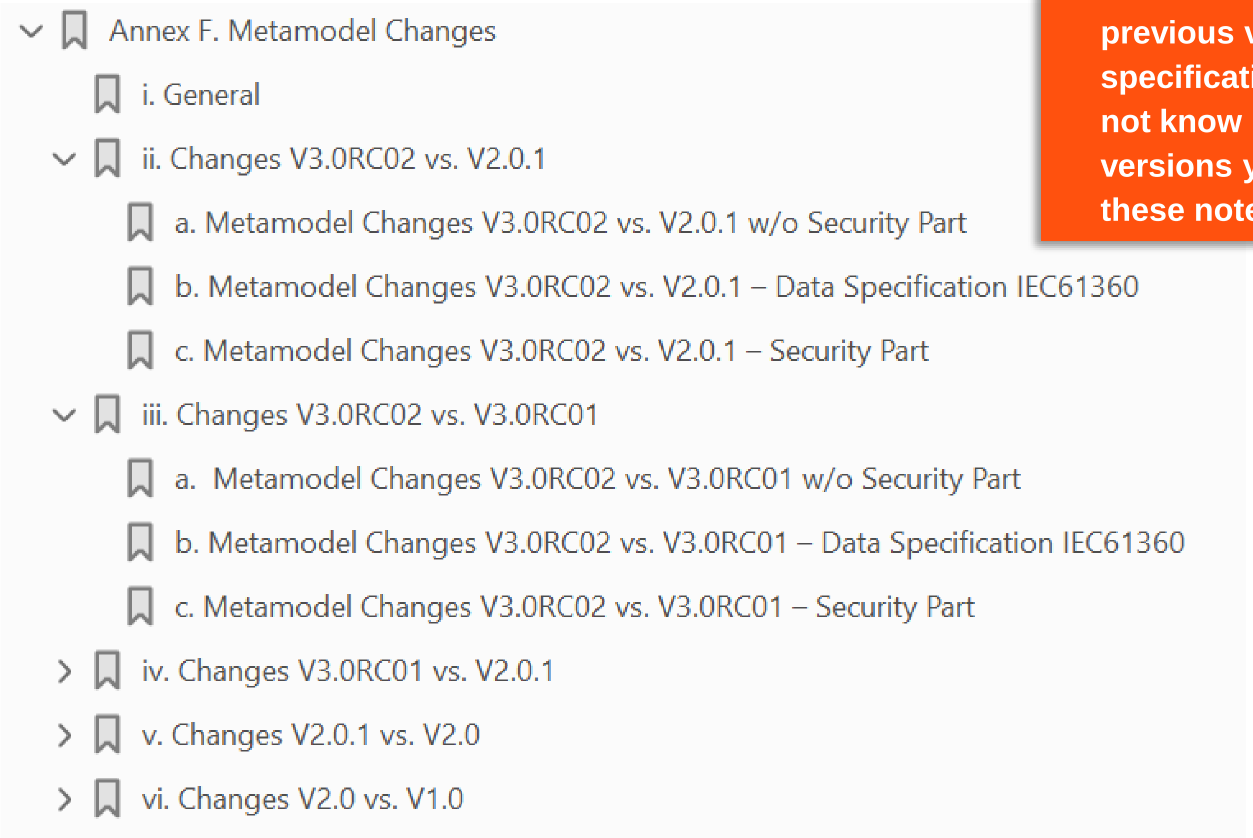
2.



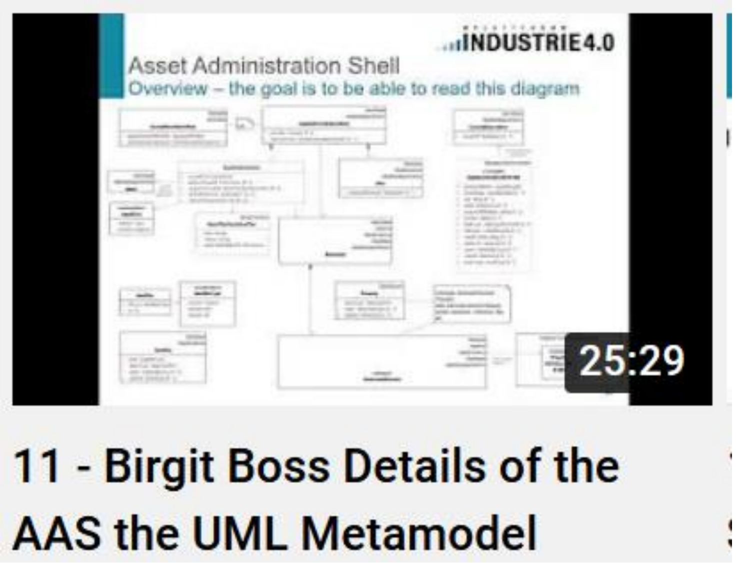
1. Fetch release of AAS you are interested in
2. Import xmi file into UML tool (best with Enterprise/Architect)

Metamodel Changes

Note for Experts: <Notes for tutorial listeners who have knowledge of previous versions of the specification. If you do not know previous versions you can ignore these notes.>



- Annex F. Metamodel Changes
  - i. General
  - ii. Changes V3.0RC02 vs. V2.0.1
    - a. Metamodel Changes V3.0RC02 vs. V2.0.1 w/o Security Part
    - b. Metamodel Changes V3.0RC02 vs. V2.0.1 – Data Specification IEC61360
    - c. Metamodel Changes V3.0RC02 vs. V2.0.1 – Security Part
  - iii. Changes V3.0RC02 vs. V3.0RC01
    - a. Metamodel Changes V3.0RC02 vs. V3.0RC01 w/o Security Part
    - b. Metamodel Changes V3.0RC02 vs. V3.0RC01 – Data Specification IEC61360
    - c. Metamodel Changes V3.0RC02 vs. V3.0RC01 – Security Part
  - iv. Changes V3.0RC01 vs. V2.0.1
  - v. Changes V2.0.1 vs. V2.0
  - vi. Changes V2.0 vs. V1.0


INDUSTRIE4.0
Asset Administration Shell
Overview – the goal is to be able to read this diagram
25:29

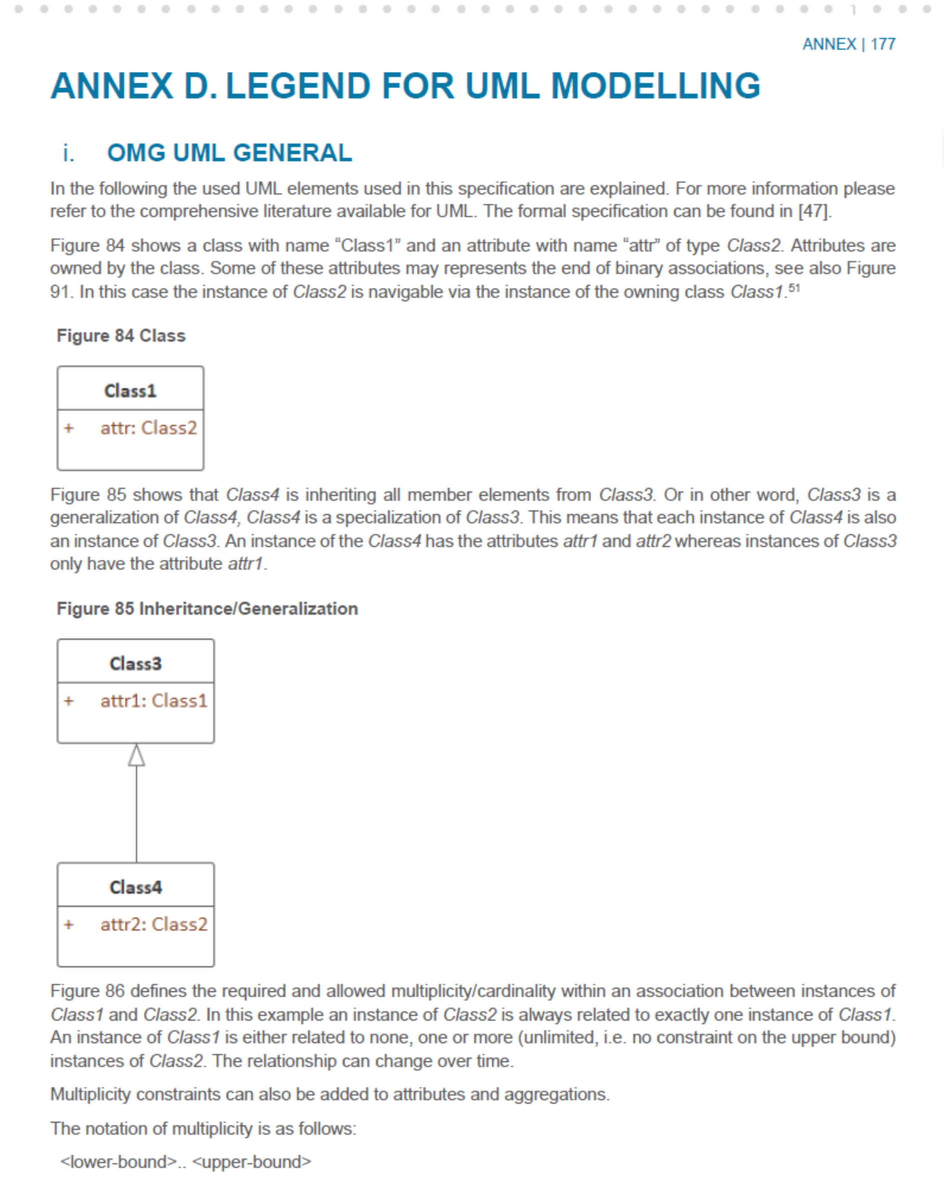
11 - Birgit Boss Details of the AAS the UML Metamodel

V3.0RC01

https://www.youtube.com/channel/UCvYk-bRkF9-x0HdpJfEcV7g

UML Legend

1. Get (re-)familiar with general UML modeling rules
2. Get familiar with specific graphical representation of UML in the specification (partly tool specific)



ANNEX | 177

ANNEX D. LEGEND FOR UML MODELLING

i. OMG UML GENERAL

In the following the used UML elements used in this specification are explained. For more information please refer to the comprehensive literature available for UML. The formal specification can be found in [47].

Figure 84 shows a class with name "Class1" and an attribute with name "attr" of type *Class2*. Attributes are owned by the class. Some of these attributes may represents the end of binary associations, see also Figure 91. In this case the instance of *Class2* is navigable via the instance of the owning class *Class1*.[51]

Figure 84 Class

Figure 85 shows that *Class4* is inheriting all member elements from *Class3*. Or in other word, *Class3* is a generalization of *Class4, Class4* is a specialization of *Class3*. This means that each instance of *Class4* is also an instance of *Class3*. An instance of the *Class4* has the attributes *attr1* and *attr2* whereas instances of *Class3* only have the attribute *attr1*.

Figure 85 Inheritance/Generalization

Figure 86 defines the required and allowed multiplicity/cardinality within an association between instances of *Class1* and *Class2*. In this example an instance of *Class2* is always related to exactly one instance of *Class1*. An instance of *Class1* is either related to none, one or more (unlimited, i.e. no constraint on the upper bound) instances of *Class2*. The relationship can change over time.

Multiplicity constraints can also be added to attributes and aggregations.

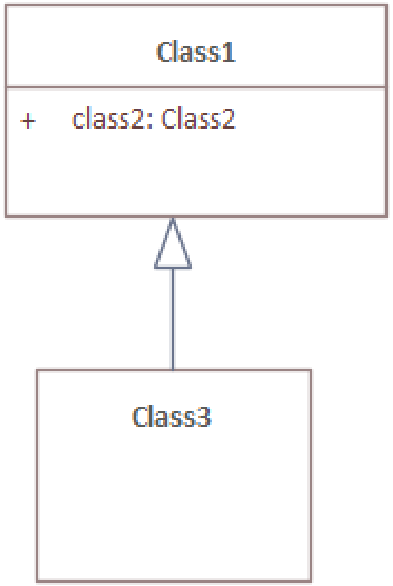
The notation of multiplicity is as follows:

<lower-bound>.. <upper-bound>

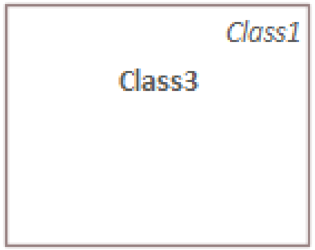
UML Generalization/Inheritance

Hint: Graphical representation tool specific

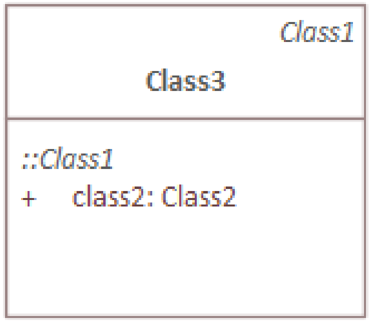
A)



B)

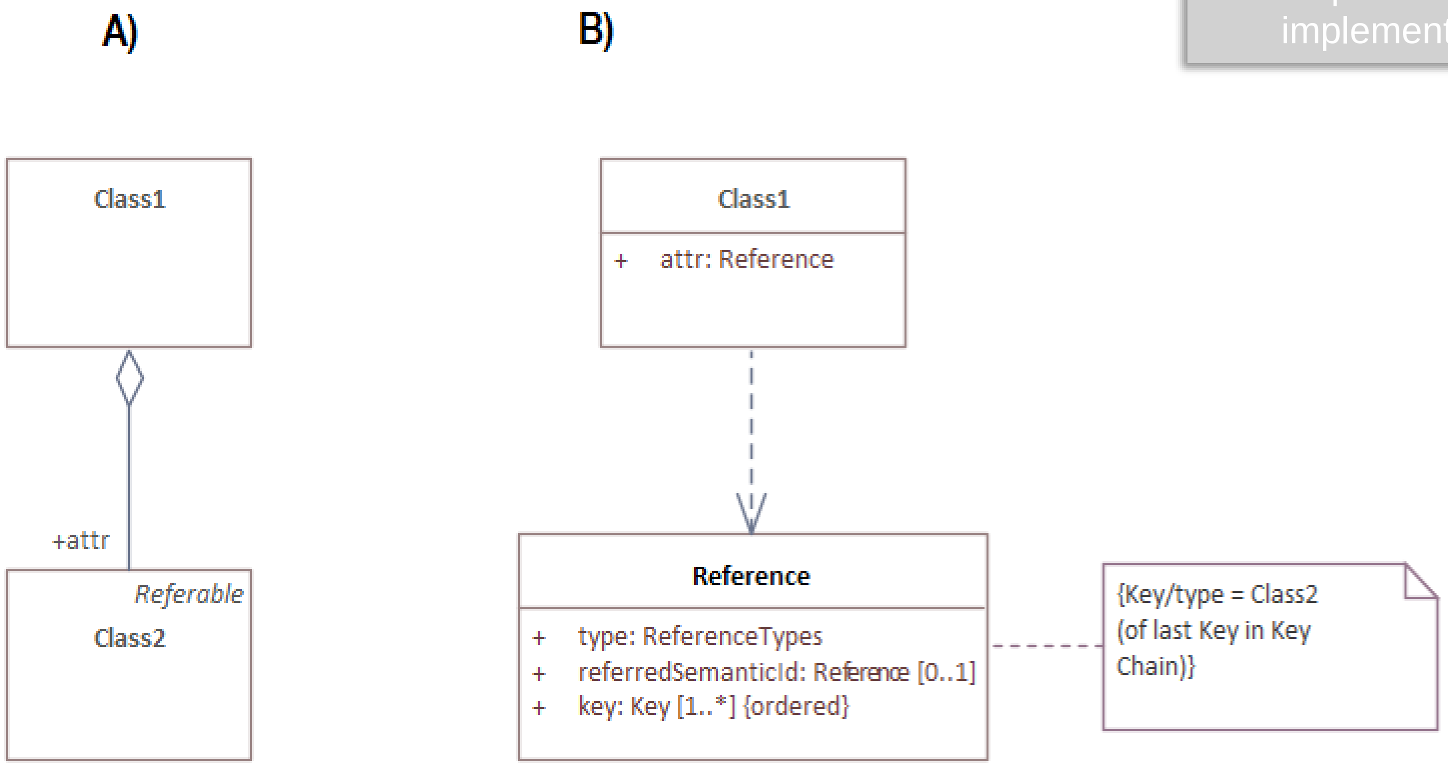


C)



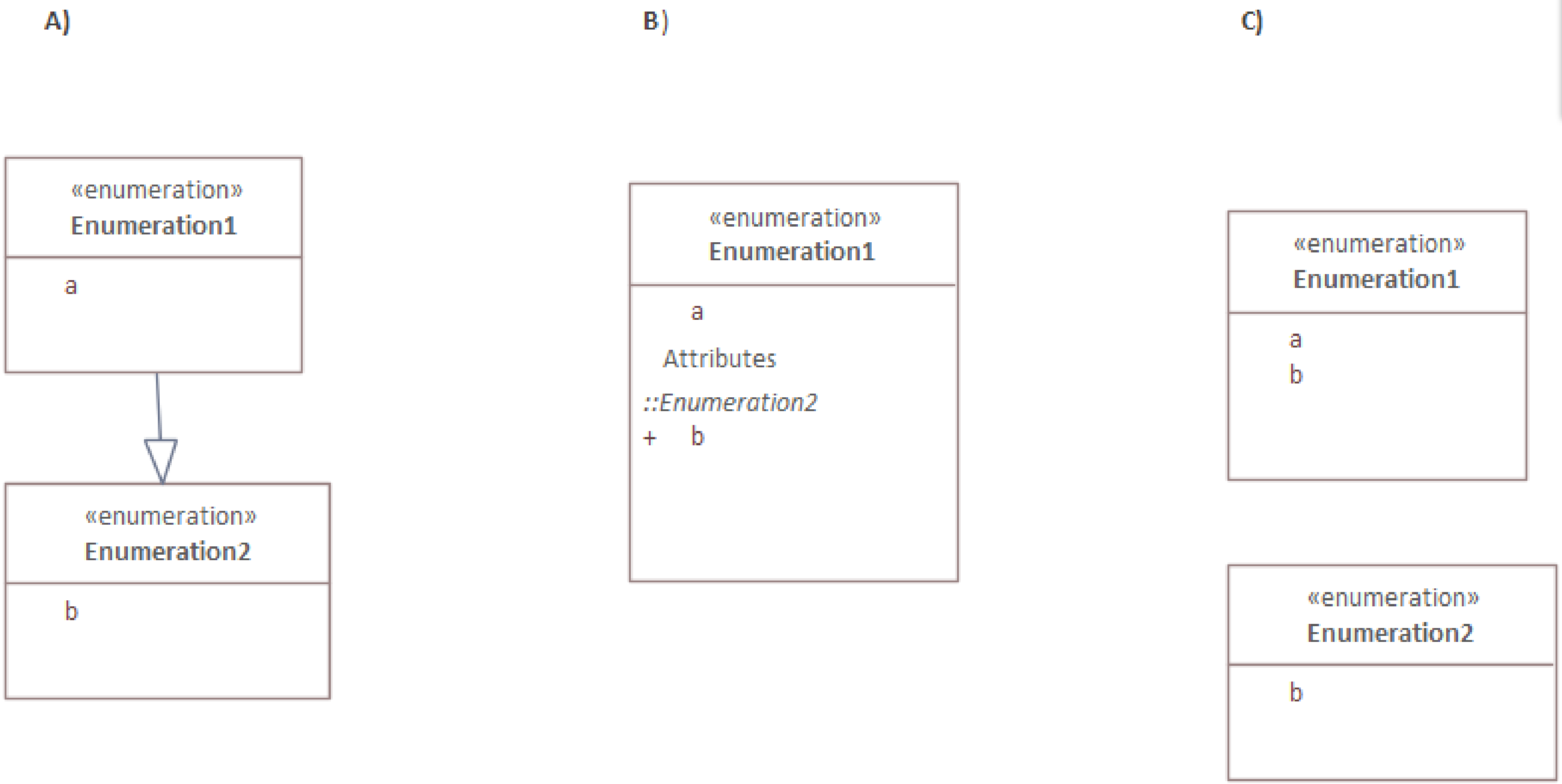
UML Shared Aggregation

Note: Referencing of Referables is an important concept to understand when implementing the AAS



