

Received October 6, 2021, accepted October 18, 2021, date of publication October 22, 2021, date of current version October 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3122175

# An Asset Administration Shell Method for Data Exchange Between Manufacturing Software Applications

XUN YE<sup>1</sup>, MENG MENG YU<sup>1</sup>, (Member, IEEE), WON SEOK SONG<sup>2</sup>,  
AND SEUNG HO HONG<sup>1</sup>, (Senior Member, IEEE)

<sup>1</sup>Department of Electronic Engineering, Hanyang University, Ansan 15588, South Korea

<sup>2</sup>Nestfield Company Ltd., Gyeonggi Technopark, Ansan 15588, South Korea

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

This work was supported by the National Research Foundation of Korea through the International Cooperation Program under Grant NRF-2018K1A3A1A61026320.

**ABSTRACT** The multifaceted industrial application of cyber-physical systems (CPSs) has increased, enabling tight interactions between physical and computational elements. Industry 4.0 aims to promote industrial application of CPSs. The asset administration shell (AAS) represents the practical embodiment of the CPS, and can be obtained by integrating information and communication technologies. Most prior studies concentrated on AASs for hardware assets in the manufacturing domain; they seldom considered AASs for software applications. To fill this gap, we present a method for implementing AASs for software, i.e., a manufacturing execution system (MES) combined with an enterprise resource planning (ERP) system. This method allows runtime data exchange between an MES and ERP; this is better than the conventional file-based method. Finally, a data exchange case (a manufacturing order) is presented to validate our AAS method.

**INDEX TERMS** Asset administration shell, cyber-physical system, industry 4.0, data exchange, automationML, OPC UA.

## I. INTRODUCTION

The Internet of Things (IoT) and cyber-physical systems (CPSs) have emerged as key technologies meeting various user and application requirements of the era of the fourth industrial revolution, also termed Industry 4.0 [1], [2]. In Industry 4.0, digitalization is frequently considered essential to facilitate information (data) exchange throughout the production lifecycle [3]. Given the emergence of CPSs and applications thereof in various domains (manufacturing, energy, and logistics) [4], a new opportunity has arisen for the development of digital solutions, i.e., digital twins (DTs). The asset administration shell (AAS) [5], as a digital representation of industrial components, fulfils the requirement for digital transformation. AASs can be used to implement DTs [6] and support certain key DT functionalities (e.g., physical object encapsulation, data gathering and communication, collaboration and negotiation, reconfiguration, and intelligent

decision support) [7]. Therefore, AASs are expected to transform industries and businesses into large, digitally networked and adaptable knowledge-based enterprises.

As defined by the International Electrotechnical Commission (IEC) Technical Committee 65 Working Group 24, an AAS is a standardized digital representation of an asset with industrial applications, allowing uniform access to information and functions and thereby facilitating interoperability among the applications of a manufacturing company [8]. As shown in Figure 1, an AAS helps transform raw data from physical assets into digitally comprehensible information for stakeholders (domain experts and system developers). An asset with a proprietary communication interface (e.g., PROFIBUS) is linked to a uniform application programming interface (API), through which the asset data can be accessed in the form of an AAS; AASs work together to capture key information pertaining to assets (i.e., their intrinsic properties, operational parameters, and technical functionalities), enabling straightforward interaction over standardized communication interfaces with other Industry 4.0 components.

The associate editor coordinating the review of this manuscript and approving it for publication was Stefano Scanzio<sup>1</sup>.

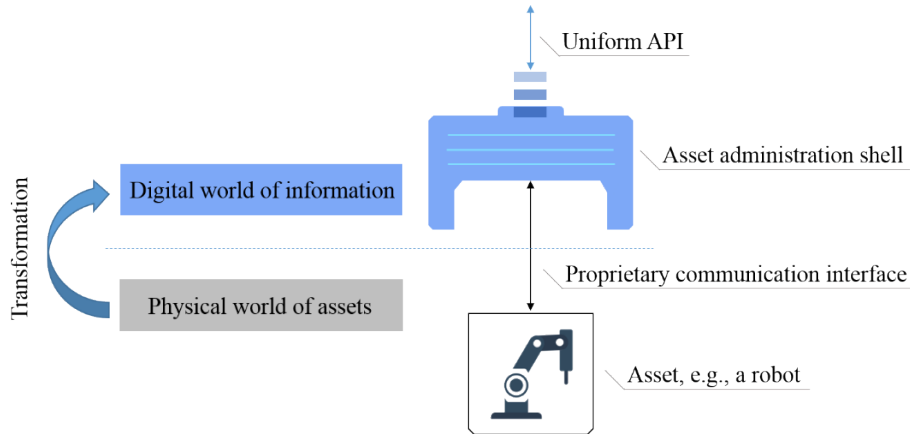


FIGURE 1. Role played by the asset administration shell in digitalization.

