Digital Twin technologies for immersive virtual reality training environments

Ricardo Rodrigues[1], Rui Machado[1], Pedro Monteiro[1], Miguel Melo[2], Luís Barbosa[1,2], and Maximino Bessa[1,2]

[1] University of Trás-os-Montes e Alto Douro, Vila Real, Portugal
rrodrigues1999.rr@gmail.com
[2] INESC TEC, Porto, Portugal

Abstract. With industry evolution and the development of Industry 4.0, manufacturers are trying to leverage it and find a way to increase productivity. Digital Twins (DT) technologies allow them to achieve this objective and revolutionize Product Life-cycle Management as they provide real-time information and insights for companies, allowing real-time product monitoring. Virtual Reality (VR) is a technology that permits users to interact with virtual objects in immersive environments; even under constant development, VR has proven efficient and effective in enhancing training. DT integration into immersive VR environments is constantly developing, with many challenges ahead. This study aims the development of an immersive virtual world for training integrated with DT technologies to handle all users' input using the simulator. Those were subject to a performance evaluation to understand how the application handles different input types, which confirmed the viability and reliability of this integration.

Keywords: digital twin, virtual reality, immersive environments

1 Introduction

With the evolution of technologies for the Industry and the development of Industry 4.0, manufacturers have incorporated several types of sensors into their equipment that allow them to collect data that leads to obtaining relevant information about the machine's operation status or measuring the machine's past, present, or future performances. As new sensors for equipment development grows, it is also necessary to increase the capacity to communicate the data obtained to central systems, preferably in real-time [1]. Digital Twin (DT) technologies have revolutionized Product Lifecycle Management (PLM) as they provide live, or near real-time, information and insights for manufacturers that allow real-time monitoring and design of all PLM-related processes [2].

DT data is real-time data that reflects the physical object state. The DT integrates and converges real-time data and uses previously processed data to provide more helpful information to the system. This technology represents real entities synchronized through various sources such as sensors and continuous collection of information, allowing real-time representation of their state [3]. This data is processed and synchronized with the DT applications to provide more helpful information to the system and can be used to understand how the physical object works, detect anomalies and prevent malfunctions through simulations.

Along the DT, there are two other ways of representing reality in a digital system, the Digital Model and the Digital Shadow. Fig. 1 represents the three models, the flow of information, and the direction of communications between the real and virtual objects [4].

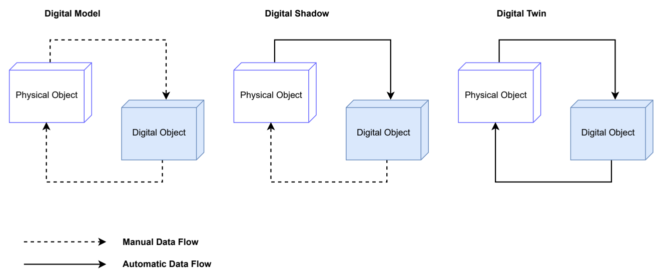


Fig. 1: Digital Twin Models

While Digital Model is used to present a concept and to create production and construction documentation, where data flow between the physical and the digital model as to be made manually, the Digital Shadow concept refers to the representation of the physical model on a digital model, where data flows automatically in a single direction, from physical to digital model, unlike the DT model where communication is bi-directional [5].

VR is the representation of a virtual environment and 3D models that can be interacted by the users with various input sources, such as using hand controllers, tracking gloves, or with Head-Mount Displays (HMDs). In these environments, various senses, such as vision, hearing, touch, and even smell, can be stimulated to give the user a greater sense of presence in the virtual environment [6]. Therefore, this technology can improve training and be used in several areas like medicine, marketing, entertainment, and others [7]. Different authors have defined VR according to their views on this technology. Some define "virtual" as something false or not real, and "reality" as a reference to the real world, which the conjunction might be interpreted as something like an imitation of reality [8]. A definition proposed by Steve Bryson is “The use of computer technology to create the effect of an interactive three-dimensional world in which the objects have a sense of spatial presence.” [9], that is, simulation of a 3D environment generated through a computer, where the user can interact, move and become immersed in this virtual world.