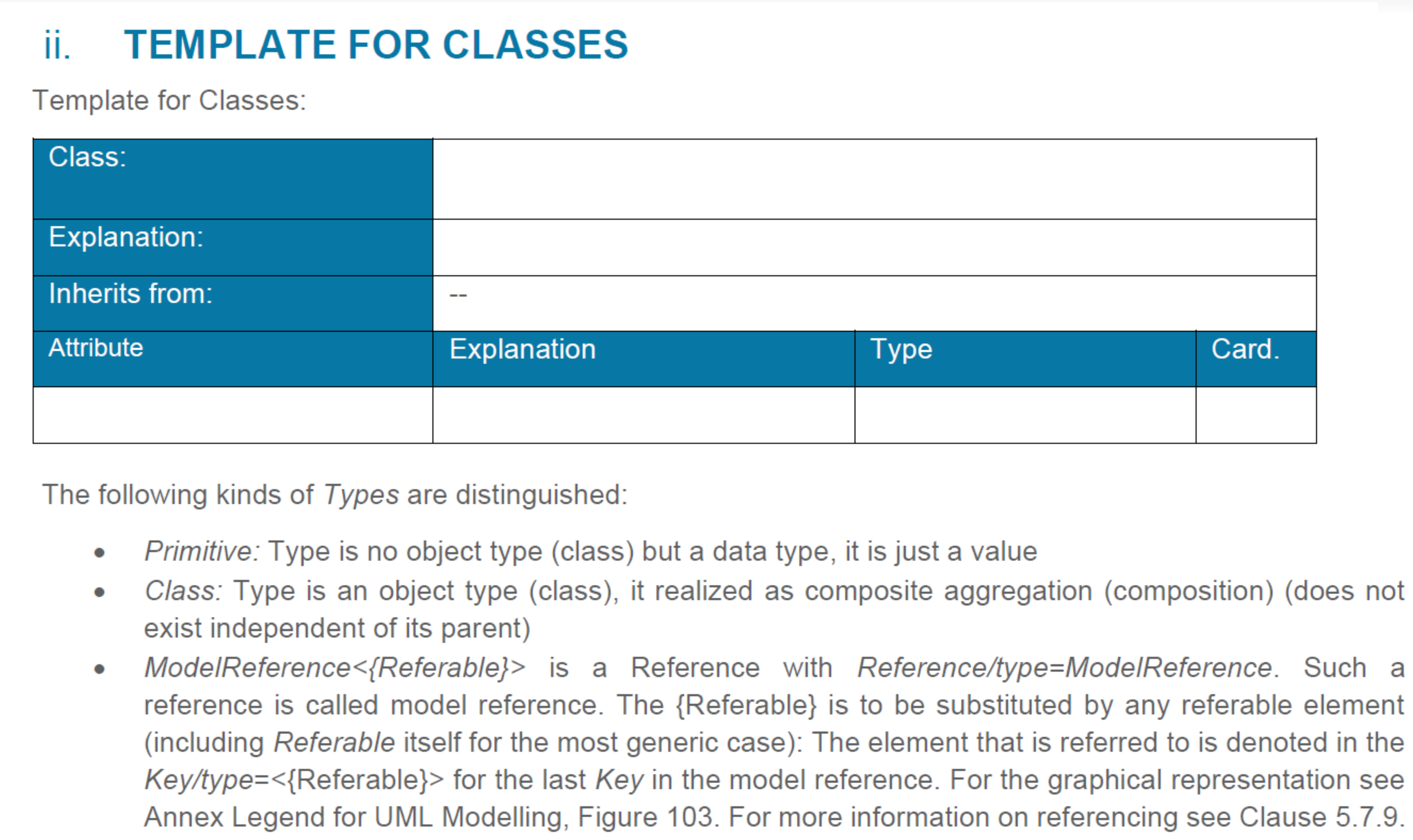
Note for experts: In previous versions a notation of class attributes with reference (*) was used additionally to the notation with the association with the diamond.

Enumerations



Note 1: A) and B) are just introduced for easier maintenance of enumerations

Templates



ii. TEMPLATE FOR CLASSES

Template for Classes:

Class:			
Explanation:			
Inherits from:	--		
Attribute	Explanation	Type	Card.

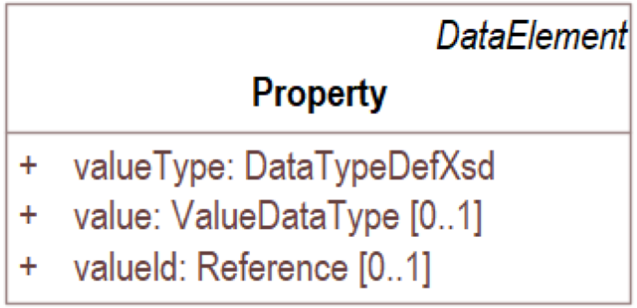
The following kinds of *Types* are distinguished:

- *Primitive:* Type is no object type (class) but a data type, it is just a value
- *Class:* Type is an object type (class), it realized as composite aggregation (composition) (does not exist independent of its parent)
- *ModelReference<{Referable}>* is a Reference with *Reference/type=ModelReference*. Such a reference is called model reference. The {Referable} is to be substituted by any referable element (including *Referable* itself for the most generic case): The element that is referred to is denoted in the *Key/type*=<{Referable}> for the last *Key* in the model reference. For the graphical representation see Annex Legend for UML Modelling, Figure 103. For more information on referencing see Clause 5.7.9.

Note for experts: ModelReference<SubmodelElement> is equal to former notation SubmodelElement*

Note for experts: no kind column any longer, instead different notation for Type

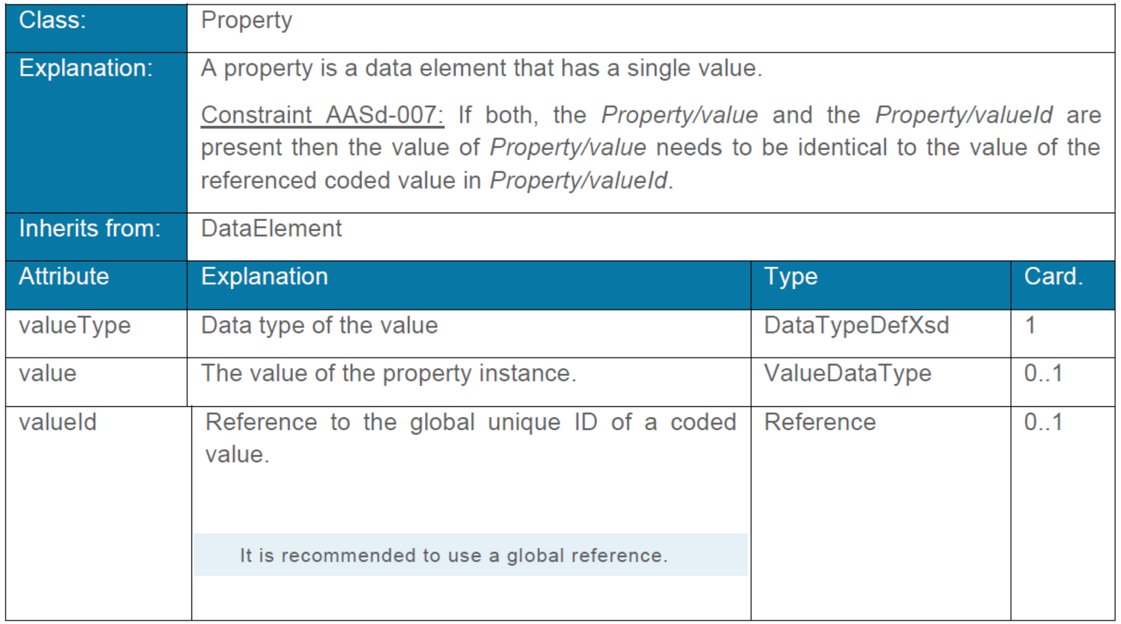
Example for Class Specification



DataElement

Property

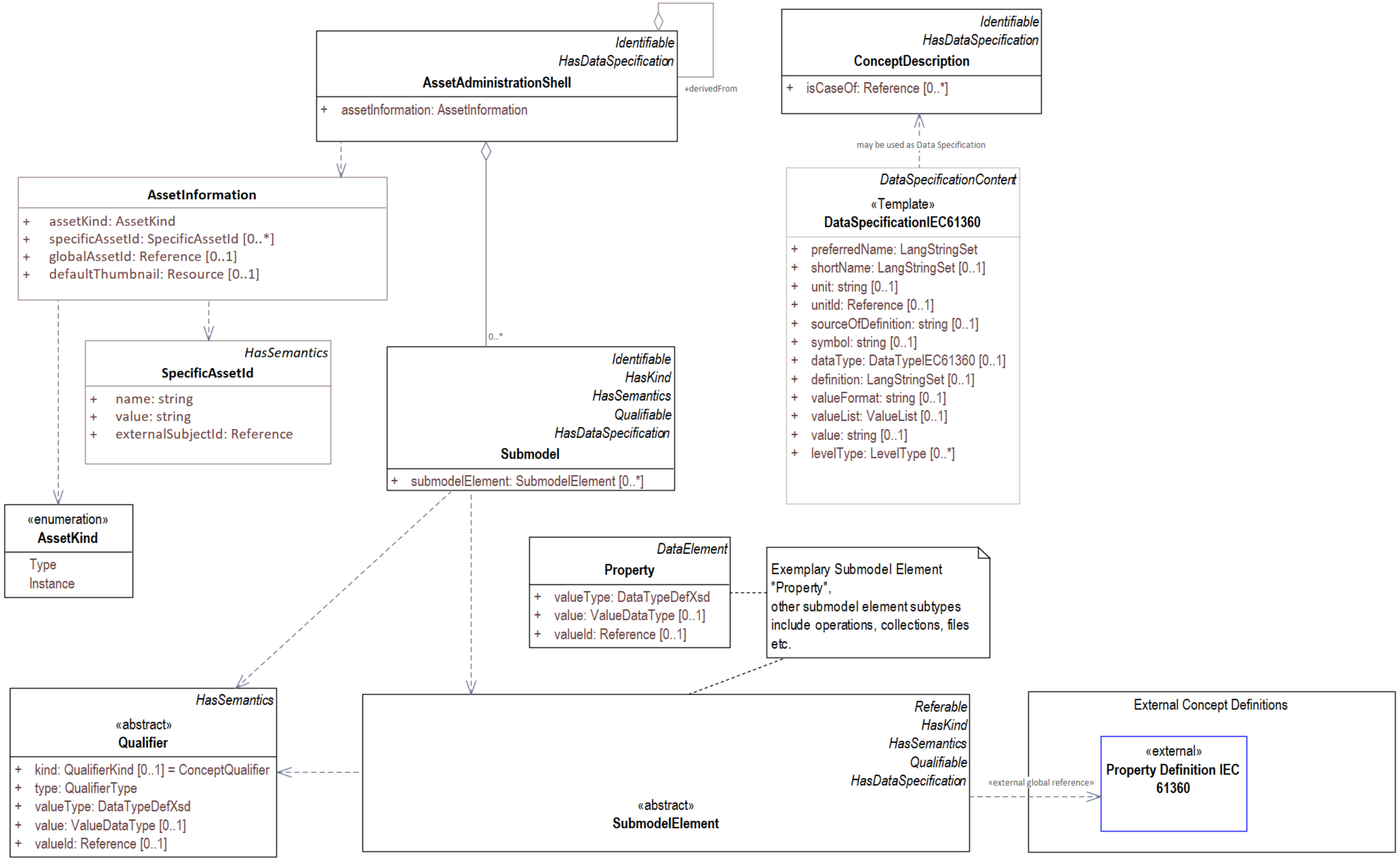
+ valueType: DataTypeDefXsd
+ value: ValueDataType [0..1]
+ valueId: Reference [0..1]



Class:	Property		
Explanation:	A property is a data element that has a single value. Constraint AASd-007: If both, the *Property/value* and the *Property/valueId* are present then the value of *Property/value* needs to be identical to the value of the referenced coded value in *Property/valueId*.		
Inherits from:	DataElement		
Attribute	Explanation	Type	Card.
valueType	Data type of the value	DataTypeDefXsd	1
value	The value of the property instance.	ValueDataType	0..1
valueId	Reference to the global unique ID of a coded value. It is recommended to use a global reference.	Reference	0..1

Get warm

Overview – the goal is to be able to read this diagram



Identifiable
HasDataSpecification
AssetAdministrationShell
+ assetInformation: AssetInformation

+derivedFrom

Identifiable
HasDataSpecification
ConceptDescription
+ isCaseOf: Reference [0..*]

may be used as Data Specification

DataSpecificationContent
«Template»
DataSpecificationIEC61360
+ preferredName: LangStringSet
+ shortName: LangStringSet [0..1]
+ unit: string [0..1]
+ unitId: Reference [0..1]
+ sourceOfDefinition: string [0..1]
+ symbol: string [0..1]
+ dataType: DataTypeIEC61360 [0..1]
+ definition: LangStringSet [0..1]
+ valueFormat: string [0..1]
+ valueList: ValueList [0..1]
+ value: string [0..1]
+ levelType: LevelType [0..*]

AssetInformation
+ assetKind: AssetKind
+ specificAssetId: SpecificAssetId [0..*]
+ globalAssetId: Reference [0..1]
+ defaultThumbnail: Resource [0..1]

HasSemantics
SpecificAssetId
+ name: string
+ value: string
+ externalSubjectId: Reference

0..*

Identifiable
HasKind
HasSemantics
Qualifiable
HasDataSpecification
Submodel
+ submodelElement: SubmodelElement [0..*]

«enumeration»
AssetKind
Type
Instance

DataElement
Property
+ valueType: DataTypeDefXsd
+ value: ValueDataType [0..1]
+ valueId: Reference [0..1]

Exemplary Submodel Element "Property", other submodel element subtypes include operations, collections, files etc.

HasSemantics
«abstract»
Qualifier
+ kind: QualifierKind [0..1] = ConceptQualifier
+ type: QualifierType
+ valueType: DataTypeDefXsd
+ value: ValueDataType [0..1]
+ valueId: Reference [0..1]

Referable
HasKind
HasSemantics
Qualifiable
HasDataSpecification
«abstract»
SubmodelElement

«external global reference»

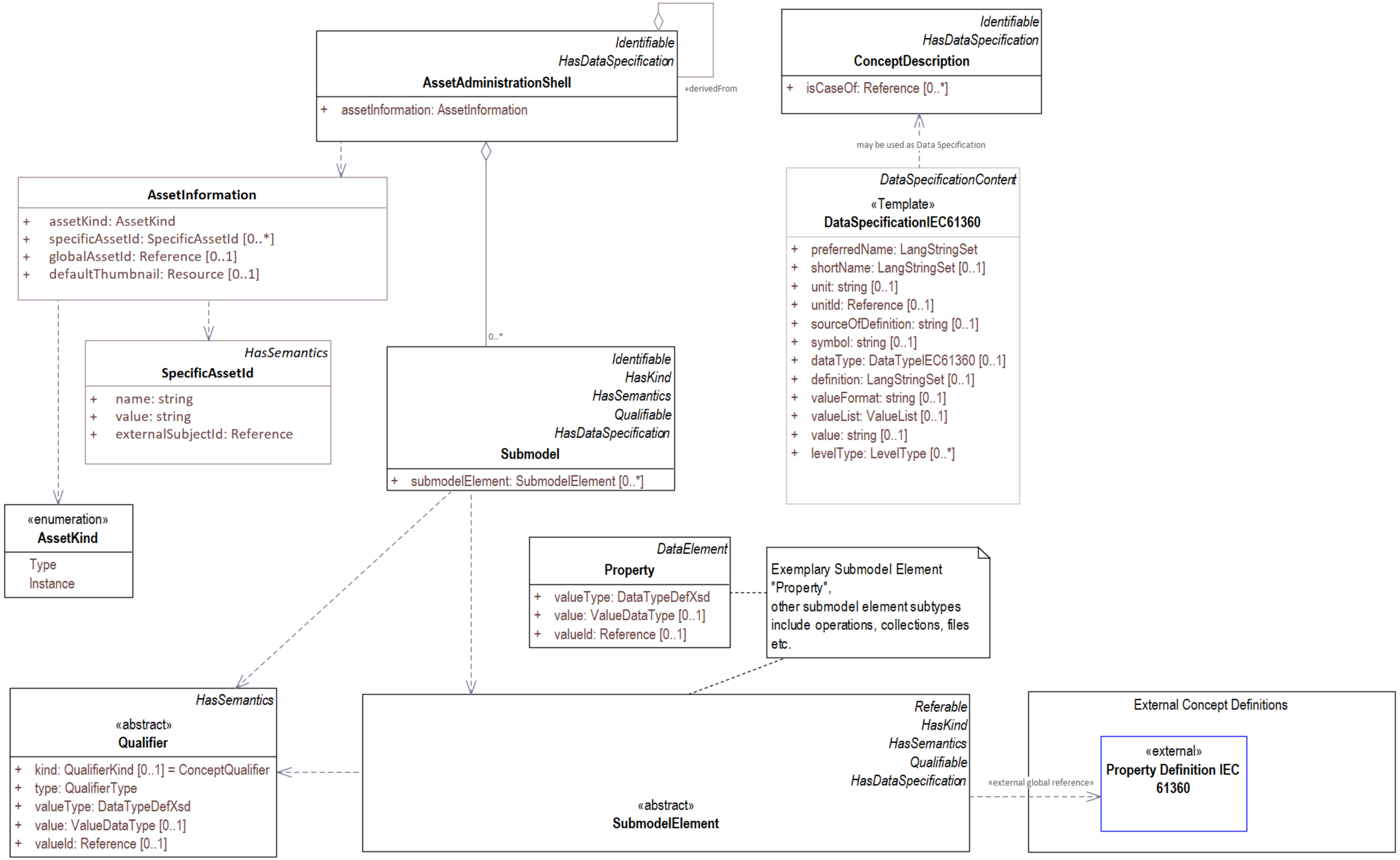
External Concept Definitions
«external»
Property Definition IEC 61360

Overview – the goal is to be able to read this diagram

Note for Experts: no views supported any longer

Note for Experts: no assets supported any longer, assetInformation introduced instead

Note for Experts: Security and Asset Administration Shell now loosely coupled only



Identifiable
HasDataSpecification
AssetAdministrationShell
+ assetInformation: AssetInformation

+derivedFrom

Identifiable
HasDataSpecification
ConceptDescription
+ isCaseOf: Reference [0..*]

may be used as Data Specification

DataSpecificationContent
«Template»
DataSpecificationIEC61360
+ preferredName: LangStringSet
+ shortName: LangStringSet [0..1]
+ unit: string [0..1]
+ unitId: Reference [0..1]
+ sourceOfDefinition: string [0..1]
+ symbol: string [0..1]
+ dataType: DataTypeIEC61360 [0..1]
+ definition: LangStringSet [0..1]
+ valueFormat: string [0..1]
+ valueList: ValueList [0..1]
+ value: string [0..1]
+ levelType: LevelType [0..*]

AssetInformation
+ assetKind: AssetKind
+ specificAssetId: SpecificAssetId [0..*]
+ globalAssetId: Reference [0..1]
+ defaultThumbnail: Resource [0..1]

HasSemantics
SpecificAssetId
+ name: string
+ value: string
+ externalSubjectId: Reference

0..*

Identifiable
HasKind
HasSemantics
Qualifiable
HasDataSpecification
Submodel
+ submodelElement: SubmodelElement [0..*]

«enumeration»
AssetKind
Type
Instance

DataElement
Property
+ valueType: DataTypeDefXsd
+ value: ValueDataType [0..1]
+ valueId: Reference [0..1]

Exemplary Submodel Element "Property", other submodel element subtypes include operations, collections, files etc.

