



(RESEARCH ARTICLE)



Digital twin of smart classroom

Bhargavi Jangam, Praharika Reddy Gongidi *, Sindhu Kancharla and Nikhil Talari

Department of CSE (AI and ML) of ACE Engineering College, Hyderabad, India.

World Journal of Advanced Research and Reviews, 2026, 30(01), 172-178

Publication history: Received on 24 February 2026; revised on 01 April 2026; accepted on 03 April 2026

Article DOI: <https://doi.org/10.30574/wjarr.2026.30.1.0796>

Abstract

This project presents the design and development of a Digital Twin of a Smart Classroom that creates a virtual 3D replica of a real classroom environment. The system simulates sensor data such as temperature, light intensity, and occupancy, which is processed using AI-based decision logic. Based on these inputs, the digital twin dynamically controls classroom devices such as lights and fans while displaying real-time classroom status through 3D visualization. The proposed system aims to improve energy efficiency, enhance classroom comfort, and demonstrate the application of digital twin technology in smart education systems.

Keywords: Digital Twin; Smart Classroom; Artificial Intelligence; 3D Visualization.

1. Introduction

Traditional classrooms lack automation and real-time monitoring systems, which leads to inefficient energy usage and reduced comfort for students. With the advancement of smart technologies, digital twin systems have emerged as a powerful solution to simulate and monitor real-world environments in a virtual space.

A Digital Twin is a virtual representation of a physical system that updates dynamically based on real-time or simulated data. In this project, a smart classroom is modelled in Blender, where environmental factors such as temperature and student presence influence device behaviour like fans and lights.

The system provides an interactive and visual simulation where fans rotate when temperature rises, lights turn ON/OFF based on occupancy, and system status is displayed dynamically. This approach improves understanding of automation systems and demonstrates how AI and simulation can be integrated into smart infrastructure.

2. Literature Survey

2.1. M. Grieves & J. Vickers (2017) – Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior

Grieves and Vickers (2017) introduced the concept of Digital Twin as a virtual replica of a physical system that continuously updates using real-time data. Their work emphasizes the importance of synchronization between the physical and digital environments to monitor system performance, predict outcomes, and improve decision-making. The digital twin enables organizations to simulate real-world conditions without directly interacting with the physical system, thereby reducing risk and improving efficiency. This concept forms the foundation for modern smart systems and is directly relevant to the proposed project, where a virtual classroom simulates real-time environmental conditions such as temperature and occupancy.

* Corresponding author: Praharika Reddy Gongidi

2.1.1. Methodologies and Algorithms

The methodology proposed by Grieves and Vickers is based on real-time data synchronization between physical and virtual systems using sensors and data integration techniques. The system continuously collects data, processes it, and updates the digital model accordingly. Algorithms are used to analyse system behaviour, predict future states, and optimize performance. This approach relies on continuous feedback loops and data-driven decision-making, which are fundamental principles applied in the proposed smart classroom system, where simulated data is used to dynamically control devices such as fans and lights.

2.2. S. Boschert & R. Rosen (2016)– Digital Twin: The Simulation Aspect

Boschert and Rosen (2016) focused on simulation-based digital twin models that emphasize visualization and analysis of system behaviour. Their work highlights how digital twins can be used to simulate complex systems and observe their performance under different conditions. This approach allows users to understand system dynamics without affecting the actual system. The study demonstrates that visualization plays a key role in improving system understanding and decision-making. This concept is utilized in the proposed project through the use of Blender, where a 3D classroom environment visually represents system behaviour such as fan movement and lighting changes.

2.2.1. Methodologies and Algorithms

The methodology in this approach involves creating simulation models that replicate real-world systems using computational techniques. Algorithms are used to simulate different scenarios and analyse system responses based on input parameters. These models rely on mathematical and logical representations rather than real-time sensor data. Visualization tools are integrated to display system behaviour dynamically. In the proposed project, similar methodology is used where Python-based logic simulates environmental conditions and updates the 3D classroom model accordingly.

