

Received March 6, 2022, accepted March 28, 2022, date of publication March 31, 2022, date of current version April 7, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3163738

Toward Data Interoperability of Enterprise and Control Applications via the Industry 4.0 Asset Administration Shell

XUN YE¹, WON SEOK SONG², SEUNG HO HONG^{2,3}, (Senior Member, IEEE),
YU CHUL KIM², AND NAM HYUN YOO⁴, (Member, IEEE)

¹Department of Information Engineering, Wuhan University of Technology, Wuhan 430070, China

²Nestfield Company Ltd., Gyeonggi Technopark, Ansan 15588, South Korea

³Department of Electronic Engineering, Hanyang University, Ansan 15588, South Korea

⁴School of Computer Science & Engineering, Kyungnam University, Changwon 51767, South Korea

Corresponding author: Seung Ho Hong (shhong@hanyang.ac.kr)

This work was supported by the Institute for Information & Communications Technology Promotion (IITP) funded by the Korean Government [Ministry of Science, ICT and Future Planning (MSIP)] through the Development & Proof of Open Manufacturing Operation System with Information and Communications Technology (ICT) Convergence under Grant 2020-0-00299-002.

ABSTRACT In a cyber-physical system, enterprise applications, control functions, and physical manufacturing assets are interconnected to seamlessly exchange information. However, the software and hardware often differ in terms of their data interfaces, data formats and semantics. Conventional enterprise applications usually employ file-based interfaces (e.g., Excel); control applications always support Ethernet-based protocols (e.g., OPC UA). Thus, data interoperability is currently lacking. The asset administration shell (AAS) is a new approach for achieving data interoperability across the automation pyramid as the AASX data format represents AAS information and supports data communication via OPC UA. In this study, we first provide insights into the AASX format and give exemplary applications on AASX; we then present an AASX-based solution for bidirectional data exchange between enterprise and control applications. In this solution, the conversion between AASX and Excel spreadsheets is completed to enable interoperable data exchange. Finally, a data exchange scenario for motor control is constructed to validate the effectiveness of the solution. The result shows no data were lost during the two-way conversions.

INDEX TERMS Asset administration shell, AASX, data interoperability, data exchange, enterprise and control applications.

I. INTRODUCTION

The Internet of Things (IoT) and cyber-physical systems (CPSs) have emerged as key technologies that fill many user and application requirements in the era of the fourth industrial revolution, also termed Industry 4.0 [1]. Soon, manufacturing operations will be digitalized and decentralized from the shop floor to the office, and across entire enterprise networks [2]. Traditional manufacturing systems must change.

Modeling, simulation, and integration of product, process, and resource (PPR) throughout the production lifecycle is a promising approach for enhancing manufacturing efficiency [3]. Manufacturing assets such as sensors, machines,

and control systems (applications or software) from different manufacturers/suppliers generate massive data during production. The system integrator or plant operator must expend considerable effort to integrate the data into higher-level enterprise systems that assist decision-making [4]. Manufacturing data can be represented and stored in various forms [5], further hindering efficient and transparent data transmission between control systems and enterprise applications. Specifically, industrial control systems often have Ethernet-based interfaces, such as the open platform communications unified architecture (OPC UA), which support runtime data communication. Conventional enterprise applications, such as enterprise resource planning systems (ERPs), usually employ file-based methods to communicate data, such as comma-separated value (CSV) files and Microsoft Excel spreadsheets. A file is exchanged between two different applications

The associate editor coordinating the review of this manuscript and approving it for publication was Gianmaria Silvello¹.

in a direct manner (from end-to-end) or indirectly via an intermediate data hub. Data exchange is often done offline, which is time-consuming and error-prone.

Moreover, the end-application must have an identical file-based interface, which requires significant system development. Data homogeneity (consistent semantic production data comprehensible to all users) is essential to fulfil the needs of all engineering chain participants. Data acquisition must be simple, and it must be possible to rapidly and selectively access information [6]. Thus, a data interoperability solution ensuring effortless data integration, comprehensive semantic data homogenization, delivery of contextual information, and full access to and control of data flow is required for industry [7].

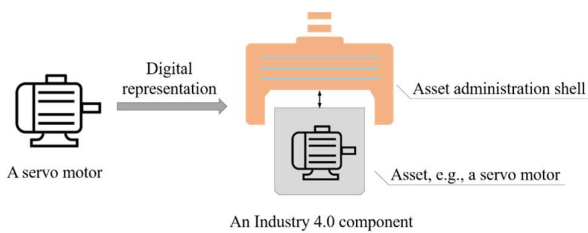


FIGURE 1. An AAS transforms a manufacturing asset into an Industry 4.0 component.

Recently, the use of digital twins (DTs) to model on-site manufacturing resources has improved data interoperability [8]. The Industry 4.0 asset administration shell (AAS), a practical example of the DT, can make data discoverable, identifiable and accessible, thus facilitating interoperability among the applications of a manufacturing company [9], [10]. AASX is a package file format for representing an AAS. An AASX interface for field hardware assets was developed previously [24]. Here, we focus on the data interconnections between high- and mid-level control applications. As explained above, existing enterprise applications usually employ file-based methods (e.g., an Excel spreadsheet) to exchange information, whereas control applications always support Ethernet-based protocols (e.g., OPC UA). AASX can support data communication via OPC UA. Hence, the goal of this study is to present an AASX-based solution that allows bidirectional, lossless information exchange between enterprise and control applications. The information semantics are implemented in the AASX to ensure data interoperability.

