
Digital fluid technology in the data space: From semantic component search to the development of mobile hydraulic systems using AAS

Paul Lukas Remde¹, Benjamin Beck^{1,2}, Denis Ritz², Jürgen Weber^{1,2},
Tom Gneuß³, Nico Braunisch³, Martin Wollschlaeger³

¹ Construction Future Lab gGmbH, benjamin.beck@cflab.de, paul.remde@cflab.de

² TU Dresden, Chair of Fluid-Mechatronic Systems, denis.ritz@tu-dresden.de,
fluidtronik@mailbox.tu-dresden.de

³ TU Dresden, Chair of Industrial Communications, tom.gneuss@tu-dresden.de,
nico.braunisch@tu-dresden.de, martin.wollschlaeger@tu-dresden.de

Abstract.

The development of hydraulic systems is becoming increasingly complex, especially in mobile machines with a wide range of fluid technology components. The joint research project Fluid 4.0, funded as part of kopa35c and embedded in the objectives of Manufacturing-X, is developing an end-to-end digital approach for the collaborative engineering of fluid power systems. The focus is on the Asset Administration Shell (AAS) as an interoperable digital twin. The article presents solutions for industry-relevant User Stories like “Machine AAS with linkage and physical composition of component AAS”, “Semantic Search within the dataspace to find suitable products” and “Automatic control software parametrization”. Therefore, occurring pain points inside the industry will be highlighted and helpful technologies based on the AAS introduced. This paper provides input to the application of existing and new approaches of the AAS to reduce manual effort and enhance collaborative engineering mainly inside the design process of fluid-mechatronic systems.

Keywords. Fluid 4.0, Digitalisation, Asset Administration Shell, Collaborative Engineering, Dataspace, Submodels, industry-relevant Use Cases

1. INTRODUCTION

The requirements for the development of stationary and mobile machines with hydraulic or pneumatic drive technology are increasing due to targets set by customers and legislation as well as technological developments and product diversity on the market. Particularly in the context of collaborative engineering (across departments and companies), standardised data helps to make efficient development decisions. The Asset Administration Shell (AAS) as a data container with a machine-readable interface and industry-standardised semantics

represents a technological option for implementing this data-based decision-making basis. The publication of [1] provides interesting insights about the characteristics and technologies about the AAS with a high number of references to various tools and information about the usage inside the industry and applied technologies. The fluid power industry can already supply many components with digital representation in the form of an AAS. However, the added value only arises when the data is utilised in the development of the machine.

The Fluid 4.0 joint research project is developing solutions for the digitalisation of the following four key areas at the interface of fluid technology components and their use in stationary and mobile machines:

1. Control development of machines
2. Energy monitoring
3. Cross-industry CO₂-balancing
4. Circular economy

The content of this publication will take part on the first key area “Control development of machines”, which also includes system layout design. At first in chapter 2, the state of the art and technology will be presented by describing the development of mechatronic systems and interoperable data models. The next chapter follows with the introduction of the main User Stories to support engineering processes based on the AAS. The relevant technologies, that have been used and developed to achieve the desired improvements inside those User Stories will be presented in chapter 4. Chapter 5 shows the usage of those technologies by presenting client applications, applied for the User Stories respectively. Therefore, a simplified example system will be used in addition to a demonstrator, resembling a mobile machine, for further validation. The conference paper will be finished with a conclusion and an outlook in chapter 6.

2. STATE OF THE ART AND TECHNOLOGY

2.1. Development of Mechatronic Systems

As described in VDI Guideline 2206, the development of modern mechatronic systems is subject to a staggered process. Basically, this can be described as a V-model. On the left branch, a system design with rough boundary conditions is repeatedly refined with increasingly fine-grained components until the lowest point of the “V” – the point of greatest detail – is reached. In the development phase – the right branch of the “V” – each hierarchical level is then checked from the lowest level upwards to ensure it fulfils the required function. This involves first testing individual components, subsystems, systems, and finally the entire machine/plant [2]. This so-called macrocycle of the V-Modell is shown as the inner cycle in Figure 2.1.

What also can be seen in the figure is that the inner cycle is layered with more outer cycles, which can be interpreted as different stages of a product, from a (digital) functional mock-up to a (digital) demonstrator, a prototype, a pilot series, and the “final” series product itself. There could be more outer cycles, like for each step of a “model facelift” or development of a new product version release. During the development process, multiple different domains must work together to create a suitable product. Typically, each domain works with its own tools; see [3], which is a major flaw during the design process, because of the limited

possibilities for the exchange of data between the different domains. This problem is often more complex when you look at the whole chain from the different companies that are involved for one product, starting at a “low” level from a TIER-X supplier, which may offer “simple” components like screws or cable connectors, along the path to the TIER II and TIER I suppliers, which offer complex elements and subsystems that the OEM has to integrate into their product.

One common way for exchanging data is PDF files, which often include unstandardized content and are not machine-readable by default. One point at which this circumstance becomes apparent in the development process is the parameterization of a lookup table to describe the often non-linear behaviour of a component during the application software development of a machine. Product data sheets often contain two-dimensional curves describing the control-relevant behaviour of components such as valves and pumps, which provide a good basis for parameterization. However, these curves are in a lot of cases only available as graphical representations of the data. The values for such a lookup table must be recreated manually. This is highly inaccurate and prone to errors. Another problem system developers face is the connection flanges of hydraulic components, which are also represented as a PDF graphic. Those flanges are essential when creating manifolds for complex systems. A remarkable fact is that Automation Studio has a function that has a large database for creating manifolds automatically [4]. Such a high integration of connection interfaces is lacking in many other tools.

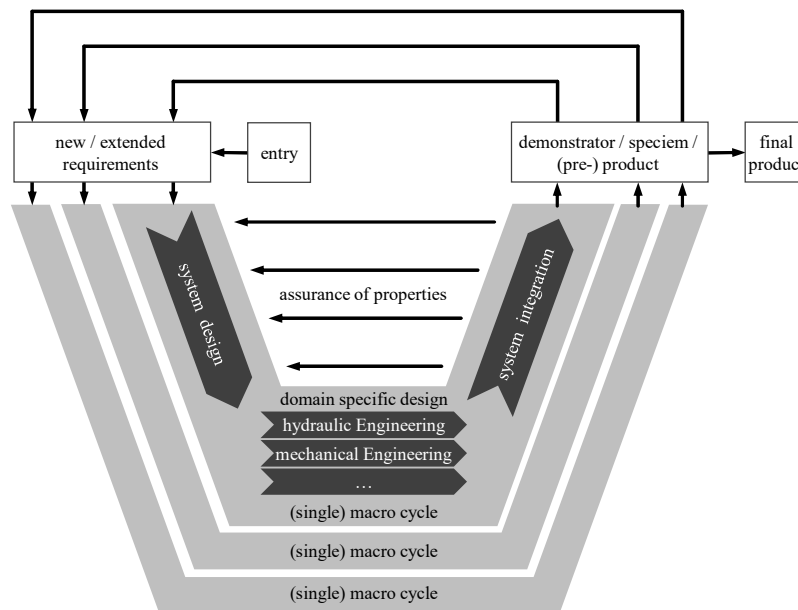


Figure 2.1. Multiple macro-cycles in the V-Model for the product development of a complex mechatronic product according to [2]

