



**HAL**  
open science

# Immersive Innovation: Exploring Interactive Virtual Reality through Distributed Simulations

Jalal Possik, Adriano Solis, Grégory Zacharewicz

► **To cite this version:**

Jalal Possik, Adriano Solis, Grégory Zacharewicz. Immersive Innovation: Exploring Interactive Virtual Reality through Distributed Simulations. Simulation Modeling - Recent Advances, New Perspectives, and Applications [Working Title], 2023, 978-1-83769-705-2. 10.5772/intechopen.1002004 . hal-04180841

**HAL Id: hal-04180841**

**<https://imt-mines-ales.hal.science/hal-04180841>**

Submitted on 14 Aug 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

## Chapter

# Immersive Innovation: Exploring Interactive Virtual Reality through Distributed Simulations

*Jalal Possik, Adriano O. Solis and Gregory Zacharewicz*

## Abstract

This chapter explores the significant role of modeling and simulation techniques in various sectors, focusing particularly on distributed simulation (DS). The increasing importance of DS has been emphasized in response to evolving industrial, healthcare, and services settings. By leveraging DS, the integration of heterogeneous simulations enhances the effectiveness and efficiency of individual and classical simulations. In particular, this chapter introduces a DS that seamlessly combines two distinct simulation methods within a virtual reality (VR) environment. This integration enables users to fully immerse themselves in a 3D digital twin environment. Two case studies were conducted to evaluate the effectiveness of the developed DS system. The first case study focused on the implementation of DS in a hemodialysis unit, while the second case study examined its application in an intensive care unit. AnyLogic has been utilized for developing both discrete event and agent-based simulations, while the Unity platform has been employed for VR environment creation. In order to ensure smooth integration and synchronization, as well as address the demanding computational requirements, a network-based DS system has been implemented based on the high-level architecture—an IEEE standard for DS.

**Keywords:** distributed simulation, discrete event simulation, agent-based modeling and simulation, virtual reality, immersive technology

## 1. Introduction

Modeling and simulation (M&S) involves techniques widely used in various fields to enhance analysis and decision-making, improve system performance, as well as optimize operations. These techniques provide valuable insights into complex real-world systems by creating representations and conducting virtual experiments. While M&S offers numerous advantages, it does come with limitations, especially when dealing with complex models that require high computational resources and extensive interconnections between different components. To address the complexities associated with intricate models, the concept of distributed simulation (DS) has emerged as a promising solution in recent years. DS involves the integration of heterogeneous simulations, which serves to enhance the usability and efficiency of individual

simulation components. This collaborative approach enables the synchronization and coordination of multiple simulation elements across diverse systems and platforms. By embracing DS, researchers and practitioners can effectively tackle challenges posed by complex models while leveraging the benefits of distributed computing and communication.

In order to harness the advantages of DS, the adoption of the high-level architecture (HLA) has become commonplace as a recognized international standard for facilitating interoperability and integration within DS environments. The HLA for DS emerged as a pivotal development in the field of computer simulation during the late 1990s. It was created as an open standard by the U.S. Department of Defense to address the increasing complexity and interoperability challenges associated with large-scale DSs. The origins of HLA can be traced back to the need for efficient simulation of complex systems such as military operations, where multiple simulations running on different computers need to seamlessly interact and exchange information in real-time. The HLA framework provides a standardized set of rules and protocols for achieving this interoperability, enabling simulations developed by different organizations, using different software and hardware, to work together seamlessly. By defining a common language for simulation components to communicate, HLA facilitates the integration and synchronization of diverse simulations, enabling the creation of comprehensive and realistic virtual environments. The HLA has been widely adopted in various domains, including defense, aerospace, transportation, and healthcare, among others, to support collaborative and DS environments (e.g., see [1–6]). Its open and extensible nature has contributed to its longevity and widespread use, making it a foundational framework in the field of DS. HLA enables the seamless integration of simulations developed using diverse tools and technologies. This integration allows for smooth communication and coordination among the DS components, promoting effective collaboration and ensuring efficient interoperability within the DS ecosystem.

In this chapter, we introduce a pioneering methodology that amplifies the interactive capabilities of virtual reality (VR) environments through the integration of M&S techniques using the DS paradigm. Our approach capitalizes on the utilization of the AnyLogic simulation software [7] in conjunction with the Unity platform [8] for the development of immersive digital twin VR environments. Through the seamless integration of these tools using DS principles, and leveraging the powerful capabilities of HLA, we showcase the seamless fusion of discrete event and agent-based simulations (ABSs) with VR.

The unique aspect of our approach lies in its application to two distinct case studies: a hemodialysis unit and an intensive care unit (ICU) both situated in a hospital located in Toronto, Canada. Staff members and trainees can immerse themselves in the VR environment using VR headsets, allowing them to explore and interact with a digital twin of the respective unit's environment. This integration of M&S and VR technology using DS opens up new possibilities for training, experimentation, and decision support in healthcare settings.

In the following sections, we will provide an overview of the advantages and limitations of M&S, discuss the concept of DS and its use of HLA, detail the integration of AnyLogic M&S with the Unity VR environment, and present the results and insights obtained from the case studies conducted in the hemodialysis unit and the ICU. Overall, this chapter serves to underscore the immense potential of integrating DS with VR technology in order to create captivating and interactive environments tailored for training and decision-making within complex systems, such as healthcare

units. By merging the collaborative nature of DS with the immersive capabilities of VR, new avenues are opened for enhancing user engagement and facilitating realistic experiences in critical domains like healthcare. The seamless integration of DS and VR holds promise for revolutionizing training methodologies and augmenting decision-making processes within complex systems.

## 2. State of the art

Classical M&S techniques have been widely employed in various fields to enhance performance, ensure safety, refine processes, validate theories, train personnel, and explore new scenarios. By scientifically modeling systems, users can gain valuable insights into the outcomes of different conditions and actions. Simulation provides a powerful tool for fine-tuning performance metrics, identifying optimization opportunities, and predicting the consequences of potential interventions. Moreover, simulations can help uncover hidden patterns, assess risk factors, and support decision-making processes by providing a virtual environment for experimentation and analysis. This enables researchers, engineers, and practitioners to make informed choices, evaluate alternative strategies, and improve overall understanding of complex systems. From engineering and manufacturing to healthcare and finance, simulation techniques play a vital role in driving innovation, advancing knowledge, and achieving desired outcomes. By simulating real-world scenarios and exploring a multitude of possibilities, classical M&S techniques empower professionals to uncover new insights, optimize processes, and foster continuous improvement in a wide range of domains. However, the escalating complexity of these models has exposed the shortcomings of traditional simulation approaches in capturing the intricate details and nuances inherent in these intricate systems. Numerous studies in the literature have acknowledged the limitations of classical simulation techniques when dealing with highly intricate models and have put forth alternative solutions to tackle this challenge.

One prevalent solution that has gained traction in addressing the limitations of classical simulation techniques for complex models is the adoption of DS methodologies. DS involves integrating multiple simulations to enhance the effectiveness and efficiency of individual simulations by facilitating collaboration and synchronization of heterogeneous simulation components across different systems or platforms. Researchers have highlighted the advantages of DS in the literature. For instance, the application of DS in simulating large-scale manufacturing systems has been investigated, emphasizing its ability to distribute computational load across multiple machines and improve simulation performance while reducing the need for expensive server scaling [9, 10]. Similarly, the use of DS in simulating complex transportation networks, leveraging existing computational resources effectively and enhancing the efficiency of simulating large-scale systems, has been demonstrated [11]. The application of DS in healthcare systems has been studied in [12], showcasing how DS enables the integration of disparate simulation models to create a comprehensive representation of the healthcare environment. These studies [9–12] exemplify the potential benefits of DS in simulating complex models in manufacturing, transportation, and healthcare. DS offers a cost-effective approach by leveraging distributed resources, enabling scalability, and effectively handling large-scale systems across various domains. There are other domains where DS has also proven to be a valuable technology, such as in the defense [13, 14] and disaster management [15] domains.