In the context of Industry 4.0, assets are manufacturing components that can be in the form of hardware (e.g., sensors and machines) or software (e.g., applications and programs). Most previous publications on AASs focused exclusively on hardware assets [9]–[11]. For example, the authors of [12] and [13] developed AASs for hardware assets in the field layer of the automation pyramid, including robots and turntables. As shown in Figure 2 [14], the present challenge is AAS implementation of high-level software assets [e.g., manufacturing execution systems (MESs) and enterprise resource planning systems (ERPs)].

Currently, few researchers focus on AAS applications for software. The authors of [15] developed an AAS model enabling the integration of two different software tools, while the authors of [16] mapped the proprietary “ABB Ability” information model to a standard AAS information model. The authors of [17] developed a data administration shell based on AAS to manage cross-enterprise engineering datasets. However, no study has focused on practical data exchange between different software applications (e.g., an MES and ERP). Conventional data exchange between these systems often uses file-based methods, which is time-consuming. Furthermore, existing studies neglect that the AAS can accelerate the data exchange for software. Here, we develop AASs for high-level MES and ERP systems; the aim was to replace file-based static data with AAS-based runtime data. Given an AAS interface, the operation data can be seamlessly transferred from the field to the enterprise level during runtime. Our goal is to interconnect the entire automation pyramid, allowing information exchange between uniform AAS interfaces (Figure 2). To this end, the main contributions of this paper are:

- First, we present an AAS-based method for data exchange between software assets for the first time.
- Second, we developed two independent AAS models for both MES and ERP data.
- Third, we also present AAS interfaces for an MES and ERP; these allow conversion of static file data into AAS data, and then communicate the latter data.

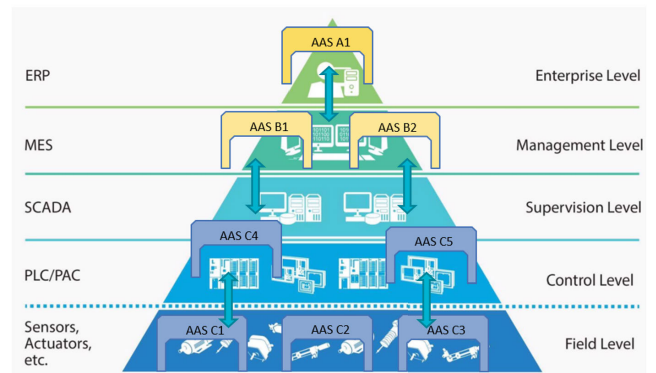


FIGURE 2. An automation pyramid with AAS interfaces (adapted from [14]).

- Fourth, we present a test case of data exchange between MES and ERP tools during execution of a manufacturing order; this validates the feasibility of our AAS data exchange method. The results show that our AAS effectively eliminates data siloes between different software systems.

The rest of the paper is organized as follows. Section 2 introduces data exchange models meeting existing industrial standards for enterprise and control systems. Section 3 introduces our AAS method for data exchange between MES and ERP systems; two research questions are addressed. Section 4 describes the test case and shows the data exchange results. Section 5 provides the conclusion and describes our plans for future work.

## II. DATA EXCHANGE MODELS FOR ENTERPRISE AND CONTROL SYSTEMS

In a fully integrated system, business and manufacturing functions are aligned for efficient data flow. In terms of data exchange between MES and ERP systems, the IEC 62264 series of standards is based on the data standard of the International Society of Automation (ISA), i.e., ISA-95 [18]. IEC 62264 describes the structures of common object models

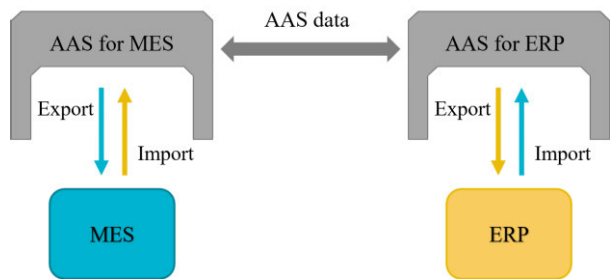


FIGURE 3. An AAS method for data exchange between an MES and an ERP.

