Proc. of the 11th International Conference on Engineering and Emerging Technologies (ICEET)
22-23 October 2025, Kuala Lumpur, Malaysia

2025 International Conference on Engineering and Emerging Technologies (ICEET) | 979-8-3315-6755-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICEET67911.2025.11424056

Leveraging WebGL for Interactive and Real-Time Data Visualization through Digital Shadows in Automotive Industry

André Moreira
CeDRI,
Instituto Politécnico de Bragança,
5300-253 Bragança, Portugal
andre-moreira@ipb.pt

Ana Mara Moreira
CeDRI,
Instituto Politécnico de Bragança,
5300-253 Bragança, Portugal
ana.mmoreira@ipb.pt

José Barbosa
CeDRI, SusTEC,
Instituto Politécnico de Bragança,
5300-253 Bragança, Portugal
jbarbosa@ipb.pt

Rogério Tavares
CeDRI, SusTEC,
Instituto Politécnico de Bragança,
5300-253 Bragança, Portugal
rogerio.tavares@ipb.pt

Carlos R. Cunha
CeDRI, SusTEC,
Instituto Politécnico de Bragança,
5300-253 Bragança, Portugal
crc@ipb.pt

Paulo Leitão
CeDRI, SusTEC,
Instituto Politécnico de Bragança,
5300-253 Bragança, Portugal
pleitao@ipb.pt

***Abstract*—The rise of Industry 4.0 and the increase of data captured, as well as its diversification, has intensified the need for advanced data visualization methods that support this volume of information, real-time monitoring and decision-making in industrial manufacturing environments. Traditional dashboards, while widely adopted, often rely on static 2D representations that lack spatial context, limiting their effectiveness in scenarios where complex and three-dimensional physical objects, such as vehicles, are involved. To address this gap, this work presents a WebGL-based module integrated into a dashboard system to enhance the visualization of data captured in vehicle production lines. The proposed module leverages interactive 3D models, designed as digital shadows of the physical counterpart, to provide a more intuitive and immersive understanding of the production state. This system establishes seamless bidirectional communication between the dashboard and the WebGL module, enabling real-time updates and user interactions, while the data captured in the industrial manufacturing workstations by methods such as geometric measuring systems and sensors is mapped to the 3D model, allowing overseers to identify measurement points and potential defects with improved clarity. Based on the combination of interactive visualization with rule-based monitoring, the system supports the pursuit of zero-defect manufacturing. This work contributes with a replicable use case of integrating WebGL into dashboards for industrial applications, highlighting both technical implementation and practical benefits.**

Keywords—WebGL, 3D Scanning, 3D Models, Data Visualization, Digital Shadows, Industry 4.0

I. INTRODUCTION

During the last years, the rapid evolution of digital technologies has transformed the way industrial systems are designed, monitored, and optimized. This transformation, known as Industry 4.0, integrates modern technologies such Cloud Computing, Internet of Things, and advanced data analytics to achieve smarter, more autonomous, and highly efficient manufacturing environments [1]. Being a central element in this transformation the vast amount of data generated by devices and sensors, which can be leveraged to improve productivity and quality [2]. However, extracting meaningful insights from this data requires effective strategies for visualization and interaction, ensuring that decision-makers can interpret complex information quickly and accurately.

Traditionally, dashboards have been one of the most widely used tools for data visualization, including in the industry sector, their capacity to integrate heterogeneous data and present them in accessible formats has made them essential in different contexts. However, despite their effectiveness, conventional dashboards often rely on two-dimensional charts, graphs, and static images, which, while suitable for some monitoring tasks, it can be insufficient when the data is related to three-dimensional elements or when the monitored processes involve complex physical assets - in which cases, relying exclusively on 2D abstractions may result in the occlusion of important contextual information and in limiting the ability of overseers and engineers to identify problems, analyze trends, or make timely interventions, highlighting the need for more advanced visualization techniques that combine the accessibility of web applications with the depth of three-dimensional representation.

