Process Simulation in Industry 4.0: Validation of Asset Administration Shell Communication

Leandra Karolina Batista Leal[1[0000-0002-9920-4016]], Atila Martins Lucas[2[0000-0003-0495-7486]], Isaque Vilson Batista da Costa[1[0000-0003-1025-0356]], Francisco Assis Barros de Oliveira[1[0000-0002-4964-198X]] and Roberto Higino Pereira Da Silva[1[0009-0001-9998-0373]]

[1] Universidade do Estado do Amazonas, Manaus AM 69050-030, BRA
`lkbl.eng18@uea.edu.br, faoliveira@uea.edu.br, rpsilva@uea.edu.br`
[2] Universidade Federal do Amazonas, Manaus AM 69067-005, BRA
`atila.lucas@hotmail.com`

Abstract. Industry 4.0 represents a shift in fundamental beliefs in the Industrial Revolution, driven by technologies like the Internet of Things (IoT), Cyber-Physical Systems (CPS), Cloud Computing, and Big Data. In this context, the Reference Architecture Model for Industry 4.0 (RAMI 4.0) and Asset Administration Shell (AAS) concepts emerge in standardizing integration and communication for developing and deploying intelligent systems. This paper explores how Industry 4.0 concepts, RAMI 4.0 and AAS, are integrated and addresses the gap in effectively validating these concepts before real-world implementation by employing simulation-based approaches, within a virtualized manufacturing environment using Flexsim® and the communication between AAS and two corresponding assets: a collaborative robot and a laser engraver. The research outlines the conceptual model of the manufacturing process, the software architecture development, the creation of an observability dashboard, and the simulation of the virtualized process. While recognizing the limitations inherent in simulation, this research underscores their significance in identifying challenges of the system interaction.

Keywords: Industry 4.0, RAMI 4.0, Asset Administration Shell (AAS), Computer Simulation

1 Introduction

The concept of Industry 4.0 is defined by [1] as: “the fourth industrial revolution applying the principles of cyber-physical systems (CPS), internet and future-oriented technologies and smart systems with enhanced human-machine interaction paradigms”—industry 4.0 fosters interconnected systems and intelligence across the entire value chain, developing industrial production processes.

A reference architecture serves as a model representing real-world concepts [2]. Equipped with rules, it aims to accurately mirror the physical world within the realm of information technology. Therefore, reference architecture provides solutions that assist companies in establishing their entire production based on an agreed standard architecture [3]. In this scenario, the concepts of the Reference Architecture Model for Industry 4.0 (RAMI 4.0) and the Asset Administration Shell (AAS) can be applied to meet the requirements of intelligent systems and assure interoperability between the applications and manufacturing.

RAMI 4.0 has the purpose of establishing a shared understanding of the standards, models, and use cases for the development of smart manufacturing stations, setting guidelines for integrating processes [4]. The AAS, on the other hand, standardizes communication and data exchange between different components and industrial systems, facilitating interoperability and integration across systems.

The research problem centers on the need to assess communication and integration between AAS and their assets, through a simulation-based approach using Flexsim®, a discrete event simulation software, before its deployment in a real robotic cell located in a laboratory setting.

The study establishes a theoretical foundation by discussing concepts such as RAMI 4.0, AAS, and previous studies that utilized simulation with AAS. Following, it outlines the development of the software architecture, the creation of an observability dashboard, and the virtualization of a smart assembly cell. Finally, it explains the outcomes of the integration between the AAS and the simulated environment.

2 Theoretical Background

Industry 4.0 is an umbrella term encompassing the technologies that enable smart production. [5] and [6] present these technologies, which encompass big data, cloud computing, and manufacturing, Internet of Things (IoT), Cyber-Physical Systems (CPS), and smart factories, as integral elements of the Industry 4.0 concept. The RAMI 4.0 architecture integrates key Industry 4.0 elements into a three-dimensional layered model (see Fig. 1).

