15thIFK
Proc. of the 15th International Fluid Power Conference <Symposium/Conference>
<DOI>

AAS-Orchestrated Production – End-to-End Digital Integration from Machine to Visualization

Micha Koebele[1], Jan Edler[2], Philipp Niemann-Stryczek[3], Heiko Baum[1]

[1] *FLUIDON GmbH, micha.koebele@fluidon.com, heiko.baum@fluidon.com*
[2] *Emerson Aventics GmbH, jan.edler@emerson.com*
[3] *IQstruct Engineering GmbH, philipp.stryczek@iqs-e.de*

Abstract.

This paper demonstrates how the Asset Administration Shell (AAS) can be used as an integration and orchestration layer for automated, product-specific processing of real machine data. Using a ballpoint pen assembly machine, standardized AAS mechanisms enable cloud-based visualization, energy analysis, and CO_2 accounting. The approach highlights product-related sustainability assessment in fluid power systems and illustrates the AAS as a key enabler for interoperable and scalable digital engineering workflows.

Keywords. Fluid 4.0, Asset Administration Shell (AAS), AAS-orchestrated data processing, Digital Twin, Fluid Power Systems, Energy and CO_2 analysis, Sustainability assessment, Industry 4.0

1. Introduction

The increasing demands for transparency, energy efficiency, and sustainability in industrial production require new, integrated digital concepts for information and data processing. In fluid-based production systems in particular, there has been a lack of standardized approaches for consistently utilizing real machine data for analysis, visualization, and product-related sustainability assessments.

This article presents an end-to-end solution orchestrated entirely by the Asset Administration Shell (AAS) [1], which is demonstrated using a ballpoint pen assembly machine as an example. The aim is to present the end-to-end flow of information from the real machine via cloud-based services to product-related CO_2 accounting and visualization in a comprehensible manner.

To provide context for the application example, the following chapter first introduces the machine used.

2. Machine and System Overview

The left-hand side of **Figure 2.1** shows the machine used to demonstrate the AAS-orchestrated data flow. The machine is an *Industry 4.0* [2] demonstrator from the *Fluid 4.0*

[3] research project. The machine, built by project partner *Emerson Aventics* [4], produces ready-to-use ballpoint pens from four basic components: the housing, refill, spring, and cap. The actual assembly process is carried out by valve-controlled pneumatic grippers and rotary actuators, while electromechanical linear axes transfer the components from the transfer points of the storage magazines to the respective assembly positions. After successful assembly, the machine assigns a unique serial number to each manufactured ballpoint pen, which is printed on a Fluid 4.0 product sticker and attached to the pen.
This serial number then allows the owner to access the product-related production data for their ballpoint pen using any AAS-enabled application.

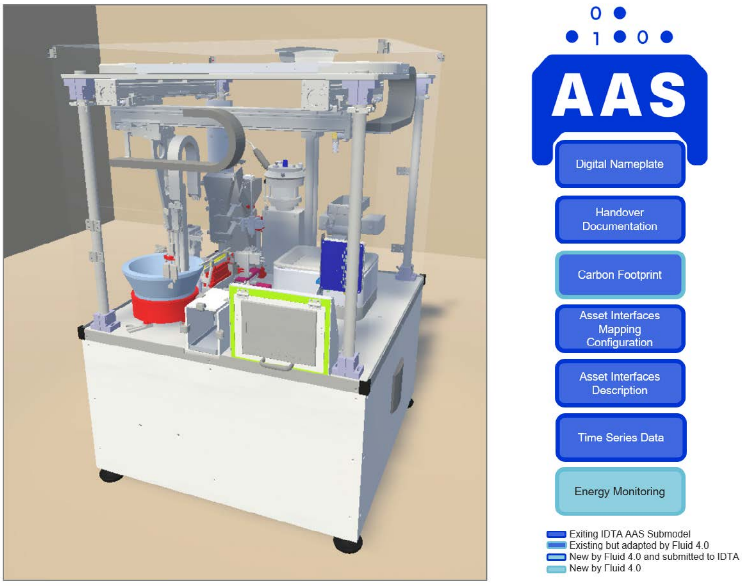

Figure 2.1 : Ballpoint pen assembly machine and AAS submodels used

During the assembly process, the machine collects the data listed below and continuously transmits it to the *MQTT broker* [5] of the project partner *Fluidon* [6]:

- pneumatic air consumption,
- electrical energy consumption,
- valve signals
- Positions of the linear axes
- serial number of the ballpoint pen being produced.

This production data forms the basis for AAS-orchestrated data processing.
The right-hand side of Figure 2.1 shows an overview of the AAS submodels used in the application example. The legend distinguishes between submodels that have already been standardized by the *IDTA* [7] and those that have been newly developed or expanded as part of the Fluid 4.0 research project.

3. AAS-ORCHESTRATED DATA PROCESSING WORKFLOW

Data processing is carried out via a data processing workflow created in the SaaS development environment *Fluidon Cube* [8]. For the sample application, the processing of production data is deliberately implemented in small steps in order to present the individual interactions with the AAS submodels involved in a didactically comprehensible manner. **Figure 3.1** shows the sequence of process steps with which the project partners implemented AAS-orchestrated data processing.

