

Design and Deployment of Digital Twins for Programmable Logic Controllers in Existing Plants

Stephan Schäfer¹, Dirk Schöttke¹, Thomas Kämpfe¹, Kiril Ralinovski¹,
Bernd Tauber² and Ralf Lehmann²

¹Hochschule für Technik und Wirtschaft (HTW) Berlin, Berlin, Germany

²EAW Relais-technik GmbH, Berlin, Germany

Keywords: Digital Twin, Asset Administration Shell, Middleware BaSys 4.

Abstract: Digitization and network integration of production environments are two core prerequisites for changeable production environments according to a digital factory. In addition, consistent and cross-system descriptions of the equipment and abilities are required over the entire life cycle of plants. These descriptions can be done with Asset Administration Shells, which are a useful tool for the design of new facilities and their system components. Currently, industrial plants usually use programmable logic controllers, which are programmed with domain-specific programming languages. The use of these programming languages leads to an inflexible coupling of programmable logic controllers and their sensors and actuators with the process. If the plant configuration is changed later, a new engineering process with extensive reprogramming is required. However, in many cases, clarity and completeness of changes are not sufficiently addressed in the accompanying documentation. The paper discusses the results of the project "OpenBasys 4.0". A project goal is to retrofit a conventional existing plant with programmable logic controllers into a reconfigurable plant design by using the BaSys 4 middleware. A method was developed and exemplarily implemented on a prototype. The focus of the method is on the automated generation of Asset Administration Shells and their use.

1 INTRODUCTION

In the Reference Architecture Model Industrie 4.0 (RAMI4.0), the concept of the Asset Administration Shell (Bader et al., 2020) is presented as an essential basis for interoperability. The Asset Administration Shell (AAS) is the digital representation (Digital Twin) of an asset in the Industry 4.0 (I4.0) environment and enables communication with other assets. Among other things, machines, products or control systems are regarded as assets. An AAS consists of several submodels in which information and functionalities of an object are described. The information provided by AAS consists of documents, properties, parameters and other functions. (Kuhn et al., 2020)

Programmable logic controllers (PLC) are assigned to the lowest RAMI level and have a close relationship to production system equipment. (Cavalieri and Salafia, 2020) This results in significant advantages in the description of the commissioning and in the process of reconfiguration of the plant components. Despite the close association, there are gaps in the use. (Bedenbender et al., 2021)

On the one hand, it lacks consistent self-description and representation to neighboring systems and objects. On the other hand, there are gaps in the consistent preparation and use of information in the engineering phase. In addition, there is currently no established procedure for integrating existing solutions within PLCs in I4.0 system environments, but rather a variety of proprietary methods. (DIN e. V., 2020) Some gaps can be compensated for by using BMBF-funded research project Basys 4.0 middleware. (Grothoff, 2018) Fig. 1 shows how the Basys 4 middleware fits into the plant context and will be explained later for a specific scenario in the paper.

In order to reduce the deficits described above, a methodical approach is required in order to be able to generate a high benefit from the available data. For this purpose, it is first necessary to consider the general state of the art in the project planning of control environments.

In various papers, the use cases are based on the initial configuration of equipment and required structures. For example, in the modeling of a reference plant (Pick & Place - Station) and its components (Belyaev et al.,