that can be used to control enterprise systems. Specifically, part 2 of IEC 62264 (IEC 62264-2) defines the attributes of those models. There are five types of resources: personnel, equipment, physical assets, materials, and process segments; all of these are involved in generic interface content exchange between manufacturing control functions and other enterprise functions [19]. IEC 62264 is widely used in many industries to reduce the risks, costs, and errors associated with implementation of MES and ERP interfaces.

Several works have aligned IEC 62264 with other relevant standards. Automation Markup Language (AutomationML) is an XML-based standard used to model production environments, i.e., products, processes, and resources [20]. IEC 62264-2 and AutomationML can operate together, although they view the same production system differently. The former is slightly biased toward the upper levels of the automation hierarchy, and the latter toward the lower levels [21]. Recently, AutomationML has been aligned with IEC 62264-2; IEC 62264-2 information can be accepted by AutomationML models. This alignment gave rise to a public specification, ‘Provisioning for MES and ERP’ [22], which specifies the rules for modeling IEC 62264-2 elements within AutomationML documents. Also, a list of role class libraries that classify the five types of resources defined in IEC 62264-2 is presented.

OPC Unified Architecture (UA) is a platform-independent, service-oriented, machine-to-machine communication standard for data exchange among industrial automation devices [23]. The OPC Foundation has released a companion specification for ISA-95 [24], which defines an information model that conforms to the common object models specified in ISA-95. The ISA-95 standard sets out how to describe the flow of information between manufacturing operations management (MOM) and ERP systems. Mapping of ISA-95 to OPC UA allows secure information flow between the MES and ERP levels; this can also be extended downward to lower levels of the automation hierarchy.

Both the AutomationML and OPC UA methods for implementation of IEC 62264 and ISA-95 complement existing business to manufacturing markup language (B2MML) implementations [25], by providing a reliable data exchange environment widely accepted in the manufacturing industry. The difference is that the AutomationML method focuses principally on information modeling, whereas the OPC UA

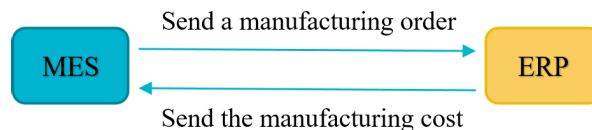


FIGURE 4. Test case of data exchange for a manufacturing order.

Manufacturing Orders / WH/MO/00004	
Product	example_bearing
Quantity	0.00 / 10.00 To Produce
Bill of Material	steel_ball
Scheduled Date	06/01/2021 10:22:23
Responsible	Xun Ye
State	Confirmed

FIGURE 5. Example manufacturing order using odoo.

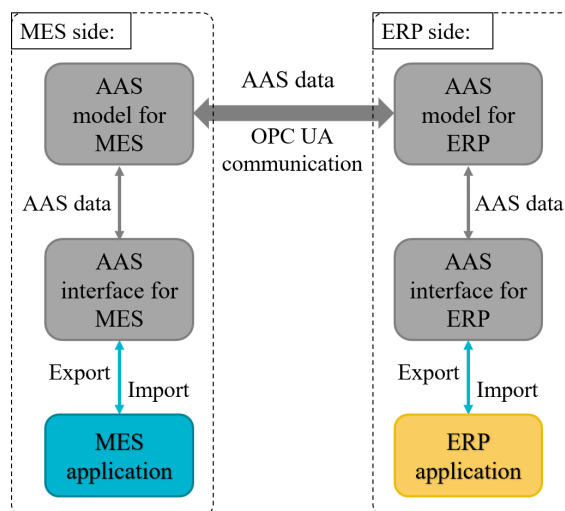


FIGURE 6. Implementation of an AAS method for an MES and ERP.

method is mainly concerned with interfaces for data communication.