HasSemantics
«abstract»
Qualifier
+ kind: QualifierKind [0..1] = ConceptQualifier
+ type: QualifierType
+ valueType: DataTypeDefXsd
+ value: ValueDataType [0..1]
+ valueId: Reference [0..1]

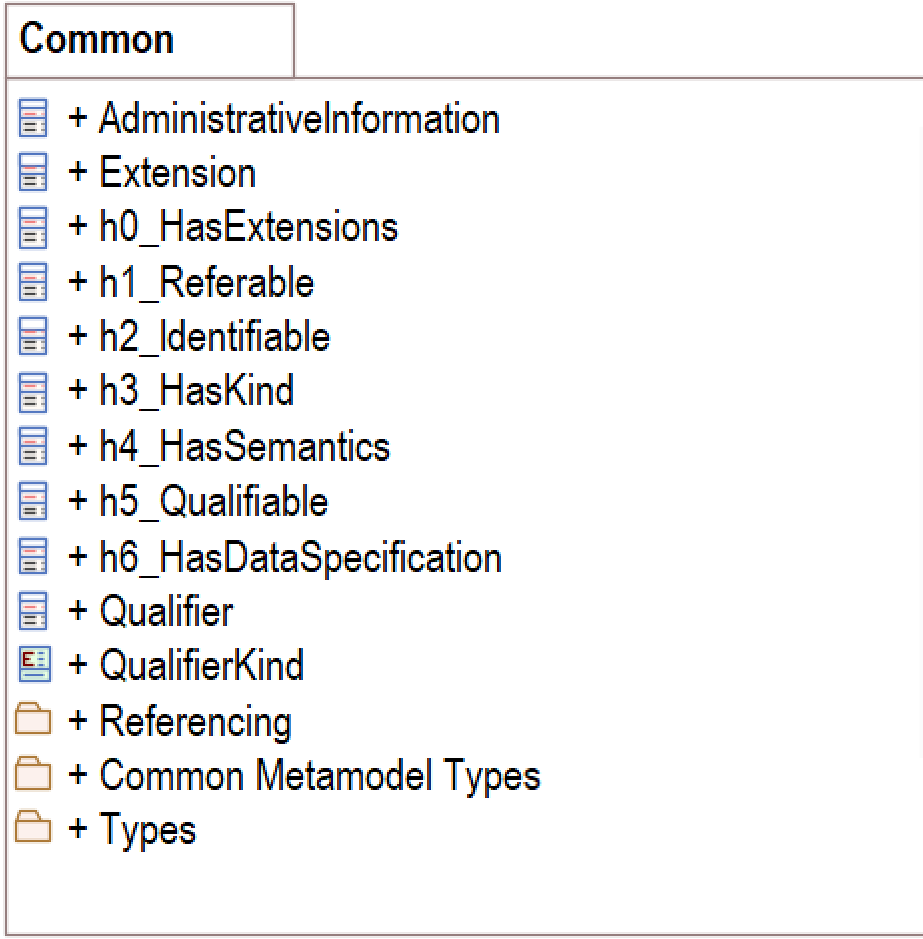
Referable
HasKind
HasSemantics
Qualifiable
HasDataSpecification
«abstract»
SubmodelElement

«external global reference»

External Concept Definitions

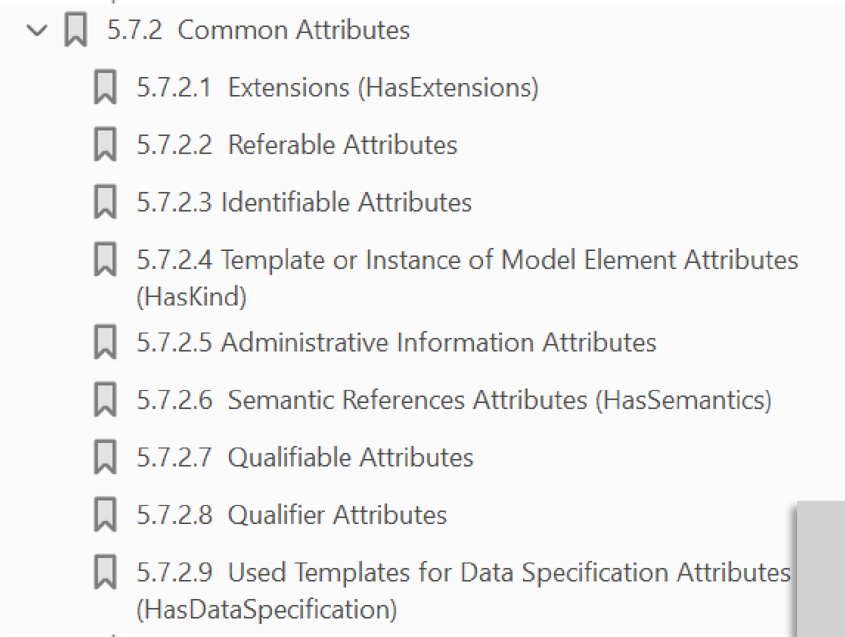
«external»
Property Definition IEC 61360

Common



Common

- + AdministrativeInformation
- + Extension
- + h0_HasExtensions
- + h1_Referable
- + h2_Identifiable
- + h3_HasKind
- + h4_HasSemantics
- + h5_Qualifiable
- + h6_HasDataSpecification
- + Qualifier
- + QualifierKind
- + Referencing
- + Common Metamodel Types
- + Types

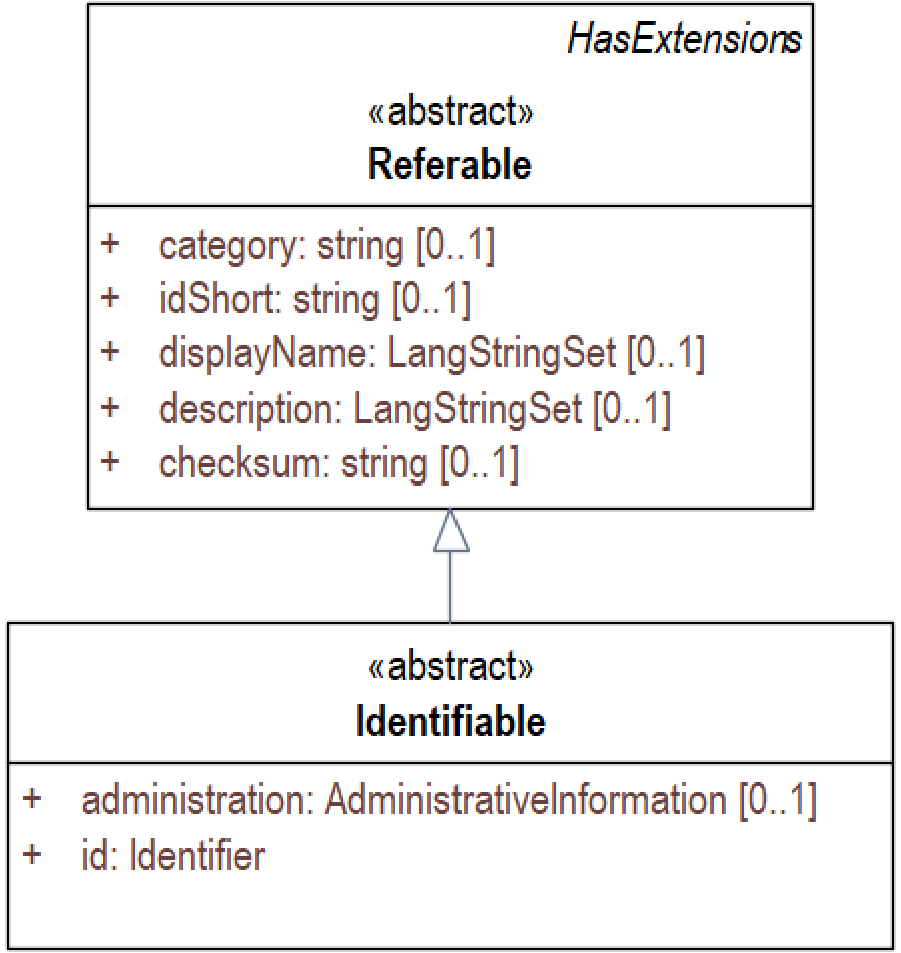


5.7.2 Common Attributes

- 5.7.2.1 Extensions (HasExtensions)
- 5.7.2.2 Referable Attributes
- 5.7.2.3 Identifiable Attributes
- 5.7.2.4 Template or Instance of Model Element Attributes (HasKind)
- 5.7.2.5 Administrative Information Attributes
- 5.7.2.6 Semantic References Attributes (HasSemantics)
- 5.7.2.7 Qualifiable Attributes
- 5.7.2.8 Qualifier Attributes
- 5.7.2.9 Used Templates for Data Specification Attributes (HasDataSpecification)

Note: h0_ h1_ are just added for sorting. In diagrams alias are used without this prefix. Only in package overview and inheritance alias are not used in the tooling.

Common – Identifiables and Referables



HasExtensions

«abstract»
Referable

+ category: string [0..1]
+ idShort: string [0..1]
+ displayName: LangStringSet [0..1]
+ description: LangStringSet [0..1]
+ checksum: string [0..1]

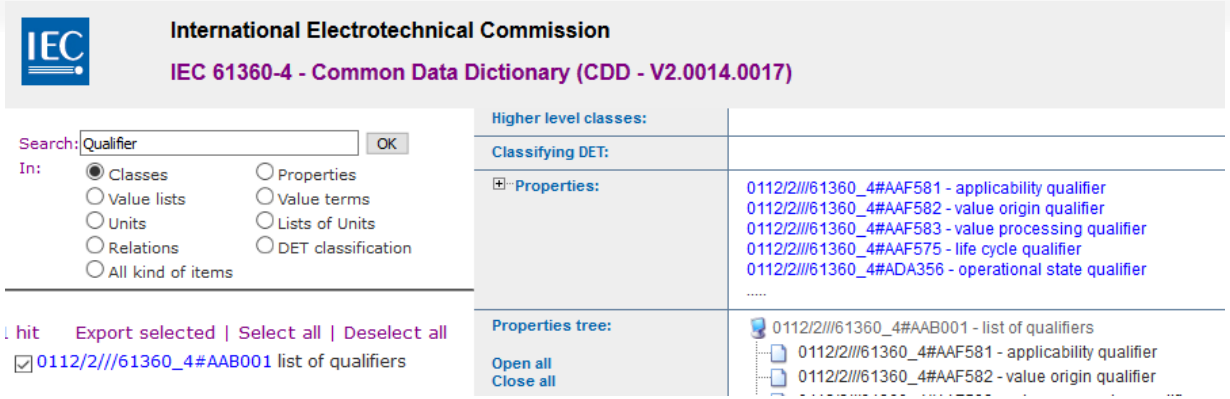
«abstract»
Identifiable

+ administration: AdministrativeInformation [0..1]
+ id: Identifier

Note for Experts: Identifier in previous versions of the specification had two attributes: the ID itself and the ID type (IRI, IRDI, Custom). The ID type was removed from the model.

Note for Experts: idShort now optional but still required for non-identiable referables. DisplayName introduced and checksum.

Common - Qualifiables



International Electrotechnical Commission

IEC 61360-4 - Common Data Dictionary (CDD - V2.0014.0017)

Search: Qualifier OK

In:
- ● Classes
- ○ Value lists
- ○ Units
- ○ Relations
- ○ All kind of items
- ○ Properties
- ○ Value terms
- ○ Lists of Units
- ○ DET classification

hit Export selected | Select all | Deselect all

☑ 0112/2///61360_4#AAB001 list of qualifiers

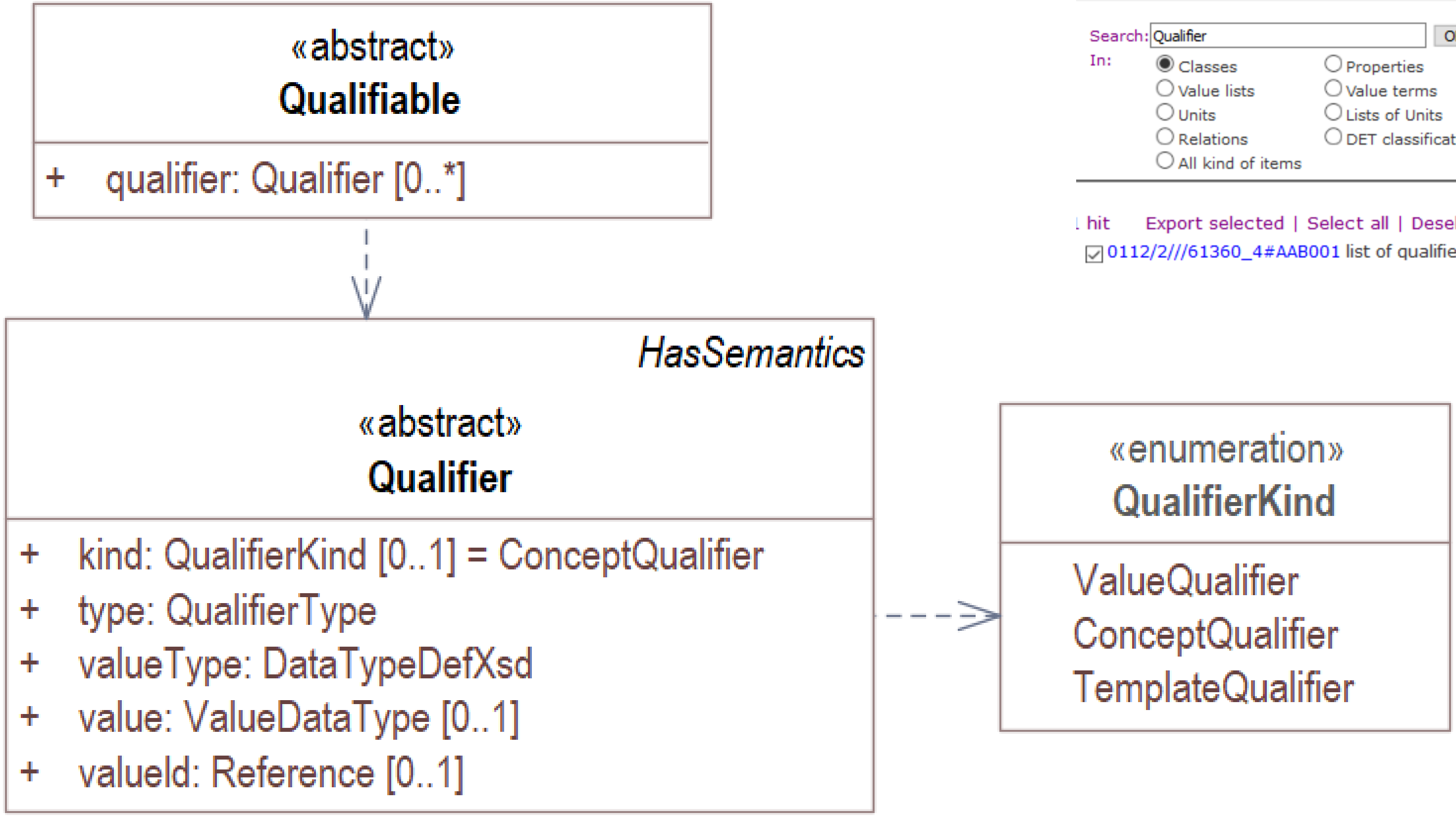
Higher level classes:

Classifying DET:

Properties:
0112/2///61360_4#AAF581 - applicability qualifier
0112/2///61360_4#AAF582 - value origin qualifier
0112/2///61360_4#AAF583 - value processing qualifier
0112/2///61360_4#AAF575 - life cycle qualifier
0112/2///61360_4#ADA356 - operational state qualifier
.....