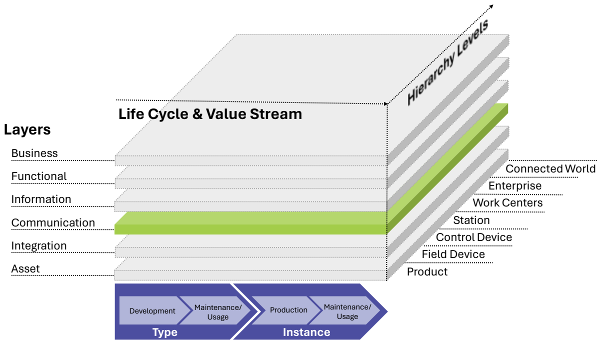


Fig. 1. RAMI 4.0. Adapted from [2].

As described by [7] and [8], each axis presented symbolizes a distinct domain or perspective within the scope of Industry 4.0. The horizontal axis indicates the product life-cycle management, adhering to the International Electrotechnical Commission (IEC) 62890 standards. The rightmost segment represents functional hierarchy according to the IEC 62264 standard, including elements such as 'Connected World,' 'Field Device,' and 'Product'. The layers in the vertical axis represent different perspectives, such as data maps, functional descriptions, communication behavior, hardware/assets, or business processes.

One of the fundamental concepts of RAMI 4.0 is the industry 4.0 components, characterized by their composition comprising a tangible entity (such as a machine, field device, factory, or software) and an Asset Administration Shell (AAS) [9]. According to the International Electrotechnical Commission (IEC), an AAS serves as a standardized digital interpretation of an asset. This portrayal ensures consistent access to information and functionalities, thereby enhancing interoperability among various applications [10]. The successful deployment of an Industry 4.0 manufacturing system centers on the accurate implementation of an AAS [11], in other words, adherence to a set of implementation standards is essential for the information models, including structured data and services.

[12] highlights the integration of Administration Shells (AASs) into a virtual assembly line for smart manufacturing, emphasizing adherence to standards. Meanwhile, [13] offers a case study illustrating a simulated production system to evaluate the utilization of an asset management node based on AAS. Both cases focus on the application of AAS in virtualized lines. Applying virtual preparation before the implementation and integration of new production systems and technologies offers numerous advantages [14], one significant benefit is the ability to test and verify solutions virtually before actual commissioning [15].

3 Method

This chapter explores the conceptualization of the manufacturing procedure, the development of a software architecture tailored to enhance communication between different components involved in the manufacturing process, the designing of a real-time observability dashboard, and the virtualization of the manufacturing process.

3.1 Conceptual Model

The case study is based on a production cell of an automated manufacturing segment, focusing on a prototype named 'Customized LED Decoration.' This prototype consists of two cases (upper and lower parts), an acrylic display featuring the project mark, and a printed circuit board assembly (PCBA) for LED light generation. The PCBA, acrylic displays, and cases constitute prefabricated production inputs. Additionally, a 3D printing machine produces the cases in two colors: white and gray. Within the assembly cell, there is a robot, a laser engraver, a color sensor, and magazines for housing the PCBAs and cases.

The assembly procedure is initiated upon the customer's personalized order, succeeding the sequence: the robot loads the upper case, verifies its color via a sensor, activates the laser engraving process on the case, and assembles the part. Then, it loads the PCBA, reads its barcode, and assembles. Lastly, it loads the lowercase, verifies its color, and assembles. Afterward, the robot transfers the final product to an area for visual inspection, acrylic display placement, and functional tests, performed by an operator (see Fig. 2).

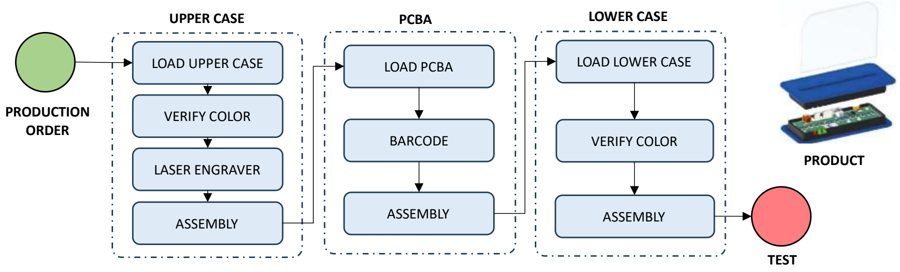


Fig. 2. Process flowchart.

The proposed system covers product customization as a strategy to match individual customer preferences, this enables users to select colors, text engraving, and quantity, via a mobile application. The cell is also capable of supporting flexible production cycles (1 to 12 products), as it automatically adapts the equipment to the manufacturing operations required by the product variant or specifications. The robot has the responsibility of receiving and executing instructions while maintaining continuous communication with the system through cloud computing. The case study utilizes Flexsim® communication with the AASs to demonstrate the product’s manufacturing process, including variations in colors and engraved text.