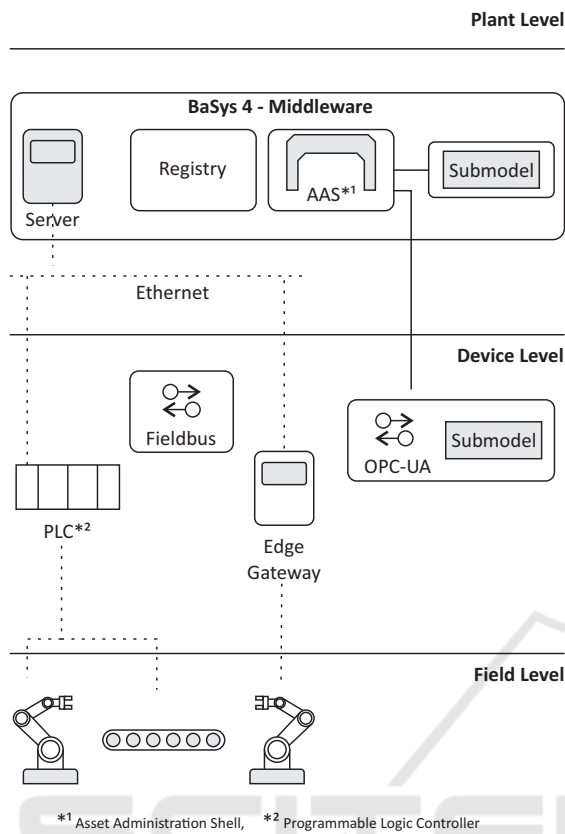


Figure 1: Integration of BaSys 4 middleware.

2021) extensive descriptions were made. The modeling of abilities as sub models was also discussed there. The migration/transition of existing plants to an I4.0 environment presents a different situation and a challenge.

Because the implementation phase of functions of a production line involves a significant engineering effort, a review of the necessary requirements and existing reference implementations should be performed in advance. It is possible that the requirements and the design of the system environment have changed during the plant life cycle, so that system components have to be adapted.

On the one hand, not all realized requirements were documented or functions were implemented or that functions have been implemented in the life cycle that were not specified in the requirements phase. This situation can lead to unpredictable side effects when redesigning the plant functionality. If, on the other hand, existing and tested function modules (templates) are already used in the modeling of the plant behavior, this results in a large number of synergies for future adaptations. This aspect should not be underestimated when transferring existing plants with programmable logic controllers to modern I4.0 structures.

2 USE CASE: SORTING

In the specific use case of high-quality thermal switches, products are tested in an industrial environment during production or after the manufacturing process. Fig. 2 shows an overview of the environment as an example. The products are placed on a special pallet carrier and made available for further processes. In this environment, each product runs through predefined test scenarios. At the test stations, the required test scope, sequence and time vary according to the product batch. The following processes are handled in the existing environment:

- Bring and Pick up service
- Pallet handling
- Pallet delivery
- Sorting of the thermal switches

The existing classic automation solution is to be gradually converted into an I4.0-based solution with Asset Administration Shell (AAS) in order to achieve synergies with the use of production data at a manageable cost.

In the use case, different abilities or groupings of abilities are required for the processes, which can be described as assets via metamodels. For example, the "pallet delivery" process includes the ability to fix and transport objects. The technological components are shown in excerpts in Fig. 3.

The abilities can be seen in isolation or as a combination of abilities. In the first case, the communication relationships and dependencies for higher-level coordination must be known. In the second case, the grouping can be based on the totality of abilities, so that the relationships can already be implemented on the programmable logic controller (PLC).

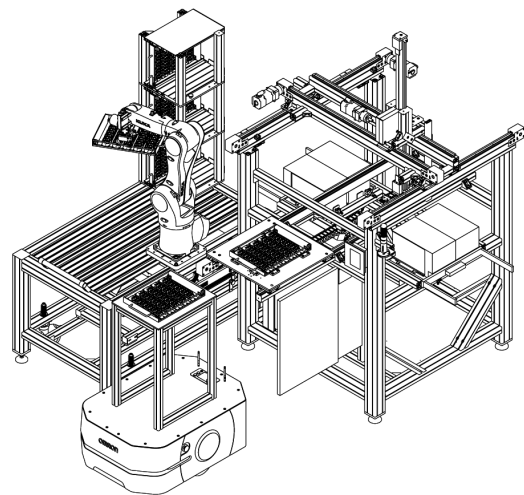


Figure 2: Technological scheme (excerpt).

