XR Interaction in Industrial Applications: Unity as a Platform for Working with Digital Twins*

Andrii Yarmilko[1,†], Dmytro Buriak[1,*,†] and Andrii Danylov[1,†]

[1] *Bohdan Khmelnytsky National University of Cherkasy, 81, Shevchenko Blvd., 18031, Cherkasy, Ukraine*

Abstract

This article examines the challenges associated with digital twins in industrial applications. Here we analyze advantages of using the Unity game engine to support ineractions with digital twins. Requirements of environments for interaction with digital twins designed for monitoring and controlling technological processes are formulated. An XR interface architecture is proposed for working with digital twins using the visual tools of the Unity platform. The proposed architecture allows to separate data, simulation, and visualization layers. The application of the developed architecture for creating an XR-enabled applications to support additive manufacturing processes has been investigated. Here, we used a 3D model of the Prusa MK4S 3D printer as the simulated digital twin. It was established that the basic Unity tools, supplemented with XR tools and animation plugins, provide a visual representation of technological parameters and convenient user interaction with the digital twin. This confirms the positive impact of the proposed architectural solutions on the flexibility of the system and the simplification of its future expansion.

Keywords

Digital Twin, Unity, Additive Technology, XR Interface

1. Introduction

The use of virtual models of physical objects, processes, and systems - known as digital twins - enables the real-time reproduction of their state, behavior, and lifecycle through the integration of data, gathered from sensors on physical models or their simulated copies, and the subsequent analysis of this information [1]. Digital twins are necessary to overcome the limitations of traditional monitoring and forecasting methods, allowing for the simulation of physical copies, testing of operational scenarios, prediction of potential failures or errors, and optimization of operations without interacting with the physical object, which significantly reduces potential risks and resource costs.

Originally proposed by Michael Greaves in the context of product lifecycle management, the concept was further developed and adopted by NASA and other organizations in the aerospace technology sector [1, 2]. A quarter of a century later, we are witnessing the implementation of digital twins across a wide variety of human activities. At the same time, the research towards the effective methods of supporting the concept in practical applications remains a pressing task.

The aim of this article is to investigate the use of the Unity game engine as a platform for supporting the concept of digital twins in industrial applications. The research focused on finding innovative solutions, tailored to manufacturing processes involving additive technologies.

CEUR-WS.org/Vol-4326/paper4.pdf

* *The Third International Conference of Young Scientists on Artificial Intelligence for Sustainable Development: Intelligent Systems and Artificial Intelligence (YAISD-WS 2026:ISAI) May 7-8, 2026, Ternopil, Ukraine*

[1,*] Corresponding author.

[†] These authors contributed equally.

a-ja@ukr.net (A. Yarmilko); buriak.dmytro1121@vu.cdu.edu.ua (D. Buriak); danylov.andrii1122@vu.cdu.edu.ua(A. Danylov)

0000-0003-2062-2694 (A. Yarmilko); 0009-0009-4180-3436 (D. Buriak); 0009-0003-4651-6598 (A. Danylov)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-11

2. Theoretical Background and Related Works

There are various aspects to the classification of digital twins. For instance, based on their purpose, we distinguish between twins of products (components), processes (production lines or operations), and systems (complex interconnected entities). In terms of integration level, solutions range from static digital models with manual data updates to full-fledged twins with automatic bidirectional real-time information exchange[1].

The application of digital twins spans across a wide range of industries, such as manufacturing, energy, transportation, healthcare, construction, and others. They enable real-time monitoring of various systems, resources optimization, virtual testing of potential changes to production processes, and support for decision-making based on statistical data collected during the operation of the digital twins[2].

Today, there are numerous examples of digital twins being used in industrial manufacturing technologies. In particular, in additive manufacturing, also known as 3D printing, they provide a dynamic virtual copy of the physical printer, the printing material, and various printer components, which are linked in real time to the printing processes through synchronized sensor data, visual simulation support, and data analysis [3]. Such twins are subject to strict requirements regarding bidirectional information exchange, simulation of physical processes (thermal, mechanical, hydrodynamic), high computing power for real-time simulations, transmission of accurate sensor data, and the use of machine learning for printer calibration and conditions prediction [3, 4].