Recent research points to the potential of using 3D elements (e.g., 3D models, spatial text) combined with Extended Reality (XR) technologies, such as Mixed Reality (MR) and Augmented Reality (AR), as tools capable of addressing this challenge and innovating current methods of data visualization [3], [4], [5]. While XR uses specific devices and differs from the traditional dashboards, 3D models can be integrated into dashboards, enriching the visual representation of manufacturing data while also enabling new forms of interaction between users and the digital counterparts of real-world objects. In this context, WebGL emerges as a particularly promising technology to promote this integration, by being a JavaScript API for rendering interactive 3D elements within browsers, WebGL allows complex visualizations to be deployed without the need for external software or specialized hardware beyond a standard computer or mobile device, making it a potential technology to be embedded in dashboards, to provide the lacking 3D interactivity and functionalities. Such integrations can also promote concepts like digital shadows and digital twins, which provide synchronized virtual replicas of physical assets and allow for real-time monitoring, simulation, and even bidirectional control.

This paper explores the development and implementation of a WebGL-based module designed to enhance industrial dashboards with interactive 3D visualization capabilities, being for test case integrated in a dashboard made with Next.js. Specifically, the module is applied to the monitoring of vehicle production lines, where data collected from geometric measuring systems is mapped in a digital shadow of the physical object.

II. Background

A. Data Visualization and Industry 4.0

With the Industry 4.0 and the integration of new data sources (e.g., sensors), it becomes more important to evolve the current methods of data visualization and monitoring, in a way to also answer the recurrent need of the manufacturing industries to optimize their production processes and to support decision-makers [6].

A common tool for data visualization are dashboards due to their simplified and attractive interface, while presenting themselves as utilitarian tools to visualize trends, segmentations and real-time data, in the form of visual components, which are able to create narratives that can be easily interpreted [7]. Aside from being able to reproduce visual outputs based on raw data captured, dashboard also are powerful tools to be combined with technologies such as Business Intelligence [8], which is the study of past and present performance.

B. Digitalization and Manufacturing Industry

Current technological evolution has incorporated processes like digitalization in various sectors, such as Tourism [9] and Education [10], and at different levels, ranging from basic task automation to the development of complete Digital Twins. When applied to industrial manufacturing, this process may involve standardized digital descriptions of assets, such as the Asset Administration Shell (AAS), as well as the use of 3D models, either designed from scratch or obtained through scanning processes.

While AAS serves as a standardized digital representation of an asset, containing metadata, properties, operational parameters, and links to related resources [11] - which makes it crucial for ensuring interoperability in Industry 4.0 environments [12] - 3D models provide the visual and geometric representation of the same asset, supporting tasks that require spatial awareness.

These models, then combined with technologies such as Extended Reality (e.g., Virtual Reality) or WebGL can be integrated in systems/applications and then used for different purposes such as education/training in manufacturing processes while providing safer environments [13], inventory management without the ambiguity and risks from inefficient documentation [14] or to improve human-robot interaction with techniques such as virtual commissioning [15]. Together, the AAS and the 3D model form complementary elements that can be integrated into digital twins or digital shadows, each addressing different but interconnected needs.

C. Digital Twins and Digital Shadows

Among the technologies associated with the Industry 4.0 are Digital Shadows and Digital Twins, which are an upgrade of the traditional Digital Model, being the latter the one most commonly known and approached, however the popularization "Digital Twins" is also the result of some misconceptions, being sometimes used as a term to describe what are in reality Digital Shadows [16], [17].

A shared characteristic of both technologies is the use of a digital replication of the real-world object, commonly a 3D model or scanning of the object and, aside from the digital replication, is also the presence of data and interaction with the object, however the difference between those technologies starts in how the flow of this data and interaction works. An example of Digital Shadows can be a scenario where the digital object replicates the state of the real-world objects (e.g., position/rotation) and the user is able to modify the digital counterpart of the object by interacting with the real-world object, being the data flow unidirectional [18], however when the user is also able to modify the state of the real-world object through interaction with the digital replication and the data flow become bidirectional, then the scenario becomes of Digital Twins, which can include use cases like controlling excavators [19] or industrial robots [20].

III. Module Implementation

The implemented module goal is to provide a dynamic and innovative way of visualizing data captured from sources like geometric measuring systems in industrial manufacturing workstations, more specifically in production lines of vehicles [21], being part of the goal of this project the monitoring of the production lines to achieve a zero defect manufacturing scenario through the of a rule-base monitoring tool [22]. This module is then seamlessly integrated in a dashboard made with Next.JS, which serves as principal end-user application in the system (see Fig. 1) [23], being inserted on a system structured based on a microservice architecture [24].