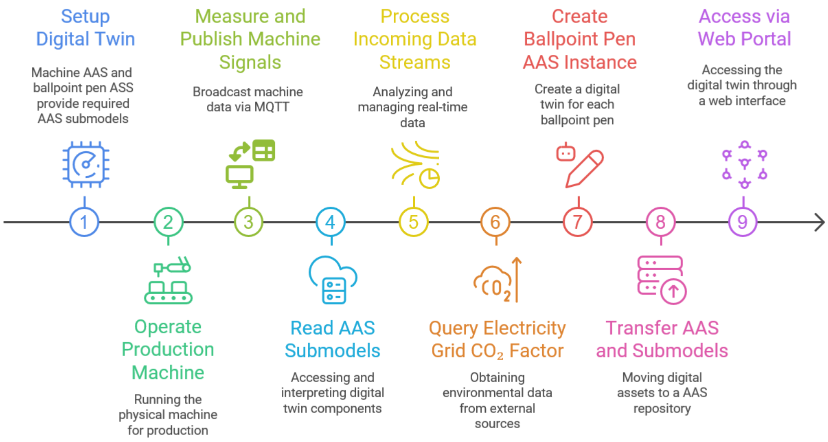

Figure 3.1 : Overview of the process steps

3.1. *Preparatory work*

In the first process step, an instance AAS of the machine and a type AAS of the ballpoint pen were first created (see Figure 3.1-1, *"Setup Digital Twin"*). The following process step (see Figure 3.1-2, *"Operate Production Machine"*) is representative of the preparatory work on the machine. Pneumatic pressure and flow sensors, a system for recording the electrical power consumption of the linear axes, and a label printer were retrofitted. The third process step (see Figure 3.1-3, *"Measure and Publish Machine Signals"*) represents the integration of all necessary signals into the control system's data acquisition, their processing, and their transmission to the MQTT broker.

3.2. *The data processing workflow*

The fourth process step (see Figure 3.1-4, *"Read AAS Submodels"*) symbolizes the start of the actual AAS-orchestrated data processing. After the workflow has retrieved the machine's AAS submodels and connected to the MQTT broker, signal processing takes place (see Figure 3.1-5, *"Process Incoming Data Streams"*). Following the query of the current CO_2 emission factor (see Figure 3.1-6, *"Query Electricity Grid CO_2Factor"*) and calculating the CO_2 parameters, an instance AAS of the ballpoint pen just produced is created in the seventh process step (see Figure 3.1-7, *"Create Ballpoint Pen AAS Instance"*).

3.3. *Provision and use of data*

In the eighth process step, the generated instance AAS and the associated submodels are transferred to any AAS repository server (see Figure 3.1-8, *"Transfer AAS and Submodels"*).

The information contained therein or linked to it is then available for interoperable access, for example via a web portal or other AAS-enabled applications (see Figure 3.1-9, *"Access via Web Portal"*).

4. PARTICIPATING ASSET ADMINISTRATION SHELLS

At the heart of AAS-orchestrated data processing are the instance AAS of the machine and the type AAS of the ballpoint pen. As already presented in the right-hand section of the image at Figure 2.1, both standardized AAS submodels and those developed jointly with other Fluid 4.0 project partners were used to create them.

4.1. Instance AAS of the ballpoint pen assembly machine

With the aim of graphically illustrating the structural design of an AAS, **Figure 4.1** shows a screenshot of the instance AAS of the machine. In the context of the application example, the instance AAS can be accessed via the AAS repository server of the project partner *IQstruct Engineering* [9]. Access to the machine's AAS instance is controlled by IQstruct Engineering via an access rights management system based on preliminary work from *Eclipse BaSyx* [10] and prototypically implemented as part of the Fluid 4.0 research project.

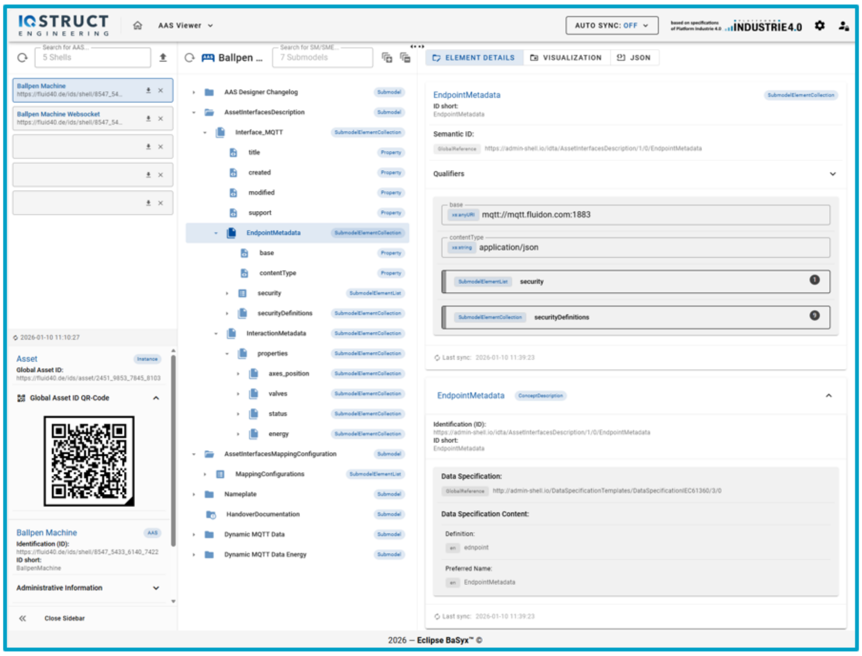

Figure 4.1 : Instance AAS of the ballpoint pen assembly machine

The middle section of the image at Figure 4.1 shows all the submodels contained in the AAS. The two AAS submodels *Asset Interface Description* (AID) [11] and *Asset Interface Mapping Configuration* (AIMC) [12] are used for the application example.
The right-hand side of the image shows elements from the *Submodel Element Collection* (see[1, p. 56]) *"EndPointMetadata"* of the AID submodel (see[11, p. 16]). All information