Modern digital twins are capable of capturing key process characteristics of the system, such as temperature fields and gradients, the distribution of remaining stresses and strains in printed parts, the evolution of the material's microstructure, potential defect formation (pores, cracks, warping), as well as printing process parameters —laser power, scanning or extrusion speed, melt viscosity, and cooling [4, 5]. An example of such a digital twin is the one for the laser powder bed fusion (LPBF/SLM) process, where thermal simulations are combined with sensor data to detect defects and correct parameters in real time [5, 6].

Another example is a digital twin ecosystem for FDM printing, which monitors nozzle temperature, filament feed rate, and platform position, enabling defect prediction and virtual system testing [7].

Such systems support platforms based on MATLAB/Simulink, Ansys Additive, or open-source tools such as OctoPrint with Unity, as well as cloud architectures with IoT protocols (MTConnect) that enable integration with production lines [3, 7]. As a result, digital twins significantly reduce the number of trial iterations, improve the accuracy of manufactured parts, and promote the "first-time-right" principle in manufacturing [3, 4]. They also enable the optimization of the entire 3D printing production lifecycle, from design to post-processing [6].

Among the key features that address common challenges with digital twins in 3D printing are built-in physics engine for accurate simulation of nozzle kinematics, material extrusion, and cooling; Line Renderer components for color visualization of G-code; adapted UnityVolumeRendering with multithreaded processing (Thread Ninja) for volumetric visualization of internal pores; and VR/AR support for layer-by-layer inspection and collaborative work via Photon Unity Networking. Easy integration with IoT protocols (Firebase, MQTT, OctoPrint), and Raspberry Pi allows data to be collected from low-cost sensors (magnetic encoders, vibration sensors) and displayed with a latency of only 500–700 ms, which is critical for real-time parameter correction and defect prevention [8, 9].

3. Unity as a Platform for Working with Digital Twins

Software tools for working with digital twins require a visual interface capable of displaying specific analytical data or visual representations. For the visually appealing interfaces, gaming

platforms are worth considering, as this is, in part, their functional purpose — to provide high-quality graphical representation.

A review of the major giants in the gaming industry highlights two dominant platforms: Unreal Engine [10] and Unity [11].

If we briefly compare those, Unreal Engine is a more complex platform to master and develop on. It provides a highly detailed and high-quality visual representation. For tasks involving digital twins, this platform is better suited for the modeling stages due to its high-quality graphics tools.

The Unity platform, on the other hand, is a simpler and more accessible option to understand and use. But it also has its own advantages in terms of presentation, namely the ability to work with virtual reality. That is, when working with digital twins, the Unity platform, using VR/XR technologies, allows you to create an interactive interface that displays technical metrics for the digital twins by overlaying this data onto a visual representation of the real world. The availability of a large number of libraries and plugins for working with VR/XR (Meta SDK, OpenXR SDK, Oculus SDK, XR Interaction Toolkit) allows for the organization of convenient interaction with both the interface and individual visual elements: dragging UI blocks in XR space, moving interactive objects, and interacting with UI components (sliders, drop-down lists, etc.).

Interaction with digital twins is an industrial task that relies on large amounts of numerical data. The Unity platform also supports various libraries for tasks such as real-time data acquisition and output.

These features of the Unity platform have sparked interest in it as a foundation for modern industrial XR solutions. An example is a sensor-based digital twin for FDM printers [8]. Here, Unity serves as the visualization and control interface: data from magnetic encoders (filament flow with an error of only 0.73%), vibration sensors, and nozzle positioning are sent to Firebase, processed, and displayed in real time with a delay of 500–700 ms. The system automatically detects layer shifts, bad printing, or excessive vibration and allows for remote adjustment of parameters from a phone or laptop. The total cost of such a system is less than $50, making it accessible for small and medium-scale production where constant quality control is required without expensive industrial platforms.

Another example is a collaborative VR twin for inspecting additive manufacturing parts [9]. An immersive environment was created using Unity (version 2019.4.20f1), where G-code trajectories (green/yellow lines with red error markers), visualization of the printing process, and 3D scans showing defects (pores, voids) are overlaid in real time. Unity's tools for rendering volumes and lines, along with multithreaded processing, allow users to examine a part layer by layer, cut cross-sections, and collaboratively annotate issues in virtual space with multiple users simultaneously. This is particularly valuable for aerospace and medical parts, where precise inspection of the internal structure resolves quality issues even before post-processing.