Properties tree:
Open all
Close all

0112/2///61360_4#AAB001 - list of qualifiers
- 0112/2///61360_4#AAF581 - applicability qualifier
- 0112/2///61360_4#AAF582 - value origin qualifier



«abstract»
Qualifiable
+ qualifier: Qualifier [0..*]

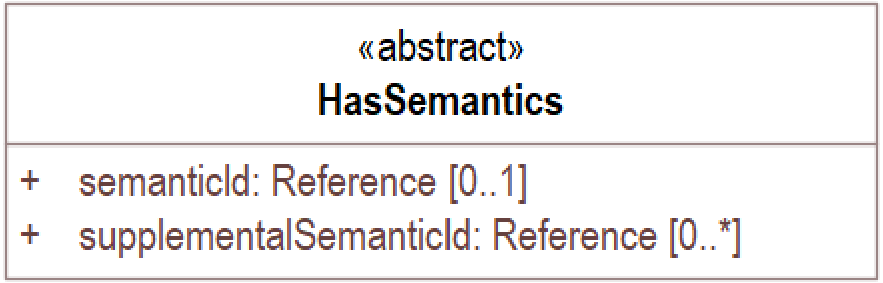
HasSemantics
«abstract»
Qualifier
+ kind: QualifierKind [0..1] = ConceptQualifier
+ type: QualifierType
+ valueType: DataTypeDefXsd
+ value: ValueDataType [0..1]
+ valueId: Reference [0..1]

«enumeration»
QualifierKind
ValueQualifier
ConceptQualifier
TemplateQualifier

Note for Experts: No Formulas (or other Constraints) supported for Qualifiables any longer,

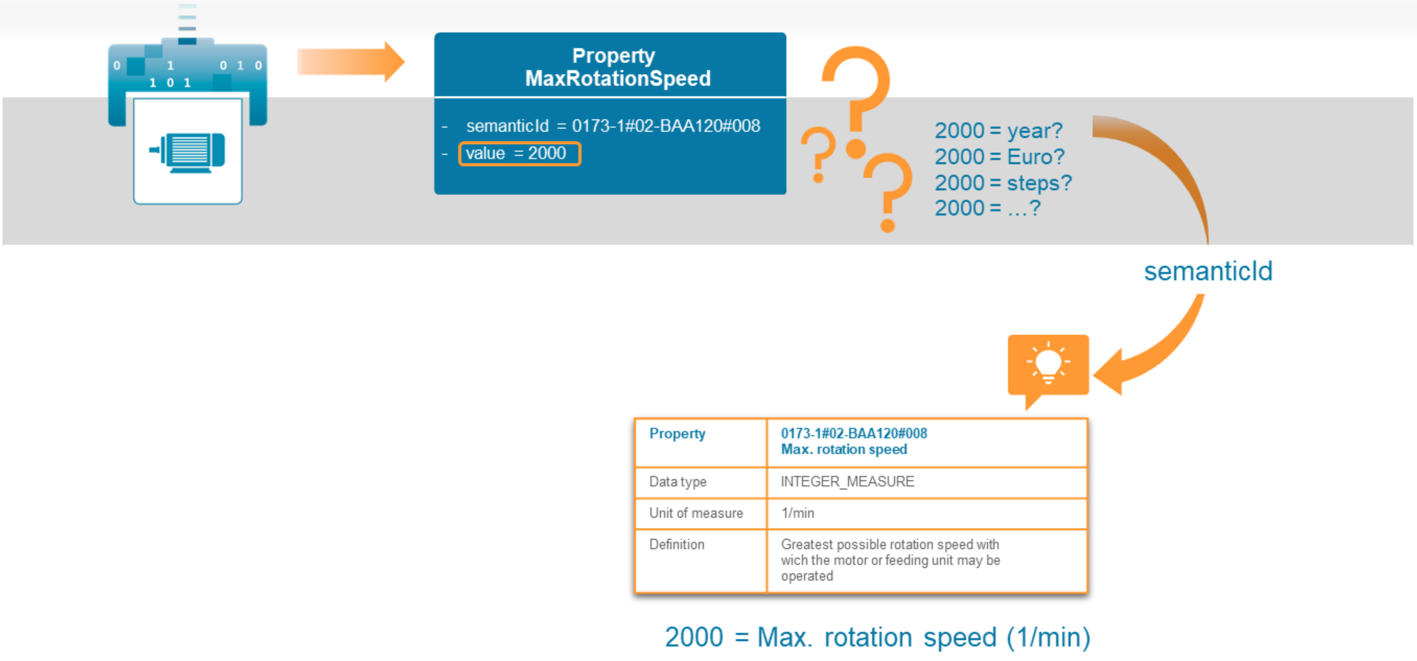
Note for Experts: Qualifier kind introduced

Common - HasSemantics



«abstract»
HasSemantics

+ semanticId: Reference [0..1]
+ supplementalSemanticId: Reference [0..*]



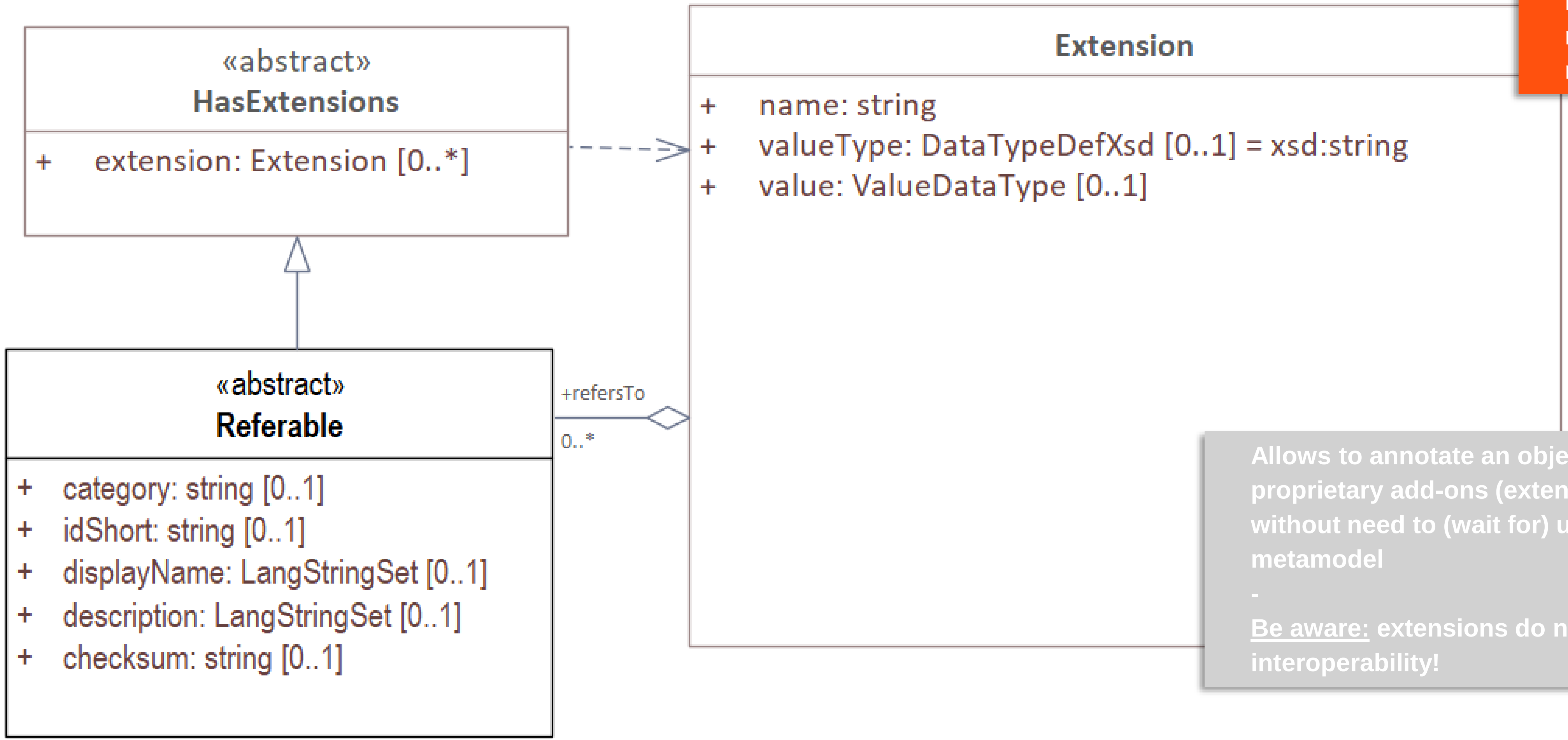
Property	**0173-1#02-BAA120#008** **Max. rotation speed**
Data type	INTEGER_MEASURE
Unit of measure	1/min
Definition	Greatest possible rotation speed with wich the motor or feeding unit may be operated

2000 = Max. rotation speed (1/min)

The semanticId is a reference, either referring to a semantic definition in an external Data Dictionary (like for example ECLASS) or to an AAS Concept Description.

Note for Experts: Besides the semanticId supplemental semantic IDs are now possible to be added.

Common - HasExtensions



«abstract»
HasExtensions

+ extension: Extension [0..*]

Extension

+ name: string
+ valueType: DataTypeDefXsd [0..1] = xsd:string
+ value: ValueDataType [0..1]

«abstract»
Referable

+ category: string [0..1]
+ idShort: string [0..1]
+ displayName: LangStringSet [0..1]
+ description: LangStringSet [0..1]
+ checksum: string [0..1]

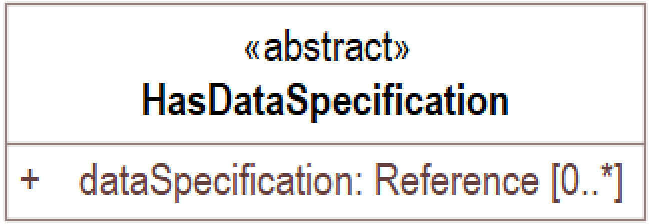
Note for Experts: refersTo now only supports references to Referables, not to external sources.

Allows to annotate an object with proprietary add-ons (extensions) without need to (wait for) update the metamodel

-

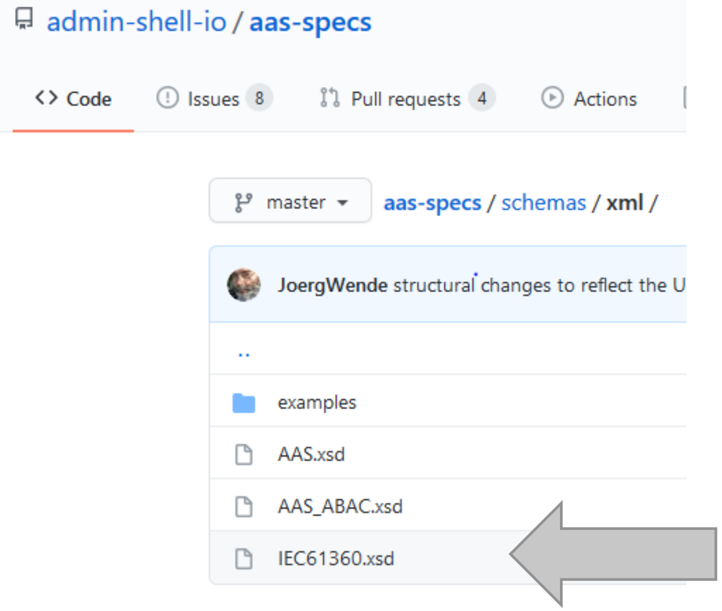
<u>Be aware:</u> extensions do not support interoperability!

Common - HasDataSpecification



«abstract»
HasDataSpecification

+ dataSpecification: Reference [0..*]



admin-shell-io / **aas-specs**

aas-specs / schemas / **xml** /

JoergWende structural changes to reflect the U

..

examples

AAS.xsd

AAS_ABAC.xsd

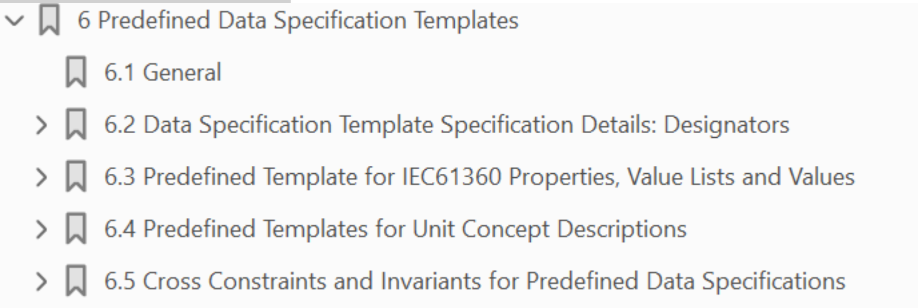
IEC61360.xsd

Predefined data specification IEC61360 in XML Serialization

Allows to define standardized templates for data specification

Data specification reference shall be globally unique and identifies which data specifications are used for an object

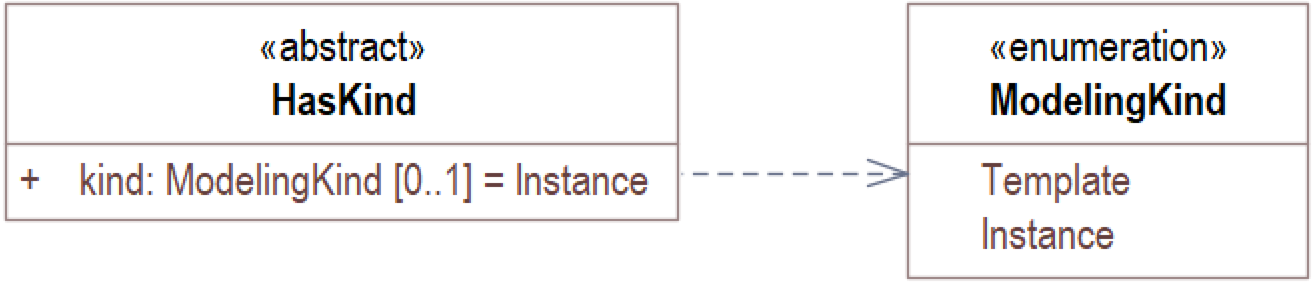
Attributes defined in template are added to the object



6 Predefined Data Specification Templates

- 6.1 General
- 6.2 Data Specification Template Specification Details: Designators
- 6.3 Predefined Template for IEC61360 Properties, Value Lists and Values
- 6.4 Predefined Templates for Unit Concept Descriptions
- 6.5 Cross Constraints and Invariants for Predefined Data Specifications

Common - HasKind



«abstract»
HasKind

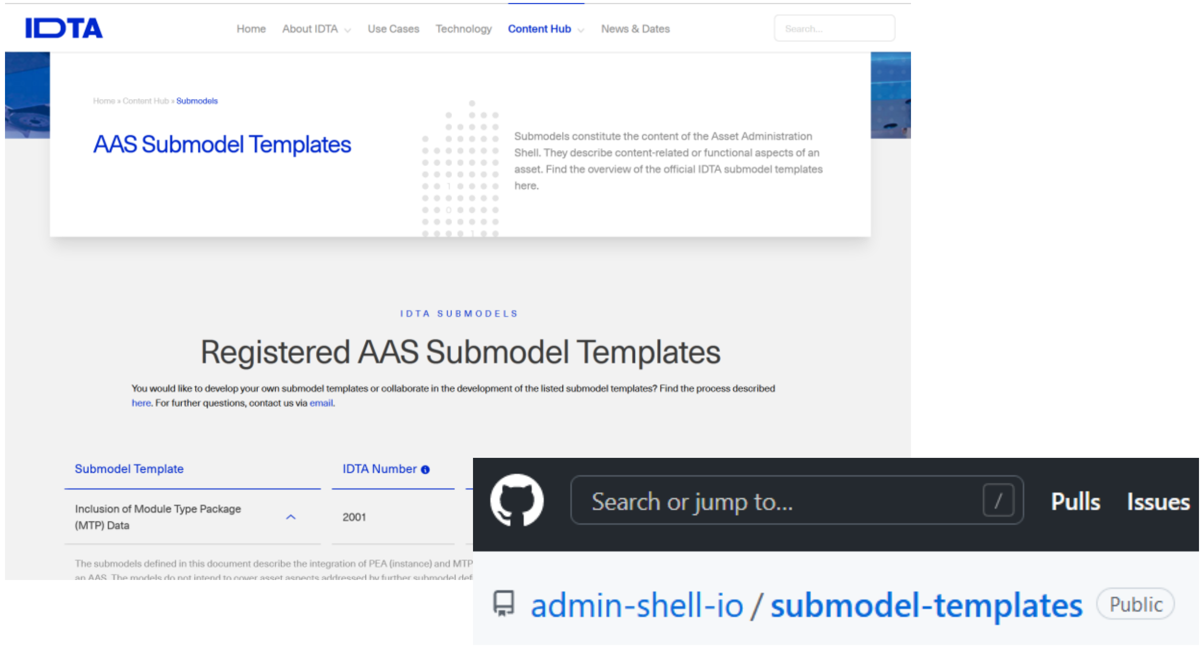
+ kind: ModelingKind [0..1] = Instance

«enumeration»
ModelingKind

Template
Instance

ModelingKind = Instance provides values for all data elements within a submodel

ModelingKind = Template is used to specify Submodel Templates, e.g. for the Digital Nameplate or Technical Data



AAS Submodel Templates

Submodels constitute the content of the Asset Administration Shell. They describe content-related or functional aspects of an asset. Find the overview of the official IDTA submodel templates here.

IDTA SUBMODELS

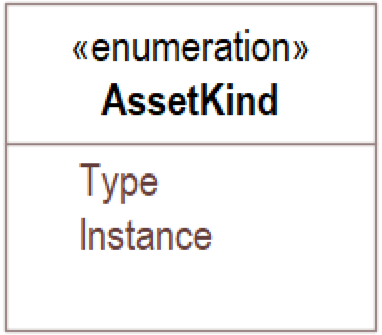
Registered AAS Submodel Templates

You would like to develop your own submodel templates or collaborate in the development of the listed submodel templates? Find the process described here. For further questions, contact us via email.

Submodel Template	IDTA Number
Inclusion of Module Type Package (MTP) Data	2001

The submodels defined in this document describe the integration of PEA (instance) and MTP
an AAS. The models do not intend to cover asset aspects addressed by further submodel def

admin-shell-io / submodel-templates Public



«enumeration»
AssetKind

Type
Instance

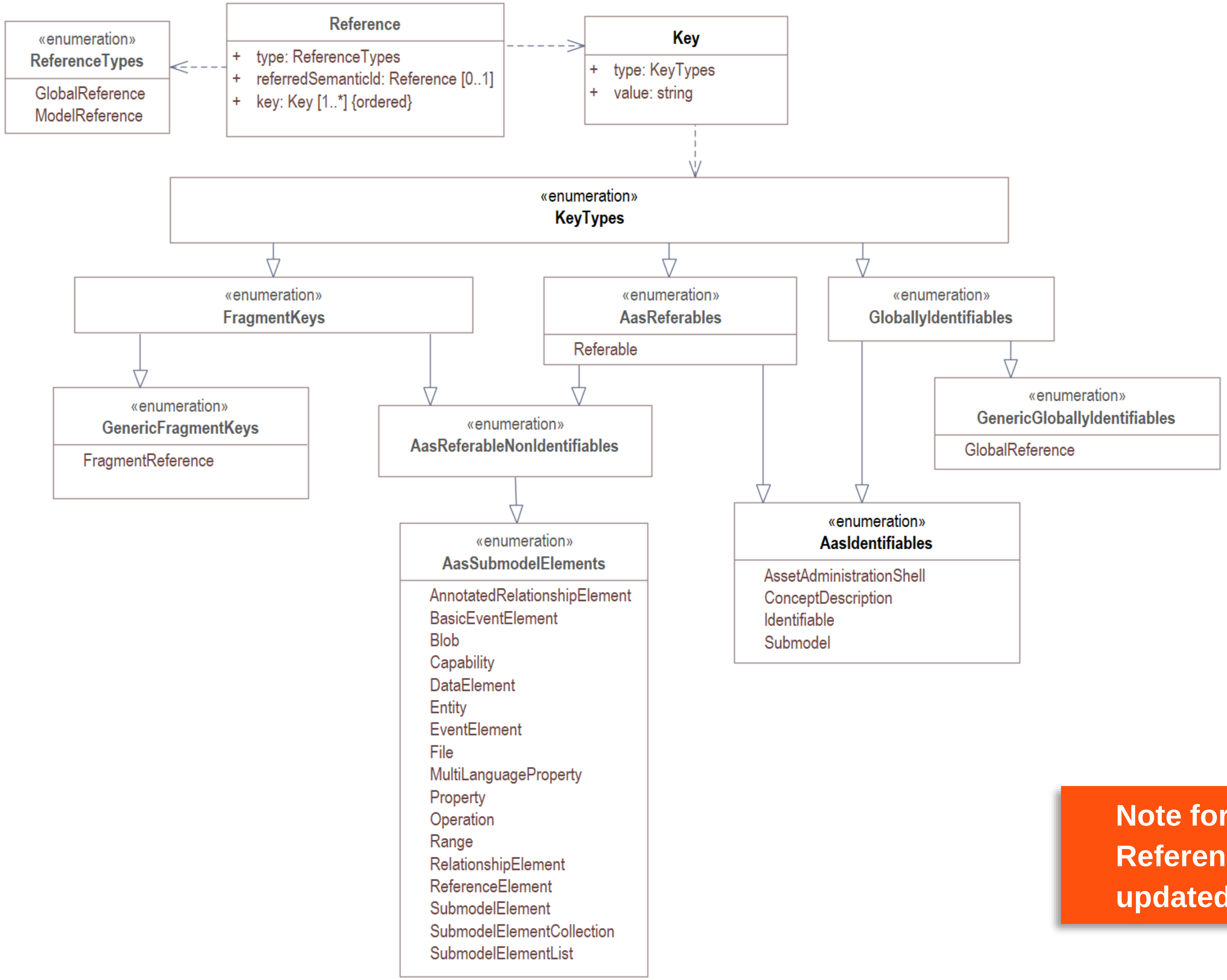
Note: Do not mix up with AssetKind.
AssetKind reflects the time in the life cycle of a product, e.g. in Engineering phase it is a product type, in production it is an product instance that is produced.

