Accelerating Product Development: The Effort of Asset Administration Shells for aggregated Systems

Denis Ritz, Malte Becker*, Jürgen Weber and Katharina Schmitz*

Technische Universität Dresden, Germany
*RWTH Aachen University, Germany
E-mail: denis.ritz@tu-dresden.de, malte.becker@ifas.rwth-aachen.de, mailbox.fluidronik@tu-dresden.de, katharina.schmitz@ifas.rwth-aachen.de

Abstract

When developing and revising products, a wide variety of engineers usually work together. Such heterogeneous development teams often work on the same product but with different models. Typically, each team works on its own partial version of a specific product.

At the same time, finding the right components for each subsystem is another challenge. The technical data sheets are often only available as PDF files. This makes it difficult to access the data they contain by machine. Furthermore, there is no standardized designation of product properties across the various manufacturers.

The joint research project "Fluid 4.0" is working on standardized semantics for fluid power technology. Among other things, existing standards such as "ECLASS" are utilized. This means that a certain basic stock of properties is already available. At the same time, the list of properties is analyzed initially, and missing features are detected.

This publication will show how these extended semantics can be used to create administration shells for aggregated systems. To this end, selected aspects of the specific extension of the semantics are explained. During the project, it is planned to submit this extension as a proposal to the standardization committees in order to provide a coordinated, machine-readable database for development processes.

An important point here is the linking of the individual components to form an aggregated system. For this purpose, a proposal for the creation of asset administration shells for aggregated systems is explained. The versioning and access control mechanisms of the administration shells make them ideal as a single source of truth for the collaborative development of complex, mechatronic systems. All of this is discussed using simplified examples.

Keywords Asset Administration Shell (AAS), System Development, Semantic

1 Introduction

In the product development process, the layout of the hydraulic powertrain is usually selected at the beginning. Choosing the layout is not a trivial part of this process, but it is usually much more straightforward than choosing the right components. During the development process, it starts with a few requirements at an abstract level of analysis, such as a specific hydraulic power. As the development process continues, decisions are made, such as setting the system pressure to a certain level. This then leads to more and more specific requirements for each component. This is where the tricky part begins: searching for components that fit the specified requirements. Among others, this is the widely varying quantity and quality of product data, which is generally difficult to compare across different manufacturers.

If a semantic search, which is possible across manufacturers, existed for this purpose, this step would be significantly simplified. This could be made possible with asset administration shells and a correspondingly specified submodel (SM) for the components.

In view of the wide range of specialist areas addressed, this paper attempts to discuss the various topics in such a way that they are not only comprehensible to the individual experts, but also to people from outside the field. To conclude, the hydraulic engineer must be able to engage in meaningful discussions with the computer scientist and vice versa.

This paper presents the potential of the Asset Administration Shell to accelerate product development. Therefore, section 2 references the related work. After that, section 3 introduces the Asset Administration Shell, followed by a description of relevant terminology and standards in section 4. Section 5 integrates the AAS into the context of the product life cycle, after which the contribution is summarized in section 6, and potential benefits are highlighted.

2 Related Work

AAS has already been used in scientific work in the context of fluid power. In the publications [1–3], the AAS is studied in the context of the simulation-based development process. Use cases such as the requirements-based search for AAS in repositories and the parametrization of simulation models are initially considered. The dissertation [4] investigates the use case of commissioning and investigates the extent to which fluid power systems can be commissioned automatically.

In the research projects [5], [6], aspects for the development of AAS and an OPC UA specification for pumps [7] were shown. These two research projects, as well as project [8], are being worked on at the same chair. The latter project dealt with the OPC UA specification for compressed air stations [9]. It can be assumed that significant preliminary work for the two OPC UA specifications [7], [9] was carried out as part of these projects. Unfortunately, the authors only have insufficient sources for the content created within the aforementioned projects.

The papers [10–12] deal with the comparison of different semantics, the mapping of a source semantics into a target semantics, as well as simplified interoperability. In [13], a search algorithm for mapping proprietary features to ECLASS properties is described.

In addition to its application in fluid power, the AAS is already being used in various branches of industry. In electrical engineering, for example, the AAS was used in a control cabinet demonstrator to enable the automatic calculation of the product carbon footprint (PCF), whereby components from various manufacturers were included [14]. In addition, the AAS is also used in research activities in other areas, e.g., in the manufacture of electric motors [15] and water supply [16].

3 Asset Administration Shell

The Asset Administration Shell (AAS) is a virtual representation of an asset in the context of Industry 4.0 (I4.0), see Figure 1. In general, assets are physical, digital, or intangible objects that represent values for one or more stakeholders. An example of an asset in fluid power is a pump or a valve that is represented virtually with an AAS. In the context of the AAS, assets are described by SMs and submodel elements (SME) which represent properties. A SM is a set of

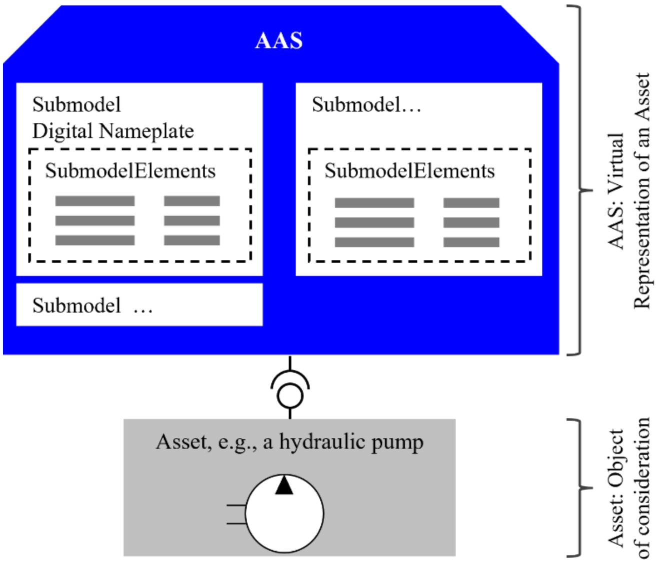

Figure 1: Exemplary AAS according to [20]

properties, collections, methods, etc. for a specific use case. SMs can then be standardized as templates to consider a range of relevant properties required for specific use cases. The SM template “IDTA 02006-3-0: Digital Nameplate for Industrial Equipment” [17] for an AAS represents a digital data sheet of the asset, while the SM template “IDTA 02005-1-0: “Provision of Simulation Models“ [18] can be used to exchange simulation-based properties and models of the asset. Another example for a SM template is “IDTA 02011-1-0: Hierarchical Structures enabling Bills of Material,” [19] which can be used to describe hierarchical structures of fluid power systems. Both SM and SMEs can be annotated with semanticIds, which will be discussed further in section 4.2. [20] Assets and thus also the corresponding AAS are distinguished in type and instance [21]. A type asset describes a product or component's general, abstract definition and is usually created during the development phase. An instance asset is a concrete, physical realization of a type asset. In production, instance assets are created based on the specifications of the type asset and linked to product-specific data (e.g., production, logistics, and test data). As a rule, there is a relationship between instance assets and a type asset. However, every instance asset does not need a corresponding type asset. [22]

In the context of fluid power, the type refers to a pump variant from the manufacturer's range, while an instance is a specific, individually available pump that has been installed in a specific application, for example. This work focuses on type AAS, which is decisive for the development process.