required for the automatic connection of the data evaluation workflow can be read from the elements in this collection. In the application example, the MQTT broker of the project partner Fluidon is entered here.
The AIMC submodel is used for the formal description of the mapping of technical machine interfaces to semantically defined AAS properties. It enables a clear decoupling between the physical or IT data source (e.g., fieldbus, MQTT, OPC UA) and the semantic structure of the administration shell. This allows production data to be automatically collected, interpreted, and further processed by the data processing workflow without project-specific adjustments. The submodel thus forms the central basis for AAS-orchestrated, scalable data processing.

4.2. Type AAS of the ballpoint pen

Figure 4.2 shows a screenshot of the type AAS of the ballpoint pen. The submodels required for the application example are listed in the left-hand part of the image at Figure 4.2. In addition to the *Nameplate* submodel [13], these are the *Handover Documentation* [14] and *Time Series Data*[15] submodels, as well as the *Carbon Footprint* [16] submodel, which was expanded as part of Fluid 4.0.

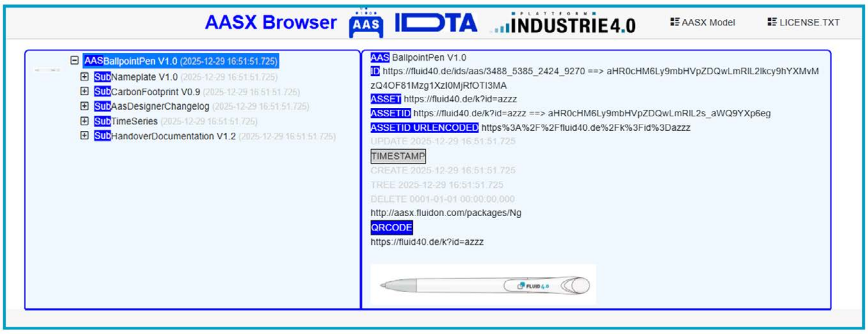

Figure 4.2 : Type AAS of the ballpoint pen

To demonstrate interoperability, the Type AAS in the application example is provided via Fluidon's AAS repository server and not via IQstruct Engineering's AAS repository server. A direct comparison of the screenshots from Figure 4.1 and Figure 4.2 shows that the two AAS are also managed on two different AAS repository server implementations. IQstruct Engineering uses the Eclipse *BaSyx Java V2* [17], while Fluidon uses the *Eclipse AASX Server* [18]. This is to illustrate that it is irrelevant for AAS-orchestrated data processing where and with which server implementation the participating AAS are hosted. Interoperable access to information is possible at any time via the standardized AAS mechanisms.
The following chapter describes how the two administration shells and their submodels are used in AAS-orchestrated data processing.

5. AAS-ORCHESTRATED DATA PROCESSING

The AAS-orchestrated data processing workflow is explained using the sequence diagram shown in **Figure 5.1**. Four actors are involved in the workflow:

- *Operator*, who interacts with the data processing workflow
- *Fluidon Cube*, which hosts the workflow
- *Web APIs*, representing the web services involved
- *Real Machine*, as the process data source

It is clear that the actors *Operator* and *Real Machine* are only actively involved in the process at the beginning and end. The actual data processing, on the other hand, is fully automated, as the distributed systems involved can be accessed via the standardized, interoperable access mechanisms of Asset Administration Shell.