Thanks to these capabilities, Unity transforms 3D printing into a predictable, efficient Industry 4.0 technology, where visualization, forecasting, and collaboration take place in a single environment.

Thus, using the Unity game engine to build digital twins in 3D printing is highly advantageous due to its real-time interaction architecture, which allows for the creation of dynamic virtual replicas of equipment, processes, and components with minimal latency. Unlike heavy-duty industrial simulators such as ANSYS [12] or COMSOL [13], Unity combines powerful 3D visualization, a physics engine, and flexible data integration, making it accessible to both research laboratories and small-scale manufacturing. It is ideal for additive processes where thousands of layers, temperature gradients, and potential defects need to be tracked in real time, transforming static models into a live, interactive monitoring and prediction system [7, 9].

As a platform, Unity provides a balance between real-time rendering capabilities, interactivity, and ease of deployment compared to traditional engineering tools. This makes Unity particularly well-suited for creating user-centric XR interfaces, rather than for high-precision physical simulation.

4. An Architectural Approaches for Working with Digital Twins Using Unity's Visual Tools

The approaches described in the previous section do not fully meet the requirements for working with digital twins, as they typically focus on either visualization or the computational side, failing to provide sufficient separation between the data layer, simulation, and user interface. Such solutions often lack mechanisms for local caching, normalization, and data reuse in the event of losses or delays from the primary source, which reduces system reliability in near-real-time mode. Furthermore, they do not always support seamless integration with the XR environment and do not provide sufficient interactivity for working with the digital twin's UI elements. This is precisely why an architecture is needed that combines flexible data processing, fault tolerance, and the ability to visually present technical metrics in Unity.

Implementations of the Unity platform in projects involving digital twins [8, 9] are more complex and customized. When used to support production-scale technological workflows, processing such a large volume of information places a heavy burden on the operator and increases the demands on their skills and knowledge. Additionally, such detailed applications may face challenges with performance optimization and the use of VR headsets. Expected directions for the architectural modernization of digital twins include simplifying control and interaction with the operator. This involves the implementation of an immersive HMI, in which most of the responsibility for maintaining interaction is shifted to the hardware.

Therefore, we are considering a solution for multiple digital twins (workshops or production lines) that features as simple as possible interface, yet provides the minimum necessary functionality and information. Such approaches require a mechanism for obtaining data from the digital twin itself or from a physical model (REST API, MQTT, etc.). An important component is the consolidation of incoming data from data integration sources in near real-time. Another obvious requirement for the interface is ensuring an intuitive level of understanding and visual comfort for the operator. And the use of various plugins and packages within the application's structure should significantly reduce its own weight, ensuring its optimization even for relatively older VR headset models (such as the Oculus Quest 2).

In response to these requirements, an architecture has been proposed that meets them through a division into the physical layer, data layer, simulation layer, visualization layer, and XR interface. This approach allows for the separate processing of input data via REST API or other integration channels, performing its consolidation, normalization, and local caching, and then transferring it for visualization in a Unity scene. Additionally, the system provides cyclic data updates at intervals close enough to real time, which is critical for industrial scenarios. As a result, the system becomes not just a display tool, but a full-fledged interactive environment for monitoring, analyzing, and interacting with the digital twin.

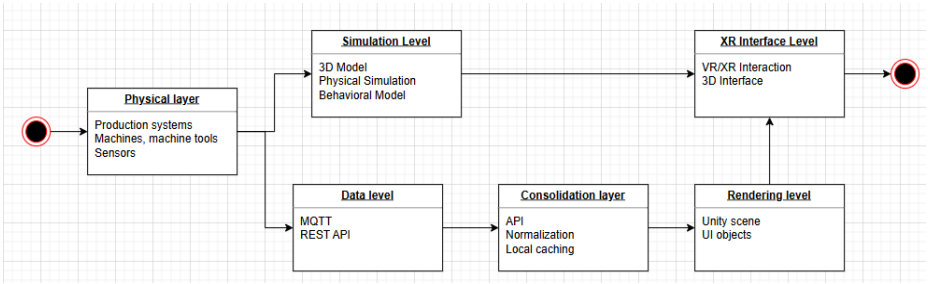


Figure 1: An architectural proposal for an XR interface to interact with digital twins using Unity's visual tools.

