Building joint forces for the digital twin
The Asset Administration Shell Metamodel

From Concept to Specification

April 2023
V3.0

2018
V1.0
This Tutorial is about Part 3a
For whom is this tutorial?

• You should have an idea of the benefits of introducing digital twins to your domain
• You should know Tutorial Part 1 Metamodel
• You should have basic knowledge in UML modeling
• You are an architect and want to learn more about the underlying information model of the Asset Administration Shell
• You are a developer and want to upgrade to the new version or start your first implementation
For whom is this tutorial?

- You feel disappointed and do not know how to start?
Have a look at the Asset Administration Shell Reading Guide!
It is updated on a regular basis.

- 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 4.0
- 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.
Download Specification

https://industraldigitaltwin.org/content-hub/
1. Fetch release of AAS you are interested in
2. Import xmi file into UML tool (best with Enterprise/Architect)

Note: The data specifications are embedded into the Part 1 schemas
Metamodel Changes Part 3a

Note for Experts: <Notes for tutorial listeners who have knowledge of previous versions of the specification (V2.0 or Release Candidates of V3.0).

If you do not know previous versions you can ignore these notes.>

Annex E. Metamodel Changes
- General
- Changes V3.0 vs. Part 1 V2.0.1
- Changes V3.0 vs. Part 1 V3.0RC02

V3.0RC02
https://www.youtube.com/watch?v=QR-nOl6cuOI
1. Get (re-)familiar with general UML modeling rules
2. Get familiar with specific graphical representation of UML in the specification (partly tool specific)
UML Generalization/Inheritance

A) Class1
   + class2: Class2

B) Class1
   Class3

C) Class1
   Class3
   ::Class1
   + class2: Class2

Hint: Graphical representation tool specific
UML Shared Aggregation

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

A) Class1

B) Class1

+ attr: Reference

Class2

+ attr

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: Inheritance between enumerations is not widely used. It is only used for graphical illustration of relationships between enumerations.

```
<<enumeration>>
Enumeration1

literals
a

<<enumeration>>
Enumeration2

literals
b

identical to

<<enumeration>>
Enumeration1

literals
a
b
```
### Template for Classes

**Class:**

- `<Class Name> [<<abstract>>] ["<<Experimental>>"] ["<<Deprecated>>"] ["<<Template>>"]`

**Explanation:**

- `<Explanatory text>`

**Inherits from:**

- `{<Class Name> ";" } | ";"`

**Attribute**

<table>
<thead>
<tr>
<th>Attribute or association name</th>
<th>Explanation</th>
<th>Type</th>
<th>Card.</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;&lt;&lt;ordered&gt;&gt;&quot;</td>
<td><code>&lt;Explanatory text&gt;</code></td>
<td><code>&lt;Type&gt;</code></td>
<td><code>&lt;Card&gt;</code></td>
</tr>
<tr>
<td>&quot;&lt;&lt;Experimental&gt;&gt;&quot;</td>
<td>&quot;&lt;&lt;Experimental&gt;&gt;&quot;</td>
<td>&quot;&lt;&lt;Experimental&gt;&gt;&quot;</td>
<td>&quot;&lt;&lt;Experimental&gt;&gt;&quot;</td>
</tr>
<tr>
<td>&quot;&lt;&lt;Deprecated&gt;&gt;&quot;</td>
<td>&quot;&lt;&lt;Deprecated&gt;&gt;&quot;</td>
<td>&quot;&lt;&lt;Deprecated&gt;&gt;&quot;</td>
<td>&quot;&lt;&lt;Deprecated&gt;&gt;&quot;</td>
</tr>
</tbody>
</table>

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

Note for experts: no kind column any longer, instead different notation for Type.
### 5.3.7.12 Property Attributes

**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/valued` are present, the value of `Property/value` needs to be identical to the value of the referenced coded value in `Property/valued`.

**Inherits from:** DataElement

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Explanation</th>
<th>Type</th>
<th>Card.</th>
</tr>
</thead>
<tbody>
<tr>
<td>valueType</td>
<td>Data type of the value attribute</td>
<td>DataTypeDefXsd</td>
<td>1</td>
</tr>
<tr>
<td>value</td>
<td>The value of the property instance</td>
<td>ValueDataType</td>
<td>0..1</td>
</tr>
<tr>
<td>valued</td>
<td>Reference to the global unique ID of a coded value</td>
<td>Reference</td>
<td>0..1</td>
</tr>
</tbody>
</table>