Therefore, the main contributions of this paper are as follows:

- We review state-of-the-art research on AASX using several approaches to AASX implementation as examples.
- We present an AASX-based data exchange solution that connects enterprise- and control-level applications.
- Our solution guarantees bidirectional, lossless data conversion between AASX and Excel.
- We use a test case to validate the feasibility of our solution.

- We find that AASX serves as an intermediary in terms of data exchange, thus effectively eliminating the data siloes that keep enterprise and control applications separate.

The rest of the paper is organized as follows. Section II gives key background knowledge on AAS and AASX. Section III describes state-of-the-art data exchange solutions. Section IV introduces our AASX-based data exchange solution and details the data conversion between AASX and Excel. Section V describes the test case (with the data conversion results). Section VI draws conclusions and describes our planned future work.

II. BACKGROUND ON AAS AND AASX

As shown in Figure 1, an AAS is a standardized digital representation of an asset with industrial applications that allows uniform access to information and functions, thereby facilitating interoperability among the applications of a manufacturing company [11]. An AAS offers an interoperable way to capture key information on assets, including their intrinsic properties, operational parameters, and technical functionalities; and enables straightforward interactions over standardized and secure communication links with other Industry 4.0 components [12].

The structure of an AAS is defined in Part 1 of the AAS specification [13]. As shown in Figure 2(a), the AAS meta-model has a ‘Submodel’ that defines a specific aspect of ‘AssetInformation’. The ‘Submodel’ contains the ‘SubmodelElement’ that orchestrates asset information into, for example, a ‘Property’. The ‘Property’ definition is linked to that of ‘ConceptDescription’, which references a semantic ‘DataSpecification’ such as the common data dictionary (CDD) [14] or ECLASS [15]. AAS model elements allow several types of identifiers (IDs), including internationalized resource identifiers (IRIs), international registration data identifiers (IRDIs), and custom IDs; these identify elements within a single AAS, or in an AAS network within (or extending beyond) a company, depending on the purpose.

Currently, an AAS is considered as a promising concept within the Industry 4.0 initiative. Information and communication technologies promote practical usage of AASs. State-of-the-art studies on AASs in academia and industry generally focus on specific cases [16]–[18]; methods of AAS implementation differ [19]–[21]. AutomationML, OPC UA, and web ontology languages such as the resource description framework (RDF) have been employed. A standardized guideline (or consensus technical approach) is urgently required for AAS implementation.

As shown in Figure 2(b), in line with the AAS meta-model, the AASX file encapsulates AAS information including the AAS itself, the submodel element, the property, a concept description, and supplementary files, in a structurally robust manner. Compared to existing engineering approaches to data interoperability (e.g., AutomationML), the AASX not only connects to field-level hardware assets (e.g., motors), but also enables data integration into enterprise-level applications

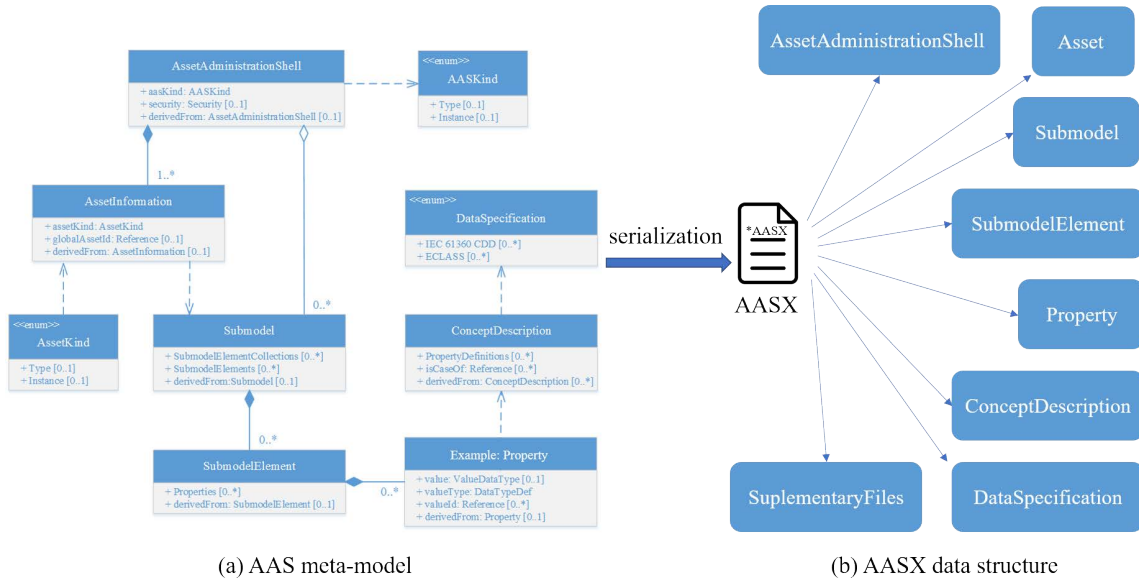


FIGURE 2. An AAS meta-model and serialization thereof in AASX format.

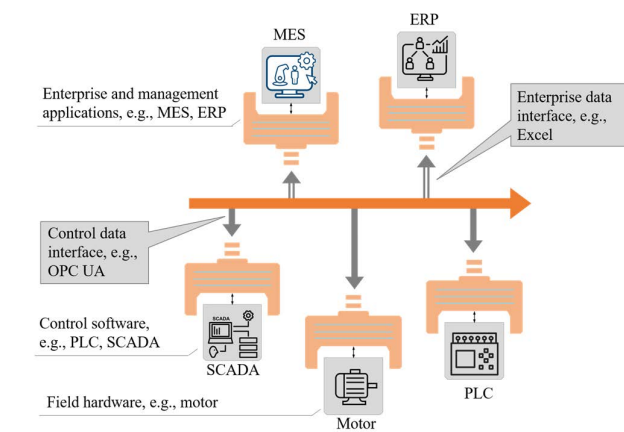


FIGURE 3. An intermediate AASX interface connecting hardware and software.

