



Synergistic Fusion of Simulation and Virtual Reality: A new proposed Approach for Collaborative Integration

Joseph Jabbour^{1,2}, Jalal Possik¹, Charles Yaacoub¹, Adriano O. Solis^{3,*}, Danny Kieken¹, Thierry Sobanski⁴ and Gregory Zacharewicz²

¹LITL, Université Catholique de Lille, Lille, 59000, France

²Laboratoire des Sciences des Risques, IMT Mines Alès, Alès, 30100, France

³School of Administrative Studies, York University, Toronto, M3J 1P3, Canada

⁴Campus Numérique, Université Catholique de Lille, Lille, 59000, France

*Corresponding author. Email address: asolis@yorku.ca

Abstract

This paper presents a comprehensive comparative analysis and a new proposed approach for the integration of Discrete Event Simulation (DES) with Virtual Reality (VR) environments, focusing on addressing challenges encountered in prior architectural approaches. These challenges include redundant environmental design efforts in both DES and VR, the establishment of a unified reference point spanning both architectures, resource allocation concerns, limitations in graphical interfaces, extended development timelines, and heightened network resource consumption. The central aim of this project is to offer effective remedies for overcoming these obstacles, thereby enhancing the overall efficiency of distributed DES/VR environments. It introduces an innovative architecture and approach to mitigate the identified challenges. The proposed method seeks to streamline simulation design, optimize resource allocation, standardize the integration of simulation and VR components, enhance network communication between the VR and DES components, and reduce the time required for DES model development. Experimental results validate the efficacy and efficiency of the proposed solution in effectively addressing the aforementioned challenges.

Keywords: Discrete event simulation, Virtual reality, Distributed simulation, Ownership management, High-Level Architecture (HLA).

1. Introduction

Distributed simulation (DS) has garnered substantial interest across diverse domains such as virtual reality, gaming, and industrial simulations, owing to its potential for enhancing scalability, realism, and efficiency (Strassburger et al., 2008). This study centers on the realm of distributed simulation by leveraging the capabilities of virtual reality (VR) and discrete event simulation (DES). It seeks to tackle the

obstacles that were prevalent in prior architectural designs (e.g., Possik et al., 2021, 2022, 2023), offering innovative solutions for refinement. The expanding significance of DS in numerous applications underscores the relevance of harnessing the capabilities of VR and DES as simulation platforms. While this approach presents promising prospects, it also introduces intricate challenges that need to be effectively navigated.

In previous methodologies (Possik et al., 2021, 2022, 2023), modellers grappled with the duplicative task of



reproducing the physical architecture in both VR and DES, resulting in duplicated effort and potential discrepancies. A unified design process becomes crucial to tackle redundancy and ensure consistent replication. Aligning coordinate systems between VR and DES demands the establishment of a shared reference point and translation vector, serving as a pivotal step for precise synchronization and seamless integration between the two simulation platforms.

Resource consumption poses a significant challenge in distributed simulation, affecting overall performance and scalability. Striking a balance between resource reduction and maintaining fidelity is paramount for optimizing simulation efficiency. The utilization of graphical interfaces within VR and DES introduces the risk of resource-intensive consumption, potentially impacting simulation performance. Strategic optimization of these interfaces is imperative to mitigate resource usage while preserving visual quality, thus ensuring an optimal simulation experience.

Employing techniques to simplify complex simulation models while retaining their essence is essential for enhancing operational efficiency. The prolonged timeline for designing, implementing, and integrating distributed simulations can impede progress. Providing tools to address this is crucial for mitigating time-to-deployment challenges and improving efficiency.

The complexities of distributed simulations, particularly those involving heterogeneous systems, necessitate robust communication mechanisms. The High-Level Architecture (HLA) offers a standardized framework for creating HLA classes that seamlessly facilitate communication across diverse systems. This strategic adoption of the HLA framework enhances communication efficacy and coherence within distributed simulation contexts.

In this paper, we present a comprehensive framework for enhancing the efficiency and effectiveness of integrating DES in VR environments. Our approach involves a series of techniques and methodologies aimed at addressing key challenges and optimizing various aspects of the DS that will be managing both DES and VR.

2. Materials and Methods

To address all aspects that we have mentioned, we propose an upgrade for the work of Possik et al., 2021, 2022), as well as an architecture that combines the introduction of transport paths to simplify simulation models and the applying of Ownership Management techniques within the HLA framework.

DS is a computational methodology that involves running a simulation model across multiple interconnected computing entities. This approach allows for the simulation of complex systems where

components could be geographically dispersed (Topçu et al., 2016). DS aims to capture the interactions and dependencies between different simulation components, facilitating insights into system performance, communication dynamics, and emergent behaviors.

Discrete event simulation is a computer-based modeling methodology renowned for its capability to replicate the intricate dynamic behaviors of complex systems, as well as the intricate interactions among individuals, populations, and their respective environments (Fishman, 2001). At its core, DES involves the replication of events and activities within a system, considering their occurrence in discrete time intervals. Rather than portraying continuous changes, DES focuses on moments when significant events transpire, thereby enabling an in-depth understanding of system dynamics. The primary objective of DES is to facilitate comprehensive comparisons of various potential practices or strategic options within a given system (Goti, 2010). By employing DES, researchers and decision-makers seek to identify the most efficient and effective strategies, ultimately enhancing overall system performance. The realm of healthcare has found valuable utility in employing Discrete Event Simulation. This technique finds practical application in diverse areas within the healthcare domain: patient scheduling and admissions, patient routing and flow schemes, Scheduling and Availability of Resources, Allocation of Resources and much more. In essence, DES provides healthcare professionals with a powerful tool to analyze, predict, and optimize the performance of complex healthcare systems. By simulating various scenarios and testing strategies, DES contributes to informed decision-making, ultimately leading to improved patient outcomes and operational efficiency (Robinson, 2005).