Note: it is recommended to use an external reference.
Get warm – Prerequisites from PART 1 Metamodel
Data Specification for Concept Descriptions (see Part 1)
Part 1 – Data Specifications & Concept Descriptions

The Asset Administration Shell Metamodel

External Concept Definitions

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

«abstract»
DataSpecification
+ administration: AdministrativeInformation [0..1]
+ id: Identifier
+ dataSpecificationContent: DataSpecificationContent
+ description: MultiLanguageTextType [0..1]

«Template»
DataSpecificationContent

«Template»
DataSpecification

AssetInformation
+ assetKind
+ specificAssetId: SpecificAssetId [0..*]
+ globalAssetId: Identifier [0..1]
+ assetType: Identifier [0..1]
+ defaultThumbnail: Resource [0..1]

HasSemantics
SpecificAssetId
+ name: LabelType
+ value: Identifier
+ externalSubjectId: Reference [0..1]

«abstract»
Property
+ valueType: DataTypeDefXsd
+ value: ValueDataType [0..1]
+ valueId: Reference [0..1]

«Experimental»
+ kind: QualifierKind [0..1] = ConceptQualifier

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

«enumeration»
AssetKind
literals
Type
Instance
NotApplicable

Exemplary Submodel Element
“Property”, other submodel element subtypes include operations, collections, lists etc.

AssetInformation
+ assetKind
+ specificAssetId: SpecificAssetId [0..*]
+ globalAssetId: Identifier [0..1]
+ assetType: Identifier [0..1]
+ defaultThumbnail: Resource [0..1]

HasSemantics
SpecificAssetId
+ name: LabelType
+ value: Identifier
+ externalSubjectId: Reference [0..1]

«abstract»
Property
+ valueType: DataTypeDefXsd
+ value: ValueDataType [0..1]
+ valueId: Reference [0..1]

«Experimental»
+ kind: QualifierKind [0..1] = ConceptQualifier

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

«enumeration»
AssetKind
literals
Type
Instance
NotApplicable

Exemplary Submodel Element
“Property”, other submodel element subtypes include operations, collections, lists etc.

The Asset Administration Shell Metamodel
The semanticId is the identifier of the semantic definition of the element. Supplemental semantic IDs can be added. The semanticId can be an External Reference or a Model Reference to a **Concept Description**.
Part 1 Common - HasDataSpecification

«abstract» HasDataSpecification

+ dataSpecification: Reference [0..*]

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

Note for Experts: Data Specifications are not part of Part 1 any longer: They are part of data specifications series Part 3
Part 1 - Embedded Data Specifications

The Asset Administration Shell Metamodel

- Data Specification Handling - Metamodel

  - HasDataSpecification
    - dataSpecification: Reference [0..1]

  - DataSpecification
    - administration: AdministrativeInformation [0..1]
    - id: Identifier
    - dataSpecificationContent: DataSpecificationContent
    - description: MultiLanguageTextType [0..1]

- Data Specification Handling - Serialization

  - HasDataSpecification
    - embeddedDataSpecification: EmbeddedDataSpecification [0..1]

  - EmbeddedDataSpecification
    - dataSpecification: Reference [0..1]
    - dataSpecificationContent: DataSpecificationContent

- ConceptDescription

  - HasDataSpecification
    - isCaseOf: Reference [0..*]

- Data Specifications in Part 3 – data specification template IEC 61360 defined in Part 3a

  - DataSpecificationContent
    - DataSpecificationIec61360
      - preferredName: PreferredNameTypeIec61360
      - shortName: ShortNameTypeIec61360 [0..1]
      - unit: string [0..1]
      - unitId: Reference [0..1]
      - sourceOfDefinition: string [0..1]
      - symbol: string [0..1]
      - dataType: DataTypesIec61360 [0..1]
      - definition: DefinitionTypesIec61360 [0..1]
      - valueFormat: ValueFormatTypesIec61360 [0..1]
      - valueList: ValueList [0..1]
      - value: ValueTypeIec61360 [0..1]
      - levelType: LevelType [0..1]

In formats like xml, JSON, rdf the embedded data specification approach is implemented.
IEC 61360

Intended use

The vocabulary specified in IEC 61360 may be used to define ontologies for use in the field of electrotechnology, electronics and related domains.¹

Structure

The IEC 61360 series is structured into different parts:
- IEC 61360-1: Definitions - Principles and methods
- IEC 61360-2: EXPRESS dictionary schema
- IEC 61360-4: IEC Common Data Dictionary (IEC CDD)