### III. AN AAS METHOD FOR MES AND ERP DATA EXCHANGE

Conventional data exchange between MES and ERP systems usually employs file-based methods, including the use of online or offline comma-separated value (CSV) files, which can be in Microsoft Excel format. When working online, a file can be digitally exchanged between an MES and ERP in a direct (from end to end) or indirect (via an intermediate data hub) manner. When working offline, the file is transferred from one software engineer to another. As the data exchange process is both static and discontinuous, it is also time-consuming and error-prone. Data exchange during runtime is not assured. Moreover, MES and ERP systems must possess

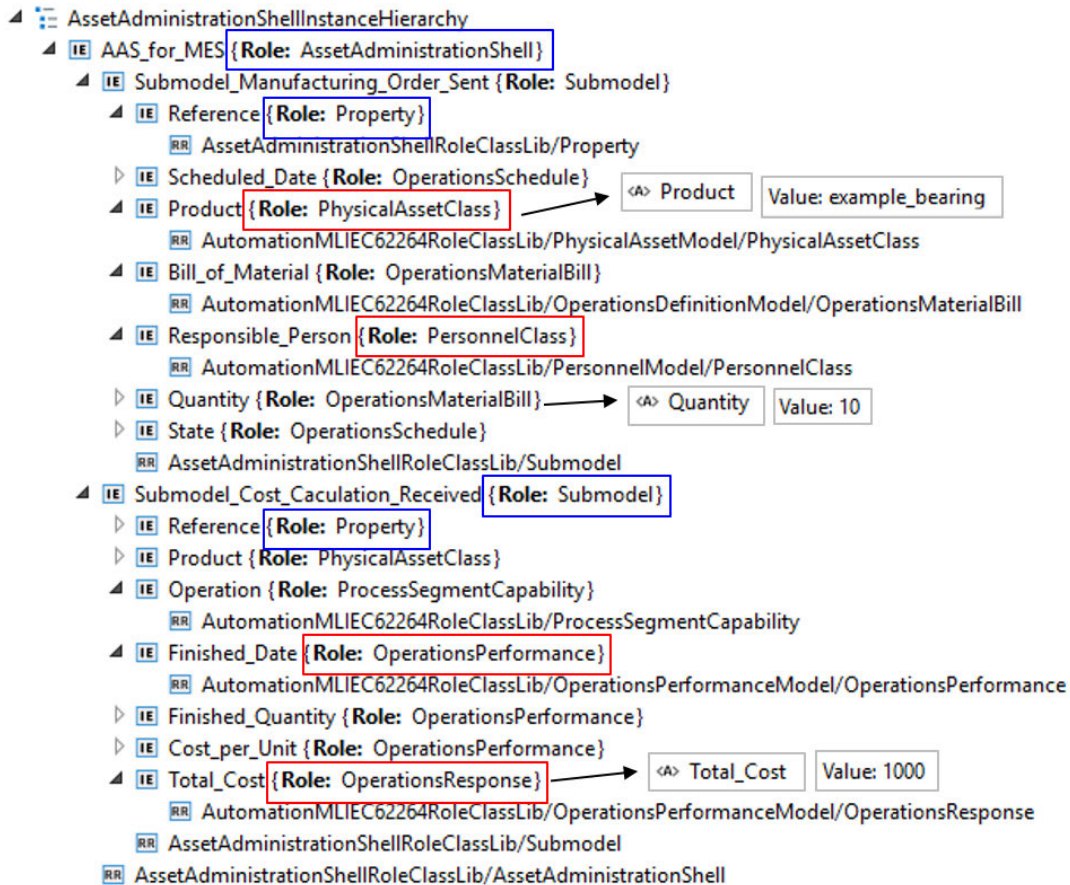


FIGURE 7. AutomationML-based AAS model for the MES.

dedicated file-based interfaces, which demands significant system development. A new digitalized method proposed supporting runtime data exchange between an MES and ERP (i.e., when manufacturing is underway) is urgently required. An AAS divides runtime data into submodels and uses inbuilt interfaces to transfer the data. Therefore, we developed an AAS-based method for runtime data exchange between an MES and ERP (Figure 3). We devised two independent AASs that link to both the MES and ERP. The AAS has two functions: to store MES or ERP data, and exchange them during runtime. We had two research questions:

1) *How can MES and ERP data be converted into AAS format?*

All information and data generated during production should be managed in a structured way. The AAS submodels gather data that belong together. Various MES and ERP data can be modeled into different AAS submodels. This requires a management mechanism for representing MES and ERP data. An implementation can be mapped to existing data modeling methods based on XML or JSON.