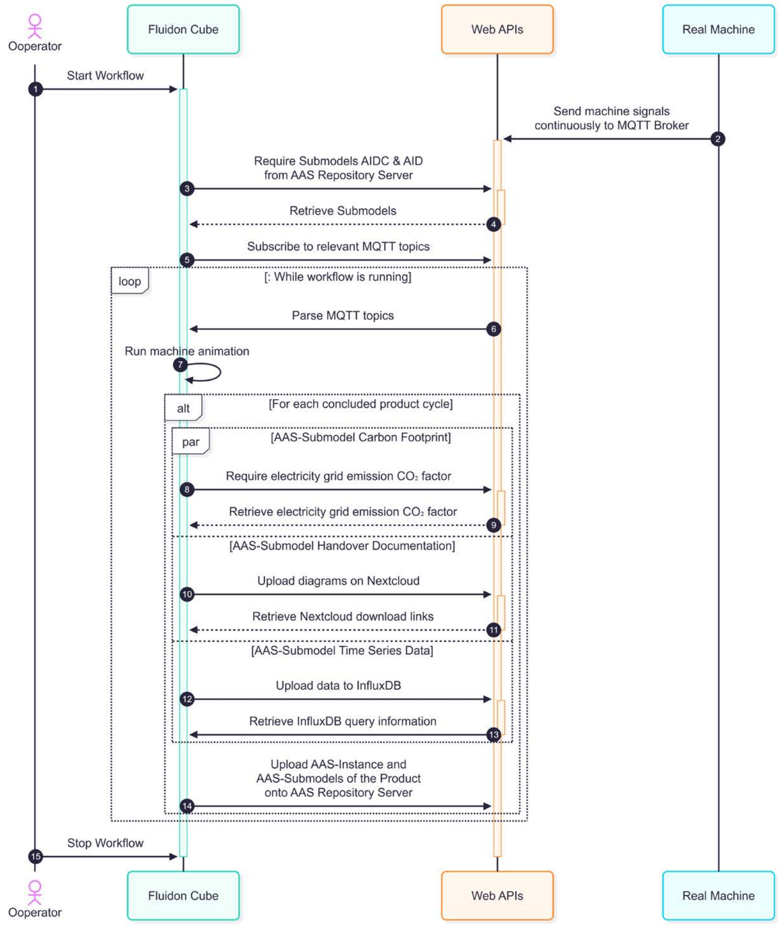

Figure 5.1 : Sequence diagram of AAS-orchestrated data processing

5.1. Start of AAS-orchestrated data processing

In the first step (see Figure 5.1-1), the operator starts the data processing workflow. It is irrelevant whether the machine is already in production at this point or – as shown in the figure (see Figure 5.1-2) – only starts the production process after the workflow has been started.

5.2. Data transfer and data preprocessing

The data processing workflow consists of several tasks, the respective functional scope of which is tailored to the process steps described in Figure 5.1. Each task performs a maximum of one AAS-supported interaction with external services in order to keep the data processing sequence comprehensible for the user of the application example. In the example, the workflow has 8 processor cores at its disposal, so that data processing takes place in parallel tasks. **Figure 5.2** shows the view of the data processing workflow in the *Fluidon Cube* development environment [8].

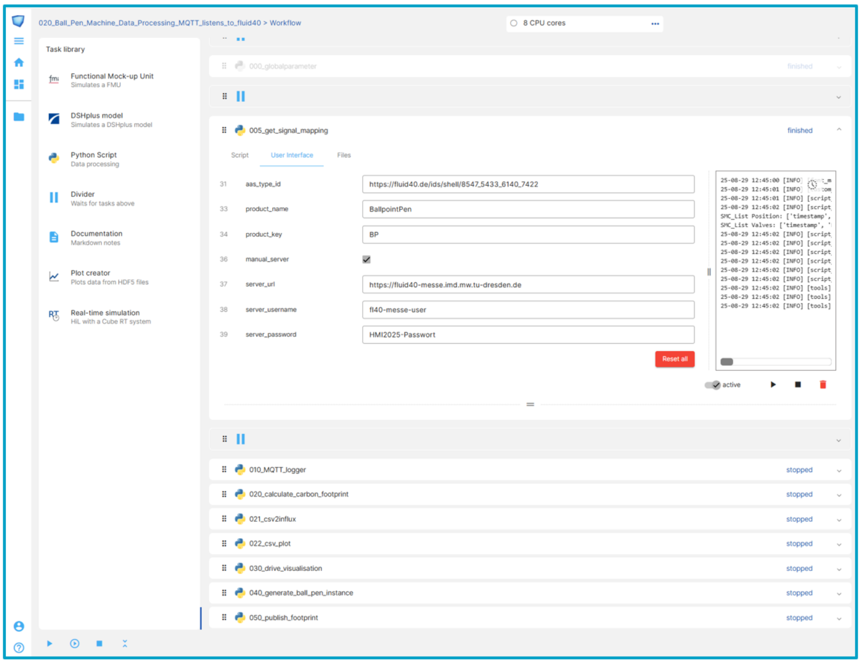

Figure 5.2 : Fluidon Cube data processing workflow [8]

The individual data processing steps were implemented using Python scripts, specifically using freely accessible libraries from the BaSyx environment [10], such as *AAS Python SDK* [19] or *aas-core-python* [20], as well as AAS-specific libraries [21], which were developed as part of Fluid 4.0 and will be handed over to the BaSyx community after the project is completed. The data-processing Python scripts can be viewed by the user on request and can serve as a template for using the BaSyx libraries in their own applications.

The user specifies under *31 aas_type_id* which machine the workflow is connected to and defines the name of the product under *33 product_name*. In addition, they specify on which AAS repository server the generated product instances are to be stored. In the application example, a repository server of the project partner *TU Dresden* [22] is used for this purpose. At the beginning of the process flow (see Figure 5.1-3), the workflow first authenticates itself with the AAS repository server and receives the AAS submodels AID and AIMC from it (see Figure 5.1-4). After successfully connecting to the MQTT broker (see Figure 5.1- 5), the workflow subscribes to the corresponding topics and starts continuous data recording.

The workflow receives the current production data (see Figure 5.1-6) and stores it temporarily.

5.3. Visualization of production data

Some of the data is processed numerically and forwarded to a 3D visualization of the assembly machine in the workflow dashboard (see Figure 5.1-7). **Figure 5.3** shows this dashboard, which integrates online graphics to display the temporal progression of selected process signals in addition to the 3D visualization.