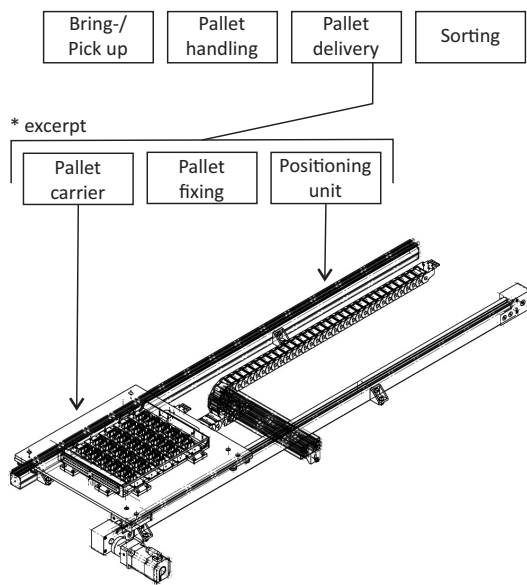


Figure 3: Subscheme - pallet delivery (excerpt).

In the use case, the functions have already been encapsulated on the PLC, so that the functions are used as assets on the AAS. All abilities or groupings of abilities are also to be synchronized at the AAS level. For this purpose, a Pack ML state machine is to be established for each controller and process or subprocess. In the case that several subprocesses are implemented on one controller, an additional higher-level Pack ML state machine is used.

3 APPROACH

3.1 Existing Plant & Asset Administration Shells

In order to successfully convert a plant into an I4.0-based solution, the plant description and the programs used must be documented in a proper manner. In the use case, the project planning of the existing plant was carried out with PLCs, which were programmed with domain-specific programming languages such as IEC 61131-3. The "Codesys" platform was used, which is popular and is based on a PLCopen compliant notation. In this platform, a large number of libraries and various communication protocols can be used. (Rayment, 2004) However, there are still no solutions that allow easy integration into an I4.0 environment via Asset Administration shells (AAS).

The challenge is the effort-reduced realization of Digital Twins (Koulamas and Kalogeras, 2018) as I4.0 components. These can be designed as Asset Adminis-

tration shells (AAS). AASs can be separated into three types based on a unified information metamodel. (Bundesministerium für Wirtschaft und Energie (BMWi), 2020) Type 1 AAS represents a passive AAS, which consists of an asset description. Type 2 AAS represents a reactive AAS, which in addition to the asset description, also contains a communication channel. Only Type 3 AAS is a proactive AAS and is capable of independently communicating with other AAS. Submodel templates are provided for modeling frequently used asset aspects. They are equivalent to the profiles for fieldbuses.

One tool for the description of AAS is the AASX Package Explorer. (Belyaev et al., 2021) This enables a structured description to be implemented using submodels and other structural elements. The AASX Package Explorer supports extensive export/import functions. The entire asset descriptions or individual submodels can be exported for further processing in an XML or JSON format, and node sets for the assets can be imported. Detailed and further information can be found on the AASX Package Explorer GitHub website. (Github Repository - AASX Package Explorer, 2021)

3.2 Challenge

The SDK BaSyx 4.2 (Eclipse BaSyx Platform, 2021) was used in the project to implement the representative. It offers the possibility of implementing AAS in different programming environments (Java, C++, C#).

AASs can be configured manually by using SDK according to the general description of the structure and its submodels in the C#. However, this means that an IT expert has to take over the engineering sector in the company. In many cases, however, appropriate employees with the required knowledge are not available in the company. However, automation engineers are already employed for the existing machinery and equipment, who are responsible for the support of the plant. Their knowledge and skills can be drawn upon with regard to the design of solutions.

The mentioned method of manual project engineering of AAS is established and will not be considered further. The question arises of how the user of industrial control is enabled to use AAS in the industrial environment. A solution is the generic preparation of AAS without knowledge of the SDK. For this, only the description of the Type 1 AAS asset and the modification of the PLC program should be as requirements.

In the research project "OpenBasys 4.0" (Bundesministerium für Bildung und Forschung (BMBF), 2019), an environment for generic generation of AAS was implemented. Type 2 AAS can be implemented in this environment without knowledge of the SDK.