Referencing

External Global References (e.g. to ECLASS IRDI, manfuacturer Web-Site)

References into a File (Fragment)

Model References to any Referable in an AAS model (e.g. to define relationships between elements)



«enumeration»
ReferenceTypes
GlobalReference
ModelReference

Reference
+ type: ReferenceTypes
+ referredSemanticId: Reference [0..1]
+ key: Key [1..*] {ordered}

Key
+ type: KeyTypes
+ value: string

«enumeration»
KeyTypes

«enumeration»
FragmentKeys

«enumeration»
AasReferables
Referable

«enumeration»
GloballyIdentifiables

«enumeration»
GenericFragmentKeys
FragmentReference

«enumeration»
AasReferableNonIdentifiables

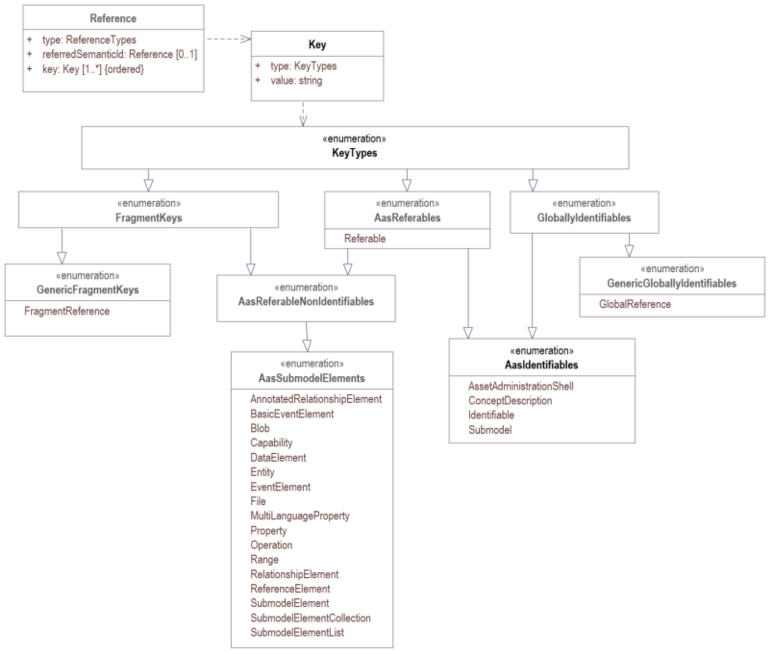
«enumeration»
GenericGloballyIdentifiables
GlobalReference

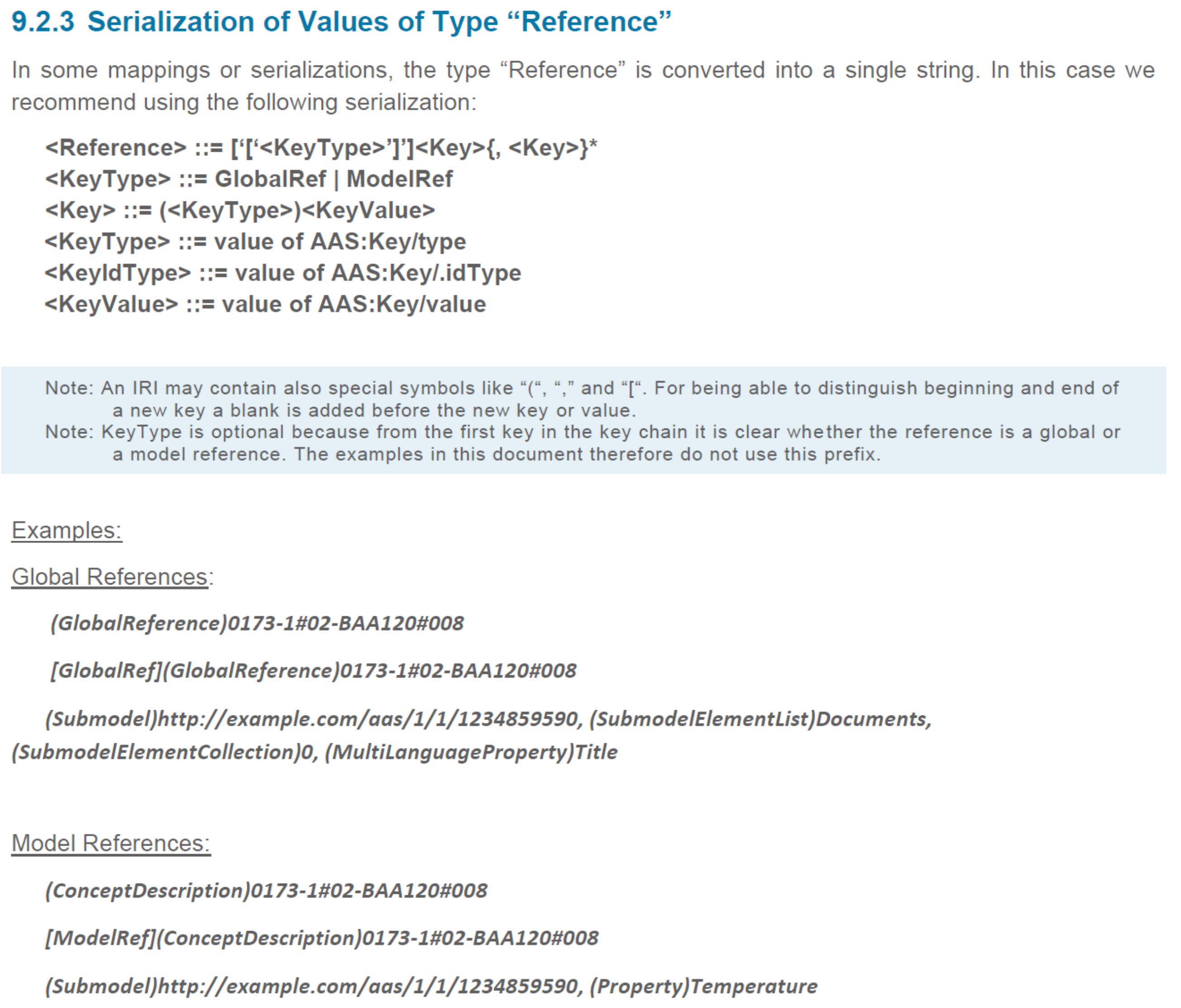
«enumeration»
AasSubmodelElements
AnnotatedRelationshipElement
BasicEventElement
Blob
Capability
DataElement
Entity
EventElement
File
MultiLanguageProperty
Property
Operation
Range
RelationshipElement
ReferenceElement
SubmodelElement
SubmodelElementCollection
SubmodelElementList

«enumeration»
AasIdentifiables
AssetAdministrationShell
ConceptDescription
Identifiable
Submodel

Note for Experts: Reference Concept updated

Referencing - Examples





9.2.3 Serialization of Values of Type “Reference”

In some mappings or serializations, the type “Reference” is converted into a single string. In this case we recommend using the following serialization:

<Reference> ::= [‘[‘<KeyType>’]’]<Key>{, <Key>}*
<KeyType> ::= GlobalRef | ModelRef
<Key> ::= (<KeyType>)<KeyValue>
<KeyType> ::= value of AAS:Key/type
<KeyIdType> ::= value of AAS:Key/.idType
<KeyValue> ::= value of AAS:Key/value

Note: An IRI may contain also special symbols like “(“, “,” and “[“. For being able to distinguish beginning and end of a new key a blank is added before the new key or value.

Note: KeyType is optional because from the first key in the key chain it is clear whether the reference is a global or a model reference. The examples in this document therefore do not use this prefix.

Examples:

Global References:

(GlobalReference)0173-1#02-BAA120#008

[GlobalRef](GlobalReference)0173-1#02-BAA120#008

(Submodel)http://example.com/aas/1/1/1234859590, (SubmodelElementList)Documents, (SubmodelElementCollection)0, (MultiLanguageProperty)Title

Model References:

(ConceptDescription)0173-1#02-BAA120#008

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

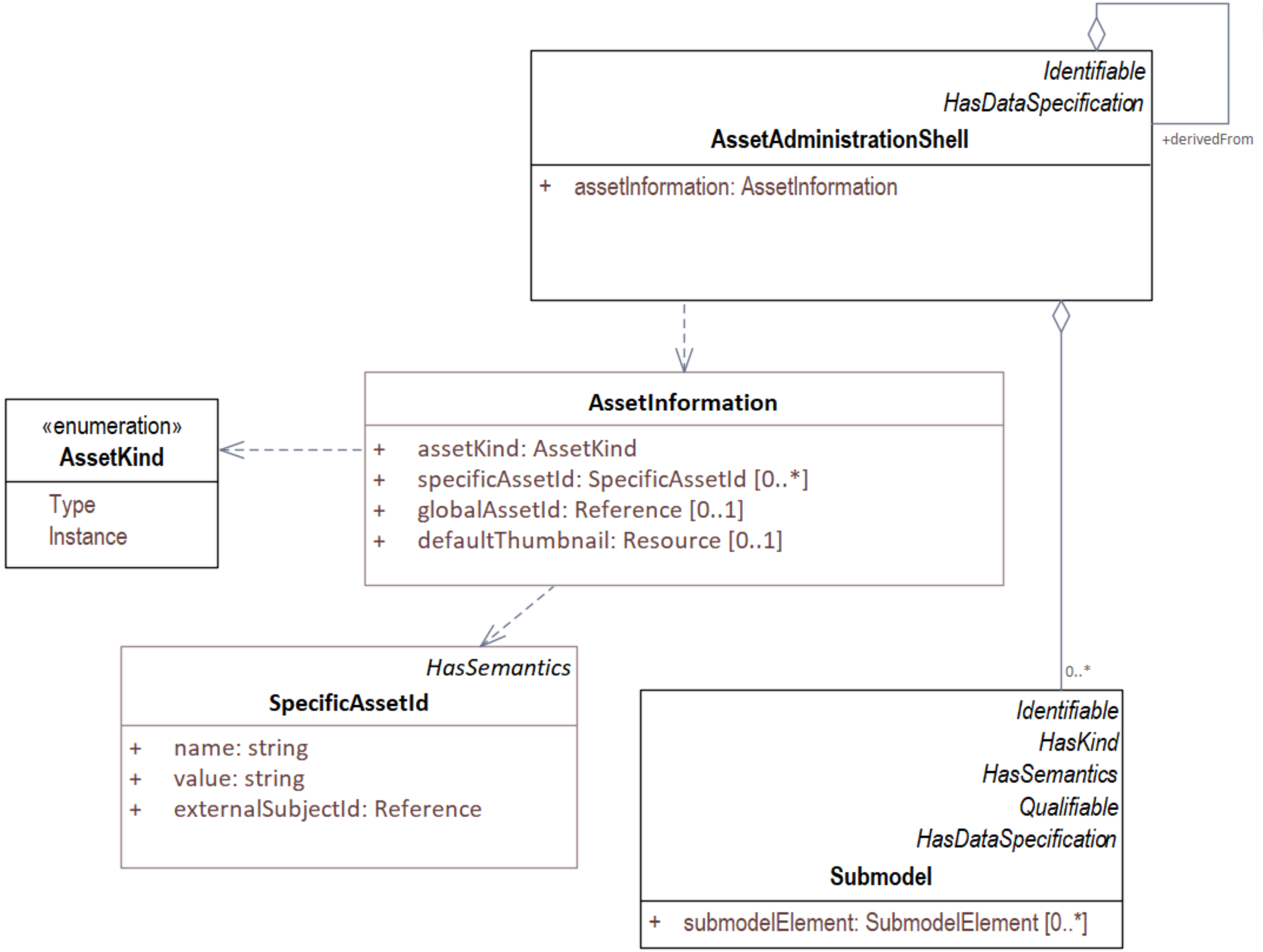
(Submodel)http://example.com/aas/1/1/1234859590, (Property)Temperature

Administration shell
Serial number
Inverter current
i700E70
231231
0.02
Administration shell
Serial number
Inverter current

Now dive in

The Asset Administration Shell

Note for Experts: Security and Asset Administration Shell now loosely coupled only



Identifiable
HasDataSpecification
AssetAdministrationShell
+ assetInformation: AssetInformation

+derivedFrom

AssetInformation
+ assetKind: AssetKind
+ specificAssetId: SpecificAssetId [0..*]
+ globalAssetId: Reference [0..1]
+ defaultThumbnail: Resource [0..1]

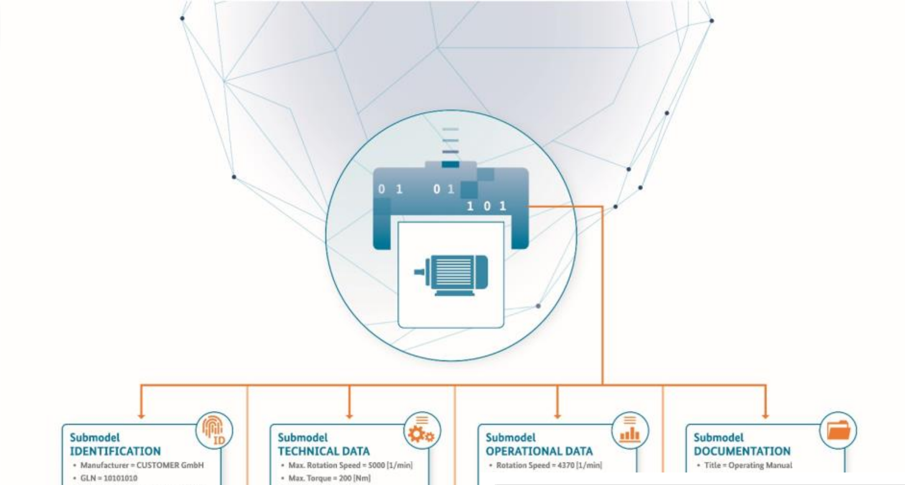
«enumeration»
AssetKind
Type
Instance

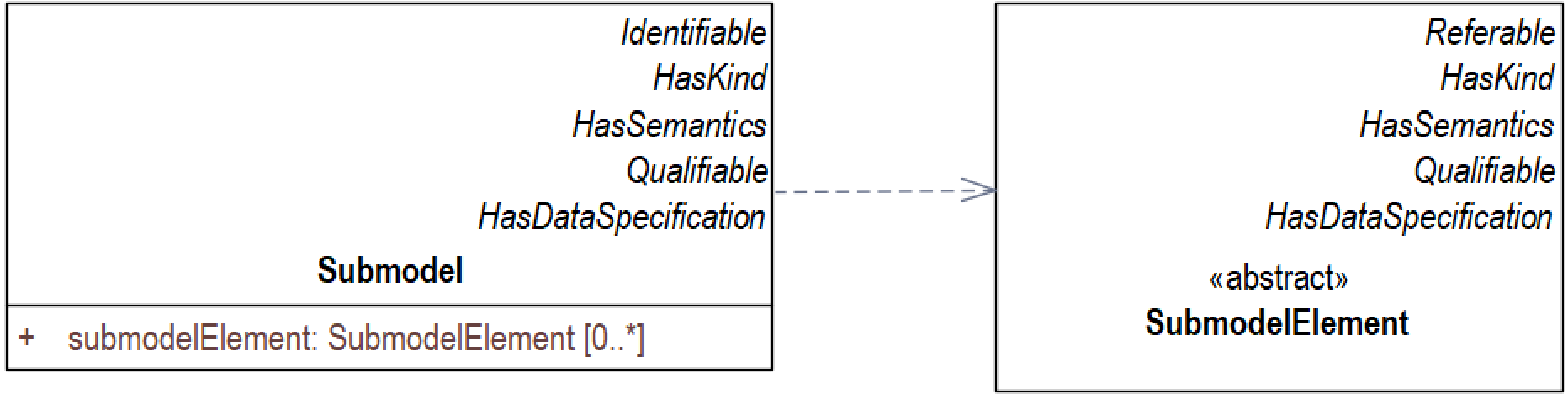
HasSemantics
SpecificAssetId
+ name: string
+ value: string
+ externalSubjectId: Reference

0..*

Identifiable
HasKind
HasSemantics
Qualifiable
HasDataSpecification
Submodel
+ submodelElement: SubmodelElement [0..*]

Submodel


0 1 0 1
1 0 1
Submodel
IDENTIFICATION
• Manufacturer = CUSTOMER GmbH
• GLN = 10101010
• Manufacturers Product Designation = I40 Capable Servo Motor
• Serial Number = P12345678I40
Submodel
TECHNICAL DATA
• Max. Rotation Speed = 5000 [1/min]
• Max. Torque = 200 [Nm]
• Cooling Type = BAB657
Submodel
OPERATIONAL DATA
• Rotation Speed = 4370 [1/min]
Submodel
DOCUMENTATION
• Title = Operating Manual
Submodel
Submodel