(e.g., ERP) and promotes data transmission between such software applications. Unambiguous data semantics are guaranteed; all hardware and software of an entire automation pyramid can be interconnected in a networked and non-hierarchical system, allowing information exchange via uniform AASX interfaces. As shown in Figure 3, field hardware, control software, and enterprise and management applications can be connected to an intermediate AASX interface, although their data interfaces vary.

The Industrial Digital Twin Association (IDTA) has developed a series of AASX-based open-source applications that can create, edit, and view AAS information [22]. These applications allow users to obtain an intuitive understanding of how an AAS is structured, and information accessed and exchanged. The applications have different AAS data access methods, all of which are in line with Parts 1 and 2 [23]

of the AAS specification. Below, we summarize two key applications, the first of which is an AASX package explorer. This C#-based AAS editor manipulates AASX information and implements several application programming interfaces (APIs), including representational state transfer (REST), message queuing telemetry transport (MQTT), and OPC UA. These APIs allow other applications to access AASX information. Security mechanisms (e.g., token-based authentication) are also embedded. As shown in Figure 4, an asset and its AAS information, as shown in an AASX package explorer in Figure 4(a), can be accessed and browsed by an OPC UA client (e.g., UaExpert) [Figure 4(b)]; the information can also be selectively accessed via the REST interface by a web browser. The result (e.g., the submodel data) is shown in a JSON file in Figure 4(c).

The second application, i.e., an AASX server, is derived from the AASX package explorer. The AASX server is a C#-based program that implements a server hosting AAS information (an AASX model). The AASX server can be accessed via the REST, MQTT, and OPC UA protocols; secure access is guaranteed.

However, these two programs work on only high-level devices such as personal computers. Prior to integration into user-specific applications and deployment over resource-constrained devices, the AASX model must first be converted into a machine-interpretable data format, such as extensible markup language (XML) or JavaScript object notation (JSON). This function is embedded in the AASX package explorer. In our previous study [24], we applied AASX to a robot-based manufacturing system. We mapped runtime data from field-level robots to an AASX model and integrated the model into OPC UA for data communication with web applications in the cloud.



FIGURE 4. An AASX package explorer and the AASX information accessed via OPC UA and REST. (a) AAS information shown in the AASX package explorer. (b) AAS information shown in UaExpert. (c) AAS information shown in JSON.

Several open-source projects focusing on AASX implementation are listed in the GitHub repository [22]. For example, the ‘java-serializer’ project serializes/deserializes JAVA classes into/from AASX instances, while the ‘aas-transformation-library’ project transforms AutomationML models to AASX models; the results are shown in JSON format.

III. STATE-OF-THE-ART

Many researchers have worked on data exchange among industrial applications or engineering tools. For example, automation markup language (AutomationML) is a neutral data format that connects heterogeneous modern engineering tools from different disciplines. AutomationML describes real plant components in different levels of detail [25]. However, data semantics have not received much attention; AutomationML focuses principally on the engineering phase of the plant life-cycle. The MTConnect data exchange standard uses a domain-specific vocabulary to devise data models of factory shop-floor equipment [26]. However, MTConnect is a read-only protocol employed principally to monitor data from machine tool components. Therefore, both AutomationML and MTConnect

could not prove the data interoperability between different applications.

In academic research, AASX is widely used as a data representation and exchange format for AASs. It is highly likely that AASX will become an official data format for AASs. IEC project 63278-1 aims to develop an international standard of AAS [11], while another study [27] described a lightweight software framework (in line with AAS specifications) for implementation of an AAS; AASX was used to show such an implementation. Finally, open-source solutions for AAS implementation were comprehensively analyzed and a method by which an AASX-based AAS could be implemented into an embedded device was presented, dramatically reducing the computational requirement [28]. However, above studies only concentrated on AASX itself rather than data exchange scenarios between applications using AASX.

Only a few studies focused on proposing AASX-enabled data exchange solutions. For example, one study [29] developed an AAS model for two different software tools; AASX was employed for integration, while another [30] mapped the proprietary ABB Ability information model to the standard AAS information model using AASX. In [31], AASX was used to develop a data AAS for managing cross-enterprise

engineering data sets. These studies promote the adoption of AASX for solving various data interoperability problems. However, it is of great significance to design a general AASX-enabled data interoperability solution, which not only enables consistent data exchange between enterprise and control applications but also extends to the control of field devices.

IV. AASX DATA EXCHANGE METHOD FOR ENTERPRISE AND CONTROL APPLICATIONS

We first present the data exchange challenge facing the enterprise and control applications, then propose a solution for data conversion between AASX and Excel, and finally detail data conversion steps.