When looking at the V-Model again, it is hard to find a program architecture for an application, which is suitable to cover the whole development process and fits all requirements of each system level and domain. To describe a machine with its drive system

and the process, a system simulation tool with lumped parameters can be a good option. The product itself is often represented in a 3D CAD program, which is a geometrical description. Both describe the same product in different ways. But when changing major parameters of the product based on one of those systems, there is just manual feedback and no automatic functions that trigger a redesign. As an example, the systems engineer simulated the lumped model and identified that the system's dynamics are too slow; he decided to use a larger hydraulic cylinder, which offers higher forces. The larger force of the actuator is thus a changed requirement of the frame, which holds the actuator. This may lead to several changes in the mechanical / geometrical design to support the force into the frame of the machine.

It is therefore clear that the development of complex systems such as the hydraulic drive system of machines requires a large amount of data and information that must be exchanged across companies and departments. How can this be done efficiently while avoiding errors?

2.2. *Interoperable Data Models*

Interoperability can be defined as the ability of two or more systems to exchange and use information seamlessly. [5] In the context of engineering, product and machine data, i.e. layout, interfaces, and connectors, represent a crucial form of information that must be interpreted and applied by a wide range of stakeholders. These include designers and producers, who generate and maintain product data, as well as customers and OEMs, who need to consume the same data in their respective tool environments.

To support this exchange, several standardized formats for engineering data have been developed. Common examples include STEP, AutomationML (AML), and OPC UA, which are reviewed in detail by Ritz [3]. Each of these formats is formally specified and standardized, yet their adoption in practice remains heterogeneous. Not all manufacturers rely on the same data format, and the distribution of data is likewise inconsistent: some producers send engineering files by email after purchase, others make them available for download on websites, and in many cases, product specifications are still provided only as PDFs, which has been discussed above. Furthermore, different formats tend to dominate in different engineering domains, limiting seamless cross-domain data usage [6]. Technologies such as RDF offer higher semantic expressiveness and support reasoning mechanisms, but their adoption in industrial practice has been limited so far [7]. The resulting multitude of formats hinders comparability across components and complicates the composition of systems, a task that OEMs in particular must regularly perform. In contrast, interoperable and vendor-independent data formats have the potential to streamline engineering workflows, enabling consistent data exchange and integration across tools and domains.

Against this background, the AAS has emerged as a particularly promising approach. Developed in context of Plattform Industrie 4.0 and the Industrial Digital Twin Association (IDTA), the AAS provides a standardized digital representation of assets that is semantically enriched and interoperable across domains [8]. Its standardized metamodel, defined in IEC 63278, ensures syntactic interoperability. The AAS organizes asset-specific information into Submodels (SMs), which can be further standardized as Submodel Templates (SMTs) to guarantee the well-defined location of particular data elements. Finally, a standardized application programming interface (API) provides a uniform access

mechanism for reading and utilizing this information across various tools and platforms. Refer to chapter 4.1 for further details on the technical realization of the AAS.

3. USER STORIES

As described in chapter 2, the development process of mobile machines is highly collaborative and data intensive which leads to communication effort and is error prone. During an intensive user story development phase within the Fluid 4.0 project under the support of associated partners the most relevant processes with high potential for data usage and automation were identified. Thereby, three main problems were identified:

1. data on components and the machine are distributed
2. technical properties are rarely semantically standardised and usually not available in machine-readable form
3. there is a diverse tool landscape

Based on this, five user stories have been developed to accommodate this.

User Story I: Machine AAS

Based on the AAS of hydraulic components, the whole drive system of a machine should be described within a machine AAS as well. This would provide a single source of truth for machine-readable information about the system for all developers. The machine AAS should represent the physical and data linkage between the used components and their digital representation. Designers of hydraulic systems should be able to create a layout as the basis for further development steps.

User Story II: Semantic Search

Based on the layout and calculation of the size of the components, the engineers must search for suited products, which is a very manual and time-consuming task today. Hydraulic engineers shall be able to search for fitting products within the dataspace based on standardized technical properties of public information, such as data sheets. This approach has been applied in [9] for the parameterization of simulation models regarding Model Based System Engineering.

User Story III: Tool interfaces and automation

The automated information transfer between different tools shall be possible. Various applications in the engineering process can benefit from standardized data provision and take advantage of automating workflows, which required manual input before. By using freeware and other work-specific software tools, the applied usage of the AAS will be demonstrated and will contribute as an example and motivation for future AAS application integrations.

User Story IV: Simulation

Even though simulation models can be unique and complex, data are required to define the values of the components. Likewise, data from the AAS can be inserted automatically to save time and effort. On top of that, since simulation models often resemble the shape of a hydraulic schematic, a first approach to generate the first instance of a simulation model

automatically can be carried out, which will be presented in additional papers. The applicability of the AAS for the Simulation has been shown in [9], [10].

User Story V: Control development

As basis for virtual commissioning, control relevant parameters of the used components shall be imported automatically into the IDE (integrated development environment). Software developers benefit from fast and safe parametrization of the control units' I/O's.

Figure 3.1. shows the implementation of these User Stories through five client applications and their interaction via a central AAS – the machine AAS as the single source of truth during the product development. In the focus of this contribution, the clients I, II III and V will be described in detail.

To implement these User Stories with the corresponding Clients a few building blocks and technologies are needed. In [3] a mock-up for a single application was shown in the outlook. This publication focuses on the identified clients, which are described in the following chapter. Basically, there are always three main building blocks:

1. Data modelling
2. Infrastructure and Interfaces
3. Clients and Tools

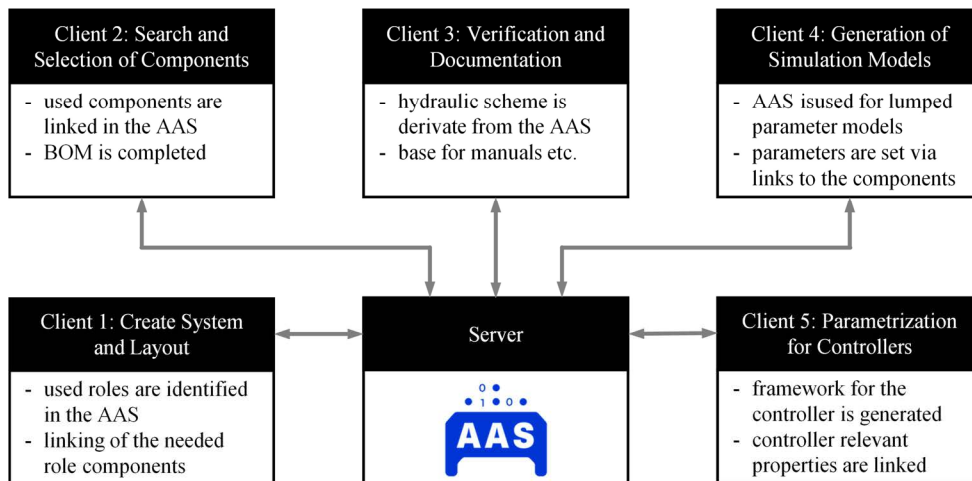


Figure 3.1. Client Applications for the implementation of the defined User Stories