2.3. A. Fuller et al. (2020)– Digital Twin: Enabling Technologies, Challenges and Open Research

Fuller et al. (2020) explored the enabling technologies behind digital twin systems, including Internet of Things (IoT), Artificial Intelligence (AI), and 3D visualization. Their study explains how these technologies work together to create intelligent and automated systems. The integration of IoT allows real-time data collection, AI enables predictive analysis, and 3D visualization provides interactive representation of systems. This combination enhances automation and decision-making in smart environments. The proposed project reflects these concepts by combining simulation, automation, and visualization to represent a smart classroom digitally.

2.3.1. Methodologies and Algorithms

The methodology involves integrating multiple technologies such as IoT for data acquisition, AI for data processing, and visualization tools for system representation. Algorithms are used for data analysis, pattern recognition, and decision-making processes. These systems often use rule-based logic or machine learning techniques to automate system behaviour. In the proposed project, a simplified version of this methodology is implemented using rule-based Python logic, where conditions such as temperature and occupancy determine the operation of fans and lights.

2.4. Y. Lu et al. (2019) – Digital Twin-driven Smart Building Management

Lu et al. (2019) developed a digital twin model for smart building management, focusing on monitoring energy consumption and indoor environmental conditions. Their work demonstrates how digital twins can optimize resource usage and improve occupant comfort through automated control systems. By continuously analysing environmental data, the system can adjust lighting, ventilation, and other parameters to maintain optimal conditions. This approach is highly relevant to the proposed project, where similar automation is applied to control classroom lighting and fans based on simulated conditions.

2.4.1. Methodologies and Algorithms

The methodology involves collecting environmental data such as temperature, occupancy, and energy usage, and applying control algorithms to manage building systems efficiently. These algorithms use threshold-based logic and feedback mechanisms to automate device operations. Optimization techniques are also used to reduce energy consumption while maintaining comfort. In the proposed system, a similar approach is followed using rule-based algorithms, where fan rotation and light intensity are controlled based on predefined conditions, simulating smart building automation.

Table 1 Comparison Table of Literature Survey

S. No	Author(s)	Title	Methodology Used	Findings from the Reference Paper
1.	M. Grieves & J. Vickers (2017)	Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior	Proposed the digital twin concept for creating virtual replicas of physical systems using real-time data synchronization.	Introduced the concept of digital twin for real-time monitoring and system optimization. Improves decision-making but requires continuous data integration and high implementation cost.
2.	S. Boschert & R. Rosen (2016)	Digital Twin - The Simulation Aspect	Used simulation-based digital twin models to visualize and analyze system behavior.	Demonstrated how simulation improves system understanding and predictive analysis. Accuracy depends on model quality and lacks real-time data integration.
3.	A. Fuller et al. (2020)	Digital Twin: Enabling Technologies, Challenges and Open Research	Reviewed enabling technologies like IoT, AI, and 3D visualization for digital twin systems.	Highlighted the role of IoT and AI in building intelligent systems. Supports automation but introduces complexity in integration and data security challenges.
4.	Y. Lu et al. (2019)	Digital Twin-driven Smart Building Management	Developed a digital twin model for smart buildings to monitor energy usage and indoor conditions.	Showed how digital twins improve energy efficiency and occupant comfort. Requires infrastructure investment and faces scalability issues.