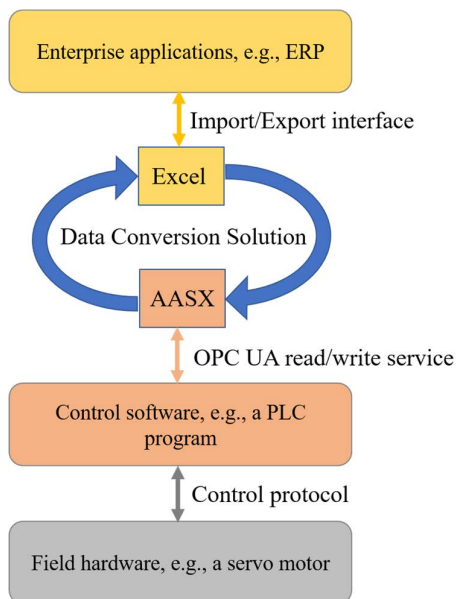


FIGURE 5. AASX solution for data exchange between enterprise and control applications.

production data. Control systems transmit the collected data to enterprise applications for analysis. The enterprise applications then create a manufacturing schedule and instruct the control systems to handle the new production task that the field devices will perform. Such two-way information exchange lacks consistent data exchange between the enterprise applications and control systems because enterprise-level applications usually use file-based interfaces, such as Excel, whereas control-level systems employ industrial communication interfaces such as OPC UA. AASX can support data communication via OPC UA; the AASX data format can serve as an intermediary between enterprise applications and control systems. The only remaining challenge is data conversion between AASX and Excel.

B. THE SOLUTION

The entire two-way information exchange in the above scenario is shown in Figure 5. We designed an AASX solution permitting two-way information exchange. The AASX solution is implemented as follows. Working from the bottom up, all field asset information and data collected by the control systems are first converted into an AASX file using AASX package explorer. Then, a web-converting tool [32] is used for data conversion from the AASX file to Excel. Finally, the Excel file is imported into the enterprise applications. From the top down, the tool converts Excel data into AASX format. Here, we focus on data conversion because we achieved AASX data communication via OPC UA in our previous publication [24]; moreover, Excel file importation/exportation is within the scope of specific industrial enterprise-level applications, so is not considered here.

C. DATA CONVERSION STEPS

Figure 6 shows the data conversion process between AASX and Excel, also the design process of the web-converting tool. The requirement of designing the solution is realizing the bidirectional and lossless data conversion between the source and destination. The tool has two independent parsing programs written in Python. One tool parses the AASX model (a JSON file) and generates a well-formatted Excel file, and the other performs the opposite function. The two Python programs have been deposited in the GitHub repository [33].

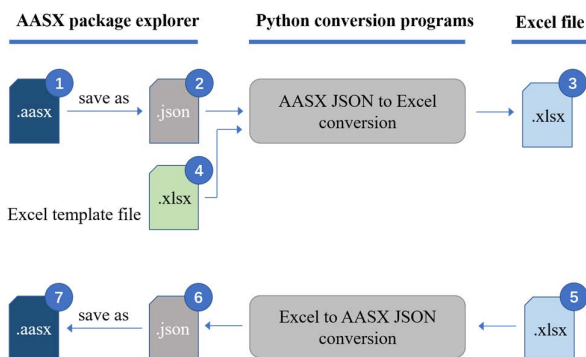


FIGURE 6. Data conversion from AASX to Excel and vice versa.

A. THE PROBLEM

Consider the following manufacturing scenario. Field devices perform specific tasks and generate large amounts of

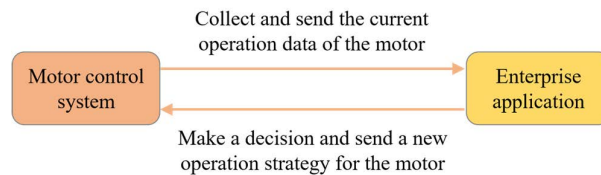


FIGURE 7. Data exchange for the motor control scenario.

1) AASX TO EXCEL

As shown in the upper part of Figure 6, first, an AASX file (No. 1 in Figure 6) is prepared using the AASX package

TABLE 1. AASX items included in the Excel file.

AASX elements:	Asset	
	AssetAdministrationShell	
	Submodel	
	SubmodelElementCollection	
	SubmodelElement (Property)	

(a)

Semantic ID:	Referable:	idShort
		category
	Kind:	kind
	Identifiable:	idType
		local
	id	

(b)

Semantic Contents:	shortName
	preferredName
	dataType
	value
	valueType
	unit
	definition

(c)

explorer. This file must be validated using the modeling rules of the AASX package explorer; otherwise, warnings or error messages will pop up. Second, the AASX file is exported as a JSON file (No. 2), which is supported in AASX package explorer. Third, the JSON file is uploaded to the web-converting tool that deserializes JSON data to Python objects. Finally, the Python program outputs an Excel file (No. 3) based on a prespecified Excel template (No. 4) that has a hierarchical structure for storage of parsed AASX information. In the third step, the tool parses every element of the AASX model in the JSON file. Table 1 shows the items output to the Excel file. Specifically, Table 1(a) lists key elements of the AASX file, Table 1(b) lists the identification information for the AASX elements, and Table 1(c) lists the semantic contents of the ‘ConceptDescription’ in the AASX file.

2) EXCEL TO AASX

This inverse procedure can be briefly described as follows. As shown at the bottom of Figure 6, the Excel template is filled with data from the enterprise applications. Then, the Excel file (No. 5) is uploaded to the web tool for conversion. Python parses the Excel file, engages in serialization, and outputs the result to a JSON file (No. 6). Finally, the AASX package explorer loads the JSON file and saves AASX information as an AASX file (No. 7). Note that the tool can only handle files that meet the requirements of the prespecified Excel template.

