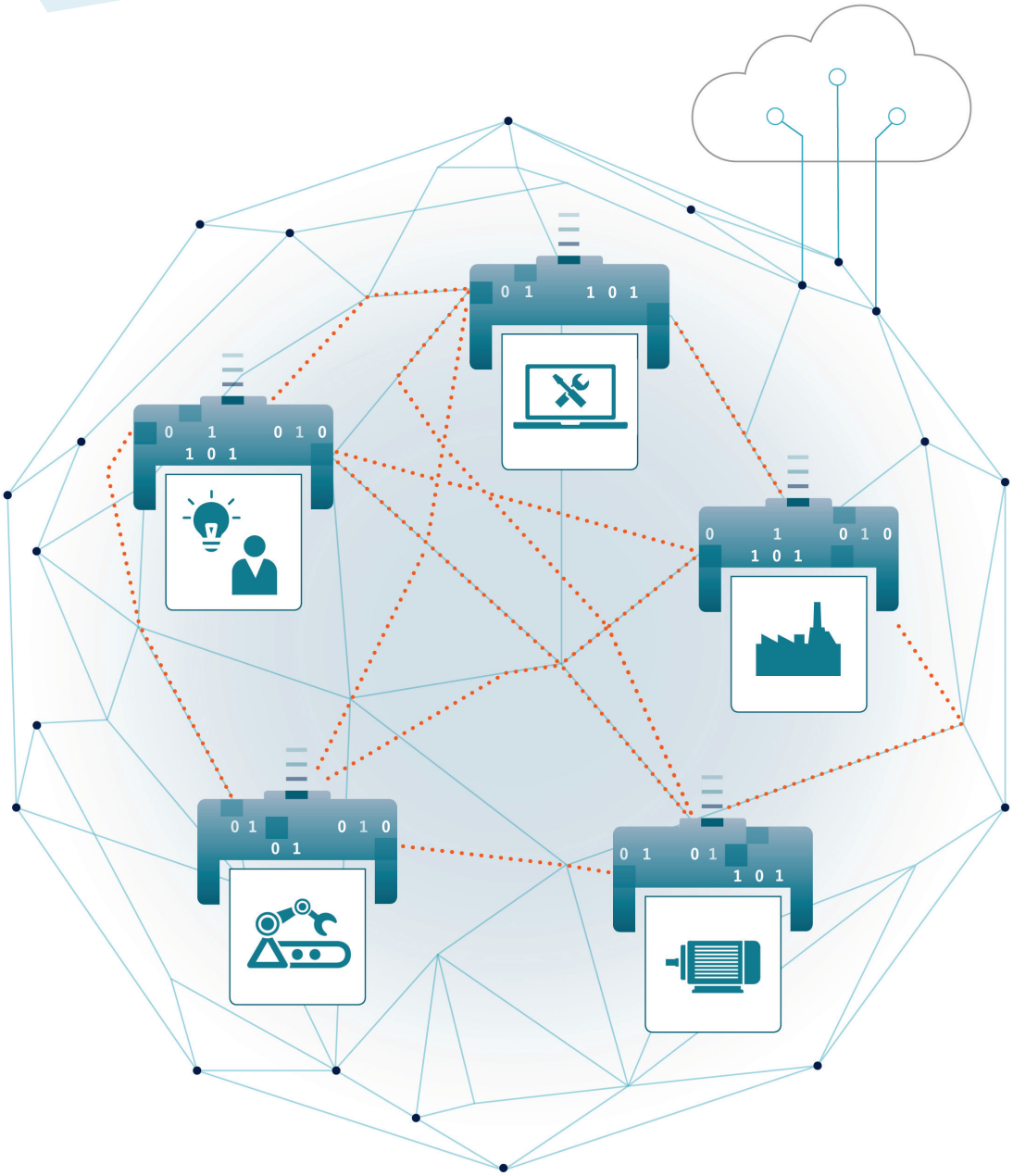
PLATTFORM INDUSTRIE4.0

Discussion paper



AAS Reference Modelling

Exemplary modelling of a manufacturing plant with AASX Package Explorer based on the AAS metamodel

Imprint

This document is a draft, generated by translation of the paper “VWS-Referenzmodellierung - Exemplarische Modellierung einer fertigungstechnischen Anlage mit AASX Package Explorer auf Basis des VWS-Metamodells“, published by the Plattform Industrie 4.0 in April 2021.

Publisher

Plattform Industrie 4.0
Bülowstraße 78
10783 Berlin
geschaeftsstelle@plattform-i40.de

Editing, layout and photos/graphics

Plattform Industrie 4.0
Bülowstraße 78
10783 Berlin

Status: April 2021

Content

(1) Aim and procedure of the document

The goal is to systematically use the modelling means of the Asset Administration Shell's (AAS) metamodel [PLA20] so that the quality of the AAS modelling elements can be tested. This document aims to provide both a guide on how to sort asset data in the AAS (cookbook for AAS configurations) and a guide for users of AAS who use the data in the applications. In addition, it is intended to be a reference work for finding exemplary applications of the elements of the AAS metamodel.

This document can only provide guidance, as there are many ways to use the model elements and it is difficult to maintain consistency with the evolving AAS specification.

This document shows best practice patterns for the implementation of AAS of concrete assets and therefore addresses the following user:

- Product managers
- Developers who have to implement the details in the configuration of the AAS
- Machine builders who have to deal with the information of the supplier assets and their documentation
- Developer of technical documentation

The document is structured as follows:

Chapter 2: A brief description of the demonstrator with its automation architecture is presented. Components of the demonstrator are described as AAS in two ways and the application of AAS metamodel is explained in further chapters using this exemplary AAS.

Chapter 3: AAS should be able to provide information about an asset over its entire lifecycle. It may well be that different AAS users or AAS interaction partners have different views of an asset and expect different representations of the asset in its AAS. To illustrate the diversity and strong expressiveness of the AAS metamodel, three AAS interaction partner roles are introduced as examples in this chapter. The defined AAS interaction partner roles do not claim to be generally valid and complete. In the context of this document, they serve to better illustrate AAS modelling concepts.

Chapter 4: Modelling of different AAS interaction partner-related aspects (see Chapter 3) of the selected assets (see Chapter 2) as AAS submodels. No explanation of the elements of the AAS metamodel is given. It is only shown which AAS metamodel elements were used in the respective submodels. Reference is made to the AAS metamodel and explanations are given in chapter 5.

Chapter 5: Includes explanations of the AAS metamodel and its use in mapping the facts of the physical world into the information world.

Chapter 6: Contains an alphabetical list of important model elements used in the previous chapters. References to the corresponding subchapters in chapters 4 and 5 are provided in each case.

Chapter 7: Summary and outlook

Version 3.0 RC01 of the AAS metamodel is the basis of the model elements of the metamodel used here in the textual descriptions [PLA20].

All AASs are provided in the AASX exchange format of the open-source tool AASX Package Explorer. At the time of writing the examples, Package Explorer had not yet correctly implemented all the details of V3.0 RC01 metamodel. Therefore, there may be deviations from the textual descriptions. Since this is a matter of basic understanding, any inconsistencies should be tolerated.

(2) Subject of the modelling

The Pick and Place Station (P&P station) at Otto von Guericke University Magdeburg (OvGU) is used as an example for the modelling.

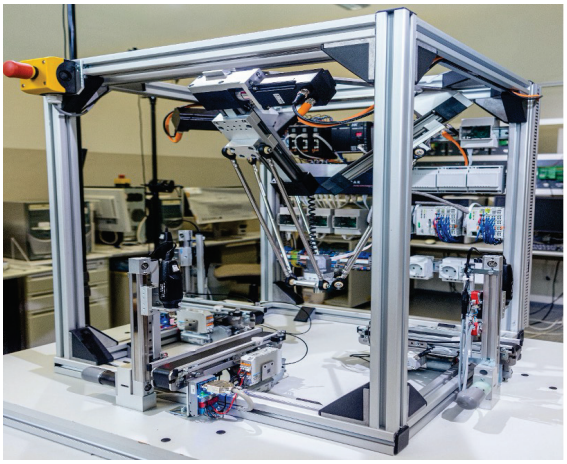


Figure 0-1: Pick and Place demonstrator of the OvGU

Figure 2-2 presents the automation architecture of the P&P station. The demonstrator consists of the three processing stations (workstation) as well as the PLCs for the delta robot and the processing stations. The delta robot includes three motors with their respective motor controllers. Several products can be processed by the P&P station. Each controller controls the respective station or motor. They are all connected to an AAS. Each product also has an AAS. The AAS interact with each other using the I4.0 language and organize the process in the P&P station.

This document will show how selected components (assets) of this demonstrator are converted into I4.0 components by incorporating them with an AAS of their own. The assets mentioned are represented in an I4.0 system by the corresponding AAS. In the process, relevant information about these assets is mapped in the information world as submodels according to the AAS metamodel and the mapped information in the corresponding AAS is made available to the potential AAS users.

First, the entire demonstrator as a whole is incorporated with an AAS of its own. Since the P&P station is quite complex and consists of components that can possibly be considered as independent I4.0 components, some selected components are incorporated with an AAS of their own. The AAS of the P&P station contains the references to the individual AAS of the selected components.

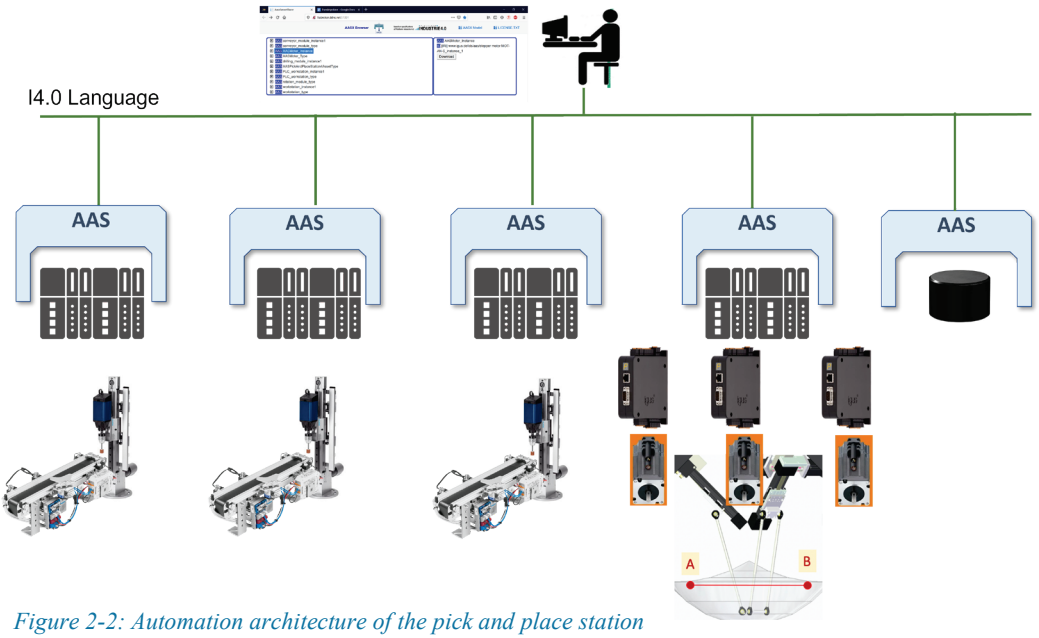


Figure 2-2: Automation architecture of the pick and place station

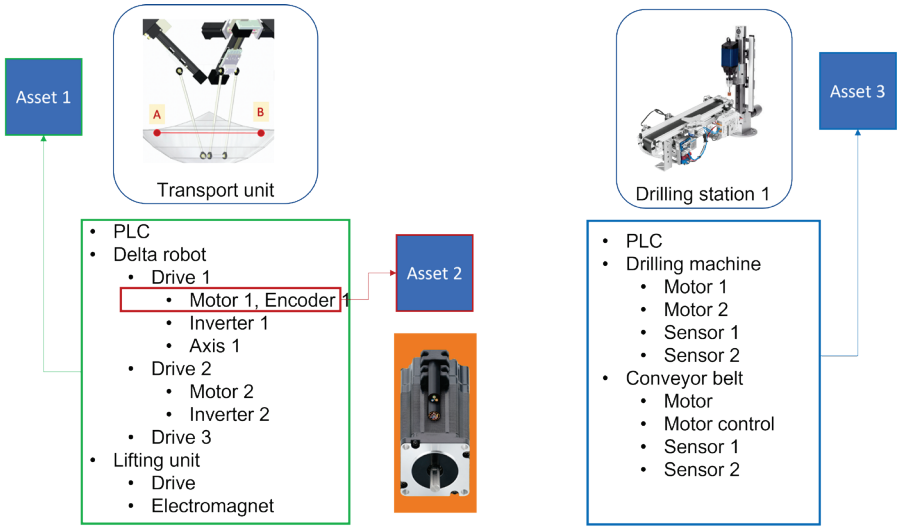


Figure 2-3: Selection of assets to be provided with their own asset administration shells in this document (for refinement see chapter 3.2)

This is to demonstrate the implementation of the principle of the nested/ composite administration shell (composite component).

For the exemplary modelling, the transport unit and the processing station were selected as independent subcomponents of the P&P station. The transport unit consists of a delta robot, an electromechanical linear actuator and associated control units and is considered as a single asset in the modelling. Similarly, a workstation is considered as a single asset, consisting of a conveyor belt and a drilling machine.

The examples of these assets are intended to show the expressiveness of the modelling capabilities of the AAS metamodel, various levels of detail and modelling techniques in creation of submodels and AAS. This should be clarified by the introduction of different AAS interaction partner roles, which require submodel contents tailored to these roles. Likewise, it is intended to illustrate the flexibility of the AAS metamodel with respect to the classification of AAS content for different stakeholders, e.g. by using the concept of View.

In addition, the concept of technology-neutral, consistent and interoperable description and access to information is demonstrated using the example of modelling an electric motor.

Overall, the following assets are selected for modelling in this document:

- Asset 1: The entire transport unit
- Asset 2: Motor with encoder of the delta robot
- Asset 3: The entire workstation
- Asset 4: The entire P&P station

These assets are incorporated with their own AAS. The submodels created during the modelling of the AAS do not claim to be technically correct and complete. The complete and consistent submodels for the representation of various aspects of assets considered in this document are currently being developed in working groups and committees of the respective expert organizations. Where possible and according to the authors' state of knowledge, reference is made in the document to the respective work.

(3) Modelling method

In this chapter, specifications are made for the general procedure for modelling and creation of AAS for the exemplarily selected assets of the demonstrator. It has to be pointed out once again that this is an exemplary

application of the modelling possibilities of AAS. There is no normative claim. Nevertheless, the intention is to offer good practical example solutions.

AAS interaction partner role description

AAS are intended to provide information about an asset throughout its entire lifecycle. It may well be that different interaction partners of AAS have different views of an asset and expect different representations of the asset in its AAS. Depending on this, different features and submodels may be relevant. To reflect the diversity and strengths of the expressive capabilities of the AAS metamodel, this chapter introduces three roles of AAS interaction partners. First, tools for planning and commissioning a plant in which the respective asset is to be used are seen as possible interaction partners. Service requesters and service providers were introduced as the third interaction partner role. Since it is conceivable that some components of the P&P station can be considered as autonomous service providers in certain I4.0 application scenarios, such as transport unit and workstation as providers of transport or drilling services. To characterize these AAS interaction partner roles, the eventual requirements for the AAS content from the perspective of these roles are specified. The definition of AAS interaction partner roles, the selection of submodels, and their contents are not of a universal nature and only apply in the context of this document.

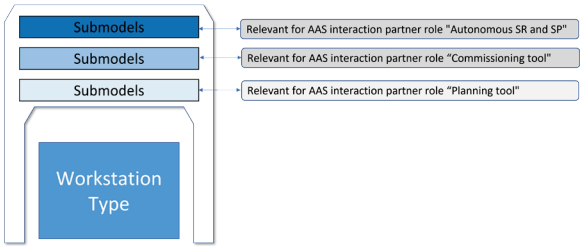


Figure 3-1: AAS interaction partner roles

The planning of a complex plant plays a prominent role within the life phases of a plant. The planner of a plant or the plant designer creates a specification sheet in which the basic requirements for the properties of the planned plant are defined.

Manufacturers of components create a technical data description of their components. This description is made available using Industry 4.0 concept in the form of AAS with its corresponding submodels.

The comparison of the requested and assured properties of components, or the verification of the fulfilment of the required properties, in the sense of I4.0, is to be as automated or semi-automated as possible with the help of a planning tool. This requires a consistent and interoperable description of the properties in the AAS of the asset type in the form of corresponding submodels of the components. This description should be read in and correctly interpreted by the planning tool (Figure 3-2).

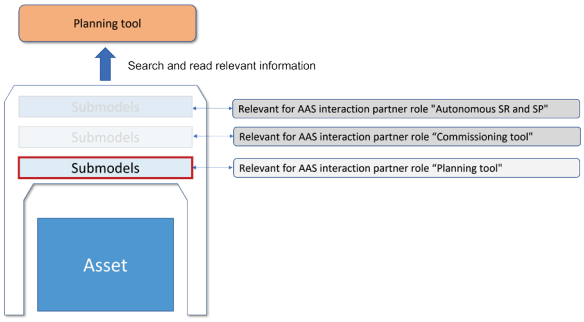


Figure 3-2: Selection of submodels for the plant design tool

- Technical data for the planners of a system
  - Electrical, pneumatic, hydraulic and mechanical interfaces
  - Operating conditions (e.g. climatic, operating temperatures)
  - Functional inputs and outputs
- Assembly instructions

- Maintenance management

AAS Interaction Partner Role “Commissioning Tool”

An AAS can also provide a technology-neutral interoperable interface of an asset that allows coupling of the asset with engineering and control software for testing and commissioning.

The AAS, as an asset interface, should enable the commissioning engineer to parameterize the asset and carry out a functional test of the asset. For this purpose, it should be possible to call up and control the functionalities of the asset (Figure 3-3).

To prove the required process capability of the plant, it should be possible for the commissioning engineers to check the asset functionalities and read the relevant sensor data. This data can be used, for example, to create characteristic curves that provide further information about the functionality of the plant.

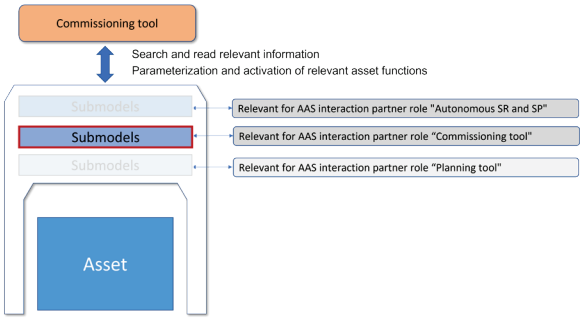


Figure 3-3: Selection of submodels for the commissioning tool

- Nameplate
- Documentations
- Technical data for commissioning engineers
  - Measuring and control ranges
  - Electrical and communication interfaces
- Submodels for functional data
  - Actual data of measured and control values, e.g. "motor current", "position"
  - Setting parameters of the devices and components
  - Status and diagnostic values about the plant and about the devices and components

AAS Interaction Partner Role "Service Requester and Service Provider”

A semantically unambiguous machine-readable description of the properties and capabilities of assets, as provided by the submodels for planners and commissioning engineers, is an important step towards increasing the interoperability and integration of automation technology components. The area of application is the production site of the operator, who wants to include the actual production capability of the asset in his production and work plans and thus aims for easy integration in ERP or MES systems.