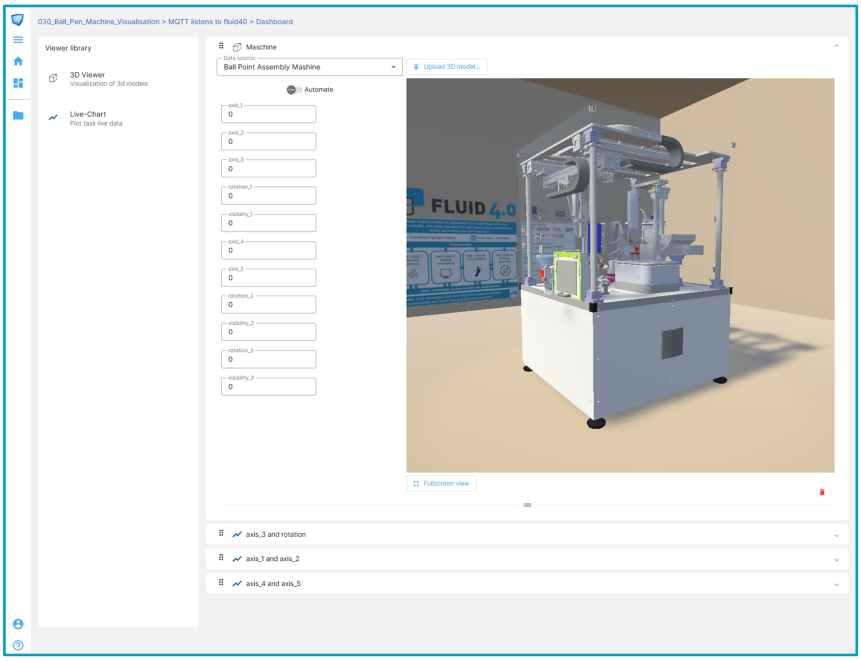

Figure 5.3 : Visualization of the ballpoint pen assembly machine

The visualization can be displayed:

- **live,** parallel to the ongoing production process, or
- **retrospectively** based on the stored time series data.

This creates a digital twin of the machine that visualizes the current or historical machine status regardless of location.

5.4. Energy analysis, CO_2 accounting, and documentation

As soon as the machine reports that the production of a ballpoint pen has been completed, energy analysis, CO_2 accounting, and product-related documentation are initiated in parallel tasks. Based on the recorded production data, the energy consumption per production cycle is determined, distinguishing between

- pneumatic energy (amount of compressed air used) and
- electrical energy (movements of the axes).

This differentiation is particularly important for fluid technical systems, as pneumatic energy is often insufficiently taken into account in classic CO_2 balancing. In the application example, this circumstance is taken into account by using a *conversion factor* that converts the consumed pneumatic energy into an electrical power equivalent for its generation. **Figure 5.4** shows the corresponding input field for this conversion factor in the workflow task. The conversion factor is multiplied by the balanced pneumatic energy; the resulting electrical equivalent is then added to the balanced electrical energy.

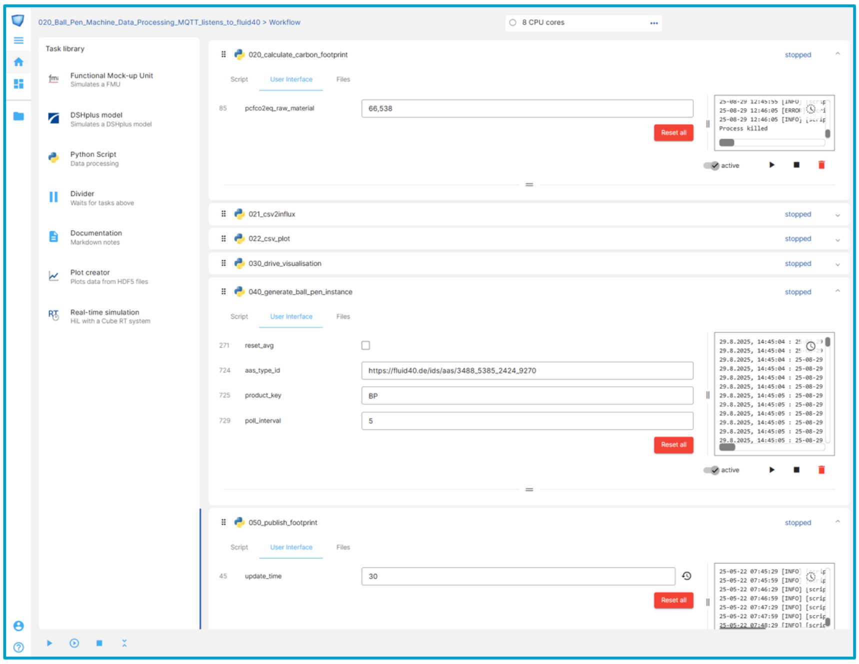

Figure 5.4 : Workflow detail for CO2 accounting

To determine the CO_2 footprint, the CO_2 emission factor of the power grid valid at the respective time of production is automatically queried via external servers (see Figure 5.1-8 and -9). Multiplying this emission factor by the determined electrical energy results in an individual, product-related CO_2 value for each ballpoint pen. This is stored in the *Carbon Footprint* submodel of a newly created product instance AAS for the ballpoint pen.
At the same time, another task generates diagrams (see Figure 5.1-9) that show, for example, the axis positions of the handling system or the development of pneumatic and electrical energy consumption over time. The diagrams are stored on a Nextcloud drive belonging to the project partner Fluidon; the download links generated (see Figure 5.1-11) are stored in the *Handover Documentation* submodel of the product instance AAS.

In addition, the production data is stored long-term in a time series database (InfluxDB) (see Figure 5.1-12). The associated database queries (see Figure 5.1-13) are stored in the *Time Series Data* submodel of the product instance AAS. An AAS-based interface thus enables targeted access to historical production data – for example, based on the serial number or a time period – which supports further analysis even long after production has been completed.