2) *How are the MES and ERP data in the AAS communicated?*

The core is to devise end-to-end runtime data exchange interfaces for the MES and ERP. The data communication

method must be compatible with the data management mechanism developed when answering the first research question. In other words, the MES and ERP data should first be represented in the AAS model. Then, the chosen communication method must be able to access the AAS model and communicate the data accordingly. Implementation could involve request/response- or client/server-based communication, such as OPC UA.

#### IV. TEST CASE: DATA EXCHANGE SCENARIO FOR A MANUFACTURING ORDER

This section describes a data exchange scenario between a MES tool and an ERP tool, which features the data exchange for a manufacturing order. This test case shows that the two research questions can be answered with detailed technology implementations.

##### A. DATA EXCHANGE SCENARIO

A data exchange scenario between ERP and MES when a manufacturing order is placed is shown in Figure 4. This scenario includes two data exchange steps. First, the MES sends the ERP a manufacturing order; then, the ERP calculates the manufacturing cost and sends it back to the MES.



FIGURE 8. AutomationML-based OPC UA information models for the MES and ERP.

Both MES and ERP focus on data management; the difference is that the MES is concerned with production, whereas the role of the ERP is to facilitate business and resource management. Microsoft Excel can be used to organize and analyze data, and can thus serve as either the MES or ERP. In this case, we assigned Excel to the ERP and used an online data management tool (Odoo [26]) for the MES. A sample manufacturing order was created (Figure 5). The order includes a reference number, the product name, the quantity required, a list of materials, the schedule, the responsible person, and the order status. The document for the manufacturing cost calculated by the ERP includes the calculation number, product name, operation, finish date, final quantity, cost per unit, and total cost.

**B. IMPLEMENTATION**

The key elements to implement the AAS method and their interconnections are shown in Figure 6. The data exchange process can be described as follows. The MES first exports the manufacturing order to a CSV file, which is converted into an AAS model via the AAS interface. This model is then mapped to the AAS model of the ERP to enable runtime data exchange. Finally, the AAS interface of the ERP stores the exchanged data as a CSV file that will be imported

```

Data from MES_application are:
parsing the Reference is: WH/MO/00004
parsing the Scheduled Date is 6/1/2021 10:22
parsing the Product is: example_bearing
parsing the Bill_of_Material is: steel_ball
parsing the Responsible is: Xun Ye
parsing the Quantity is: 10
parsing the State is: Confirmed

Data from AAS_model_MES are:
reading the Calculation Number is: CN00312
reading the Product is: example_bearing
reading the Operation is: Turning and Polishing
reading the Finished Date is: 6/3/2021 11:01
reading the Finished Quantity is: 10
reading the Cost_per_Unit is: 100
reading the Total Cost is: 1000
    
```

FIGURE 9. Outputs of the AAS interface for the MES.

into the ERP. The role of the AAS interface for the ERP is same as that of the interface for the MES. The same process applies during feedback data exchange from the ERP to the MES.

The development of the AAS model and the AAS interface are described below.