An AAS can represent rudimentary components, such as a “simple” non-return valve, the drive of a partial function of a production machine, but also the hydraulic ram of a deep-drawing press, a complex machine, as well as a hydraulic excavator, or even an entire production plant, such as a rolling mill with a connected press line for the production of semifinished products for the automotive industry.

4 Relevant Terminologies and Standards

According to the authors of [23], the AAS could have a huge impact on businesses and business models in general. The AAS can act as the interface between "our physical" world and the "digital" world. So, if a semantic is created for a specific SM, this is the vocabulary which has to be used in every domain. Therefore, naming the individual elements should follow a meaningful convention, from the point of view of every domain involved. Therefore, in the following existing standards, dictionaries, and semantics are reviewed. In the context of semantic descriptions, terms such as data, information, and knowledge are often used misleadingly [24]. To create a uniform understanding, this paper uses the definitions made by North in [25], which is explained in the following paragraph.

Data is formed from characters using syntax rules. It consists of arbitrary and uninterpreted character strings, such as any sequence of numbers and becomes information via semantics. Information is, therefore, data that has a context of meaning. It is created by linking information with current information or information stored in the past. Knowledge is characterized by the fact that it is shaped by individual experience, context-specific and personal. This hierarchy of terms is illustrated in Figure 2. [25]

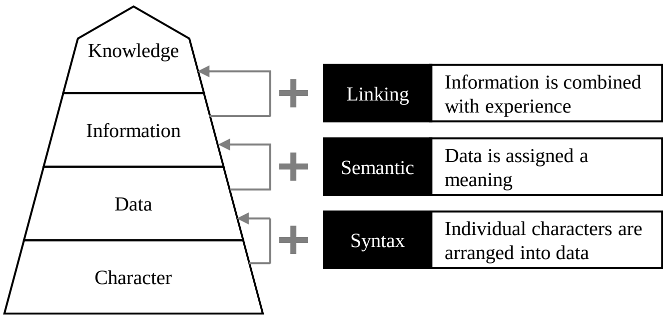

Figure 2: Conceptual hierarchy of data, information and knowledge according to [25], [26]

4.1 ISO 1219

The ISO 1219 standard [27] or the corresponding national implementations of the standard, like DIN ISO 1219 [28], represents the graphic symbols of all fluid technology components. Every hydraulic engineer and every service technician understands these symbols and can use them to identify the function of a component and a system. At the same time, the standard provides a structural classification of all standard components. Therefore, the contents of this standard should have a strong influence when defining the semantics, product groups of hydraulics and the description for the topology of systems.

Figure 3 shows the relevant component classifications as defined in ISO 1219-1[27]. It is striking that the standard explicitly divides directional, pressure, and flow control valves into "normal" and proportional valves. The vertical lines in the illustration indicate the related product groups.

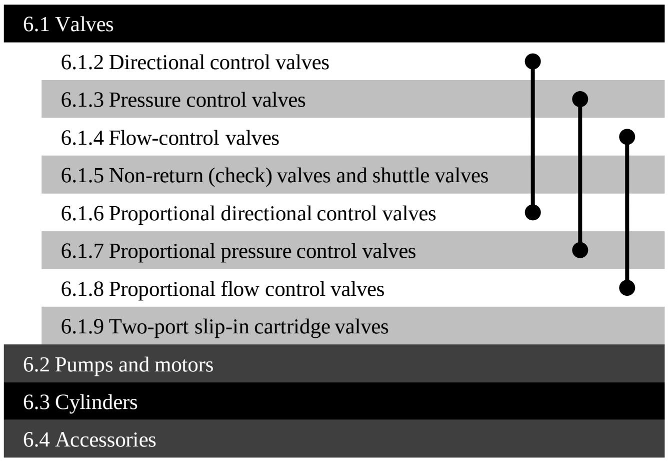

Figure 3: Component classification from the table of contents of ISO 1219-1 [27]

From the authors' point of view, this duplication between "normal" and proportional raises whether proportionality is not more than a property of the valves and, therefore, does not require a separate subdivision.

Another point that stands out is the category "Cartridge valves," which is the only place in the standard where the design of a valve is explicitly mentioned. These valves can also be described purely in terms of their function. All valves in this group can be classified according to their function, either as 2/2-way valves or as pressure-reducing or limiting valves.

ISO 1219-2 [29] explains the rules to be followed when presenting a hydraulic diagram. This is explicitly required in section 5.2 of this standard, which is necessary in order to be able to clearly identify the specific connections on a valve. Figure 4 shows this for a simple system.

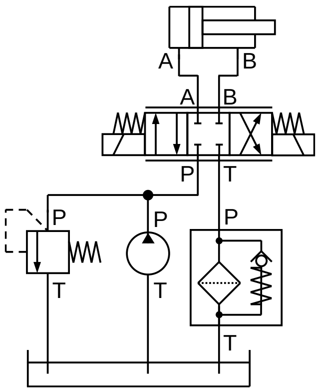

Figure 4: Example of the hydraulic scheme for a small aggregated hydraulic system

The last part of the standard, ISO 1219-3 [30], explains so-called symbol modules. These modules generally describe complex products that are offered by manufacturers of hydraulic components as ready-to-sell products. This often applies to the valve sections of mobile hydraulic systems. Figure 5 shows the module symbol for a LUDV valve disk from Bosch Rexroth [31].

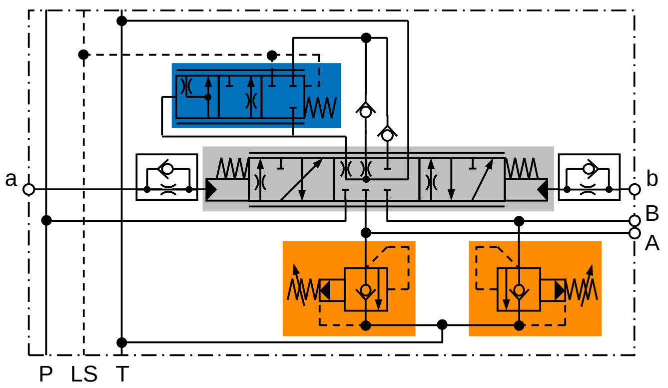

Figure 5: Module Symbol for the valve section of the LUDV system by Bosch Rexroth, based on [31]

This and similar valve systems are frequently used in construction machinery. Figure 5 shows that the LUDV valve itself is already a subsystem consisting of several components. The main valve, which controls the direction and speed of the connected actuator, is shown in gray. The pressure compensator is shown in blue. This valve system consists of at least these two components. The valves shown in orange are optional additional functions for secondary pressure limitation and post-suction, which often can be found in mobile machines.

For module symbols, only the connections need to be marked on the outside, but a frame must be used to delimit the module. This example also makes it clear that semantics must be flexible enough to describe valves that are not directly represented in the standard, such as the gray marked 7/3 directional valve.

4.2 Semantic Dictionaries

A unique semantic understanding of data is a prerequisite for implementing I4.0 use cases. According to Wang [32], this is referred to as “semantic interoperability” and is made possible using a unique semantic reference model.