5.5. *Uploading the instance AAS of the ballpoint pen*

Once the data analysis and submodel creation are complete, a product-related instance AAS for the ballpoint pen is generated in the final process step (see Figure 5.1-14). To do this, the type AAS presented in Figure 4.2 is loaded and instantiated with the results determined in the workflow, creating a concrete product instance AAS. Finally, this is transferred to the previously specified AAS repository server.

6. USER ACCESS TO THE BALLPOINT PEN AAS

The owner of a ballpoint pen can use a web application to retrieve the corresponding product instance AAS by entering the serial number printed on the product. This access provides both product-related sustainability indicators such as the carbon footprint and process-related visualizations and documents. **Figure 6.1** shows an example of user access via the Fluid 4.0 research project website. The image shows an entry from the *Carbon Footprint* submodel.

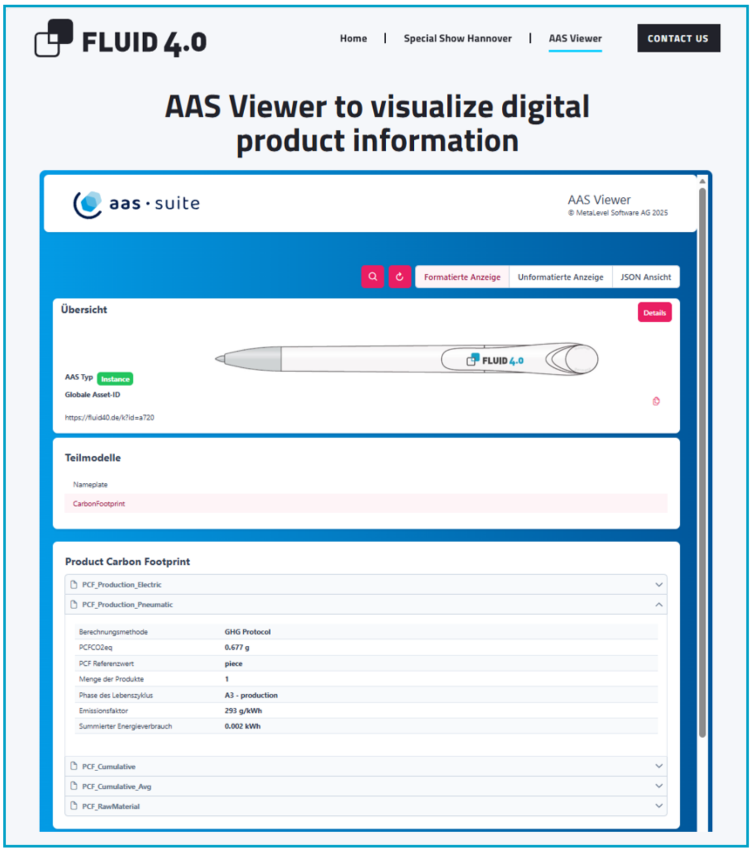

Figure 6.1 : User access to the ballpoint pen AAS

The functionality was implemented using a browser plugin provided by the associated project partner *Metalevel* [23]. This plugin can be integrated into any website with minimal effort and is connected to any AAS repository server on the backend side by Metalevel. **Figure 6.2** illustrates user access using a sequence diagram. The *client application* represents the browser plugin in the user's web application and the backend at Metalevel.

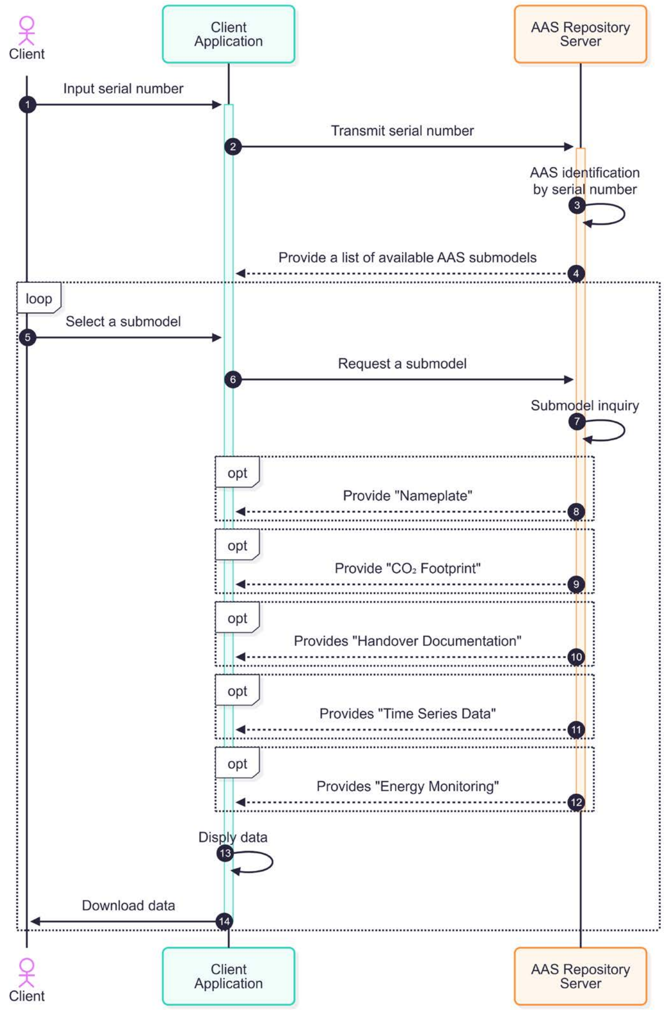

Figure 6.2 : Sequence diagram of user access

The serial number entered by the user (see Figure 6.2-1) is transmitted to the *client application*, which then queries the AAS repository server for the AAS associated with the serial number (see Figure 6.2-2). The AAS repository server determines the appropriate product instance AAS (see Figure 6.2-3) and transmits the metadata required to access the contained submodels to the *client application* (see Figure 6.2-4).
The user selects the relevant submodel in the browser plugin (see Figure 6.2-5), which is then requested by the *client application* from the AAS repository server (see Figure 6.2-6). The AAS repository server determines the requested submodel and transfers its contents to