The aforementioned studies provide valuable insights into the utilization of DS in various domains, underscoring its significance in addressing complex real-world challenges while facilitating informed decision-making.

The integration of M&S with VR technology through DS has garnered attention in numerous studies for simulating complex models. Researchers have explored combining discrete event simulation (DES) with VR using DS in various research endeavors, highlighting the synergistic benefits of this integration. These studies demonstrate the seamless fusion of DES with VR, enabling users to immerse themselves in a virtual environment and interact with simulated models.

For instance, in [16], the integration of DES with VR for simulating manufacturing processes has been investigated. A DS framework that combined the capabilities of DES with the immersive experience of VR was developed. This integration enabled decision-makers and trainees to navigate and interact with the virtual environment, gaining valuable insights into complex manufacturing systems. Similarly, a cross-disciplinary team of researchers focused on integrating DES with VR for simulating healthcare scenarios [17]. They developed a DS system that combined patient flow simulations with a VR environment, particularly in a hospital hemodialysis unit. This integration provided an immersive and interactive experience, allowing healthcare professionals to explore different scenarios, evaluate system performance, and make informed decisions in a virtual setting. Furthermore, a design method using DES and VR for hot cell facility design has been developed [18]. The integrated DES-VR platform facilitated collaborative process simulation and design evaluation, improving material flow and obstacle checking compared to traditional approaches. The methodology proved effective for designing teleoperation-based nuclear facilities with varied equipment functions, structures, and degrees of automation. In summary, various studies have explored the integration of DS with VR, showcasing the seamless fusion of DES with VR to simulate complex models. These works [16–18] exemplify the immersive and interactive capabilities of the integration. By enabling users to explore and experience complex systems in a realistic and intuitive manner, the integration of DS with VR empowers decision-makers and trainees to gain valuable insights and make informed decisions.

The existing literature acknowledges the limitations of classical simulation techniques in dealing with complex models. As a result, researchers and practitioners have sought alternative solutions, with DS emerging as a promising approach. DS offers scalability and improved efficiency by leveraging the cost-effective distribution of resources, surpassing traditional simulation methods. Additionally, the integration of DS with VR presents significant potential, seamlessly combining DES with immersive VR environments. This integration enhances interactivity and realism in complex model simulations, opening new avenues for training and decision-making across various domains.

While previous studies have explored the benefits of integrating DES with VR for various applications, there is a notable gap in research regarding the integration of agent-ABS with DES and VR. ABS, which emerged in the 1990s, focuses on modeling individual agent behavior within a simulation, enabling a comprehensive representation of complex systems [19]. By incorporating ABS into the DES-VR framework, the dynamic interactions and decision-making processes of agents can be accurately captured, resulting in a more realistic simulation. This integration has the potential to significantly enhance the fidelity and accuracy of simulations by considering agent behavior and interactions within the virtual environment. The combination of these three technologies to create an immersive VR representation of a digital twin

environment has thus far remained largely unexplored. This integration holds great promise in allowing users to not only interact with simulated models but also immerse themselves in a VR environment that faithfully reflects real-world dynamics. By leveraging the strengths of ABS, DES, and VR, researchers can usher in a new era of simulation, enabling users to gain deeper insights into complex systems and make informed decisions based on the digital twin representation. In this chapter, we present the methodology for unlocking the full potential of this integrated approach and demonstrate its applications within healthcare settings.

### **3. Materials and methods**

The integration of DES, ABS, and VR represents a significant advancement in the field of simulation. This innovative approach aims to deliver a comprehensive and immersive simulation experience by leveraging the strengths of each paradigm. With DES, the complex and dynamic nature of events and system interactions can be accurately captured and analyzed, providing valuable insights into the behavior of the simulated system. Agent-based modeling assumes a crucial role in modeling the behaviours and decision-making processes of individual agents within the simulated environment, adding a layer of realism and complexity to the simulation. Furthermore, VR technology takes the experience to new heights by creating an interactive and immersive environment, enabling users to engage with the simulation in a more intuitive and realistic manner.

By carefully integrating these paradigms, this groundbreaking work seeks to unlock unparalleled potential in the realm of simulation. It provides a dynamic platform for exploring and understanding complex systems while enhancing decision-making processes. However, achieving seamless integration and interoperability among these heterogeneous components poses a significant challenge. This is where the HLA comes into play as the orchestrator. HLA acts as the mediator, coordinating the operations of the three components and facilitating data synchronization. It addresses interoperability issues at various levels, including data, middleware, operating system, and application heterogeneities. Through the efficient orchestration of these components, HLA ensures a cohesive and synchronized simulation experience, enabling researchers and practitioners to harness the full capabilities of this integrated simulation framework.

This integration has undergone extensive testing in two distinct healthcare environments within a hospital located in Toronto. Specifically, the first case involves the integration within a hemodialysis unit, while the second case focuses on the integration within an ICU. In the subsequent section, we delve into the technical aspects of integrating these three aforementioned technologies in these two unique healthcare settings.

#### **3.1 High level architecture**

The initial iteration of the HLA standard was introduced as HLA US DoD 1.3. Subsequently, it gained recognition from IEEE in 2000 and was adopted and renamed as HLA IEEE 1516 [1, 2, 4]. In 2010, the standard underwent revisions and updates, incorporating significant improvements, leading to the version known as HLA Evolved. Recently, the much-anticipated release of HLA4 made its debut, marking a significant milestone in the field of M&S. This latest version builds upon the

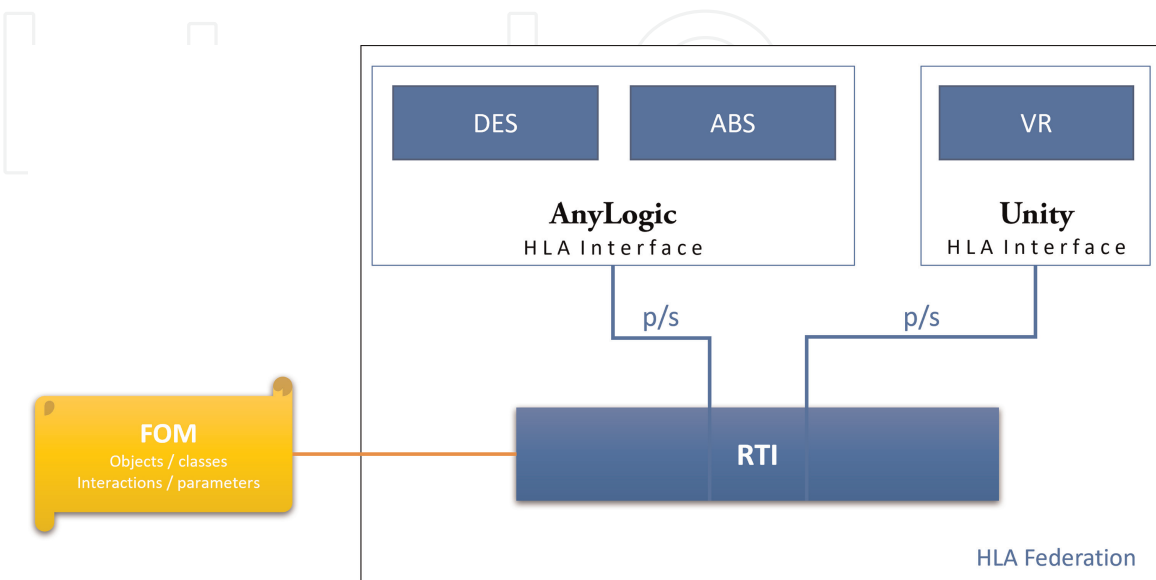
foundation of its predecessors, incorporating new and advanced features to further enhance the capabilities of the HLA standard.