3.2 Software Architecture and AAS Models

The software architecture framework contains the factory operation flow, from the purchase order to the final product, including its essential functionalities: streamlined systems designed to replicate an Enterprise Resources Planning (ERP) and Manufacturing Execution System (MES), an AAS Robot and AAS Laser Engraver (see Fig. 3). The decision to incorporate the AAS robot and laser engraver was driven by their potential to be virtualized, based on the components of the robotic cell.

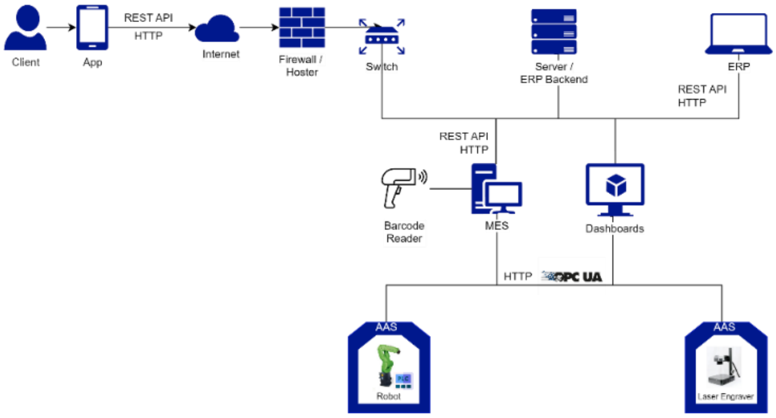


Fig. 3. Software architecture.

The process starts with the customers purchasing orders through an Android application, which offers a range of functionalities, including access login, product selection, customization options, order placement, and tracking of completed purchases. Therefore, ERP receives requests, and after approval, forwards them to MES, who oversees the management of the robotic cell. Acting as an intermediary, the MES facilitates interaction between assets and AASs, translating and routing messages between physical and virtual devices (see Fig. 4).

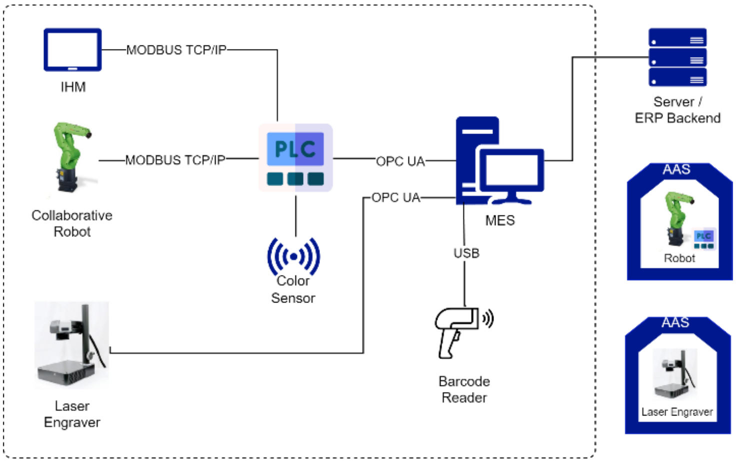


Fig. 4. Robotic cell communication architecture.

MES communicates with both the AAS Robot and the AAS Laser Engraver via an OPC-UA Client, facilitating the exchange of information. This allows the MES to issue production management commands, including initiating production, specifying product quantities, requesting engraving for the laser engraver, and notifying the robot to proceed with the manufacturing process upon completion of engraving. After receiving the production order and its specifications, the AAS robot and the AAS laser engraver begin their respective tasks. Both AAS consist of the following submodels, each with specific properties: Identification; Data; Operation; Documentation; Real-time parameters; Energy efficiency.

To integrate the laser engraver, a dedicated Python script serves as an intermediary between the AAS and the asset application. By employing the OPC-UA protocol, this software enables real-time control and monitoring of the equipment. This setup allows the engraver application to receive commands from the AAS and/or transmit relevant data to the project using its methods. To validate the laser engraver AAS functionality, the operation submodel incorporates the Laser request parameter with values of 'START' and 'DONE'. In this case, Flexsim® simulates the laser engraver and communicates with the Python script via OPC-UA.