However, data is often available in proprietary formats, e.g., based on manufacturer-specific data descriptions. An example illustrates this problem: One manufacturer's data sheet uses pressure-related properties like “Max continuous pressure”, “Peak pressure”, “Minimum Inlet Pressure”, “Maximum Inlet Pressure” [33] to describe a hydraulic pump, while another manufacturer's data sheet describes the hydraulic pump with pressure-related properties like “Nominal pressure,” “Minimal pressure,” and “Maximum pressure” [34]. This example alone demonstrates the potential added value of the algorithm presented in [13].

In addition to the challenge of the overarching comparability of components from different manufacturers, imposed by the need-to-know which property is related to the other, these different descriptions stand in the way of semantic interoperability. To counteract this challenge, there are semantic dictionaries such as ECLASS [35] or IEC 61360 CDD [36]. Semantic interoperability is given, under the assumption that corresponding stakeholders are committed to a semantic dictionary. In general, different semantic dictionaries are not interoperable with each other [11]. ECLASS itself is a non-profit association, and the steering committee is largely made up of companies from the pharmaceutical industry, automation technology and fluid power [37].

ECLASS provides defined properties for various domains, including fluid power. Within a domain, ECLASS is hierarchically structured. For example, an axial piston pump with variable displacement and bent design (product variant) belongs to the product type of axial piston pump. This product type, in turn, falls under the broader product family of pumps within the fluid power domain, which is shown in Figure 6. ECLASS properties or classifications are assigned an IRDI (International Registration Data Identifier), enabling unique identification. For instance, the shown product variant is uniquely identified by the IRDI 0173-1#01-AGW107#006.

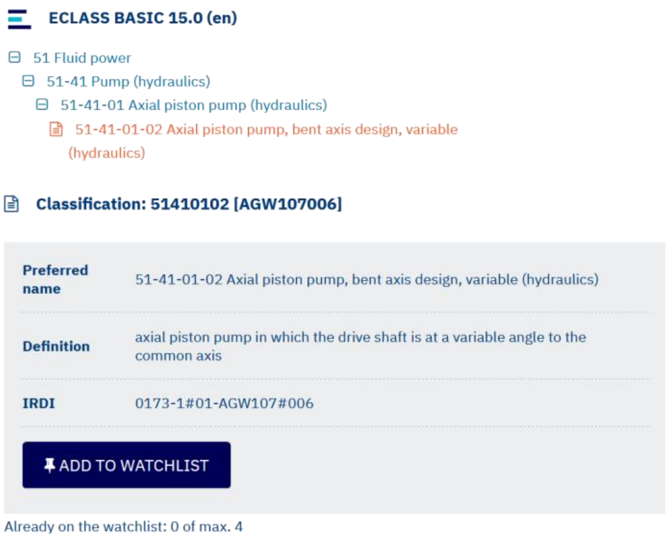

Figure 6: Exemplary Product Classification in ECLASS [35]

Furthermore, properties are assigned to the product variant. For example, the property “nominal pressure” with the IRDI 0173-1#02-ABL782#001 is assigned to the product variant. In the context of I4.0, the IRDI can be used as a semanticId [20]. In addition to the example above, ECLASS has numerous other product families, such as motors and valves with respective product types and product variants from fluid power.

ECLASS version 15 already has numerous general properties for fluid power. Missing properties are continuously being added, for example, the property displacement was not yet defined as a central feature for describing pumps in the previous version. In addition, duplications can still be found in ECLASS, such as the property “set pressure”, which appears under both IRDIs 0173-1#02-ABC506#003 and 0173-1#02-AAZ819#003, even though they represent the same concept. Discrepancies are to be expected if manufacturers refer to different semantics for this property.

Besides ECLASS, there is also IEC 61360 CDD [36], which serves as a semantic dictionary with a strong focus on electrical engineering. The properties defined in IEC are also assigned unique identifiers, such as the "actuating force" with the identifier AAE932-005. This characteristic can be expanded into a complete IRDI by specifying the standard source: 0112/2///61360_4#AAE932-005 [38]. In contrast to ECLASS, however, IEC does not provide specific semantic definitions for fluid power components.

4.3 Semantic Modelling

The publication [39] provides an overview of several different languages for semantic modeling. The authors of [39] also refer to ISO 15926 in their publication. The ISO 15926 standard describes the semantic modeling of data and integration during the product life cycle in detail. Part 2 of this standard [40] covers data modeling, which is particularly relevant for the development of semantic models, from the point of view of the authors of this paper. In [41], it is shown, among other things, how an AAS must be structured to comply with the rules of ISO 15926.

The Eclipse Semantic Modeling Framework (ESMF) [42] offers a suitable option for creating semantic models for administration shells. The Eclipse Foundation itself has been active since 2001 and has set itself the goal of providing open-source software for many areas of technical development. The members of the foundation cover many different industries, with many well-known, large I4.0 companies. [43].

The ESMF's basis is the Semantic Aspect Meta Model (SAMM), which forms the basis for the semantic description of SMs in so-called aspect models. These models express domain-specific semantics by linking structural elements, relations, namespaces, and verified concepts.

Figure 7 shows an example of modeling technical data according to the suggested structures of the SAAM [44]. In addition to integrating the ECLASS IRDI, the property's minimum and maximum value range and the properties' underlying unit are also considered. This comprehensive structure enables companies to prepare their proprietary data uniquely and semantically consistently. This significantly improves interoperability between different systems and data utilization in different contexts.

4.4 schema.org

The schema.org initiative provides a vocabulary for structuring data. The main drivers of this initiative are large digital companies such as Google, Microsoft, and Yahoo, to name just a few. The vocabulary can be used to represent entities of a wide variety of objects and link them together. According to schema.org, its vocabulary is already used by more than 45 million websites. [45]

The vocabulary of schema.org offers to categorize products hierarchically using the types "ProductGroup", "Product" and "ProductModel". Figure 8 shows a possible structure for classifying an axial piston pump with inclined axis, which also corresponds to the example in Figure 6.

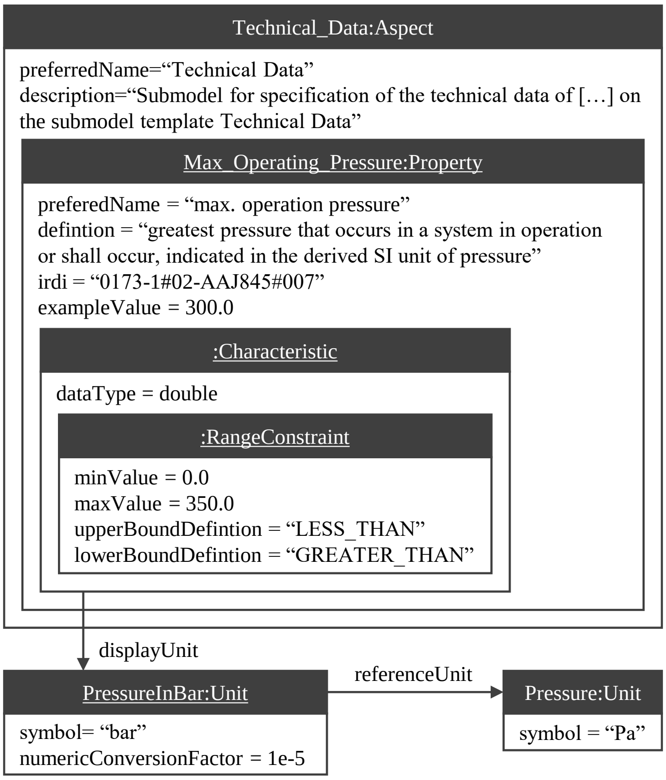

Figure 7: UML-like Object Diagram for a Semantic Modeling using the terms of the SAAM [44]