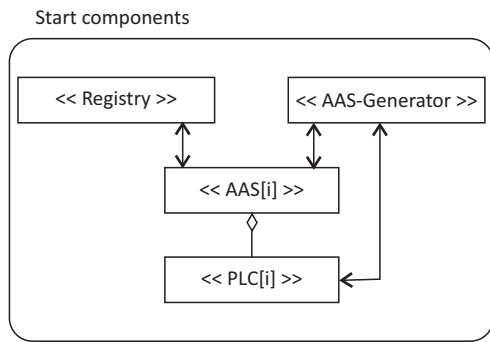


Figure 4: Minimum environment in I4.0.

The company’s employees only need to make minor corrections and additions to their PLCs. These changes include for example the global variable list (GVL) and the login data (AAS ID, ip address, etc.) as http request with a JSON structure. In most cases a http requests (GET) are supplied by libraries of PLC manufacturer. This means that a Type 2 AAS can be generated and stored in a container largely automatically on request, e.g. by the PLC. The approach involves the generic generation of AAS for a variety of use cases using established controls.

The potential user should be able to design AASs relatively easily with the help of this approach and transfer it into an I4.0 compliant environment. A machine interpretation of the contents/abilities is not yet given due to the different use of the terminology. In the project, a separate structure was created for the use case. In the future, each generated AAS will receive a GUI generated to match the AAS. At the current stage of the project, the Pack ML state machine data (Mušič, 2015) is transferred from the PLC and synchronized with the AAS. With the Pack ML state machine and the GUI, pre-testing of the components used with any end devices is possible.

4 DESIGN OF GENERATION PROCESS

A functioning I4.0 system requires at least a minimal environment (Fig. 4). In the use case, this consists of a registry and an AAS generator, which converts Type 1 AAS into Type 2 AAS. The registry is a directory where the Type 2 AAS can register with their submodels. The AAS generator has a database in which Type 1 AASs have already been stored for later use. The sequence diagram (Fig. 5) explains the generation process for an AAS and a required communication interfaces. In order to convert Type 1 AAS into Type 2 AAS, the required AAS generator was developed.

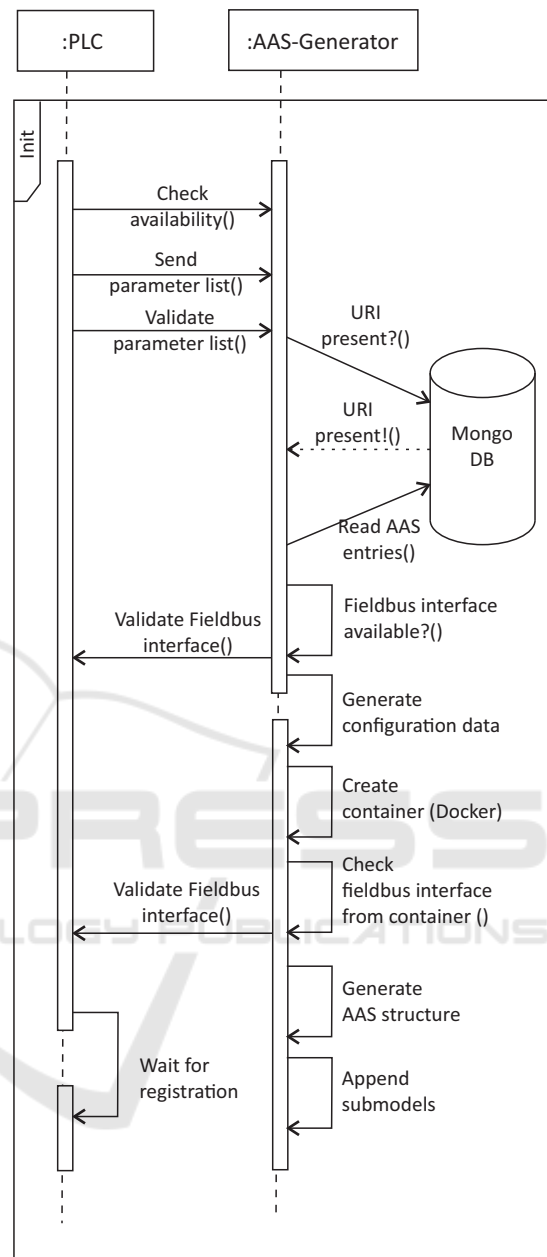


Figure 5: Sequence of generation process.

