A Methodology for the Design and Creation of Asset Administration Shell for Manufacturing Systems

Tasnim A. Abdel-Aty *· Simone Galparoli** · Elisa Negri* · Luca Fumagalli*

** Department of Management, Economics and Industrial Engineering, Politecnico di Milano, P.za Leonardo da Vinci, 32, 20133 – Milano – Italy (tasnim.ahmed@polimi.it, elisa.negri@polimi.it, luca1.fumagalli@polimi.it)*

***Miraitek srl, Via Colico, 36, 20158 Milano- Italy (simone.galparoli@miraitek.com)*

Abstract: Within Industry 4.0 the communication between the physical and the cyber part of manufacturing systems is in growing rise in complexity. The Asset Administration Shell (AAS) is an information framework that represents the technological features of an asset. This work addresses the design of AAS by proposing a methodology to guide practitioners through the process of creating AAS models for manufacturing systems, and populating them with real-time data from the field. The aim of the paper is to design a methodology for the creation of AAS that is user friendly and functional to be followed by non-IT experts. The proposed methodology has been applied and validated within the Industry 4.0 Lab of the School of Management of Politecnico Di Milano.

Keywords: Asset Administration Shell; AAS; data model; Industry 4.0; Manufacturing; information model

I. Introduction

Technology has always been a driving force for industrial evolution. The adoption of Information and Communication Technologies (ICT) into manufacturing allows the integration of industrial operational technologies with information technologies (IT) (Inigo *et al.*, 2020), hence driving mass customization of manufacturing for customer-driven products. For the purpose of attaining various business objectives, smart manufacturing networks combine heterogeneous data gathered across the value chain. Among other things, the Industry 4.0 vision encompasses this trend by developing intelligent goods and processes (Lelli, 2019). To facilitate this vision, models, architectures, and smart technologies need to enable horizontal and vertical communication and interoperability (Inigo *et al.*, 2020; Lelli, 2019). However, the implementation within industry is not easy as industrial assets are characterized by high specificity, low technological flexibility, and information silos. For describing an Industry 4.0 asset there exist multiple initiatives that try to use semantic annotation for driving device interoperability. Lelli (2019) has studied how industry professionals and academics address these initiatives and concluded that they are 'fragmented' and do not deliver a standardized understanding. The current way manufacturing plants facilitate operations causes information silos and a lack of standardization in the presentation of process, product, and service data. However, this needs to change as customers are increasingly demanding improved flexibility, transparency within the manufacturing process, and adaptability (Ye and Hong, 2019). To address this problem, intelligent manufacturing reference models exist, such as the Reference Architectural Model for Industry 4.0 (RAMI 4.0) (Lins and Oliveira, 2020), and the Industrial Internet Reference Architecture (IIRA) (Monteiro *et al.*, 2018; Pivoto *et al.*, 2021). RAMI 4.0 is one of the architecture models that aim at aligning a holistic consolidation of business information, automation, and manufacturing execution functions, and provides the description for Industry 4.0 with uniform structure and uniform wording (Hosseini, Sauter and Kastner, 2021). RAMI 4.0 is a three-dimensional map that depicts the key elements of Industry 4.0 and provides a specification for the industrial component of Industry 4.0 as being composed of the physical asset and its digital part named Asset Administration Shell (AAS) (Wei, Sun and Liu, 2019). The AAS, introduced by the German Platform for Industrie 4.0 (https://www.plattform-i40.de), has a key role of the exchange of asset-related data among assets, along the supply chain, or between production orchestration systems and engineering tools (Bader *et al.*, 2020). This logical representation of the asset aids in satisfying the guidelines of Industry 4.0 by allowing the asset to communicate its real-time parameters, production data, and state of the asset in a standard form (Wein *et al.*, 2020). The way the data is structured within the AAS leads a path towards standardization within industrial systems and hence could be used as data model for Digital Twin (DT). For example, Platenius-Mohr et al. (2019) have demonstrated how the AAS could enable interoperable DT. The relationship between the AAS and DT is complex and has taken many different forms within literature. This work focuses on RAMI 4.0 since contributions for extending and complementing the architecture are important to provide practitioners with a completely integrated reference model to support the implementation of digital and smart manufacturing. In order to encourage the adoption of AAS, as part of RAMI 4.0, in manufacturing companies, this paper presents a methodology for industrial professionals to create and design AAS models with the aid of easy-to-use tools.