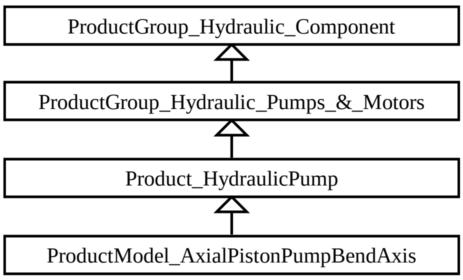

Figure 8: UML-like Class Diagram for structuring an axial piston pump in bent axis design

The hydraulic components correspond to the entire content of ISO 1219-1 [27]; the pumps and motors group is equivalent to section 6.2 of the standard. The symbol for a variable displacement pump can be found in subsection 6.2.1 of the standard. At this point, the standard does not differentiate between the possible designs of a pump, as this has no influence on the defined function of the component. In the schema.org description, a product model can also be derived from a product and thus represents the fact that every axial piston pump in bent axis design is a pump, but not every pump is an axial piston pump in bent axis design.

Compared with the classification according to ECLASS in Figure 6, some small differences are recognizable. For example, ECLASS already differentiates between pumps and motors at a high level and then displays the pump type in a detailed manner over several steps. Likewise, adjustability is not used as a property of the pump as discussed in section 4.1, but as a distinction between several types.

When creating schema.org-compliant semantics, the question also arises, whether it makes sense to explicitly differentiate between pumps and motors, as the majority of all pumps can also be used as motors and vice versa. Whereby a "real" component is usually optimized for its primary purpose, which also must be readable in data of the component. This allows the number of product groups to be reduced, as long as the various products have comparable interfaces. From the authors' point of view, it makes sense to combine all valves in a "ProductGroup_Valves" and to provide this product group with subgroups for each configuration of the number of connections. This means that such a structure must be suitable for all components and flexible and expandable for components that do not fit into the standard or are not yet included. An example of this is the so-called "hydraulic transformer" from INNAS BV, which is described in [46].

As a final note in this section, it should be mentioned that the schema.org vocabulary, with "IndividualProduct", contains an equivalent for the instance of an AAS.

4.5 Structured Information in Communication protocols

Another important aspect of I4.0 is the connectivity of systems. In [47], the possibilities for communication between a physical asset and its AAS, among other things, are summarized. The EDC (Eclipse Dataspace Connector) project [48] attempts to create a simple way of combining the various possible communication channels into an AAS with this software project. The authors will therefore discuss what they consider to be the most relevant communication standards, which are OPC UA and MQTT.

The origins of the OPC UA communication protocol date back to the 1990s. The main drivers at the time came primarily from automation technology. Even back then, there was a strong focus on interoperability. The first version of the OPC UA standard was released in 2006. [49]

The OPC Foundation has now published over 150 specifications. Within these specifications, basic properties are defined that build on and reference each other. In the "Data access" specification [50], for example, a proposal is made on how physical units should be represented. The advantage of this is the unambiguous assignment of a specific value based on the SI base units. This is a clear advantage over purely semantic definitions, which only describe the physical unit of a value within a text. One other core specification is "Devices" [51], it already contains a definition for "VendorNameplate", which is similar to the IDTA SM Template "Digital Nameplate for industrial equipment" [17]. The recently published specification for "Powertrain" [52] contains more component definitions for the elements of electro-mechanical drives. Here, explicit attention is paid to object orientation for structuring; each derived class in turn contains objects which, if the link is followed, ultimately lead to formally described properties. Whenever possible, these properties should be linked with a semanticId, for example ECLASS.

Another important specification in this context is "Pumps and Vacuum Pumps" [7]. This specification defines many important properties and statuses of a pump. The specification explicitly references the respective physical unit of values. However, no distinctions are made regarding the design, but the different hydraulic connections are modeled. Based on this document, a proposal for the design of hydraulic systems has already been created in [53] using the example of a mobile excavator. The concrete modeling of the hydraulic connections and their linking is also used there. Another advantage of OPC UA is its capability to provide an automation interface. This was realized during the joint research project "Bauen4.0" [54]. As an extension of this system, a 3D safety system for a mobile excavator was implemented, which is described in [55].

The MQTT protocol was originally developed to reliably send messages in networks with low bandwidth and high latency. The "Technical Committee" of MQTT is mainly made up of software companies. [56]

The possibility of dividing the individual messages into different "topics" allows the messages to be structured and filtered for the corresponding recipients [57]. Although the protocol offers a good possibility for structuring, it has no explicit rules for structuring, for example which names are assigned and how. There is no recommendation from the "Technical Committee" as to how "topics" should be further subdivided. Even if this offers an exciting opportunity for distributed communication in an I4.0 network, few specific points can be derived for structuring. However, the division of messages into "topics" represents an interesting possibility for setting up field communication. This could be interesting when setting up field device communication with structure based on the semantic description.

From the authors' point of view, however, OPC UA is much better suited for this, as this protocol can already map a clear, unambiguous, semantically describable structure through the specifications.

4.6 Modeling and Model-Integration

In the context of this topic, two standards should be mentioned. The first one is the FMI/FMU standard, which allows integrating of nested simulation models in different tool. The other one is the Modelica language, which is suited for every domain, as long as the model can be described with ordinary differential equations. Both allowing to combine different model domains in one simulation environment. FMI/FMU is a way of exchanging simulation models between different tools. The respective simulation part model is packaged as a container, so to speak. [58]

These containers are described via their interfaces. It is important that the type of modeling is taken into account: what are model inputs, what are model outputs. The containers conceal the internal structure. There are now 235 different programs that support at least one version of the FMI standard, 69 of which support the current version 3.0 [59]. The consortium behind FMI consists largely of simulation software manufacturers and large companies that use simulation in the automotive, automation and aviation sectors [60].

In summary, the FMI standard enables the simple cross-domain use of simulation models. However, the internal structure of the respective model is hidden, which is often in the interest of the providing companies. This means that the interfaces must be precisely described in order to be able to work with these models.

The object-oriented modeling language Modelica makes it possible to model complex technical systems using equations, regardless of the physical domain. In contrast to other simulation tools, which are typically geared towards a specific problem domain, e.g., for multi-body systems, electrical circuits or fluid dynamics. The Modelica language itself is maintained and further developed by the Modelica Association. In addition to simulation software manufacturers that distribute Modelica-compliant programs, the association also brings together several companies from the fields of automation technology and automotive engineering. The association is also involved in the development of the FMI/FMU standard. [61]

Two another relevant features of the Modelica language are that it explicitly supports the use of physical units based on SI-units and conversion to other units. Models of a system are based on components, which are linked via connectors. For hydraulic components those connectors could match exactly to the fluid ports and the electric pins. Modelica itself is also mentioned in [18] as an option for providing a simulation model from an AAS.

