<AutomationML/>

The Glue for Seamless Automation Engineering

Interrelation of Asset Administration Shell and AutomationML

Position paper

State: May 2021

Background

The AutomationML association has been developing a standard for modelling processes and resources in the production and industrial environment for more than 10 years. Both, the data format and the content standardization of different disciplines involved in the engineering and their domain models, are considered. This also includes the description of components with a focus on the engineering process, covering capabilities to represent all essential engineering information syntactically and semantically.

The results of the work performed in AutomationML association are a valuable resource for future endeavours in information modelling as well as for digitalisation efforts in automation engineering. This paper aims at presenting the value available through the results of AutomationML workgroups and shows ways in which collaboration with other organisations in the standardization of Industry 4.0 can benefit users.

For several years now, more and more organizations in the Industry 4.0 environment have been working on the description of components for the operation and maintenance processes. The focus here is on use cases from the Industry 4.0 environment such as predictive maintenance or the lifecycle information management of production equipment. This interconnects efficiently with the goal of the AutomationML community to provide a lifecycle file for an automation system based on AutomationML.

Major features of AutomationML that are proven and standardised, and which will likely benefit other digitalisation efforts in the context of Industry 4.0, are:

- A rich model for engineering information with many specialised discipline-specific submodels that can be utilised freely without usage barriers
- A robust serialisation for engineering toolchains that has already been adopted by some of the major automation vendors
- Mechanisms for traceability and versioning
- Coverage of topologies and cross-referencing of models built-in to the AutomationML base model
- Robustness to proprietary contents and incomplete information

These features can facilitate discussions in how to implement solutions for digital twins by providing a free-to-use, existing and proven baseline.

Prerequisites for successful AAS standardisation

Most prominent amongst topics is the standardisation of an Asset Administration Shell as pushed by the Platform Industry 4.0. For this standardisation, we consider as necessary prerequisites:

- The standardization shall follow a sequential process developing consistent versions of the standard and integrating more advanced features step by step.
- The standard shall be robust towards the addition of proprietary parts without loss of generality. If the proprietary parts become proven it shall be possible to take them into the standard.
- The standard shall enable the development and application of domain specific profiles. These profiles shall be designed in a way that they reflect existing proven industry standards, can be applied in combination, and are, therefore, compatible to each other.
- There shall be an integrated mechanism ensuring, that standardized parts, domain specific profiles, and proprietary parts will be compatible with each other.
- Engineering tools and other information processing elements shall be enabled to utilise the standard and change elements within its implementation with as little effort as possible.

The goal of these prerequisites is to avoid unproductive overlaps.

This is especially relevant in the field of Industry 4.0.

Use Cases for AutomationML in the AAS context

As for the interrelation of the Asset Administration Shell and the AutomationML format and potentially other technologies like OPC UA, there is great potential for mutual enablement. The benefits for utilising AutomationML when implementing an AAS structure with submodels are from our point of view:

- Domain specific models and corresponding technologies / modelling languages are in place for different phases of a plant's lifecycle.
- Several existing tools are already equipped with AutomationML importers and exporters and are thus capable of handling submodels for an AAS without additional effort, if these are based on AutomationML. In addition, translation mechanisms to other technologies (e.g. between AutomationML and OPC UA) are in place.
- The highly modular approach of AutomationML data structures translates to native support for modular AAS implementations.
- The modularity of AutomationML also allows the easy and seamless integration of proprietary information with AAS standard information.

For implementing AAS ecosystems, we recommend the following approach:

- Engineering Phase:
  - complex information exchange between tools up to and including the construction phase
  - leading technology: AutomationML
- Operation Phase:
  - dynamic exchange of data with focus on timely delivery
  - leading technology: OPC UA
- Administration Shell as both unified access point and framework for information over the lifecycle
  - Publication/mapping of relevant basic data from AutomationML to AAS can be used to start off an AAS.
  - AutomationML models can be used as complex submodel(s) for the Engineering phase.
  - AAS, OPC UA and AutomationML should be employed in a “mix and match” between AAS core, submodels and serialisations, to allow cross-integration to whatever format fits best for a particular application case.
  - AutomationML can be employed as a native format for engineering-heavy submodels, deep crosslinking with OPC UA or other serialisations is possible and encouraged!