IEC 61360-1 provides a detailed introduction to the structure of the dictionary and its use.² IEC 61360-2 specifies the detailed dictionary data model and IEC 61360-6 stipulates quality criteria for the content of the dictionary.³ The data model defined in IEC 61360-2 is also published in ISO 13584-4.³

The IEC provides a technical dictionary for the use in the electro-technical and electronic domain which is published as IEC 61360-4. This dictionary is called IEC Common Data Dictionary (IEC CDD) and can be accessed as a web page (https://cdd.iec.ch/).⁴

See also

IEC 61360 also defines the base for other product taxonomies like eCl@ss. Industrie 4.0 uses product property description based on IEC 61360.⁵

¹Note: Always be aware of usage policies and licensing before using a predefined/standardized concept description!
### IEC 61360 Property

**Example measurable Property from ECLASS**

<table>
<thead>
<tr>
<th>Preferred name</th>
<th>Max. rotation speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRDI</td>
<td>0173-1102-BAA120008</td>
</tr>
<tr>
<td>Definition</td>
<td>Greatest permissible rotation speed with which the motor or feeding unit may be operated</td>
</tr>
<tr>
<td>Short name of unit</td>
<td>1/min</td>
</tr>
<tr>
<td>Quantity</td>
<td>frequency</td>
</tr>
<tr>
<td>Type of Property</td>
<td>Non-dependent</td>
</tr>
<tr>
<td>Valency type</td>
<td>Multivalent</td>
</tr>
<tr>
<td>Definition class</td>
<td>ECLASS (0173-1101-RAA0011001)</td>
</tr>
<tr>
<td>Property data type</td>
<td>Integer (measure)</td>
</tr>
<tr>
<td>Class type code</td>
<td>F03 - frequency, rotational frequency</td>
</tr>
<tr>
<td>Allow negative values</td>
<td>False</td>
</tr>
<tr>
<td>Property Original Identifier</td>
<td>BAA1200001</td>
</tr>
</tbody>
</table>
### IEC 61360 Value List

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>short name</td>
<td>-</td>
</tr>
<tr>
<td>Format</td>
<td>STRING</td>
</tr>
<tr>
<td>Definition</td>
<td>Summary of various types of cooling, for use as search criteria that limit a selection</td>
</tr>
</tbody>
</table>

#### Example Value List from ECLASS

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0173-1907-BA686600001</td>
<td>Air-air heat exchanger</td>
</tr>
<tr>
<td>0173-1907-BA685040001</td>
<td>Air-water heat exchanger</td>
</tr>
<tr>
<td>0173-1907-BA659220001</td>
<td>alien</td>
</tr>
<tr>
<td>0173-1907-BA681110001</td>
<td>closed, external air-cooling</td>
</tr>
<tr>
<td>0173-1907-BA686104001</td>
<td>closed, internal air-cooling</td>
</tr>
<tr>
<td>0172-1907-BA689110003</td>
<td>free cooling</td>
</tr>
<tr>
<td>0179-1907-BA687024003</td>
<td>Heat exchanger against other cooling medium</td>
</tr>
<tr>
<td>0193-1907-BA685770008</td>
<td>open circuit, external cooling</td>
</tr>
<tr>
<td>0173-1907-BA685660002</td>
<td>open circuit, internal cooling</td>
</tr>
<tr>
<td>0173-1907-BA685540003</td>
<td>other form of cooling with primary air coolant</td>
</tr>
<tr>
<td>0173-1907-BA685360003</td>
<td>other primary non-air coolant</td>
</tr>
<tr>
<td>0173-1907-BA687440003</td>
<td>self</td>
</tr>
</tbody>
</table>
IEC 61360 Value