In principle, the Modelica standard libraries already provide many basic building blocks for creating complex simulation models. However, the standard library contains just few basic elements that can be used to build hydraulic systems directly. For more hydraulic component-oriented models, these must either be implemented by the user or by using third-party solutions such as the freely available “OpenHydraulics” package [62]. The package itself uses object-oriented, abstracted approaches for the efficient implementation of different but similar components. For example, there is a common base class for different 4/3 directional valves, which contains all the connections for such valves. The valves derived from this then differ, for example, in the linking of the connections in the center position. For “more complex” elements, such as a valve section for an LS system, this element is aggregated from existing models of individual components. The collection of examples in this library already includes some simulation models for hydraulic systems, which are also made up of individual components.

4.7 Conclusion to the reviewed Terminologies and Standards

Compared to Figure 2, each of the topics studied brings aspects of syntax, semantics or links to the hydraulic components and systems. With SAMM (section 4.3), it is possible to introduce syntax and semantics for parameters of components to fully describe those with standardized physical units. Ideally, such semantics are standardized with a semantic dictionary, such as ECLASS (section 4.2). For structuring and comparing components, it would be a good option to structure products in a hierarchical relationship to each other in order to make them easier to compare. Such a product structure can be realized with schema.org (section 4.4) and should be based on the product groups of ISO 1219-1 [27] (section 4.1) for hydraulics. Many of these functionalities are implemented in Modelica (section 4.6). At the same time, Modelica also allows components to be linked in sub-models like “module symbols” in accordance with ISO 1219-3 [30] or a complete hydraulic system to be assembled from individual components in accordance with ISO 1219-2 [29] (both section 4.1). In principle, compatibility and transferability to other standards, such as the Companion Specifications of OPC UA, should be sought wherever possible (section 4.5).

5 Integrating AAS into the Product Data Backbone and Product Life Cycle

The use cases described here address the product lifecycle phases of product planning and development. This is usually also the area in which PDM (product data management) is traditionally used [63]. This means that integration into the development processes also requires a link to the PDM system, and ideally also to the company's CAD program. Each reference in the AAS of the aggregated system should therefore correspond to a (purchased) part or an assembly in the PDM and CAD system. This has a deep impact on the product data backbone of a company. Thanks to the possibility of versioning and revising an AAS, this can also be used as a data basis for system engineering, which can thus be integrated into the typical working method of a PDM system. Figure 9

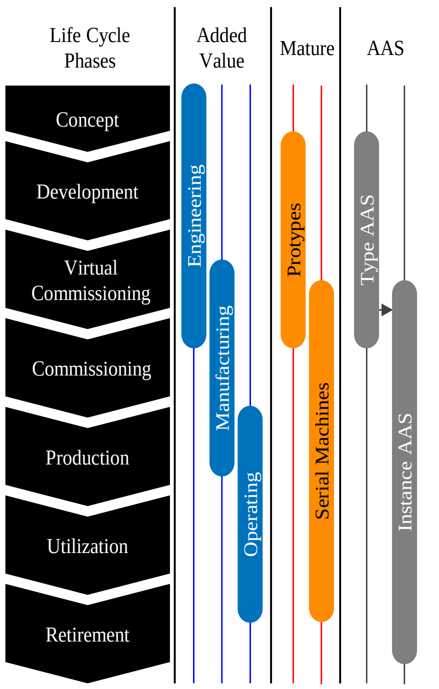

Figure 9: Use of AAS for a specific product during the product life cycle, the maturity of the product and the phases of value creation based on [21], [63–66]

shows the possible contexts of AAS along the product life cycle. For the product life cycle, a comparison was made using the documents [21], [63–66], as each of these documents has a slightly different focus.

Next to the administration shells is the maturity of the product and the value creation that takes place. It shows that the AAS type is relevant to large parts of engineering and can also be used for production planning. A type AAS of an aggregated system could be used to create virtual prototypes based on their linkage. At the same time, the type AAS should continue to mature with physical prototypes to serve as a blueprint for instance AAS, which are automatically derived from the type AAS during manufacturing. Ideally, the data from the manufacturer's ERP (Enterprise Resource Planning) system is linked during derivation. Figure 10 shows a proposal of which types of AAS should be used within a company's data and software architecture.

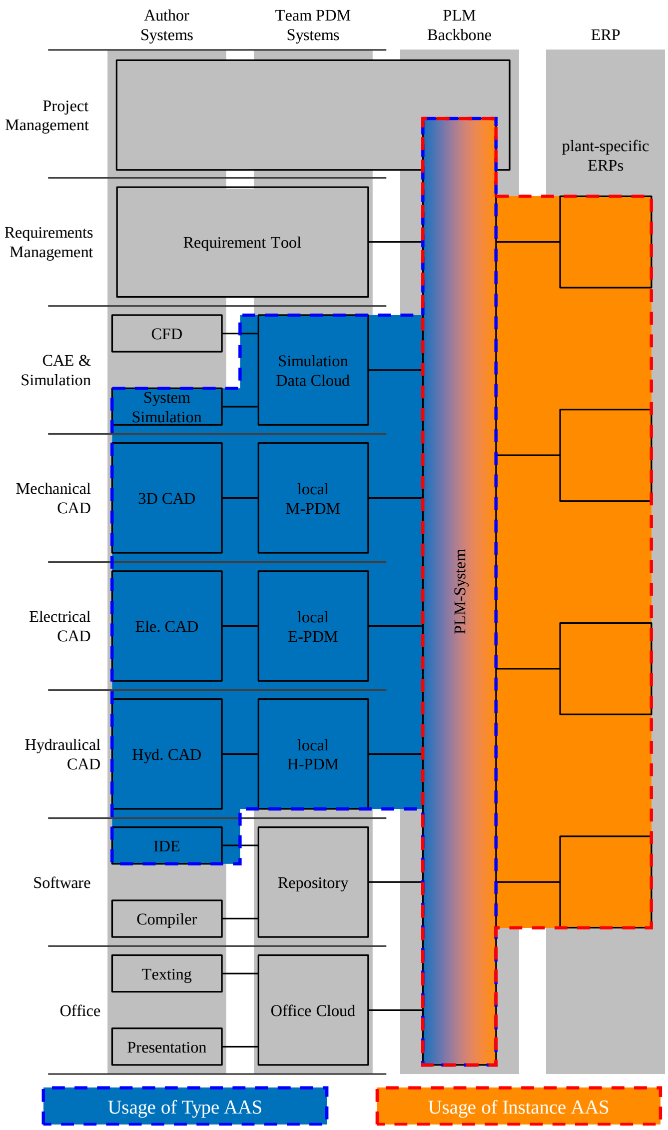

Figure 10: Use of AAS types in the various program systems in an exemplary PLM architecture based on [63]

As shown in the figure, all authoring systems that can map individual components should have a reference to the type shells of the corresponding components. Whether it is the system simulation with lumped parameters, which either loads concrete models of the components from the SM "Provision of Simulation Model" [18] or loads them from the SM "Technical Data for Fluid Components", or the "classic" 3D CAD program which loads the 3D model via the SM "Provision of 3D Models" [67] and automatically creates or updates the SM "Hierarchical Structures enabling Bills of Material" [19] when assemblies are created. This list could be extended further, but essential is that consistent identification of the same component must be ensured in all authoring systems. This could be done with the type AAS of the product as the "single source of truth" during the development process.