4. INDUSTRY 4.0 TECHNOLOGY

4.1. AAS Infrastructure

The construction of an AAS follows the rules defined in its standardized metamodel [11]. An AAS is composed of SM, each of which contains hierarchically structured SubmodelElements (SMEs). These SMEs can take on various types, such as Properties, Ranges, References, or Collections. The structure and nesting depth of SMEs is highly variable and depends on the specific information that needs to be represented. Different SMEs can be grouped in a SubmodelElementCollection (SMC), which also can include

other SMCs. To provide consistency across use cases, SMTs define a pre-specified structure for SMs and their elements. Such templates are tailored to specific domains or application scenarios, for example the *Digital Nameplate for Industrial Equipment* or the *Generic Frame for Technical Data for Industrial Equipment in Manufacturing* [12]. Refer to chapter 4.3.3 for an in-depth elaboration on the information that these provide. Templates are not developed in isolation but are discussed and refined in dedicated working groups, ensuring cross-industry applicability and acceptance (standardization process).

Two types of AAS infrastructure are distinguished. A Type-1-AAS is exchanged in the form of a serialized file, typically encoded as XML, JSON, or RDF, and often wrapped in an AASX package for convenient bundling [13]. In contrast, a Type-2-AAS is hosted on a server, which provides an application programming interface (API) via HTTP for reading and modifying AAS contents [14]. This includes operations such as changing the value of SMEs, adding or rearranging SMEs, or adding and removing complete AASs and SMs. Beyond basic access, a Type-2-AAS also supports a query language that allows clients to request AASs or SMs from a server that match certain criteria [14]. There exists a vast landscape of open-source AAS applications that implement the AAS metamodel and API specification to varying degrees. Among them, some have just recently included support for the AAS query language.

From a security perspective, the AAS infrastructure already integrates attribute-based access control (ABAC) mechanisms [15]. These mechanisms allow fine-grained regulation of who can access or modify specific AAS contents. However, information signing for ensuring the integrity and authenticity of exchanged data is not yet part of the standard. At the time of writing, there is no publicly available off-the-shelf AAS server application that reliably and completely implements this specification.

For implementing business logic and user stories working with the AAS, the community has created several Software Development Kits (SDKs). The SDKs provide the necessary functions to insert, edit, delete, as well as serialize and deserialize the information inside an AAS. For example, SME can be added to SMs, their values edited, and the result serialized to an exchangeable format like JSON.

4.2. Dataspaces

A dataspace is a decentralized technology for sharing information. Its central focus lies in ensuring data ownership and sovereignty, meaning that data providers retain full control over the conditions under which their data is made available and used [16].

To support interaction among participants, dataspace typically include centralized services such as a catalogue, which enables the discovery of connector endpoints of other parties, and a wallet, which stores participant certificates. Each participant in a dataspace can act as a data provider, a data consumer, or both, depending on their role in each exchange.

On the provider side, internal rules are defined that govern the conditions under which a consumer can access information. Consumers can issue requests to provider endpoints but must present authentication information; the provider then decides whether the consumer is eligible to receive the requested data. These access rules can go beyond simple eligibility checks and include detailed usage policies, such as restrictions on how often the data may be accessed, how it may be processed, or whether it can be forwarded to third parties. In this

way, policy enforcement mechanisms ensure that data sovereignty is preserved even after access has been granted [17].

The technical realization of these interactions is achieved through dataspace connectors. These connectors must be configured either as providers or consumers, although some implementations support a dual configuration in which a connector can serve both roles. A provider-side connector must be connected to backend resources, such as databases or repositories, to expose the relevant information. At the architectural level, contracting takes place on the control plane, while the actual data exchange occurs on the data plane [18].

Within the Fluid 4.0 key areas and User Stories, each vendor maintains an AAS repository and a SM repository containing AAS (and their linked SMs) for all offered products. At a minimum, each AAS is required to include the SM *Generic Frame for Technical Data for Industrial Equipment in Manufacturing* [12]. One dataspace connector is deployed per vendor, linked to the AAS/SM repositories, and another connector is operated by the customer. The customer issues a query, routed through its connector, to all vendor connectors to retrieve *Technical Data* SM that meet certain criteria – for example, a specific ECLASS number identifying pumps or valves (User Story II). Based on the retrieved SMs, the customer then issues a second query to the AAS repositories (or rather, the dataspace connectors acting on their behalf) to collect all AASs that reference these SMs. The result is a consolidated list of AAS associated with the targeted product category. The whole process and involved components are visualized in Figure 4.1.

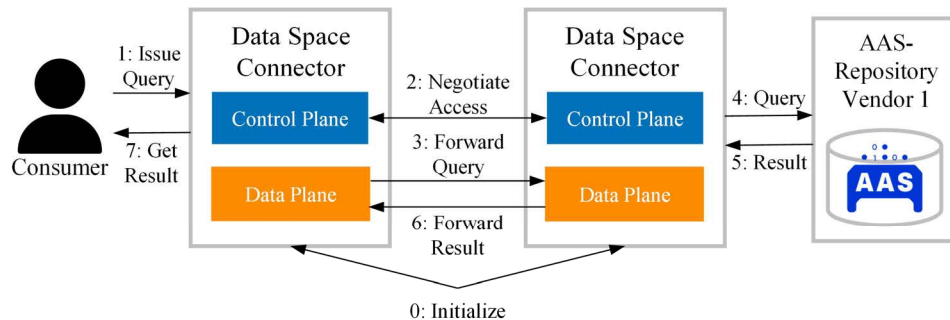


Figure 4.1. Workflow for querying through the Dataspace

4.3. Relevant Submodels for collaborative engineering

SMs are the key elements of an AAS containing domain-specific information in a structured way. Given the well-known location of data according to the SMT, the extraction of information can be carried out automatically, regardless of the manufacturer who populated the information. The diversity in the presentation of product-specific details inside the uniquely designed datasheets of the companies will be bypassed completely.