In a VR system, all human senses can be stimulated, which creates different impressions and sensations in the user inside the virtual world, increasing the user's sense of presence and immersion when using the VR application. Thus, the quality and immersion levels are determined by how involved and engaged the user is with the virtual environment during the VR experience and how these impression types stimulate all user's sensations[10][11]. The VR systems can be divided by the immersion that users feel when using the application in three main levels[11]: immersive, semi-immersive, and non-immersive. In particular, immersive systems allow users to feel fully immersed in the virtual environment, using HMDs responsible for tracking head movements and the user's position in the virtual world. In these systems, and to increase the level of immersion, different systems can stimulate different sensations. [11]

VR technologies can benefit from other technologies, such as DT, though most DTs, depending on the data's characteristics, are presented as text or graphics in non-immersive environments. Through the integration of DT and VR, users may have a more in-depth understanding of the dynamics of the represented model by experiencing deep immersion and a sense of reality [12].

One example of an application combining DT and VR is "Virtual Assembly Line Training". This application is a car assembly line simulator created to reduce critical errors, assembly costs and training time. This software allows customization of the assembly line to each scenario by analyzing the DT data.[13].

Another application that uses VR and DT is Flexsim. This solution permits the design and development of 3D factory simulations. FlexSim includes a standard models library that can replicate real-world operations functionalities so that users can build different scenarios. Flexsim is used in different areas such as industrial manufacturing, healthcare, and robotics, among others [14]

This work focuses on the integration of the DT in immersive virtual environments, having as its basis the SMARTCUT project, which intends to develop a solution that envisages making the most of these two technologies by simulating forestry and agricultural equipment based on CAN technology to allow the execution of telediagnostic tasks, remote maintenance and, monitoring of the forest. This project will use information from an existing DT, which will be processed and used in immersive VR environments. These tasks will be complemented with the use of the DT. A performance evaluation will be made to understand how the application handles different input types to validate and evaluate the proposed solution.

The next sections present the proposal and integration of DT technologies in a training VR application for training.

2 Digital Twin integration in a VR application for training

The main objective of this work is to integrate a DT into an immersive virtual environment (simulator), mainly to give visual feedback to the user. The DT reads inputs made at a physical emulator and sends them to the simulator that applies them to the corresponding virtual elements. The emulator replicates a real forestry machine with all associated controls, such as pedals, joysticks, and buttons.

In this work, the case study focuses on the operation of forest machines and harvester heads, where users control all those components. Users can use these 3D components to complete tasks associated with individual training sessions in the immersive environment. The training session is customized for each user and different scenarios by modifying the environment props, terrain, lighting, harvester model, and tasks. Upon completion of all tasks, the application generates results for each completed task and compares them with those of other users.

Through DT technology, several tasks can be accomplished during the training. They include the representation of the operation of the entire forestry machine and a harvesting head representing its operation and malfunctions that can be associated with each component. Those training courses consist of a group of tasks, including the operation and maintenance of forest machines.

The simulator and all virtual worlds were developed with the UNITY 3D graphics engine.

2.1 Communication Architecture

As shown in figure 2, the emulator and simulator communicate through the DT with a broker to emulate all machine and harvesting head movements.

There are two types of users: the trainee and the trainer. When using the application, the trainee does not interact directly with the simulator but instead utilizes all available controls in the emulator that sends all necessary information to DT. The simulator uses this information to update all virtual models. A trainer panel is built into the simulator and is used to interact directly with the application by changing the environment variables, such as time of day and weather, and affecting forest machine and harvesting head functionality.

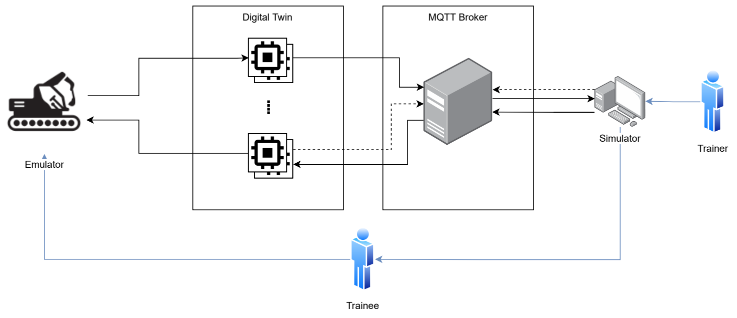


Fig. 2: MQTT Architecture

Upon initialization, the simulator subscribes to multiple topics (represented by dashed lines) and receives values from each (incoming solid lines). The simulator can also send data to Broker (outgoing solid line) to notify and synchronize the DT about what is happening with the virtual machine and harvesting head.