However, certain I4.0 application scenarios require a certain autonomy of the production systems and their components, e.g. dynamic optimization of production utilization, avoidance of downtime due to machine failures, order-driven production as well as the dynamic orchestration of production resources for cost-optimized production in lot size one. To this end, the submodels of the AAS should be able to provide information about the asset capabilities or the services offered by assets to the interaction partners.

In such scenarios, the assets become autonomous service providers or service requesters (Figure 3-4). The AAS enable participation in horizontal protocol-based interactions, such as those defined in the VDI/VDE 2193 - guideline for the bidding procedure.

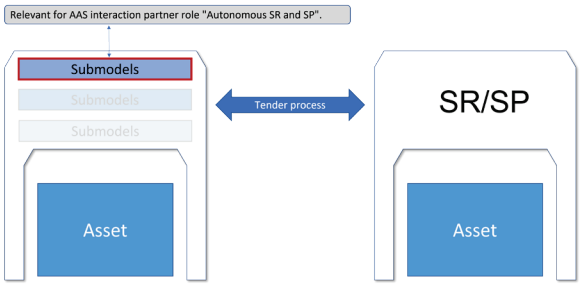


Figure 3-4: Selection of submodels for the autonomous AAS

- What purpose does the asset serve?
- What are the capabilities of the asset and what services can the asset/ I4.0 component offer?
- Technical data for detailed verification of the requirements (see planner)

Specifications on the general procedure for modelling

This document describes how a P&P station (the basic structure shown in Figure 2-2) can be digitally represented in an I40 system and can be incorporated with its own AAS. The following assumptions and requirements built the structure of the P&P station's AAS:

- I4.0 components consist of an asset and its AAS. Assets can be described by their own AAS or in submodels of an AAS in which this asset is a component. This is described by SelfManagedEntity and CoManagedEntity. This is exampled here in a Bill of Material (BoM).
- Some AAS of a "self-managed" asset have a BoM (submodel with semanticId BoM)
- The framework with its mechanical parts is described as a submodel of the P&P station AAS and does not have its own AAS.
- The components used in this document are the components of the P&P station, or the components of the assets in general.
- The assets are provided with one or more AAS, which are considered as a stand-alone component. The idea behind this is that the stand-alone component also contains a consistent description in itself, e.g. provided by the supplier. These assets are called "Self-ManagedEntity" (5.5).
- The entire P&P station is considered as an asset and receives its own AAS. The AAS of the P&P station contains references to the AAS of the other modelled assets as a so-called composite component in a special submodel.
- Assets are understood to be the automation components actually built into the P&P station. Accordingly, the AAS of these assets are understood as the AAS of asset instances.
- Additionally, it is assumed that asset types exist. For the sake of completeness, selected AAS of the corresponding asset types are modelled in this document in addition to AAS of the asset instances. AAS is a digital representation of an asset. In different life cycle phases, this digital representation is used by different interaction partners.
- To enable the different interaction partners to identify the submodels that are relevant to them, the concept of "view" for structuring AAS content is introduced as an example. Views can also be used to group information that is relevant to the asset type and the asset instance. Then both the aspects can be contained within an AAS.
- Each AAS has a submodel for Nameplate, Documentation and TechnicalData.

The Figure 3-5 shows an excerpt of a possible mapping of the selected components of the P&P station in AAS. In principle, a distinction is made between the asset type and

the asset instance. The asset type is described by the catalogue data and other planning documents, which do not yet have any specific information about the individual device or component (e.g. the possible voltage supply variants).

Figure 3-5 shows an excerpt of the P&P station only (PandPtype) which is consisting of the delta robot (DeltaRobot), the workstation and the motor. Each asset type is represented by an AAS (PaP: AAS), which is supplied, for example, with the sale. Asset types considered include the P&P station itself (PaP1: PandPtype), a delta robot (Robot1: DeltaRobot), a workstation (Workstation1: Workstation, ...) and a motor (Motor1: Motor, ...). There are relationships between asset instance and AAS of an asset instance and AAS of an asset type which are described in more detail in chapter 5.6. These are described here as links (between the asset instances and the AAS - solid line) and as dependencies (between the asset types and AAS - solid line).

Figures 3-6 to 3-8 show the selection of AAS described in this document as examples.

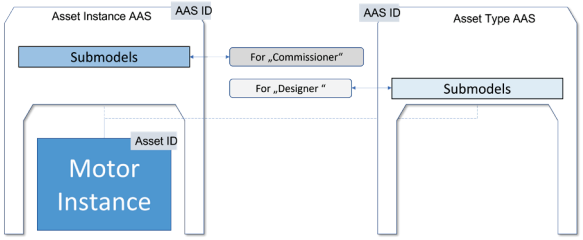


Figure 3-6: The submodels created in chapter 4 for representing the motor within the AAS

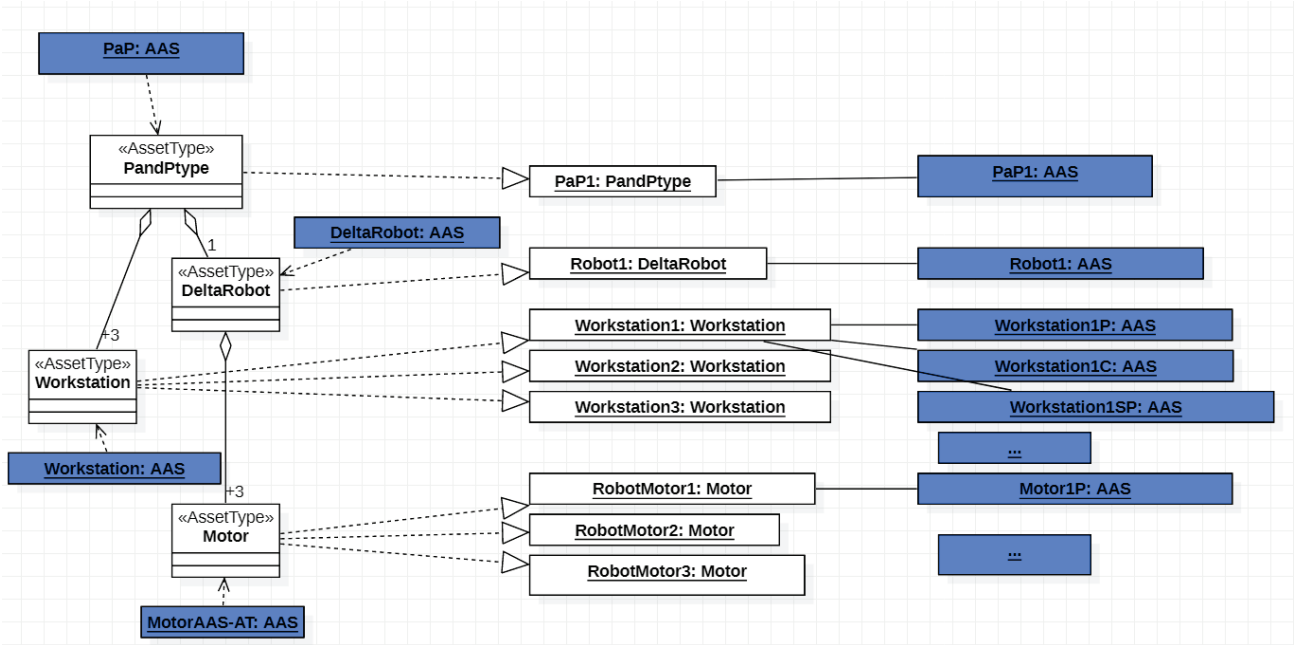


Figure 3-5: Overview of the AAS modelling of the P&P station (detail)

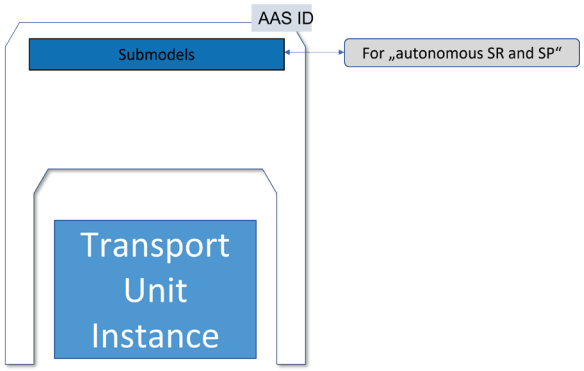


Figure 3-7: The submodels created in chapter 4 for mapping the transport unit

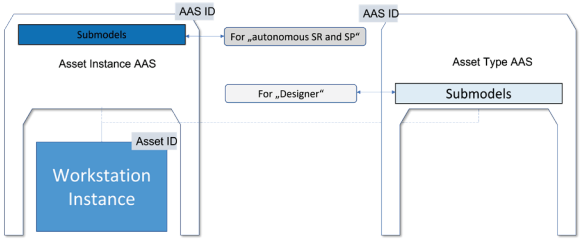


Figure 3-8: The submodels created in chapter 4 for mapping the workstation

For the modelling of composite assets two different approaches are used here. This is to illustrate the scope for design.

Figure 3-9 shows how further information on individual components of the motor (here the encoder) can be described in a submodel of the assembled asset. This makes sense if the individual parts are permanently connected to each other and the asset (here the motor) is usually only purchased assembled. This modelling approach also allows the motor manufacturer to include information about the encoder (e.g. the positioning accuracy) without passing on its AAS to the next value creation partner. The different subcomponents are described in separate submodels. This makes sense if the component can only be acquired as a whole. The description can be found in chapters 4.3 and 4.4.

Figure 3-10 shows another approach. Here the subcomponents are equipped with their own AAS. This makes sense if the subcomponents are purchased separately and assembled at the user's site. The description can be found in chapters 4.6 and 4.7.

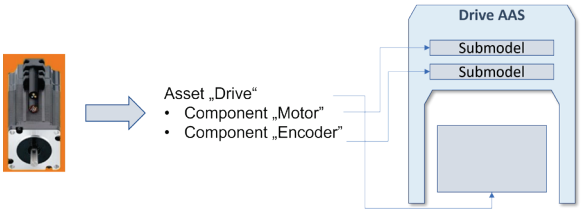


Figure 3-9: Built-in assets of the I4.0 component "Motor with integrated encoder" are modelled as submodels in the AAS

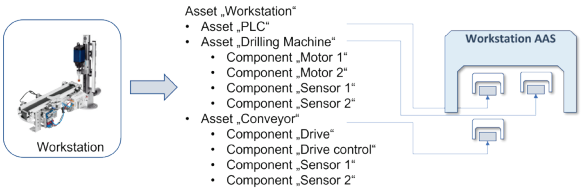


Figure 3-10: AAS of the workstation refers to the AAS of Built-in Assets of the workstation

Use of the AASX Package Explorer for illustration purposes

The modelling presented are generally to be understood as examples that illustrate the modelling elements and possibilities. They are not specifications. Therefore, different modelling options are sometimes used in different examples.

For illustration purposes, all examples are created with the AASX Package Explorer in the current available version. The AASX files can be viewed at http://liabroker.ddns.net:51001/.

Not all exemplary implementations in Package Explorer currently comply with version 3.0 RC01 of the AAS specification. Nevertheless, the authors believe that the AASX models and the resulting screenshots support understanding.

(4) Modelling of the selected components of the P&P station

This chapter describes the exemplary modelling of the assets selected in chapter 2 in the form of AAS and submodels. The modelling presented is based on the modelling method described in chapter 3.

P&P station from a planning perspective

The demonstrator is a P&P station, in which three workstations offer different capabilities. A delta robot handles the transport tasks between the workstations. A planner first has a rough view and considers the delta robot and the workstations as separate components. In this respect, the P&P station is an I4.0 composite component (Figure 4-1). A Bill of Material (BoM) lists all the components.

This is explained in more detail in chapter 5.6. As specified for this modelling (chapter 3.2), the AAS of the P&P station has the following submodels Nameplate, Documentation and TechnicalData. The Nameplate and Documentation submodels are based on the corresponding submodel templates. More detailed information can be found in [DNI20] and [DD20]. In these submodels, the station is described as a whole, such as the electrical connection values and the geometric dimensions. However, the planner also needs the technical details of the components. Therefore, the AAS of the P&P station has relationships to the AAS of the components. These are stored in a separate submodel "CompositeRelationship" (more detailed descriptions in chapter 5.8).

Via the entry point of the P&P station AAS, all components belonging to the station as well as their data can be found. They contain the type-related data of the components.

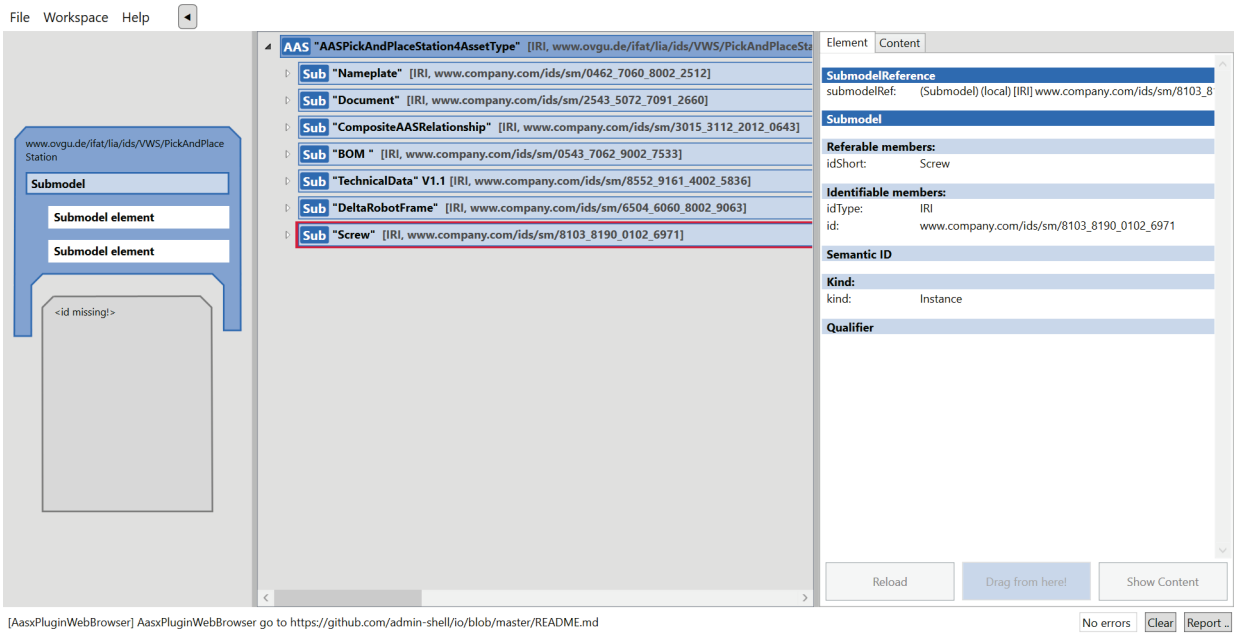


Figure 4-1: Submodel overview P&P station type

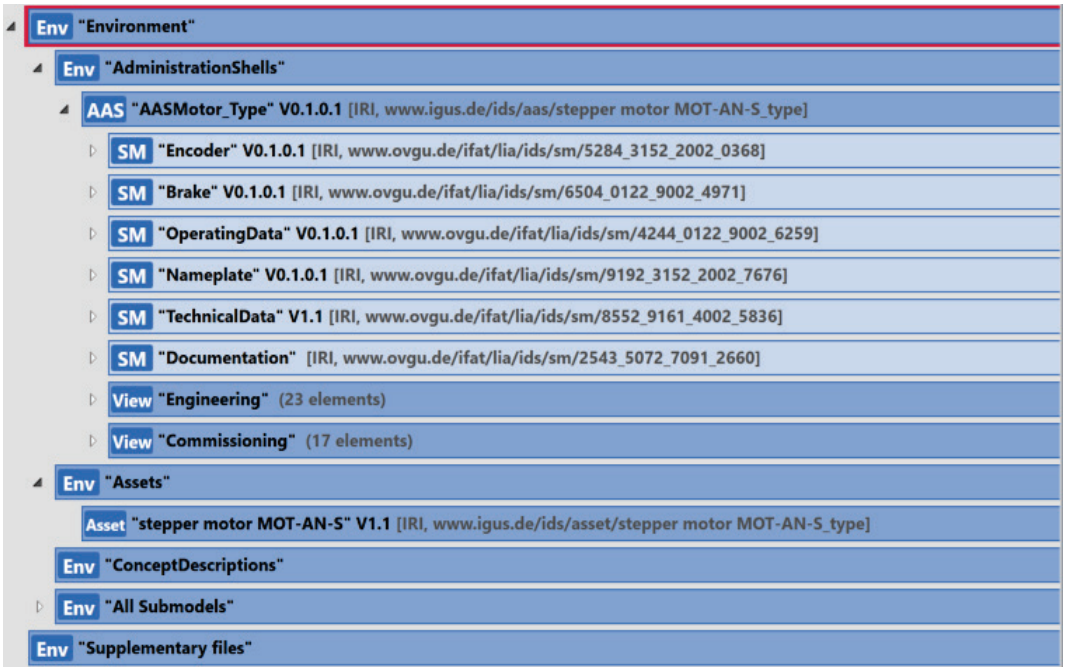


Figure 4-2: View of the AAS of the motor type

P&P station from a commissioning perspective

Once the design work has been completed and the components have been ordered and delivered, the assembly and commissioning must be carried out. The AAS of the P&P station instance refers to all instance-related AAS, in this example on three workstations and one delta robot. As described in the following chapters, the submodel entries of the I4.0 components differ between those describing the asset type and the asset instance. The AAS of the P&P station instance uses the AAS attribute "derivedFrom", in which the AASId of the AAS of the P&P station type is entered. "derivedFrom" does not describe a classic object-oriented type/instance relationship.

Here it is only expressed that the asset instance AAS takes over elements from the asset type AAS once during instantiation or makes them accessible by reference to the data of the other AAS. In the process, the elements of the AAS can be changed, shortened or supplemented.

Motor type from the perspective of planning and commissioning

The demonstrator contains three stepper motors for controlling the Delta robot. All three motors are manifestations of the same product type. Therefore, an AAS is modelled for this type. Figure 4-2 shows the representation of the AAS of the motor type in the AASX Package Explorer.

An asset and an associated AAS are created. The asset is of the AssetKind Type, as it represents the type of the motor. The AAS contains six submodels, the choice of which is based on the structure of the motor's documentation [IGUS03]. All IDs of the submodels are preceded by the domain www.ovgu.de.