3. System Architecture

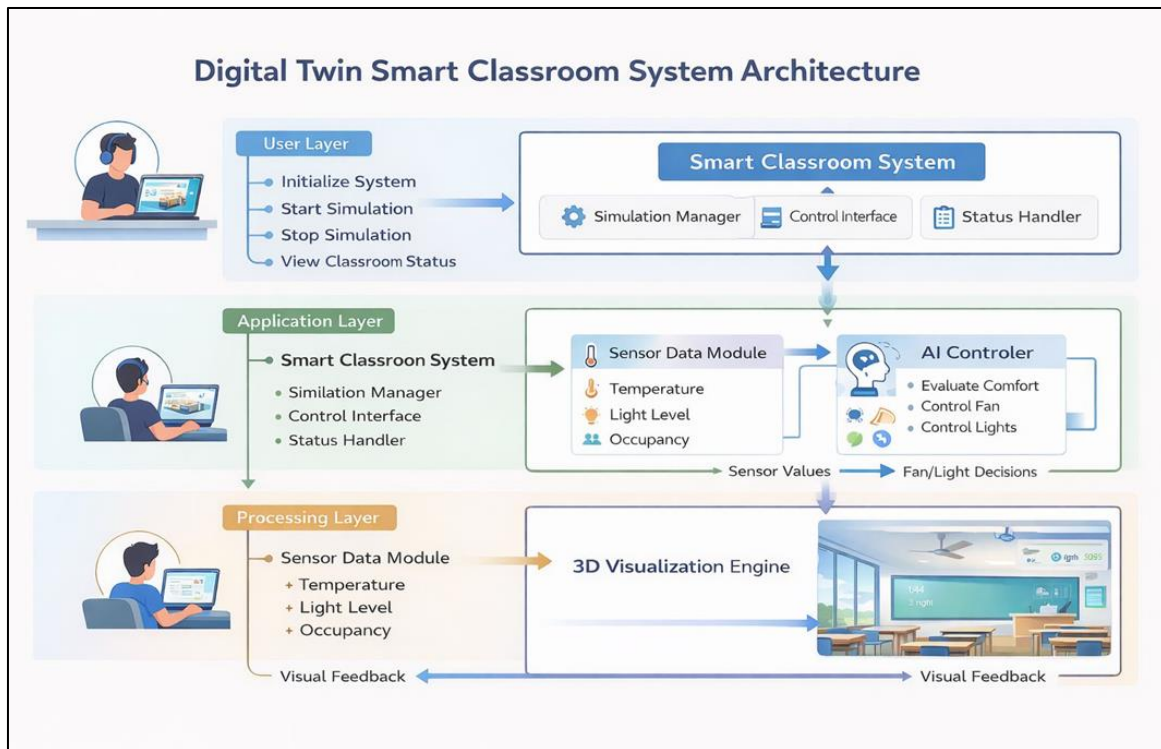


Figure 1 System Architecture

The system architecture of the Digital Twin Smart Classroom is designed based on three main layers: User Layer, Application Layer, and Processing Layer, which together simulate and control the classroom environment using Blender and Python.

The User Layer represents the interaction of the user with the system. In this project, the user starts or stops the simulation directly within the Blender environment. The user can view the classroom status visually, including fan movement, lighting conditions, and real-time data displayed on the board. This layer acts as the entry point for controlling the simulation.

The Application Layer acts as the core logic of the system. It is implemented using Python scripting in Blender. The Simulation Manager generates dynamic values such as temperature and student count using random functions. The Control Logic (implemented through conditional statements) evaluates these values and decides whether fans should rotate or lights should turn ON/OFF. The Status Handler updates the system state and sends the current status to the board display inside the classroom.

The Processing Layer is responsible for executing the simulation and visualization. The Sensor Data Module is simulated in this project by generating temperature and occupancy values instead of using real sensors. These values are processed using rule-based logic to control classroom devices. The 3D Visualization Engine is implemented using Blender, where the classroom model updates in real time. Fan objects rotate continuously based on temperature conditions, light intensity changes based on occupancy, and student objects appear or disappear to represent presence.

The system runs continuously using timer-based functions, where data is generated, processed, and visualized in a loop. This creates a real-time simulation of a digital twin, where the virtual classroom behaves like a real classroom. The architecture ensures proper interaction between data generation, system logic, and visual output, making the system realistic and interactive.

4. Proposed Methodology

4.1. 3D Classroom Modeling

The first step in the methodology involves designing a realistic 3D classroom environment using Blender. Various classroom components such as walls, floor, desks, chairs, fans, lights, board, and other objects are created or imported as assets. These elements are properly positioned and scaled to resemble an actual classroom setup.

Special attention is given to placing lights and fans correctly on the ceiling and arranging student seating systematically. The classroom model acts as the foundation of the digital twin, where all further simulation and automation are performed.