For robot integration, the AAS communicates directly with the Programmable Logic Controller (PLC) via the OPC-UA protocol. The PLC, in return, manages communication with the robot, ensuring the robot's control and execution of tasks. Flexsim® is utilized to simulate both the PLC and robot movements, as well as the OPC-UA protocol. In the AAS Robot, the submodel operation that has the state parameter receives the PLC values presented in Table 1.

Table 1. States of Collaborative Robot

#	Referent State
11-12	Removing the upper case from the magazine
13	Upper case wrong color
15	Checking the color of the upper case
21	Engraving text on the upper case
31	Placing the upper case on the assembly jig
41	Removing the PCBA from the magazine
42	Reading PCBA barcode
51	Placing the PCBA on the assembly jig
61-62	Removing the lowercase from the magazine
63	Lowercase wrong color
66	Checking the color of the lowercase
64	Scanning the barcode of the lower case
71	Fitting the lower case into the upper case
72	Taking the assembled product and removing it from the jig.

3.3 Observability Dashboard

Observability is the ability to understand what is happening inside a system by observing its external outputs, such as logs, metrics, traces, or other signals. In the context of the Industry 4.0 paradigm, observability becomes a key component for the continuous improvement of production processes.

The observability dashboard provides a comprehensive and real-time perspective of the simulated production process. The MES assists with the real-time acquisition of data, overseeing general operations and the AAS. These data streams are integrated into InfluxDB, a time series database, for storage.

Moreover, the integration with Grafana, an observability platform, enables the visualization of these data through dynamic and interactive displays, thereby improving data comprehension. The dashboard for the robot and laser engraver incorporates key performance indicators such as average assembly time and engraving time, essential for planning and analysis of the product assembly process.

3.4 Process Virtualization

The process virtualized in Flexsim® simulates the system and its communication with AAS Robot and AAS Laser Engraver. The model development utilizes Flexsim's tools, featuring three-dimensional representations of system components, and programmed logic to replicate the process and emulation of the OPC-UA. Validation and testing procedures were employed to verify the model fidelity in portraying the actual system and the AAS communication within the simulated environment.

The simulation process required incorporating the cell layout (see Fig. 5) into the simulator to facilitate the establishment of predefined positions for the cases and PCBAs magazines, laser engraver, and the simulation of the robot's movements.

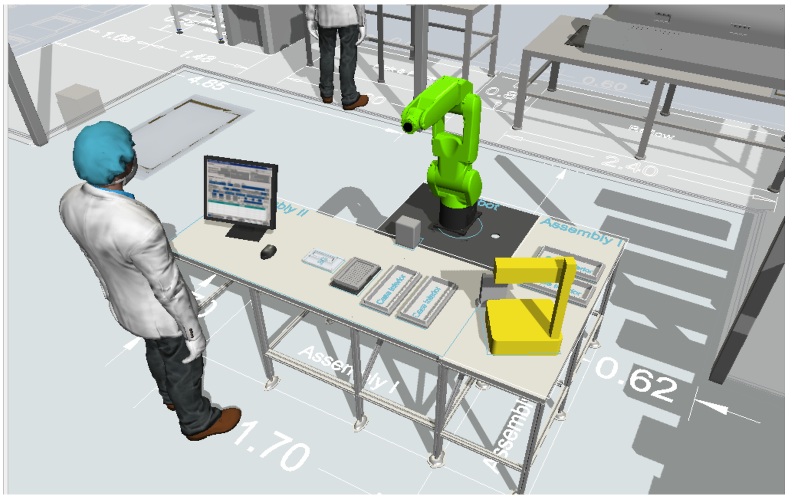


Fig. 5. Robotic cell virtualization.

The existing robot in the software was adapted to align with the collaborative robot's specifications for integration into the cell, achieved by modifying 3D joints using components created on Inventor software (see Fig. 6). Each action, as per the process flowchart, is allocated as an individual motion path with specific path positions relating to each of the robot's six axes (see Fig. 7).