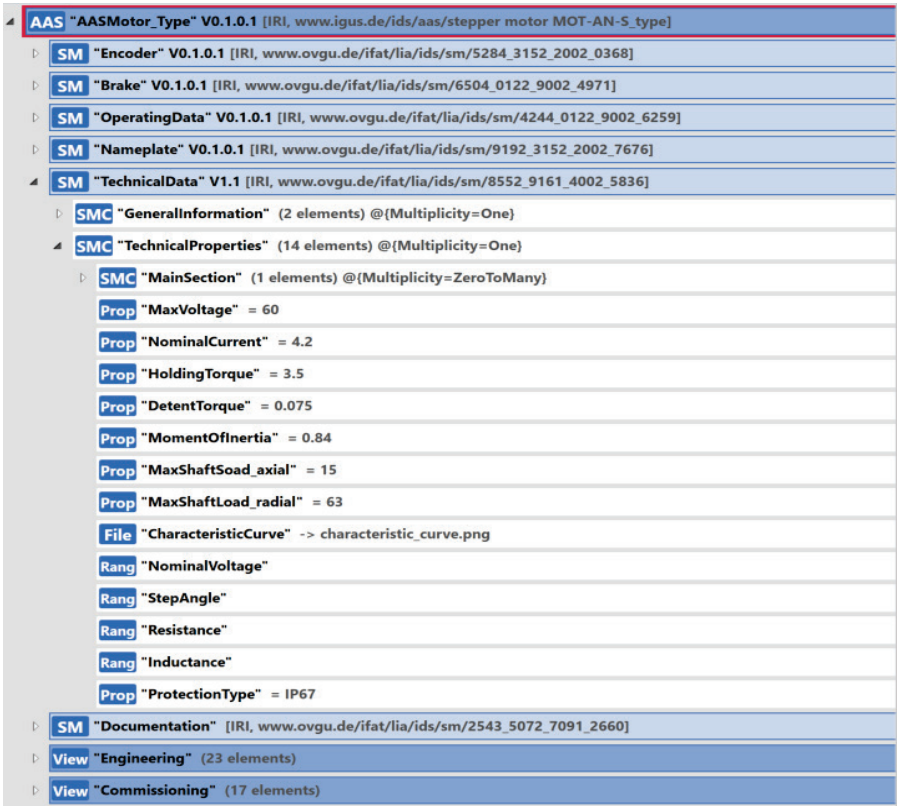


Figure 4-3: Submodel TechnicalData with different SubmodelElement types

All submodels here are from the Modeling-Kind Instance. It should be noted that the submodeElements inherits its kind from the submodel modelling kind. The submodel elements of the submodel of the kind template are also from the kind template. All SubmodelElements carry semanticIds that refer as a GlobalReference of the type IRDI or IRI to a standard such as ECLASS or IEC61360-CDD.

Figure 4-3 shows an example of the TechnicalData submodel where different subtypes of SubmodelElements are used. In addition to the frequently used property type, this submodel also contains the types range and file in order to assign a range to the value attribute of a property or to refer to a file. The property with the identifier protection type also uses a ValueID to refer to a standardised value from a ValueList (e.g. IP65, IP67). Individual submodel elements are assigned to one of the two views "Engineering" or "Commissioning". Different user roles can thus specifically access the submodel elements they consider relevant.

Motor instance from the perspective of planning and commissioning

Figure 4-4 shows the view of the AAS of a motor instance in the AASX Package Explorer. There are in total three of these. An asset and an associated AAS have been created. The asset is from the ModelingKind Instance, as it represents an instance of the motor. The AAS now contains only five submodels, since with respect to submodel nameplate none of the Modelling Kind Template is now required. Instead, all submodel elements of Nameplate have now concrete values and are therefore combined in a single submodel from the Modeling-Kind Instance.

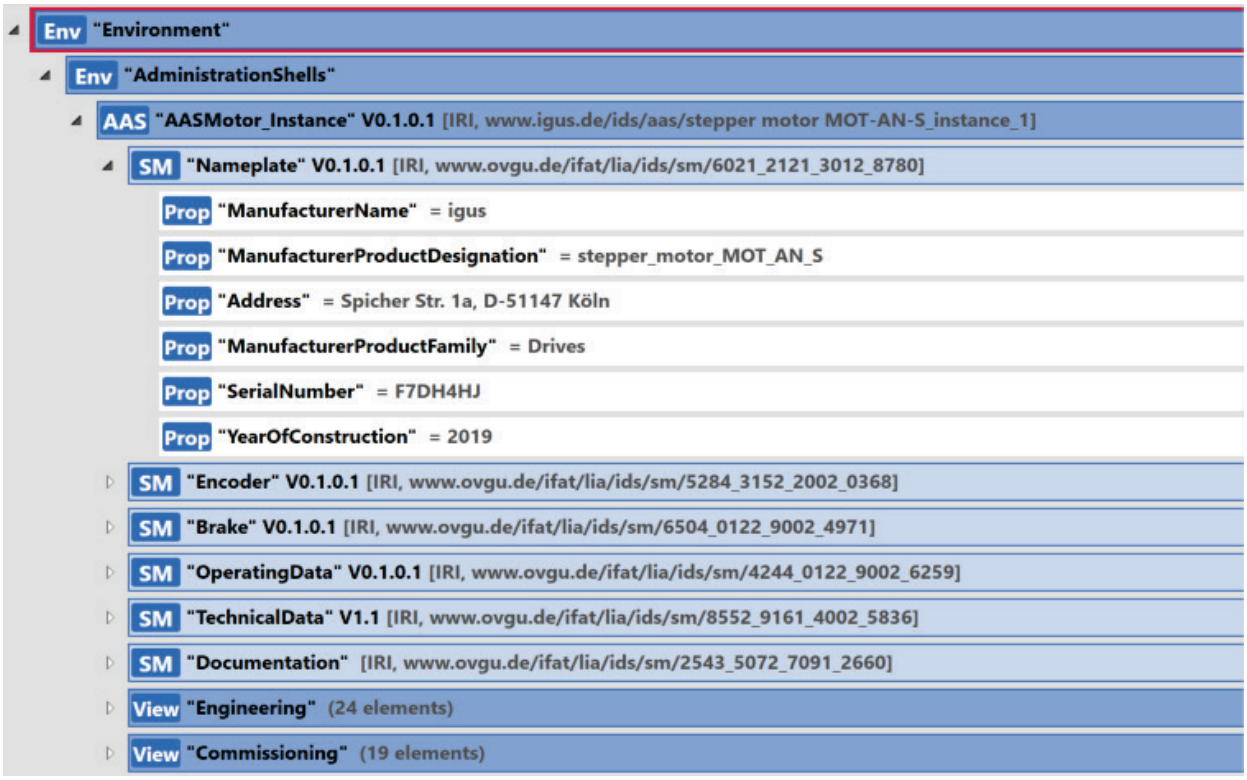


Figure 4-4: View of the AAS of a motor instance

The instance AAS also contains the two views "Engineering" and "Commissioning". Figure 4-5 shows an example of the "Commissioning" view. The references to submodel elements of the various submodels can be seen. The assignment to the views is regarded as exemplary and will vary depending on the intended use of the asset and its AAS. The metamodel itself cannot provide criteria for this. In the other aspects, the instance AAS does not differ from the type AAS described above.

Figure 4-5: View of the AAS of a motor instance with the "Commissioning" view expanded

AAS of the transport unit: submodels for SP/SR

In the previous chapters it was explained how the features, parameters and states of assets can be modelled with properties as submodels. The next challenge is to describe what the assets are suitable for in order to be able to define their use precisely. For an asset to be integrated into a CPS, this object must provide relevant information and description of its capabilities in its AAS. In I4.0 systems, some I4.0 components can support contractual negotiations to bring in production services. In a service system applied in Industrie 4.0, some I4.0 components can be understood as service providers or users. The information about the available services and provided functions must be made available in such a way that all interaction partners understand them in the same way. This means that the semantics of the service and function description must be unambiguous.

In today's discussion in the I4.0 community, the terms "function", "functionality", "capability", "skill", "operation", "process" and "service" play a central role in describing the effect of an asset. However, the authors are not aware of a concept for a clear definition of these terms.

In this document, terms "function", "functionality", "capability" are seen as synonyms and are modelled with the AAS metamodel element "capability". The terms "service" is used for both the ability to provide a certain offer to a client as well as the access to these offer. If, for example, a service is to be modelled with properties, a capability submodel element can be used for this purpose. The properties can either be in the submodel in which the capability submodel element is entered or the properties are in another submodel that is linked by reference from the submodel to the capability submodel element by reference form the submodel.

A capability submodel element describes the ability of an asset or component to perform a specific function. The feature sets for describing different capability submodel elements are function-specific and can be provided for each technical function in product description standards (e.g. ECLASS).

As shown in Figure 4-6, the asset considered in this chapter is a transport unit. The task or expected effect of this asset is to transport a work piece from A to B within the P&P station.

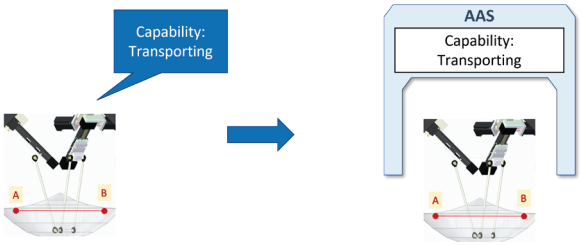


Figure 4-6: If the transport unit in the I4.0 system acts as an autonomous service provider, its capabilities and offered services are described in the AAS.

Accordingly, the ability of the transport unit transport things from A (start position) to B (destination position) is to be modelled with the capability concept [PLA20].

Figure 4-7 shows an exemplary modelling approach. In this example, the "Capability" submodel contains all the capabilities that an asset has. In this case, the capability in question is "Transport". SemanticId of the capability "Transport" refers to the corresponding semantic description in the ECLASS standard. The capability modelling concept, described in the relevant white paper of the I4.0 platform [DCI20], establishes the basis for a second submodel that contains the properties for the description of the asset capability.

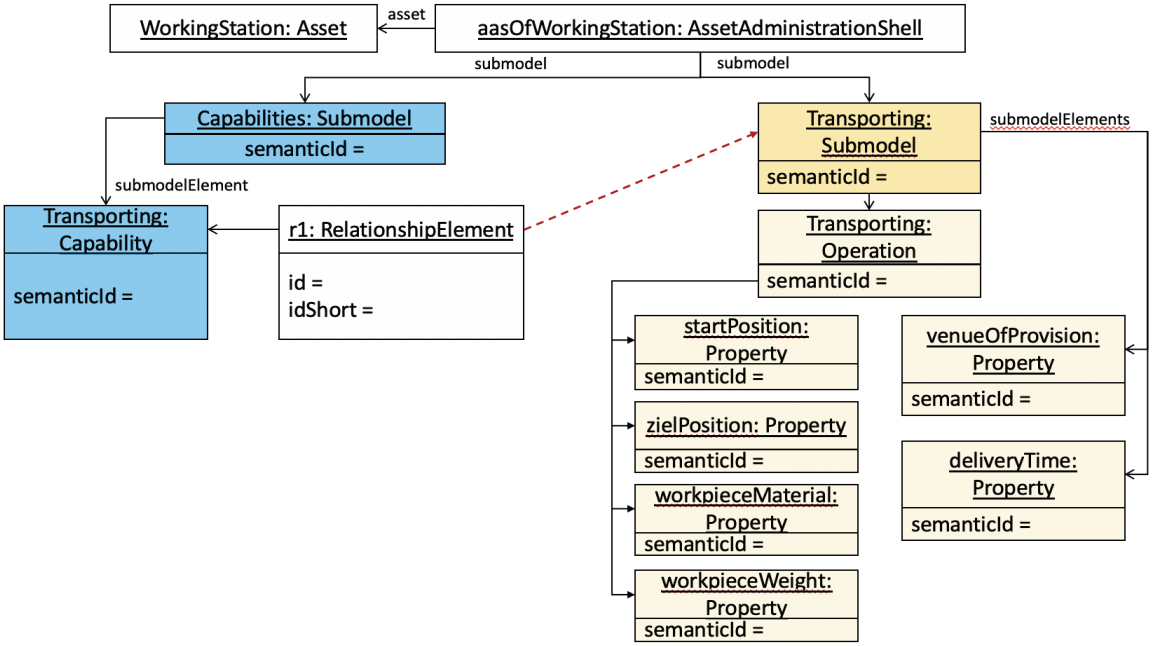


Figure 4-7: Description of the capability "Transport" based on the concept of capability description in the common white paper Platform Industrie 4.0 and BaSys 4.2 project [DCI20].

The modelling concept provides for a structuring of some features for the individual operations in order to facilitate the execution of the capability. In this example, these include start and target position, material and weight of the transporting work piece. If a service is to be modelled, the service-describing (commercial) characteristics can be added to the submodel in addition to the function-describing (technical) characteristics (e.g. price, place and time of the provision of a service).

The concepts of AAS capability and AAS operation presented in the capability white paper [DCI20] can also be interpreted differently, so that an alternative simplified variant of the capability description could be imagined. The capability description shown in Figure 4-8 does not require the use of operations in the modeling. The properties assigned to the operation in the previous example are directly assigned to the "Transporting" submodel in the alternative proposal.

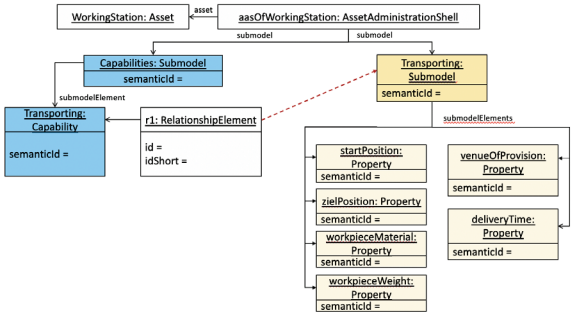


Figure 4-8: Alternative way to describe the "Transport" capability

Workstation type from a planning perspective

The AAS concept supports the aggregation of I4.0 components into a new, higher-level I4.0 component [PLA17]. As a result, there are different ways to model the asset and AAS structure of the processing station (workstation).

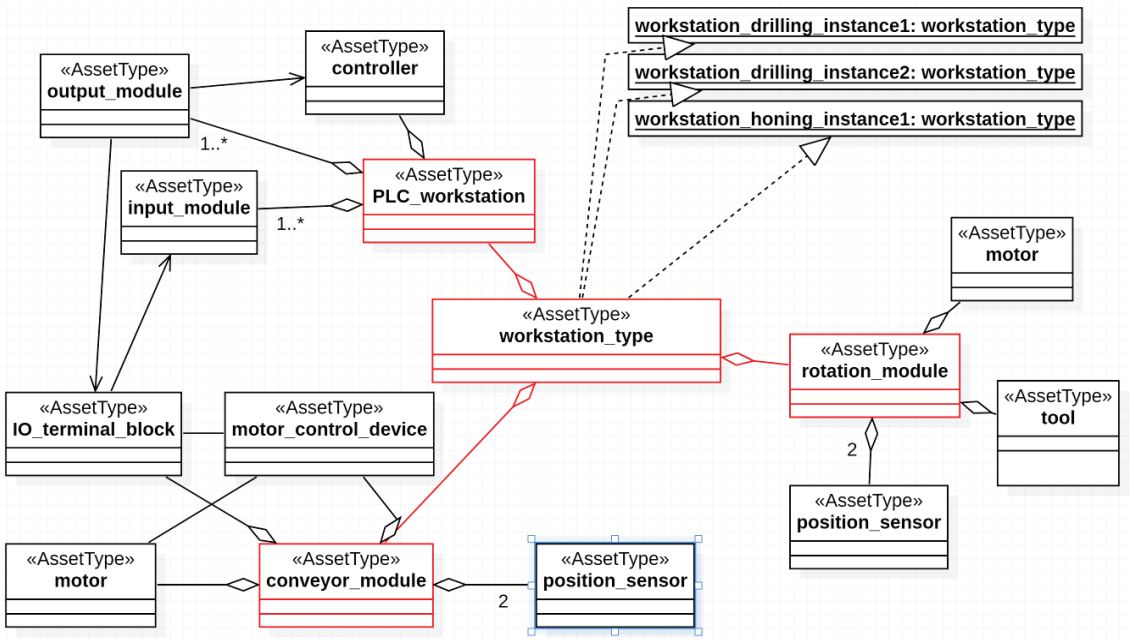


Figure 4-9 - Workstation class diagram

In this section, the components that make up the workstation are each individually modelled as a separate AAS (cf. Figure 4-9). The workstation is then an asset that is composed of various sub-assets, such as the controller, conveyor belt and drilling module.

This is quite conceivable in practice because the workstation is constructed from a set of different supplier components. In a possible future scenario, the AAS of the corresponding components can be supplied by the suppliers and integrated into the AAS of the workstation by its manufacturer (or system integrator), in this case OvGU.

Three instances of the AssetType workstation_type are built into the P&P station. According to RAMI 4.0, it makes sense to separate the product life cycle of assets according to type and instance information (see section 5.6). This section shows the modelling of an AAS for the workstation as an AssetType. A proposal on how to derive the AAS of an AssetInstance from the AAS of an AssetType is described in section 4.6.

In addition to the PLC (PLC_workstation), the workstation_type further consists of a conveyor (conveyor_module) and a rotation module (rotation_module). Furthermore, these assets consist of another sub-assets, such as actuators and sensors. Due to the effort involved, only the sub-assets of the first level are modelled (cf. classes and aggregation relationships marked in red in Fig. 4-9).

The relationships for this I4.0 component are assigned in a separate submodel. This submodel contains relationships between the assets that belong together in the implemented example (marked red in Figure 4-10) workstation_type. The relationships are entered at the asset level (see Figure 4-10) and not at the AAS level. The reason is that an asset can in principle have several AAS and an exchange would lead to higher effort. Presumably, an exchange of assets is less frequent. Another variant intended specifically for this task is modelling as a composite component (see Chapter 5.7) in which the bill of material (BoM, Chapter 5.5) is used.