As already indicated, an instance AAS, that could be derived from the type AAS, should also be composed by the PLM system with all available data. This is also shown in Figure 10. In this case, the instance shell is created as a "copy" of the type shell. However, this copy receives some additional data, such as its own serial number. It is also possible to link contained parts and assemblies with their specific instance shell, which could be the hydraulic pumps, valves or even the combustion engine of a mobile machine. With this it all relations form parts and subassemblies could be described.

6 Conclusions and potential Benefits

In this paper, a wide variety of documents and templates for the development of semantics for specific products, the classification of products and the topology of systems were reviewed. Most of these documents are largely driven by the automation technology sectors and large digital companies. These documents can also be transferred to fluid power systems. In the opinion of the authors, the following statements should apply to the terminology of AAS for fluid power components and systems:

A <u>hierarchical classification</u> of the individual components, as already intended in ISO 1219-1 and the product classification according to ECLASS. However, the design solution does not play a role in the classification, as is envisioned e.g., in the OPC UA specification for "Pumps and Vacuum Pumps". The hierarchy is compatible with the conventions of schema.org. The <u>principles of object orientation</u> are used for inheritance from an abstract to a concrete description.

The characteristics of each component are <u>uniquely described via an identifier</u> that refers to an <u>SI-compliant physical unit.</u> As in Modelica, for example, this unit is specified to be converted into any unit describing the same physical quantity. The <u>unit's assignment is not purely based on a text entry</u> in the description of the identifier; this can lead to errors if a physical unit is written in different ways.

A <u>component's connections</u> (hydraulic ports, electrical pins, etc.) <u>are part of the elements</u>. By linking the connections, a more complex element can be composed of several sub-elements, so aggregated systems can also be represented. This should also fulfill the requirements of ISO 1219-2 and ISO 1219-3.

From the authors' point of view, this would result in the following benefits.

With this semantic definition, components from different manufacturers can be directly compared based on their characteristic values. This enables a cross-manufacturer semantic search.

These aggregated systems also provide knowledge about the system's internal structure, which can also be used to utilize this structure for further applications. This can be used to derive technical documents such as the hydraulic circuit diagram or the BOM, or to create and parameterize simulation models as part of the development process, or to load the relevant parameters into the machine program code. The permissibility of the connection can also be checked by linking the connections. Errors can thus be detected much earlier in the development process. For instance, this can be the comparison of the burst pressure of a component with the set value of the pressure relief valve or the comparison of the signal type of the sensor and the input of an electrical controller.

When a product is manufactured, automatic parameterization for the machine can be carried out based on the instance AAS of the subcomponents by linking them to the individual instance AAS of the subcomponents. Thus, time-consuming calibration during the commissioning of a series product can be significantly reduced.

These aggregated systems also provide knowledge about the internal structure of the system, which makes the utilization of this structure for further applications possible. This can be employed for example, to derive technical documents such as the hydraulic circuit diagram, or the parts list, or to create and parameterize simulation models as part of the development process, or to load the relevant parameters into the machine program code.

7 Outlook

As part of the Fluid4.0 research project, the authors will attempt to implement the conclusions reached in this work and develop a proposal for a partial model for the technical data of hydraulic components and systems. During the project, this proposal will be suggested to IDTA as the basis for an official partial model.

In addition, the advantages also explained in the context of this work are to be demonstrated using an exemplary software implementation. Figure 11 shows a mock-up of the user interface of this application. Although there are already commercial solutions (e.g. Eplan [68]) for creating system structures, the planned solution primarily offers advantages in analyzing basic systems and component dependencies in the product development process independent of commercial

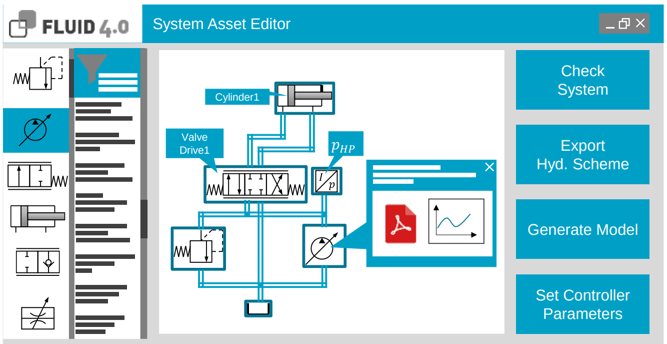

Figure 11: Mock-Up for the prototypical application, which is planned during the Fluid4.0 joint research project

solutions. At the same time, the tool will especially provide smaller and medium-sized companies with an opportunity for creating system descriptions based on the AAS, independent of license costs.

Another critical point is the standardization of the vocabularies and definitions and the definition of uniform translations. From the authors' point of view, this is currently one of the main challenges for the far-reaching implementation of I4.0.

8 Acknowledgement

This work has been funded by the European Union (EU) and supported by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) in the project “Fluid4.0 – Revolutionizing Fluid Engineering: Implementation of digitalization for fluid power 4.0 in the cross-industry and cross-manufacturer data space using asset administration shells, submodels and demonstrators” (project number 13IK039L). The views and opinions expressed are solely those of the authors and do not necessarily reflect the views of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

Funded by the European Union
NextGenerationEU

Supported by:
Federal Ministry for Economic Affairs and Climate Action
on the basis of a decision by the German Bundestag

References

[1] M. Becker, S. Heppner, R. Alt, T. Kleinert, and K. Schmitz, "Improving Fluid Power System Simulation Through an AAS-Based Simulation Framework," presented at the ASME/BATH 2023 Symposium on Fluid Power and Motion Control, American Society of Mechanical Engineers Digital Collection, Nov. 2023. doi: 10.1115/FPMC2023-109807.

[2] S. Heppner, T. Miny, T. Kleinert, M. Becker, K. Schmitz, and R. Alt, "Asset Administration Shells as Data Layer for Enabling Automated Simulation-based Engineering," in *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2023, pp. 1–7. doi: 10.1109/ETFA54631.2023.10275474.

[3] R. Alt *et al.*, "Seamless integration of device and field data into the system simulation of a hydraulic servo-press using aas," in *14th International Fluid Power Conference*, Dresden, Germany: River Publishers, 2024, pp. 548–565. doi: 10.13052/rp-9788770042222C43.