The article is structured as follows. In Section II is the research background about the AAS that sets the base for this work. Section III details the proposed methodology of the AAS creation. Section IV is a presentation of the application of the proposed methodology within the Industry 4.0 Lab at the School of Management of Politecnico Di Milano, and the tools that have been used to allow the methodology to be user friendly for non-IT experts. Section V follows the discussed contribution of this work within an industrial and research environment, as well as discussing the limitations of the proposed methodology. Section VI concludes by summarizing this work and discussing possible future developments.

II. RESEARCH BACKGROUND: ASSET ADMINISTRATION SHELL

The AAS is defined as "A structured digital representation of an asset that serves as the foundation for interoperability between applications that manage manufacturing systems. AAS defines the Administration Shell and the assets it represents, stores digital models of various aspects (submodels), and explains the technological features exposed by the Administration Shell or the assets in question" (Bader *et al.*, 2020). Accordingly, AAS can be grouped into three types (Hosseini, Sauter and Kastner, 2021):

- Passive AAS are serialized files, like JSON or XML. These models contain serialization of static information describing an asset.
- Reactive AAS are hosted on servers. They may demonstrate both static data and real-time exchange of information by interacting with other components. Accordingly, they are runtime instances that provide http/REST or OPC-UA interface for communication.
- Proactive AAS are extensions of reactive AAS, that initiate and implement an active behaviour, which means they could communicate and negotiate on their own.

A metamodel of the AAS has been introduced by the Industrie 4.0 Platform containing a 'header' with the Asset and AAS information, and a 'body' with Submodels and Submodel Elements (Bader *et al.*, 2020), as can be seen in Fig. 1.

Within literature the AAS is considered an emerging topic that has been receiving a rising rate of attention. It has been studied as a tool for service-oriented Industry 4.0 Architecture where Kuhn, Schnicke and Oliveira Antonino (2020) explain and illustrate their experience of transferring service-oriented architecture principles to the automation industry using AAS for the realization

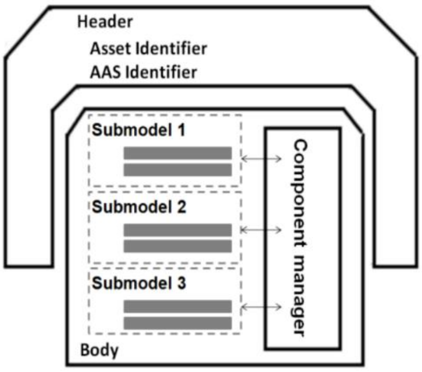

Fig. 1. Metamodel Structure from (Hang et al., 2022)

of DT, they found that this integration leads to benefits like end-to-end communication and unified interfaces for applications and dashboards.

The use of AAS for interoperability within Industry 4.0 has been analyzed by Platenius-Mohr *et al.* (2020) and Cavalieri and Salafia (2020). Platenius-Mohr *et al.* (2020) present a solution that flexibly transforms the information models for DT. The solution also allows for file and Application Programming Interface (API)-based bidirectional exchange of information. They applied their solution to a real-world case by transforming ABB AbilityTM DT to the AAS format. They demonstrated that their approach enables connections between industries using smart use cases. Cavalieri and Salafia (2020) approach interoperability by introducing a definition of a generic and technology-independent model for predictive maintenance using AAS. The standardization of the AAS model itself has been explored by Lüder *et al.* (2020) with the aim at facilitating the collection of engineering data sets for Industry 4.0 applications and their export into the AAS serialization using AutomationML based domain-specific languages. Wein *et al.* (2020) have also addressed AAS standardization by presenting a framework that is lightweight for executing active AAS based on inversion of control.