This architecture treats the Unity platform as a way to present various types of performance metrics for digital twins. The operator's interaction with the technological process begins at the

physical level, which provides real-world models, data, and metrics for subsequent simulation. From there, the process branches out: transitioning to the simulation level and the data level. The simulation level is responsible for generating or transmitting 3D models of process equipment, performance metrics, and so on. In effect, it enables the visualization of real, physical objects. The data level retrieves data from the physical level via REST API or other data integration methods. The next step involves data consolidation with local caching and normalization. After that, the data is sent to the visualization layer, where it is transferred to Unity scene UI objects for display (numeric outputs, sliders, indicators of process parameters, etc.). In the final stage, the results obtained at the simulation and visualization layers are transferred to the XR interface layer, where they form a complete, interactive structure. The logical structure of the described process of interaction with the Unity platform is illustrated in the activity diagram.

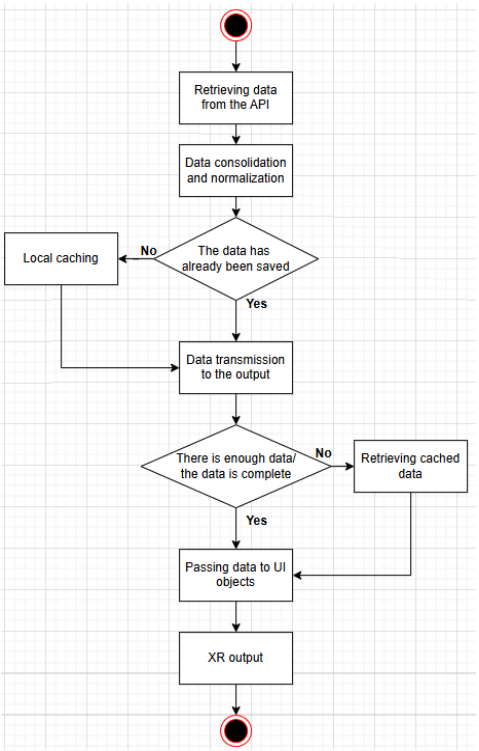


Figure 2: A structure of data transmission and rendering processes for Unity's XR component.

As shown in API data (simulated sensor readings and/or metrics) is first obtained, which then undergoes normalization and consolidation. The system then checks whether a local copy of the data exists. If none is found, local caching is performed. Next, calculations, unit conversions, and other operations are performed on the data, along with the analysis of its sufficiency. If data loss or corruption is detected, data from the cache is used instead of the direct source. In the next step, the data is transmitted to the Unity scene's UI objects. The next and final stage is rendering in the XR environment, where the UI is provided with additional interactivity. The algorithm in must run cyclically, with a data refresh interval that ensures maximum real-time performance.

5. Experiments and Results

The following tools were used in the experimental study of the developed XR interface architecture for working with digital twins on the Unity platform:

1. OpenXR SDK [14] and XR Interaction Toolkit [15] – tools for integrating VR and utilizing its features;
2. DoTween – a plugin for creating smooth UI animations [16].

The Prusa MK4S model was considered as a simulated digital twin [17, 18]. Its 3D model was used as a representation.

The Prusa MK4S 3D printer is a modern example of a 3D printer leveraging additive manufacturing technology based on the FDM (Fused Deposition Modeling) method, which involves the layer-by-layer deposition of thermoplastic material. The device is an evolutionary continuation of previous models in the i3 series, specifically the MK3S+, developed by Prusa Research [19, 20]. The printer's build volume is approximately 250 × 210 × 220 mm, allowing for the production of both small parts and medium-sized objects. The Prusa MK4S supports a wide range of printing materials, including PLA, PETG, ABS, and other engineering plastics, which expands its potential for use in research and manufacturing applications [17].

From the perspective of a digital twin, a 3D printer model serves as a foundational geometric representation that can be integrated with the device's physical parameters, such as kinematic characteristics, temperature profiles, print head movement speed, and other operational data. This allows for the creation of a virtual environment to simulate the printing process, analyze potential defects, and optimize parameters even before the physical fabrication of the object.