[4] R. M. Alt, "Service-oriented architecture for automated commissioning of fluid power systems," Shaker Verlag, 2023. Accessed: Feb. 23, 2025. [Online]. Available: https://publications.rwth-aachen.de/record/958758

[5] "Industrie 4.0-Verwaltungsschale für Pumpen." Accessed: Mar. 01, 2025. [Online]. Available: https://www.th-koeln.de/anlagen-energie-und-maschinensysteme/verwaltungsschale-pumpen-2_115652.php

[6] "Companion Specification for Pumps." Accessed: Mar. 01, 2025. [Online]. Available: https://www.th-koeln.de/anlagen-energie-und-maschinensysteme/companion-specification-for-pumps_115547.php

[7] OPC 40223, *OPC UA for Pumps and Vacuum Pumps*, Apr. 19, 2021. Accessed: Feb. 06, 2025. [Online]. Available: https://opcfoundation.org/developer-tools/documents/view/244

[8] "Companion Specification for CAS." Accessed: Mar. 01, 2025. [Online]. Available: https://www.th-koeln.de/anlagen-energie-und-maschinensysteme/companion-specification-for-cas_115546.php

[9] OPC 40250-1, *Compressed Air Systems - Main Control System*, Jul. 03, 2021. Accessed: Mar. 01, 2025. [Online]. Available: https://opcfoundation.org/developer-tools/documents/view/245

[10] M. Both, J. Müller, and C. Diedrich, "Automatisierte Abbildung semantisch heterogener I4.0-Verwaltungsschalen durch Methoden des Natural Language Processing," - *Autom.*, vol. 69, no. 11, pp. 940–951, Nov. 2021, doi: 10.1515/auto-2021-0050.

[11] S. Heppner, M. A. Both, T. Miny, J. Müller, C. Diedrich, and T. Kleinert, "Semantic Matching Interface für semantisch heterogene Industrie 4.0 Verwaltungsschalen," - *Autom.*, vol. 72, no. 10, pp. 919–927, Oct. 2024, doi: 10.1515/auto-2023-0225.

[12] A. Cartus, M. Both, N. Maisch, J. Müller, and C. Diedrich, "Interoperability of semantically heterogeneous digital twins through Natural Language Processing methods," in *CLIMA 2022 conference*, May 2022. doi: 10.34641/clima.2022.143.

[13] J. Beermann, M. Both, C. Block, J. Nagel, and J. Müller, "Erstellung eines semantischen Suchalgorithmus zur Abbildung proprietärer Merkmale auf den ECLASS-Standard," Jul. 2024.

[14] "ZVEI-Show-Case PCF@ControlCabinet," ZVEI. Accessed: Apr. 27, 2025. [Online]. Available: https://www.zvei.org/en/subjects/zvei-show-case-pcfcontrolcabinet

[15] N. Klein, J. Gerner, A. Wang, M. Herm, F. Küßler, and J. Fleischer, "Implementation of an Information Model for Electric Motor Production Using the Asset Administration Shell," in *2024 14th International Electric Drives Production Conference (EDPC)*, Nov. 2024, pp. 1–7. doi: 10.1109/EDPC63771.2024.10932846.

[16] S. Cavalieri and S. Gambadoro, "Digital Twin of a Water Supply System Using the Asset Administration Shell," *Sensors*, vol. 24, no. 5, Art. no. 5, Jan. 2024, doi: 10.3390/s24051360.

[17] IDTA 02006-3-0, *Digital Nameplate for Industrial Equipment*, Frankfurt Am Main, Germany., Nov. 2024. [Online]. Available: https://industrialdigitaltwin.org/en/wp-content/uploads/sites/2/2024/11/IDTA-02006-3-0_Submodel_Digital-Nameplate.pdf

[18] IDTA 02005-1-0, *Provision of Simulation Model*, Frankfurt Am Main, Germany., Dec. 2022. [Online]. Available: https://industrialdigitaltwin.org/en/wp-content/uploads/sites/2/2023/01/IDTA-02005-1-0_Submodel_ProvisionOfSimulationModels.pdf

[19] IDTA 02011-1-1, *Hierarchical Structures enabling Bills of Material*, 06/24. Accessed: Feb. 23, 2025. [Online]. Available: https://industrialdigitaltwin.org/wp-content/uploads/2024/06/IDTA-02011-1-1_Submodel_HierarchicalStructuresEnablingBoM.pdf

[20] IDTA 01001-3-0-1, *Specification of the Asset Administration Shell Part 1: Metamodel*, IDTA 01001-3-0-1, Frankfurt Am Main, Germany., Jun. 01, 2024. [Online]. Available: https://industrialdigitaltwin.org/wp-content/uploads/2024/06/IDTA-01001-3-0-1_SpecificationAssetAdministrationShell_Part1_Metamodel.pdf

[21] DIN SPEC 91345, *Reference Architecture Model Industrie 4.0 (RAMI4.0) — English translation of DIN SPEC 91345:2016-04*, Apr. 2016.

[22] Industrial Digital Twin Association, "General :: Asset Administration Shell Documentation." Accessed: Feb. 28, 2025. [Online]. Available: https://admin-shell-io.github.io/aas-specs-antora/IDTA-01001/v3.1/annex/general.html#types-and-instances

[23] E. Tantik and R. Anderl, “Potentials of the Asset Administration Shell of Industrie 4.0 for Service-Oriented Business Models,” *Procedia CIRP*, vol. 64, pp. 363–368, Jan. 2017, doi: 10.1016/j.procir.2017.03.009.

[24] R. H. L. Chiang, T. M. Barron, and V. C. Storey, “Data, Knowledge & Information in Database and Knowledge-Based Systems,” *J. Database Manag. JDM*, vol. 3, no. 3, pp. 12–20, Jul. 1992, doi: 10.4018/jdm.1992070102.

[25] K. North, *Wissensorientierte Unternehmensführung: Wissensmanagement im digitalen Wandel*. Wiesbaden: Springer Fachmedien, 2021. doi: 10.1007/978-3-658-32771-2.

[26] F. Bodendorf, *Daten- und Wissensmanagement*, 2., Aktualisierte u. erw. Auf. Berlin: Springer Berlin, 2005.

[27] ISO 1219-1, *Fluid power systems and components — Graphical symbols and circuit diagrams — Part 1: Graphical symbols for conventional use and data-processing applications*, Jun. 2012.

[28] DIN ISO 1219-1, *Fluidtechnik – Graphische Symbole und Schaltpläne Teil 1: Graphische Symbole für konventionelle und datentechnische Anwendungen*, Jan. 2019.

[29] ISO 1219-2, *Fluid power systems and components — Graphical symbols and circuit diagrams — Part 2: Circuit diagrams*, Sep. 2012.

[30] ISO 1219-3, *Fluid power systems and components — Graphical symbols and circuit diagrams — Part 3: Symbol modules and connected symbols in circuit diagrams*, Sep. 2016.

[31] Bosch Rexroth AG, *LUDV control block in mono block/sandwich plate design M7-20*. 2013.

[32] W. Wang, A. Tolk, and W. Wang, “The Levels of Conceptual Interoperability Model: Applying Systems Engineering Principles to M&S,” presented at the Spring Simulation Multiconference (SpringSim’09), San Diego, USA, 2009, p. 9. doi: 10.1145/1639809.1655398.

[33] Parker Hannifin Corporation, *P2/P3 Series Piston Pumps Variable Displacement*. Marysville, Ohio USA, 2009.

[34] Bosch Rexroth AG, *Axial piston variable pump A10VG series 60*. Elchingen, Germany, 2024.