The DT is composed of several controllers, some responsible for storing and sending to the server the different states of each emulator component and others that will subscribe to multiple topics and are responsible for receiving information sent by the simulator and acting on the emulator. An MQTT broker transmits messages using the MQTT Protocol, a lightweight messaging protocol for IoT devices. By implementing the Observer pattern, the Broker effectively exchanges every message between the DT and simulator. [15].

According to listing 1.1, MQTT messages are JSON messages, composed of a topic that identifies the component, along with three values: (a) the current component value, the actual input value of the component at the emulator, (b) the default component value, the input value of component at emulator when not being used, and (c) the maximum component value, the maximum input value of component at the emulator. This message structure allows us to easily handle all needed values to apply to the virtual model. This data is continuously sent to the MQTT Broker, sending the message to all listeners who have subscribed to the current topic.

Listing 1.1: MQTT Message Json Schema

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "properties": {
    "atualValue": {
      "type": "integer"
    },
    "defaultValue": {
      "type": "integer"
    },
    "maxValue": {
      "type": "integer"
    }
  },
  "required": [
    "atualValue",
    "defaultValue",
    "maxValue"
  ]
}
```

2.2 Application Flowchart

To better understand and interpret the entire operation of starting and executing a training session in the application, a flowchart (Fig. 3) was designed that shows how the entire process will work.

When starting the application, the user is presented with an initial menu with several options, one of which is the training menu. When entering this menu, if there are already courses created, these are listed, and the user can choose one of them and start the train. Otherwise, it is possible to create training using the training creation menu, where the user chooses the environment and the desired harvesting head for the training execution. When starting the training, the trainee interacts with the machine and the harvesting head through the emulator until all the tasks required for the training are completed. At the end of the training, the obtained results are generated and saved, and the user is redirected back to the initial menu.

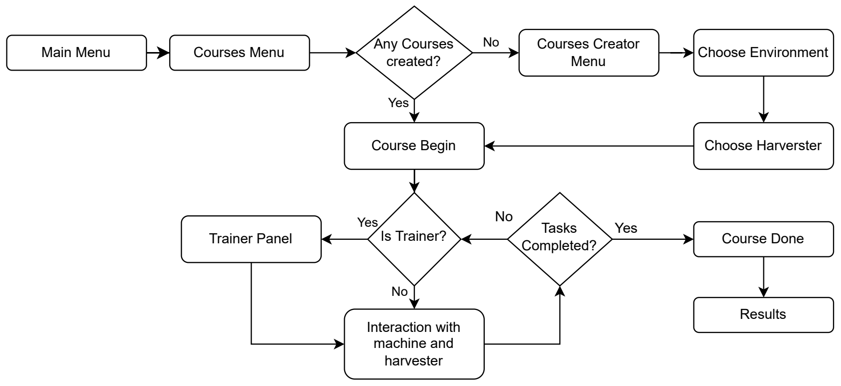


Fig. 3: Simulator's Flowchart

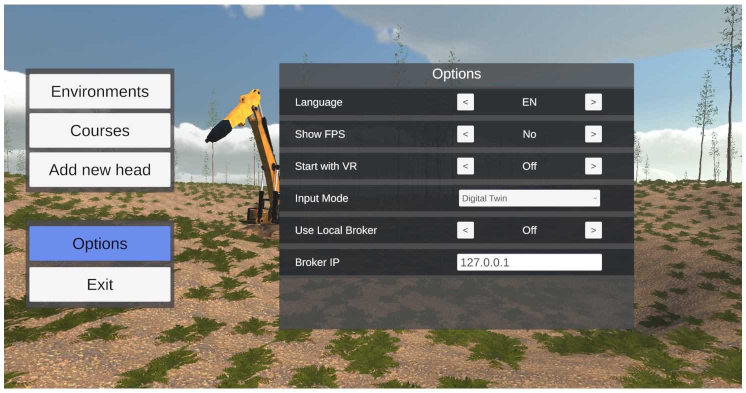
2.3 Integration Development

One of the purposes of the tool is to serve as a base for the operation of forest machines and harvesting heads in virtual environments, integrated into a DT system that will process all the control inputs and send the result to the simulator that will apply the proper effect to the forest machine.

This subsection will focus on how the communication between the DT and the simulator was implemented and how the messages are processed by the simulator and applied to the machine and the harvesting head to perform the training tasks.

To integrate Digital Twin with Unity 3D, a client was created to connect to the MQTT Broker and receive all messages for each topic using an external library.

Before starting the training, the user can access the options menu (Fig. 4) and choose, in the "Input Mode" option, to use the Digital Twin as input. When using Digital Twin as input, two other options are presented to define if the Broker Server is local, that is, if the server is executed on the machine that is running the simulator or if you want the Broker Server to be remote and executed on a remote machine and accessed through the IP Address defined in the “IP Broker” option. If you choose a local server, the simulator starts the Broker without any action from the user. All these options are saved in an external configuration file in JSON format, which allows these options to be set automatically whenever the application is started.



Options

Language EN
Show FPS No
Start with VR Off
Input Mode Digital Twin
Use Local Broker Off
Broker IP 127.0.0.1

Fig. 4: Application options menu