4.2. Data Simulation and Input Generation

Since real-time sensors are not used in this project, environmental data such as temperature and student occupancy is simulated using Python. Random values are generated at regular intervals to represent changing classroom conditions.

Student presence is simulated by dynamically showing or hiding student objects within the classroom. The number of visible student objects represents occupancy. This simulated data acts as input for the system, similar to how real sensor data would function in an actual smart classroom.

4.3. Control Logic and Automation

The control logic is implemented using Python scripting within Blender. Conditional statements are used to evaluate simulated data and control classroom devices accordingly. If the temperature exceeds a predefined threshold and students are present, the fans are activated and rotate continuously. If students are present, lights are turned ON, otherwise they are turned OFF.

Smooth animation is applied to enhance realism. Fan rotation is implemented using continuous rotation updates, while light intensity is gradually increased or decreased to simulate real-world lighting behavior. This automation mimics intelligent system behavior in a smart classroom environment.

4.4. Real-Time Visualization and System Update

The system uses Blender as a 3D visualization engine to display real-time updates. A text object placed on the classroom board shows the current system status, including temperature, number of students, fan status, and light status.

The entire system runs in a loop using timer-based functions, where data is generated, processed, and visualized continuously. This creates a dynamic and interactive simulation where the virtual classroom behaves like a real-world system, demonstrating the concept of a digital twin effectively.

5. Algorithmic Flow of the System

The working of the Digital Twin Smart Classroom system is explained through the following steps:

- Initialize the 3D classroom environment in Blender with objects such as fans, lights, desks, and student models.
- Start the simulation using Python scripting within Blender.
- Generate dynamic input data such as temperature and student count using random functions.
- Update student presence by showing or hiding student objects based on the generated student count.
- Check temperature conditions:
 - If temperature is greater than the threshold value, activate the fans.
 - Otherwise, keep the fans OFF.
- Check occupancy conditions:
 - If students are present, turn the lights ON.
 - If no students are present, turn the lights OFF.
- Apply continuous animation to fans by rotating them smoothly based on temperature conditions.
- Adjust light intensity gradually to simulate realistic ON/OFF transitions.
- Update the classroom board text with current system status including temperature, number of students, fan status, and light status.
- Repeat the above steps using timer-based execution to maintain continuous real-time simulation.

6. Output Screens

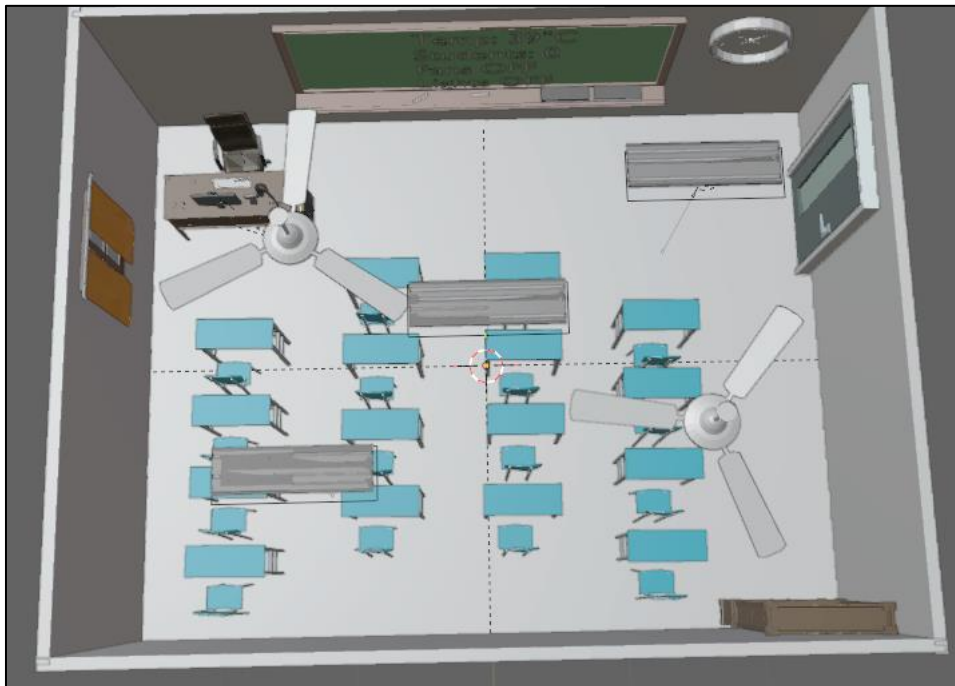


Figure 2 Overall 3D Model of the Smart Classroom Environment (Solid View)