Among the technologies available to implement the intended module - a component for 3D visualization to be integrated in a web application - were alternatives such as Three.js and WebGL, both capable of executing natively within any modern web browser and thus ensuring immediate cross-platform availability without additional setup. While Three.js offers higher-level abstractions and a simplified development process, it is ultimately built on top of WebGL, meaning that the underlying rendering pipeline remains the same. By working directly with WebGL, it becomes possible to achieve finer control over performance optimization, rendering strategies, and seamless integration with other standard web technologies (e.g., HTML, CSS, and JavaScript). In addition, WebGL reduces dependency on external libraries, which contributes to greater flexibility and efficiency in tailoring the visualization to the specific requirements of the project. For these reasons, WebGL was selected as the most suitable option, combining lightweight implementation with scalability and broad accessibility in the context of the proposed solution.

For the present work, Unity was adopted as the development environment, as it provides a comprehensive set of tools for 3D modeling, interaction design, physics, and asset management, which significantly accelerates prototyping and implementation. Once the module was created in Unity, it was exported to WebGL, enabling it to run natively in modern web browsers without the need for additional installations. This approach combines the robust authoring capabilities of Unity with the accessibility and cross-platform scalability of WebGL, ensuring both efficient development and broad availability of the final solution.

As for the integration of the module, it was made through the use of the package *jeffreylanters/react-unity-webgl*, being currently used the version 10.1.5, this package is supported both in React.js as well in Next.js web applications and it provides both a component (Unity) to render the HTML necessary to integrate the WebGL, as well the methods to configurate and interact with it through the hook *useUnityContext*, which includes methods like *sendMessage()* and *addEventListener()*, which are required to allow communication between the dashboard and the module.

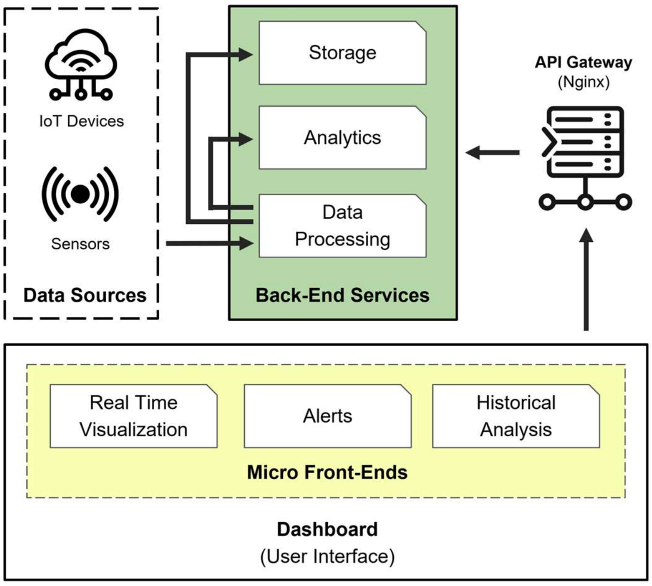

Fig. 1. Data Flow Overview (Adapted from [23]).

A. Front-End Components

The implemented module serves as a component in the dashboard to visualize the vehicle in production as well the measurement captured in the workstation, while providing a higher level of interaction and dynamization in comparison to static imagens, being then required a different approach in the design of the front-end when compared with normal (serious) games or general end-user applications.

1) Station 3D Scanning: The 3D scans of the workstations, currently three different workstations, allow a better understanding of the production process and its existent machinery. Using the 3D editing tool Meshlab, it is possible to perform a simple cleanup of the scans and to convert them into a file type that is compatible with the 3D modeling tool Blender.

Through Blender, each point of the point cloud is converted into a sphere for better readability, with the use of geometry nodes (see Fig. 2).