When starting the training scene, the application connects to the Broker Server and subscribes to all topics of the ***”digital_twin/controlo/”*** type. After these steps, the application waits until the DT connects to the Broker. When connected to the Broker, the DT starts sending messages to the topics associated with each control that contains all the information necessary for the application to work correctly. These topics, as well as all associated values, are defined in the DT on each controller. Each of these controllers can be associated with an extra controller, called an error block, which can change the original value of the input to another value. These blocks are helpful to replicate malfunctions and problems in the operation of some of the components of the forest machine, as well as in the harvesting head.

After receiving the values sent by the DT, the client processes the MQTT message and associates an event to each topic responsible for processing and applying the due effect to the intended component. These events have, as input parameters, the values sent in the MQTT message (Current Value, Default Value, and Maximum Value).

3 Performance Evaluation

To test the viability and reliability of the integration of the DT and the interactive immersive virtual environment, a protocol was designed with all the actions possible in the simulator, which is the basis of the performance evaluation. The main purpose of this evaluation is to assess how a DT-based approach performs against a conventional approach and validate its viability in terms of latency and responsiveness.

3.1 Procedure

To ensure that all protocol steps were executed uniformly, a script was developed to simulate user inputs through a direct input to the virtual environment or via a DT. In addition, each test was executed five times in two different environments for consistency purposes.

For evaluation purposes, two environments were created: one empty scene only with the forest machine (baseline) and the harvesting head attached, and the other was a highly populated scene with many trees, grass, and rocks (representative of a real case scenario). In addition, the input mode was also considered as a variable, direct input (representative of the conventional solutions). This resulted in eight different test conditions:

- Keyboard Inputs + Non VR mode + Empty Scene
- Keyboard Inputs + Non-VR mode + High Populated Scene
- Keyboard Inputs + VR mode + Empty Scene
- Keyboard Inputs + VR mode + High Populated Scene
- Digital Twin Inputs + Non-VR mode + Empty Scene
- Digital Twin Inputs + Non-VR mode + High Populated Scene
- Digital Twin Inputs + VR mode + Empty Scene
- Digital Twin Inputs + VR mode + High Populated Scene

The considered performance variables were Input Latency, FPS, RAM Usage, CPU Usage Percentage, CPU Time, GPU Usage Percentage, and GPU Time. The performance tests were executed on a Desktop Computer with the following specifications: Intel(R) Core(TM) i7-6700K CPU, NVIDIA GeForce GTX 1080 GPU, and 32 GB of RAM.

3.2 Results

The obtained results are shown in Table 1. The best performance was achieved when non-immersive displays and testing the application in an empty scene (ES), reaching an average of more than 300 FPS, and the worst when using VR displays and a High Populated scene (HPS), which was not able to reach 20 FPS in most of the cases. Looking at CPU Usage Percentage (CPU %) and CPU Time, we can see that they are slightly higher for the same scene type and display mode combinations when the DT makes application inputs. This was also expected since the application must process all MQTT messages. In terms of latency, the values are lower when using the keyboard compared with DT when executing the application in an empty scene, but this changes when we run the application in a complex scene and when FPS are lower, where DT inputs are more responsive. This mainly happens because, even though the application has to process all MQTT messages, this type of input allows us to transport input processing from the main application to the DT application.

Table 1: All average variables values of each test

	Digital Twin				Keyboard			
	VR		NON VR		VR		NON VR	
	ES	HPS	ES	HPS	ES	HPS	ES	HPS
Latency (ms)	11.53	31.72	6.55	17.31	4.03	47.48	1.52	22.81
FPS	72	16	309	32	72	18	334	37
RAM (Mb)	738	10771	724	10619	781	11540	750	11157
CPU %	10.53	25.32	21.11	34.21	8.97	24.24	19.09	32.28