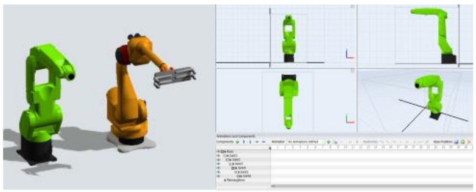
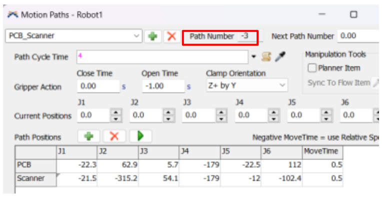


Fig. 6. Robot adaptation.



Motion Paths - Robot1

PCB_Scanner
Path Number -3
Next Path Number 0.00
Path Cycle Time 4
Manipulation Tools
☐ Planner Item
Sync To Flow Item
Gripper Action: Close Time 0.00 s; Open Time -1.00 s; Clamp Orientation Z+ by Y
Current Positions: J1 0.0; J2 0.0; J3 0.0; J4 0.0; J5 0.0; J6 0.0
Path Positions
Negative MoveTime = use Relative Sp

	J1	J2	J3	J4	J5	J6	MoveTime
PCB	-22.3	62.9	5.7	-179	-22.5	112	0.5
Scanner	-21.5	-315.2	54.1	-179	-12	-102.4	0.5

Fig. 7. Example of Motion path created.

A Path is a set of movements determined by a sequence of robot positions. The Robot stores these positions in a tabular format, with each row representing the six joint angles constituting the given position. When executing this path, the Robot moves from its current position to Row 1, followed by Row 2. Each motion path is associated with a unique number, incorporated into the model logic. Following the completion of virtualization, simulated motion tests were performed by clicking on the start button in the motion path panel for each of the six axes, to validate the robot's movements aligned with the expectations within a real 'Customized LED Decoration' assembly scenario.

The implementation of logic is within the process flow tool, designed to build custom logic for simulation models (see Fig. 8). The process starts with the ‘START PROCESS’ to create the objects in 3D. The robot’s logic included four distinct subflows to depict various activities: assembling the upper-case component (UPPER CASE), assembling the lower-case component (LOWER CASE), integrating PCBA, and transferring for final testing (ASSEMBLY). Likewise, a logic flow was created for the laser engraver to simulate the engraving process duration.

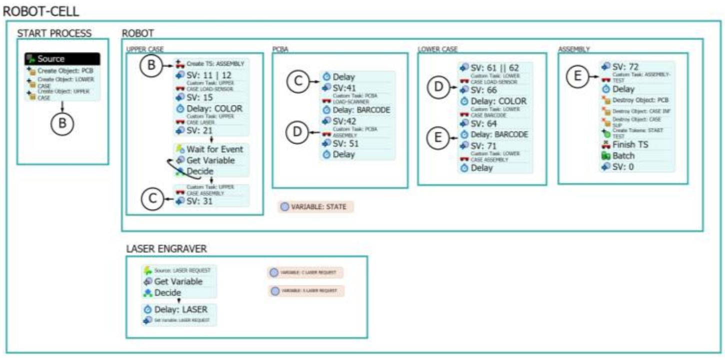


Fig. 8. Computer simulation logic.

The Custom Task activity initiates a task sequence comprising a single task and dispatches it to a task executer. In this case, the task executor is the robot, linked to a specific path movement number in the 'end speed' parameter, determined by the activity in the process flow (Fig. 9). To track the status, an entity called 'variable' was employed, this shared asset stores any kind of data, allowing for subsequent reading or modification of that data. The update of the 'STATE' variable value is upon completion of the movements, whereas the update of the 'LASER REQUEST' variable follows the simulated recording duration (Fig. 10).

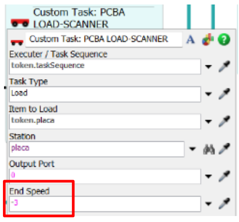

Custom Task: PCBA LOAD-SCANNER
Custom Task: PCBA LOAD-SCANNER
Executer / Task Sequence
token.taskSequence
Task Type
Load
Item to Load
token.placa
Station
placa
Output Port
0
End Speed
-3

Fig. 9. Custom Task parameters example.