V. TEST CASE: MOTOR CONTROL

Given the manufacturing scenario described in section III, we now give an example of real data exchange between a motor control system and enterprise application. We examine the AASX solution and show lossless data conversion between AASX and Excel. We do not develop enterprise or motor control applications.

A. THE DATA EXCHANGE SCENARIO

The data exchange scenario for motor control is shown in Figure 7. After a specific motor operation, the control system

must send the operation data of the motor (e.g., rotation speed) to the enterprise application; enterprise collects the data, analyzes it, and sends a new command (e.g., control mode) to the motor control system. A new task can then be performed by the motor. Detailed AASX model information are introduced in the following.

B. THE AASX MODEL

Figure 8 shows the model of motor control system in AASX package explorer. The AASX model serves as a benchmarking file in our example of data exchange. The file contains all AAS information that will be exchanged between the enterprise application and motor control system. Our AASX model was created based on the official AASX example provided by IDTA, which illustrates every discipline/rule that users must follow to create their own AASX models. We adopt this AASX example because it is both elaborate and comprehensive, and allows us to validate the effectiveness of our data conversion solution.

As shown at the top of Figure 8, the AASX model describes an AAS ‘ExampleMotor’ for the asset ‘ServoDC-Motor’. The AAS contains five well-defined submodels that describe the properties and capacities of motors used in different disciplines. ‘Identification’ contains supplier information (e.g., the name of the manufacturer). ‘TechnicalData’ defines the technical parameters of the motor (e.g., the maximum torque). ‘OperationalData’ contains the runtime parameters of the motor (e.g., the rotation speed). ‘Documentation’ stores and classifies the documents of an asset (e.g., the operation manual). The above four submodels are indispensable for describing an asset. The last submodel, ‘EnterpriseData’, is customized for the motor control scenario, which contains a ‘SubmodelElement’ or ‘Property’ termed ‘ControlMode’; the enterprise user employs this to control motor operation.

The bottom half of Figure 8 shows the detailed attributes of ‘ControlMode’ (e.g., the semantic ID and contents). The semantic ID links the ‘SubmodelElement’ to its definition (e.g., ‘ConceptDescription’). Moreover, ‘ConceptDescription’ contains semantic descriptions of the ‘SubmodelElement’; it is possible to refer to semantic ‘DataSpecifications’ (i.e., data standards such as ECLASS and IEC 61360 that allow for formal, common, and unambiguous definitions of properties). If no appropriate definition can be found in the standards, users can create and reference a public ‘DataSpecification’ by extending IEC 61360. The property should have the data value and value type. Supplementary files attached to the AASX model do not need to be converted, and are not worked on by the conversion tool.

C. DATA EXCHANGE RESULT

The data exchange results of the motor control scenario are shown in Figure 9. From AASX to Excel, the example AASX model is first exported to a JSON file. Figure 9(a) shows an extract of the Excel file converted from the JSON file; the extract contains the motor’s AAS, submodels, and properties. This Excel file can be further imported into enterprise

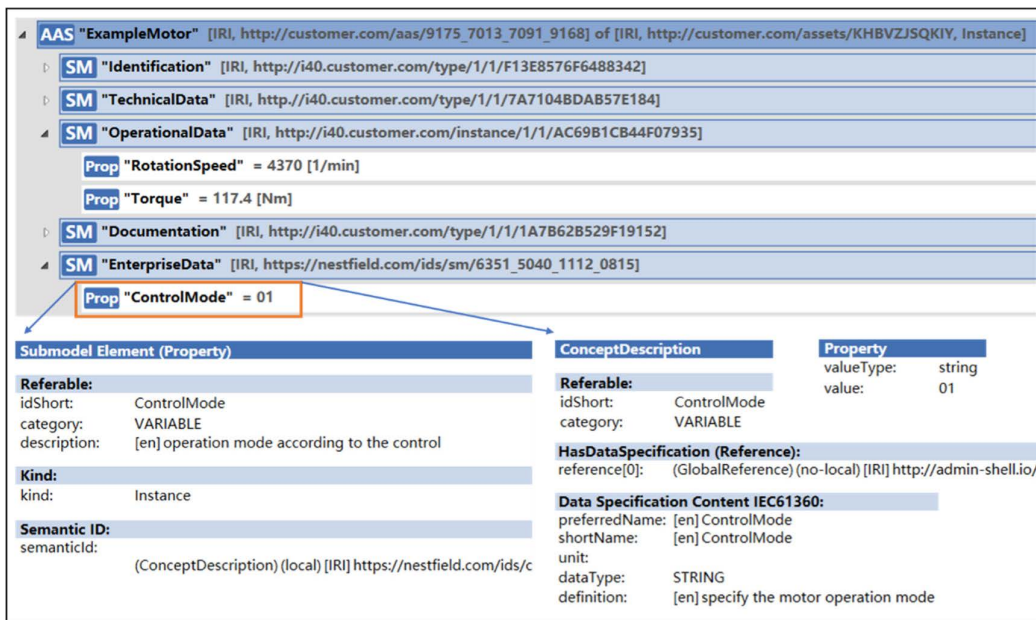


FIGURE 8. The AASX model and its attributes, as shown by the AASX package explorer.

