2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA) | 978-1-6654-9996-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ETFA52439.2022.9921637

Automated generation of Asset Administration Shell: a transfer learning approach with neural language model and semantic fingerprints

Yuchen Xia
Institute of Industrial Automation and Software Engineering, GSaME
University of Stuttgart
Stuttgart, Germany
yuchen.xia@ias.uni-stuttgart.de

Nasser Jazdi
Institute of Industrial Automation and Software Engineering
University of Stuttgart
Stuttgart, Germany
nasser.jazdi@ias.uni-stuttgart.de

Michael Weyrich
Institute of Industrial Automation and Software Engineering
University of Stuttgart
Stuttgart, Germany
michael.weyrich@ias.uni-stuttgart.de

***Abstract*— The Asset Administration Shell (AAS) is a standardized data container for sharing data in the context of Industry 4.0. It allows different participants or systems to communicate their information based on shared meaning. Today, however, AAS makes data interoperable at the cost of extra development effort. Developers must map a proprietary information model to a standardized AAS model during the data transformation. In this work-in-process paper, a novel data transformation method based on transfer learning with neural language model is proposed to automatically map the data properties from an arbitrary information model into a standardized AAS model. The term "semantic fingerprint" is used to characterize a pivot intermediate vector generated by a neural network, containing latent conceptual meaning about a data property, which is in turn used for generating the mappings between data properties with similar conceptual meaning. By this means, the proposed approach fills the research gap on automated generation of AAS models with semantic analysis and is able to lower the barrier to adopting AAS with tool support.**

Keywords*— *Semantic fingerprint, Asset Administration Shell, neural language model, data transformation, Industry 4.0

I. Introduction and Background

Modern industrial manufacturing has been benefiting from the trends of automation and digitalization. Digitalization is deemed a strategical measure for manufacturing companies to improve process efficiency, decrease communication cost, cope with domain complexity, and adapt to market fluctuation. Whether to conduct these measures is an investment decision, and digitalization is expected to bring a more sustainable payoff than it would cost. The digital transformation often implies that the data need to be exchanged between different domain applications, and engineering effort must be made to ensure that the meaning of the data is correctly understood. This process is highly dependent on the available domain knowledge possessed by a development team, and the raw data from a manufacturing company may not be originally designed to be scalable as well. Thus, making systems interoperable can be time-consuming if the expert knowledge is absent, resulting in escalated costs in mapping different interfaces and disambiguating the transmitted information. Without understandable semantics of the data, the integration of a system node into a global network would be more expensive, and the long-term payoff would be reduced due to the diminished reusability and manageability. In other words, the lack of semantic interoperability reduces the return on investment and impedes the industrial digitalization process, especially when interconnecting heterogeneous systems and sharing data across different domains.

A. What is Asset Administration Shell (AAS)?

The asset administration shell is the central corner stone of Platform Industry 4.0 to enable interoperability in the industry and is now under iterative development. [1]

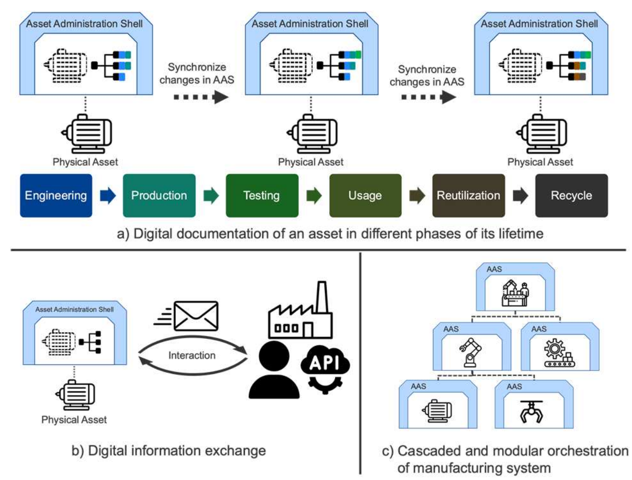

Fig. 1. Prominent features of AAS summarized from [1], [2]

Fig. 1 introduces some prominent features of AAS derived from the AAS specification [1], [2]: it can be used as a data container, which is a digital duplicate of a physical asset to exchange the product information, containing standardized templates (sub-models) of various aspects to describe a physical or logical object; the AAS can also be considered as a digital file that accompanies the product's lifetime to allow interoperable access by different domain applications; last but not least, the AAS can also be deployed in a cascaded and modular way to describe the manufacturing system and to provide scalable information access.