The HLA standard plays a pivotal role in advancing the field of DS by enabling the creation of simulations comprising diverse simulation components, referred to as "Federates." A federation consists of multiple federates, a runtime infrastructure (RTI), and a federated object model (FOM). The RTI provides a standardized set of services for data exchange, synchronization, and collaborative management. On the other hand, the FOM defines the objects/classes and interactions/parameters utilized for communication.

The primary objective of HLA is to ensure M&S compatibility and facilitate the reuse of models. These models are executed on separate computers equipped with different operating systems, implemented in various programming languages, and connected through a local area network (LAN) or wide area network (WAN). Despite these differences, all these components seamlessly integrate within a federation. Leveraging a publish/subscribe (p/s) mechanism based on FOM and HLA object management, these heterogeneous federates can efficiently exchange and synchronize simulation data.

**Figure 1** illustrates the comprehensive global system architecture of the DS system. Within this architecture, two key components, namely DES and ABS, are utilized in the AnyLogic platform. These components seamlessly communicate with the VR environment, which is built using Unity.

In this integrated system, AnyLogic plays a crucial role by publishing the positions of each agent in the unit present within the simulation. This real-time tracking of agent positions allows for the precise monitoring of their movements and interactions within the simulated healthcare unit. Simultaneously, Unity, as part of the system, publishes the particle system results of these agents. This information is critical for AnyLogic, as it enables the real-time calculation of statistical studies related to COVID-19 transmission within the healthcare unit. Based on the p/s mechanism of the HLA standard, AnyLogic publishes the necessary information to the RTI. The RTI ensures that all subscribed components receive the specific data at a synchronized time. Similarly, Unity follows the same process, publishing relevant information to the



**Figure 1.**  
DS system architecture.

RTI for dissemination. To enable communication between two distinct components, or federates, with different characteristics, an HLA interface has been specifically configured. This interface plays a vital role as an intermediary between the RTI and the HLA federate. By serving as a link, it facilitates seamless interaction and data exchange between these components, despite their heterogeneity.

In this scenario, time regulation and constraints are disabled, indicating that the time management mechanism of HLA is not currently being utilized. This means that the system does not enforce strict synchronization of time between the federates. However, this does not hinder the seamless communication between AnyLogic and Unity. Whenever AnyLogic or Unity publishes an object/class or interaction/parameter, the RTI triggers a callback function, transmitting the corresponding data to the connected federate. The Java pRTI library has been used to develop the HLA integration. The Java pRTI library served as an effective facilitator, offering a range of methods tailored to support the HLA-related functionalities. By employing this library, we were able to expedite the development process and enhance the efficiency of the HLA implementation. Algorithm 1 outlines the steps required to exchange data between AnyLogic and Unity using the PRTI library. AnyLogic sends agent positions to Unity, while Unity sends virus particle data to AnyLogic, enabling synchronized visualization and interaction between the two systems.

---

**Algorithm 1:** HLA configuration and Data Exchange Between AnyLogic and Unity

---

**Input:** AnyLogic model and Unity project with PRTI library integration

**Output:** Data exchange between AnyLogic and Unity

### **I. Initialize AnyLogic-Unity Connectivity**

1. Import PRTI library into AnyLogic model and Unity project.
2. Initialize variables: **federationExecutionName**, **federateName**, **simulationTime**.

### **II. Create and Connect to HLA Federation**

1. Create an instance of the RTI ambassador using the PRTI library in AnyLogic and Unity.
2. Connect AnyLogic and Unity to the HLA federation:
  - a. Call the **connect()** function of the RTI ambassador in AnyLogic, passing **federationExecutionName**, **federateName**, and relevant parameters.
  - b. Call the **connect()** function of the RTI ambassador in Unity, passing **federationExecutionName**, **federateName**, and relevant parameters.

### **III. Join the Federation as Federates**

1. Join the federation as federates:



- a. Call the **joinFederationExecution()** function of the RTI ambassador in AnyLogic, specifying **federationExecutionName**, **federateName**, and additional parameters.
- b. Call the **joinFederationExecution()** function of the RTI ambassador in Unity, specifying **federationExecutionName**, **federateName**, and additional parameters.

#### **IV. Subscribe and Publish Object Classes/Attributes**

##### 1. AnyLogic:

- a. Subscribe to Unity's virus particle attributes:
  - Call the **subscribeObjectClassAttributes()** function of the RTI ambassador in AnyLogic, providing the appropriate object class and attribute handles for virus particles.
- b. Publish AnyLogic's agent positions:
  - Call the **publishObjectClassAttributes()** function of the RTI ambassador in AnyLogic, specifying the object class and attribute handles for agent positions.

##### 2. Unity:

- a. Subscribe to AnyLogic's agent position attributes:
  - Call the **subscribeObjectClassAttributes()** function of the RTI ambassador in Unity, providing the appropriate object class and attribute handles for agent positions.
- b. Publish Unity's virus particle attributes:
  - Call the **publishObjectClassAttributes()** function of the RTI ambassador in Unity, specifying the object class and attribute handles for virus particles.

#### **V. Enter Main Simulation Loop**

##### 1. while simulation is not complete:

###### a. Update Simulation Time

- Update simulation time based on AnyLogic model time or Unity game time.

###### b. Send Agent Position Updates from AnyLogic to Unity

- Get agent positions from AnyLogic model.

- Call the **updateAttributeValues()** function of the RTI ambassador in AnyLogic, passing the object and attribute handles for agent positions and the updated values.

#### c. Send Virus Particle Updates from Unity to AnyLogic

- Get virus particle data from Unity.
- Call the **updateAttributeValues()** function of the RTI ambassador in Unity, passing the object and attribute handles for virus particles and the updated values.

#### d. Receive Agent Position Updates in Unity

- Call the **tick()** function of the RTI ambassador in Unity to receive the latest data for subscribed agent positions.
- Process the received updates and update the corresponding objects in Unity.

#### e. Receive Virus Particle Updates in AnyLogic

- Call the **tick()** function of the RTI ambassador in AnyLogic to receive the latest data for subscribed virus particles.
- Process the received updates and update the corresponding objects in AnyLogic.

### VI. Disconnect from HLA Federation

#### 1. Disconnect from the HLA federation:

- a. Call the **resignFederationExecution()** function of the RTI ambassador in AnyLogic.
- b. Call the **resignFederationExecution()** function of the RTI ambassador in Unity.