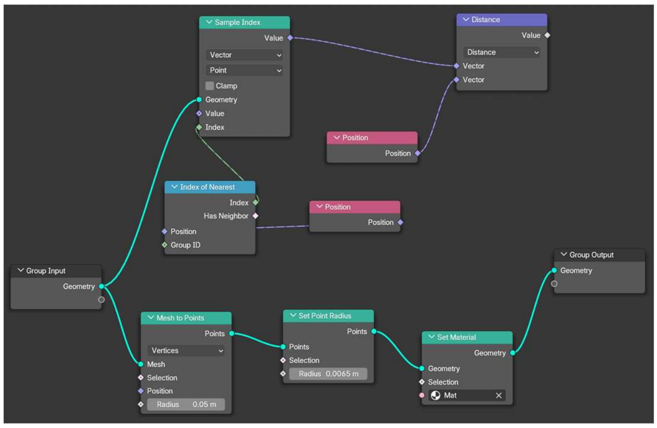

Fig. 2. Geometry Nodes to transform the Point Cloud.

After which, by accessing the color attribute present in the original scans, it is also possible to create a custom shading material that accurately colors each sphere based on the colors of the real world (see Fig. 3).

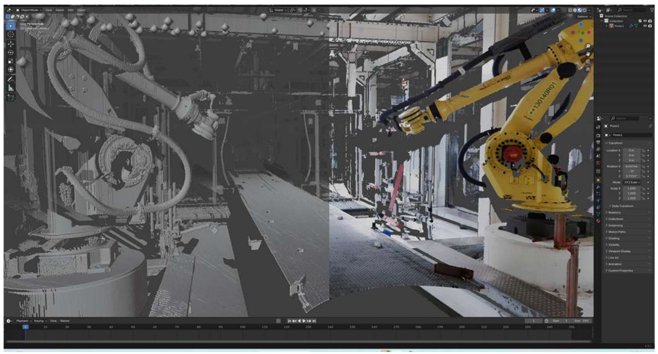

Fig. 3. Scanning Solid Mode / Material Preview Mode.

The scanning of the station serves to view the real world through 3D compatible software, while giving the possibility to develop 3D meshes that replicate the real world machinery and components used in the production process.

Through the use of the current 3D scans as base, it is planned to produce digital shadows of the industrial robots, to provide the overseers and general users of the dashboard a quicker understanding of the workstations progress.

2) Vehicle Digital Shadow: As to replicate the real word vehicle for the digital shadow feature, a 3D model was made, being used the modeling tool Blender (see Fig. 4) and the image reference modeling techniche, based on blueprints and schematics provided by the manufacturer. The model itself was divided by components (e.g., doors, framing, …) to allow a simplified modification of the visual outputs and updates of the models, that is, while the WebGL module contains just one 3D model of the vehicle, the dashboard can specify which state of the production is the vehicle, and with that, which components should be visible.

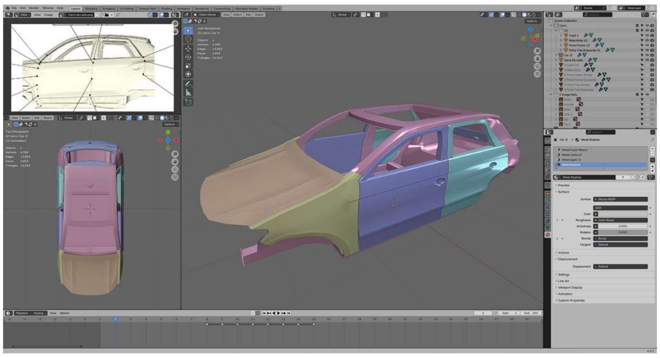

Fig. 4. Vehicle 3D Modelling.

3) User Interface: While the main interface used by the end-user is the one present in the dashboard which contains the WebGL module, some components were created based on defined requirements, such as visual tips of the measure, point which required the creation of specific shaders, as well models that are used to provide "fast information" and to provide options of interations, which will then trigger events in the dashboard, such as opening a different webpage or a pop-up.

B. Back-End Processes

While acting as module in the dashboard, to provide dynamic data visualization and to complement the end-user web application, it also consists in an application with its own logic, to be able of processing data, as well the communication between both applications (see Fig. 5).

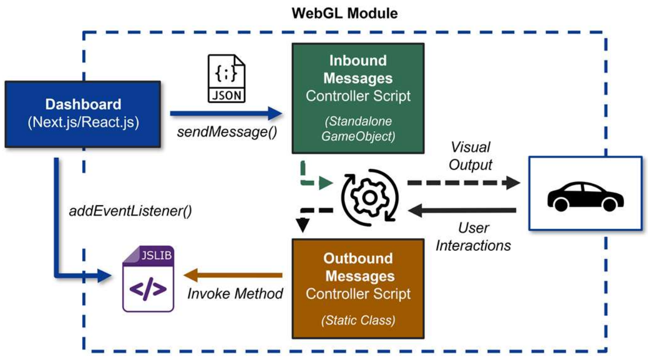

Fig. 5. Dashboard/WebGL Communication Model.