B. What are the typical use cases of AAS so far?

The AAS provides a solution to implement modular and decentralized digital twin. The use cases of AAS can mostly be clustered into two groups:

1) AAS shares the product information as a data container, facilitating the information exchange across the companies in supply-chain [1], especially within a long supply chain with multiple participants. Typical use cases are, e.g., digital product passport [3], AI data management in manufacturing lifecycle [4], carbon footprint [5] and so on.

2) AAS can describes the components and systems in a production facility with a standardized interface [2], facilitating the information exchange in the manufacturing company and providing a communication interface, especially where the production includes diverse machines, different materials, complex processes and heterogeneous data. Typical use cases are, e.g., automatic device configuration [6], agent-based production systems [7], predictive maintenance [8]. In these use cases, the AAS is seen as a standardized data container and interface complemented with other communication technologies, typically with OPC UA [9] and underlying communication stacks.

II. Existing Challenges

A. Which challenges could inhibit the potential of AAS?

C 1: Typically, the creation of AAS demands considerable effort. The engineer must read and understand the specifications of AAS and collect the required data accordingly. The AAS provides standardized sub-models [10], each of which contains a set of recommended data properties depicting an aspect of a use case. During the data transformation. The concept of the data property must be interpreted and disambiguated properly. Lack of domain knowledge in manufacturing as well as the absence of competence in IT and automation could make deployment of AAS hardly possible.

C 2: Expert consensus for data definition needs to be reached. Users from different domains have different understanding of data properties. Even data properties with the same name can have different semantics in different contexts, for example, considering a data property called *"current_measurement"* for monitoring electrical power supply and contrastively for recording production capacity history; data properties with different names can also have the same semantics, e.g., "rotation rate" and "motor speed". The identifiable semantics need to be well specified during the information exchange between different domains.

C 3: Diverse standardized sub-models in AAS are still being developed [10] and they may not be optimally defined for all specific use cases at the moment. Effort needs to be made to ensure that the standardized data properties should be neither too specific nor too general, or neither redundant nor missing, depending on the concrete use cases. Different participators in the industry should be encouraged to collaborate on developing best practice standards.

C 4: The data transformation is generally still laborious. Today, to create AAS for an asset, the developers need either to manually enter the required data with a graphical editor *AAXS-Package-Explorer* for a small number of assets, or to specifically define the model transformation rules to create the ad-hoc data mappings between the proprietary data model and AAS model, in order to batch-process all the AASs for assets of the same type, as in [11]–[13].

C 5: The commercial value of AAS needs to be realized and proven. It is believed that AAS has the potential to enable a seamless information exchange in the global supply chain [1]. However, the number of AAS users needs to reach a critical point to achieve the network effect. It is urgent to reduce the cost to deploy AAS and to convince the potential users to try out the AAS. Theoretically, the value of AAS increases with the growing number of users.

Other issues are also being handled by the Platform Industry 4.0, e.g., the data security [14] and trust in industrial data-sharing [5].

B. Contribution of this paper, and how the challenges can be addressed

This paper proposes a novel approach to automate the creation of asset administration shell, as matching the semantics of data properties demands considerable effort in manual searching, disambiguating and mapping the data. The term "semantic fingerprint" is brought up in this paper to characterize an auxiliary indexing vector generated from a neural network. The semantic fingerprint encodes latent semantic information about a data property to enable the machine to process this information and to map the data properties into AAS. The proposed approach contributes to solving or alleviating the 5 aforementioned challenges.

Inspired by the quotation "what we can say at all can be said clearly" from philosopher Ludwig Wittgenstein, the concept of a data property can be specified clearly, and thus the information exchangeability can be improved by providing the transmitted data with a machine-readable meaning. The work described in this paper is proceeding with the vision of making machine able to figure out this clear meaning autonomously and fill the research gap on automated generation of AAS model with semantic analysis in the context of Industy 4.0.

III. Conceptual Design

In AAS, a property element can be referenced to a data definition from an extern dictionary that specifies the concept of the data, e.g., from the *ECLASS-dictionary* or *IEC CDD*. This allows data property to be specified with a complementary standardized concept description in text that specifies the meaning of the data property.