Identifiable
HasKind
HasSemantics
Qualifiable
HasDataSpecification

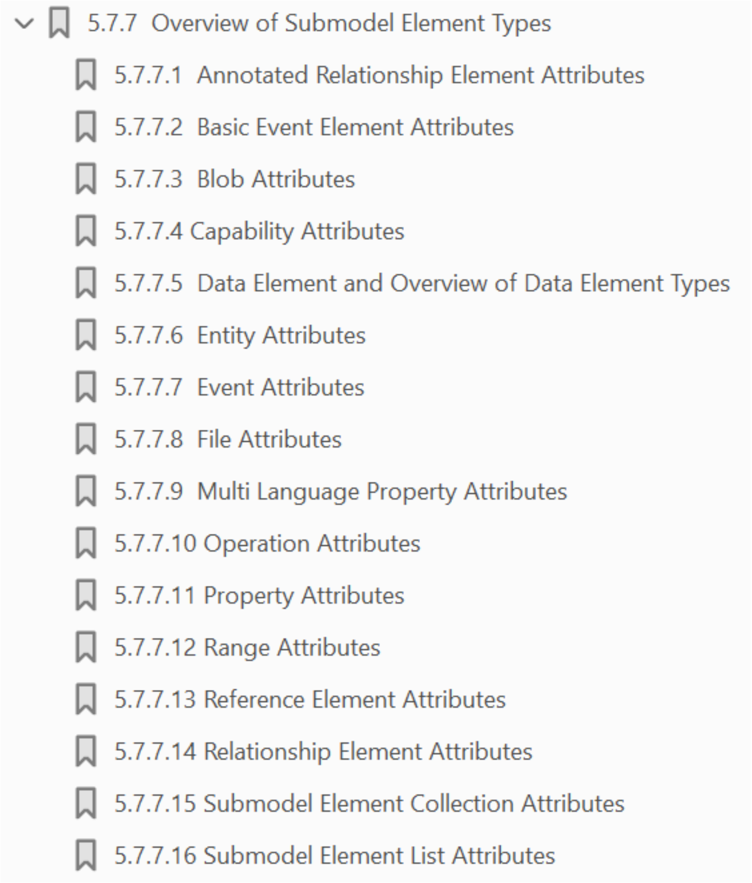
Submodel

+ submodelElement: SubmodelElement [0..*]

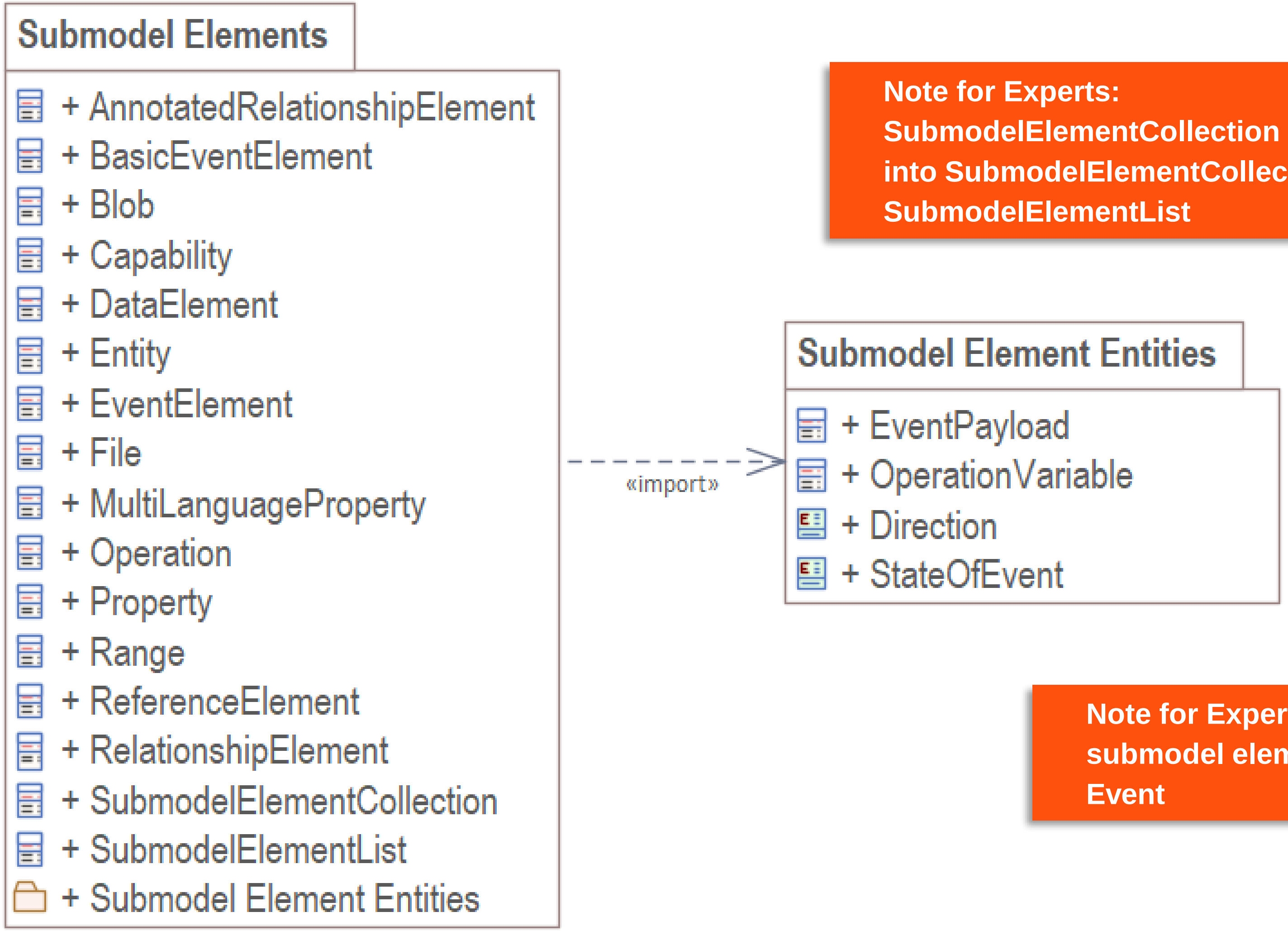
Referable
HasKind
HasSemantics
Qualifiable
HasDataSpecification

«abstract»
SubmodelElement

Overview – Package Submodel Elements



- 5.7.7 Overview of Submodel Element Types
  - 5.7.7.1 Annotated Relationship Element Attributes
  - 5.7.7.2 Basic Event Element Attributes
  - 5.7.7.3 Blob Attributes
  - 5.7.7.4 Capability Attributes
  - 5.7.7.5 Data Element and Overview of Data Element Types
  - 5.7.7.6 Entity Attributes
  - 5.7.7.7 Event Attributes
  - 5.7.7.8 File Attributes
  - 5.7.7.9 Multi Language Property Attributes
  - 5.7.7.10 Operation Attributes
  - 5.7.7.11 Property Attributes
  - 5.7.7.12 Range Attributes
  - 5.7.7.13 Reference Element Attributes
  - 5.7.7.14 Relationship Element Attributes
  - 5.7.7.15 Submodel Element Collection Attributes
  - 5.7.7.16 Submodel Element List Attributes



Submodel Elements

- + AnnotatedRelationshipElement
- + BasicEventElement
- + Blob
- + Capability
- + DataElement
- + Entity
- + EventElement
- + File
- + MultiLanguageProperty
- + Operation
- + Property
- + Range
- + ReferenceElement
- + RelationshipElement
- + SubmodelElementCollection
- + SubmodelElementList
- + Submodel Element Entities

«import»

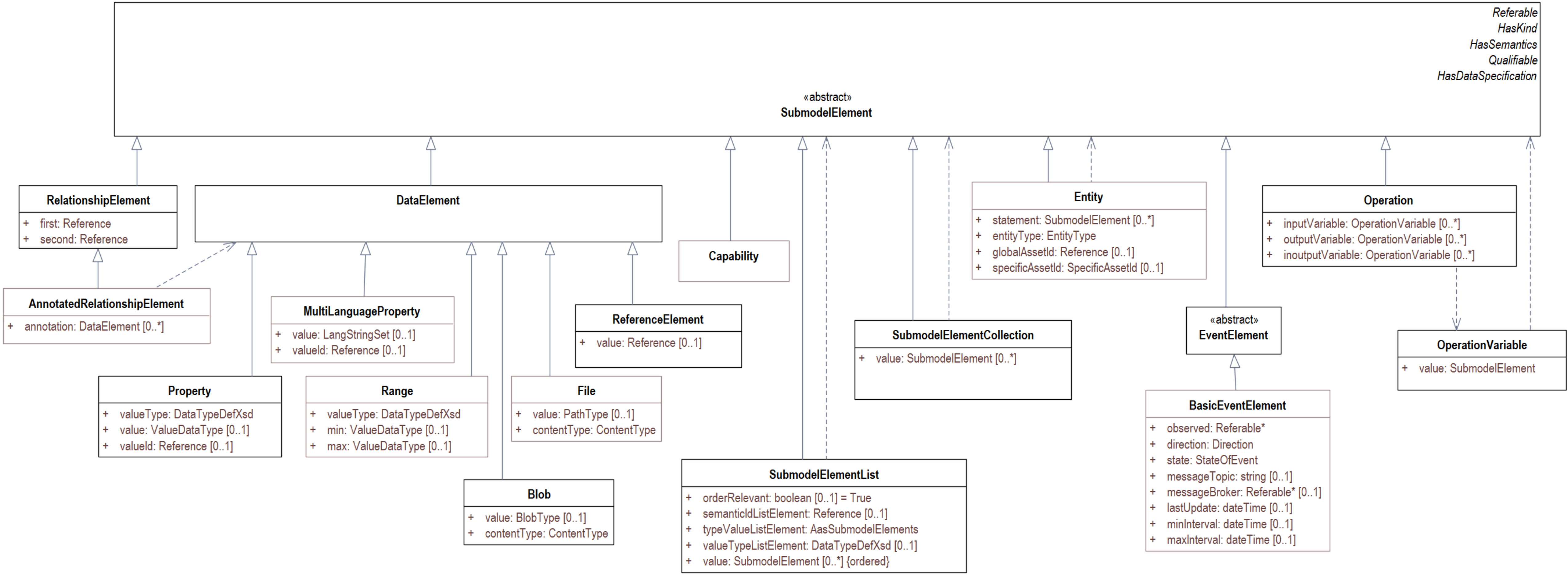
Submodel Element Entities

- + EventPayload
- + OperationVariable
- + Direction
- + StateOfEvent

Note for Experts: SubmodelElementCollection splitted into SubmodelElementCollection and SubmodelElementList

Note for Experts: updated submodel element Basic Event

Submodel Element Subtypes



Referable
HasKind
HasSemantics
Qualifiable
HasDataSpecification

«abstract»
SubmodelElement

RelationshipElement
- + first: Reference
- + second: Reference

AnnotatedRelationshipElement
- + annotation: DataElement [0..*]

DataElement

MultiLanguageProperty
- + value: LangStringSet [0..1]
- + valueId: Reference [0..1]

ReferenceElement
- + value: Reference [0..1]

Property
- + valueType: DataTypeDefXsd
- + value: ValueDataType [0..1]
- + valueId: Reference [0..1]

Range
- + valueType: DataTypeDefXsd
- + min: ValueDataType [0..1]
- + max: ValueDataType [0..1]

File
- + value: PathType [0..1]
- + contentType: ContentType

Blob
- + value: BlobType [0..1]
- + contentType: ContentType

Capability

SubmodelElementList
- + orderRelevant: boolean [0..1] = True
- + semanticIdListElement: Reference [0..1]
- + typeValueListElement: AasSubmodelElements
- + valueTypeListElement: DataTypeDefXsd [0..1]
- + value: SubmodelElement [0..*] {ordered}

SubmodelElementCollection
- + value: SubmodelElement [0..*]

Entity
- + statement: SubmodelElement [0..*]
- + entityType: EntityType
- + globalAssetId: Reference [0..1]
- + specificAssetId: SpecificAssetId [0..1]

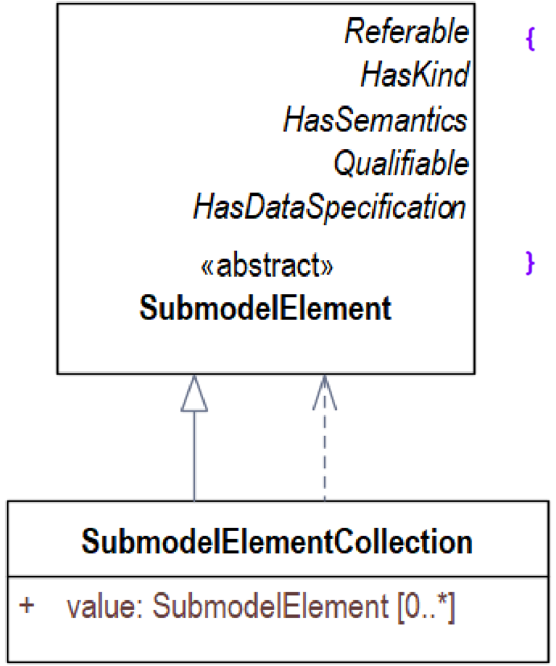
«abstract»
EventElement

BasicEventElement
- + observed: Referable*
- + direction: Direction
- + state: StateOfEvent
- + messageTopic: string [0..1]
- + messageBroker: Referable* [0..1]
- + lastUpdate: dateTime [0..1]
- + minInterval: dateTime [0..1]
- + maxInterval: dateTime [0..1]

Operation
- + inputVariable: OperationVariable [0..*]
- + outputVariable: OperationVariable [0..*]
- + inoutputVariable: OperationVariable [0..*]

OperationVariable
- + value: SubmodelElement

Submodel Element Collections and Lists



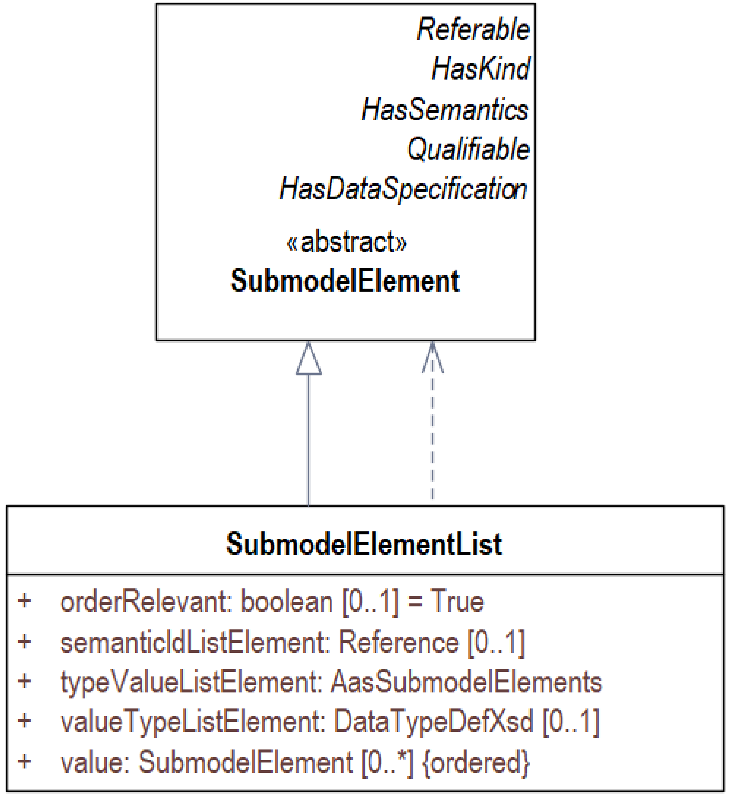
Referable
HasKind
HasSemantics
Qualifiable
HasDataSpecification
«abstract»
SubmodelElement

SubmodelElementCollection
+ value: SubmodelElement [0..*]

```
{
    "NamesOfFamilyMembers": {
        "NameOfMother": "Martha ExampleFamily",
        "NameOfFather": "Jonathan ExampleFamily",
        "NameOfSon": "Clark ExampleFamily"
    }
}
```

Difference in serialization in ValueOnly Format

```
{
    "NamesOfFamilyMembers": [
        "Martha ExampleFamily",
        "Jonathan ExampleFamily",
        "Clark ExampleFamily"
    ]
}
```