### VII. Cleanup Resources

1. Release the resources used by the RTI ambassador and the PRTI library in AnyLogic and Unity.

---

This bidirectional communication ensures that relevant information is shared efficiently between AnyLogic and Unity, enabling both components to work in harmony within the DS system. By analyzing the data collected from both AnyLogic and Unity, the DS system facilitates the determination of important metrics such as the contact matrix in the healthcare unit. The contact matrix provides valuable insights into the rate of contacts between agents, aiding in the identification of potential hotspots for

disease transmission. This information proves vital for hospital managers as it helps pinpoint the sources of disease transmission within the healthcare unit.

Equipped with this knowledge, hospital managers can take proactive measures to control and mitigate the spread of diseases. By identifying high-risk areas and focusing efforts on implementing targeted interventions, such as improved hygiene protocols or physical layout modifications, hospitals can effectively minimize the impact of disease transmission. Ultimately, this integrated DS system empowers healthcare facilities to make informed decisions based on accurate data, leading to improved patient safety and enhanced overall healthcare outcomes.

### **3.2 Discrete event simulation**

DES is a widely utilized computational modeling approach employed to replicate the dynamic behavior and operational performance of systems across a span of time. It entails representing the system as a sequence of distinct events that take place at precise points in time, effectively capturing the interplay and interdependencies between various entities and resources. Each event signifies a noteworthy alteration in the system's state, such as an arrival, departure, or completion of a task. Through simulating the system's behavior under diverse scenarios and parameter configurations, DES enables the analysis, optimization, and evaluation of intricate systems without the need for resource-intensive and time-consuming real-world experimentation.

#### *3.2.1 Hemodialysis case study*

The DES technique was applied within the hemodialysis unit to model the dynamic processes and interactions involving hemodialysis patients, nephrologists and other healthcare professionals, and available resources. The simulation encompassed various discrete events, such as patient arrivals, assignment to hemodialysis machines, duration of dialysis sessions, and other relevant activities. Through accurate representation of event timings and dependencies, the simulation provided valuable insights into the unit's performance metrics, including patient waiting times, utilization rates of hemodialysis machines, and overall operational efficiency. By exploring diverse scenarios and resource allocation strategies, the DES facilitated the evaluation of different approaches and identified potential enhancements to optimize the effectiveness of the hemodialysis unit and enhance patient care. Detailed implementation of the hemodialysis model within the DES system, including entities, resources, parameters, assumptions, and model dynamics, is elaborated in **Table 1**.

#### *3.2.2 ICU case study*

In the context of the ICU management system, DES was leveraged to simulate the dynamic processes and interactions involving patients, healthcare professionals, and the available resources. The discrete events encompassed patient arrivals, consultations with medical professionals, administration of treatments, diagnostic procedures, and other relevant activities. By accurately modeling the timing and interdependencies of these events, the simulation provided valuable insights into the system's performance metrics, including patient waiting times, utilization rates of healthcare resources, and overall operational efficiency. Through the exploration of different scenarios and resource allocation strategies, the DES enabled the evaluation of various approaches and the identification of potential improvements to enhance the

---

Entities:

---

- a. Patients: Individuals requiring hemodialysis treatment.
  - b. Nurses: Healthcare professionals responsible for administering treatment and monitoring patients.
  - c. Dialysis machines: Medical devices used for hemodialysis treatment.
  - d. Supplies: Consumables and equipment necessary for the process.
  - e. MRP (medical renal physicians): Specialists responsible for overseeing the treatment plan and making medical decisions.
  - f. UT nephrologists: Nephrologists affiliated with the University Hospital responsible for consultation and treatment oversight.
  - g. Residents: Medical residents involved in the patient care process.
  - h. Housekeepers: Staff responsible for maintaining cleanliness and order in the unit.
- 

Resources:

---

- a. Nurse capacity ( $C_{\text{nurse}}$ ): The number of available nurses.
  - b. Dialysis machine capacity ( $C_{\text{machine}}$ ): The number of available dialysis machines.
  - c. Supplies availability: The availability of necessary supplies and equipment.
  - d. MRP availability: The availability of MRP for treatment plan review.
  - e. UT nephrologist availability: The availability of UT Nephrologists for consultation and treatment oversight.
  - f. Resident availability: The availability of residents for assisting in patient care.
  - g. Housekeeper capacity ( $C_{\text{housekeeper}}$ ): The number of available housekeepers.
- 

Parameters and variables:

---

- a. Arrival rate ( $\lambda$ ): The average rate at which patients arrive at the unit.
  - b. Service rate ( $\mu$ ): The average time it takes to complete a hemodialysis treatment for a patient.
  - c. Consultation time ( $\tau$ ): The average time taken by a UT Nephrologist to consult and provide oversight.
  - d. Treatment plan review time ( $\theta$ ): The average time taken by an MRP to review and update the treatment plan.
  - e. Nurse availability ( $A_{\text{nurse}}$ ): The proportion of time nurses are busy.
  - f. Dialysis machine availability ( $A_{\text{machine}}$ ): The proportion of time dialysis machines are busy.
  - g. MRP availability ( $A_{\text{mrp}}$ ): The proportion of time MRP are busy.
  - h. UT nephrologist availability ( $A_{\text{nephrologist}}$ ): The proportion of time UT Nephrologists are busy.
  - i. Resident availability ( $A_{\text{resident}}$ ): The proportion of time residents are busy.
  - j. Housekeeper availability ( $A_{\text{housekeeper}}$ ): The proportion of time housekeepers are busy.
- 

Model assumptions:

---

- a. Patients arrive according to a Poisson process with an average arrival rate of  $\lambda$ .
  - b. The service time for each patient follows an exponential distribution with a mean of  $1/\mu$ .
  - c. The consultation time for each patient with a UT Nephrologist follows an exponential distribution with a mean of  $1/\tau$ .
  - d. The treatment plan review time for each patient by an MRP follows an exponential distribution with a mean of  $1/\theta$ .
  - e. Patients join the corresponding queues when resources are not immediately available.
  - f. The number of nurses, dialysis machines, supplies, MRP, UT nephrologists, residents, and housekeepers remain constant throughout the simulation.
- 

Model dynamics:

---

Initialization:

- Set simulation time to  $t = 0$ .
  - Generate the first patient arrival time based on the exponential distribution.
- 

Patient arrival process:

---

- At time  $t$ , a patient arrives.
  - If there is an available UT nephrologist for consultation:
    - The patient consults the UT nephrologist and receives treatment oversight.
    - Calculate the treatment completion time for the patient based on the service time.
-

- 
- Update the availability of the UT nephrologist.
  - If there are available nurses, dialysis machines, and supplies:
    - Assign a nurse, a dialysis machine, and necessary supplies to the patient.
    - Update the availability of the assigned resources.
  - If any of the resources are not available:
    - The patient joins the corresponding queues.
- 

Treatment process:

- 
- When a nurse, dialysis machine, or supplies become available:
    - Assign the available resource to the patient at the front of the corresponding queue (if any).
    - Start the treatment for the patient, setting the treatment completion time based on the service time.
    - Update the availability of the assigned resource.
- 

Consultation process:

- 
- When a UT nephrologist becomes available:
    - Assign the UT nephrologist to the patient at the front of the consultation queue (if any).
    - Conduct the consultation and provide treatment oversight.
    - Update the availability of the UT nephrologist.
  - If there are available nurses, dialysis machines, and supplies:
    - Assign the necessary resources to the patient.
    - Update the availability of the assigned resources.
  - If any of the resources are not available:
    - The patient joins the corresponding queues.
- 

Treatment plan review process:

- 
- When an MRP becomes available:
    - Assign the MRP to the patient at the front of the treatment plan review queue (if any).
    - Conduct the treatment plan review and update the treatment plan if necessary.
    - Update the availability of the MRP.
  - If there are available UT nephrologists, nurses, dialysis machines, and supplies:
    - Assign the necessary resources to the patient.
    - Update the availability of the assigned resources.
  - If any of the resources are not available:
    - The patient joins the corresponding queues.
- 

Resident support process:

- 
- When a resident becomes available:
    - Assign the resident to assist in patient care for the patient at the front of the corresponding queue (if any).
    - Assist in patient care as directed by the nurses, UT nephrologists, or MRP.
    - Update the availability of the resident.
- 

Treatment completion:

- 
- When the treatment completion time for a patient is reached:
    - Release the nurse, dialysis machine, and supplies assigned to the patient.
    - Update the availability of the released resources.
    - If there are patients in the consultation, treatment plan review, or resident support queues:
      - Assign the next available resources to the respective patients.
      - Start the corresponding process for the assigned patients.
- 

Housekeeping process:

- 
- Housekeepers continuously monitor and maintain cleanliness and order in the unit.
  - Update the availability of the housekeepers.
- 

Performance metrics:

- 
- Track and record the following metrics during the simulation:
    - Average patient waiting time in the queues.
    - Average patient treatment time.
-

<ul style="list-style-type: none"> <li>◦ Resource utilization rates (nurses, dialysis machines, MRP, UT nephrologists, residents, housekeepers).</li> </ul>
Simulation termination:
<ul style="list-style-type: none"> <li>• Continue the simulation until a specified simulation time is reached or a predetermined number of patients have been treated.</li> </ul>
Data analysis:
<ul style="list-style-type: none"> <li>• Collect simulation outputs and calculate statistics for performance metrics.</li> <li>• Conduct sensitivity analysis and optimization techniques to improve the efficiency of the hemodialysis unit process.</li> </ul>

**Table 1.**  
*Hemodialysis model.*

effectiveness of the ICU management system and improve patient care. A comprehensive description of the ICU model implementation, including entities, resources, parameters, assumptions, and model dynamics, is provided in **Table 2**.

The DES played a crucial role as an initial step in the analysis, setting the foundation for further exploration using ABS. By simulating discrete events and their interdependencies, the DES provided valuable insights into system behavior and performance. However, to capture the intricate dynamics of individual agents, their behaviors, and decision-making processes, the subsequent subsection will delve into ABS. ABS offers a more nuanced approach by representing each entity as an autonomous agent, capable of interacting with other agents and responding to their environment. This advanced M&S technique enables a deeper understanding of emergent system behaviors, complex interactions, and the impact of individual agent actions. The following subsection will present the details of the agent-based modeling framework, shedding light on its advantages and its potential applications in the context of the system under study.

### 3.3 Agent-based simulation

In both the hemodialysis unit and ICU case studies, a unified agent-based model was employed to investigate disease transmission within the healthcare facility, with a particular focus on the challenging period characterized by the COVID-19 pandemic. This critical phase necessitated hospitals' proactive pursuit of effective measures to mitigate the spread of the virus and ensure the safety of patients and healthcare professionals. Leveraging the agent-based modeling approach, our objective was to gain valuable insights into the complex dynamics of disease transmission within these units while considering the unique characteristics of the pandemic context.

To effectively capture the behavior and interactions of human agents, including healthcare workers and patients, we implemented a hierarchical structure within the model. This hierarchical structure allowed for the extension and inheritance of human properties across agent types. Specifically, a specialized entity, referred to as the "human agent framework," served as the overarching representation of human characteristics and served as the foundation for modeling disease transmission.

Within the human agent framework, a dedicated module focused on disease transmission was developed. Drawing inspiration from established epidemiological models, we devised an expanded version of the SEIR (susceptible, exposed, infected,

---

Entities:

---

- a. Patients: Individuals requiring intensive care treatment, including cardiac patients.
  - b. Nurses: Healthcare professionals responsible for patient care and monitoring.
  - c. Doctors: Medical professionals responsible for diagnosing and prescribing treatment.
  - d. Respiratory therapists: Healthcare professionals specializing in respiratory care.
  - e. Pharmacists: Professionals responsible for medication management and dispensing.
  - f. Lab technicians: Personnel responsible for conducting laboratory tests.
  - g. Housekeepers: Staff responsible for maintaining cleanliness and order in the unit.
  - h. Medical equipment technicians: Personnel responsible for maintaining and repairing medical equipment.
  - i. Family members: Relatives or guardians of patients providing support and information.
- 

Resources:

---

- a. Nurse capacity ( $C_{\text{nurse}}$ ): The number of available nurses.
  - b. Doctor capacity ( $C_{\text{doctor}}$ ): The number of available doctors.
  - c. Respiratory therapist capacity ( $C_{\text{therapist}}$ ): The number of available respiratory therapists.
  - d. Pharmacist capacity ( $C_{\text{pharmacist}}$ ): The number of available pharmacists.
  - e. Lab technician capacity ( $C_{\text{technician}}$ ): The number of available lab technicians.
  - f. Equipment availability: The availability of medical equipment required for intensive care, including cardiac monitoring equipment.
  - g. Medication availability: The availability of necessary medications, including cardiac medications.
  - h. Housekeeper capacity ( $C_{\text{housekeeper}}$ ): The number of available housekeepers.
  - i. Technician capacity ( $C_{\text{technician}}$ ): The number of available medical equipment technicians.
- 

Parameters and variables:

---

- a. Arrival rate ( $\lambda$ ): The average rate at which patients arrive at the ICU, including cardiac patients.
  - b. Service rate ( $\mu$ ): The average time it takes to complete treatment for a patient.
  - c. Consultation time ( $\tau$ ): The average time taken by a doctor to consult and prescribe treatment.
  - d. Respiratory therapy time ( $\theta$ ): The average time taken by a respiratory therapist to administer respiratory care.
  - e. Medication dispensing time ( $\psi$ ): The average time taken by a pharmacist to dispense medication.
  - f. Laboratory test time ( $\omega$ ): The average time taken by a lab technician to conduct laboratory tests.
  - g. Nurse availability ( $A_{\text{nurse}}$ ): The proportion of time nurses are busy.
  - h. Doctor availability ( $A_{\text{doctor}}$ ): The proportion of time doctors are busy.
  - i. Respiratory therapist availability ( $A_{\text{therapist}}$ ): The proportion of time respiratory therapists are busy.
  - j. Pharmacist availability ( $A_{\text{pharmacist}}$ ): The proportion of time pharmacists are busy.
  - k. Lab technician availability ( $A_{\text{technician}}$ ): The proportion of time lab technicians are busy.
  - l. Housekeeper availability ( $A_{\text{housekeeper}}$ ): The proportion of time housekeepers are busy.
  - m. Technician availability ( $A_{\text{technician}}$ ): The proportion of time medical equipment technicians are busy.
- 

Model assumptions:

---

- a. Patients arrive according to a Poisson process with an average arrival rate of  $\lambda$ .
  - b. The service time for each patient follows an exponential distribution with a mean of  $1/\mu$ .
  - c. The consultation time for each patient with a doctor follows an exponential distribution with a mean of  $1/\tau$ .
  - d. The respiratory therapy time for each patient follows an exponential distribution with a mean of  $1/\theta$ .
  - e. The medication dispensing time for each patient follows an exponential distribution with a mean of  $1/\psi$ .
  - f. The laboratory test time for each patient follows an exponential distribution with a mean of  $1/\omega$ .
  - g. Patients join the corresponding queues when resources are not immediately available.
  - h. The number of nurses, doctors, respiratory therapists, pharmacists, lab technicians, housekeepers, and medical equipment technicians remains constant throughout the simulation.
- 

Model dynamics:

---

---

Initialization:

- Set simulation time to  $t = 0$ .
- Generate the first patient arrival time based on the exponential distribution.