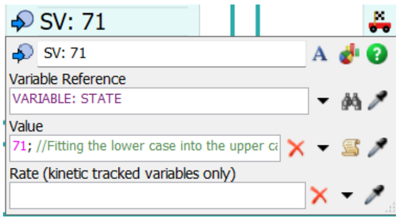
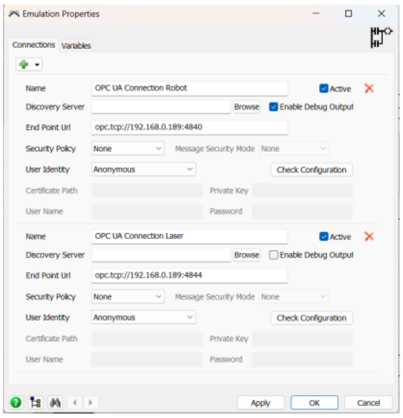

SV: 71
SV: 71
Variable Reference
VARIABLE: STATE
Value
71; //Fitting the lower case into the upper c
Rate (kinetic tracked variables only)

Fig. 10. Set state variable example

After completion of the logic development and simulation execution to verify the movement of the 3D robot upon receiving commands from the process flow, the emulation tool is utilized. This tool establishes connections between the simulator and external servers linked to the AAS. It enables the creation of multiple connections, with variables defined for each. Specifically, for the model, there are two OPC UA connections: one for the AAS Robot and another designated for the AAS Laser Engraver (see Fig. 11).



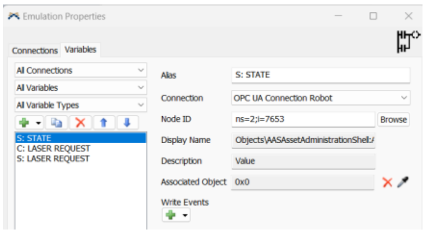
Name: OPC UA Connection Robot ☑ Active
Discovery Server: Browse ☑ Enable Debug Output
End Point Url: opc.tcp://192.168.0.189:4840
Security Policy: None Message Security Mode: None
User Identity: Anonymous Check Configuration
Certificate Path Private Key
User Name Password

Name: OPC UA Connection Laser ☑ Active
Discovery Server: Browse ☐ Enable Debug Output
End Point Url: opc.tcp://192.168.0.189:4844
Security Policy: None Message Security Mode: None
User Identity: Anonymous Check Configuration
Certificate Path Private Key
User Name Password

Fig. 11. OPC UA connection configured.

The integration between the AAS Robot and AAS Laser Engraver occurs when the robot reaches the state 21 - Engraving text on the upper case. At this moment, the MES System stops the robot's movements and transmits the information regarding the name chosen by the client to the AAS Laser Engraver, beforehand modifying the "Laser Request" variable to "START" to initiate the engraving process. Upon completion of the engraving, the asset (laser engraver) communicates with its AAS through the same "Laser Request" variable, altering the value to "DONE". Subsequently, the AAS notifies the MES, then instructs the AAS Robot, and therefore, the robot resumes its programmed movements.

To configure variables, there are two types to the OPC UA connection: sensors for writing data to the server and controls for reading data from the server. To fulfill communications requirements, the server utilizes a sensor variable created for monitoring the "State" and a control and sensor variable for "Laser Request ", which is responsible for both monitoring and changing value as required (see Fig. 12).



S: STATE
C: LASER REQUEST
S: LASER REQUEST

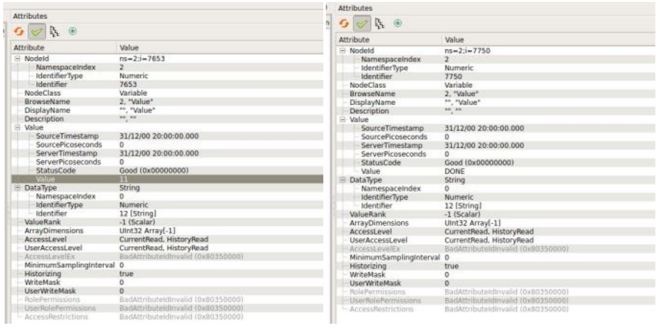
Alias: S: STATE
Connection: OPC UA Connection Robot
Node ID: ns=2;i=7653
Display Name: Objects\AASAssetAdministrationShell:/
Description: Value
Associated Object: 0x0

Fig. 12. OPC UA variables.

Following the configuration of emulation properties, validation tests were performed to determine the proper functionality of communication between Flexsim and the AAS server, including changes in the variable’s values. By verifying the simulated movements, the logic applied in a process flow, and the communication with the Asset Administration Shell (AAS), the simulation successfully accounted for the reception and transmission of messages within the AAS.