There are current tools available for the creation and use of AAS, each with different features. For example, regarding the creation of AAS package files: the AASX Package Explorer (https://github.com/admin-shell-io/aasx-package-explorer) is an editor and a viewer with a user friendly interface, and PyI40AAS (https://git.rwth-aachen.de/acplt/pyi40aas) is a python module introduced by a research group in RWTH Aachen. For the implementation of the AAS and the connection with the physical asset: NOVA Asset Administration Shell (NOVAAS) (https://gitlab.com/novaas) is an open source environment that uses Node-Red, and BaSyx (https://wiki.eclipse.org/BaSyx) is an open-source platform that supports implementation of Industry 4.0. Most of the tools present require high IT expertise or are inefficient for the creation of complex models. Additionally, no work has been done in the literature addressing the design and creation of the AAS. Specifically for industrial implementation, where assets need to be modeled with a high level of detail and complexity. Therefore, this paper aims to fill this gap by proposing a methodology for the design and creation of the AAS and demonstrating the tools used that facilitate the creation of AAS for non-IT experts.

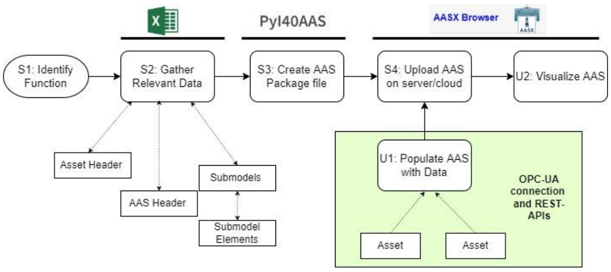

Fig. 2. Proposed AAS Methodology

III. Proposed AAS Methodology

The methodology proposed, displayed in Fig. 2, contains four steps (S1, S2, S3, S4) that guide the user in the design process of the AAS. The methodology follows the structure of the metamodel proposed by Industrie 4.0 Platform, previously described as containing AAS, Asset, and submodel related information. U1 and U2 represent two utilization possibilities of the created AAS. A relationship exists between these two choices of the utilization possibilities and the AAS design process, specifically for the choice of the submodels. A submodel framework has been proposed, in S2, to highlight this connection and facilitate the structured choice of the submodels.

Fig. 2 also presents the tools that the authors deem most appropriate as they are easy to use, particularly by non-IT experts. They have been used in every step for the application case within Industry 4.0 Lab of Politecnico di Milano and are explained in further details in section IV. A detailed description of each step and utilization possibilities is provided below.

A. *S1: Identifying the function*

Identifying the function, or the intended use, of the AAS is the first requirement in the methodology. Depending on the function identified the design and choice of the information model would be based. The AAS could represent, but not limited to, the following:

- A resource/asset in an assembly line for operational information;
- A product for the product lifecycle information;
- A process for Enterprise Resource Planning (ERP) / Manufacturing Execution System (MES) data.

B. *S2: Gathering relevant data*

The data that need to be collected for AAS will be presented as header data and body data. The header needs to contain a minimum level of mandatory information defined for the asset and the AAS (Bader *et al.*, 2020). These include:

- Unique Identifier (idType and id): Identifiers are important elements within the Industry 4.0 environment, hence every AAS and asset need to have unique identity information that can be used to uniquely reference entities, and used to relate elements to external definitions (Bader *et al.*, 2020). Id types can be IRDI (https://www.iso.org/standard/50773.html) , IRI (https://tools.ietf.org/html/rfc3987), or Custom.
- Name Identifier (idShort): A string identifier for each element describing the asset/AAS.
- Asset Reference (assetRef): Each AAS needs to reference the asset it encompasses by including the
  - reference id of the asset.
  - type of this reference (IRI, Custom, etc.), and the
  - Kind: Specifies whether the asset is of kind "Instance" or "Type".

The body data are made up of submodels containing submodel-elements. Submodels make up a large part of the AAS body and hence need to be diligently crafted. The submodel-elements contained within the submodels could be a relationship, operation, capability, entity, event, or data element. The AAS metamodel, which has been introduced in background section, does not dictate mandatory submodels. However, various efforts exist for the standardization of submodel templates. For example, the Industrie 4.0 Platform has introduced a submodel template for Contact Information, Technical Data, and Digital Nameplate (Platform Industrie 4.0, 2021). Ye et al. (2020) proposed the identification of the submodels according to the type of data they enclose, hence introducing the division of 'Generic submodel' and 'Asset-Specific Submodel'. The authors propose an expansion on this classification according to whether the submodel contains variable submodel-elements that receive real-time data from the asset. This classification aids in the design process of the AAS information model to support the vision of standardised AAS. The submodels could be defined as, Fig. 3:

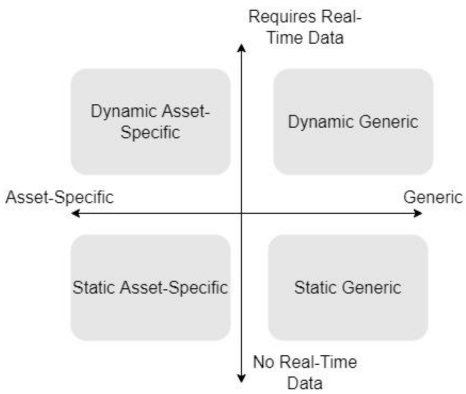

Fig. 3. Submodel Framework

- Dynamic Asset-Specific Submodel: These submodels contain data that receive real-time updates of the data from the asset and are particular for each individual asset. This submodel could serve for monitoring various operational activities like the health status of the asset, or the sensors data. Dynamic Asset-Specific submodels open pathways for AAS to be a vital tool for interoperability and virtual monitoring of the asset.

- Dynamic Generic Submodel: Within this submodel the data exchange might not be at the same rate as the Dynamic Asset-Specific submodel. The main difference is that this submodel contains data that are generic and could be gathered from all assets. An example of this submodel could be a 'Security' submodel that details the access rights and will need real-time data to update the access log.
- Static Asset-Specific Submodel: Asset specific submodels contain data that are particular to the asset and do not benefit from real-time data exchange. An example could be a submodel that contains specific information about the capabilities of the asset.
- Static Generic Submodel: This submodel contains data that do not benefit from real-time data exchange and could be gathered from all assets. An example could be a Documentation submodel containing files from the supplier of the asset.

The choice of the submodels along the y-axis (i.e the real-time data requirement) is related to the intended use of the AAS. If the AAS will be used for real-time data updates (U1), it would be expected that either Dynamic Asset-Specific or Dynamic Generic submodels are present. Otherwise, the presence of Dynamic submodels would not be required.

C. *S3: Create AAS Package*

The creation of the AAS package, in *.aasx* format, allows the user to have the information in a format that is light, interoperable, and easily accessible by various stakeholders in the value-chain. Various methodologies and tools exist to create the AAS package format, which have been discussed in the research background section. For this work the python library PyI40AAS was used for its flexibility and customization characteristics.

D. *S4: Upload AAS on Server/Cloud*

Uploading the AAS packages on a server or a cloud makes it possible to extend the usability of the models by allowing the connection with real-time data (U1: Data population) from the asset, or to visualize the AAS models (U2: Visualization) for activities that require a standardized and readable view of the asset data. Both of the uses have been demonstrated and further explained in the next section.

IV. APPLICATION

At the School of Management of Politecnico di Milano, a proof of concept for the proposed methodology was developed in the Industry 4.0 Laboratory (I4.0 Lab). The laboratory is fitted with a didactic assembly line that creates a streamlined mobile phone. Each module of the production line is equipped with a collection of sensors that can be accessed and read in real-time via OPC-UA protocol (Fumagalli *et al.*, 2016). The line consists of one conveyor belt that connects seven workstations, each of which is dedicated to one or more steps in the cell phone assembly process. Within this section is a detailed description of how each step from the methodology has been applied and the tools used. These tools simplified the creation process of the AAS and allowed it to be user-friendly.

A. *S1: Identifying the function*

The AAS developed were for the resources of the I4.0 Lab assembly line. The function of the AAS created is to be an information model of the I4.0 Lab assets that can be used to collect non-homogenous asset data and be used to support the data flow for DT. The functional and operational information of assets were encompassed within each AAS. The AAS have been designed to follow the hierarchy in Fig. 4 that is composed of 13 resource AAS models.