A neural network can be trained to process these texts and generate a semantic fingerprint, capturing the latent semantic meta-information about a data property. Today, some state-of-the-art neural language models from natural language processing can outperform humans in semantic understanding tasks [15]. It is supposable that the neural language model should also be capable to fulfill the semantic mapping task and at least partially take over the manual effort on making heterogeneous systems interoperable.

A. Transfer learning paradigm and neural language model

For training the neural network, the requirement to collect abundant comprehensive data from industry companies cannot be easily achieved; nevertheless, benefiting from the paradigm of transfer learning, pre-trained neural language models released from the natural language processing community are available to embed the sentences into vectors with up-to-date performance [15]. These models are also referred to as “foundation models” [16] that can learn general semantics in natural language from abundant text data collected from the web, and they can be transferred and used for the downstream applications with or without fine-tuning.

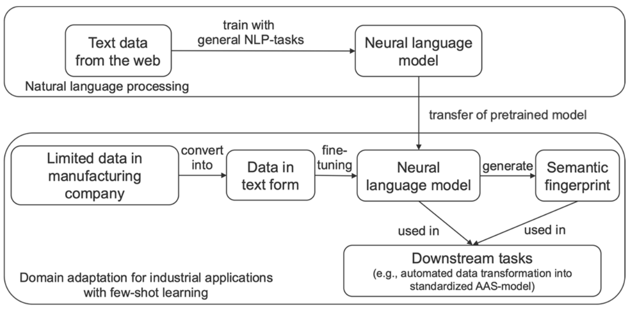

Fig. 2. Applying the neural language model and the transfer leaning paradigm

As shown in Fig. 2, the neural model taken over can be trained afterward. Complaint with the transfer learning paradigm, the model can be fine-tuned with limited training examples for a specific domain and application, also referred to as few-shot learning [17]. The fine-tuning aims to improve the performance of a general neural language model in extracting domain specific semantics.

B. The generation and usage of semantic fingerprints

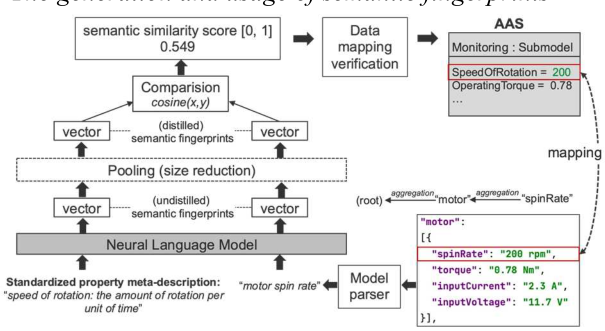

Fig. 3. The system design to map data properties with semantic fingerprints

Fig. 3 presents the overall architecture of the proposed approach to map a data property from arbitrary information model into a predefined AAS model. First, a parser analyzes the information model and extract the data name and the contextual relationships. The name and other text-based meta-description are concatenated into a pseudo-sentence. Symmetrically, the standardized data description in the sub-model is also transformed into a pseudo-sentence. A neural language model is applied to embed the sentences into vectors, i.e., into so-called semantic fingerprints. A pooling layer can be added to reduce the size of the vectors to produce distilled semantic fingerprints.

The semantic fingerprints of the two pseudo-sentences are compared to determine a similarity score. The comparison can be done by calculating the similarity score of the two vectors, preferably with cosine-similarity as evaluated in [18] for NLP-tasks. The higher the semantic similarity score is, the more likely there should be a mapping between the proprietary data property and the standardized data property in the AAS-model. By quantitatively measuring the semantic equivalency between one given property and other properties in descending order, the model can recommend the best matchings.

From the point of view of information theory, the semantic fingerprints of data properties keep the latent semantic information of the meta-description of the data. In this sense, the neural language model can be considered as a neural network memorizing knowledge on how to extract the source information into a unified format, i.e., into semantic fingerprint. The upper layer can be customized to exploit the latent encoded information in vectors for a dedicated application, e.g., similarity search and classification.

IV. Initial Results

The semantic fingerprint can be considered as an index of a standardized concept description that specifies a data property. Based on the semantic fingerprint, we implement a prototype tool that can generate mappings between the data properties from manufacturer’s specification that describes a technical component into a standardized AAS model with standardized ECLASS-definitions (cf. Fig. 4).