To build a digital twin, data from open sources was used, including official manufacturer documentation, technical specifications, and CAD models available in community repositories. Integrating this data with the simulation environment enables a more accurate reproduction of the real device's behavior and enhances the effectiveness of research in the field of additive technologies [18].

For simulation purposes, the input data includes both the device's own parameters and data from the model being printed (number of layers). The key process parameters are following:

3. nozzle temperature;
4. cooler rotation speed;
5. print progress (current layers and their total number, percentage);
6. print duration (current, not estimated);
7. extruder position;
8. length of extruded material (mm);
9. feed rate (mm/s).

Together, these parameters allow for an assessment of both the physical condition of the equipment and the dynamics of the object's manufacturing process. Key parameters include nozzle temperature, which directly affects the quality of melting and material adhesion, as well as the cooler rotation speed, which determines the intensity of cooling for the printed layers. Print progress is displayed as the current layer number, the total number of layers, and their percentage completion, allowing you to assess the stage of the process. Additionally, the current print time is taken into account, which reflects the actual duration of the process without using predictive models. The extruder's position is specified as coordinates in three-dimensional space and allows tracking the print head's movement path. The length of extruded material (in millimeters) characterizes the total volume of filament used and is an important parameter for cost analysis. The material feed rate (mm/s) determines the extrusion intensity and affects the accuracy and quality of layer formation.

Together, these parameters form a complete set of input data for the digital twin, ensuring an accurate reproduction of the real 3D printing process in a simulation environment. This, in turn, allows for the analysis of system behavior, the identification of potential anomalies, and the optimization of printer operating modes.

Figure 3 shows an overview of the project's UI. The entire UI shown here was implemented using Unity's basic tools.

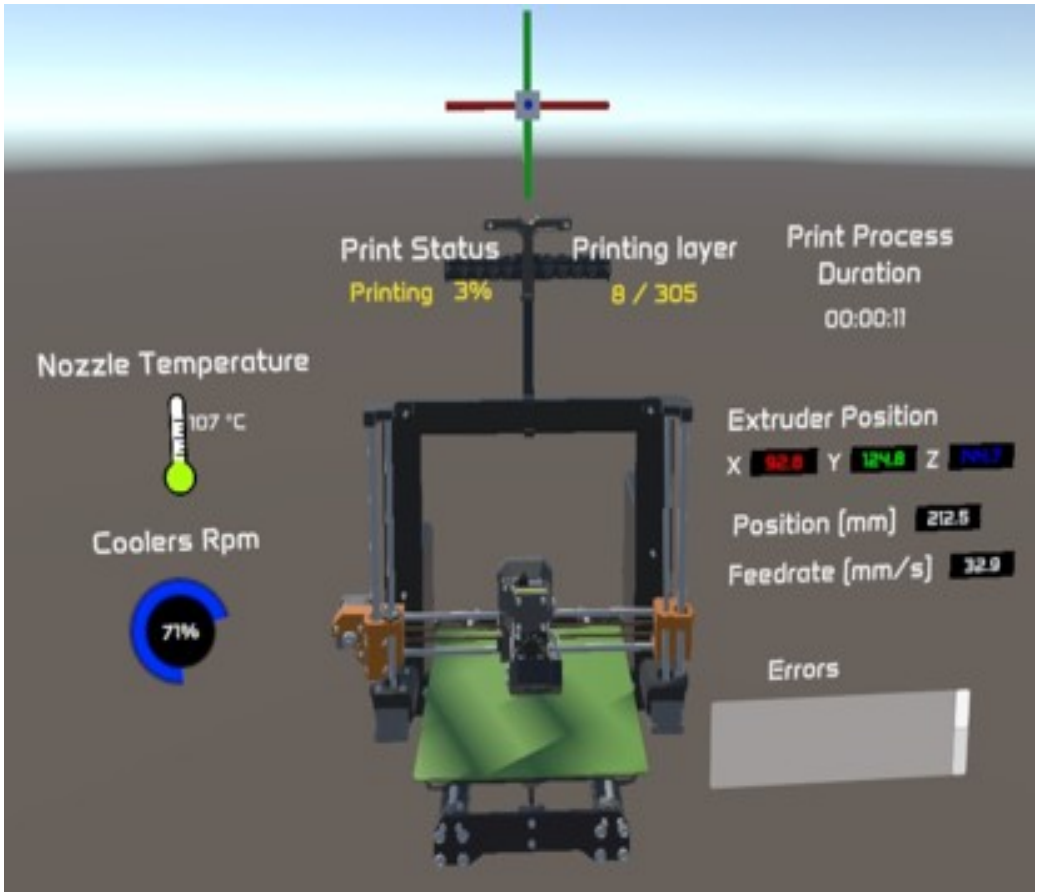


Figure 3: A digital twin UI settings.