1) Inbound Messages Controller: The inbound communitcation is done through the method *sendMessage()*, which requires the specification of the method to be invoked inside the WebGL, the arguments and the name of the GameObjetct which has as component the script with the defined method to be invoked, being a GameObject a building block in the scene. This GameObject is also defined by the script as a standalone GameObject, which means that there should only be one per time. The method *sendMessage()* however has limitations in terms of arguments which can sent to the WebGL, being limited by one argument, requiring the use of alternatives such as JSON (see Table I) to allow sending more complex data. After the JSON be deserialized, and based on the method invoked, the data is then processed which can result in alterations of the visual output in the dashboard.

TABLE I. EXAMPLE OF INBOUND MESSAGE

Method Name	JSON Content		
	Attribute	***Value Type***	***Example***
WebGL_SwitchDataProcessing	toProcess	Bool	False
	keepPoints	Bool	True
	resetPoints	Bool	False

2) Outbound Messages Controller: To send messages from the WebGL to the dashboard a different process is required, being necessary the creation of a JSLIB plugin file which will contain a set of methods that can be called by the WebGL application after done the build, the methods can then make use of the method *window.dispatchReactUnityEvent()* from the *jeffreylanters/react-unity-webgl* package, which allow to call methods from the dashboard as well sent arguments through events, this however requires the definition of a listener for each method though the use of the method *addEventListener()*, as well as a *removeEventListener(),* these listeners, aside from requiring the specification of the event to listen, which in this case will be the same defined in *window.dispatchReactUnityEvent()* (e.g., "WebGL_PointId") it also requires the definition of the method which will be invoked when triggered the event, being used an *useCallback()* hook. Among the existent outbound messages existent, the most used is the *WebGL_ChangeCursorIcon* (see Table II), being this method necessary to change the cursor visual with the support of the CSS in the dashboard - since by the default WebGL does not modify the cursor – which provides a better user experience, since the user will be able to identify what visual components can be clicked or grabbed.

TABLE II. EXAMPLE OF OUTBOUND MESSAGE

Method Name	Parameters	
	Value Type	***Example***
WebGL_ChangeCursorIcon	String	"grab"

3) Messages Processing: Different actions can be triggered through the messages, some can modify/configutate the interaction with the WebGL modules, such as limiting the manipulation of the model 3D or changing what event is triggered when done some interaction (e.g., click in measure mark). Among the principal messages being processed is the ones related to the visualization of data, as well the mapping of strategic points (see Fig. 6) which are being measured by sensors in the real world workstations, this data is firstly received by the dashboard for specific uses, and then forward for the WebGL module. The data received by the module is a JSON, which is structured based on the Avro schema, being processed by the logic in the module and then, based on the specifications defined by the dashboard, will provided visual outputs of that data.

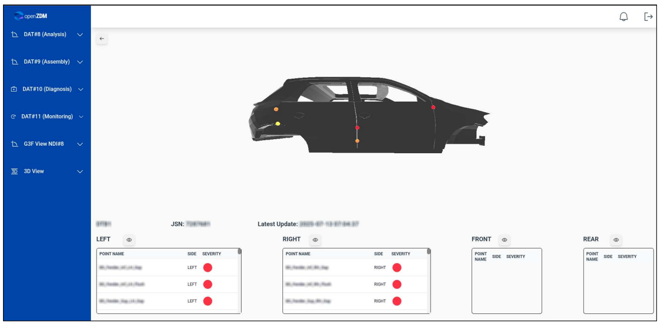

Fig. 6. Dashboard with WebGL Module.

IV. Conclusion and Final Remarks

This work presents a replicable use case of WebGL as a complementary component of a dashboard, being used to present interactable digital shadows and to display real-time data captured from external data sources.