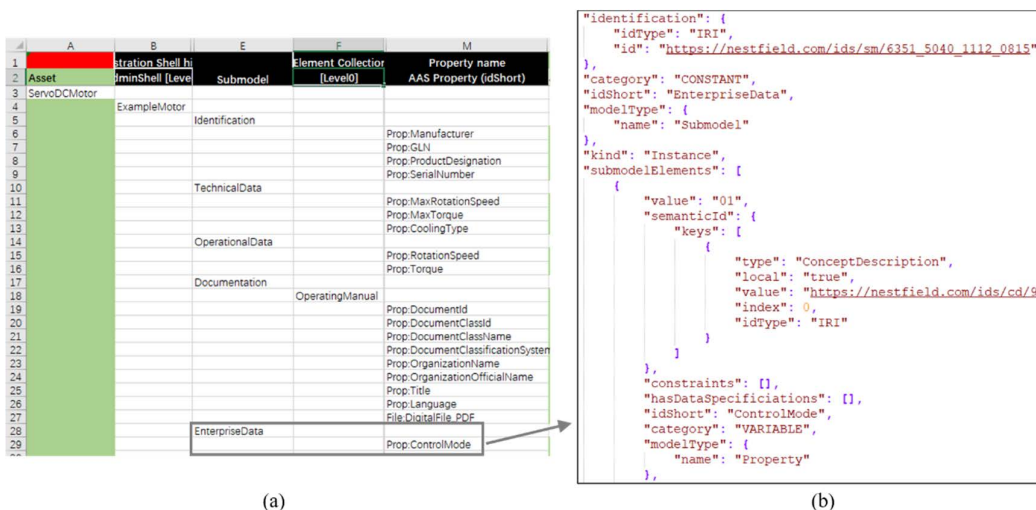


FIGURE 9. Data exchange results of the motor control scenario. (a) An extract of the Excel file. (b) An extract of the JSON file converted from the Excel file.

applications. On the other hand, when converting from Excel to AASX, we use the Excel file generated in the first round as the input. Figure 9(b) shows an extract of the JSON file converted from the Excel file. This JSON data block contains information on the motor’s AAS submodel ‘EnterpriseData’. The JSON file can be imported and displayed in the AASX package explorer.

We compared the generated JSON file to the original JSON file exported from the AASX model, and found no data loss after the two-way conversion. Thus, the two AASX models (before and after the conversions) are identical. This proves the validity of our conversion tool, and the feasibility of our AASX solution for lossless data exchange between enterprise

and control applications. All files have been uploaded to the GitHub repository [34] and are freely available.

To ensure successful conversion using the online converting tool, certain rules must be followed when preparing the AASX file. For example, both the value type and initial value of a property must be assigned. The property in the submodel must be mapped to the corresponding ‘ConceptDescription’, and the definition of the ‘ConceptDescription’ must not be empty.

VI. CONCLUSION AND FUTURE WORK

In terms of the data interoperability required by Industry 4.0, the AAS is emerging as a digitalized approach that enables

intimate interactions between cyber applications and physical devices. Given a uniform AAS interface, all manufacturing components (hardware devices and software applications) can be interconnected. In a fully integrated system, enterprise applications and manufacturing functions are aligned for efficient data flow. However, existing research has paid little attention to the semantic aspects of information exchange. To ensure robust data exchange between enterprise and control applications, we describe data conversion between AASX and Excel; the latter is generally used to manage the business-related data of enterprise applications. AASX is an Industry 4.0 AAS-compliant data representation format supporting data communication with field-level manufacturing assets via OPC UA, for example. Moreover, appropriate semantic descriptions of information are provided by the AASX model, which can reference data standards such as IEC 61360. Therefore, data interoperability between enterprise and control applications is guaranteed if bidirectional lossless conversion between AASX and Excel is possible.

We validated our data conversion solution using a motor control scenario. We developed an AASX model for motor control based on the AASX file provided by IDTA. No data were lost during the two-way conversions. We focused on data conversion, and are not aiming to develop enterprise or control applications at present. Although our case study is simple, it proves the effectiveness of our AASX-based data conversion method. One limitation of the current solution is that the data conversion is completed by human operators. In future, we will develop an automatic converting solution which can support runtime data conversion. We will also apply the method to enterprise and control applications in real industries. Conversion between AASX and other data formats, such as IEC 62264 Business to Manufacturing Markup Language (B2MML) [35], will also be explored for two organizations to exchange business information.