Example Value from ECLASS and ECLASS Advanced
**IEC 61360 Level Type**

<table>
<thead>
<tr>
<th>Code</th>
<th>01123/61360_4F4AE022</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>001</td>
</tr>
<tr>
<td>Revision</td>
<td>05</td>
</tr>
<tr>
<td>IROID</td>
<td>01123/61360_4F4AE022001</td>
</tr>
<tr>
<td>Preferred name</td>
<td>outside diameter</td>
</tr>
<tr>
<td>Synonymous name</td>
<td></td>
</tr>
<tr>
<td>Symbol</td>
<td>d_out</td>
</tr>
<tr>
<td>Synonymous symbol</td>
<td></td>
</tr>
<tr>
<td>Short name</td>
<td>d_out</td>
</tr>
<tr>
<td>Definition</td>
<td>Value as specified by level (millimeters) of the outside diameter of a component with a body of circular cross-section</td>
</tr>
<tr>
<td>Note</td>
<td></td>
</tr>
<tr>
<td>Remark</td>
<td>m</td>
</tr>
<tr>
<td>Primary unit</td>
<td>m</td>
</tr>
<tr>
<td>Alternative units</td>
<td></td>
</tr>
<tr>
<td>Level</td>
<td>m&lt;sub&gt;out&lt;/sub&gt;</td>
</tr>
<tr>
<td>Data type</td>
<td>LEVEL(MIN(MAX,NOM)) OF REAL_MEASURE_TYPE</td>
</tr>
<tr>
<td>Format</td>
<td>NR3.3,IEC622</td>
</tr>
<tr>
<td>Property constraint</td>
<td></td>
</tr>
<tr>
<td>Definition source</td>
<td></td>
</tr>
<tr>
<td>Value source</td>
<td></td>
</tr>
<tr>
<td>Property data element type</td>
<td>NOME_DEPENDENT_P_SET</td>
</tr>
<tr>
<td>Drawing</td>
<td></td>
</tr>
<tr>
<td>Formula</td>
<td></td>
</tr>
<tr>
<td>Value set code</td>
<td></td>
</tr>
<tr>
<td>Value set</td>
<td></td>
</tr>
<tr>
<td>DET class</td>
<td>T00</td>
</tr>
<tr>
<td>Applicable classes</td>
<td>01123/61360_4F4AE022 - component</td>
</tr>
<tr>
<td>Definition class</td>
<td>01123/61360_4F4AE022001</td>
</tr>
<tr>
<td>Code for unit</td>
<td>01123/61360_4F4AE022 - mout</td>
</tr>
<tr>
<td>Codes for alternative units</td>
<td></td>
</tr>
<tr>
<td>Code for unit list</td>
<td></td>
</tr>
</tbody>
</table>

**Example Property with Level Type MIN, MAX and NOM from IEC CDD**
The Asset Administration Shell Metamodel

Data Specification IEC 61360

DataSpecificationContent

<Template>

DataSpecificationIec61360

+ preferredName: PreferredNameTypeIec61360
+ shortName: ShortNameTypeIec61360 [0..1]
+ unit: string [0..1]
+ unitId: Reference [0..1]
+ sourceOfDefinition: string [0..1]
+ symbol: string [0..1]
+ dataType: DataTypeIec61360 [0..1]
+ definition: DefinitionTypeIec61360 [0..1]
+ valueFormat: ValueFormatTypeIec61360 [0..1]
+ valueList: ValueList [0..1]
+ levelType: LevelType [0..1]

<enumeration>

DataTypeIec61360

literals

DATE
STRING
STRING_TRANSLATABLE
INTEGER_MEASURE
INTEGER_COUNT
INTEGER_CURRENCY
REAL_MEASURE
REAL_COUNT
REAL_CURRENCY
BOOLEAN
IR
IRDI
RATIONAL
RATIONAL_MEASURE
TIME
TIMESTAMP
HTML
BLOB
FILE

ValueList

ValueReferencePair

+ value: ValueTypeIec61360
+ valueId: Reference

LevelType

+ min: boolean
+ max: boolean
+ nom: boolean
+ typ: boolean

Property

0173-1 #02-BAA120 #008
Max. rotation speed

Data type
INTEGER_MEASURE

Unit of measure
M/min

Definition
Greatest possible rotation speed with which the motor or feeding unit may be operated
Note 1: IEC 61360 also requires a globally unique identifier for a concept description. This ID is not part of the data specification template. Instead, the ConceptDescription/id as inherited via Identifiable is used. The same applies to administrative information like the version and revision.

Note 2: ConceptDescription/idShort and DataSpecificationIec61360/shortName are very similar. However, in this case, shortName is explicitly added to the data specification.

Note 3: the same applies to ConceptDescription/displayName and DataSpecificationIec61360/preferredName.

Note 4: the same applies to ConceptDescription/description and DataSpecificationIec61360/definition.
Data Specification IEC 61360

For measurable properties (dataType = *_MEASURE)

For values (within value lists)
# IEC 61360 Level Types and Ranges

**DataSpecificationContent**

- preferredName: PreferredNameTypeIec61360
- shortName: ShortNameTypeIec61360 [0..1]
- unit: string [0..1]
- unitId: Reference [0..1]
- sourceOfDefinition: string [0..1]
- symbol: string [0..1]
- dataType: DataTypeIec61360 [0..1]
- definition: DefinitionTypeIec61360 [0..1]
- valueFormat: ValueFormatTypeIec61360 [0..1]
- valueList: ValueList [0..1]
- value: ValueTypeIec61360 [0..1]
- levelType: LevelType [0..1]