---

Patient arrival process:

- At time  $t$ , a patient arrives, including cardiac patients.
- If there is an available doctor for consultation:
  - The patient consults the doctor and receives treatment prescription.
  - Calculate the treatment completion time for the patient based on the service time.
  - Update the availability of the doctor.
  - If there are available nurses, respiratory therapists, pharmacists, lab technicians, equipment, and medications:
    - Assign a nurse, a respiratory therapist, a pharmacist, a lab technician, necessary equipment, and medications to the patient.
    - Update the availability of the assigned resources.
  - If any of the resources are not available:
    - The patient joins the corresponding queues.

---

Treatment process:

- When a nurse, respiratory therapist, pharmacist, lab technician, equipment, or medication becomes available:
  - Assign the available resource to the patient at the front of the corresponding queue (if any).
  - Start the treatment for the patient, setting the treatment completion time based on the service time.
  - Update the availability of the assigned resource.

---

Consultation process:

- When a doctor becomes available:
  - Assign the doctor to the patient at the front of the consultation queue (if any).
  - Conduct the consultation and prescribe the treatment plan.
  - Update the availability of the doctor.

---

Respiratory therapy process:

- When a respiratory therapist becomes available:
  - Assign the respiratory therapist to the patient at the front of the respiratory therapy queue (if any).
  - Administer respiratory therapy to the patient.
  - Update the availability of the respiratory therapist.

---

Medication dispensing process:

- When a pharmacist becomes available:
  - Assign the pharmacist to the patient at the front of the medication dispensing queue (if any).
  - Dispense the required medications to the patient.
  - Update the availability of the pharmacist.

---

Laboratory testing process:

- When a lab technician becomes available:
  - Assign the lab technician to the patient at the front of the laboratory testing queue (if any).
  - Conduct the necessary laboratory tests for the patient.
  - Update the availability of the lab technician.

---

Treatment completion:

- When the treatment completion time for a patient is reached:
  - Release the nurse, respiratory therapist, pharmacist, lab technician, equipment, and medications assigned to the patient.
  - Update the availability of the released resources.
  - If there are patients in the consultation, respiratory therapy, medication dispensing, or laboratory testing queues:
    - Assign the next available resources to the respective patients.
    - Start the corresponding process for the assigned patients.

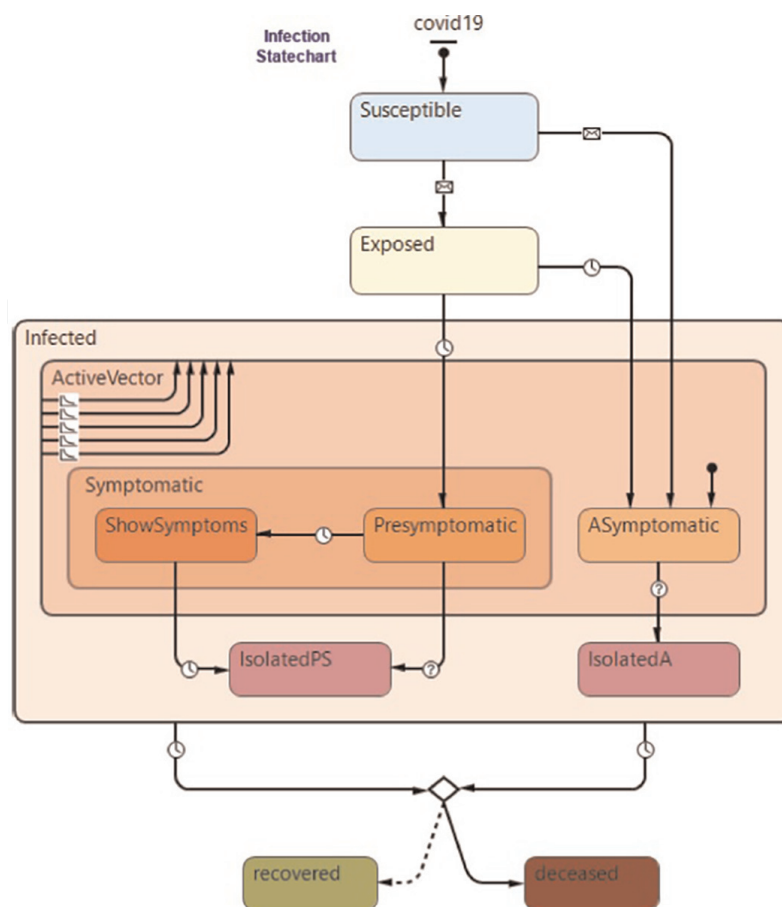


Housekeeping process:
<ul style="list-style-type: none"> <li>• Housekeepers continuously monitor and maintain cleanliness and order in the unit.</li> <li>• Update the availability of the housekeepers.</li> </ul>
Medical equipment maintenance process:
<ul style="list-style-type: none"> <li>• When a medical equipment technician becomes available: <ul style="list-style-type: none"> <li>◦ Assign the technician to inspect, maintain, or repair medical equipment as required.</li> <li>◦ Update the availability of the technician.</li> </ul> </li> </ul>
Family support process:
<ul style="list-style-type: none"> <li>• Family members of patients provide support and information as needed.</li> </ul>
Performance metrics:
<ul style="list-style-type: none"> <li>• Track and record the following metrics during the simulation: <ul style="list-style-type: none"> <li>◦ Average patient waiting time in the queues.</li> <li>◦ Average patient treatment time.</li> <li>◦ Resource utilization rates (nurses, doctors, respiratory therapists, pharmacists, lab technicians, housekeepers, medical equipment technicians).</li> </ul> </li> </ul>
Simulation termination:
<ul style="list-style-type: none"> <li>• Continue the simulation until a specified simulation time is reached or a predetermined number of patients have been treated.</li> </ul>
Data analysis:
<ul style="list-style-type: none"> <li>• Collect simulation outputs and calculate statistics for performance metrics.</li> <li>• Conduct sensitivity analysis and optimization techniques to improve the efficiency of the ICU process.</li> </ul>

**Table 2.**  
*ICU model.*

and recovered) model, as depicted in **Figure 2** [12]. This enhanced model incorporated additional features to account for asymptomatic and presymptomatic infected healthcare workers, who served as potential sources of viral contamination and transmission. Their interactions with susceptible individuals within the defined physical space of the model played a crucial role in simulating the spread of the disease. Similar modeling approaches have proven effective in other simulations, validating their applicability and relevance.

Tracking the different stages of infection within the human agent framework, including periods of active viral transmission, was accomplished through a comprehensive state chart. This state chart facilitated the precise monitoring of disease progression, enabling the identification of crucial time points when human agents became potential sources of contamination and transmission. Symptomatic healthcare workers underwent testing, and upon confirmation of a positive diagnosis, they were isolated to prevent further virus transmission. During the quarantine or recovery period, these symptomatic agents were designated as inactive viral vectors and temporarily replaced within the model. Conversely, for ICU patients, the virus transmission component remained active throughout the simulation, reflecting the assumption that the ICU ward exclusively admitted COVID-19 patients. Healthcare workers entering the state of active viral transmission were governed by the same state chart within the human agent framework, ensuring consistent modeling of their behavior. Furthermore, the model recorded the sources of contamination for exposed/infected medical staff agents, facilitating the identification of significant contributors to viral spread.