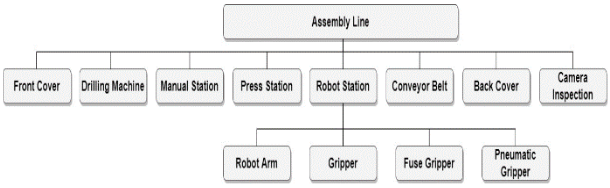

Fig. 4. I4.0 Lab AAS Hierarchy

B. *S2: Gathering relevant data*

All the data gathered have been stored in a Microsoft Excel (MS Excel) file that structures the information model by connecting between the AAS, the asset, the submodels, and their submodel-elements using the unique identifiers as shown in Fig. 5. MS Excel has been chosen for the data gathering step, as it is a relatively known tool and does not require high computational expertise to operate. This connection is crucial as it sets the base for S3, and accordingly demonstrates the ease-of-use of the proposed methodology.

The information gathered includes the sensors connected to every asset and the kind of data that the OPC-UA reads, which helped the mapping of the submodels needed for every asset. In Table 1 is a list of all submodels with their type and the assets they represent. Submodels with their relative meaning, are the following:

- "Technical Documentation" contains all the files provided by the manufacturer for the asset.
- "Communication" includes the OPC-UA and theWeb-Interface addresses.
- "Technical Data" contains the technical data about the asset provided by the manufacturer.

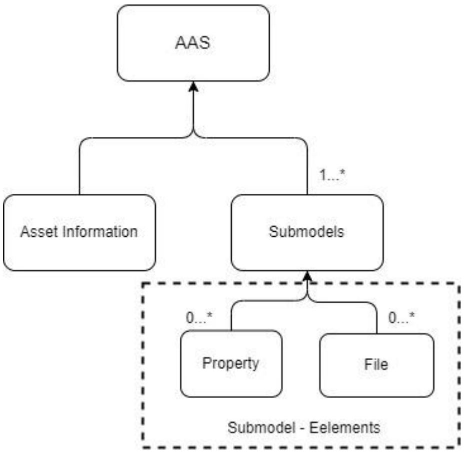

Fig. 5. Unique identifiers used in MS Excel to structure AAS

- "Energy Management" monitors the properties related to the energy efficiency of the asset, like active power, energy saving mode etc.
- "Operational" monitors the operational properties of the asset including the presence of errors.
- "Predictive Maintenance" is related to the health state of the asset and the scheduled maintenance.
- "Part Data" monitors the properties related to the part and its movement along the assembly line.
- "Bill of Materials" is a special submodel designed for complex assets to include the unique identifiers of other AAS that make up the original asset. This submodel is responsible for creating a hierarchical structure of the assembly line.

TABLE 1
SUBMODEL LIST FOR I4.0 LAB

Submodel Name	**Submodel Type**	**Assets**
Technical Documentation	Static Generic	All
Communication	Static Generic	All
Technical Data	Static Generic	All
Energy Management	Dynamic Generic	Drilling, Camera Inspection, Front Cover, Manual Station, Press Station,Back Cover
Operational	Dynamic Generic	All
Predictive Maintenance	Dynamic Generic	All
Part Data	Dynamic Asset-Specific	Drilling, Camera Inspection, Front Cover, Manual Station, Back Cover
Bill of Material	Special Submodel	Assembly Line, Robot Station

C. S3: Create AAS Package

Using the PyI40AAS library, a python script has been coded to read the MS Excel file, created in S2, and to automatically extract the data needed from it to create an AAS serialization as a .aasx package file, without requiring manual coding interference from the user. A graphical user interface (GUI) has been added to present the script as a one-click tool, as seen in Fig. 6. This allows any user with little to no coding experience to create an AAS within a few minutes, being of great benefit to any practitioner in industry. The script has been used to create AAS files following the hierarchy seen in Fig. 4 of the I4.0 Lab assembly line.