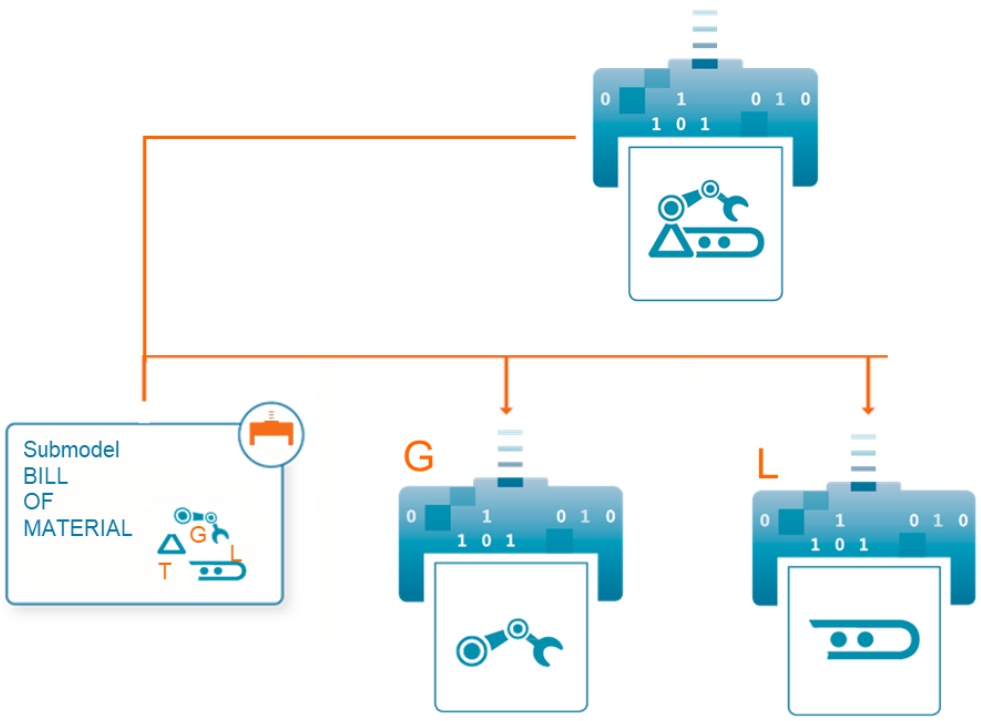


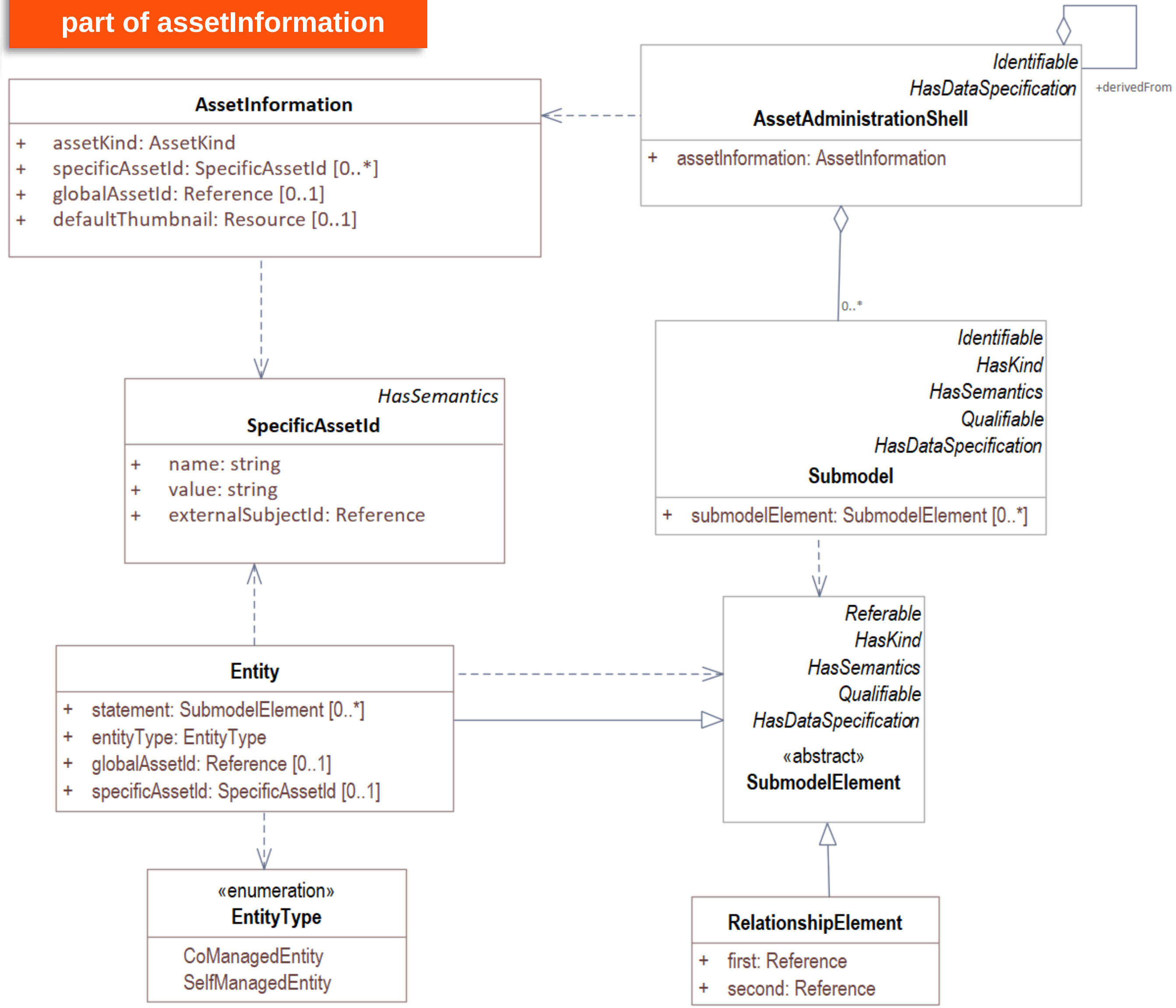
Referable
HasKind
HasSemantics
Qualifiable
HasDataSpecification
«abstract»
SubmodelElement

SubmodelElementList
+ orderRelevant: boolean [0..1] = True
+ semanticIdListElement: Reference [0..1]
+ typeValueListElement: AasSubmodelElements
+ valueTypeListElement: DataTypeDefXsd [0..1]
+ value: SubmodelElement [0..*] {ordered}

Composite I4.0 Components

Note for Experts: no bill of materials any longer as part of assetInformation





AssetInformation
+ assetKind: AssetKind
+ specificAssetId: SpecificAssetId [0..*]
+ globalAssetId: Reference [0..1]
+ defaultThumbnail: Resource [0..1]

Identifiable
HasDataSpecification
AssetAdministrationShell
+ assetInformation: AssetInformation

+derivedFrom

0..*

HasSemantics
SpecificAssetId
+ name: string
+ value: string
+ externalSubjectId: Reference

Identifiable
HasKind
HasSemantics
Qualifiable
HasDataSpecification
Submodel
+ submodelElement: SubmodelElement [0..*]

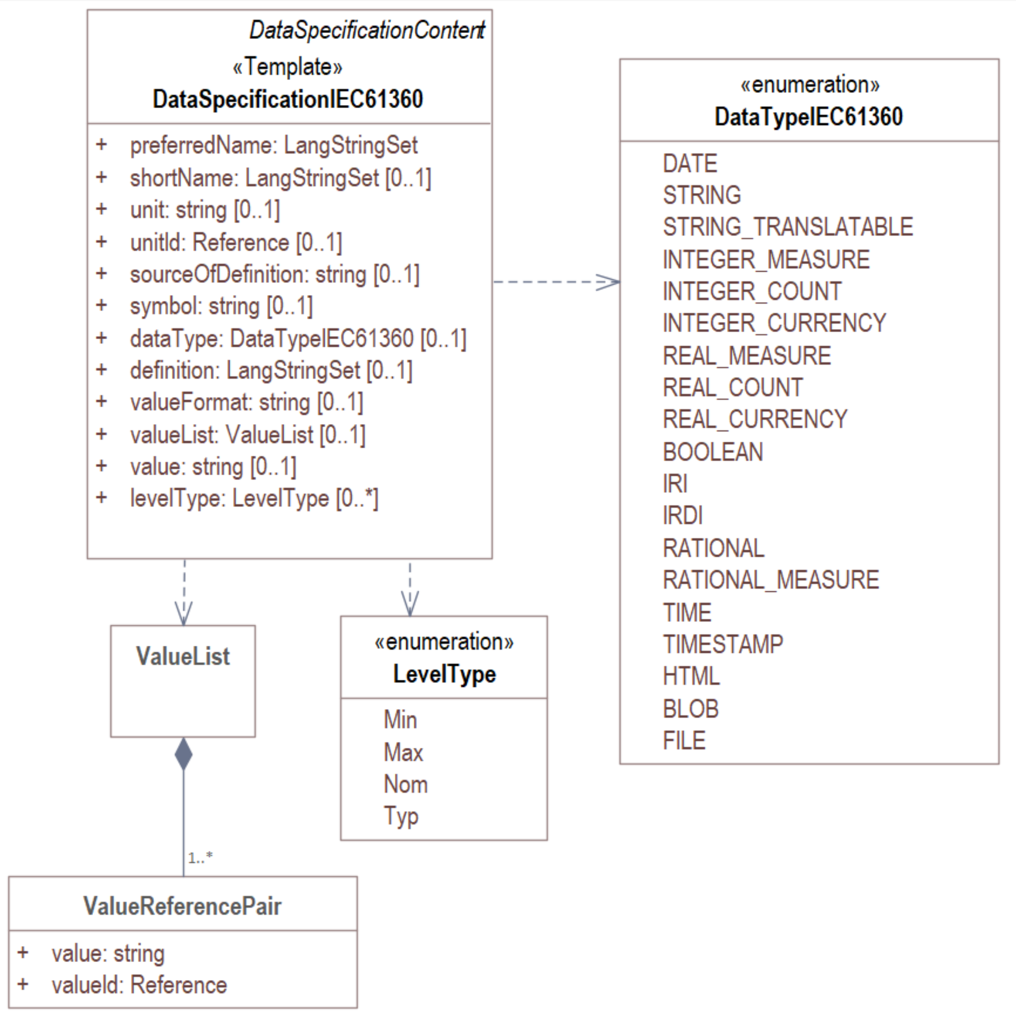
Entity
+ statement: SubmodelElement [0..*]
+ entityType: EntityType
+ globalAssetId: Reference [0..1]
+ specificAssetId: SpecificAssetId [0..1]

Referable
HasKind
HasSemantics
Qualifiable
HasDataSpecification
«abstract»
SubmodelElement

«enumeration»
EntityType
CoManagedEntity
SelfManagedEntity

RelationshipElement
+ first: Reference
+ second: Reference

Predefined Data Specification Template IEC61360

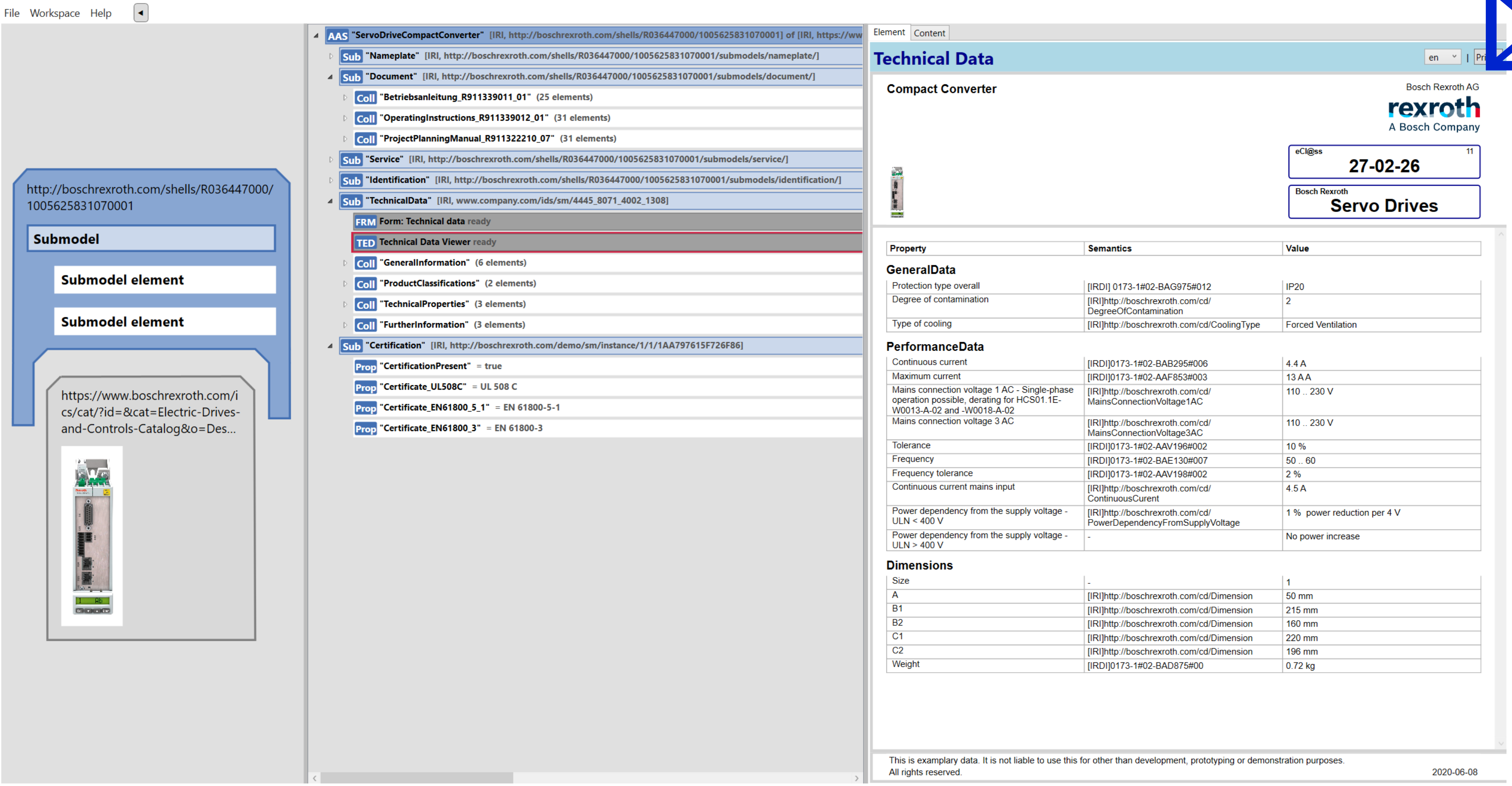


Property	**0173-1#02-BAA120#008** **Max. rotation speed**
Data type	INTEGER_MEASURE
Unit of measure	1/min
Definition	Greatest possible rotation speed with wich the motor or feeding unit may be operated

Note for Experts: additional data types

Create your first digital twin

AASX Package Explorer


File
Workspace
Help
http://boschrexroth.com/shells/R036447000/1005625831070001
Submodel
Submodel element
Submodel element
https://www.boschrexroth.com/ics/cat/?id=&cat=Electric-Drives-and-Controls-Catalog&o=Des...
Element
Content
en
Pri

AAS "ServoDriveCompactConverter" [IRI, http://boschrexroth.com/shells/R036447000/1005625831070001] of [IRI, https://ww

Sub "Nameplate" [IRI, http://boschrexroth.com/shells/R036447000/1005625831070001/submodels/nameplate/]

Sub "Document" [IRI, http://boschrexroth.com/shells/R036447000/1005625831070001/submodels/document/]

Coll "Betriebsanleitung_R911339011_01" (25 elements)

Coll "OperatingInstructions_R911339012_01" (31 elements)

Coll "ProjectPlanningManual_R911322210_07" (31 elements)

Sub "Service" [IRI, http://boschrexroth.com/shells/R036447000/1005625831070001/submodels/service/]

Sub "Identification" [IRI, http://boschrexroth.com/shells/R036447000/1005625831070001/submodels/identification/]

Sub "TechnicalData" [IRI, www.company.com/ids/sm/4445_8071_4002_1308]

FRM Form: Technical data ready

TED Technical Data Viewer ready

Coll "GeneralInformation" (6 elements)

Coll "ProductClassifications" (2 elements)

Coll "TechnicalProperties" (3 elements)

Coll "FurtherInformation" (3 elements)

Sub "Certification" [IRI, http://boschrexroth.com/demo/sm/instance/1/1/1AA797615F726F86]

Prop "CertificationPresent" = true

Prop "Certificate_UL508C" = UL 508 C

Prop "Certificate_EN61800_5_1" = EN 61800-5-1

Prop "Certificate_EN61800_3" = EN 61800-3

Technical Data

Compact Converter

Bosch Rexroth AG

rexroth
A Bosch Company

eCl@ss 11
27-02-26

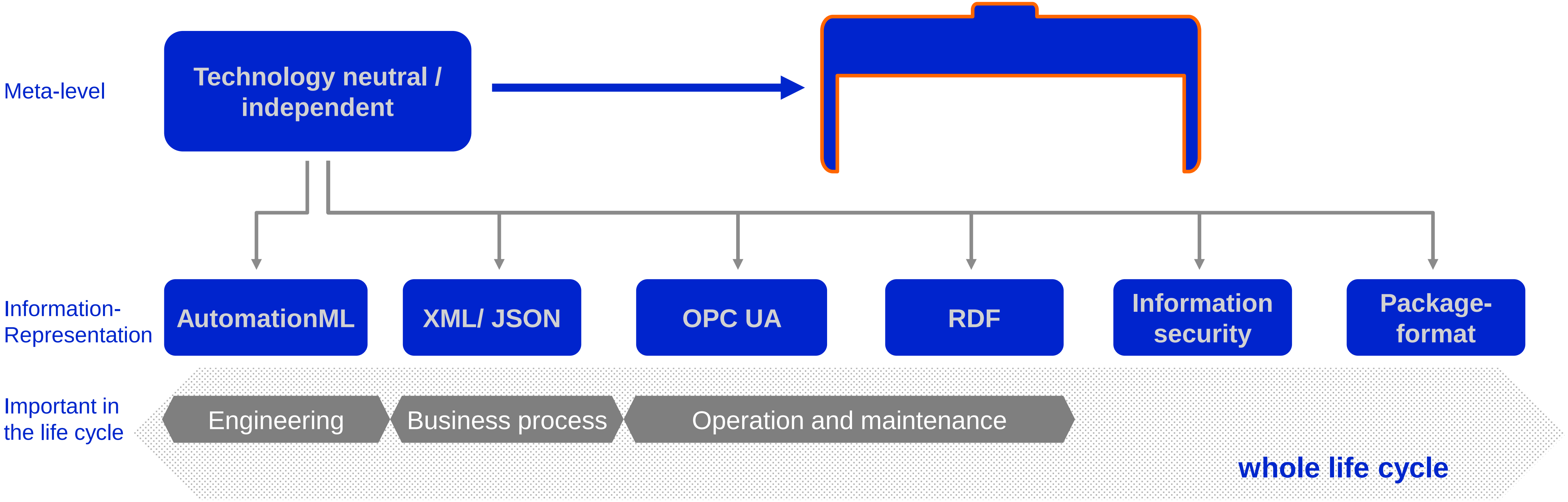
Bosch Rexroth
Servo Drives

Property	Semantics	Value
GeneralData		
Protection type overall	[IRDI] 0173-1#02-BAG975#012	IP20
Degree of contamination	[IRI]http://boschrexroth.com/cd/DegreeOfContamination	2
Type of cooling	[IRI]http://boschrexroth.com/cd/CoolingType	Forced Ventilation
PerformanceData		
Continuous current	[IRDI]0173-1#02-BAB295#006	4.4 A
Maximum current	[IRDI]0173-1#02-AAF853#003	13 A A
Mains connection voltage 1 AC - Single-phase operation possible, derating for HCS01.1E-W0013-A-02 and -W0018-A-02	[IRI]http://boschrexroth.com/cd/MainsConnectionVoltage1AC	110 .. 230 V
Mains connection voltage 3 AC	[IRI]http://boschrexroth.com/cd/MainsConnectionVoltage3AC	110 .. 230 V
Tolerance	[IRDI]0173-1#02-AAV196#002	10 %
Frequency	[IRDI]0173-1#02-BAE130#007	50 .. 60
Frequency tolerance	[IRDI]0173-1#02-AAV198#002	2 %
Continuous current mains input	[IRI]http://boschrexroth.com/cd/ContinuousCurent	4.5 A
Power dependency from the supply voltage - ULN < 400 V	[IRI]http://boschrexroth.com/cd/PowerDependencyFromSupplyVoltage	1 % power reduction per 4 V
Power dependency from the supply voltage - ULN > 400 V	-	No power increase
Dimensions		
Size	-	1
A	[IRI]http://boschrexroth.com/cd/Dimension	50 mm
B1	[IRI]http://boschrexroth.com/cd/Dimension	215 mm
B2	[IRI]http://boschrexroth.com/cd/Dimension	160 mm
C1	[IRI]http://boschrexroth.com/cd/Dimension	220 mm
C2	[IRI]http://boschrexroth.com/cd/Dimension	196 mm
Weight	[IRDI]0173-1#02-BAD875#00	0.72 kg