**Figure 2.**  
*Agent-based model for COVID 19 transmission inside the units.*

By utilizing this unified agent-based model, we aimed to provide a comprehensive understanding of disease transmission dynamics within both the hemodialysis unit and the ICU, particularly during the challenging period of the COVID-19 pandemic. The integration of advanced modeling techniques and the meticulous representation of human agent behaviors enabled us to evaluate various scenarios and devise strategies for improving infection control measures, optimizing patient care, and safeguarding the well-being of healthcare workers.

#### 4. Results and discussion

The aforementioned case studies involved building digital twin models specific to the units under investigation, enabling hypothesis testing and contextual exploration to understand their impact on system performance. To enhance the capabilities of the digital twin frameworks, we introduced a VR layer. This immersive layer comprises VR environments that faithfully replicate the physical units in a three-dimensional space. The VR environments showcase empty 3D models without agents, providing a platform to gain insights into unit behavior and dynamics across different scenarios.

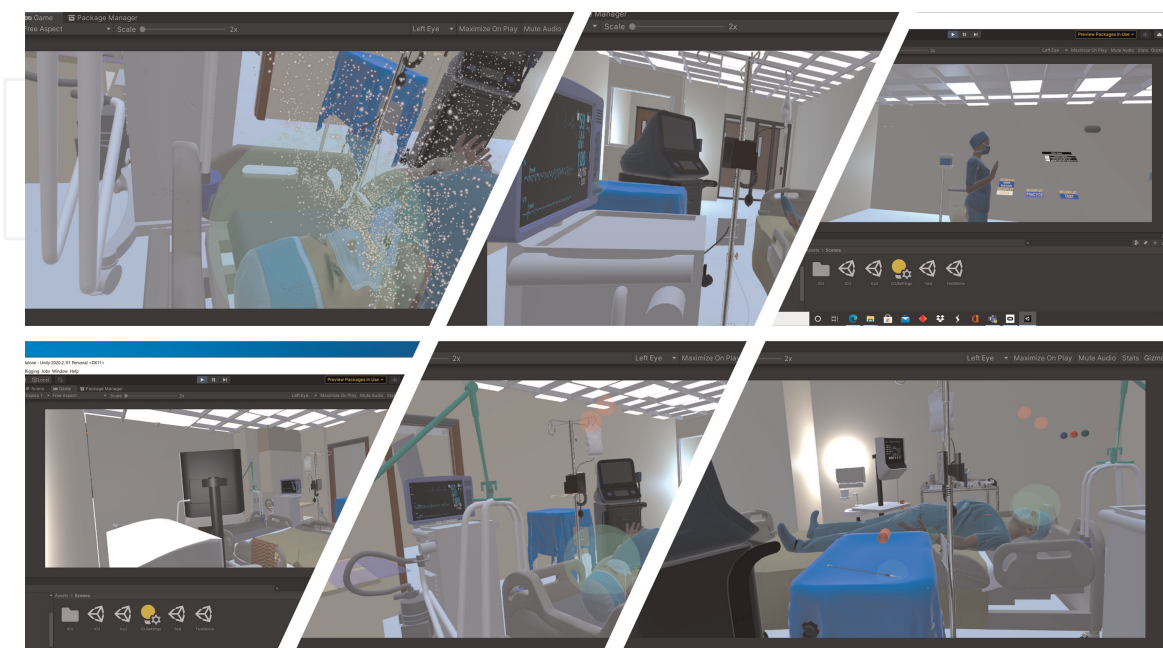
Integration between the DES system, ABS system, and the VR layer was achieved through the utilization of HLA. The DES system facilitates real-time agent position transmission, enabling us to receive crucial information such as agent type, unique

identifier, and position within the system. This data plays a pivotal role in accurately defining and locating agents within the VR model.

To achieve synchronization between the digital twin and the VR environment, we utilized Unity, a powerful game development engine. Leveraging the agent information obtained from the DES and ABS system, we imported the corresponding agent avatars into the VR environment. By aligning the agents' positions from the DES and ABS system with their virtual counterparts, we established a realistic and interactive experience, facilitating comprehensive monitoring and analysis of agent behavior within the simulated environment.

The immersion inside the units using VR headsets offers a unique and engaging experience, allowing researchers and stakeholders to gain a profound understanding of system dynamics. The VR environment provides a sense of presence and realism that traditional desktop simulations cannot replicate, enabling intuitive exploration and comprehension of complex system behaviors (**Figure 3**).

By immersing oneself in the virtual environment, users can visually inspect the units from various perspectives, observing intricate details of the spatial layout, environmental conditions, and architectural features. This level of immersion enhances contextual understanding, enabling researchers to identify potential design flaws, safety hazards, or operational inefficiencies that may not be evident in traditional simulations or physical environments. The interactive nature of the VR environment empowers researchers to manipulate objects, simulate interactions, and trigger events, replicating real-world scenarios within the digital twin. This interactivity facilitates hypothesis testing, validation of assumptions, and exploration of the consequences of different actions, leading to informed decision-making and system optimization. The immersive experience in the VR environment also enhances collaboration and communication among stakeholders. By donning VR headsets, researchers, domain experts, and decision-makers can convene within the same virtual space, inspect the units together, and discuss findings and insights. This collaborative environment promotes interdisciplinary discussions, knowledge sharing, and improved collective



**Figure 3.**  
*An immersive experience inside the ICU.*

understanding of the system, ultimately facilitating more effective decision-making processes.

While the immersion within the units using VR headsets offers significant advantages, there are considerations to address. Factors such as user comfort, mitigation of motion sickness, and the availability of high-quality VR hardware can impact the adoption and accessibility of this approach. Furthermore, ongoing advancements in VR technology should be considered to enhance the realism and fidelity of the virtual environment, further improving the accuracy and usefulness of the digital twin framework.

## 5. Conclusion

In conclusion, this research has made significant contributions to the advancement of digital twin frameworks by seamlessly integrating DES, ABS, VR based on HLA standard. The utilization of these technologies has allowed for the simulation and analysis of a wide range of systems, services, and processes in a scientifically rigorous manner. The construction of accurate digital twin models using DES and ABS has laid the groundwork for hypothesis testing, contextual exploration, and performance evaluation. DES enables the modeling of system dynamics based on discrete events, while ABS facilitates the representation of individual agent behaviors and interactions. This integration provides a comprehensive understanding of system-level behavior and the emergence of complex phenomena.

The inclusion of the VR layer within the digital twin framework has greatly enhanced the simulation experience by providing an immersive virtual environment. By faithfully replicating the physical units in a three-dimensional space, the VR environments enable researchers to visually and interactively explore the simulated systems, leading to a deeper understanding of their intricacies. The ability to experiment with different scenarios within the VR environment increases the realism and fidelity of the simulations, enabling more accurate analysis and decision-making. The seamless integration facilitated by HLA between the DES/ABS systems and the VR environment has enabled real-time data exchange, ensuring synchronization between the digital twin and the immersive virtual environment. By transmitting agent positions from the DES system to the VR layer, precise agent placement and visualization within the virtual environment are achieved, resulting in a coherent and lifelike representation of the system. This integration is crucial for tracking and analyzing agent behavior, providing valuable insights into system performance and facilitating evidence-based decision-making.