REFERENCES

- [1] M. Sanchez, E. Exposito, and J. Aguilar, "Industry 4.0: Survey from a system integration perspective," *Int. J. Comput. Integr. Manuf.*, vol. 33, nos. 10–11, pp. 1017–1041, 2020, doi: [10.1080/0951192X.2020.1775295](https://doi.org/10.1080/0951192X.2020.1775295).
- [2] A. W. Colombo, S. Karnouskos, X. Yu, O. Kaynak, R. C. Luo, Y. Shi, P. Leitao, L. Ribeiro, and J. Haase, "A 70-year industrial electronics society evolution through industrial revolutions: The rise and flourishing of information and communication technologies," *IEEE Ind. Electron. Mag.*, vol. 15, no. 1, pp. 115–126, Mar. 2021, doi: [10.1109/MIE.2020.3028058](https://doi.org/10.1109/MIE.2020.3028058).
- [3] J. Jasperneite, T. Sauter, and M. Wollschlaeger, "Why we need automation models: Handling complexity in industry 4.0 and the internet of Things," *IEEE Ind. Electron. Mag.*, vol. 14, no. 1, pp. 29–40, Mar. 2020, doi: [10.1109/MIE.2019.2947119](https://doi.org/10.1109/MIE.2019.2947119).
- [4] B. Cheng, J. Zhang, G. P. Hancke, S. Karnouskos, and A. W. Colombo, "Industrial cyberphysical systems: Realizing cloud-based big data infrastructures," *IEEE Ind. Electron. Mag.*, vol. 12, no. 1, pp. 25–35, Mar. 2018.
- [5] R. S. Peres, M. Parreira-Rocha, A. D. Rocha, J. Barbosa, P. Leitao, and J. Barata, "Selection of a data exchange format for industry 4.0 manufacturing systems," in *Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2016, pp. 5723–5728, doi: [10.1109/IECON.2016.7793750](https://doi.org/10.1109/IECON.2016.7793750).
- [6] *Data Homogenization in the Age of Industry 4.0*, Bosch Connected Ind., Stuttgart, Germany, Jul. 2019. [Online]. Available: <https://www.bosch-connected-industry.com/de/en/latest-news/discover-our-nexeed-white-papers/data-homogenization>
- [7] T. Burns, J. Cosgrove, and F. Doyle, "A review of interoperability standards for industry 4.0," *Proc. Manuf.*, vol. 38, pp. 646–653, Jan. 2019.
- [8] R. Minerva, G. M. Lee, and N. Crespi, "Digital twin in the IoT context: A survey on technical features, scenarios, and architectural models," *Proc. IEEE*, vol. 108, no. 10, pp. 1785–1824, Oct. 2020, doi: [10.1109/JPROC.2020.2998530](https://doi.org/10.1109/JPROC.2020.2998530).
- [9] C. Wagner, J. Grothoff, U. Epple, R. Drath, S. Malakuti, S. Gruner, M. Hoffmeister, and P. Zimmermann, "The role of the industry 4.0 asset administration shell and the digital twin during the life cycle of a plant," in *Proc. 22nd IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2017, pp. 1–8.
- [10] X. Ye, J. Jiang, C. Lee, N. Kim, M. Yu, and S. H. Hong, "Toward the plug-and-produce capability for industry 4.0: An asset administration shell approach," *IEEE Ind. Electron. Mag.*, vol. 14, no. 4, pp. 146–157, Dec. 2020, doi: [10.1109/MIE.2020.3010492](https://doi.org/10.1109/MIE.2020.3010492).
- [11] *Industrial-Process Measurement, Control and Automation-Asset Administration Shell for Industrial Applications—Part 1: Administration Shell Structure*, IEC Standard CD 63278-1, 2020.
- [12] X. Ye and S. H. Hong, "Toward industry 4.0 components: Insights into and implementation of asset administration shells," *IEEE Ind. Electron. Mag.*, vol. 13, no. 1, pp. 13–25, Mar. 2019, doi: [10.1109/MIE.2019.2893397](https://doi.org/10.1109/MIE.2019.2893397).
- [13] *Details of the Asset Administration Shell: Part1-the Exchange of Information Between Partners in the Value Chain of Industrie 4.0 (Version 3.0RC01)*, Plattform Industrie 4.0, Berlin, Germany, May 2020. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.html
- [14] *Common Data Dictionary*, IEC Standard 61360 (CDD, V2.0014.0017), 2017. [Online]. Available: <https://cdd.iec.ch/cdd/iec61360/iec61360.nsf>
- [15] ECLASS. *ECLASS Standard*. Accessed: Nov. 2021. [Online]. Available: <https://www.eclass.eu/en/html>
- [16] M. Birtel, W. Motsch, J. Popper, and M. Ruskowski, "Method for the development of an asset administration shell in a product-driven modular production—realizing an active digital object memory," in *Proc. IEEE Conf. Ind. Cyberophys. Syst. (ICPS)*, Jun. 2020, pp. 583–588, doi: [10.1109/ICPS48405.2020.9274753](https://doi.org/10.1109/ICPS48405.2020.9274753).
- [17] B. Lv, Y. Zhang, and W. Dai, "Integrated asset management model based on IEC 61499 and administration shell," in *Proc. 47th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2021, pp. 1–6, doi: [10.1109/IECON48115.2021.9589597](https://doi.org/10.1109/IECON48115.2021.9589597).
- [18] L. Sakurada, P. Leitao, and F. De la Prieta, "Towards the digitization using asset administration Shells," in *Proc. 47th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2021, pp. 1–6, doi: [10.1109/IECON48115.2021.9589370](https://doi.org/10.1109/IECON48115.2021.9589370).
- [19] A. Sidorenko, M. Volkmann, W. Motsch, A. Wagner, and M. Ruskowski, "An OPC UA model of the skill execution interaction protocol for the active asset administration Shell," *Proc. Manuf.*, vol. 55, pp. 191–199, Jan. 2021.
- [20] A. Luder, A.-K. Behnert, F. Rinker, and S. Biffl, "Generating industry 4.0 asset administration shells with data from engineering data logistics," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 867–874, doi: [10.1109/ETFA46521.2020.9212149](https://doi.org/10.1109/ETFA46521.2020.9212149).
- [21] S. R. Bader and M. Maleshkova, "The semantic asset administration Shell," in *Proc. Int. Conf. Semantic Syst.*, vol. 11702. Cham, Switzerland: Springer, 2019, pp. 159–174, doi: [10.1007/978-3-030-33220-4_12](https://doi.org/10.1007/978-3-030-33220-4_12).
- [22] *Admin-Shell-IO by IDTA*. Accessed: Nov. 2021. [Online]. Available: <https://github.com/admin-shell-io>
- [23] *Details of the Asset Administration Shell: Part2-Interoperability at Runtime—Exchanging Information via Application Programming Interfaces (Version 1.0RC02)*, Plattform Industrie 4.0, Berlin, Germany, Nov. 2021. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part2_V1.html
- [24] X. Ye, S. H. Hong, W. S. Song, Y. C. Kim, and X. Zhang, "An industry 4.0 asset administration shell-enabled digital solution for robot-based manufacturing systems," *IEEE Access*, vol. 9, pp. 154448–154459, 2021, doi: [10.1109/ACCESS.2021.3128580](https://doi.org/10.1109/ACCESS.2021.3128580).
- [25] A. Garmendia, M. Wimmer, A. Mazak-Huemer, E. Guerra, and J. de Lara, "Modelling production system families with AutomationML," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2020, pp. 1057–1060, doi: [10.1109/ETFA46521.2020.9211894](https://doi.org/10.1109/ETFA46521.2020.9211894).
- [26] C. Liu, H. Vengayil, Y. Lu, and X. Xu, "A cyber-physical machine tools platform using OPC UA and MTCConnect," *J. Manuf. Syst.*, vol. 51, pp. 61–74, Apr. 2019.