The structure of the AAS generator itself is a Type 3 AAS. PLCs use the generator’s RESTful interface. (Eclipse BaSyx Wiki, 2021) It also has knowledge of the submodels and their properties for communication.

PLCs first check the availability of the generator by sending a request. If the generator is reachable, the PLC then sends essential information for identification. Furthermore, additional information is sent to the AAS generator for later fieldbus communication between the industrial controller and the assigned AAS. These are, for example, IP address, port number and

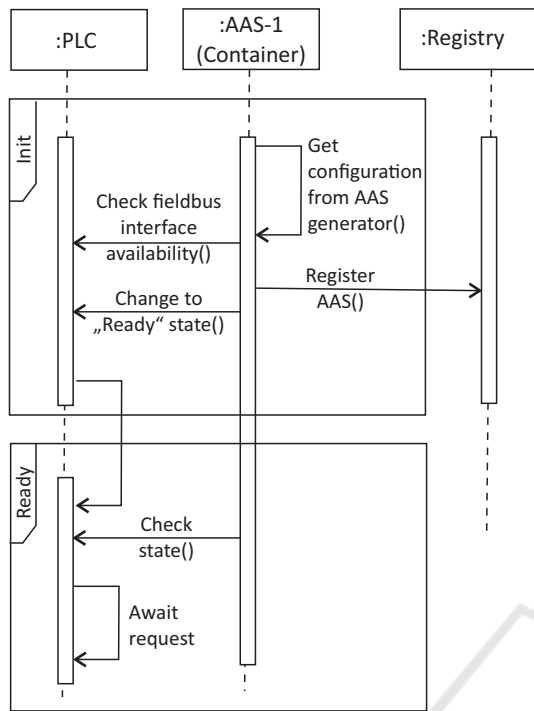


Figure 6: Sequence of registry process.

communication protocol to be used.

The AAS generator reads corresponding Type 1 AAS from the MongoDB database. If an AAS with a matching ID exists, the AAS generator reads the entries for the AAS and submodels from MongoDB and creates an instance. Now the AAS generator checks the availability of the fieldbus communication. For example, in the case of OPC UA communication, the AAS generator creates a session and establishes a connection to the PLC. The AAS generator now creates a configuration file, which contains all information of the Type 1 AAS and the submodels together with fieldbus interfaces.

The configuration file also contains the information needed for the communication between the AAS and the registry. Then the AAS generator creates a Docker container. Inside the Docker container, a Type 2 AAS is created and filled with information.

5 DEPLOYMENT OF AAS

The following sequence diagram (Fig. 6) represents the communication between AAS, registry and the respective PLC.

First, a process is started which reads the previously created configuration file. The process checks the fieldbus communication again from the Docker container. This renewed communication test is neces-

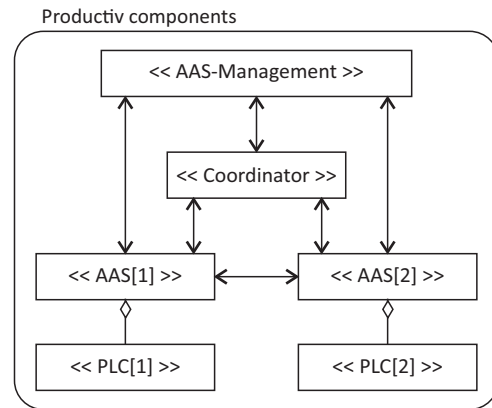


Figure 7: Productive environment (excerpt).

sary to ensure that the Docker container can establish fieldbus communication to the PLC. If the fieldbus communication was successful, the AAS logs on to the registry and transfers the descriptions to the registry. The descriptions of the AAS, its submodules and their communication endpoints are now known and the AAS is now available as an active resource.

The resources (AAS) can then be booked for use in the productive environment (Fig. 7) in accordance with the use case. In the current phase, their overall coordination is still being handled by a coordinator. The relationships between the AAS are currently still being established manually.