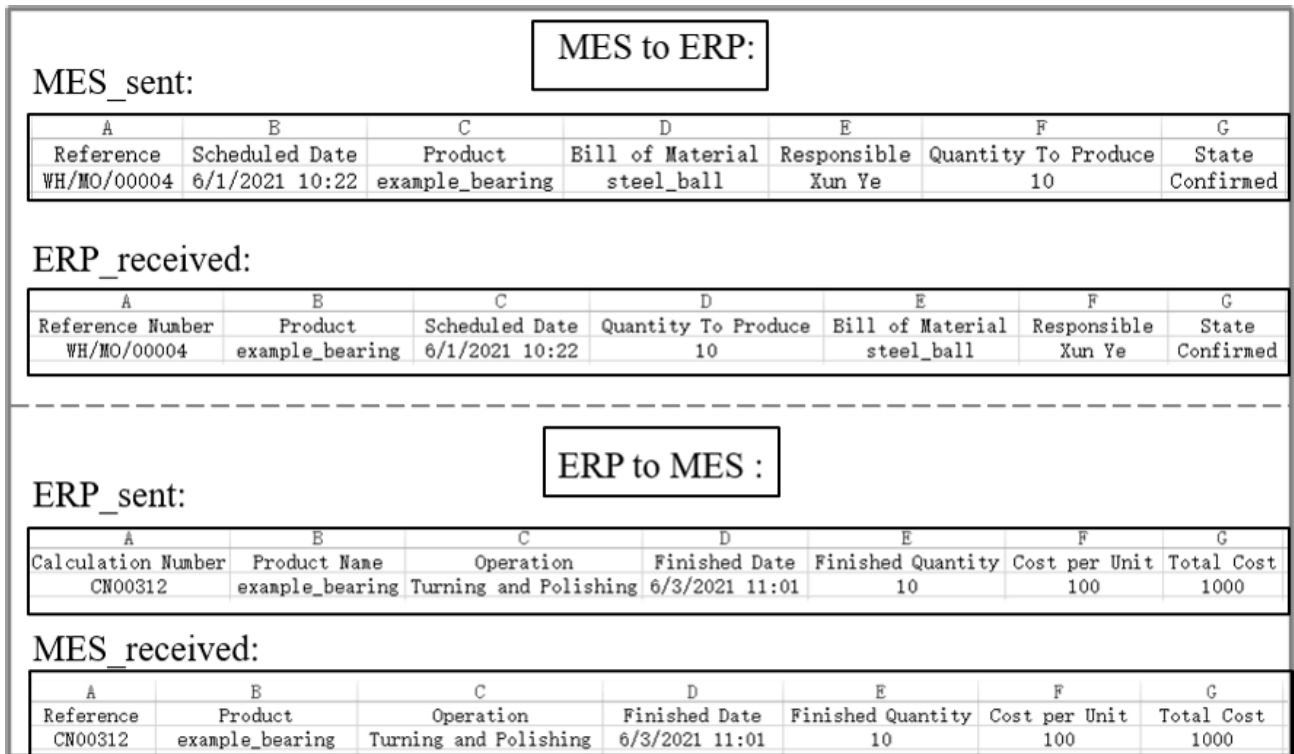


FIGURE 10. Two set of data exchange results between the MES and ERP.

1) AUTOMATIONML FOR AAS DATA MODELING

We developed AAS models for MES and ERP using the method of AutomationML for IEC 62264, because AutomationML provides a visual editing tool for modeling the AAS information. Based on the specification of AutomationML for IEC 62264, we implemented the AutomationML model for MES and ERP respectively. Each AutomationML model describes the AAS data that need to be communicated. The AutomationML-based AAS models of the MES and ERP should be matched to ensure consistent data exchange. Here, the AAS data for MES are shown in Figure 7 as an example.

Figure 7 shows that each AAS contains two submodels, i.e., the manufacturing order and cost calculation. The former submodel stores the data associated with manufacturing execution and the latter calculates the manufacturing cost. Each submodel element was modelled as an internal element (IE) in AutomationML, and every element had a relevant role requirement (RR), thus complying with the IEC 62264, e.g., physical asset class and personnel class (red box in Figure 7). The RR indicates the role an element fulfils and realizes the mapping between IEC 62264 and AutomationML. We also adopted the role class from the specification of AutomationML for AAS representation [27], e.g., submodel class and property class (blue box in Figure 7). These standardized classes allow mapping between the AAS model and AutomationML elements. The property data of the AAS are stored as attributes (grey box in Figure 7) of the AutomationML model.

2) OPC UA FOR AAS DATA COMMUNICATION

After representing the AAS information using AutomationML, the next challenge is to exchange and communicate the AAS data. OPC UA can be used because it provides interfaces for data exchange between industrial devices and systems [28]. OPC UA has a typical client-server communication scheme that can be adopted to become an AAS communication interface. OPC UA also has an information model that organizes static and dynamic data into structured information. Therefore, we integrated the AutomationML-based AAS model with OPC UA to enable data exchange.

Specifically, the two AutomationML models for the MES and ERP were integrated with respective OPC UA applications. Figure 8 shows the integration results accessed by an OPC UA software (UA Expert). The OPC UA properties of ‘bill of material’ and ‘cost per unit’ are displayed as examples. The two OPC UA applications are mutually mapped; the OPC UA information model for the MES [Figure 8(a)] is also matched with the AAS model for the MES in Figure 7. Figure 8(b) shows the OPC UA information model for ERP. Integration of AutomationML with OPC UA used an online conversion tool (AutomationML-OPC UA Converter) [29] that generates an OPC UA server from an existing AutomationML model. Data exchange between the two OPC UA applications proceeded via an OPC UA client. In our case, the client was embedded in the OPC UA server of the MES, and “subscribed” to the data from the OPC UA server of the

ERP. This allows runtime data mapping between the two AAS models for the MES and ERP, as described in Figure 6.