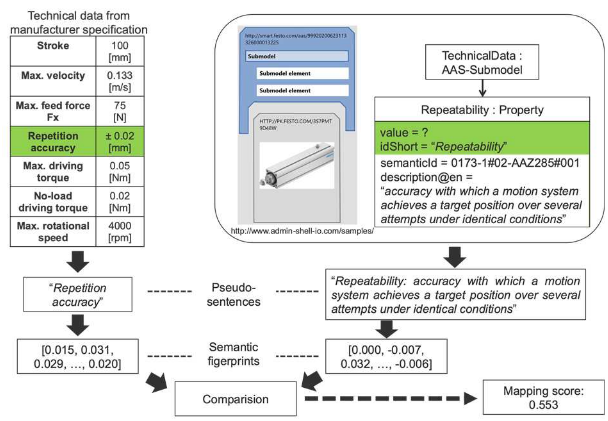

Fig. 4. The mechanism of generating AAS instance with semantic fingerprints

TABLE I. Prototype Results of Data Transformation

Manufacturer's specification — Technical data of an electric cylinder (http://www.admin-shell-io.com/samples/) \ Property template in AAS with standardized ECLASS-definitions	"stroke length"	"maximum piston speed"	"maximum feed force"	"repeatability"	"specified maximum drive torque"	"specified minimum output torque"	"max. rotation speed"	...
"Stroke"	0.241	0.075	0.024	0.085	0.033	0.006	-0.003	
"Max. velocity"	0.118	0.515	0.408	0.200	0.326	0.168	0.436	
"Max. feed force Fx"	0.046	0.261	0.615	0.028	0.249	0.260	0.128	
"Repetition accuracy"	-0.008	0.000	-0.011	0.553	0.004	-0.002	0.009	
"Max. driving torque"	0.142	0.411	0.463	0.108	0.694	0.527	0.457	
"No-load driving torque"	0.055	0.158	0.327	0.034	0.523	0.531	0.238	
"Max. rotational speed"	0.180	0.531	0.415	0.163	0.410	0.232	0.755	

Table I shows the mapping scores in the data transformation. Green color marks the greatest similarity scores for each of the manufacturer’s data properties, which lead to correct mappings in data transformation (red dash line, unidirectional), while yellow color marks the scores of close data properties that almost confuse the algorithm.

V. Discussion and Future Works

In this paper, an approach for automated generation of AAS model with semantic analysis based on neural network is presented. However, there are some limitations to reveal: to analyze the data semantics, this approach has the requirement that the meaning of the data properties shall be verbalized in an intelligible form, e.g., by having a meaningful name, description, comments, or so on. A data property simply named "*m_v*" without any other context information will confuse the algorithm. This approach based on neural model also has more computational complexity for calculating the semantic similarity. An intuitive fact of this is that the neural model has about 300 MB in size and takes a computation time in the range of about 30 milliseconds to compare two pseudo-sentences with a regular PC in Python environment, and it would take minutes to analyze a digital technical data sheet or a detailed AAS model.

Nevertheless, the first results suggest that this approach is feasible, and these limitations can be addressed by the future works based on practical use cases. This method can be used to save manual effort to link heterogeneous data properties together, since it requires expert knowledge as well as effort at reading, understanding and editing. Two types of use cases are presented as follows:

1) To facilitate the information exchange for smart manufacturing: In a connected factory, a harmonized data representation and interfaces are essential to increase the degree of intelligence. Dynamical insights in a comprehensive data space require conceptually well-defined data. For instance, different applications like equipment and process monitoring, quality evaluation, and cost estimation demand sharing the data that have the equivalent conceptual meaning (e.g., "*tool wear-out*"). The semantic fingerprint can be allocated to each data property for classifying the equivalent data together and identifying the inherent correlation, allowing heterogeneous data to be managed, retrieved, and utilized based on the semantics that human experts interpret. Promising use cases under investigation are, for instance, smart search engine, question-answering system, recommendation system and so on.

2) To map the data instance to a knowledge base for reasoning: With semantic fingerprints, instance data can be easily mapped to knowledge concepts in an ontology or knowledge graph in an automated way. The relations defined on the abstract conceptual layer of a knowledge base can be applied to the instance layer to perform verification and reasoning, and conversely, the discovered correlation in the data instances can be generalized on the conceptual level to enrich the knowledge base with dynamical insights.