The data consists in measurements captured by sensors in industrial manufacturing workstations, that is saved and kept on a database for future use, the data is then retrieved by the dashboard, through the use of WebSocket's, and forwarded to the WebGL module, being this feature supported by a bidirectional communication between the WebGL module and the dashboard, without the requirement of external tools/applications (e.g., WebSocket's), which also allows a seamless utilization of both components.

With the use of a 3D model to map and visualized the measurement points instead of static 2D images, the overseers of the production line have a better notion of the identified problems, this module also provides new potential use cases aside from real-time data visualization.

While the data displayed is captured in real-time, and the WebGL module functionalities are configurable by the dashboard, based on each webpage requisite, the initial mapping of the measurement points is done in a pre-build stage, being saved on *Scriptable Objects* in the format of dictionaries, similarly to JSON structures, which limits the modification and addition of measurement points, future work would then involve the transaction of this data to a Document Database (e.g., MongoDB) to store the list of the measurement points and crucial data such as their position and workstation to which they are assigned.

Aside from this transaction, the current WebGL module would also require new features to allow the overseers to modify the lists of measurement points.

Acknowledgment

This work was partially supported by the HORIZON-CL4-2021-TWIN-TRANSITION-01 openZDM project, under Grant Agreement No. 101058673. Also, it was supported by national funds: UID/05757 - Research Centre in Digitalization and Intelligent Robotics (CeDRI); and SusTEC, LA/P/0007/2020 (DOI: 10.54499/LA/P/0007/2020)

References

[1] R. Pandey et al., "A systematic review of industry 4.0 technologies in the production and manufacturing sector," in Materials Research Proceedings, 2025, pp. 197–206. doi: 10.21741/9781644903438-20.

[2] S. Lahmine and F. Bennouna, "Transforming Quality Management with Industry 4.0 Technologies: A Meta-Analytic Review of AI, Blockchain, IoT, and Big Data," International Journal of Industrial Engineering and Production Research, vol. 36, no. 2, pp. 68–79, 2025, doi: 10.22068/ijiepr.36.2.2271.

[3] S. Vasudev, P. Kumari, J. Sawlani, and P. Sharma, "Immersive Urban Narratives: Augmented Reality Data Visualization for Smart City Engagement," in 2025 IEEE International Students' Conference on Electrical, Electronics and Computer Science, SCEECS 2025, 2025. doi: 10.1109/SCEECS64059.2025.10940973.

[4] M. Perno, L. Hvam, and A. Haug, "Uses and challenges of digital twins-based augmented reality in operator training and data visualization in process manufacturing lines," Flex Serv Manuf J, 2025, doi: 10.1007/s10696-025-09605-w.

[5] M. Zheng, D. Lillis, and A. G. Campbell, "Current state of the art and future directions: Augmented reality data visualization to support decision-making," Visual Informatics, vol. 8, no. 2, pp. 80–105, 2024, doi: 10.1016/j.visinf.2024.05.001.

[6] V. R. Nair and V. V Panicker, "Power of Data Visualization in Industry 4.0: Leveraging Quality Management," in Lecture Notes in Mechanical Engineering, 2023, pp. 583–599. doi: 10.1007/978-981-19-6945-4_44.

[7] A. Lavalle, A. Maté, M. Y. Santos, P. Guimarães, J. Trujillo, and A. Santos, "A methodology for the systematic design of storytelling dashboards applied to Industry 4.0," Data Knowl Eng, vol. 156, 2025, doi: 10.1016/j.datak.2025.102410.

[8] M. Belghith, H. Ben Ammar, F. Masmoudi, and A. Elloumi, "Data Visualization for Industry 4.0: Developing Dashboards with Power BI – A Case Study in a Pharmaceutical Company," in Lecture Notes in Mechanical Engineering, 2023, pp. 402–408. doi: 10.1007/978-3-031-14615-2_45.

[9] R. Hammady, M. Ma, C. Strathern, and M. Mohamad, "Design and development of a spatial mixed reality touring guide to the Egyptian museum," Multimed Tools Appl, vol. 79, no. 5–6, pp. 3465–3494, Feb. 2020, doi: 10.1007/S11042-019-08026-W.

[10] C. R. Cunha, A. Moreira, S. Coelho, V. Mendonça, and J. P. Gomes, "Converging extended reality and Machine Learning to improve the lecturing of geometry in basic education," Journal of Engineering Research (Kuwait), 2024, doi: 10.1016/j.jer.2024.10.016.