[35] ECLASS, “ECLASS Datenstandard für Waren & Dienstleistungen,” ECLASS. Accessed: Feb. 04, 2025. [Online]. Available: https://eclass.eu/

[36] “IEC - Common Data Dictionary (CDD).” Accessed: Feb. 04, 2025. [Online]. Available: https://cdd.iec.ch/cdd/iec61360/iec61360.nsf/TreeFrameset?OpenFrameSet

[37] ECLASS, “ECLASS e.V. members,” ECLASS. Accessed: Feb. 10, 2025. [Online]. Available: https://eclass.eu/en/about-us/members/eclass-ev-members

[38] IDTA 01001-3-a-0-2, *Specification of the Asset Administration Shell Part 3a: Data Specification – IEC 61360*, Frankfurt Am Main, Germany., Jul. 01, 2024. [Online]. Available: https://industrialdigitaltwin.org/en/wp-content/uploads/sites/2/2024/07/IDTA-01003-a-3-0-2_SpecificationAssetAdministrationShell_Part3a_DataSpecification_IEC613601.pdf

[39] A. L. Opdahl, “A Platform for Interoperable Domain-Specific Enterprise Modelling Based on ISO 15926,” in *2010 14th IEEE International Enterprise Distributed Object Computing Conference Workshops*, Oct. 2010, pp. 301–310. doi: 10.1109/EDOCW.2010.29.

[40] ISO 15926-2, *Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities — Part 2: Data model*, Dec. 15, 2003.

[41] B. Kim *et al.*, “Use of Asset Administration Shell Coupled with ISO 15926 to Facilitate the Exchange of Equipment Condition and Health Status Data of a Process Plant,” *Processes*, vol. 10, p. 2155, Oct. 2022, doi: 10.3390/pr10102155.

[42] M. T. Delgado, “Eclipse Semantic Modeling Framework (ESMF) | projects.eclipse.org.” Accessed: Feb. 05, 2025. [Online]. Available: https://projects.eclipse.org/projects/dt.esmf

[43] M. Milinkovich, “About the Eclipse Foundation | The Eclipse Foundation.” Accessed: Feb. 10, 2025. [Online]. Available: https://www.eclipse.org/org/

[44] “Semantic Aspect Meta Model (SAMM) : Bosch Semantic Stack Documentation.” Accessed: Feb. 09, 2025. [Online]. Available: https://docs.bosch-semantic-stack.com/oss/samm-specification.html

[45] “schema.org.” Accessed: Feb. 05, 2025. [Online]. Available: https://schema.org/

[46] P. A. J. Achten, Z. Fu, and G. E. M. Vael, “Transforming future hydraulics : a new design of a hydraulic transformer,” *Fifth Scand. Int. Conf. Fluid Power SICFP 97 Swed. Linköping*, p. 24, 1997.

[47] S. Bader *et al.*, “What is the AAS.” Accessed: Feb. 09, 2025. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/2021_What-is-the-AAS.pdf?__blob=publicationFile&v=1

[48] C. Schmidt and J. Müller, “GitHub - FraunhoferIOSB/EDC-Extension-for-AAS: EDC Extension supporting usage of Asset Administration Shells.” Accessed: Feb. 09, 2025. [Online]. Available: https://github.com/FraunhoferIOSB/EDC-Extension-for-AAS?tab=readme-ov-file

[49] “History,” OPC Foundation. Accessed: Feb. 06, 2025. [Online]. Available: https://opcfoundation.org/about/opc-foundation/history/

[50] OPC 10000-8, *OPC Unified Architecture - Part 8: Data Access*, Oct. 05, 2024. Accessed: Feb. 06, 2025. [Online]. Available: https://opcfoundation.org/developer-tools/documents/view/165

[51] OPC 10000-100, *OPC Unified Architecture - Part 100: Devices*. Accessed: Feb. 06, 2025. [Online]. Available: https://opcfoundation.org/developer-tools/documents/view/197

[52] OPC 40400-1, *OPC UA for the Powertrain - Part 1: Asset ManagementPower*, Nov. 01, 2024. Accessed: Feb. 06, 2025. [Online]. Available: https://opcfoundation.org/developer-tools/documents/view/240

[53] S. Köhler and J. Weber, "Integration of fault detection and diagnosis methods into OPC UA for mobile machinery," 2022, pp. 457–472.

[54] B. Beck *et al.*, "Bauen 4.0 - Effizientere und produktivere Bauprozesse durch Vernetzung und Kommunikation mobiler Maschinen : Abschlussbericht : Bauen 4.0 Bauprozesse - Vernetzung - Maschinen : Laufzeit des Vorhabens: 01.07.2019 bis 31.12.2022," Dresden, 2023. doi: 10.2314/KXP:1881651258.

[55] S. Köhler *et al.*, "Digital assisted collision avoidance for mobile machinery," in *14th International Fluid Power Conference*, Dresden, Germany: River Publishers, 2024, pp. 923–943. doi: 10.13052/rp-9788770042222C73.

[56] "FAQ." Accessed: Feb. 10, 2025. [Online]. Available: https://mqtt.org/faq/

[57] OASIS Open, "MQTT Version 5.0." Mar. 07, 2019. Accessed: Feb. 10, 2025. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf

[58] "Functional Mock-up Interface." Accessed: Apr. 27, 2025. [Online]. Available: https://fmi-standard.org/

[59] "Tools - Functional Mock-up Interface." Accessed: Apr. 27, 2025. [Online]. Available: https://fmi-standard.org/tools/

[60] "About - Functional Mock-up Interface." Accessed: Apr. 27, 2025. [Online]. Available: https://fmi-standard.org/about/

[61] "Modelica." Accessed: Feb. 07, 2025. [Online]. Available: https://modelica.org/association/#organizational-members-of-the-modelica-association-listed-alphabetically

[62] D. Winkler, P. Harman, M. Tiller, K. Davies, and M. Sjölund, "GitHub - modelica-3rdparty/OpenHydraulics: A free Modelica package providing components describing 1-dimensional fluid flow in hydraulic circuits." Accessed: Feb. 07, 2025. [Online]. Available: https://github.com/modelica-3rdparty/OpenHydraulics

[63] M. Eigner and R. Stelzer, *Product Lifecycle Management – Ein Leitfaden für Product Development und Life Cycle Management*. 2008.

[64] ISO/IEC/IEEE 24748-2, *Systems and software engineering — Life cycle management — Part 2: Guidelines for the application of ISO/IEC/IEEE 15288 (system life cycle processes)*, Mar. 2024.

[65] ISO 15226, *Technical product documentation — Life cycle model and allocation of documents*, Apr. 01, 1999.

[66] R. Heidel, M. Hoffmeister, M. Hankel, and U. Döbrich, *Industrie 4.0 – Basiswissen RAMI4.0*. 2017.

[67] IDTA 02026-1-0, *Provision of 3D Models*, Jun. 2024. [Online]. Available: https://industrialdigitaltwin.org/wp-content/uploads/2024/06/IDTA_02026-1-0_Submodel_ProvisionOf3DModels.pdf

[68] EPLAN GmbH & Co. KG, "EPLAN – efficient engineering." Accessed: Feb. 12, 2025. [Online]. Available: https://www.eplan.de/