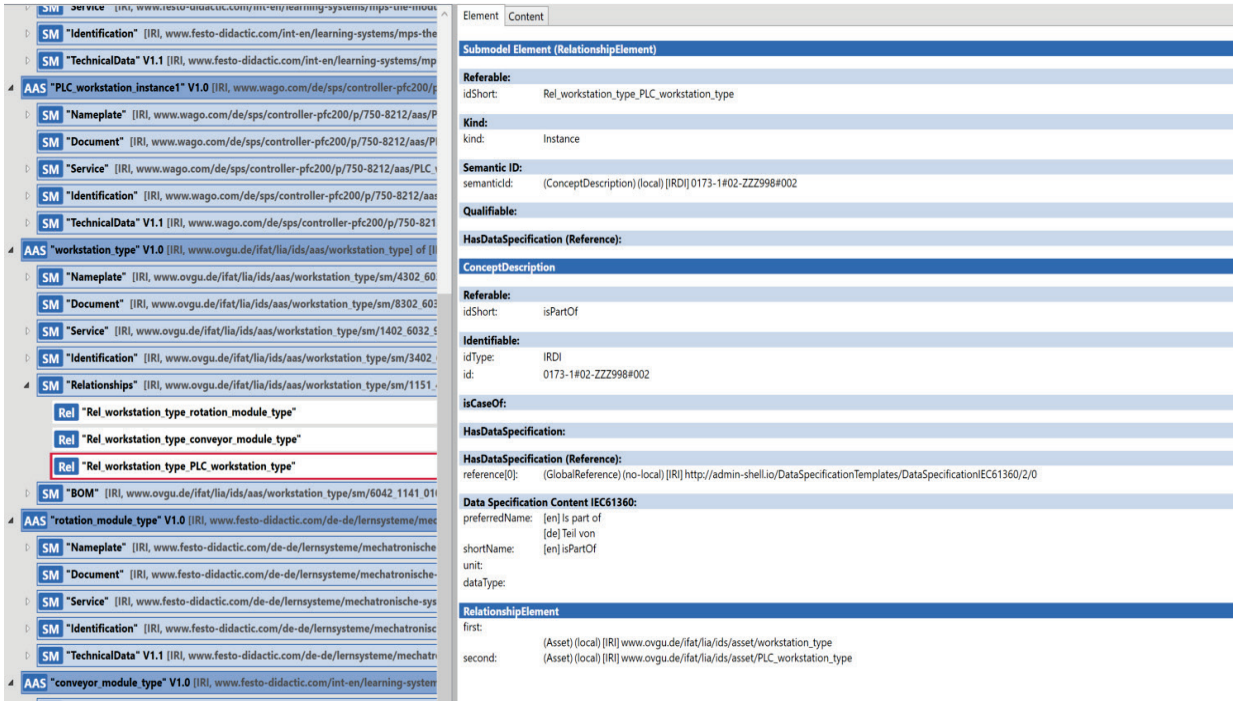


Figure 4-10 - Definition of the relationships between the sub-assets of the AAS workstation in the submodel "Relationships"

Each AAS consists of the submodels Nameplate, Document (collection of relevant documents), Service (information regarding support/maintenance), Identification (identifying properties) and TechnicalData (technical properties).

The selection of the properties and the assignment of values results from the information available from the manufacturers. This has been determined by the authors of this document or read from the nameplates of the assets (see Table 4-1). A special case is considered with the AssetType of the control PLC_workstation_type. Here, the type information is not taken from a specific manufacturer. In the considered scenario the system integrator of the workstation has decided to install PLCs of different types depending on the inventory. In principle, these can also come from different manufacturers. However, the system integrator assures in the TechnicalData submodel that the characteristics mentioned are assumed.

The joint use of nameplate and identification appears redundant, as some of the submodel elements are the same. This is due to the use in different use cases. One source of error that can lead to inconsistency is assigning different values to submodel elements with the same name.

In particular, the TechnicalData submodel appears to be of interest to a planner who is looking for components that satisfy the requirements of a planned facility. In this sense, the features of this submodel here represent assurances that the type of asset in question will meet these features.

Asset	AssetKind	Source
Workstation_type	Type	Own determination
Rotation_module_type	Type	[FES01]
Conveyor_module_type	Type	[FES02]
PLC_workstation_type	Type	Own determination
Workstation_instance1	Instance	Own determination
Drilling_module_instance1	Instance	[FES01] + Nameplate information
Conveyor_module_instance1	Instance	[FES02] + Nameplate information
PLC_workstation_instance1	Instance	[WAG01] + Nameplate information

Table 4-1 - Information sources for submodels of the workstation

Workstation instance from a planning perspective

In the following, it is shown how the AAS of the AssetInstance of the workstation was modelled. An asset of the kind instance is a concrete asset that exists only once. In the P&P demonstrator there are three instances of the type workstation_type. However, only the instance drilling_workstation_Instance is modelled as an example. The "instantiation" was carried out in the AASX Package Explorer by recursively copying the existing AAS workstation_type. In this context, recursive means that all submodels and their contents are taken over from the source AAS. idShort and id of the new AAS, its submodels and the assets must be renamed. The derivation from the AAS of the AssetType can be indicated by the DerivedFrom reference (see Fig. 4-11).

When deriving an AAS from the AAS of an AssetType, various consequences can arise with regard to the SubmodelElements:

- the type of SubmodelElement can change
    - Example: the length of the conveyor belt of the workstation_type can be configured by the customer within certain limits when ordering and is therefore modelled as a range. Finally, the conveyor belt of an instance has a fixed length and becomes a property
- the meaning of the property may change and receive a different semanticId
    - Example: the planner specifies that the PLC type requires at least 8 digital inputs (MinNumberOfIn-puts: 0173-1#02-AAP328#001). The instance of the PLC has 8 inputs (NumberOfInputs: 0173-1#02-AAP508#003).

Only properties that are relevant for the description of the respective AssetKind are modelled in an AAS. Characteristics that exist at instance level but are not relevant for the description of the AssetType must be added in a suitable submodel in the AAS of the AssetInstance. An example is the serial number of an asset. This feature is only modelled in the AAS of the AssetInstance.

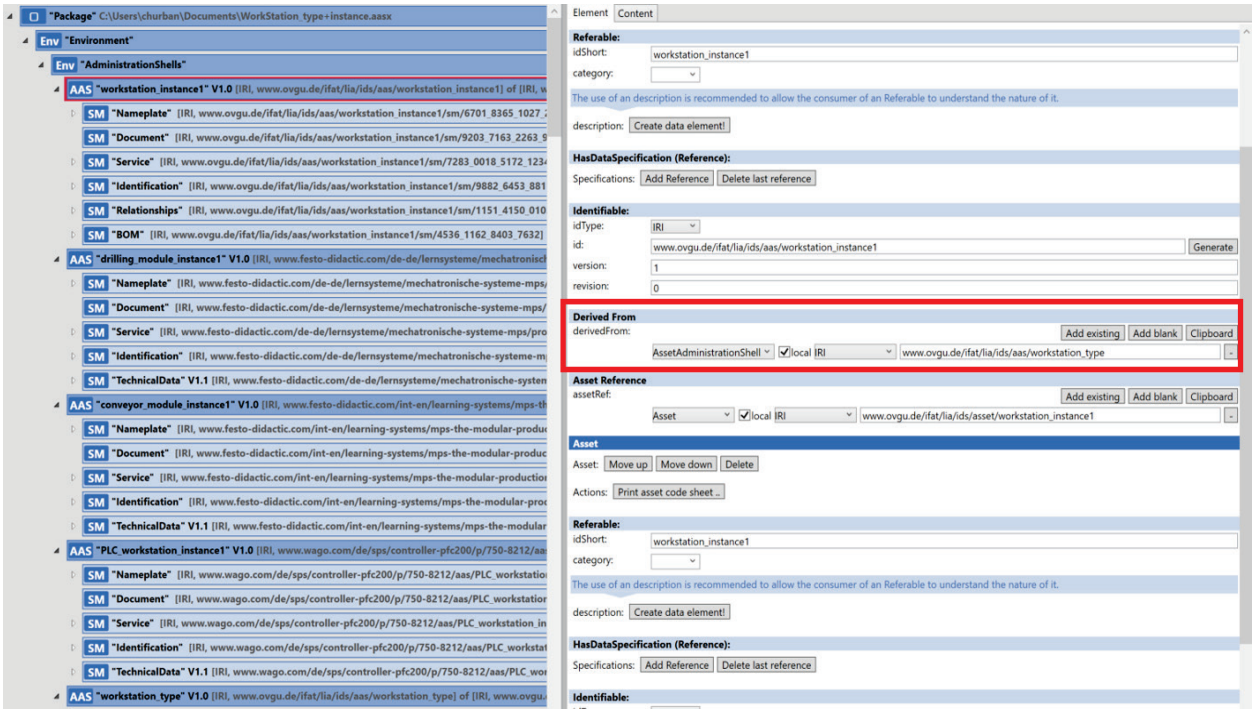


Figure 4-11: Derived From - Deriving an AAS from Asset-Kind:Type AAS-.

Workstation instance: Submodels for SP/SR

The concept of modelling the capabilities and services of an asset with capability submodel elements is introduced in chapter 4.6. To avoid repetition, reference is made to the relevant chapter for general explanations.

This chapter refers to an exemplary modelling of the capability "drilling" of the workstation. An abstract description of the capability of the workstation, in this case "drilling", based on the example in the capability white paper [DCI20], is shown in Figure 4-12 and Figure 4-13.

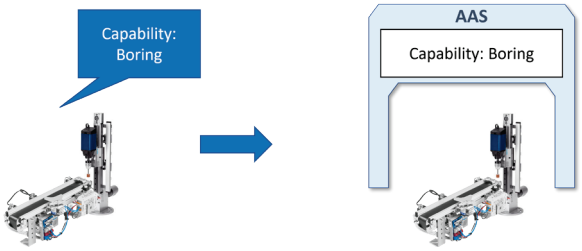


Figure 4-12: If the workstation in the I4.0 system acts as an autonomous service provider, its capabilities and offered services are described in the AAS.

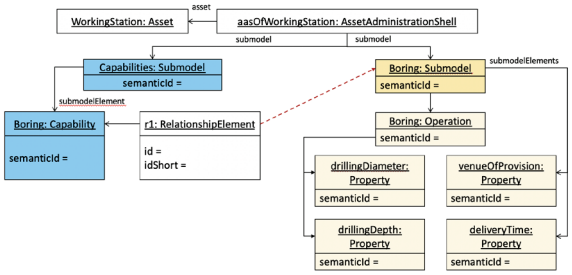


Figure 4-13: Description of the capability "Drilling" based on the capability white paper of the Platform Industrie 4.0 and BaSys 4.2 project [DCI20].

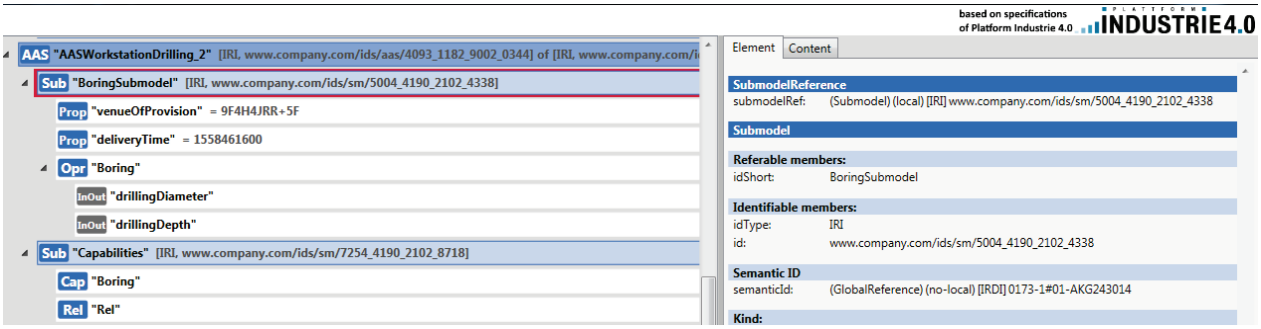


Figure 4-14: Implementation of the description of the "Drilling" capability in the AASX Package Explorer

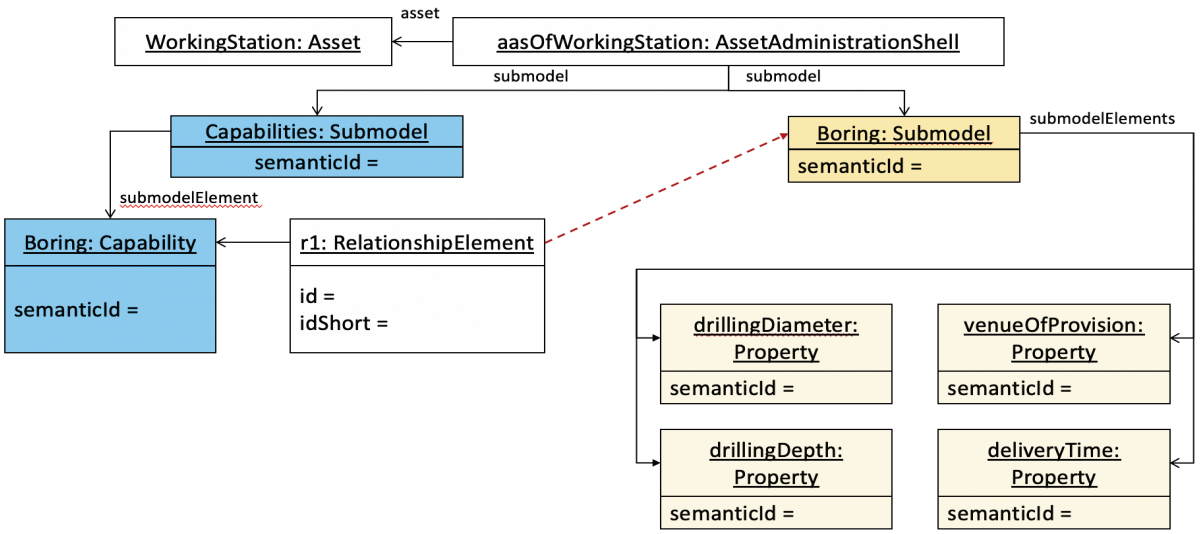


Figure 4-15: Alternative proposal for the description of the "Drilling" capability

Figure 4-14 shows how the description of the "Drilling" capability is implemented in the AASX Package Explorer.

The AAS capability and AAS operation concepts can also be interpreted differently, so that an alternative variant of the capability description could be imagined. The capability description shown in Figure 4-15 does not require the use of operations. The properties assigned to the operation in the previous example are directly assigned to the "Boring" submodel in the alternative proposal.

Figure 4-16 shows how the description of the "Drilling" capability is implemented without using the operations in the AASX Package Explorer.

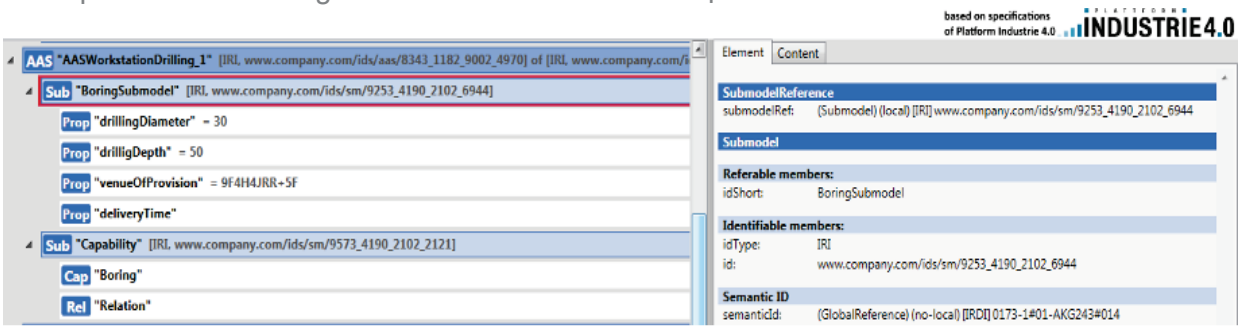


Figure 4-16: Implementation of the capability description (option 2) in the AASX Package Explorer

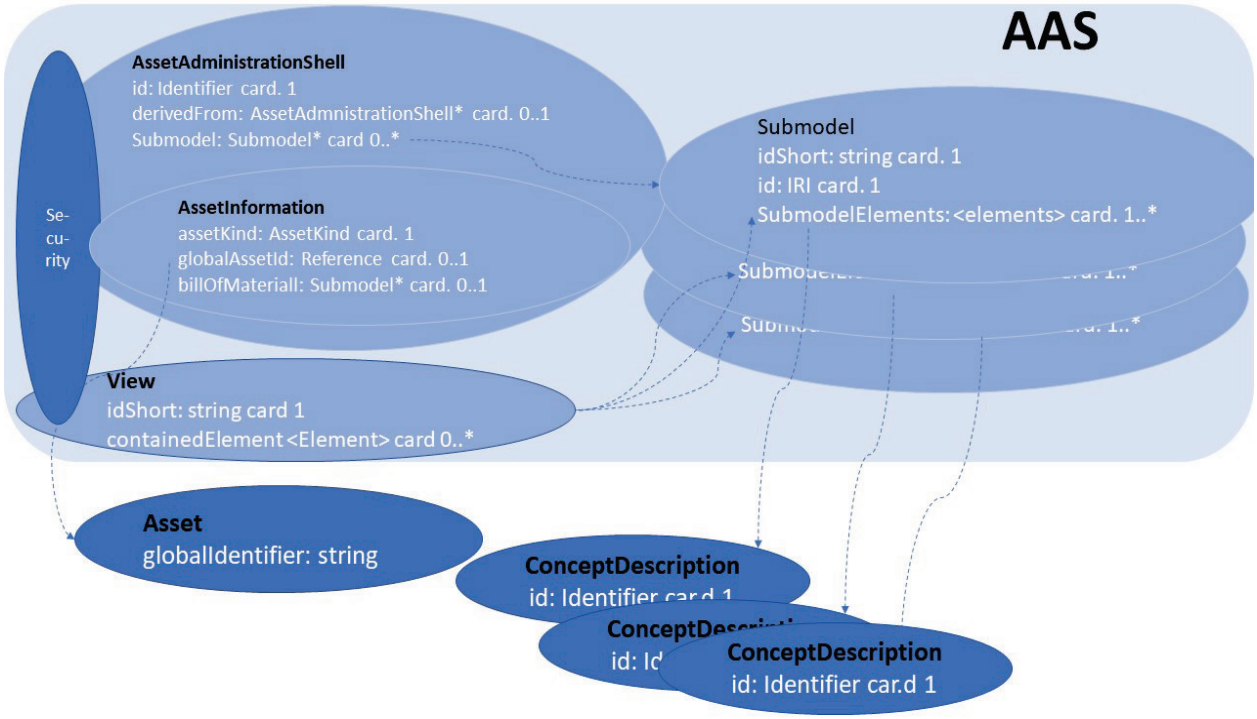


Figure 5-1: Overview of the basic structure of the AAS

(5) AAS modelling concepts

This chapter highlights essential concepts of AAS modelling and presents them as examples. The aim is to describe the interaction of different model elements for a concrete task.

The examples in chapter 5 refer to the delta robot. Therefore, its technical documentation [IGUS01] is used throughout as an asset-related source of information.

Basic structure of the AAS and selected important attributes

This chapter describes the rough structure of the AAS and presents it as an example. The goal is to describe the entry points for modelling.

The Asset Administration Shell (AAS) consists at the top level of the asset information, the submodels and the views. All these information units have attributes that describe them in more detail. A selection of the attributes with their cardinalities is shown in Figure 5-1 (white font). Security attributes and functionalities are consistently anchored in all elements.

The Asset and ConceptDescriptions do not belong inside the AAS. The Asset is described by the AssetInformation. The AssetInformation contains a reference to the asset. In addition, a Bill of Material (BoM), which is described as a submodel, can be referenced. Some SubmodelElements (e.g. Property) refer to their ConceptDescription (dashed line) with the semanticId. The ConceptDescription provides additional descriptive definitions (see chapter 5.3).

Element identification

Aim of the chapter

Presentation of the use of the concepts for the identification of AAS elements.

Underlying chapters of the "AAS in detail" specification:

- Chapter 4.4.3 "Identifiers for Assets and Administration"
- Chapter 4.4.4. "Which Identifier to use in which Elements" (Table 2)
- Chapter 4.7.2.2 "Referable"
- Chapter 4.7.2.3 "Identifiable"

Display in the Package Explorer

Using the example of the AAS of the P&P station, the attributes of "Referable" and "Identifiable" are shown in Figure 5-2. "idShort" is the essential attribute for Referable. Identifiable allows different types of identifiers to be selected, which are specified in idType as IRIs, IRDIs or Custom. The "id" is then to be filled in accordingly.