CPU TIME (ms)	14.01	64.76	3.40	31.42	13.92	55.14	3.10	27.14
GPU %	58.82	78.75	83.29	84.21	63.21	98.90	89.56	98.03
GPU TIME (ms)	3.62	23.92	0.66	17.24	1.92	33.27	0.77	6.84

4 Conclusions

The study aimed to integrate an immersive VR application for training with DT and evaluate and validate it by analyzing its performance and comparing it across different scenarios, including conventional simulation (without DT) and across VR setups (non-immersive vs immersive).

According to the results obtained, we can conclude that creating an application with the primary purpose of rendering a simulation with complex scenes that leverages DT potential is feasible. In addition, DT approaches helped to improve application stability and reliability by handling input processing since CPU time and usage percentage were lower.

The developed tool allows trainers to create different training scenarios and change, whenever they want, the type of inputs and controllers without reprogramming the simulator's entire Input System. Also, the MQTT Broker can be instantiated on a dedicated server, which allows users to use both the training application for trainees and the control of the application by the trainers remotely without having to be in the same place where the simulator is being executed.

Acknowledgments

This work is co-financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 under the PORTUGAL 2020 Partnership Agreement, and through the Portuguese National Innovation Agency (ANI) as a part of project "SMARTCUT - Diagnóstico e Manutenção Remota e Simuladores para Formação de operação e manutenção de Máquinas Florestais: POCI-01-0247-FEDER-048183"

References

1. Melesse, T.Y., Di Pasquale, V., Riemma, S.: Digital twin models in industrial operations: State-of-the-art and future research directions. IET Collaborative Intelligent Manufacturing **3**(1), 37–47 (2021). DOI https://doi.org/10.1049/cim2.12010
2. Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., Sui, F.: Digital twin-driven product design, manufacturing and service with big data. The International Journal of Advanced Manufacturing Technology **94**(9), 3563–3576 (2018). DOI https://doi.org/10.1007/s00170-017-0233-1
3. Bergs, T., Gierlings, S., Auerbach, T., Klink, A., Schraknepper, D., Augspurger, T.: The Concept of Digital Twin and Digital Shadow in Manufacturing. Procedia CIRP **101**, 81–84 (2021). DOI 10.1016/j.procir.2021.02.010
4. Seppälä, L.: Data-driven shipbuilding. https://www.cadmatic.com/en/resources/publications-and-brochures/cadmatic-data-driven-shipbuilding-article-collection.pdf
5. Sepasgozar, S.: Differentiating digital twin from digital shadow: Elucidating a paradigm shift to expedite a smart, sustainable built environment. Buildings **11**, 151 (2021). DOI 10.3390/buildings11040151
6. Wheeler, A.: Understanding Virtual Reality Headsets (2016). URL https://www.engineering.com/story/understanding-virtual-reality-headsets
7. Sherman, W.R., Craig, A.B.: Understanding virtual reality—interface, application, and design. Presence **12**(4), 441–442 (2003). DOI 10.1162/105474603322391668
8. Muhanna, M.A.: Virtual reality and the CAVE: Taxonomy, interaction challenges and research directions. Journal of King Saud University - Computer and Information Sciences **27**(3), 344–361 (2015). DOI 10.1016/j.jksuci.2014.03.023
9. Virtual reality: Definition and requirements (2022). URL https://www.nas.nasa.gov/Software/VWT/vr.html. Accessed: 2022-02-19
10. Radianti, J., Majchrzak, T.A., Fromm, J., Wohlgenannt, I.: A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. Computers Education **147**, 103,778 (2020). DOI https://doi.org/10.1016/j.compedu.2019.103778
11. Mandal, S.: Brief Introduction of Virtual Reality and its Challenges. International Journal of Scientific and Engineering Research **4**(4), 304–309 (2013)
12. Zhu, Z., Liu, C., Xu, X.: Visualisation of the digital twin data in manufacturing by using augmented reality. Procedia CIRP **81**, 898–903 (2019). DOI https://doi.org/10.1016/j.procir.2019.03.223
13. Virtual assembly line training (2022). URL https://www.seriousgames.net/en/portfolio/opel-virtual-assembly-line-training/. Accessed: 2022-07-13
14. Flexsim (2022). URL https://www.flexsim.com/. Accessed: 2022-07-13
15. Martin, R.C.: Design principles and design patterns. Object Mentor **1**(34), 597 (2000)