Up until now, many SMs have been published, e.g., the *Digital Nameplate for Industrial Equipment* [19] or *Handover Documentation* [20]. As the name implies, SMs are mainly designed for specific purposes, like [20], which offers information about the asset's manufacturer and its identification. Beyond that, machine manufacturers can create their own client applications to interact with the machine-readable information of the assets for their individual purposes. In the end, one of the main purposes of the SMTs lies in the

standardized provision of asset-specific information in a machine-readable format. As a result, the engineering process can be designed more efficiently through the new possibilities in automation.

In the process of applying the User Stories by developing the client applications, that has been illustrated in Figure 3.1., required data were missing that couldn't get included in the defined structures of existing SMTs. This includes the information in diagrams, which are only graphically, but not machine-readable provided inside the datasheets of the manufacturers. Those data play a major role inside the control development of User Story V, since software developers need to extract the information manually to handle the control behaviour of hydraulic components accordingly for their unique application. This contributes to the necessity of a SMT to retrieve machine readable information from a graphic in a standardized generic syntax.

Furthermore, to realize the User Stories I, III and IV, a new approach for a SMT has been created to automatically generate a hydraulic schematic. To develop a fluid technological drive system, the hydraulic schematic contains the information about which components are included and how there are connected respectively. The provision of the information in a generic machine-readable syntax of the AAS generates many opportunities for the realization of the User Stories. By applying graphic editors to draw hydraulic schematics, the AAS of the developed system, regarding to Use Case I, can be generated in combination with the graphic information stored inside a SM. This would help to create hydraulic schematics in various data formats for communication and documentation (Use Case III), as well as serving the graphic information as basis for the simulation models of fluid-technological systems (Use Case IV), while interacting with one medium as single source of truth. [9] has been mentioned, that the automatic generation of the structure and topology of a simulation model should be achieved in the future. The content of those new SM approaches will be further discussed in the following chapters 4.3.1 and 4.3.2.

To realise User Story II, existing SMTs need to be applied, like *Generic Frame for Technical Data for Industrial Equipment in Manufacturing* [Data]. This SM contains all necessary information to classify a component and verify its system compatibility by its technical properties. As a result, search algorithms for fitting components can be implemented. A more specific insight of this SMT will be presented in the chapter 4.3.3.

4.3.1. Characteristic Curves

As explained above, the characteristics of hydraulic components and systems are usually characterized by nonlinearities. For this reason, an application was submitted to the IDTA as part of the Fluid 4.0 joint research project to create a new SMT *Characteristic Curves*. The aim is to develop a uniform and universal description of characteristic curves that can be used for the technical properties of a wide variety of products and various applications.

On the one hand, (multidimensional) characteristic curves are often a widely used approach for modelling component behaviour in system simulation. On the other hand, the inverse behaviour of the components is essential for achieving good control behaviour of the respective drive axes. Since the corresponding SM is still under development, no definitive representation can be provided at this point. However, Figure 4.2. shows an example of both a characteristic curve mapping $\mathbb{R}^1 \rightarrow \mathbb{R}^1$ and a higher-dimensional

characteristic field representing a mapping of the order $\mathbb{R}^2 \rightarrow \mathbb{R}^1$. This can be extended to any number of dimensions.

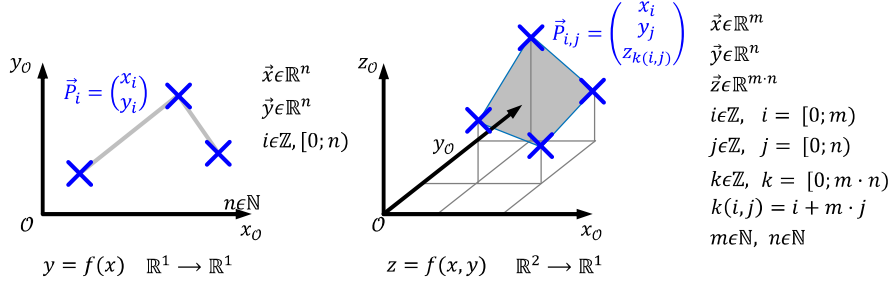


Figure 4.2. Examples for a representation of characteristic curves, according to the planned IDTA SM Characteristic Curves

Generally, it makes sense to create a link to a semantic lexicon, such as ECLASS [21], to build a unique assignment of the contained values to the corresponding variables of the respective components. At the same time, it is planned to store the boundary conditions under which the respective behaviour is mapped, such as the kinematic viscosity ν that was present during a valve measurement.

4.3.2. Model Reconstruction

In the fluid technology industry, the hydraulic schematic is a document of high significance. It contains the integrated components in a visual representation and their connections to one another. It provides the blueprint to achieve the planned movement sequences of the machine via hydraulic components. For engineering tasks, the schematic serves as a fundament for communication, planning and decision making. On top of that, they are also used for maintenance and monitoring of hydraulic systems afterward.

To improve the efficiency of the engineering process, the task occurred to develop a SM, which enables the possibility to reconstruct a hydraulic schematic generically according to User Story I. Schematics are most often available in graphic or software-specific files, especially between employees with different working backgrounds. A standardized template, which includes all necessary parameters of a hydraulic schematic, would offer the opportunity for interaction in a specific work-related environment. This will play a major role in the AAS of hydraulic systems, which will be further discussed in chapter 4.4. Since simulation models and hydraulic schematics are highly related, the information inside the SM can be used for the automated generation of simulation models as well, see User Story IV. This could also be used for the electrical engineering as well with same benefits, ideally using the same BOM as shared data for the systems engineering.

Up until now, the SM consists of two SMCs with the following names: *Model Components* and *Connections*. A *Model Component* is a graphic element inside the schematic, e.g. a pump, valve, or cylinder. It includes specific properties for a unique identification and visual placement inside the schematic. The identification includes the component name, a display designation for tags inside the model, a description for further details, and an element identification to categorize it based on ECLASS [21] standardization. Additionally, a model

component possesses attributes to describe its virtual appearance inside the model. This includes position, size, and orientation, as well as information about the unique positions and IDs of component-specific ports. This approach is visualized in Figure 4.3.