The AAS interfaces of the MES and ERP were implemented as OPC UA clients based on the open source ‘open62541’ toolkit [30]. Figure 9 shows the outputs of the AAS interface of the MES, as an example. The interface has two main functions: to connect to the MES application and then convert the CSV file to runtime AAS data (Function a); and to interact with its AAS model (i.e., the OPC UA information model) (Function b).

### C. DATA EXCHANGE RESULTS

According to data flows in Figure 6, the communicated AAS data should be exported as CSV files to allow access by the ERP or MES. Data exchange results are four CSV files generated by AAS interfaces, as shown in Figure 10. The sent/received MES data are identical to the received/sent ERP data. This proves the feasibility of the proposed AAS method for runtime data exchange between MES and ERP.

### V. CONCLUSION AND FUTURE WORK

To realize the goal of digitalization for Industry 4.0, the CPS is regarded as an emergent approach that enables tight interactions between cyber applications and physical devices. The AAS is a practical embodiment of the CPS and can be realized with the integration of information and communication technologies. With a uniform AAS interface, all the manufacturing components including hardware devices and software applications can be interconnected. However, prior research focused principally on the implementation of AASs for hardware manufacturing assets. Thus, we concentrated on AASs for software assets, and developed an AAS method allowing runtime data exchange between MES and ERP applications for the first time. We used a test case to validate our AAS approach, and developed AAS models for an MES and ERP using the AutomationML technology, IEC 62264 standard, and OPC UA communication protocol. We also developed AAS interfaces for data conversion.

A limitation of this work is that we used simple existing data management tools as MES and ERP; we did not spend efforts developing the tool itself. Although the test case is simple, it proves the effectiveness of our AAS method. In future, the proposed AAS method could be applied to the enterprise-level MES and ERP software in real industrial cases. Other approaches for implementing the IEC 62264, e.g., OPC UA for ISA-95 will also be explored.