Thus, for the serialisation of the Asset Administration Shell, we see a synergy of existing technologies as follows. The Plattform Industrie 4.0 has developed an auxiliary format for storing AAS information, the AASX container. The OPC Foundation has devised several use case specific OPC UA models that can serve as submodels. And likewise, AutomationML e.V. has standardised a number of use case specific data models for engineering information.

Proposal

It is our conviction that the best and most sustainable approach for implementing the AAS is to allow for a hybrid mix of technology formats based on these standards. For each application scenario, and thus for each submodel inside the AAS, there exists a best of breed solution. We recommend AutomationML for engineering scenarios, OPC UA for Runtime/Online scenarios and AAS to link across those life cycle stages. If this best of breed solution is represented in the data model of any one of these data formats, this format should be utilised to store the relevant information. Storage should, on a technical level, follow the most feasible solution for the AAS execution environment in question. However, the payload of this storage should be dictated by the already established data models.

In short, when deciding on what to base data models for an AAS implementation:

- We propose to use AutomationML as data representation for engineering information sub-models of the AAS.
- We propose to use OPC UA for runtime purposes and specific machine types.
- We propose AASX as backbone data format

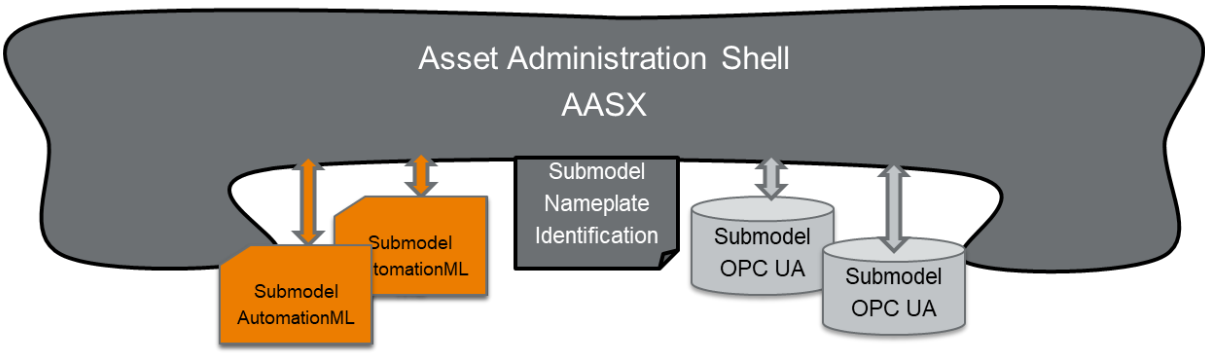

Figure 1 Composition of an AAS from a base model and submodels for engineering and operation

What should be considered before implementing the proposed approach is that potentially, for any AAS, software tools implementing interactions with the AAS would have to be capable of digesting up to three different formats of information. However, most software tools deal with specific use cases, because each use case requires a high level of expert knowledge. Furthermore, most software already uses either AutomationML or OPC UA in its data exchange handling. Thus, implementations using the AAS can build on pre-existing knowledge if the existing interfaces are respected. In addition, cross use case related information (that are relatively sparse) can be cooperatively identified by the involved users and standardisation organizations and “published” from AutomationML and/or OPC UA to AAS to simplify common use.

As a result, we recommend a use case driven technology selection to achieve:

- No confusion in formats and redundant information modelling.
- Minimized development cost.
- Reduced introduction hurdle.
- Re-use of existing know-how and existing interfaces.
- Cross use cases can be cooperatively identified and published to AAS for common use.
- Clear separation of concerns.