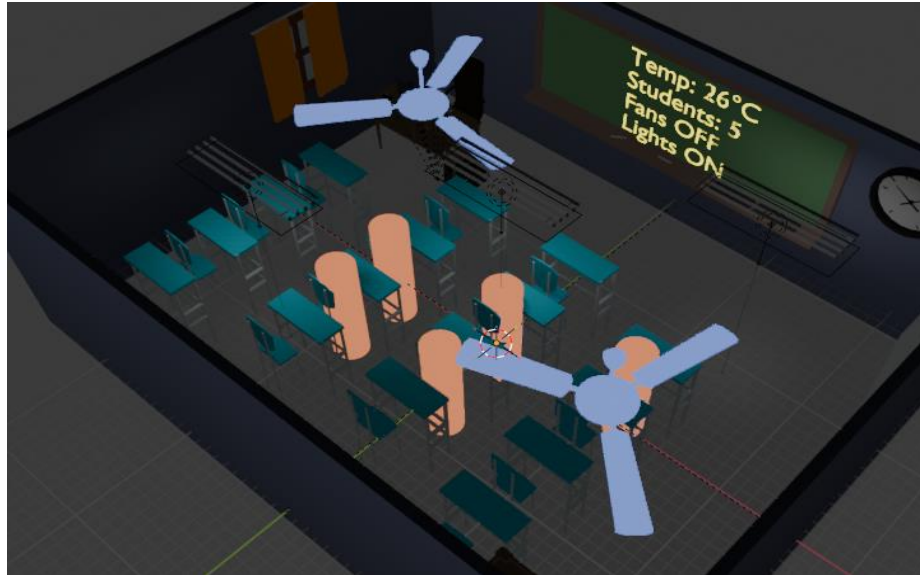


Figure 3 Simulation of Student Occupancy in the Classroom

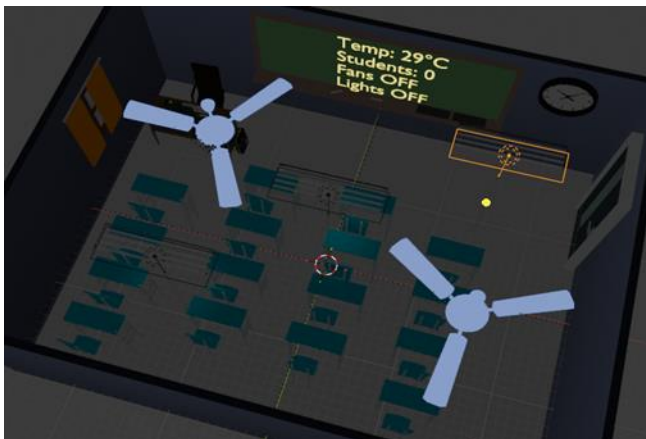


Figure 4 Fan Operation (OFF State)