The value of the Print Status field indicates the current status of the print job for a specific color (Printing – yellow, Stopped – red, Done – green). Next to it, the progress percentage is displayed, calculated based on the number of layers printed. The layer printing (Printing Layer) indicator also changes color (below maximum – yellow, maximum layers – green). The printing duration (Printing Process Duration) is visualized using a stopwatch with basic logic. It stops when errors occur and/or the process is paused (status changes to Stop).

The Nozzle Temperature display is presented as a thermometer filled with gradient colors (green, yellow, red) corresponding to the temperature value. The filling is done using the DoTween plugin [16]. The display of cooler RPM (Coolers Rpm) is presented in a speedometer design with a numerical reading. The DoTween plugin is also used for this display.

The display of extruder information consists of simply outputting floating-point numbers into text fields. It is worth noting that for the extruder position (Extruder Position), different text colors are used to indicate the axes (X – red, Y – green, Z – blue). To help visualize the physical placement of the axes, a 3D object was added above the printer object.

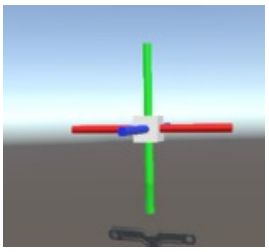


Figure 4: Visualization of the spatial coordinates of an additive process.

There is also an additional output: the "Errors" list. It is logically linked to all outputs and configured with specific limits. For example, it displays a message if the nozzle temperature exceeds 230 degrees Celsius.



Figure 5: Error indication for the temperature profile of the manufacturing process.

Also, additional logic was implemented to better immerse the operator into the workflow:

1. The UI display area for the current printer is limited; moving outside of it hides the printer. This is suitable for situations involving entire lines of machines where diagnostics need to be performed.
2. The UI responds to the camera's position (VR headset) and rotates towards the user while remaining readable. This does not restrict the user to standing directly in front of the printer.

The entire visual interface, through variations in text color under certain conditions, conveys an understanding of changes in specific process parameters, their current values, and the status of processes. According to estimates, for the operator, these additional visual elements make it approximately 40–60% easier to receive and process information on the screen compared to a 2D display on monitor screens.

Overall, the developed experimental application demonstrates the expected results: the UI provides minimal functional information about the digital twin's operation, aimed at intuitive understanding of the process by the operator, with additional features for comfortable interaction. Modeling the operation of a digital twin line using flow logic will provide a simple, brief, and clear summary of the operation of each device in the factory line.

The app's performance metrics were also analyzed: the system demonstrates an average data refresh latency of 500–700 ms, which is acceptable and close to real-time monitoring. The application maintains a stable refresh rate of 100+ frames per second on the Oculus Quest 2 headset. Since this is a relatively old model, these figures indicate good optimization and raise positive expectations regarding performance improvements when using newer and more powerful systems.

6. Conclusions

This article analyzes the concept of digital twins and their application in additive manufacturing and examines the relevance of Unity game engine as a platform for their visualization and interactive support. A new architectural solution for an XR interface is proposed for working with

digital twins using the visual tools of the Unity platform. As part of the experiment, a simulated digital twin of the Prusa MK4S 3D printer was created with a set of key printing process parameters: nozzle temperature, print progress, extruder position, and material feed rate. The results showed that Unity provides a visual representation of data, color-coded status indicators, visualization of spatial coordinates, and real-time error display. Additionally, the ability to use XR tools to enhance the user experience with the digital twin was demonstrated. This confirms Unity's suitability for building flexible and visually intuitive solutions in the field of industrial digital twins.

Further research of Unity for working with digital twins should focus on integration with real-world IoT data sources, extending the model to multi-device production scenarios, and using machine learning methods to predict defects and optimize printing modes.