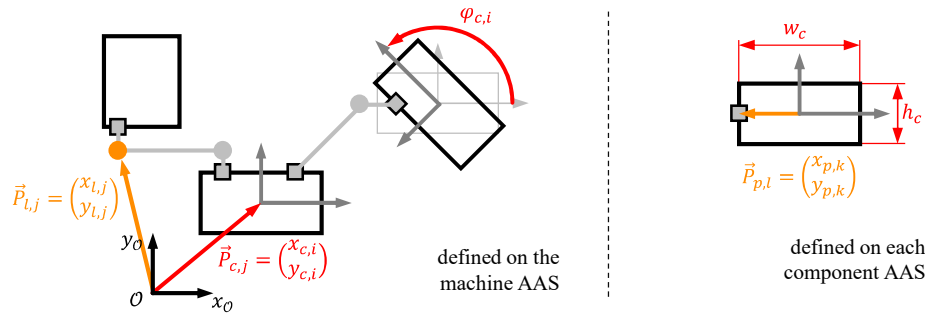


Figure 4.3. Basic approach for Semantic Position Description

Connections is the second SMC inside the *SM Model Reconstruction* and contains the information to describe the linkages between the model components respectively. Inside the SMC is a *Source* and *Destination*, which includes the name of the model component in combination with the specific ID of the port. By determining a source and destination, the direction of flow can additionally be described. To picture the hydraulic schematic more accurately, the SMC *Connections* also include *Points* and *Line*. *Points* determine the coordinates of the key points to describe the path of the line respectively. The coordinates to describe the location of those points share the same reference as the position coordinates of the model components. Additionally, *Line* includes information about the width and shape of the line in first instance.

4.3.3. Generic Frame for Technical Data for Industrial Equipment in Manufacturing

To establish User Story II the *SM Generic Frame for Technical Data for Industrial Equipment in Manufacturing* [12] is used and extended. Besides *Product Classification*, an important property to identify the specific type of hydraulic components, the SM contains most technical data, which used to be found inside the datasheets. From a different point of view, this SM is the provider of machine-readable data, ideally combined with a semantic id based on ECLASS or other dictionaries. From a general view, this SM is the source to retrieve most of the necessary information that is required for automatic product search (User Story II).

This SM includes SMC like *Product Classifications* and *Technical Properties*. *Product Classifications* associates the asset with product classes in common classifications systems like ECLASS, e.g. an axial piston pump or a hydraulic cylinder. *Technical Properties* includes technical parameters to describe the characteristics of the components respectively. These parameters relate to International Registration Data Identifiers (IRDI) [22], to enable a unique identification of the property. This SM is the basis to implement Use Case II, since the conditions are met to automate the search process for required components, because a model components suitability to be integrated inside the machine can be verified by its product group and technical parameters.

4.4. *The AAS for hydraulically-driven machines*

Machines with a hydraulic drive system are a composition of hydraulic components according to a defined layout. Even though you could treat the components like pumps or valve as systems themselves, in context of Fluid 4.0, the focus lies on stationary machines, like a hydraulic press, and mobile machines such as wheel loaders or excavators.

According to User Story I, the digital representation of the machine drive system is one fundamental basis for collaborative engineering. From a machine manufacturers perspective data consistency is required from the machine AAS to the component AAS without duplicating data. The physical and logical linkage of component AAS within the machine layout is already part of the SM *Model Reconstruction* in chapter 4.3.2. Besides the idea of a structural reconstruction of hydraulic plans, the SM offers the opportunity for an interaction with the selected system components and their unique AAS by using API. The linkage between specific component AAS and their model-based representation inside the *Model Reconstruction* has yet to be finalized. First approaches are the synergetic application of the SM *Hierarchical Structures enabling Bills of Material* [23], which contains the used components inside the systems and, if available, the reference to their specific AAS.

The combination of both SMs *BOM* and *Model Reconstruction* would enable the possibility to simultaneously provide all required data interactively, since the information have been linked through the *BOM* and an interface for interaction has been provided through the *Model Reconstruction*. A machine AAS, consisting of both SMs, would tremendously support the provision of information, to enable an automated data exchange, the creation and parametrization of simulation models, the efficient exchange of components, and to retrieve component-specific information like the characteristic curves.

5. CLIENTS AND VALIDATION

The User Stories described in chapter 3 will be addressed by the client applications shown in Figure 3.1. This chapter presents an overview of the applications in combination with an assessment of how precisely the requirements have been fulfilled. All client / user applications use the *aas-core-works* SDKs as basic communication stack with the AAS [24].

5.1. *Systems Engineering and Model Reconstruction*

The first User Story aims for a digital twin of a hydraulic system that can later be shared and detailed further. Typically, after the objective of the system has been defined and all necessary translational and rotational movement patterns have been detected, the design of a hydraulic system begins with the hydraulic schematic. It includes all necessary actuators for power transmission, e.g., cylinders or hydraulic motors, as well as the required valves and hydraulic lines to control the flow of power, and finally, the providers for hydraulic power, like pumps.

There are various software tools to generate a hydraulic schematic by dragging and dropping the hydraulic components. Usually, there are libraries of commonly used components that can easily be embedded into the own design. Connections between the components via their unique ports are established in the same graphical manner. From that schematic, the initial version of an AAS for the hydraulic system is created. To achieve it, a graph editor program is required to create the library, which contains all the necessary components to draw

hydraulic schematics. In this project, *yEd* [25] has been used, since it is a free tool that can generate various diagrams. Besides the visualization aspect, it is important to embed component parameters inside the library, which can later be retrieved through an XML or GraphML-file. To create the AAS out of the file, a Python program has been developed, which executes the deserialization process by using the *aas-core-works* SDK [24]. First, the *SM Model Reconstruction* from chapter 4.3.2 will be applied. It contains all the information to recreate the hydraulic schematic as detailed as it was created in *yEd* before. This provides the opportunity for later interactions, like the establishment of references to individual component AAS. In Figure 5.1., a cylinder drive system has been designed inside *yEd* by using the prepared library of elements and later transferred into an AAS by using the implemented Python application.

Figure 5.2. shows an example application of the *SM Model Reconstruction* by recreating the hydraulic schematic inside a vector graphic. To simulate the generation of a hydraulic system, a second Python application has been programmed to identify the parameters inside the *SM Model Reconstruction* and generate a vector graphic (*.svg) out of it. The result is shown on the right side of Figure 5.2. The hydraulic schematic has been created by retrieving information from the AAS. This shows the advantage of an AAS as single source of truth which can be used as harmonization inside multiple tools using different file formats.

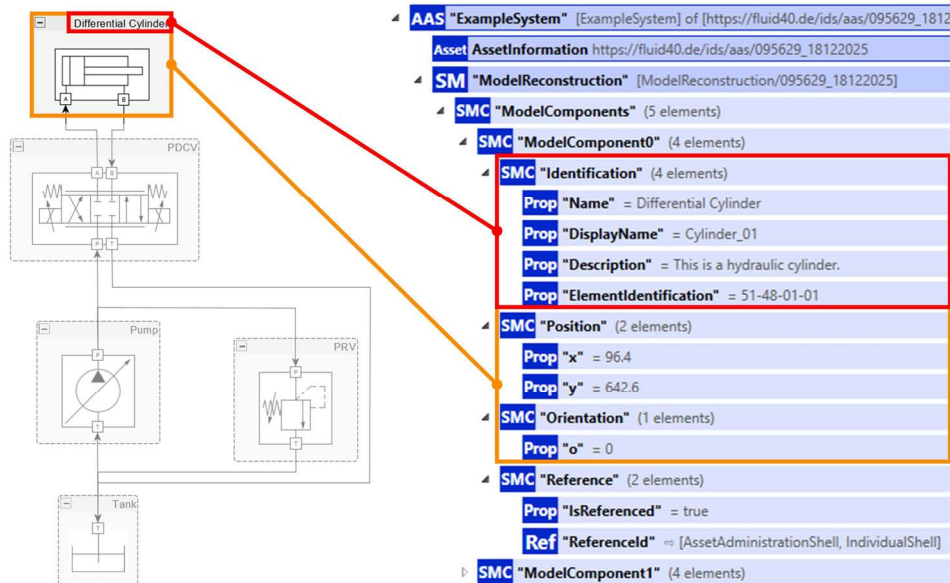


Figure 5.1. Example System of a hydraulic cylinder drive by using *yEd* (left) and its representation as AAS (right), which was generated automatically by the python script