the browser plugin (see Figure 6.2-7 to 11), where they are visualized (see Figure 6.2-12). The user can then access and download the displayed data. **Figure 6.3** shows an example of a results graph depicting the axis positions of the handling system during the production of a ballpoint pen.

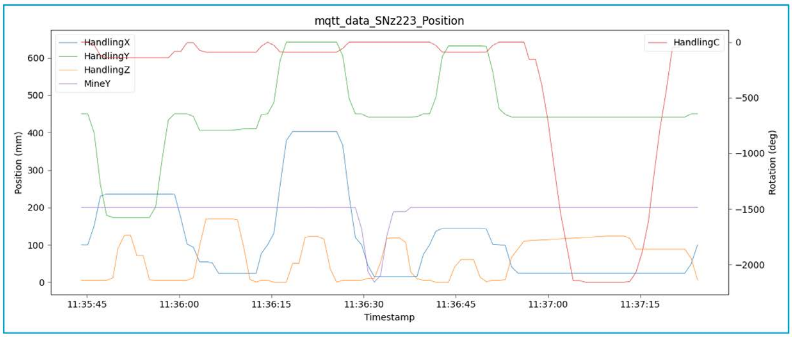

Figure 6.3 : Result diagram from the handover documentation

Furthermore, access to the product AAS is not limited to a single application. Thanks to the standardized interfaces and semantic descriptions of the Asset Administration Shell, the stored information can also be used by any AAS-enabled application. These include, for example, mobile applications for product information, analysis tools for quality or sustainability assessments, and engineering or simulation tools that access historical production data.

Another such use case is access to the time series data referenced in the *Time Series Data* submodel. Authorized users can use the database queries stored there to access specific historical production data, for example, to analyze process deviations, track individual batches, or perform further statistical evaluations.

The separation between data storage, semantic description, and access via standardized AAS mechanisms thus enables flexible and future-proof use of product data throughout the entire lifecycle. New applications or services can be connected to the product AAS without adapting the existing data processing, allowing the AAS to function as a central integration interface for product-related information.

7. SUMMARY

This article uses an industrial demonstrator to show how the Asset Administration Shell can be used as a consistent integration and orchestration layer to automatically collect, process, and provide real machine data for specific products. Using the example of a ballpoint pen assembly machine, it was demonstrated how production data can be linked to cloud-based services via standardized AAS mechanisms and used for visualization, energy analysis, and CO_2 accounting.

The solution presented illustrates in particular the interaction between machine instance AAS and product type or product instance AAS, as well as the role of standardized and project-specific extended AAS submodels for modular data processing. Through AAS-orchestrated workflow execution, central processing steps – from data acquisition and analysis to documentation – are fully automated and decoupled from specific IT or server implementations.

A particular focus was placed on the product-related sustainability assessment of fluid technology systems. The separate accounting of pneumatic and electrical energy and its consolidation into a product-specific carbon footprint shows how AAS-based data structures can be used for transparent and reproducible sustainability assessment.
Finally, it was presented how the generated product instance AAS can be used by different AAS-enabled applications via standardized interfaces. User access via a web application is only one example of a use case. Overall, the example underscores the potential of the Asset Administration Shell as a central basis for interoperable, scalable, and sustainable digital engineering and production processes, especially in the context of fluid power production systems.

8. 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 Fluid4.0 project – 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 13IK039E, 13IK039D, and 13IK039H). 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 Energy
on the basis of a decision by the German Bundestag

9. REFERENCES

[1] IDTA, 'Specification of the Asset Administration Shell - Part 1: Metamodel'. Industrial Digital Twin Association, 2025. [Online]. Available: https://industrialdigitaltwin.org/en/content-hub/aasspecifications/specification-of-the-asset-administration-shell-part-1-metamodel-idta-number-01001

[2] 'Industrie 4.0', *Wikipedia*. Dec. 17, 2025. Accessed: Jan. 10, 2026. [Online]. Available: https://de.wikipedia.org/w/index.php?title=Industrie_4.0&oldid=262510167

[3] 'Joint Research Project Fluid 4.0 – 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.' Accessed: Jan. 03, 2026. [Online]. Available: https://fluid40.de/

[4] 'AVENTICS | Pneumatische Automatisierungslösungen'. Accessed: Jan. 10, 2026. [Online]. Available: https://discreteautomation.emerson.com/de/brands/aventics

[5] 'MQTT', *Wikipedia*. Dec. 17, 2025. Accessed: Jan. 10, 2026. [Online]. Available: https://de.wikipedia.org/w/index.php?title=MQTT&oldid=262501931

[6] 'FLUIDON | Ihr Experte für Fluidtechnik und Mechatronik', Mainpage. Accessed: Jan. 03, 2026. [Online]. Available: https://fluidon.com/