Declaration on Generative AI

During the preparation of this work, the authors used AI-based tools such as Grammarly Pro for grammar correction and Strike Plagiarism for detecting potential plagiarism. After using these tools, the authors carefully reviewed and edited the content and visual materials as needed and take full responsibility for the integrity and accuracy of the publication's content.

References

[1] M. Singh, E. Fuenmayor, E. P. Hinchy, Y. Qiao, N. Murray, D. Devine, Digital twin: Origin to future, Applied System Innovation 4 (2021) 36. doi:10.3390/asi4020036.

[2] E. H. Glaessgen, D. S. Stargel, The digital twin paradigm for future NASA and U.S. air force vehicles, in: Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 2012, pp. 1–14. doi:10.2514/6.2012-1818.

[3] S. Ben Amor, N. Elloumi, A. Eltaief, B. Louhichi, N. H. Alrasheedi, A. Seibi, Digital twin implementation in additive manufacturing: A comprehensive review, Processes 12 (2024) 1062. doi:10.3390/pr12061062.

[4] L. Zhang, X. Chen, W. Zhou, T. Cheng, L. Chen, Z. Guo, B. Han, L. Lu, Digital twins for additive manufacturing: A state-of-the-art review, Applied Sciences 10 (2020) 8350. doi:10.3390/app10238350.

[5] N. Jyeniskhan, K. Shomenov, M. H. Ali, E. M. Shehab, Exploring the integration of digital twin and additive manufacturing technologies, International Journal of Lightweight Materials and Manufacture 7 (2024) 860–881. doi:10.1016/j.ijlmm.2024.06.004.

[6] L. Tudorache, Ö. Babur, S. S. Lucas, M. van den Brand, Current approaches to digital twins in additive manufacturing: A systematic literature review, Progress in Additive Manufacturing 10 (2025) 10819–10853. doi:10.1007/s40964-025-01262-7.

[7] M. Pantelidakis, K. Mykoniatis, J. Liu, G. Harris, A digital twin ecosystem for additive manufacturing using a real-time development platform, The International Journal of Advanced Manufacturing Technology 120 (2022) 6547–6563. doi:10.1007/s00170-022-09164-6.

[8] K. Shomenov, M. H. Ali, N. Jyeniskhan, A. Al-Ashaab, E. Shehab, Cost-effective sensor-based digital twin for fused deposition modeling 3D printers, International Journal of Computer Integrated Manufacturing (2025) 1–20. doi:10.1080/0951192X.2025.2504085.

[9] V. Chheang, S. Narain, G. Hooten, R. Cerda, B. Au, B. Weston, B. Giera, P.-T. Bremer, H. Miao, Enabling additive manufacturing part inspection of digital twins via collaborative virtual reality, Scientific Reports 14 (2024) 80541. doi:10.1038/s41598-024-80541-9

[10] Epic Games, Unreal Engine, 2026. URL: https://www.unrealengine.com. Last accessed 2026/03/24.

[11] Unity Technologies, Unity, 2026. URL: https://unity.com. Last accessed 2026/03/24.

[12] Ansys, Ansys, 2026. URL: https://www.ansys.com. Last accessed 2026/03/24.

[13] COMSOL Multiphysics, COMSOL, 2026. URL: https://www.comsol.com. Last accessed 2026/03/24.
[14] OpenXR Plugin, Unity Documentation, 2026. URL: https://docs.unity3d.com/Packages/com.unity.xr.openxr@1.16/manual/index.html. Last accessed 2026/03/20.
[15] XR Interaction Toolkit, Unity Documentation, 2026. URL: https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@3.0/manual/index.html. Last accessed 2026/03/20.
[16] DoTween, Demigiant, 2026. URL: https://dotween.demigiant.com/. Last accessed 2026/03/20.
[17] Original Prusa MK4S 3D printer, Prusa Research, 2026. URL: https://www.prusa3d.com/product/original-prusa-mk4s-3d-printer-5/. Last accessed 2026/03/22.
[18] MK4S – documentation, Prusa Research, 2026. URL: https://help.prusa3d.com/product/mk4s. Last accessed 2026/03/22.
[19] Prusa Research, Official website, 2026. URL: https://www.prusa3d.com/. Last accessed 2026/03/22.
[20] About us, Prusa Research, 2026. URL: https://www.prusa3d.com/page/about-us_77/. Last accessed 2026/03/22.