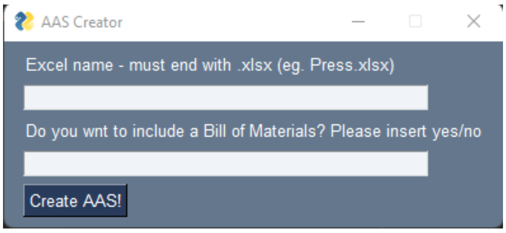

Fig. 6. One-Click Tool for AAS creation

D. S4: Upload AAS on Server/Cloud

The AAS models created in S4 have been uploaded on a server using the open source AASX Server (which can be openly found at: https://github.com/admin-shell-io/aasx-server). The server uses Blazor framework to provide a graphical user interface in the browser to allow the visualization and exploration of the AAS (U2). Real-time data update in the AAS was also made possible using REST APIs (U1); a python script was coded to connect to the desired sensors in the asset using their specific OPC-UA address and determine the rate at which it will read the data. After connecting to the sensors, it proceeds to create the necessary connections to the AAS using the native API provided by the AAS Server and starts periodically updating data in AAS, in a synchronous way, carrying out the following steps:

1. Download the submodel through a 'get' call to the API of the server.
2. Save the submodel as a JSON file.
3. Read the data from OPC-UA using the sensor address specified and stores the corresponding value.
4. Update the downloaded submodel with the new OPC-UA data.
5. Upload the updated submodel into the server, replacing the old submodel, with a 'put' call to the specific REST API.

This loop continues to iterate and to update the submodel as long as there is flow of data present, hence allowing the real-time time data population into the AAS model.

V. Discussion

The proposed methodology offers the prospects to create AAS models without the use of high IT competences. Three points of actions have been focused on to simplify the methodology:

1. The AAS design process: By creating a streamlined methodology with action steps (S1, S2, S3, S4) and utilization options (U1 and U2);
2. The Submodel choice process: Expanding the submodel definition proposed by Ye et al. (2020) to include the real-time data property axis to the submodel choice framework;
3. The AAS creation tools: The creation of a new tool is out of the scope of this paper. However, already existing tools (i.e MS Excel and PyI40AAS library) have been combined to demonstrate the potential of industrial implementation of AAS without the need for high IT expertise.

The methodology has been designed with the purpose of reducing the computational competences needed from stakeholders responsible for creating new AAS, or updating already existing AAS. This expands the discussion regarding the usability of AAS in relationship to the intended function and personnel within the manufacturing company and along the value chain.

Therefore, the validation has been applied to validate whether the methodology guides the user in the creation of a successful AAS, and whether the methodology along with the accompanying tools satisfies the requirement of being easy to use. Students within the I4.0 Lab, with no previous IT competences, have been asked to create AAS models that serve different functions, as a first demonstration for the ease in the AAS creation through the proposed methodology. First the students tried understanding and creating the models without the introduction of the methodology, which proved to be time consuming and a challenging task without the coding knowledge needed to operate the already existing tools. The introduction of the methodology and the accompanying one-click python code allowed the students to shift their focus from trying to learn and understand the tools, towards the data to be gathered and detailing the structure of the AAS to be function specific.

The methodology has been used by the students to create AAS for varying complexity of assets. The use of MS Excel for the data gathering allows the methodology to be scalable since no change to the python code is needed. However, the scalability of the methodology needs to be further tested in an industrial context, specifically the behavior of the tools with large amounts of data. Another limitation would be that the methodology does not address the full range of utilization options possible for the AAS.

VI. Conclusion

As the AAS is a relatively new subject, it has not yet received attention for adoption from manufacturing companies. There is a lack of architectures and methodologies that support its implementation. As well as tools that cater to industrial practitioners with varying competences. This work fills this gap and contributes to the literature by proposing a methodology for the creation of AAS and demonstrating the tools that could be used to allow the methodology to be user friendly and suitable for industrial applications. The ease in the use of the methodology has been validated in a proof-of-concept laboratory validation, that demonstrated the relevance of the proposed work also for industrial practitioners who want to implement Industry 4.0 RAMI platform, through AAS.

Future work could focus on expanding the methodology to involve more utilization options in relation to the environment surrounding the AAS from the point of view of stakeholders involved along the value chain. As well as an analysis on the key performance indicators and methodologies for analyzing the return on investment (ROI) needed for measuring the impact of the AAS on manufacturing companies.

Acknowledgement

This project has received funding from CanvAAS: Connected Assets iNteroperability framework Via AAS, EIT Manufacturing project. Activity Code: 20049 (https://eitmanufacturing.eu/connected-assets-interoperability-framework-via-aas/). This research work has been possible thanks to the use of the Industry 4.0 Laboratory of the School of Management of the Politecnico di Milano (www.industry40lab.org).