In chapter 2, the ineffectiveness of the process in finding fitting components for a hydraulic system by manually analysing various datasheets has been highlighted. It contributed to User Story II for a standardized data format, which can be applied for an automatized search process. By defining the relevant parameters, the values of various AAS can be analysed and automatically identified as a fitting component. The basic requirement includes the

knowledge of the exact location, where the information will be stored, a fitting identifier, and compliance with the syntactic representation.

To achieve it, the *SM Generic Frame for Technical Data for Industrial Equipment in Manufacturing* will be used to store the individual parameters of the components. By using the IRDI, the technical properties will receive a unique reference for a specific assignment. On one hand, the SM stores the information about the product class of the component by using ECLASS identification. On the other hand, the individual technical parameters for the specific product group will be listed. By using AAS API calls to retrieve the information, the suitability of the component can be assessed through the comparison of the available and required parameters. This process enables automatic search by scanning accessible servers of the component vendors, including AAS of the traded components. Even though the process works as anticipated, difficulties occurred by scanning the server for suitable components. By repetitively requesting the AAS (including *Technical Data*) from the server and checking whether the component matches the desired parameters, the time for computing increased noticeably with a higher number of components. To reduce the client-side computing effort and increase the efficiency of this process, the AAS Query Language [14] has been applied. It assists the search process tremendously, since the server pre-filters the results instead of the client application.

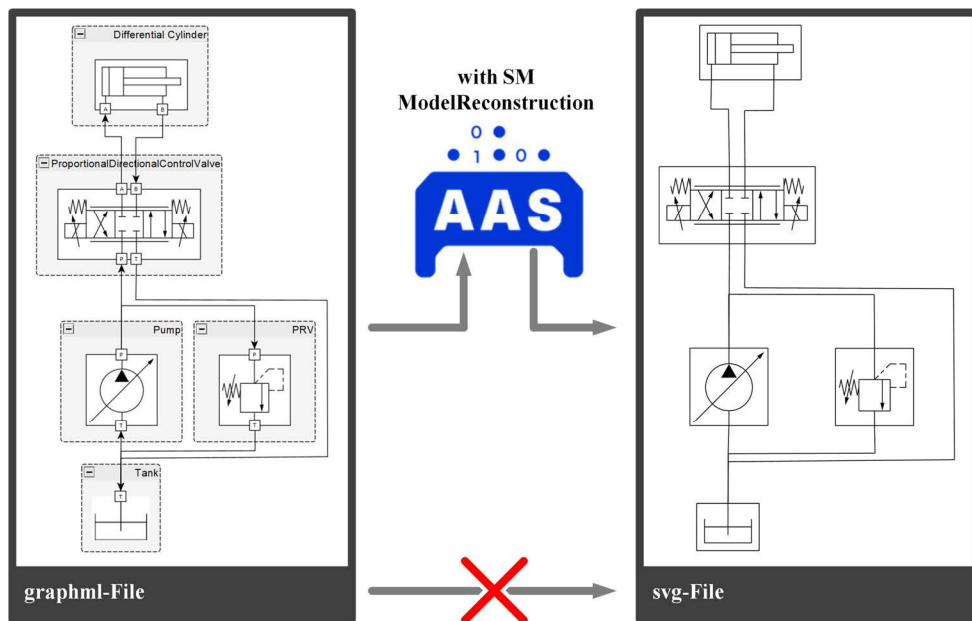


Figure 5.2. Vector graphic of a hydraulic system (right) created by the data of the Submodel Model Reconstruction inside the AAS of the system

A query request requires a payload specifying the constraints of the query. Various definitions can be integrated to specify the necessary properties that the AAS of the components need to provide in combination with the sought values. For example: “Filter for all component AAS which represents a hydraulic pump with a displacement of 100 cm³ or higher on this server.” By intuitively using logical and comparison operators in the syntax

of the Query Language, this technology can be used to extract the relevant AAS from a large pool of assets. After the relevant components have been identified, more information can be retrieved for downstream purposes. [9] has been also using the query language to filter the available assets to select the fitting components, including their parameters, for the simulation model.

User stories III, IV and V have some strong similarities. In all of them, the machine AAS is used to collect the demand data for specific customers and the connections between all components. A specific advantage of user story V is described in more detail below.

5.2. Control Development for a miniature hydraulic Wheel Loader

The implementation of User Story V is explained using a demonstrator developed in Fluid 4.0. For this purpose, a reduced and simplified version of a hydraulic wheel loader has been designed to represent the advantages of the machine AAS, which links to their component AASs. The demonstrator is shown in Figure 5.3. The main purpose of this demonstrator is to visualize the control development and highlight the advantages of using the AASs in combination with the new SM *Characteristic Curves* (refer to chapter 4.3.1).

To demonstrate this effect interactively, a user interface has been designed as visualized in Figure 5.3. A task was set to insert missing parameters to map a curve of a proportional flow control valve, which has been depicted in red colour inside the hydraulic scheme. This curve describes the amount of flow that passes through the valve, depending on the electrical current applied to the magnet from the control unit. The valve is responsible for the translational speed of the hydraulic cylinder.

As described in chapter 3 before, the values of the lookup table representing the curve need to be retrieved manually from a graphic inside the data sheet today. By applying the new SM *Characteristic Curves*, which provides the data machine-readable in a clearly defined syntax, the required information can be inserted automatically by the application with the click of a button. This leads to the desired movement of the wheel loader kinematics.

To demonstrate the occurrence of errors, a graphic for a different-sized valve has been prepared. Even though there are slight differences between the curves, the movement of the wheel loader changes tremendously. In the end, the demonstrator envisions the advantages of reducing the effort of implementing parameters from graphics and reducing the risk of mistakes. Both ways of parametrization of the control software, the manual way today and the automatically one based on component AAS with SM characteristic curve of the valve, can be compared interactively. The benefits are directly noticeable. The first concepts of this demonstration were shown at the special show during Hannover Messe 2025. The feedback from the visitors were fed back into the development steps of the workflow presented within this contribution.

