OPEN INDUSTRY 4.0 ALLIANCE

Whitepaper

Introduction to AAS

Let's Enter the World of Digital Twins and AAS

Let's begin with a basic introduction to the concept of the digital twin. No, this isn't just another definition, and we aren't assuming there's a single, definitive interpretation. Instead, we aim to establish a shared understanding of the underlying technical foundation of a digital twin: the **Asset Administration Shell (AAS).** Once we've achieved that, we'll systematically delve into practical use cases and offer explanations.

A digital twin defines both properties and functionalities. Whether we're talking about a product or a production process, the primary distinction lies in the specification of properties and functionalities. While a product may possess distinct characteristics compared to a factory manufacturing it, we can simplify matters in this way. Within the context of AAS, we'll refer to these sets of properties as "submodels" and functionalities as "skills."

The concept of AAS naturally involves a distributed information approach, meaning it's not confined to a single piece of software or a sole server containing all the details. It's important to grasp that information can be spread throughout your company and even extend to your partners in the ecosystem. Within the AAS framework, you can pinpoint where specific information is located. For instance, your engineering data is securely stored internally, while data related to your products can be accessed through your customer portal. Both types of data are connected to the same asset, however, they come from different perspectives. AAS enables the integration of these diverse details by using unique identifiers.

You might be wondering, *"is this really necessary?"* The answer is yes, indeed. While awareness of the AAS is growing, it remains somewhat isolated in its own sphere – an expanding one at that. Therefore, it's crucial to burst this bubble. Consider your suppliers, partners, and customers: are they familiar with it? Do any of your partners provide data exchange based on AAS? That being said, why should you or your partners use AAS?

Why Should I Use AAS?

If you're already dealing with various digital twins, you might be wondering*: "what sets AAS apart from existing solutions, and why should I consider making the switch?"*

Answering this question comprehensively varies and, of course, also depends on your perspective. However, let's ease into the discussion step by step: AAS represents an open approach driven by a diverse community in the Industry 4.0 area. Unlike existing, proprietary formats, AAS isn't the brainchild of a single company or association. It serves as a shared foundation to define a digital twin, offering a unified description model and, consequently, a normalized API for practical application.

In theory, anyone can create submodels for a digital twin. However, the true value emerges when these submodels are reusable and unified. When every entity employs the same submodel for a particular type of information, integration becomes more straightforward. There's no need for complex mappings between different submodels; you can simply use the unified one.

Now, here's the critical point – the acceptance of AAS is closely linked to the speed at which valuable submodels are developed and made available. If the pace of submodel development doesn't align with companies' expectations, we'll encounter challenges when it comes to integrating products and gaining acceptance in the next phase.

Types of AAS – and Why You Should Know Them

Not particularly tech-savvy? Great! Here's a brief detour into a detail of AAS that you should be aware of, regardless of your job role. It's crucial to understand and categorize approaches and solutions from both a business and technical standpoint. There are three types that we want to inspect:

- Type 1: **passive type** - via file exchange based on XML
- Type 2: **reactive type** - via microservices using a unified API
- Type 3: **proactive type** - communication between digital twins.

Type 1 can indeed have value in specific use cases but relying exclusively on type 1 is just the beginning. Let me break it down: having an XML file is the initial step. Next, you require a system for exchanging these files. This is similar to what we've been doing with FTP servers and CSV files for the past two decades. The only new aspect here is that now you're using a standardized XML file. Sounds somewhat old-fashioned, doesn't it? Of course, you can opt for a more advanced technical method of exchange. However, in the end, it still involves handling XML files, or a collection of XML files bundled together as an archive, which requires document parsing and processing.

On the other hand, with type 2, you can make use of microservices to obtain the specific data you need for processing. For example, if you're interested in a particular submodel, you can request that data directly. There is no need to parse all the other available submodels and extract them one by one. Ultimately, this will increase the effectiveness of your development process and reduces the potential for errors.

Now, let's move a step forward: if your goal is to enable machines to communicate with each other (type 3), type 2 should be seen as the necessary foundation. Here's a simple recommendation: whenever feasible, type 2 should be the preferred method for managing AAS. For a type 3 approach, we'll require skills defined for a digital twin. And yes, these types of behaviors must also be standardized to facilitate seamless integration. For now, we're focusing on properties. The next significant challenge is establishing a consensus on skills within a broader community to ensure acceptance and enable product integration.

The conclusion is quite easy: all types of approaches are valid in some way, and the choice depends on the specific use case.

Fields of Practical Approaches

There are several aspects when and how to use AAS. We have collected potential topics of interest from our Members of the Open Industry 4.0 Alliance, listed below:

- Digital nameplate (DNP): unified identification of an asset
- Digital product passport (DPP): unified product description, essential details
- Digital calibration certificate (DCC): replace paper-based certificate, rise automation
- Software bill of material (SBOM): unified base for detection and prevention
- Modular approaches for plug-n-play AAS solutions
- Central asset management based on AAS

Since every topic is worth a dedicated document, we'll enrich our series by listed topics and more. Enjoy browsing our publications! Do you think some topics are missing? Just get in contact and bring in your minds!

Author & Resources

Authors involved

- Ricardo Dunkel (ricardo.dunkel@openindustry4.com)

Resources used

The MetaModel of the AAS:
https://industrialdigitaltwin.org/wp-content/uploads/2023/06/IDTA-01001-3-0_SpecificationAssetAdministrationShell_Part1_Metamodel.pdf

Details of the Asset Administration Shell, Part 1:
https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.pdf

Details of the Asset Administration Shell, Part 2:
https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part2_V1.pdf