This is examplary data. It is not liable to use this for other than development, prototyping or demonstration purposes.
All rights reserved. 2020-06-08

https://github.com/admin-shell-io/aasx-package-explorer

Life Cycle Phases and Formats



Meta-level

Technology neutral / independent

Information-Representation

AutomationML

XML/ JSON

OPC UA

RDF

Information security

Package-format

Important in the life cycle

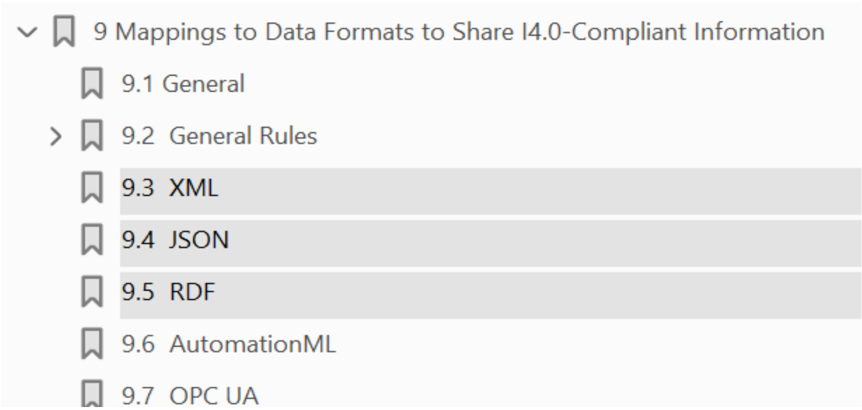
Engineering

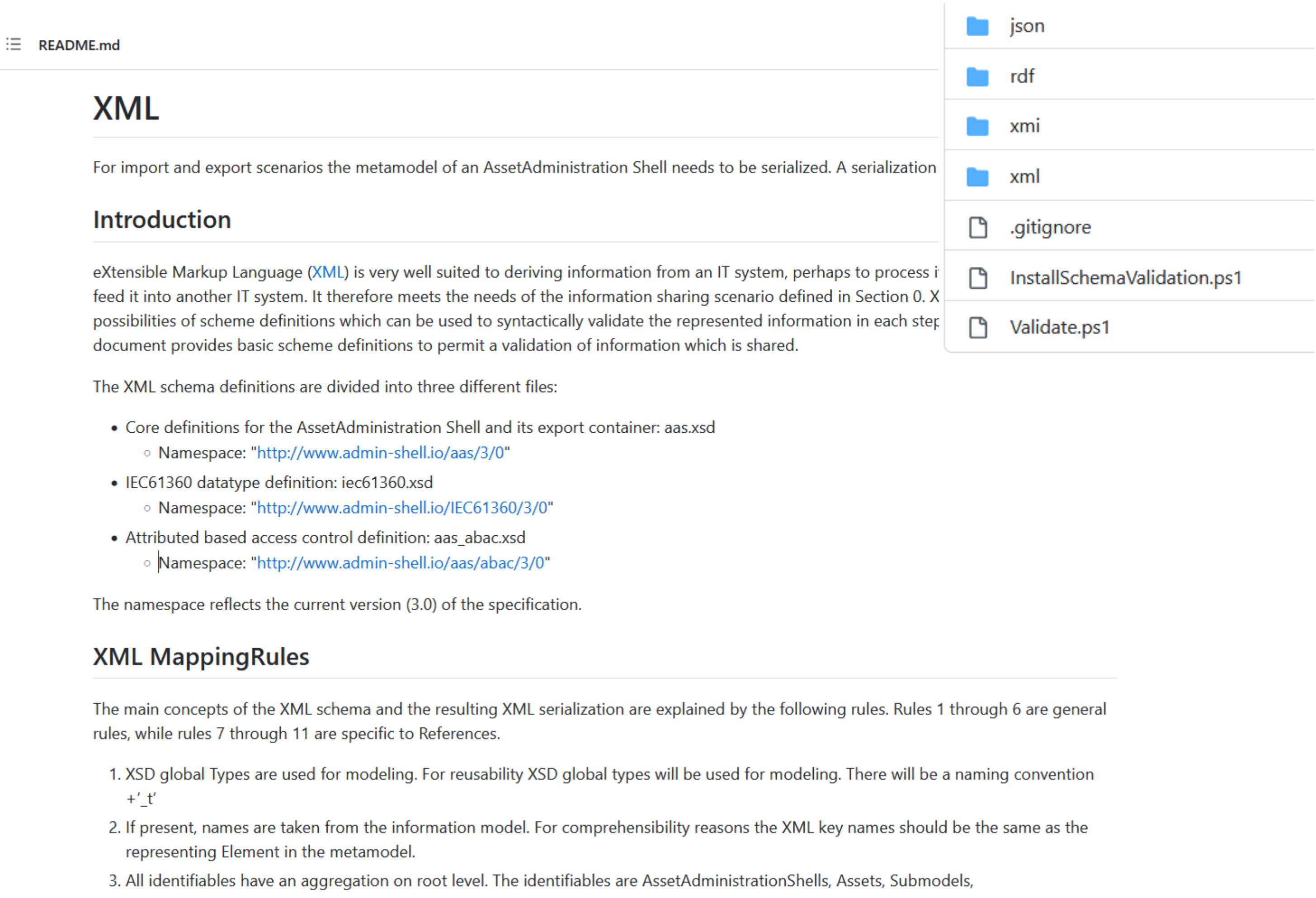
Business process

Operation and maintenance

whole life cycle

Serializations





XML

For import and export scenarios the metamodel of an AssetAdministration Shell needs to be serialized. A serialization

Introduction

eXtensible Markup Language (XML) is very well suited to deriving information from an IT system, perhaps to process i feed it into another IT system. It therefore meets the needs of the information sharing scenario defined in Section 0. X possibilities of scheme definitions which can be used to syntactically validate the represented information in each step 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/3/0"
- IEC61360 datatype definition: iec61360.xsd
  - Namespace: "http://www.admin-shell.io/IEC61360/3/0"
- Attributed based access control definition: aas_abac.xsd
  - Namespace: "http://www.admin-shell.io/aas/abac/3/0"

The namespace reflects the current version (3.0) of the specification.

XML MappingRules

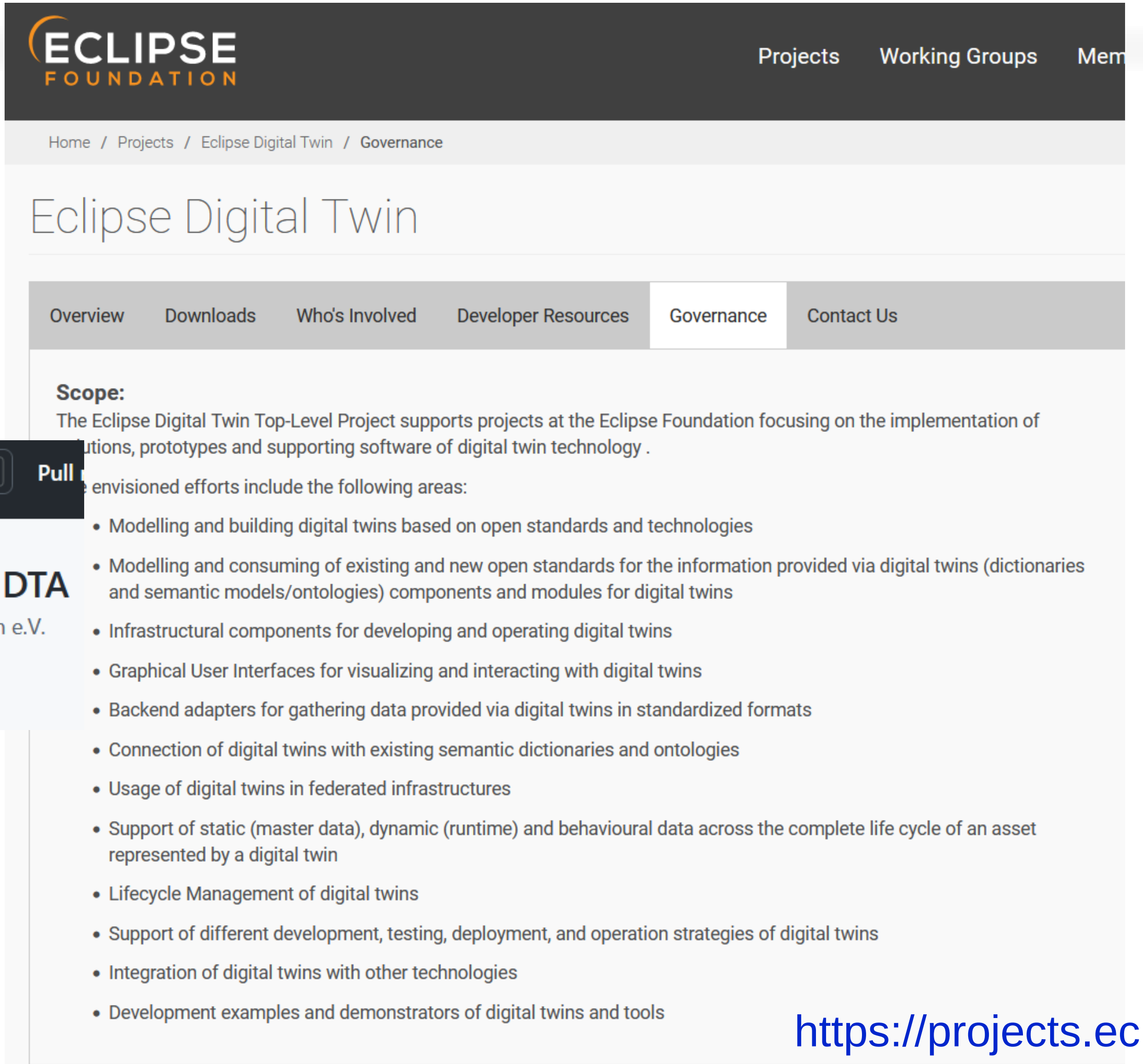
The main concepts of the XML schema and the resulting XML serialization are explained by the following rules. Rules 1 through 6 are general rules, while rules 7 through 11 are specific to References.

1. XSD global Types are used for modeling. For reusability XSD global types will be used for modeling. There will be a naming convention +'_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,

https://github.com/admin-shell-io/aas-specs/tree/master/schemas

Note for Experts: Mapping Rules and Schema as well as examples not part of specification any longer ➔ now part of open source project admin-shell-io/aas-specs or maintained at other organizations.

Open Source Support

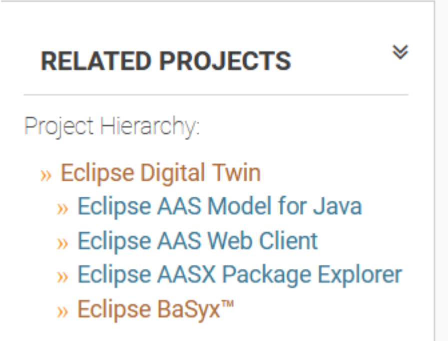


Eclipse Digital Twin

Scope:
The Eclipse Digital Twin Top-Level Project supports projects at the Eclipse Foundation focusing on the implementation of utions, prototypes and supporting software of digital twin technology .

envisioned efforts include the following areas:

- Modelling and building digital twins based on open standards and technologies
- Modelling and consuming of existing and new open standards for the information provided via digital twins (dictionaries and semantic models/ontologies) components and modules for digital twins
- Infrastructural components for developing and operating digital twins
- Graphical User Interfaces for visualizing and interacting with digital twins
- Backend adapters for gathering data provided via digital twins in standardized formats
- Connection of digital twins with existing semantic dictionaries and ontologies
- Usage of digital twins in federated infrastructures
- Support of static (master data), dynamic (runtime) and behavioural data across the complete life cycle of an asset represented by a digital twin
- Lifecycle Management of digital twins
- Support of different development, testing, deployment, and operation strategies of digital twins
- Integration of digital twins with other technologies
- Development examples and demonstrators of digital twins and tools

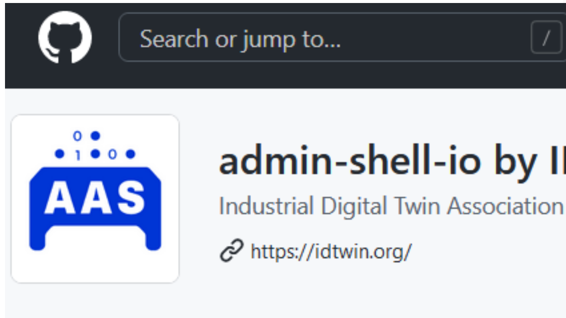


RELATED PROJECTS

Project Hierarchy:

- » Eclipse Digital Twin
  - » Eclipse AAS Model for Java
  - » Eclipse AAS Web Client
  - » Eclipse AASX Package Explorer
  - » Eclipse BaSyx™

Status: June 2022



admin-shell-io by IDTA

Industrial Digital Twin Association e.V.

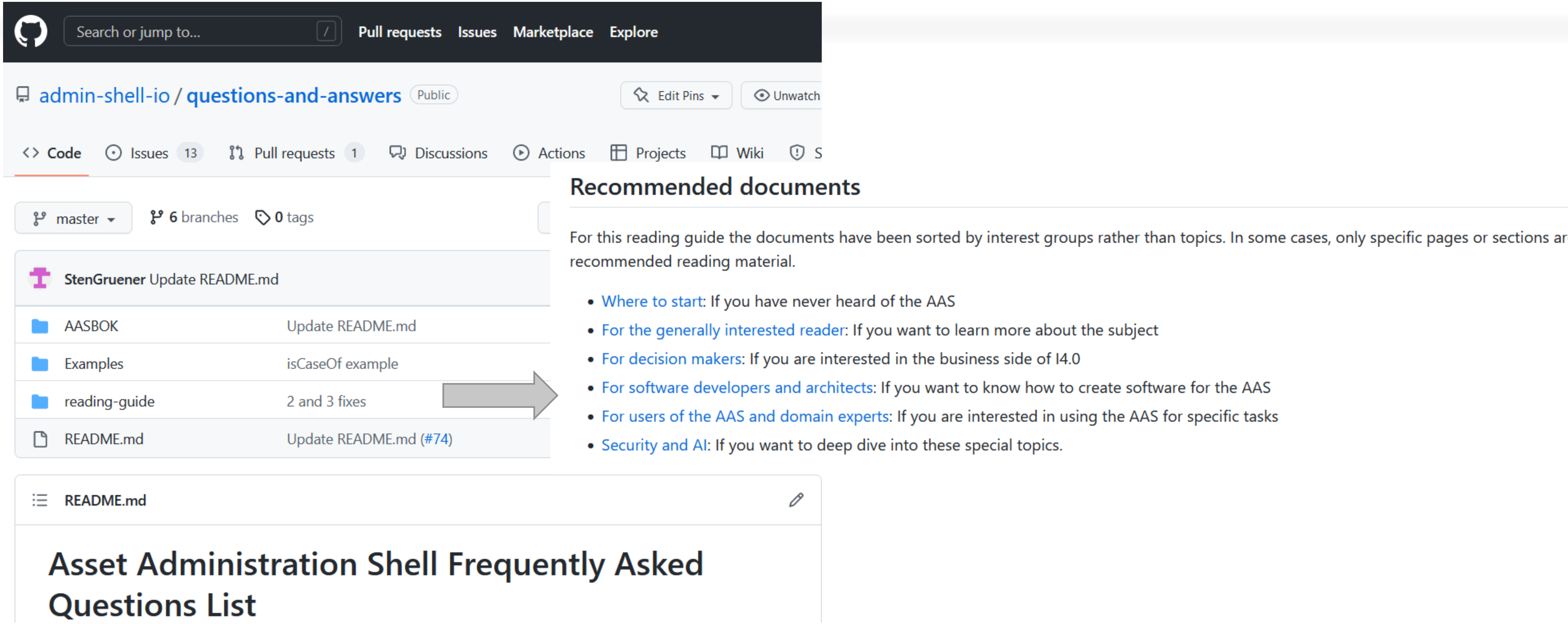
https://idtwin.org/

https://github.com/orgs/admin-shell-io/

https://projects.eclipse.org/projects/dt/

Still Questions?

Questions and Answers



StenGruener Update README.md

AASBOK	Update README.md
Examples	isCaseOf example
reading-guide	2 and 3 fixes
README.md	Update README.md (#74)

Asset Administration Shell Frequently Asked Questions List

Recommended documents

For this reading guide the documents have been sorted by interest groups rather than topics. In some cases, only specific pages or sections ar recommended reading material.

- Where to start: If you have never heard of the AAS
- For the generally interested reader: If you want to learn more about the subject
- For decision makers: If you are interested in the business side of I4.0
- For software developers and architects: If you want to know how to create software for the AAS
- For users of the AAS and domain experts: If you are interested in using the AAS for specific tasks
- Security and AI: If you want to deep dive into these special topics.

https://github.com/admin-shell-io/questions-and-answers

AAS Academy

Willkommen auf der Review-Seite zum IDTA-Training. Diese Seite ist Ihr persönlicher Startpunkt, um auf Lerninhalte von University4Industry zuzugreifen, die für Sie ausgewählt wurden.

www.u4i.io/IDTA

Let´s go!

MEMBERS SUPPLIERS USERS

ABB
accenture
asentics vision technology
bill-X Collect. Manage. Execute.
bitkom
BOSCH Technik fürs Leben
BLUE INNOVATORS
Class.Ing Ingenieur-Partnerschaft für Mediendaten-management Scherenschlich & Rukavina
CLOUD item

codewerk
conplement the digital enablers
Danfoss
dunkermotoren
ECHOPRM PRODUCT RELATIONSHIP MANAGEMENT
Endress+Hauser EH People for Process Automation
equinor
FESTO
Fraunhofer

GRi SIMULATIONS
HEGLA New Technology Innovations for Glass
HARTING Pushing Performance
HOMAG
HUAWEI
ifm
IndustryApps Appstore for Industry 4.0
INspares GmbH
INSTITUTE FOR ADVANCED MANUFACTURING AND ENGINEERING COVENTRY UNIVERSITY · UK

INTRANAV.
INTERX
ISTOS A MEMBER OF DMG MORI
KION GROUP
KUKA
Lenze
LS ELECTRIC
MCS
Microsoft

NINEFEB
nnaisense
objective partner
OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG
PEPPERL+FUCHS
PHOENIX CONTACT
PROSTEP integrate the future
rexroth A Bosch Company
SAP

MURR ELEKTRONIK stay connected
NEST+ FIELD
SCHMALZ
Life Is On Schneider Electric
SCHUNK
SDU
SICK Sensor Intelligence.
SIEMENS
sommer kennzeichnend.

STAHL
THREEDY GmbH instant 3Dhub
TRANSACTION NETWORK TN
TRUMPF
TURCK Your Global Automation Partner
TwinStore
UNITED GRINDING
UNIVERSITAS OSLOENSIS MDCCCXI
VDMA

Wiferion
WITTENSTEIN
XITASO
ZF GROUP
ZVEI: Die Elektroindustrie

Connect on

www.linkedin.com/in/birgit-boss/

Dr. Birgit Boss

Robert Bosch GmbH, Bosch Connected Industry

- Chair of the Joint Working Group “Asset Administration Shell” of the Working Group “Open Technology” of the Industrial Digital Twin Association (IDTA) and the Working Group “Reference Architecture, Standards and Norms” of the Plattform Industrie 4.0
- Board member of the Industrial Digital Twin Association (IDTA) (www.idtwin.org)
- Active in additional initiatives with focus semantics and digital twins

IDTA Industrial Digital Twin Association

www.industrialdigitaltwin.org