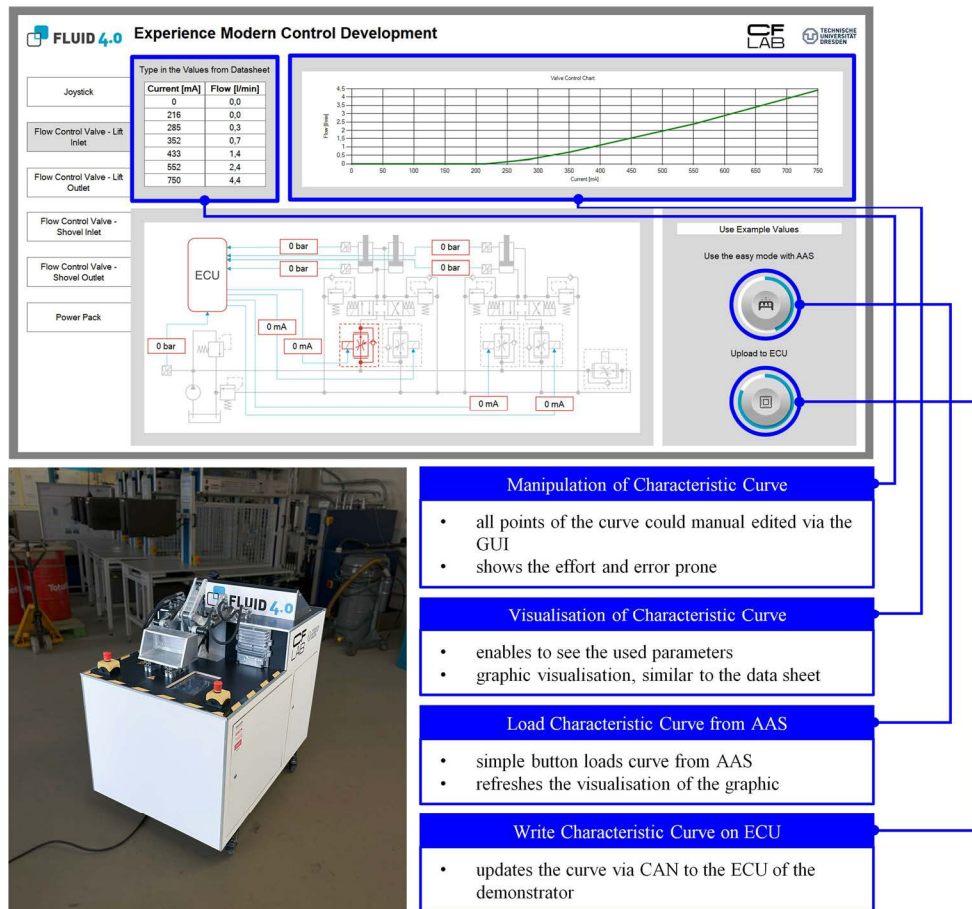


Figure 5.3. Miniature hydraulic wheel loader as demonstrator (bottom left) and interactive User Interface (top) to adjust control software manually or with the SM Characteristic Curve automatically

6. CONCLUSION AND OUTLOOK

This article highlighted the current problems in collaborative engineering of complex mechatronic systems such as hydraulic drive technology in mobile machines. Based on industry-relevant User Stories, an overall concept was developed using the AAS, and a total of five client applications. The essential building blocks for implementing collaborative engineering in fluid technology systems are information modelling, infrastructure, interfaces, and tools for automated signal flow. The administration shell technology, combined with the interaction possibilities offered by established web technologies and interfaces, forms a crucial basis for implementing these building blocks in the context of cross-company communication via dataspace. The semantic description of an asset's properties using the existing SMs allows the implementation of basic User Stories. However, there has been a lack of specific implementations for User Stories in collaborative

engineering like “Machine AAS with physical based linkage of component AAS”, “Semantic search within the dataspace” or “Automatic control software parametrisation”. With the help of the SMs “*Model Reconstruction*,” “*Generic Frame for Technical Data for Industrial Equipment in Manufacturing*,” and “*Characteristic Curves*” the most important User Stories in fluid engineering system design can be implemented. The first prototypes have already been built to carry this out.

The results of these investigations have already been incorporated into the IDTA standardization processes for partial model development. In the following work, User Stories III and IV will be implemented and the overall process will then be evaluated. The software modules will be made available as open source repositories on GitHub until the end of the project: <https://github.com/fluid40>

7. ACKNOWLEDGEMENT

This work has been funded by the European Union (EU) and supported by the German Federal Ministry for Economic Affairs and Energy (BMWE) 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 numbers 13IK039L and 13IK039C). 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.



8. REFERENCES

- [1] T. A. Abdel-Aty, E. Negri, and S. Galparoli, ‘Asset Administration Shell in Manufacturing: Applications and Relationship with Digital Twin’, presented at the 10th IFAC Conference on Manufacturing and Control MIM 2022, Nantes, France: Elsevier, Jun. 2022.
- [2] VDI 2206, *Entwicklungsmethodik für mechatronische Systeme*, Jun. 2004.
- [3] D. Ritz, M. Becker, J. Weber, and K. Schmitz, ‘Accelerating Product Development: The Effort of Asset Administration Shells for aggregated Systems’, presented at the The 19th Scandinavian International Conference on Fluid Power, SICFP’25, River Publishers, 2025, pp. 397–409. doi: 10.13052/tp-9788743808251A29.
- [4] T. Famic, ‘Live Manifold - Automation Studio™’. Accessed: Oct. 07, 2025. [Online]. Available: <https://www.famictech.com/en/Products/Automation-Studio-Live-Manifold/Product-Overview>
- [5] IEEE, ‘IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries’, *IEEE Std 610*, pp. 1–217, 1991, doi: 10.1109/IEEESTD.1991.106963.