In addition, this specification is provided with version and revision.

Explanations

Each element of the AAS model needs an identifier in order to be uniquely named in a machine-readable way. This designation is used for the different use cases in which the model elements are required.

A distinction is made between globally unique identification, denoted by "Identifiable" and locally unique identification, denoted by "Referable". This distinction makes it possible that identifiers can be used identically in several AAS (only possible for idShort). This contributes significantly to the reuse of model parts.

Almost all model elements are referable. AAS Identifiable model elements are referable and also have the attributes:

- AdministrationInformation (version and revision) and the identifier consisting of idType (IRI, IRDI and Custom) and id

Identifiable are:

- AAS, Asset, Submodel, ConceptDescription

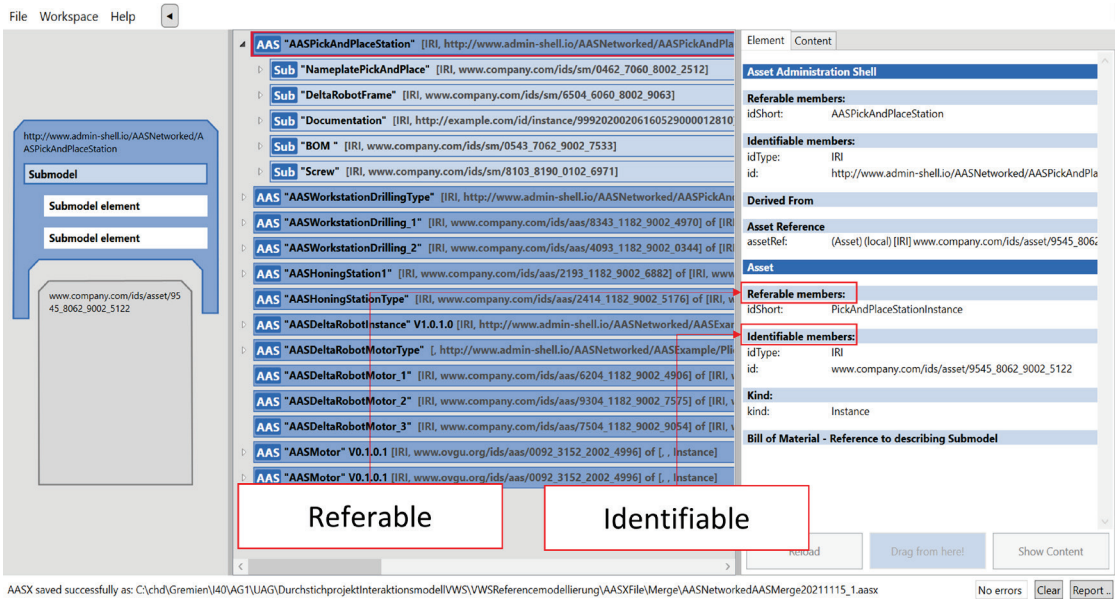


Figure 5-2 - Referable and Identifiable Data for AAS and Asset of the P&P Station

Semantic assignment to model elements - the "semanticId"

Aim of the chapter

Description of the task of the attribute "semanticId".

Input information used for the model

Relevant properties, parameters, states of the modelled assets shall be transferred into the characteristics according to IEC 61360. The unique identifiers for features of other model elements are to be taken from the standardised vocabularies.

- A vocabulary of semantic definitions favoured by Plattform Industrie 4.0 is the ECLASS standard.
- Other vocabularies: Company-specific device classifications and their feature definitions
  - A user-specific semanticId can be defined in the form of an IRI
  - PackageExplorer has the option to generate the IRI automatically

Model elements of the AAS metamodel used

- Chapter 4.7.2.7 "Semantic Reference Attributes

Visualization in the Package Explorer

Using the example of "ManufacturerTypeNameAAS" property of the nameplate submodel, a semanticId is shown in Figure 5-3. This shows the reference to the corresponding ConceptDescription, in this case an ECLASS IRDI.

Explanations

In addition to the identification of the model elements, it is intended that each model element has a complementary semantic description. Therefore, the model elements have the attribute "semanticId".

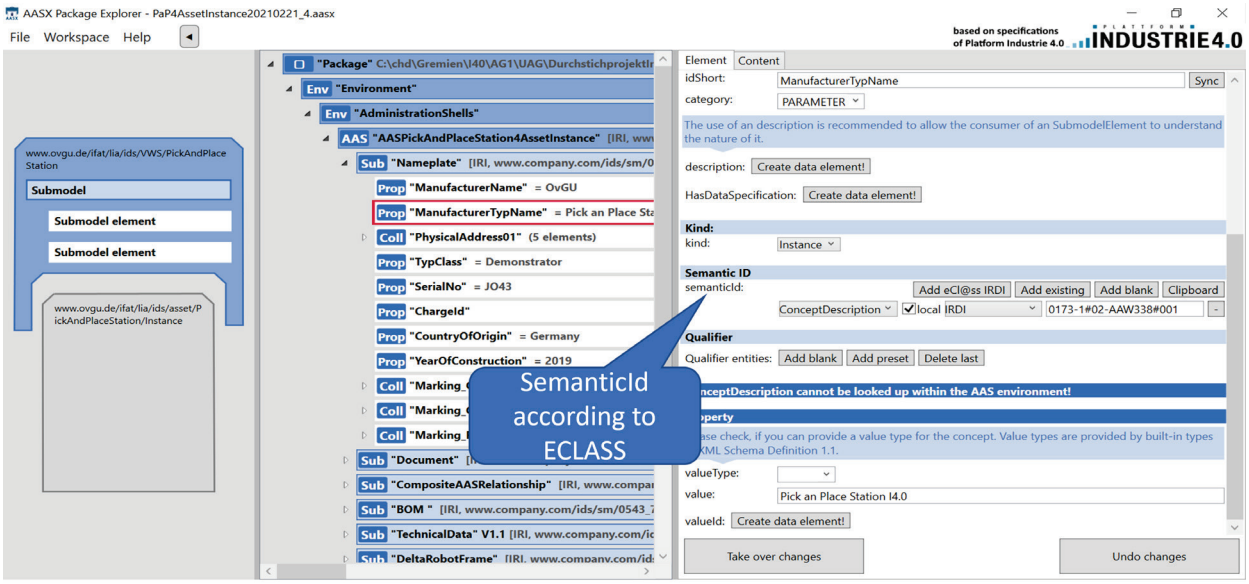


Figure 5-3 - SemanticId using the example of a property

This is a reference to the machine-readable description of the model element, i.e. it is a reference to an entry in a dictionary. The preferred case of a dictionary used is one in which the characteristics (property) are described according to the IEC 61360 standard. This standard specifies the attributes to be used for the description.

Almost all model elements (except Asset, ConceptDescription and Asset Administration Shell) have a semantic reference, i.e. the semanticId. The model elements of the AAS also have their own semanticId, which is created in the namespace www.admin-shell.io/....

The semanticId is not an identifier that is used for naming and thus for addressing in a use case. The identifiers of Identifiable and Referable are responsible for this. The semanticId is an additional attribute that is available for the content-related technical explanation of the model element. It points to the supplementary description and thus contributes to the assignment of the meaning of the model element. Thus, it clarifies, what is behind the model element. In other words, the attributes of a model element are not stored locally but are defined globally. This means that the same characteristics with the same attributes can be used in different engineering phases. The mapping of feature values during the transition from one technology to another or from one tool to another is no longer necessary.

The AAS metamodel defines entries in a dictionary based on IEC 61360. These entries are referred to as concept descriptions (see 5.8). Only individual entries in a dictionary are referenced by a model element of the AAS. Therefore, there is no dictionary itself in the AAS specification, but only the entries, the ConceptDescriptions.

Relationship between AAS, Asset and Asset Identification Submodel

Aim of the chapter

- Description of the relationship between asset and AAS in a submodel
- Modelling of the type plate/nameplate of the asset and its referencing in the AAS

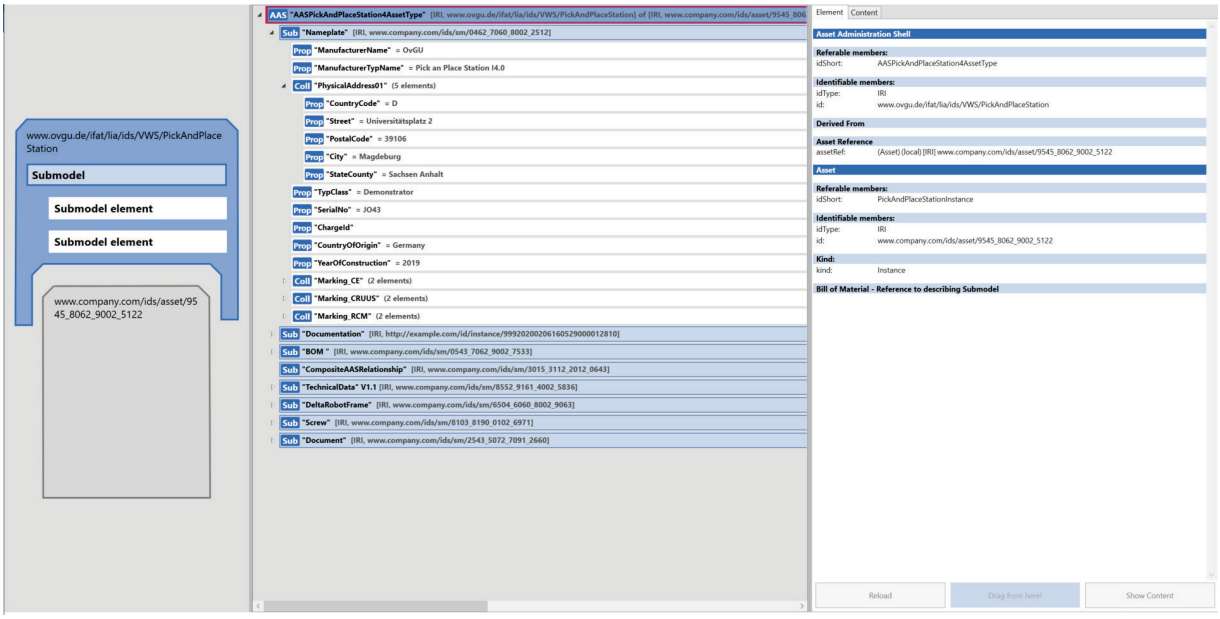


Figure 5-4 - Reference between AAS and Asset

Model elements of the AAS meta-model used

- Table P. 58 - attribute "asset" (this is a reference)
- Table P. 59 - attribute "assetIdentificationModel

Explanations and visualization in the Package Explorer

For the representation of the relationships between AAS and asset, the object "AssetInformation" is available from version 3.0 of the AAS specification. Among other things, the asset identification and the asset child are to be specified in this object. The asset identification is specified as globalAssetId, which is to enable a worldwide unique identification. A good possibility is to assign this according to DIN SPEC 91406. Companies usually have their own internal identifiers first, e.g. serial number and order code. These supplementary IDs are to be entered in the externalAssetId, of which there can be several. In addition, a submodel similar to the type nameplate is created in the AAS, which can be uniquely identified by reference from the AssetInformation object. This submodel should be derived from the submodel template "Digital Nameplate" [DNI20]. Typical information for this task is shown in the Figure 5-4.

The asset is not part of the AAS, the I4.0 component consists of asset and AAS. Nevertheless, the asset is modelled in the Package Explorer. The Environment (Env) is responsible for this.

Since the asset information is only available from V3.0 of the AAS specification, it cannot yet be used in the example with the Package Explorer. The corresponding definitions are therefore shown in Figure 5-5 with page reference of the AAS specification in which the definitions are contained.

If assets are composed of several components, their relationships must be described in the Bill of Material (BoM) (see 5.5).

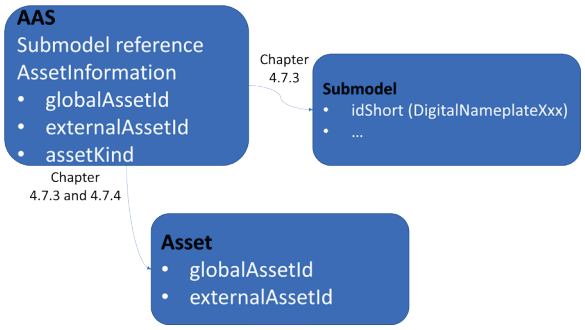


Figure 5-5 - Relationship between AAS, Asset and Asset Identification Submodel

Bill of material

Goal of this chapter

Representation of the structure of an asset as a submodel "Bill of Material" using the example of DeltaRobot 360.

Model elements of the AAS meta-model used

- Chapter 4.3: Composite I4.0 Components
- Chapter 4.7.5, p. 58 and p. 59: Definition

Explanations and visualization in the Package Explorer

The BillOfMaterial (BoM) is a submodel. It is contained in the AAS of the asset to which the components in the BoM belong. In the AssetInformation is the attribute that contains the reference to the BoM submodel.

If components are included that are composed of further components, this refinement of the corresponding AAS is assigned to the asset consisting of further components.

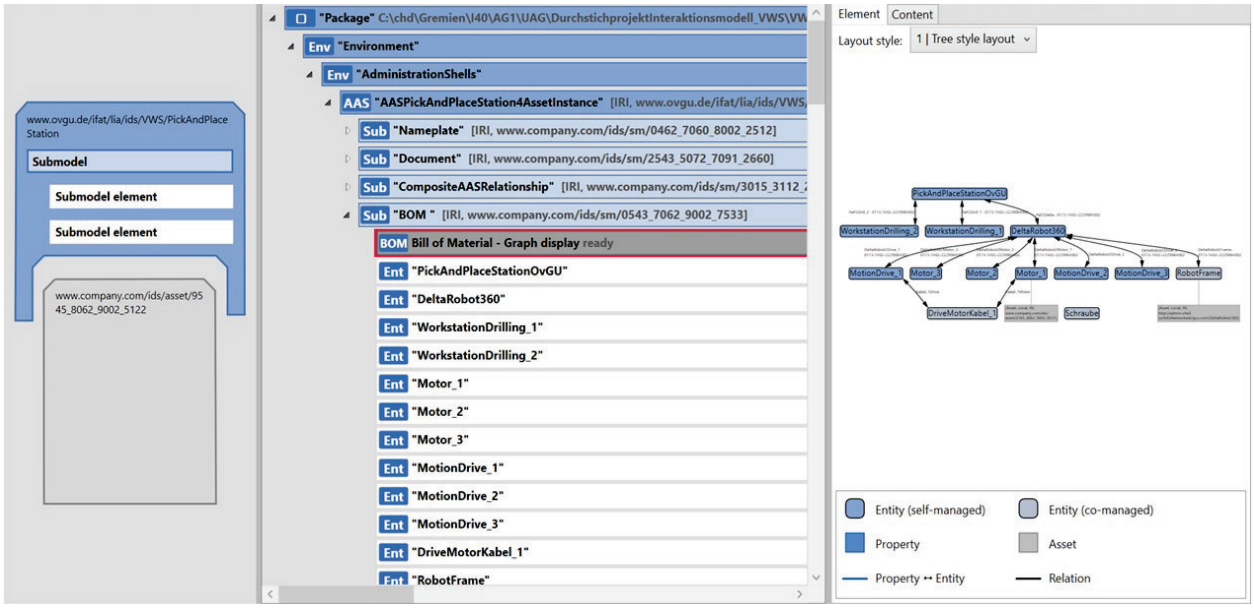


Figure 5-6: Overview BillOfMaterial P&P Station Instance

The components of the asset in the BoM are to be modelled as "Entity" in this submodel (Figure 5-6). Relationships between the components (e.g. is part of - isPartOf) are modelled as "RelationshipElement". The submodel containing the BoM must be provided with the semanticId "http://example.com/id/type/submodel/BoM/1/1" in order to be able to identify this submodel as a BoM.

The entities correspond in a 1:1 relationship to the assets to be modelled. For the entities, idShort, entityType and the reference (identifiable) to the asset must be specified (Figure 5-7). The attribute "entityType" can be "CoManagedEntity" or "SelfManagedEntity". CoManagedEntity states that the asset of this entity has no AAS and is managed in the AAS where the BoM is at. SelfManagedEntity states that the associated asset has its own AAS.

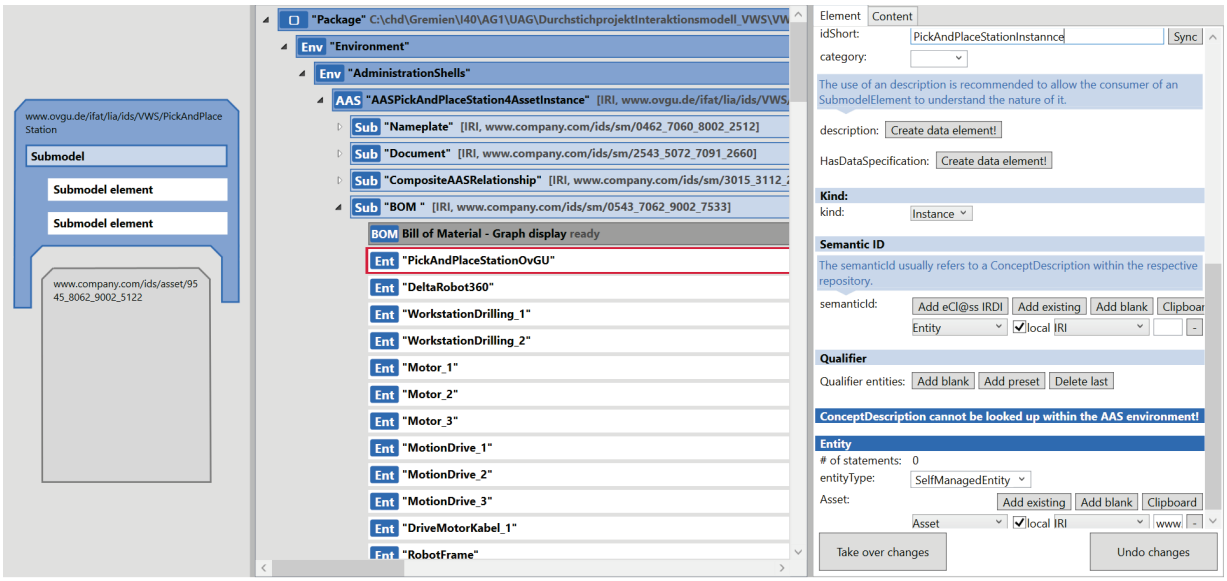


Figure 5-7: Bill of Material modelling

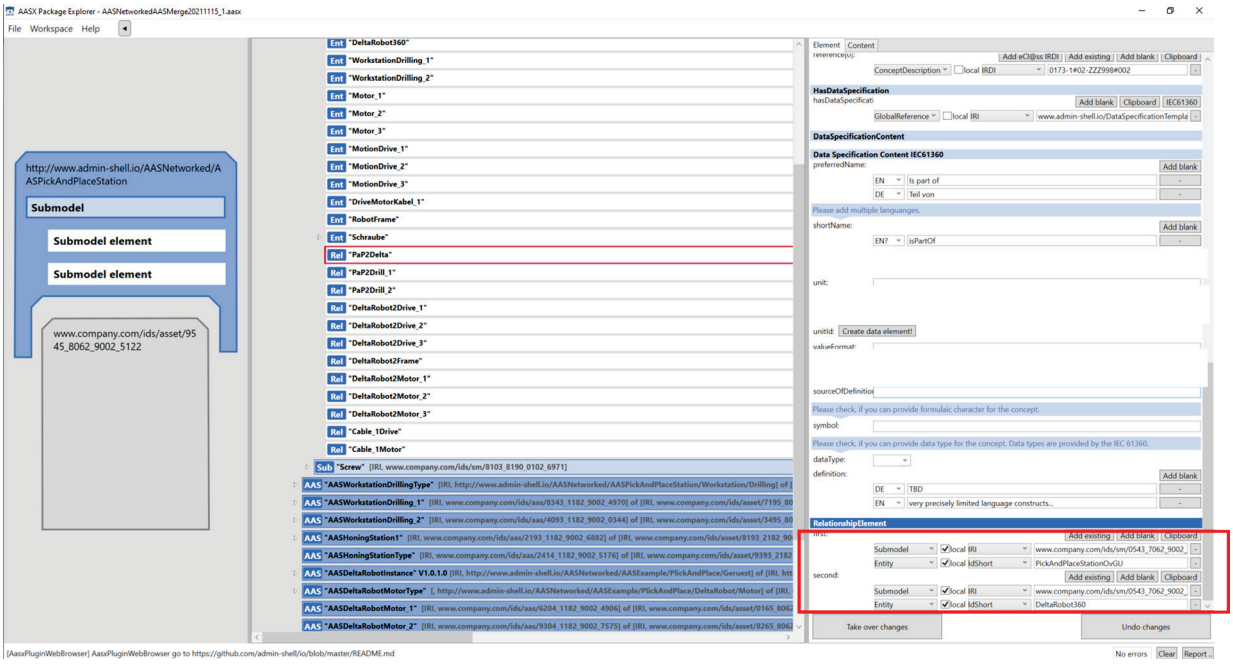


Figure 5-8: RelationshipElement example

The RelationshipElement is used to model the relationships between the entities of the BoM. The relationship is directed and has a starting point (first) and an end point (second). The identification is done by means of idShort and the identifier (IRI) of the corresponding entity (Figure 5-8).