4 Results and Discussion

The designed AAS tests were operated on the Flexsim® virtualized robot cell, covering all stages of the manufacturing process. As a product enters the cycle, it requests the production services, with parameters adapted to customer specifications. The tests involved production orders of variable quantities, entering production at separate times, as the simulation considers the AAS's ability to adapt to the changes in production status. The server receives state variables and laser requests from the simulator (Fig 13).



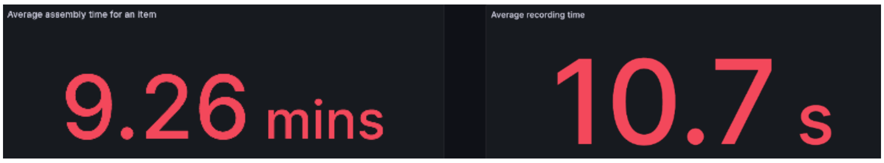
Attribute	Value
NodeId	ns=2;i=7653
NamespaceIndex	2
IdentifierType	Numeric
Identifier	7653
NodeClass	Variable
BrowseName	2, "Value"
DisplayName	"", "Value"
Description	"", ""
Value	
SourceTimestamp	31/12/00 20:00:00.000
SourcePicoseconds	0
ServerTimestamp	31/12/00 20:00:00.000
ServerPicoseconds	0
StatusCode	Good (0x00000000)
Value	11
DataType	String
NamespaceIndex	0
IdentifierType	Numeric
Identifier	12 [String]
ValueRank	-1 (Scalar)
ArrayDimensions	UInt32 Array[-1]
AccessLevel	CurrentRead, HistoryRead
UserAccessLevel	CurrentRead, HistoryRead
AccessLevelEx	BadAttributeIdInvalid (0x80350000)
MinimumSamplingInterval	0
Historizing	true
WriteMask	0
UserWriteMask	0
RolePermissions	BadAttributeIdInvalid (0x80350000)
UserRolePermissions	BadAttributeIdInvalid (0x80350000)
AccessRestrictions	BadAttributeIdInvalid (0x80350000)

Attribute	Value
NodeId	ns=2;i=7750
NamespaceIndex	2
IdentifierType	Numeric
Identifier	7750
NodeClass	Variable
BrowseName	2, "Value"
DisplayName	"", "Value"
Description	"", ""
Value	
SourceTimestamp	31/12/00 20:00:00.000
SourcePicoseconds	0
ServerTimestamp	31/12/00 20:00:00.000
ServerPicoseconds	0
StatusCode	Good (0x00000000)
Value	DONE
DataType	String
NamespaceIndex	0
IdentifierType	Numeric
Identifier	12 [String]
ValueRank	-1 (Scalar)
ArrayDimensions	UInt32 Array[-1]
AccessLevel	CurrentRead, HistoryRead
UserAccessLevel	CurrentRead, HistoryRead
AccessLevelEx	BadAttributeIdInvalid (0x80350000)
MinimumSamplingInterval	0
Historizing	true
WriteMask	0
UserWriteMask	0
RolePermissions	BadAttributeIdInvalid (0x80350000)
UserRolePermissions	BadAttributeIdInvalid (0x80350000)
AccessRestrictions	BadAttributeIdInvalid (0x80350000)

Fig. 13. Variables received by the server, robot (left), and laser engraver (right).

Across all tested scenarios, the products were manufactured without any communication failures, indicating efficient data transmission and information updating among system components. Future failures in the real factory setup are expected to be attributed primarily to the physical hardware of the PLC and laser engraver, given that the AASs underwent thorough simulation validation beforehand. This highlights the utility of simulation-based modeling for identifying potential challenges and optimizing interaction with system components.

Regarding the database within the dashboard, it coordinates the methodical gathering and storage in InfluxDB, of the simulated process data by Flexsim® and AASs, facilitating their visualization through Grafana (see Fig. 14).



Average assembly time for an item

9.26 mins

Average recording time

10.7 s

Fig. 14. Robot cell dashboard.