The scientific rigor and advancements presented in this research open up new avenues for further exploration and development of digital twin frameworks. Future research efforts can focus on refining the modeling capabilities of DES and ABS, incorporating more intricate agent behaviors and interactions. Furthermore, enhancing VR environments to include realistic avatars, environmental factors, and sensory feedback can further improve the fidelity and accuracy of the simulations. The practical implications of this research are substantial, particularly in domains such as healthcare, where digital twin frameworks can provide valuable insights into system performance, resource allocation, and optimization. By leveraging the combined power of DES, ABS, VR, and HLA, decision-makers can make informed choices, enhance operational efficiency, and mitigate potential risks.

In conclusion, this research highlights the potential of integrating DES, ABS, VR, and HLA within a digital twin framework for scientifically rigorous system simulation

and analysis. The advancements presented in this study lay a solid foundation for further research and application in various domains, with the potential to revolutionize decision-making and optimization processes in complex systems.

IntechOpen

### **Author details**

Jalal Possik<sup>1\*</sup>, Adriano O. Solis<sup>2</sup> and Gregory Zacharewicz<sup>3</sup>

1 Lille Catholic University, Lille, France


2 York University, Toronto, Canada

3 IMT Mines Alès, Alès, France

\*Address all correspondence to: [jalal.possik@univ-catholille.fr](mailto:jalal.possik@univ-catholille.fr)

### **IntechOpen**

---

© 2023 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

## References

- [1] Zeigler BP, Hall SB, Sarjoughian HS. Exploiting HLA and DEVS to promote interoperability and reuse in Lockheed's corporate environment. *Simulation*. 1999;**73**:288-295. DOI: 10.1177/003754979907300505
- [2] Zacharewicz G, Frydman C, Giambiasi N. G-DEVS/HLA environment for distributed simulations of workflows. *Simulation*. 2008;**84**:197-213. DOI: 10.1177/0037549708092833
- [3] Possik J, Amrani A, Zacharewicz G. Development of a co-simulation system as a decision-aid in Lean tools implementation. In: *Proceedings of the 50<sup>th</sup> Computer Simulation Multiconference (SummerSim '18)*; 9-12 July 2018; Bordeaux, France. San Diego, California: Society for Computer Simulation International; 2018. Article 21, pp. 1-12
- [4] Possik J. Contribution to a methodology and a co-simulation framework assessing the impact of lean on manufacturing performance [PhD thesis]. Bordeaux, France: University of Bordeaux; 2019
- [5] Possik J, D'Ambrogio A, Zacharewicz G, Amrani A, Vallespir B. A BPMN/HLA-based methodology for collaborative distributed DES. In: *Proceedings of the 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2019)*; 12-14 June 2019; Capri, Italy. Piscataway, NJ: IEEE; 2019. pp. 118-123. DOI: 10.1109/WETICE.2019.00033
- [6] Almaksour A, Gerges H, Gorecki S, Zacharewicz G, Possik, J. The use of the IEEE HLA standard to tackle interoperability issues between heterogeneous components. In: *Proceedings of the 2022 IEEE/ACM 26th International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2022)*; 26-28 September 2022; Alès, France. Piscataway, NJ: IEEE; 2022. pp. 175-178. DOI: 10.1109/DS-RT55542.2022.9932042
- [7] AnyLogic Company. AnyLogic Simulation Software [Internet]. 2022. Available from: <https://www.anylogic.com/> [Accessed: January 27, 2023]
- [8] Unity Technologies. Unity Simulation Pro [Internet]. 2022. Available from: <https://unity.com/> [Accessed: December 2, 2022]
- [9] Gorecki S, Possik J, Zacharewicz G, Ducq Y, Perry N. A multicomponent distributed framework for smart production system modeling and simulation. *Sustainability*. 2020;**12**:6969. DOI: 10.3390/su12176969
- [10] Possik J, Zouggar-Amrani A, Vallespir B, Zacharewicz G. Lean techniques impact evaluation methodology based on a co-simulation framework for manufacturing systems. *International Journal of Computer Integrated Manufacturing*. 2021;**35**:91-111. DOI: 10.1080/0951192X.2021.1972468
- [11] Ahmed, MS, Hoque MA. Partitioning of urban transportation networks utilizing real-world traffic parameters for distributed simulation in SUMO. In: *Proceedings of the 2016 IEEE Vehicular Networking Conference (VNC 2016)*; 08-10 December 2016; Columbus, OH, USA. Piscataway, NJ: IEEE; 2016. pp. 1-4. DOI: 10.1109/VNC.2016.7835942
- [12] Possik J, Asgary A, Solis AO, Zacharewicz G, Shafiee MA,

- Najafabadi MM, et al. An agent-based modeling and virtual reality application using distributed simulation: Case of a COVID-19 intensive care unit. *IEEE Transactions on Engineering Management*. 2022;**70**:2931-2943  
DOI: 10.1109/TEM.2022.3195813
- [13] Dahmann JS, Fujimoto RM, Weatherly RM. The Department of Defense high level architecture. In: *Proceedings of the 29th Conference on Winter Simulation*; 7-10 December 1997; Atlanta, GA, USA. New York, NY: Association for Computing Machinery; 1997. pp. 142-149. DOI: 10.1145/268437.268465
- [14] Kim TG, Sung CH, Hong S-Y, Hong JH, Choi CB, Kim JH, et al. DEVSim++ toolset for defense modeling and simulation and interoperability. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*. 2011;**8**:129-142.  
DOI: 10.1177/1548512910389203
- [15] Boukerche A. Smart disaster management and responses for smart cities: A new challenge for the next generation of distributed simulation systems. In: *Proceedings of the 2019 IEEE/ACM 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2019)*; 07-09 October 2019; Cosenza, Italy. Piscataway, NJ: IEEE; 2019. pp. 1-2.  
DOI: 10.1109/DS-RT47707.2019.8958693
- [16] Hutabarat W, Oyekan J, Turner C, Tiwari A, Prajapat N, Gan X-P, Waller A. Combining virtual reality enabled simulation with 3D scanning technologies towards smart manufacturing. In: *Proceedings of the 2016 Winter Simulation Conference (WSC 2016)*; 11-14 December 2016; Washington, DC, USA. Piscataway, NJ: IEEE; 2016. pp. 2774-2785. DOI: 10.1109/WSC.2016.7822314
- [17] Possik J, Gorecki S, Asgary A, Solis AO, Zacharewicz G, Tofighi M, Shafiee MA, Merchant AA, Aarabi M, Guimaraes A, Nadri N. A distributed simulation approach to integrate anylogic and unity for virtual reality applications: Case of covid-19 modelling and training in a dialysis unit. In: *Proceedings of the 2021 IEEE/ACM 25th International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2021)*; 27-29 September 2021; Valencia, Spain. Piscataway, NJ: IEEE; 2021. pp. 1-7. DOI: 10.1109/DS-RT52167.2021.9576149
- [18] Yu S, Han J. Virtual reality platform-based conceptual design and simulation of a hot cell facility. *The International Journal of Advanced Manufacturing Technology*. 2021;**116**:487-505.  
DOI: 10.1007/s00170-021-07139-7
- [19] Siebers PO, Macal CM, Garnett J, Buxton D, Pidd M. Discrete-event simulation is dead, long live agent-based simulation! *Journal of Simulation*. 2010; **4**:204-210. DOI: 10.1057/jos.2010.14