The semanticId of the relationship element contains the meaning of the relationship. In this example, "isPartOf" is identified by "0173-1#02-ZZZ998#002". There is then an entry for this in the concept description (Figure 5-9).

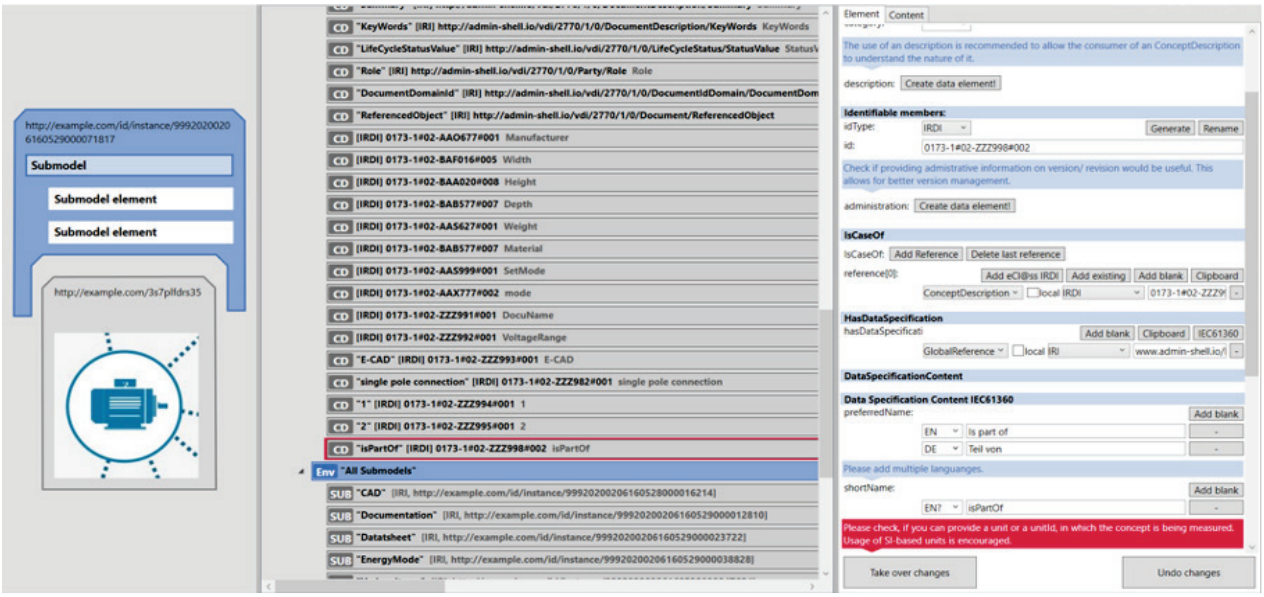


Figure 5-9: isPartOf - Property in the ConceptDescription

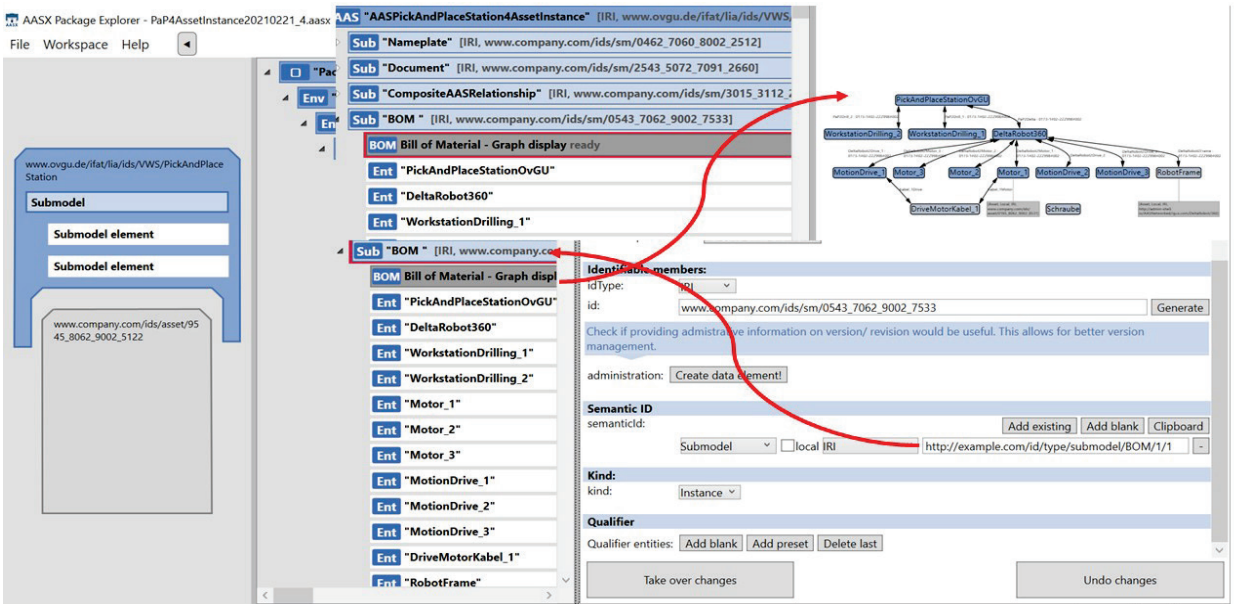


Figure 5-10: Relationship of the asset to the BoM in the Package Explorer - example

A special feature of the Package Explorer is that if semanticId = "http://example.com/id/type/submodel/BoM/1/1", a graphic representation of the BoM appears when the BoM element of the submodel is activated. This appears when the semanticId is entered correctly. In addition, further submodel elements can be specified for an entity to describe the entity, i.e. the asset, in more detail. For example, a reference to the AAS of a self-managed asset or properties for a co-managed asset. This will be explained in more detail. There are components in machines and plants that either do not have their own communication capability (e.g. screws, rods, seals), occur in large quantities or whose individual value (in terms of purchasing costs) is low.

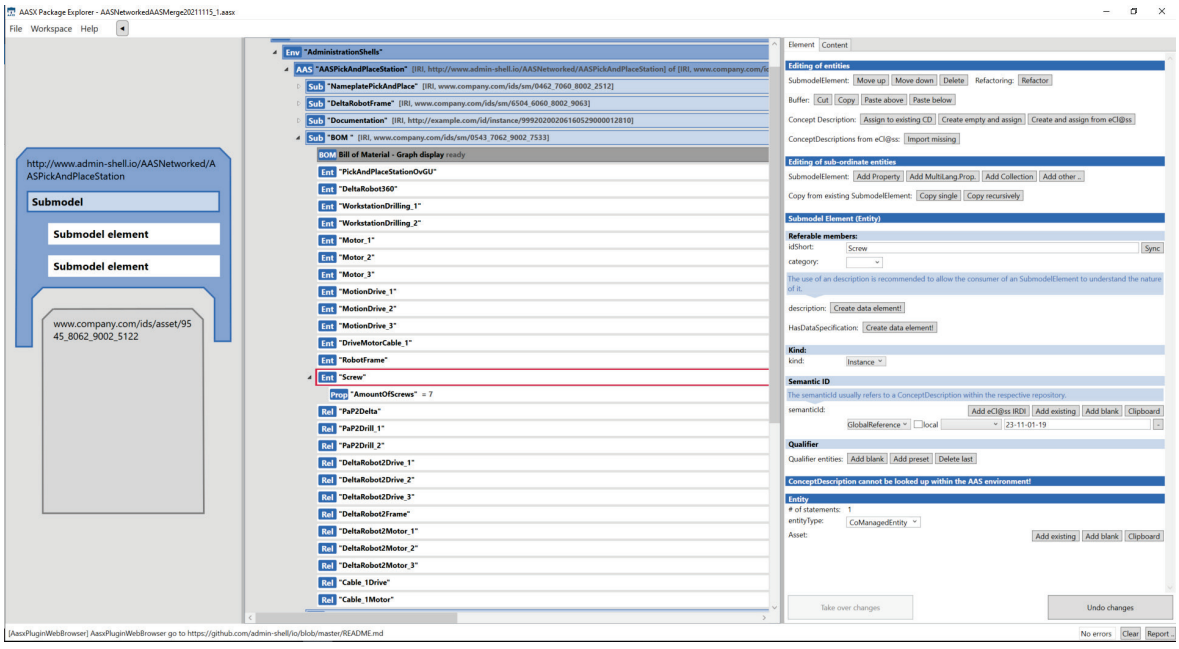


Figure 5-11: Entity overview from the BoM in the Package Explorer

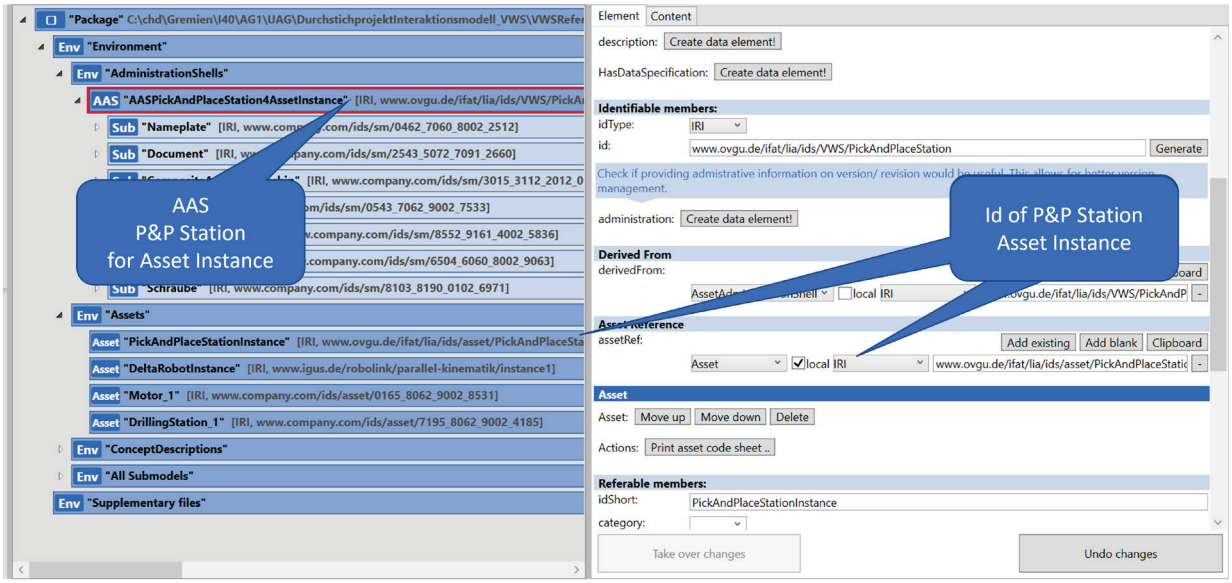


Figure 5-12: Refined characteristic (property) of the CoManagedEntity screws - here example number of screws

For these, however, information is still needed to accompany the life cycle. The entityType "CoManagedEntity" applies to these assets. In the example, screws are such "CoManagedEntity" and are modelled accordingly in the BoM (Figure 5-11). The concrete number, for example, should also be modelled and appears as a property of the entity, here in the example "Number of screws" (0173-1#01-RAA001#001), a characteristic that exists in ECLASS (Figure 5-11). The characteristic "Number of screws" is an ECLASS characteristic and is also included in the ConceptDescription (Figure 5-13).

Figure 5-12 shows that there can be a submodel in which the screw is described in more detail, here exemplarily only with the property "Execution of the thread".

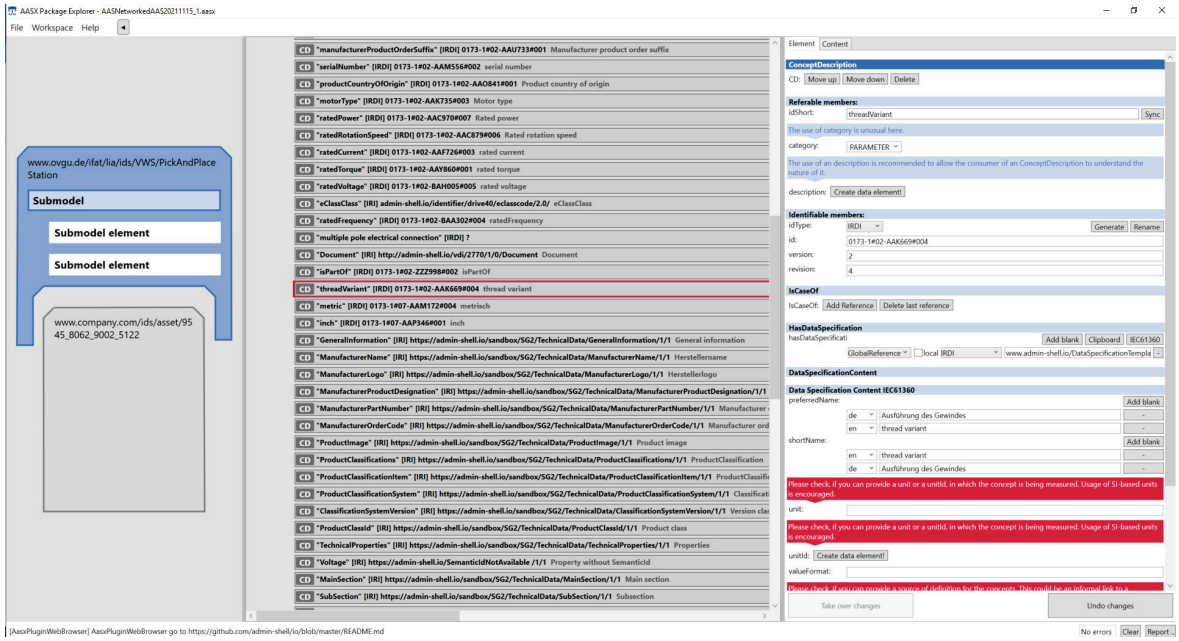


Figure 5-13: Entry of the feature "number of screws" in the concept description according to IEC 61360/ECLASS [https://www.eclasscontent.com/in-dex.php?id=23110119&version=11_1&lan-guage=de&action=det].

This submodel is accommodated in the AAS in which the BoM is located. In addition, this submodel gets a reference to the associated CoManagedEntity. This reference needs a machine-interpretable semanticId to unambiguously express the fact that it is the link between the entity contained in the BoM and its description in a submodel. However, it is also possible that the entity points to the description of entities in a submodel. The comprehensive list of properties of the screws can be found in the enclosed reference.

Figure 5-13 shows a section of the concept descriptions that are modelled in the P&P station.

The kind attribute - types/templates and instances

Aim of the chapter

Explain the use of the attribute "kind" for assets (assetKind) and submodels (modellingKind) and relationships between these concepts.

Model elements of the AAS metamodel used.

- Chapter 4.2 (general explanations and examples)
- Chapter 4.7.5, p. 66 and p. 67: Asset kind
- Chapter 4.7.2.5, p. 57: ModelingKind

Explanations and representation in the class diagram and in the Package Explorer

In this chapter, the use of different properties of the model defined by the kind attributes is explained using the P&P station example. A distinction is made between AssetKind and ModelingKind. As the name AssetKind indicates, this attribute only refers to the

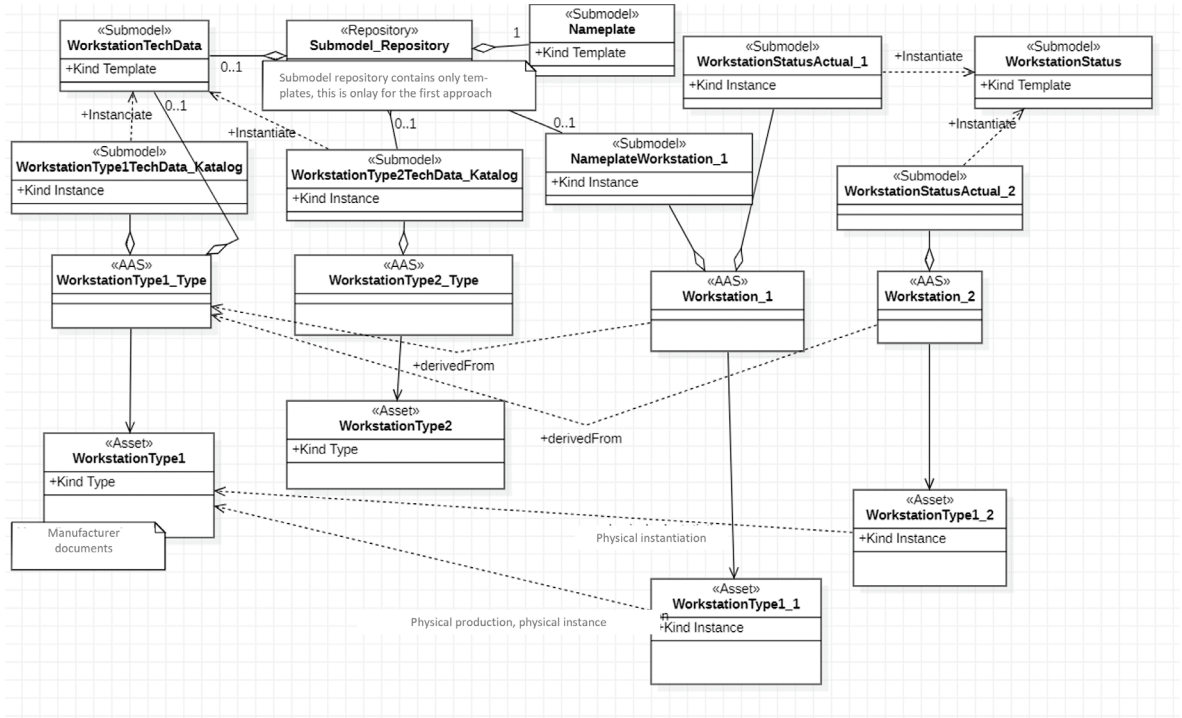


Figure 5-14: Modelling example of asset, AAS and submodels with regard to type, template and instance

distinction in assets. ModelingKind is used to distinguish AAS metamodel elements.