Acknowledgment

This work was supported by the Ministry of Science, Research and the Arts of the State of Baden-Wurttemberg within the sustainability support of the projects of the *Exzellenzinitiative II*.

References

[1] S. Bader *et al.*, "Details of the asset administration shell part 1 - the exchange of information between partners in the value chain of industrie 4.0 (Version 3.0 RC02)," 2022.

[2] S. Bader *et al.*, "Details of the Asset Administration Shell - Part 2 Interoperability at Runtime – Exchanging Information via Application Programming Interfaces (Version 1.0RC02)." 2021.

[3] C. Plociennik *et al.*, "Towards a Digital Lifecycle Passport for the Circular Economy," in *Procedia CIRP*, 2022, vol. 105. doi: 10.1016/j.procir.2022.02.021.

[4] L. Rauh, S. Gärtner, D. Brandt, M. Oberle, D. Stock, and T. Bauernhansl, "Towards AI Lifecycle Management in Manufacturing Using the Asset Administration Shell (AAS)," *Procedia CIRP*, vol. 107, pp. 576–581, 2022, doi: https://doi.org/10.1016/j.procir.2022.05.028.

[5] R. Pinto and V. Renda, "Manufacturing data-sharing as driver for sustainability in Europe," 2021. Accessed: Jun. 12, 2022. [Online]. Available: https://www.digitaleurope.org/wp/wp-content/uploads/2021/10/Manufacturing-data-sharing-as-driver-for-sustainability-in-Europe.pdf

[6] M. Wenger, A. Zoitl, and T. Müller, "Connecting PLCs With Their Asset Administration Shell For Automatic Device Configuration," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, 2018, pp. 74–79. doi: 10.1109/INDIN.2018.8472022.

[7] F. Ocker, C. Urban, B. Vogel-Heuser, and C. Diedrich, "Leveraging the asset administration shell for agent-based production systems," in *IFAC-PapersOnLine*, 2021, vol. 54, no. 1, pp. 837–844. doi: 10.1016/j.ifacol.2021.08.186.

[8] S. Cavalieri and M. G. Salafia, "A model for predictive maintenance based on asset administration shell," *Sensors (Switzerland)*, vol. 20, no. 21, 2020, doi: 10.3390/s20216028.

[9] S. Cavalieri, S. Mule, and M. G. Salafia, "OPC UA-based Asset Administration Shell," in *IECON Proceedings (Industrial Electronics Conference)*, 2019, vol. 2019-October. doi: 10.1109/IECON.2019.8926859.

[10] "AAS Submodel Templates," *Industrial Digital Twin Association e. V.* https://industrialdigitaltwin.org/en/content-hub/submodels (accessed Jun. 12, 2022).

[11] T. Miny, M. Thies, U. Epple, and C. Diedrich, "Model Transformation for Asset Administration Shells," in *IECON Proceedings (Industrial Electronics Conference)*, 2020, vol. 2020-October. doi: 10.1109/IECON43393.2020.9254649.

[12] M. Platenius-Mohr, S. Malakuti, S. Grüner, J. Schmitt, and T. Goldschmidt, "File- and API-based interoperability of digital twins by model transformation: An IIoT case study using asset administration shell," *Future Generation Computer Systems*, vol. 113, pp. 94–105, Dec. 2020, doi: 10.1016/J.FUTURE.2020.07.004.

[13] S. Cavalieri and M. G. Salafia, "Insights into mapping solutions based on OPC UA information model applied to the industry 4.0 asset administration shell," *Computers*, vol. 9, no. 2, 2020, doi: 10.3390/computers9020028.

[14] M. Redeker, S. Volgmann, F. Pethig, and J. Kalhoff, "Towards Data Sovereignty of Asset Administration Shells across Value Added Chains," in *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 2020, vol. 2020-September. doi: 10.1109/ETFA46521.2020.9211955.

[15] "GLUE Benchmark." https://gluebenchmark.com/ (accessed Jun. 12, 2022).

[16] R. Bommasani *et al.*, "On the Opportunities and Risks of Foundation Models." arXiv, 2021. doi: 10.48550/ARXIV.2108.07258.

[17] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a Few Examples: A Survey on Few-shot Learning," *ACM Computing Surveys*, vol. 53, no. 3, 2020, doi: 10.1145/3386252.

[18] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," 2019. doi: 10.18653/v1/d19-1410.