### REFERENCES

- [1] G. Manogaran, G. Srivastava, B. A. Muthu, S. Baskar, P. M. Shakeel, C.-H. Hsu, A. K. Bashir, and P. M. Kumar, “A response-aware traffic offloading scheme using regression machine learning for user-centric large-scale Internet of Things,” *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3360–3368, Mar. 2021, doi: [10.1109/JIOT.2020.3022322](https://doi.org/10.1109/JIOT.2020.3022322).
- [2] Y. Gong, L. Zhang, R. Liu, K. Yu, and G. Srivastava, “Nonlinear MIMO for industrial Internet of Things in cyber–physical systems,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5533–5541, Aug. 2021, doi: [10.1109/TII.2020.3024631](https://doi.org/10.1109/TII.2020.3024631).
- [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] A. W. Colombo, S. Karnouskos, O. Kaynak, Y. Shi, and S. Yin, “Industrial cyberphysical systems: A backbone of the fourth industrial revolution,” *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 6–16, Mar. 2017, doi: [10.1109/MIE.2017.2648857](https://doi.org/10.1109/MIE.2017.2648857).
- [5] *Smart Manufacturing-Reference Architecture Model Industry 4.0 (RAMI4.0)*, IEC Standard PAS 63088, 2017.
- [6] 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.
- [7] *Digital Twin and Asset Administration Shell Concepts and Application in the Industrial Internet and Industrie 4.0*, Industrial Internet Consortium, Norfolk, VI, USA, Sep. 2020. [Online]. Available: [https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/Digital-Twin-and-Asset-Administration-Shell-Concepts.pdf?\\_\\_blob=publicationFile&v=7](https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/Digital-Twin-and-Asset-Administration-Shell-Concepts.pdf?__blob=publicationFile&v=7)
- [8] *Industrial-Process Measurement, Control and Automation-Asset Administration Shell for Industrial Applications—Part 1: Administration Shell Structure*, IEC Standard CD 63278-1, 2020.
- [9] A. Luder, M. Schleipen, N. Schmidt, J. Pfrommer, and R. Henssen, “One step towards an Industry 4.0 component,” in *Proc. 13th IEEE Conf. Autom. Sci. Eng. (CASE)*, Aug. 2017, pp. 1268–1273.
- [10] M. Wenger, A. Zoitl, and T. Müller, “Connecting PLCs with their asset administration shell for automatic device configuration,” in *Proc. IEEE 16th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2018, pp. 74–79.
- [11] S. Cavalieri and M. G. Salafia, “Asset administration shell for PLC representation based on IEC 61131–3,” *IEEE Access*, vol. 8, pp. 142606–142621, 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] 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).
- [14] *Manufacturing Execution Systems in Industry 4.0*. Accessed: Oct. 2021. [Online]. Available: <http://www.miac-automation.com/mes-oe-track-and-trace/>
- [15] A. Deuter and S. Imort, “PLM/ALM integration with the asset administration shell,” *Proc. Manuf.*, vol. 52, pp. 234–240, Jan. 2020.
- [16] 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.
- [17] 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.
- [18] *Enterprise-Control System Integration, ANSI/ISA-95*, Int. Soc. Automat., Durham, NC, USA, 2010.
- [19] *Enterprise-Control System Integration—Part 2: Objects and Attributes for Enterprise-Control System Integration*, IEC Standard 62264-2, 2015.
- [20] *AutomationML*. Accessed: Oct. 2021. [Online]. Available: <https://www.automationml.org/>
- [21] B. Wally, C. Huemer, A. Mazak, and M. Wimmer, “IEC 62264-2 for AutomationML,” in *Proc. 5th AutomationML User Conf.*, Gothenburg, Schweden, 2018, pp. 1–7.
- [22] *Provisioning for MES and ERP-Support for IEC 62264 and B2MML*, AutomationML, Magdeburg, Germany, 2021.
- [23] *OPC Foundation*. Accessed: Oct. 2021. [Online]. Available: <https://opcfoundation.org/>
- [24] *OPC Unified Architecture for ISA-95, Release 1.00, OPC 10030*, OPC Found., Scottsdale, AZ, USA, 2013.
- [25] C. Pipero and K. Manjunath, “ISA 95 implementation best practices: Workflow descriptions using B2MML,” in *Proc. WBF North Amer. Conf.*, Atlanta, GA, USA, 2006, pp. 1–13.
- [26] *Odoo*. Accessed: Oct. 2021. [Online]. Available: <https://www.odoo.com/>
- [27] *Asset Administration Shell Representation*, AutomationML, Magdeburg, Germany, 2019.
- [28] X. Ye and S. H. Hong, “An AutomationML/OPC UA-based Industry 4.0 solution for a manufacturing system,” in *Proc. IEEE 23rd Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2018, pp. 543–550.

- [29] *AutomationML-OPC UA Converter*. Accessed: Jul. 2021. [Online]. Available: <https://aml2ua.iosb.fraunhofer.de/servlet/is/11/>
- [30] *Open62541*. Accessed: Jul. 2021. [Online]. Available: <https://open62541.org/>



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



**MENGMENG YU** (Member, IEEE) received the B.S. degree in communication engineering from the Wuhan University of Technology, Wuhan, China, in 2008, the M.S. degree in detection technique and automation equipment from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2011, and the Ph.D. degree in electronic systems engineering from Hanyang University, Ansan, South Korea, in 2015.

She was a Postdoctoral Researcher under the BK21 PLUS Program (BK21+) with Hanyang University, from 2015 to 2018, where she is currently a Research Assistant Professor with the Research Institute of Engineering and Technology. Her research interests include smart grid, game theory, artificial intelligence, and smart manufacturing. She was a recipient of the Outstanding Researcher Award of BK21+ and the Best Paper Award from the Workshop on Smart City Infrastructure and Applications, in 2016.



**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. He has been in charge of developing solutions for the smart industry. He is currently engaged in standard activities, such as IEC TC 65 and IEC TC 3 SC 3D.



**SEUNG HO HONG** (Senior Member, IEEE) received the B.S. degree in mechanical engineering from Yonsei University, Seoul, South Korea, in 1982, the M.S. degree in mechanical engineering from Texas Tech University, Lubbock, in 1985, and the Ph.D. degree in mechanical engineering from Pennsylvania State University, University Park, in 1989. 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. He is currently an Emeritus Professor with the Department of Electronic Engineering, Hanyang University, Ansan, South Korea, and an Executive Advisor/Fellow of Nestfield Company Ltd. His research interests include smart manufacturing, smart grid, the 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.

...