- [6] K. Chen and D. Schaefer, 'MCAD-ECAD integration: Overview and future research perspectives', vol. 3, Jan. 2007, doi: 10.1115/IMECE2007-41705.
- [7] S. Rongen, N. Nikolova, and M. van der Pas, 'Modelling with AAS and RDF in Industry 4.0', *Comput. Ind.*, vol. 148, no. C, Jun. 2023, doi: 10.1016/j.compind.2023.103910.
- [8] F. M. for E. Affairs and Energy (BMW), 'Usage View of Asset Administration Shell'. 2019. [Online]. Available: <https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/2019-usage-view-asset-administration-shell.html>
- [9] R. Alt *et al.*, 'SEAMLESS INTEGRATION OF DEVICE AND FIELD DATA INTO THE SYSTEM SIMULATION OF A HYDRAULIC SERVO-PRESS USING AAS', presented at the 14th International Fluid Power Conference, Dresden, Germany, Mar. 2024.
- [10] S. Heppner, T. Miny, S. Kleinert, M. Becker, K. Schmitz, and R. Alt, 'Asset Administration Shells as Data Layer for Enabling Automated Simulation-based Engineering', presented at the 28th International Conference on Emerging Technologies and Factory Automation (ETFA), Sinaia, Romania: IEEE, Sep. 2023.
- [11] IDTA, 'Specification of the Asset Administration Shell - Part 1: Metamodel'. Industrial Digital Twin Association, 2025. doi: <https://doi.org/10.62628/IDTA.01001-3-1-1>.
- [12] IDTA, *Generic Frame for Technical Data for Industrial Equipment in Manufacturing*, 02003, Frankfurt Am Main, Germany., Mar. 2025. Accessed: Oct. 07, 2025. [Online]. Available: https://industrialdigitaltwin.org/wp-content/uploads/2025/03/IDTA-02003_Generic-Frame-for-Technical-Data.pdf
- [13] IDTA, 'Specification of the Asset Administration Shell - Part 5: Package File Format (AASX)'. Industrial Digital Twin Association, 2025. doi: <https://doi.org/10.62628/IDTA.01005-3-1>.
- [14] IDTA, 'Specification of the Asset Administration Shell - Part 2: Application Programming Interfaces'. Industrial Digital Twin Association, 2025. doi: <https://doi.org/10.62628/IDTA.01002-3-1-1>.
- [15] IDTA, 'Specification of the Asset Administration Shell - Part 4: Security'. Industrial Digital Twin Association, 2025. doi: <https://doi.org/10.62628/IDTA.01004-3-0-1>.
- [16] H. Pettenpohl, M. Spiekermann, and J. R. Both, 'International Data Spaces in a Nutshell', in *Designing Data Spaces: The Ecosystem Approach to Competitive Advantage*, B. Otto, M. ten Hompel, and S. Wrobel, Eds, Cham: Springer International Publishing, 2022, pp. 29–40. doi: 10.1007/978-3-030-93975-5_3.
- [17] S. ; et. al. Steinbuss, 'Usage Control in the International Data Spaces', International Data Spaces Association, 2021. doi: <https://doi.org/10.5281/zenodo.5675884>.
- [18] M. Bacco, A. Kocian, S. Chessa, A. Crivello, and P. Barsocchi, 'What are data spaces? Systematic survey and future outlook', *Data in Brief*, vol. 57, p. 110969, 2024, doi: <https://doi.org/10.1016/j.dib.2024.110969>.

- [19] IDTA, *Digital Nameplate for Industrial Equipment*, 02006-3-0, 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
- [20] IDTA, *Handover Documentation*, 02004-1-2, Frankfurt Am Main, Germany., Mar. 2023. Accessed: Oct. 07, 2025. [Online]. Available: https://industrialdigitaltwin.org/wp-content/uploads/2023/03/IDTA-02004-1-2_Submodel_Handover-Documentation.pdf
- [21] ECLASS, ‘ECLASS Standard’, What is ECLASS? [Online]. Available: <https://eclass.eu/en/eclass-standard>
- [22] ECLASS, ‘IRDI’. Accessed: Oct. 10, 2025. [Online]. Available: <https://eclass.eu/support/technical-specification/structure-and-elements/irdi>
- [23] IDTA, *Hierarchical Structures enabling Bills of Material*, 02011-1-1, 06/24. Accessed: Feb. 23, 2025. [Online]. Available: https://industrialdigitaltwin.org/wp-content/uploads/2024/06/IDTA-02011-1-1_Submodel_HierarchicalStructuresEnablingBoM.pdf
- [24] M. Ristin, *aas-core*. Python. [Online]. Available: <https://github.com/aas-core-works>
- [25] yworks, ‘yEd - graph editor’, yEd - graph editor High-quality diagrams made easy. [Online]. Available: <https://www.yworks.com/products/yed>

Biographies



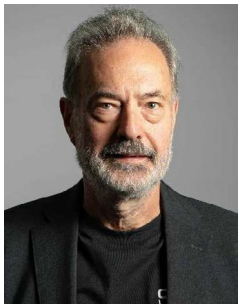
Paul Lukas Remde received the diploma degree in mechanical engineering from TU Dresden in 2024. He is working as scientific research assistant at Construction Future Lab GmbH since 2024 and is currently a developer inside the Fluid 4.0 project. His main task is the application of I4.0 technologies by developing client applications and demonstrators.



Benjamin Beck received the diploma degree in mechanical engineering from TU Dresden in 2013. He worked as research assistant and group leader for mobile hydraulic systems at the chair of Fluid-Mechatronic Systems at TU Dresden from 2013 to 2023. His research areas include functional safety, energy efficiency and function development of hydraulic drive systems with independent metering for mobile machines. Now he works in addition as strategy and sales lead at Construction Future Lab gGmbH and coordinates the joint research project Fluid 4.0.



Denis Ritz completed his mechanical engineering studies at TU Dresden in 2018 and is currently a research assistant at the Chair of Fluid Mechatronic Systems at TU Dresden. His research centers on control strategies and the development of mobile hydraulic drive systems, with a strong emphasis on systems engineering. Prior to his academic studies, he completed a vocational training as a farmer, providing him with practical insights into mobile machinery.



Jürgen Weber has studied mechanical engineering at the TU Dresden and successfully finished his doctorate in 1991. Until 1997, he was the active senior engineer at the former chair of Hydraulics and Pneumatics. This was followed by an approximately 13-year

industrial phase. He was active in various positions in the R&D department of the agricultural and construction machinery manufacturer CNH. Effective March 1st, 2010, Dr.-Ing. Jürgen Weber has been appointed university professor and chair of Fluid-Mechatronic System (Fluidtronics) at the TU Dresden. Since July 2018 he is the leader of the Institute of Mechatronic Engineering. He is also one of the founders and managing director of the Construction Future Lab gGmbH.



Tom Gneuß received the degree *Master of Science* from the TU Dresden in 2024. Since then, he is working as a research associate at the Chair for Industrial Communications. His main focus lies on the integration of brownfield devices into interoperable Industry 4.0 ecosystems.



Nico Braunisch received his diploma from the TU Dresden in 2019. Since then, he is part of the research group at the Chair for Industrial Communications. During his research, he has gained extensive knowledge in the field of IIoT, Industry 4.0, and automation.



Martin Wollschlaeger studied Electrical Engineering at Otto-von-Guericke University Magdeburg, earning his doctorate in 1991 and Habilitation in 2001. After working at ifak

Magdeburg, he became professor and chair of Industrial Communications at TU Dresden in 2003, where he also directs the Institute of Applied Computer Science. His research focuses on industrial communication and automation networks, including Ethernet-based systems, information modeling, and semantic integration. He contributes to standardization (DKE, IEC) and leads national and European research projects in areas such as Industrie 4.0 and 5G.