6 CONCLUSION AND OUTLOOK

Programmable logic controllers (PLCs) can be integrated into an I4.0 environment if they have a digital twin as a representative. The paper discussed ways of simplifying the design by a methodical approach and the use of a generic approach. This approach was implemented on a prototype and will be released as open source once the project is completed. An essential prerequisite of the approach is that the documentation is as complete and free of contradictions as possible.

Templates can now be generated with the ASS generator using the transformation models shown. Automatically generated AAS as digital twins of their PLCs enable plant operators to implement easier modifications in the plant. This helps to reduce the engineering workload.

With the embedding of the ISA-88 state machine (Pack-ML state machine) per AAS and PLC, higher-level commands can be assigned to the processes and described as an ability in the future. In addition, data from the pack tags are automatically transferred to the information model of the digital twin. This ensures

consistent and standardized process control and documentation.

With the prototype, an interface is made available via AAS, which provides further information with standardized calls (RESTful, OPC-UA).

In the future, automatically generated web interfaces can enable the pre-testing of the components. This will simplify the asset commissioning process and process coordination.

To further reduce engineering processes, automated derivation of abilities from requirements and their relationship to each other will be necessary in the future. This requires the transferability of information models at the necessary levels.

ACKNOWLEDGEMENT

This research work called "OpenBasys 4.0" has been funded by the federal ministry of education and research BMBF (01IS1900A) and is a BaSys 4 satellite project for application-oriented projects. The effort of this work was made possible by the Basys 4 framework. The authors thank for the support of the BaSys 4 team.

REFERENCES

- Bader et al., S. (2020). Details of the Asset Administration Shell : Part 1: The exchange of information between partners in the value chain of Industrie 4.0. Technical report, Berlin.
- Bedenbender et al., H. (2021). Industrie 4.0 Plug-and-Produce for Adaptable Factories: Example Use Case Definition, Models, and Implementation. Technical report, München.
- Belyaev et al., A. (2021). VWS-Referenzmodellierung : Exemplarische Modellierung einer fertigungstechnischen Anlage mit AASX Package Explorer auf Basis des VWS-Metamodells 2021. Technical report, Berlin.
- Bundesministerium für Bildung und Forschung (BMBF) (2019). BaSys 4.0 in der Anwendung - Verbundprojekt OpenBasys 4.0.
- Bundesministerium für Wirtschaft und Energie (BMWi) (2020). Verwaltungsschale in der Praxis.
- Cavalieri, S. and Salafia, M. G. (2020). Asset administration shell for plc representation based on iec 61131-3. *IEEE Access*, 8:142606–142621.
- DIN e. V. (2020). DIN and DKE Roadmap - German Standardization Roadmap Industrie 4.0, Version 4. Technical report, Berlin.
- Eclipse BaSyx Platform (2021). Ready-to-use components and extendable software development kits (SDK). <https://www.eclipse.org/basyx/>.
- Eclipse BaSyx Wiki (2021). Accessing Asset Administration Shells - AAS communication). https://wiki.eclipse.org/BaSyx/_Documentation/_AssetAdministrationShell.
- Github Repository - AASX Package Explorer (2021). A viewer / editor for the asset administration shell. <https://github.com/admin-shell-io/aasx-package-explorer>.
- Grothoff, J. A. (2018). BaSys 4.0: Metamodell der Komponenten und Ihres Aufbaus; 1st ed. Technical report, Aachen.
- Koulamas, C. and Kalogeras, A. (2018). Cyber-physical systems and digital twins in the industrial internet of things [cyber-physical systems]. *Computer*, 51(11):95–98.
- Kuhn, T., Schnicke, F., and Oliveira Antonino, P. (2020). Service-based architectures in production systems: Challenges, solutions experiences. In *2020 ITU Kaleidoscope: Industry-Driven Digital Transformation (ITU K)*, pages 1–7.
- Mušič, G. (2015). A low-cost packml-based control solution for a modular production line. *IFAC-PapersOnLine*.
- Rayment, M. (2004). Flexible motion control using iec 61131-3. In *2004 Mini Symposia UKACC Control*, pages 27–36.