[7] 'IDTA – Der Standard für den Digitalen Zwilling', IDTA. Accessed: Jan. 03, 2026. [Online]. Available: https://industrialdigitaltwin.org/

[8] 'FLUIDON | Cube'. Accessed: Jan. 03, 2026. [Online]. Available: https://cube-lab.app/

[9] 'Home < IQstruct Engineering', IQstruct Engineering. Accessed: Jan. 10, 2026. [Online]. Available: https://iqs-e.de/

[10] 'BaSyx Wiki — Eclipse BaSyx™'. Accessed: Jan. 11, 2026. [Online]. Available: https://wiki.basyx.org/en/latest/index.html

[11] IDTA, 'AAS-SM - Asset Interfaces Description'. Industrial Digital Twin Association, Frankfurt am Main, Germany, Jan. 2024. [Online]. Available: https://industrialdigitaltwin.org/en/wp-content/uploads/sites/2/2024/01/IDTA-02017-1-0_Submodel_Asset-Interfaces-Description.pdf

[12] IDTA, 'AAS-SM - Asset Interfaces Mapping Configuration'. Industrial Digital Twin Association, Frankfurt am Main, Germany, Jan. 2024. [Online]. Available: https://industrialdigitaltwin.org/en/wp-content/uploads/sites/2/2024/06/IDTA-02027-1-0_Submodel_AssetInterfacesMappingConfiguration.pdf

[13] IDTA, 'AAS-SM - Digital Nameplate for Industrial Equipment'. Frankfurt am Main, Germany, Oct. 2025. [Online]. Available: https://industrialdigitaltwin.org/en/wp-content/uploads/sites/2/2025/10/IDTA-02006-3-0-1_Submodel_Digital-Nameplate.pdf

[14] IDTA, 'AAS-SM - Handover Documentation'. Frankfurt am Main, Germany, Mar. 2023. [Online]. Available: https://industrialdigitaltwin.org/en/wp-content/uploads/sites/2/2023/03/IDTA-02004-1-2_Submodel_Handover-Documentation.pdf

[15] IDTA, 'AAS-SM - Time Series Data'. Frankfurt am Main, Germany, Mar. 2023. [Online]. Available: https://industrialdigitaltwin.org/en/wp-content/uploads/sites/2/2023/03/IDTA-02008-1-1_Submodel_TimeSeriesData.pdf

[16] IDTA, 'AAS-SM - Carbon Footprint'. Frankfurt am Main, Germany, Mar. 2025. [Online]. Available: https://industrialdigitaltwin.org/en/wp-content/uploads/sites/2/2025/03/IDTA-02023_Submodel_CarbonFootprint.pdf

[17] 'eclipse-basyx/basyx-java-server-sdk'. Accessed: Jan. 27, 2026. [Online]. Available: https://github.com/eclipse-basyx/basyx-java-server-sdk

[18] 'eclipse-aaspe/server: C# based server for AASX packages'. Accessed: Jan. 27, 2026. [Online]. Available: https://github.com/eclipse-aaspe/server

[19] *eclipse-basyx/basyx-python-sdk*. (Jan. 10, 2026). Python. Eclipse BaSyx™. Accessed: Jan. 11, 2026. [Online]. Available: https://github.com/eclipse-basyx/basyx-python-sdk

[20] 'aas-core-works', GitHub. Accessed: Jan. 05, 2026. [Online]. Available: https://github.com/aas-core-works

[21] 'Fluid4.0', GitHub. Accessed: Jan. 13, 2026. [Online]. Available: https://github.com/fluid40

[22] work T. ServiceCenterStudium, 'Technische Universität Dresden', TU Dresden. Accessed: Jan. 11, 2026. [Online]. Available: https://tu-dresden.de?set_language=de

[23] M.-L. S. AG, 'Meta-Level Software AG', Meta-Level Website. Accessed: Jan. 13, 2026. [Online]. Available: https://meta-level.de

Biographies

Micha Koebele received his M.Sc. degree from RWTH Aachen University in 2021. Since 2024, he has been working at FLUIDON GmbH as a development engineer, focusing on dynamic system simulation, model-based engineering, and automated simulation workflows for hydraulic and pneumatic systems. His work particularly addresses the integration of simulation models with digital twin concepts and Asset Administration Shell–based data structures in the context of Industry 4.0.

Heiko Baum studied mechanical engineering at RWTH Aachen University and completed his doctorate in 2001. Since 2001, he has been the managing director of FLUIDON GmbH, where he focuses on model-based engineering, digital twins, and automated simulation workflows for hydraulic and pneumatic systems. His long-standing experience in the simulation and development of fluid power systems bridges industrial applications and research projects in the context of Industry 4.0 and the Asset Administration Shell.

Jan Edler studied mechatronics at the University of Magdeburg and graduated with a Dipl.-Ing. After serving as the construction leader for all customer-specific pneumatic cylinders for Aventics GmbH, he began introducing the Emerson Intelligent Automation Software stack into global factories. His vast knowledge of pneumatic construction, data generation, and analytics converges with his close collaboration with universities and the digital twin initiatives driving the AAS.

Philipp Niemann-Stryczek has studied Computer Science at Trier University of Applied Science after graduating as a State-certified Technician, specialized in Automation Technologies at Aachen Technical College. Before founding and becoming Managing Director of IQstruct Engineering GmbH in 2018 he has been commissioning automobile production lines and developed distributed server backend systems. His passion creating digital twins is widely used within AAS related software and virtual commissioning solutions.