[27] S. Wein, L. Korfer, S. Storms, and C. Brecher, "Flexible and lightweight framework for active industrial asset administration shells," in *Proc. IEEE Conf. Ind. Cyberphys. Syst. (ICPS)*, Jun. 2020, pp. 549–554, doi: [10.1109/ICPS48405.2020.9274761](https://doi.org/10.1109/ICPS48405.2020.9274761).

[28] R. Pribiš, L. Beňo, and P. Drahoš, "Asset administration shell design methodology using embedded OPC unified architecture server," *Electronics*, vol. 10, no. 20, p. 2520, Oct. 2021, doi: [10.3390/electronics10202520](https://doi.org/10.3390/electronics10202520).

[29] A. Deuter and S. Imort, "PLM/ALM integration with the asset administration Shell," *Proc. Manuf.*, vol. 52, pp. 234–240, Jan. 2020.

[30] 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 Gener. Comput. Syst.*, vol. 113, pp. 94–105, Dec. 2020.

[31] A. Löcklin, H. Vietz, D. White, T. Ruppert, N. Jazdi, and M. Weyrich, "Data administration Shell for data-science-driven development," *Proc. CIRP*, vol. 100, pp. 115–120, Jan. 2021.

[32] *Nestfield AASX Parser*. Accessed: Nov. 2021. [Online]. Available: <https://www.aasnest.io/aasx2xls>

[33] *AASX-XLS-Parser*. Accessed: Nov. 2021. [Online]. Available: <https://github.com/admin-shell-io/aasx-xls-parser>

[34] *Testing Files for Nestfield AASX Parser*. Accessed: Nov. 2021. [Online]. Available: <https://github.com/hanyangyexun/Nestfield-AASX-solution>

[35] H. Schweizer, R. Alt, K. Schmitz, and M. Wollschlaeger, "B2MML as an exchange format for asset administration shells as part of a plug-and-produce process for a fluid power engineering application," in *Proc. 12th Int. Fluid Power Conf.*, 2020, pp. 27–33, doi: [10.25368/2020.66](https://doi.org/10.25368/2020.66).



SEUNG HO HONG (Senior Member, IEEE) received the B.S. degree from Yonsei University, Seoul, South Korea, in 1982, the M.S. degree from Texas Tech University, Lubbock, in 1985, and the Ph.D. degree from Pennsylvania State University, University Park, in 1989, all in mechanical engineering. He is currently an Emeritus Professor with the Department of Electronic Engineering, Hanyang University, Ansan, South Korea, and an Executive Advisor/a fellow of Nestfield Company Ltd. He was a Visiting Scholar with the U.S. National Institute of Standards and Technology and the Vienna University of Technology, Austria, the Shenyang Institute of Automation, and the Chongqing University of Posts and Telecommunications, China. His research interests include smart manufacturing, smart grid, Industrial Internet of Things, cyber-physical systems, and artificial intelligence. He is a member of the IEEE Industrial Electronics Society and the Co-Convenor of the IEC TC65 JWG17 System interface between industrial facilities and the smart grid.



XUN YE received the B.S. degree in mechanical engineering and automation from the Wuhan University of Science and Technology, China, in 2016, and the Ph.D. degree from the Department of Electronic Engineering, Hanyang University, Ansan, South Korea. He is currently a Lecturer with the Department of Information Engineering, Wuhan University of Technology. His research interests include information technology in industrial automation, cyber-physical production systems, and industry 4.0.



YU CHUL KIM received the M.S. degree in control and measurement engineering on the subject of industrial communication networks from Hanyang University, South Korea, in 2000. He has been developing various wired and wireless field network solutions, and is currently working as the Representative of Nestfield Company Ltd., which provides Industrial IoT and smart manufacturing solutions. He is active in IEC TC 65 and ISO TC 184.



WON SEOK SONG received the Ph.D. degree in control and measurement engineering from Hanyang University, South Korea, in 2005. He majored in industrial and building automation network technology. He has developed smart home and building solutions, and currently has been in-charge of developing solutions for the smart industry. He is currently with Nestfield Company Ltd., Ansan, South Korea. He is also engaged in standard activities, such as IEC TC 65 and IEC TC 3 SC 3D.



NAM HYUN YOO (Member, IEEE) received the M.S. and Ph.D. degrees in intelligent agent and semantic web from Suncheon National University, South Korea, in 1999 and 2007, respectively. He had conducted research on cooperative intelligence, from 2008 to 2010, with the Artificial Intelligence Super Laboratory, University of Oklahoma. Since 2011, he has been conducting research on context awareness, digital twin, and software engineering for smart manufacturing with Kyungnam University.

...