[11] L. Sakurada, F. D. L. Prieta, and P. Leitao, "A Methodology for Integrating Asset Administration Shells and Multi-agent Systems," in IEEE International Symposium on Industrial Electronics, 2023. doi: 10.1109/ISIE51358.2023.10227964.

[12] C. Liepert, C. Stary, A. Lamprecht, and D. Zügn, "Digital Twin Data Provision Within Engineering: An AAS PLM Implementation Ensuring Interoperability," in Communications in Computer and Information Science, 2025, pp. 3–23. doi: 10.1007/978-3-031-72041-3_1.

[13] S. Yang, S. A. Mirahmadi, E. Zhu, and B. Solanki, "Live digital twin with virtual reality for accessible and immersive manufacturing education," The International Journal of Advanced Manufacturing Technology, vol. 136, no. 7, pp. 3577–3590, 2025, doi: 10.1007/s00170-025-15078-w.

[14] X. Liu, Y. Zheng, Z. Liu, and G. Lai, "A Digital Management System for Equipment Inventory Maintenance," in 2025 8th International Conference on Advanced Electronic Materials, Computers and Software Engineering, AEMCSE 2025, 2025, pp. 454–457. doi: 10.1109/AEMCSE65292.2025.11042572.

[15] X. Zhou, L. Fan, K. Ding, Y. Shang, and L. Ding, "Research on Virtual Commissioning System for Human-Robot Collaboration Assembly Cell

based on AutomationML," Procedia CIRP, vol. 130, pp. 1303–1309, 2024, doi: https://doi.org/10.1016/j.procir.2024.10.243.

[16] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital Twin: Enabling Technologies, Challenges and Open Research," IEEE Access, vol. 8, pp. 108952–108971, 2020, doi: 10.1109/ACCESS.2020.2998358.

[17] S. M. E. Sepasgozar, "Differentiating Digital Twin from Digital Shadow: Elucidating a Paradigm Shift to Expedite a Smart, Sustainable Built Environment," Buildings, vol. 11, no. 4, 2021, doi: 10.3390/buildings11040151.

[18] I. Errandonea, S. Beltrán, and S. Arrizabalaga, "Digital Twin for maintenance: A literature review," Comput Ind, vol. 123, p. 103316, 2020, doi: https://doi.org/10.1016/j.compind.2020.103316.

[19] S. M. E. Sepasgozar, "Digital Twin and Web-Based Virtual Gaming Technologies for Online Education: A Case of Construction Management and Engineering," Applied Sciences, vol. 10, no. 13, 2020, doi: 10.3390/app10134678.

[20] M. Ostanin, S. Zaitsev, A. Sabirova, and A. Klimchik, "Interactive Industrial Robot Programming based on Mixed Reality and Full Hand Tracking," IFAC-PapersOnLine, vol. 55, no. 10, pp. 2791–2796, 2022, doi: https://doi.org/10.1016/j.ifacol.2022.10.153.

[21] I. M. Treuk, A. O. Júnior, R. P. Lopes, G. Mota, J. Joaquim Mira, and P. Leitao, "Digitalization of Industrial Inspection Assets through the Asset Administration Shell," in 2024 IEEE 7th International Conference on Industrial Cyber-Physical Systems (ICPS), 2024, pp. 1–6. doi: 10.1109/ICPS59941.2024.10640027.

[22] J. Costa et al., "Real-Time Rule-Based Monitoring Tool to Achieve Zero Defect Manufacturing," in Studies in Computational Intelligence, 2025, pp. 273–285. doi: 10.1007/978-3-031-85316-6_20.

[23] V. B. Dos Santos, F. de La Prieta, P. Alves, J. E. Fernandes, and P. Leitão, "Micro Frontends in Real-Time Data Analytics Dashboards For Industry 4.0," in Proceedings of the 20th Iberian Conference on Information Systems and Technologies (CISTI), Lisbon, Portugal, 2025.

[24] R. Pedro Lopes, A. Ibrahim, J. Barbosa, and P. Leitao, "Microservices architecture to enable an open platform for realizing zero defects in cyber-physical manufacturing," Log J IGPL, vol. 33, no. 4, p. jzae112, Aug. 2025, doi: 10.1093/jigpal/jzae112.