The modelling could start with the assets, which can be of AssetKind=Type and AssetKind=Instance. The modelling (Figure 5-14) provides in principle that for recurring components of the P&P station (e.g. the workstations for drilling) the submodels are to be modelled for each asset type (WorkstationType1 and WorkstationType2 each with kind = Type) and the asset instance (Workstation-Type1_1 and Workstation-Type1_2 each with kind = Instance).

This means that there is initially one AAS for the asset type (each for Workstation-Type1_Type and WorkstationType2_Type) and one AAS each for the asset instances (Workstation_1 and Workstation_2, each derived from WorkstationType1_Type (derivedFrom), WorkstationType2_Type has no instance in Figure 5-14). AAS do not have the "kind" attribute, it results from the assignment to the asset. For the asset instance, there is typically an AAS that is delivered with the asset instance and there can be further AAS with additional submodels (WorkstationStatusAc-tual_2) that describe, for example, the actual data of properties.

These can have a range in the type description. During runtime the instance has one value only so the range submodel element as to be changed to single property. The submodels that are only filled with asset data at runtime can also be contained in AAS at delivery time.

Submodels can be of Kind=Template and Kind=Instance. Submodels with Kind=Template are primarily intended for standardised submodels. For example, the nameplate or the documentation (not shown in Figure 5-14) are such submodels. All submodel elements (properties, ranges, files, selection ...) are created in the templates. The corresponding attributes may have default values. In addition to the submodel elements, the ConceptDescription entries belong to the submodel template definition, since the definition bases are stored here. All submodels of an AAS, both for AssetKind=Type or AssetKind = Instance, are defined with the submodel attribute Kind=Instance. Submodels derived from submodel templates may have additional submodel elements. Therefore, formally, they are not an instance derived from a type.

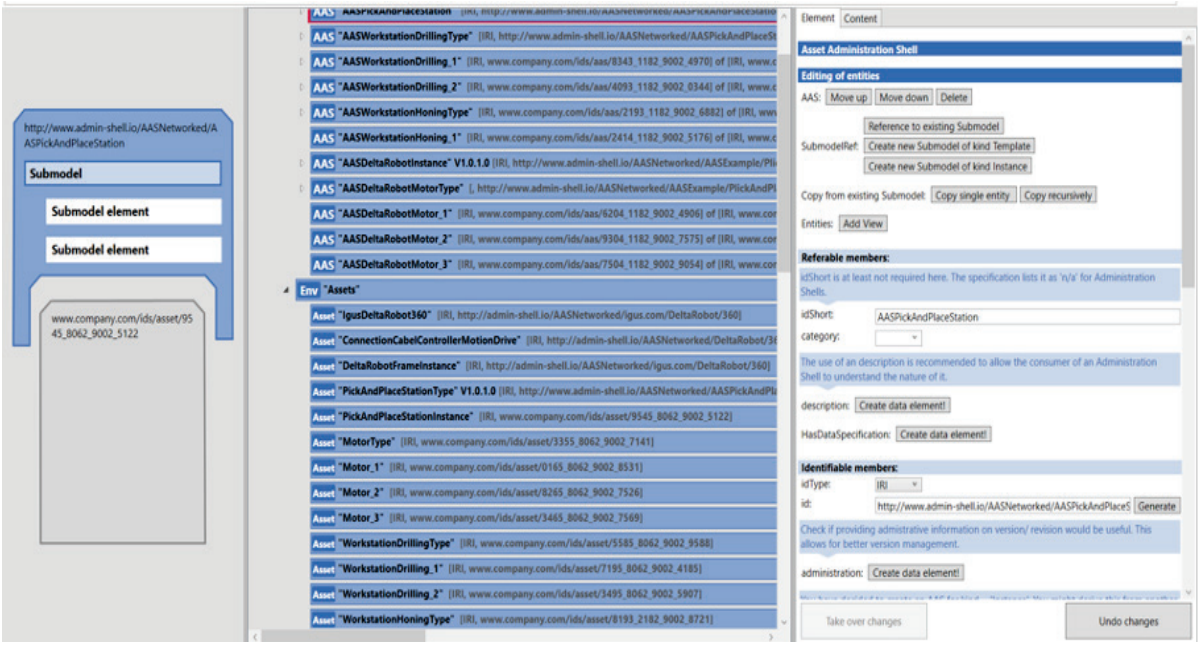


Figure 5-15: AAS overview in the Package Explorer

The structure designed in the class diagram was implemented as a composite AAS in the Package Explorer (Figure 5-15). The P&P station (AAS PickAnd-PlaceStation) as well as the types for the drilling and honing machines (AASWorkstationDrillingType and AASWorkstationHoningType) and the motor type (AASDeltaRobot-MotorType) with the corresponding instances *Drilling_1, *Drilling_2, *Honing_1, *Motor_1, *Motor_2 and *Motor_3 are implemented. This includes the corresponding assets.

Submodel templates, which serve as a standard and have been adopted by the corresponding working groups, have a semanticId that identifies them as such. This allows the user to rely on a specific information content, i.e. submodel elements. These semanticId start with http://admin-shell.io/ ... e.g. http://admn-shell.io/vdi/2770/1/0/Documentation

I4.0 Composite Component

Aim of the chapter

Explain the task and structure of the composition of AAS.

Input information used for the model

- All descriptions of the P&P station

Model elements of the AAS metamodel used

- Chapter 4.3, p. 33-34: Composite I4.0 Components

Explanations

Assets are usually composed of several components. As shown in the BoM in chapter 5.6, assets can be described by SelfManagedEntitiy and CoManageEntity. CoManagedEntity assets are described in submodels that are in the AAS in which the BoM submodel is also located. SelfManagedEntity assets have their own AAS.

The result is that several AAS of the subcomponents and those of the superordinate AAS belong together. This is referred to as a Composite I4.0 Component and ultimately as a Composite AAS. The AAS are in aggregation relationships with each other, which can also consist of more than one hierarchy level.

This requires a description in the respective composite AAS. It is explicitly pointed out that the component AAS are not in the superordinate AAS. The AAS remain independent. The aggregation relationship is described by the submodel element "Relationship".

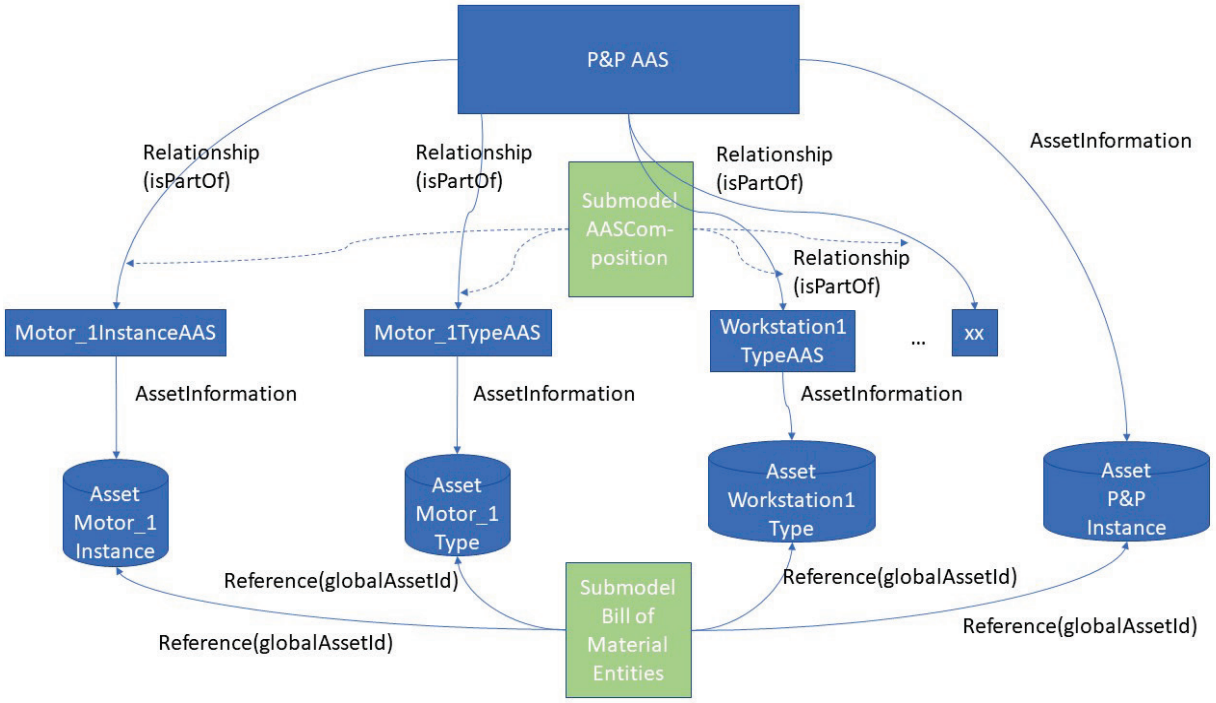


Figure 5-16: Composition of AAS as an overview

As a best practice solution, a submodel is proposed in which the relationships are entered for one "isPartOf" relationship each (see Figure 5-16, green submodel box above).

This submodel is in the AAS in which the BoM submodel with the SelfManagedEntities is located. In this CompositionAAS submodel, the relationships are each entered as a 'Relationship' with two endpoints, e.g. as a relationship between P&PAAS and Drive_1InstanceAAS. The solid line is the aggregation relationship (is-PartOf). The dashed line shows that the description as a relationship is in the submodel. It is the same RelationshipElement type that is already used in the description of the relationship of the entities in the BoM. Of course, the Ids of the subjects and objects of the corresponding AAS are to be used here. The lower part of Figure 5-16 corresponds to the description of the BoM from Chapter 5.5.

In the Package Explorer, the relationships look as shown in Figure 5-17.

Concept Descriptions and Data Specifications

Aim of the chapter

Explain the task and structure of the concept description and how the DataSpecification works.

Model elements of the AAS metamodel used

- Chapter 4.8, p. 94 ff: Predefined Data Specification Templates
- Chapter 6.2.4 p. 124: Embedded Data Specification

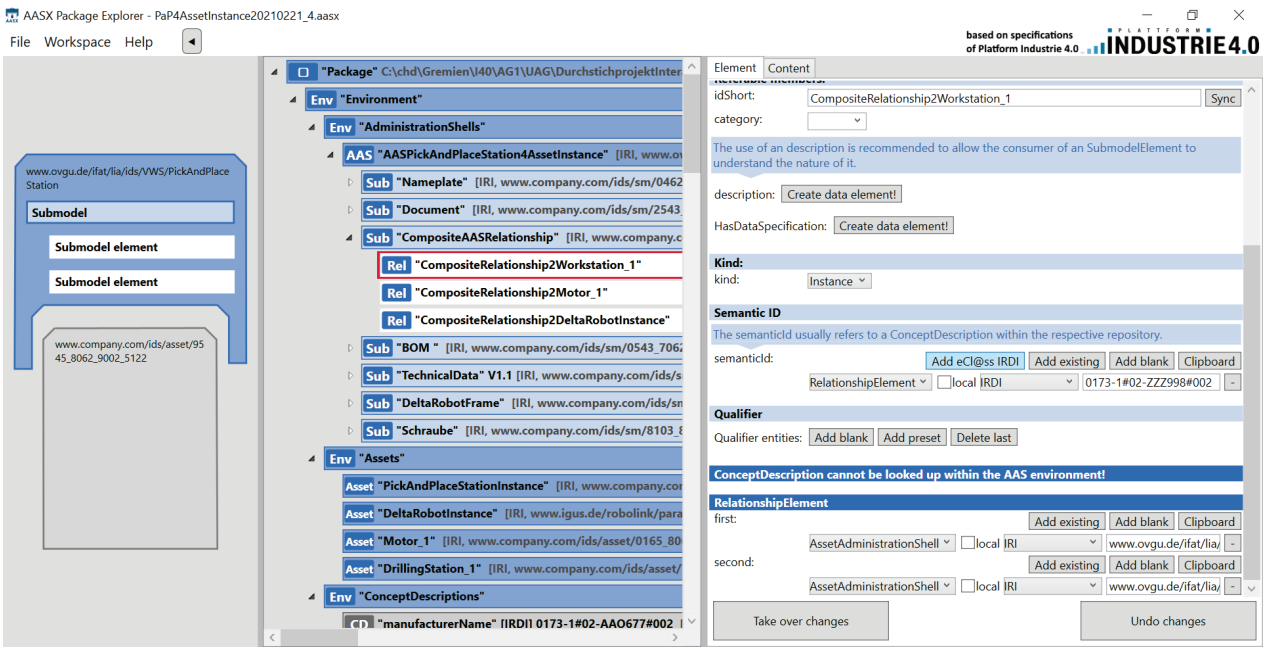


Figure 5-17: 'Relationship' describes the relationships to the AAS of the system components

Explanations

The AAS information model offers diverse types of model elements that describe the assets according to the various use cases along the life cycle. This diversity is mainly reflected in the SubmodelElements that fill out the submodels. Each model element is described in more detail by a set of attributes that characterise the model elements. The attributes depend on the type of model element.

Many of the model elements included in the AAS are already used in industrial practice. They are provided with tailored attributes for the different applications. One of the submodel elements is the property. It describes the variables, parameters and constants of the assets. One might have expected that a set of attributes would be defined for these properties in the AAS metamodel. This path was not followed. As described in chapter 5.3 and 5.4, a separation of identification of model elements and semantic definition is made. Therefore, only a minimum set of attributes is defined for the SubmodelElements and reference is made to the detailed description by using the semanticId. The definition according to IEC 61360 is used as the default specification for the attributes. This corresponds to the embedded data specification in the metamodel. This has several reasons and effects:

- It is explicitly recommended to use properties that are already contained in vocabularies. The Platform Industrie 4.0 favors industrial standards such as ECLASS or Companion Specifications of the OPC Foundation. These properties are already the result of a consensus process and therefore have a larger scope of validity than manufacturer-specific definitions.
- When defining the submodels, it becomes visible which properties have not yet gone through a standardisation process. It can then be decided whether standardisation is worthwhile or not.
- The norms and standards also contain specifications for selected values, e.g. units of measurement, colours or other enumerations belonging to an asset type. These values, which are usually represented by strings or individually

defined numbering, are then also clearly identifiable.
- The uniquely defined set of attributes allows applications to implement these specifications as "normal". This significantly limits the variety of evaluation algorithms and simplifies application development.

The AAS metamodel specifies a recommended or mandatory subset of IEC 61360 attributes for each submodel element type (chapter 4.8.2 in [BMWi 2020]).

The principle is that AAS model elements that inherit from HasDataSpecification implement a DataSpecificationContent when instantiated, which can be variable, but in the current specification is implemented according to IEC 61360. The attributes of DataSpecification-Content are therefore only added to the corresponding instances when a model element is instantiated (chapter 6.2.4 in [BMWi 2020]).

This document does not use the possibility of using attributes other than those predefined in the meta-model and those provided by IEC 61360. For a basic understanding of the details of this mechanism, the specification (metamodel 4.8 and 6.4.2) should be used.

The ConceptDescription used here is exactly the detailed attribute selection for the model elements specified by IEC 61360. An overview is given in Figure 5-18.

The number of ConceptDescriptions referenced by an AAS through a semanticId depends on the number of AAS model elements that have the semanticId. Although the ConceptDescription is not part of an AAS (it contains standardised descriptions of model elements, e.g. properties), it is included in the PackageExplorer in the Environment section (env) under ConceptDescription. It is not mandatory to include all ConceptDescriptions in the AASX description. Concept Descriptions taken from a standard (ECLASS or IEC CDD) are thus uniquely identified by their sematicId, i.e. the IRDI. If the description is correct and complete, the application can also interpret the attributes unambiguously.

The basic concept that is effective here is the so-called data specification. This is initially only used for the integration of IEC 61360 attributes. Although the data specification

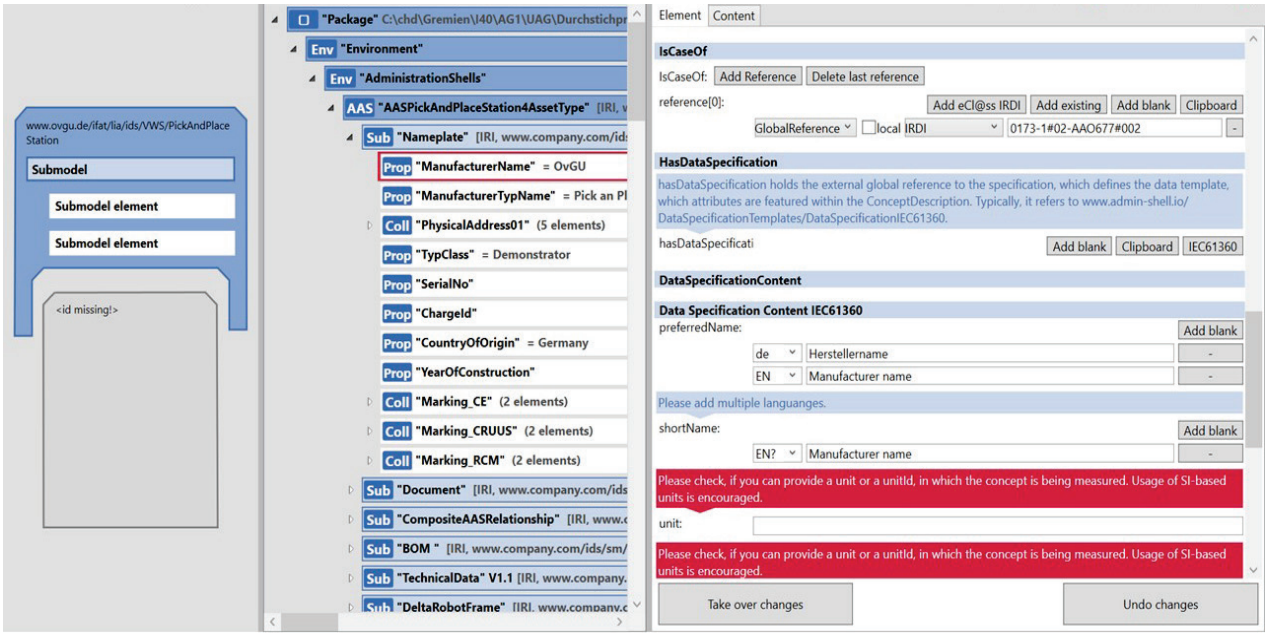


Figure 5-18: Attributes of the DataSpecification according to IEC 61360 of a property using the example of "manufacturerName

concept seems quite complicated because of the option of different attribute definitions, the implementation is very simple. The AAS model elements receive a semanticId on the ConceptDescription and can therefore use the IEC 61360 attributes in the AAS model. The corresponding IRDI or IRI of the ConceptDescription is then the value of the semanticId of the AAS model element. The effect for the application programmer is that with a standardised ConceptDescription, programming can be done with its attribute definitions and values.