For properties with Level Type

**DataElement**

- minValue: ValueDataType [0..1]
- maxValue: ValueDataType [0..1]

Use level type MIN and MAX for submodel element „Range“ (see Part 1)

It is not recommended to use standardized properties with level types except those defining ranges.
## Data Types IEC 61360

### «enumeration»

**DataTypeIec61360**

<table>
<thead>
<tr>
<th>Data Type IEC 61360</th>
<th>XSD Value Type</th>
<th>Example Values IEC 61360$^d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>xs:date</td>
<td>1979-01-15</td>
</tr>
<tr>
<td>STRING</td>
<td>xs:string</td>
<td>&quot;ON 700&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;10 Mbps&quot;</td>
</tr>
<tr>
<td>STRING_TRANSLATABLE</td>
<td></td>
<td>Mapped to MultiLanguageProperty. I.e. type MultiLanguageText</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: for details, please see Part 1 of the document series &quot;Details of the Asset Administration Shell&quot;</td>
</tr>
<tr>
<td>INTEGER_MEASURE</td>
<td>xs:integer</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>111</td>
</tr>
<tr>
<td>INTEGER_COUNT</td>
<td>xs:integer</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>111</td>
</tr>
<tr>
<td>INTEGER_CURRENCY</td>
<td>xs:integer</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>111</td>
</tr>
<tr>
<td>REAL_MEASURE</td>
<td>xs:double or xs:float (depending on needed)</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>102.35</td>
</tr>
</tbody>
</table>

Note: Data Type FILE is not explicitly supported by IEC61360 but used for data elements File
TheAssetAdministrationShellMetamodel

ValueList

For Value Lists:

The valueld references a value
## Level Type

**Class:** LevelType

**Explanation:** Value represented by up to four variants of a numeric value in a specific role: MIN, NOM, TYP, and MAX. True means that the value is available, false means the value is not available.

Note: for details, please refer to [IEC61360-1], LEVEL_TYPE

**Inherits from:** DataSpecificationContent

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Explanation</th>
<th>Type</th>
<th>Card.</th>
</tr>
</thead>
<tbody>
<tr>
<td>min</td>
<td>Minimum of the value</td>
<td>boolean</td>
<td>1</td>
</tr>
<tr>
<td>nom</td>
<td>Nominal value (value as designated)</td>
<td>boolean</td>
<td>1</td>
</tr>
<tr>
<td>typ</td>
<td>Value as typically present</td>
<td>boolean</td>
<td>1</td>
</tr>
<tr>
<td>max</td>
<td>Maximum of the value</td>
<td>boolean</td>
<td>1</td>
</tr>
</tbody>
</table>

**Range**

- value: ValueDataType [0..1]
- min: ValueDataType [0..1]
- max: ValueDataType [0..1]

Example from [IEC61360-1]: in case of a property which is of the LEVEL_TYPE min/max – Note: for details, please refer to [IEC61360-1], LEVEL_TYPE
Create your first digital twin
AASX Package Explorer

https://github.com/admin-shell-io/aasx-package-explorer
The Asset Administration Shell Metamodel

Life Cycle Phases and Formats

- Technology neutral / independent
- AutomationML
- XML/ JSON
- OPC UA
- RDF
- Information security
- Package-format

Meta-level

Information-Representation

Important in the life cycle

- Engineering
- Business process
- Operation and maintenance

whole life cycle
Serializations/Formats

Note for Experts: Mapping Rules and Schema for xml, JSON and rdf as well as examples not part of specification any longer ➔ now part of open source project admin-shell-io/aas-specs

Note for Experts: Formats like OPC UA or AutomationML are maintained in OPC Foundation and Automation e.V.

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

Note: see Readme files for different mappings to XML, JSON and RDF

Note: for data specifications the embedded approach is used
Open Source Support

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

Status: September 2023
Still Questions?
Questions and Answers

For this reading guide the documents have been sorted by interest groups rather than topics. In some cases, only specific pages or sections are 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 4.0
- **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.

Asset Administration Shell Frequently Asked Questions List

https://github.com/admin-shell-io/questions-and-answers
Let’s go!
Dr. Birgit Boss
Robert Bosch GmbH, Bosch Connected Industry

- Board member of the Industrial Digital Twin Association (IDTA)
- Chair of the Working Group “Open Technology” and its Working Stream “Specifications of the Asset Administration Shell”
- Chair of the Working Group “Semantic Layer including Digital Twins” of Catena-X
- PMC member of the Eclipse Digital Twin Top Level Project

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