References

Bader, S. et al. (2020) 'Details of the Asset Administration Shell Part 1 - The exchange of information between partners in the value chain of Industrie 4.0', Plattform Industrie 4.0, 0, p. 473. Available at: https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/Details-of-the-Asset-Administration-Shell-Part1.html.

Cavalieri, S. and Salafia, M. G. (2020) 'A model for predictive maintenance based on asset administration shell', Sensors (Switzerland), 20(21), pp. 1–20.

Fumagalli, L. et al. (2016) 'New methodology for smart manufacturing research and education: The lab approach', in Proceedings of the Summer School Francesco Turco. AIDI - Italian Association of Industrial Operations Professors, pp. 42–47.

Hang, J.H., Charles, D.S., Gan, Z.H., Gan, S.K., Lim, Y.M., Lee, W.P., Wong, T.L. & Goh, C.P. (2022). Constructing a Real-Time Value-Chain Integration Architecture for Mass Individualized Juice Production. Information. [Online]. 13 (2). p.p. 56.

Hosseini, A. M., Sauter, T. and Kastner, W. (2021) 'Towards Adding Safety and Security Properties to the Industry 4.0 Asset Administration Shell', IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS, 2021-June, pp. 41–44.

Inigo, M. A. et al. (2020) 'Towards an Asset Administration Shell scenario: A use case for interoperability and standardization in Industry 4.0', Proceedings of IEEE/IFIP Network Operations and Management Symposium 2020: Management in the Age of Softwarization and Artificial Intelligence, NOMS 2020.

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 2020. Institute of Electrical and Electronics Engineers Inc.

Lelli, F. (2019). Interoperability of the Time of Industry 4.0 and the Internet of Things. Future Internet, 11(2), p.36.

Lins, T. and Oliveira, R. A. R. (2020) 'Cyber-physical production systems retrofitting in context of industry 4.0', Computers and Industrial Engineering, 139.

Lüder, A. et al. (2020) 'Generating Industry 4.0 Asset Administration Shells with Data from Engineering Data Logistics', IEEE Symposium on Emerging Technologies and Factory Automation, ETFA, 2020-Septe, pp. 867–874

Monteiro, P. et al. (2018) 'Adoption of Architecture Reference Models for Industrial Information Management Systems', in 9th International Conference on Intelligent Systems 2018: Theory, Research and Innovation in Applications, IS 2018 - Proceedings. Institute of Electrical and Electronics Engineers Inc., pp. 763–770.

Pivoto, D. G. S. et al. (2021) 'Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review', Journal of Manufacturing Systems, 58(PA), pp. 176–192

Platenius-Mohr, M. et al. (2020) 'File- and API-based interoperability of digital twins by model transformation: An IIoT case study using asset administration shell', Future Generation Computer Systems, 113, pp. 94–105.

Platenius-Mohr, M., Malakuti, S., Grüner, S., Goldschmidt, T., 2019. Interoperable digital twins in IIoT systems by transformation of information models: A case study with asset administration shell, in: ACM International Conference Proceeding Series. Association for Computing Machinery.

Platform Industrie 4.0 (2021) *Submodel Templates for Asset Administration Shell*. Available at: www.plattform-i40.de

Wei, K., Sun, J. Z. and Liu, R. J. (2019) 'A Review of Asset Administration Shell', IEEE International Conference on Industrial Engineering and Engineering Management, pp. 1460–1465.

Wein, S. et al. (2020) 'Flexible and Lightweight Framework for Active Industrial Asset Administration Shells', in Proceedings - 2020 IEEE Conference on Industrial Cyberphysical Systems, ICPS 2020. Institute of Electrical and Electronics Engineers Inc., pp. 549–554.

Ye, X. and Hong, S.H. (2019). Toward Industry 4.0 Components: Insights Into and Implementation of Asset Administration Shells. IEEE Industrial Electronics Magazine, 13(1), pp.13–25.

Ye, X. et al. (2020) 'Toward the Plug-and-Produce Capability for Industry 4.0: An Asset Administration Shell Approach', IEEE Industrial Electronics Magazine, 14(4), pp. 146–157.