Submodel templates

Submodel templates are pattern that provide predefined submodels with the necessary model elements for frequently recurring aspects of assets or use cases. A number of submodel templates are being standardised as part of the I4.0 activities. Examples are Nameplate, Documentation, TechnicalData, Simulation and many more. They serve as master copy. The standardized submodel templates will be accessible to both the AAS developers and their users.

The procedure for using the templates is not as strict as the type/instance relationship known from object orientation. On the one hand, there are optional elements in the templates that can be omitted during use, and on the other hand, the user can add further elements or instantiate parts several times. In addition, asset manufacturers can derive their own customized templates from the standardized templates and then implement them in a manufacturer-specific way (Figure 5-19). The submodels instantiated in the AAS should indicate that they originated from this template by using the semanticId of the associated submodel template. This allows the obligatory part of the submodel to be presupposed in the application. An example of this is the BoM (see chapter 5.5).

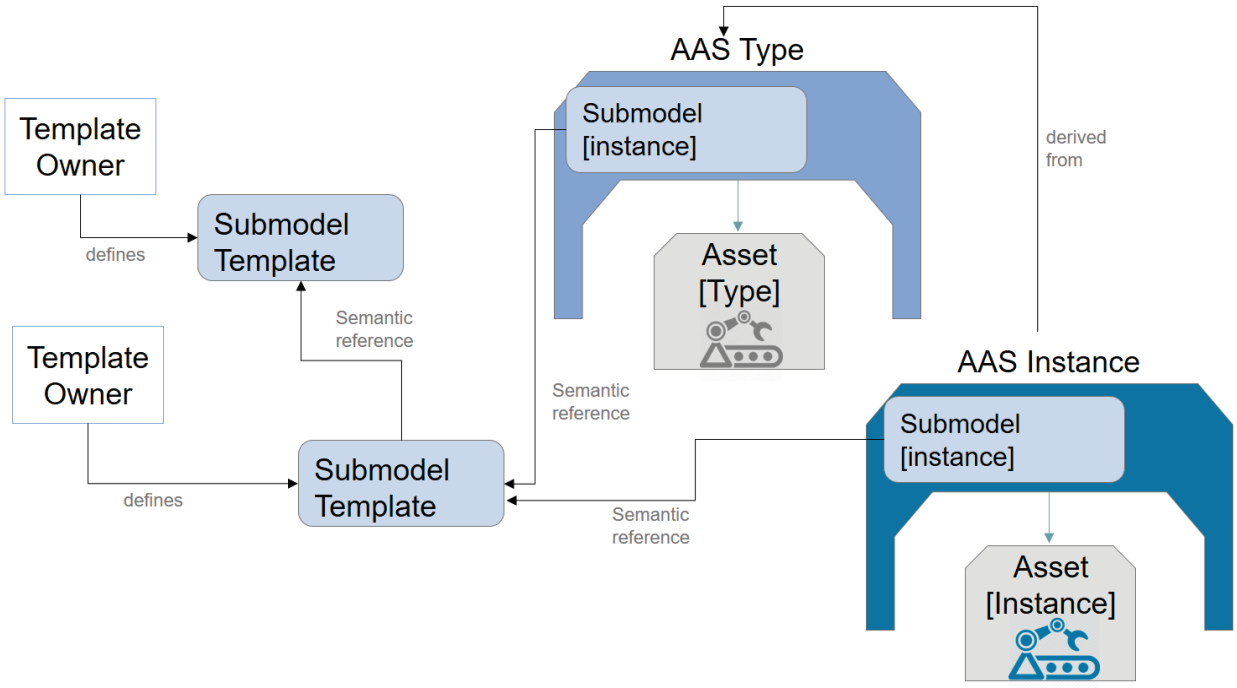


Figure 5-19: Relationship between submodel templates and submodels in the AAS

(6) Metamodel references

The following table contains the essential AAS model elements in alphabetical order. The elements are each assigned a reference within this document, which refers to a chapter in which the element and its use is described. This should be helpful to find explanations for individual elements of the AAS meta model.

Model elements	Explanations and references
Administrative Information	Administrative information are version and revision attributes of the identifiers Used in chapter 5.4
Asset	Used in chapter 5.4
AssetAdministrationShell	Used in chapter 5.4
AssetInformation	Used in chapter 5.2
AssetKind	Used in chapter 5.6
Capability	Used in chapter 4.5, 4.8
ConceptDescription	Used in chapter 5.8
Entity	Used in chapter 5.5
Capability	Used in chapter 4.5, 4.8
File	Used in submodel "Documentation"
hasDataSpecification	Used in chapter 5.8
HasKind	This attribute describes whether a model element is considered an instance or not. Assets can also be modelled as types, other model elements are marked as templates if they are not to be instances. These are then templates from which instances can be derived without observing the strict type-instance derivation rules.
	Used in 5.3
HasSemantics	Implemented as semanticId and used in chapter 5.3 as well as chapters 4.3, 4.5, 4.7
Identifiable	Used in chapter 5.2
Identifier	IRDI, IRI, Custom Identifier are values of the global identifier type "idType".
Instance	Used in chapter 5.6
ModellingKind	Used in chapter 5.6 and in chapters 4.3 and 4.4
Operation	Used in chapter 4.5, 4.8

OperationVariables	Used in chapter 4.5, 4.8
Property	Used in chapter 4.3 and 4.7
Range	Used in chapter 4.3, 4.7 and 5.6
Referable	Used in chapter 5.2
ReferenceElement	Used in chapter 5.5
RelationshipElement	Used in chapter 5.5 and 5.7
Submodel	in almost all chapters
SubmodelElementCollection	e.g. in the submodel "TechnicalData" chapter 4.3
Template	Used in chapter 5.9 and 5.6
Type	Used in chapter 5.6
View	Used in chapter 4.3 and 4.4

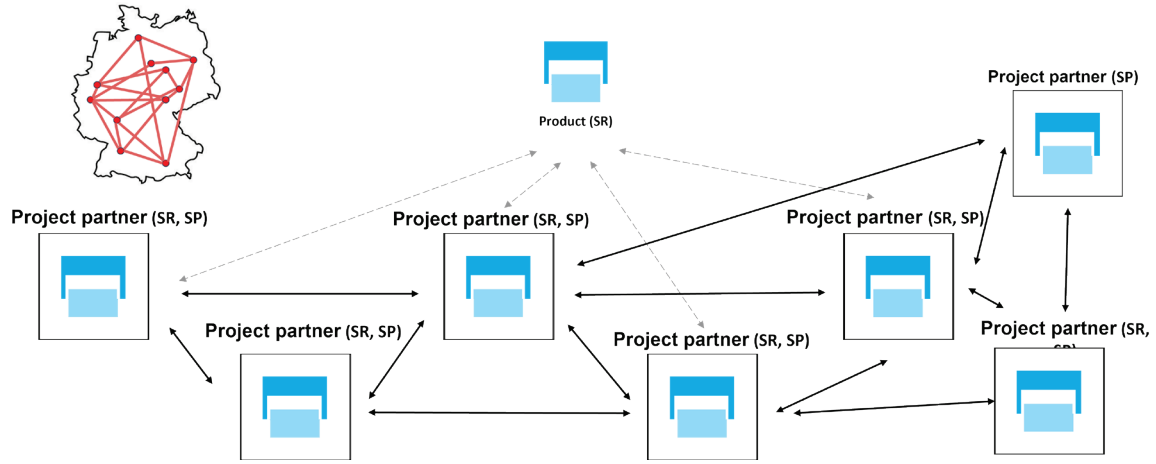


Figure 7-1: The demonstrator described in this document is part of an emerging networked Germany-wide I4.0 demonstrator.

(7) Summary and outlook

This document uses an exemplary plant to show how the modelling elements of the AAS can be used. It aims to support the description of typical components of production plants. This plant is a demonstrator that shows application scenarios for the AAS and its interoperability in a network of demonstrators that is being created throughout Germany (Figure 7-1).

After an introductory presentation of the modelling principles (chapter 3), two separate chapters describe the concrete modelling proposals (chapter 4) of selected components of the plant and essential modelling concepts (chapter 5).

Chapter 6 serves as a register if you want to get information about a single model element.

The modelling proposals have all been implemented with the AASX Package Explorer and are available as an AASX file at http://liabro-ker.ddns.net:51001/.

During the work on this document, a number of questions arose which were discussed with the relevant WGs of WG1 of the Platform I4.0. Some were included in the Q&A list.

Bibliography

[PLA17]	Plattform Industrie 4.0: Beziehungen zwischen I4.0-Komponenten - Verbundkomponenten und intelligente Produktion: Fortentwicklung des Referenzmodells für die Industrie 4.0-Komponente SG Modelle und Standards, BMWi Public Relations, June 2017, URL: https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2017/Juni/Beziehungen_zwischen_I4.0_Komponenten/Beziehungen-zwischen-I4.0-Komponenten-zvei.pdf.
[PLA20]	Federal Ministry for Economic Affairs and Energy (BMWi) (publisher): Details of the Administration Shell - Part 1: The exchange of information between partners in the value chain of Industrie 4.0; Version 3.0 RC01 Federal Ministry for Economic Affairs and Energy (BMWi). Plattform Industrie 4.0, Berlin 2020. URL: https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/Details-of-the-Asset-Administration-Shell-Part1.html
[I40Sp18]	Federal Ministry for Economic Affairs and Energy (BMWi) (publisher): I4.0 Sprache. Discussion paper Plattform I4.0. April 2018.
[FES01]	Festo Didactic SE: Module Drilling, URL: https://www.festo-didactic.com/de-de/lernsys-teme/mechatronische-systeme-mps/projektbaukaesten/komponenten-module-pro-jektbaukasten/modul-bohren.htm?fbid=ZGUuZGUuNTQ0LjEzLjE4LjcxMC4zOTcx (as at: 22.02.2021).
[FES02]	Festo Didactic SE: Module Band, URL: https://www.festo-didactic.com/de-de/lernsys-teme/mechatronische-systeme-mps/projektbaukaesten/komponenten-module-pro-jektbaukasten/modul-band.htm#prd_det_accessories (as at: 22.02.2021).
[WAG01]	WAGO Kontakttechnik GmbH & Co. KG: Controller PFC200, URL: https://www.wago.com/de/sps/controller-pfc200/p/750-8212 (as at: 22.02.2021).
[IGUS01]	IGUS: Technical D-± DC, EC/BLDC Motor Control Manual V2.3 (Manual dryve D1 EN.pdf))
[IGUS02]	IGUS: dryve D1, ST- DC- EC/BLDC-Motor Control System Manual V2.3 ((IGUS, DOM May 2020))
[IGUS03]	IGUS: stepper motor Manual, MOT-AN-S_EN_202003#1, May 2020
[DCI20]	Federal Ministry for Economic Affairs and Energy (BMWi) (publisher): Describing Capabilities of Industrie 4.0 Components. Discussion Paper Plattform I4.0. 2020
[DNI20]	Federal Ministry for Economic Affairs and Energy (BMWi) (publisher): ZVEI Digital Nameplate for industrial equipment. Discussion paper Plattform I4.0. 2020, URL: https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/Submodel_Templates-Asset_Administration_Shell-digital_nameplate.pdf?__blob=publicationFile&v=2
[DTD20]	Federal Ministry for Economic Affairs and Energy (BMWi) (publisher): ZVEI Generic Frame for Technical Data for Industrial Equipment in Manufacturing (Version 1.1). Discussion paper Plattform I4.0. 2020, URL:
[DD20]	Bundesministerium für Wirtschaft und Energie (BMWi) (Publisher): ZVEI Minimum requirements for the Handover documentation from the manufacturer to the operator based on the VDI 2770 specification. Discussion paper Plattform I4.0. 2020

Appendix

Identifiers

Table A-1 illustrates the assignment of identifiers for submodels, assets and AAS. Instances: generate and the IRI domain of the asset manufacturer for AssetId and AASId Type: IRI of the asset domain/ evt substructures /ids/AAS or asset/Product type name

Component	Asset ID	AAS ID
P&P Type	www.ovgu.de/ifat/lia/ids/asset/PickAndPlaceStation/Type	www.ovgu.de/ifat/lia/ids/VWS/PickAndPlaceStation/Type
P&P Instance	www.ovgu.de/ifat/lia/ids/asset/PickAndPlaceStation/Instance	www.ovgu.de/ifat/lia/ids/VWS/PickAndPlaceStation/Instance
Engine_Type	www.igus.de/ids/asset/stepper motor MOT-AN-S_type	www.igus.de/ids/VWS/stepper motor MOT-AN-S_type
Motor_Instance_1	www.igus.de/ids/asset/stepper motor MOT-AN-S_instance_1	www.igus.de/ids/VWS/stepper motor MOT-AN-S_instance_1
Motor_Instance_2	www.igus.de/ids/asset/stepper motor MOT-AN-S_instance_2	www.igus.de/ids/VWS/stepper motor MOT-AN-S_instance_2
Motor_Instance_3	www.igus.de/ids/asset/stepper motor MOT-AN-S_instance_3	www.igus.de/ids/VWS/stepper motor MOT-AN-S_instance_3
Workstation_type	www.ovgu.de/ifat/lia/ids/asset/workstation_type	www.ovgu.de/ifat/lia/ids/VWS/workstation_type
rotation_module_type	www.festo-didactic.com/de-de/lernsysteme/mechatronische-systeme-mps/projektbau-kaesten/komponenten-module-projektbaukasten/asset/rotation_module_type	www.festo-didactic.com/de-de/lernsysteme/mechatronische-systeme-mps/projektbau-kaesten/komponenten-module-projektbaukasten/VWS/rotation_module_type
conveyor_module_type	www.festo-didactic.com/int-en/learning-systems/mps-the-modular-production-system/mps-modules/asset/conveyor_module_type	www.festo-didactic.com/int-en/learning-systems/mps-the-modular-production-system/mps-modules/VWS/conveyor_module_type
PLC_workstation_type	www.ovgu.de/ifat/lia/ids/asset/PLC_workstation_type	www.ovgu.de/ifat/lia/ids/VWS/PLC_workstation_type
workstation_instance1	www.ovgu.de/ifat/lia/ids/asset/workstation_instance1	www.ovgu.de/ifat/lia/ids/VWS/workstation_instance1 68339440
drilling_module_instance1	www.festo-didactic.com/de-de/lernsysteme/mechatronische-systeme-mps/projektbau-kaesten/komponenten-module-projektbaukasten/asset/drilling_module_instance1	www.festo-didactic.com/de-de/lernsysteme/mechatronische-systeme-mps/projektbau-kaesten/komponenten-module-projektbaukasten/VWS

stance1	www.festo-didactic.com/int-en/learning-systems/mps-the-modular-production-system/mps-modules/asset/conveyor_module_instance1	www.festo-didactic.com/int-en/learning-systems/mps-the-modular-production-system/mps-modules/VWS/conveyor_module_instance1
PLC_workstation_instance1	www.wago.com/de/sps/controller-pfc200/p/750-8212/asset/PLC_workstation_instance1	www.wago.com/de/sps/controller-pfc200/p/750-8212/VWS/PLC_workstation_instance1
Transport	www.igus.de/robolink/parallel-kinematik/instance1	www.igus.de/robolink/parallel-kinematik/VWS

Table A-1: Assignment of identifiers for submodels, assets and AAS

Instances

Generation and the IRI domain of the asset manufacturer at AssetId and AASId

Submodel	idShort	Exemplary id	semanticId
Name tag	Nameplate	www.ovgu.de/ifat/lia//ids/ass/PickAndPlaceStation/sm/4343_5072_7091_3242	http://admin-shell.io/zvei/nameplate/1/0/nameplate
	Documentation	ditto	http://admin-shell.io/vdi/2770/1/0/Documentation
	TechnicalData	ditto	http://admin-shell.io/sandbox/SG2/TechnicalData/Submodel/1/1
Encoder	Encoder	www.ovgu.de/ifat/lia/ids/sm/5284_3152_2002_0368	http://admin-shell.io/Encoder
Brake	Brake	www.ovgu.de/ifat/lia/ids/sm/6504_0122_9002_4971	http://admin-shell.io/Brake
OperatingData	OperatingData	www.ovgu.de/ifat/lia/ids/sm/4244_0122_9002_6259	http://admin-shell.io/OperatingData
BoM	BoM	www.ovgu.de/ifat/lia//ids/ass/PickAndPlaceStation/sm/BoM/4343_5072_7091_3242	http://example.com/id/type/submodel/BoM/1/1
all others	individual	individual	

Package Explorer and AASX files

The modelling tool used is AASX Package Explorer 1.9.8.1 Build date 18.5.20.

The AASX Package Explorer is used to easily and quickly view and create AAS that are currently compliant with version 2.0.1 of the AAS metamodel. The tool enables AAS to be serialised in XML and JSON formats and read in for editing. Thus, concept descriptions with ECLASS IRDIs are automatically created and referenced. With import and export functions for e.g. BMEcat, AutomationML or OPC UA, other data formats and relevant company data can be integrated quickly. The AASX Package Explorer can be downloaded free of charge at github.com/admin-shell. The tool is an open source implementation licensed under the Eclipse Public License 2.0 (EPL-2.0).

Installation Package Explorer: https://github.com/admin-shell-io/AASX-package-explorer

Screencasts: http://admin-shell-io.com/screencasts/

AAS examples: http://liabroker.ddns.net:51001/

Authors

Alexander Belyaev, Otto von Guericke University

Christian Diedrich, Otto von Guericke University

Daniel Espen, Fraunhofer IESE

Benno Lüdicke, Expleo Germany GmbH

Torben Miny, RWTH Aachen University

Anja Mrosowski, Mitsubishi Electric Europe B.V.

Matthias Müller, Mitsubishi Electric Europe B.V.

Jörg Neidig, Siemens AG

Stefan Pollmeier, ESR Pollmeier GmbH

Johannes Reich, SAP SE

Pascal Rübel, Technology Initiative SmartFactory KL e.V.

Manuel Sauer, SAP SE

Tizian Schröder, Otto von Guericke University

Olaf Ulrich, Siemens AG

Chris Urban, Otto von Guericke University

Bernd Vojanec, Wittenstein SE

Michael Wieczorek, Siemens AG

This publication was created by the working group "Reference Architectures, Standards and Norms" of Plattform Industrie 4.0.