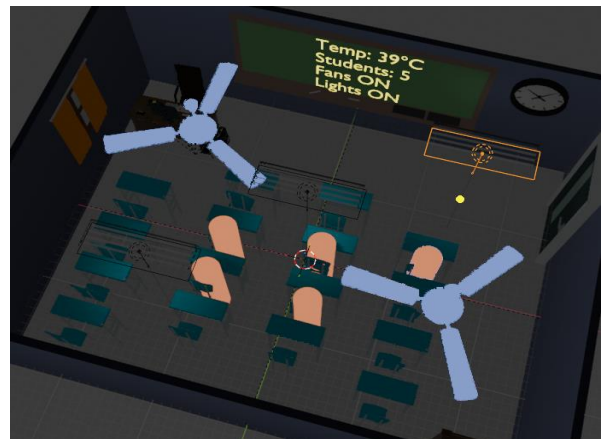


Figure 5 Activation of Ceiling Fans based on Temperature Conditions

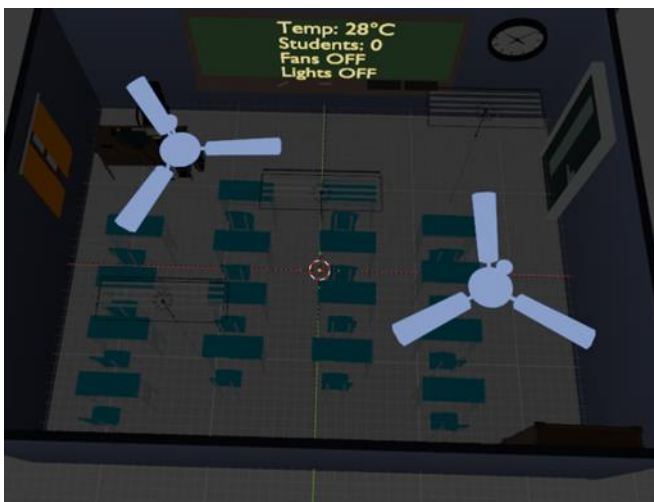


Figure 6 Light Operation (OFF State)

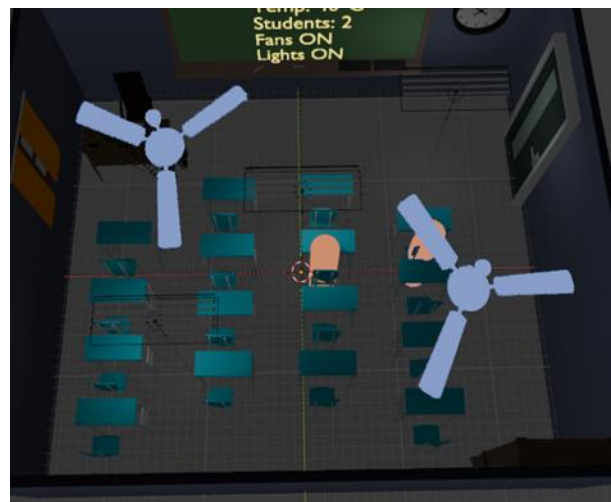


Figure 7 Classroom Lighting in ON State (Students Present)



Figure 8 Real-Time System Status Display on Classroom Board

7. Conclusion

The Digital Twin Smart Classroom project successfully illustrates the application of simulation and automation technologies in an educational environment. By integrating 3D modelling with Python-based logic, the system replicates real-world classroom behaviour in a virtual space.

The project highlights how environmental conditions such as temperature and occupancy can be used to automate devices like fans and lights, improving energy efficiency and user comfort. The use of animation and real-time updates enhances the effectiveness of the simulation and makes it visually engaging.

This work provides a strong foundation for future developments in smart classrooms, where real-time sensor data and IoT integration can further enhance system capabilities. The project demonstrates the potential of digital twin technology in building intelligent and adaptive systems.

Compliance with ethical standards

Disclosure of conflict of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

- [1] M. Grieves, "Digital Twin: Manufacturing Excellence through Virtual Factory Replication," White Paper, 2014.
- [2] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital Twin: Enabling Technologies, Challenges and Open Research," IEEE Access, vol. 8, 2020.
- [3] F. Tao et al., "Digital Twins and Cyber-Physical Systems toward Industry 4.0," IEEE Access, 2019.
- [4] Blender Foundation, "Blender – 3D Creation Suite," Available: <https://www.blender.org/>
- [5] Python Software Foundation, "Python Programming Language," Available: <https://www.python.org/>
- [6] S. Boschert and R. Rosen, "Digital Twin – The Simulation Aspect," Springer, 2016.
- [7] Y. Lu et al., "Digital Twin-driven Smart Manufacturing," Robotics and Computer-Integrated Manufacturing, 2019.
- [8] Online Resources and Tutorials on Blender and Digital Twin Simulation.