In terms of discrete event simulation, the result of average time suggests that the manufacturing cycle can meet product requirements under ideal operating conditions. Furthermore, the Flexsim® software has limitations in fully replicating the demands of the real environment due to the absence of latency during the exchange of information. Although the results showed positivity, it's important to note that the simulated environment may not capture all the complexities and variability inherent in a real production environment.

5 Final Considerations

The Reference Architecture Model for Industry 4.0 (RAMI 4.0) and Asset Administration Shell (AAS) present a framework for the understanding and deployment of intelligent manufacturing stations within Industry 4.0. The communication and integration layers facilitate data exchange, bridging the physical and cyber realms and enhancing smart production processes. In summary, this research provides a systematic approach to validate the integration and communication between AAS and their corresponding assets, exemplified by a collaborative robot and laser engraver, within a simulated manufacturing environment in Flexsim®. Further research could involve methods to enhance the fidelity of simulations, replicating the complexities and variability of real production environments to address these limitations and focus on improving communication and control logic between AASs and assets, for example, implementing advanced algorithms to handle exceptional communication scenarios.

References

1. Sanders, A., Elangeswaran, C., & Wulfsberg, J. P.: Industry 4.0 implies lean manufacturing: Research activities in industry 4.0 function as enablers for lean manufacturing. Journal of Industrial Engineering and Management (JIEM) 9(3), 811–833 (2016).
2. Heidel, R. Industrie 4.0: The reference architecture model RAMI 4.0 and the Industrie 4.0 component. 1st edn. Beuth Verlag GmbH. Germany (2019).
3. Inigo, M. A., Porto, A., Kremer, B., Perez, A., Larrinaga, F., & Cuenca, J.. Towards an Asset Administration Shell scenario: A use case for interoperability and standardization in Industry 4.0. NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium, 1-6. IEEE (2020).
4. Alemão, D., Rocha, A. D., Nikghadam-Hojjati, S., Barata, J. (2022). How to Design Scheduling Solutions for Smart Manufacturing Environments Using RAMI 4.0? IEEE Access (10), 71284–71298 (2022).
5. Herman, G., Pentek, T., Otto, B. Design principles for Industrie 4.0 scenarios: A literature review. 2015 49th Hawaii International Conference on System Sciences, 3928–3937 (2015).
6. Liu, Y., Xu, X. Industry 4.0 and Cloud Manufacturing: A Comparative Analysis. Journal of Manufacturing Science and Engineering, 139(3), 034701 (2017)
7. Febriani, R. A., Park, H.S., Lee, C.M. An approach for designing a platform of smart welding station system. The International Journal of Advanced Manufacturing Technology, 3437–3450 (2020).
8. Ye, X., Hong, S. H. Toward industry 4.0 components: Insights into and implementation of asset administration shells. IEEE Industrial Electronics Magazine, 13(1), 13-25 (2019).
9. ZVEI: Die Elektroindustrie. GMA Status Report: Reference Archtitecture Model Industrie 4.0 (RAMI 4.0) (2015).
10. Adolphs P., et al. Structure of the administration shell: continuation of the development of the reference model for the Industrie 4.0 component Federal Ministry for Economic Affairs and Energy (BMWi), Berlin (2016).
11. García, J. I., Cano, R. E., & Contreras, J. D. Digital retrofit: A first step toward the adoption of Industry 4.0 to the manufacturing systems of small and medium-sized enterprises. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 234(8), 1156-1169 (2020).
12. Arm, J., Benesl, T., Marcon, P., Bradac, Z., Schröder, T., Belyaev, A., ... & Dohnal, P. Automated design and integration of asset administration shells in components of industry 4.0. Sensors, 21(6) (2021).
13. Rehman, H. U., Chaplin, J. C., Zarzycki, L., Mo, F., Jones, M., & Ratchev, S. Service based approach to asset administration shell for controlling testing processes in manufacturing. IFAC-PapersOnLine, 55(10), 1852-1857 (2022).
14. Oppelt, M., Urbas, L. Integrated virtual commissioning an essential activity in the automation engineering process: From virtual commissioning to simulation supported engineering, in: Industrial Electronics Society, IECON 2014-40th Annual Conference of the IEEE, IEEE. pp. 2564–2570 (2014).
15. Schamp, M., Hoedt, S., Claeys, A., Aghezzaf, E.H., Cottyn, J. Impact of a virtual twin on commissioning time and quality. IFACPapersOnLine 51, 1047–1052 (2018).