JaamSim is a versatile DES software used to model and analyze complex systems in fields like healthcare and manufacturing (King & Harrison, 2013). It captures event-driven system behavior, allowing users to create custom models, validate designs, optimize processes, and visualize results. This tool is vital for understanding and improving system dynamics in a wide range of applications.

VR is an immersive technology that creates a simulated environment through the use of computer-generated visuals, sounds, and often haptic feedback (Dailey-Hebert et al., 2021). Users engage with this digital environment using specialized hardware, such as VR headsets, to experience a sense of presence and interaction within the simulated world. VR technology aims to replicate real-world experiences or create entirely fictional scenarios, enabling users to perceive and interact with the virtual environment as if it were real. In healthcare, VR has gained attention for its potential to improve medical treatment, training, and therapy (Aziz, 2018). It is used in medical education, offering realistic simulations of medical procedures,

surgeries, and anatomy. This helps healthcare professionals and students practice in a controlled, low-risk environment. VR is also applied in pain management and rehabilitation. By immersing patients in engaging virtual environments, it can distract them from pain during medical procedures (Chou et al., 2020). VR-based rehabilitation programs provide interactive exercises for physical and cognitive recovery, making therapy more engaging. Another healthcare use is exposure therapy for phobias, PTSD, and anxiety (Qian et al., 2020). Patients are gradually exposed to triggers in a controlled virtual environment, helping them confront and manage their fears. VR also enables telemedicine, allowing doctors to connect with patients virtually, regardless of location, enhancing healthcare access.

Unity3D, or Unity, is a widely used cross-platform game engine and development environment (Jerald et al., 2014). It enables creators to build interactive experiences across computers, consoles, mobile devices, and VR headsets (Kuang and Bai, 2018).

HLA serves as a standardized framework for achieving interoperability and interaction among diverse simulation components within a DS environment (IEEE, 2023). HLA was introduced by the United States Department of Defense in the early 1990s as an effort to foster seamless interoperability among different simulation systems. The purpose was to enable simulations developed by various entities to work cohesively together (Zacharewicz et al., 2008). Over time, HLA has undergone multiple iterations to enhance its functionalities and adapt to evolving technological landscapes.

Ownership management is a fundamental concept embedded within the HLA framework for managing shared resources in distributed simulations (Dingel et al., 2002). In the context of DS, ownership management within HLA ensures that specific simulation components have authorized control over certain shared resources. This mechanism prevents conflicting modifications and unauthorized access to objects within the simulation environment. Ownership management mechanisms guarantee the integrity and consistency of the simulation results, which is crucial for maintaining accurate and reliable simulation outcomes in a collaborative distributed setting (Straßburger et al., 2002).

The developed approach of this DS has been tested on the in-center dialysis case study of Possik et al. (2022) at the University Health Network (UHN). The UHN's in-center hemodialysis program, situated within Toronto General Hospital (TGH), represents one of the largest such programs in Toronto, Canada. This facility caters to a total of 308 patients, with 278 of them undergoing intermittent hemodialysis sessions three to six days a week, while the remaining 30 patients are engaged in nocturnal/evening dialysis procedures three days a week.

At TGH, there are two distinct yet interconnected units, referred to as "Hemo-east" and "Hemo-west," which are separated by a corridor. These units collectively house 58 dialysis chairs, accommodating up to 58 patients simultaneously.

Within the facility, regular dialysis stations are partitioned by curtains, while isolated dialysis stations are equipped with divider partitions. Notably, the regular dialysis stations are strategically positioned along the perimetral and central areas of the wards.

In our endeavor to depict the dynamics encompassing patient flow, staffing, and resource allocation within the dialysis unit, we constructed a representation encompassing various operational facets. These encompass activities ranging from admission and weighing to patient settling, dialysis, decannulation, and cleaning procedures. Our model will also incorporate the intricate interactions between patients and the healthcare professionals overseeing their care, including nurses, nephrologists, and nurse practitioners.

The agents integral to our model include nephrologists who serve as most responsible physicians (MRPs), nephrology residents/nurse practitioners (NPs), nurses, hemodialysis assistants, housekeepers, clerks, and patients. These agents are systematically categorized by their roles, and their respective quantities are quantified. Each agent is uniquely identified, facilitating a comprehensive indexing system, as delineated in Table 1.

There are seven distinct agent types, ranging from 1 to 7. We enumerate the precise count of agents for each agent type within the TGH hemodialysis ward, as indicated in the "Total Number" column. Subsequently, we index each agent type from 1 to the total number of agents for accurate representation.

Table 1. Indexing System of Agents

Agents	Type of Agents	Total Number	Index
MRP	1	10	1 to 10
Nephrology Resident/NP	2	12	1 to 12
Nurse	3	110	1 to 110
Assistant	4	30	1 to 30
Housekeeper	5	10	1 to 10
Clerk	6	6	1 to 6
Patient	7	308	1 to 308

We also incorporate the identification and classification of locations or destinations within our framework, which are systematically indexed in accordance with the scheme shown in Table 2.

We have categorized the hemodialysis unit at TGH into three primary sections: (i) common areas, (ii) the east ward, and (iii) the west ward, encompassing a total of 76 distinct areas or destinations. Each destination is uniquely identified using an index ranging from 1 to 76.

Table 2. Indexing System of Destinations

Total Number	Destinations	Index
	Stay in place	0
Common areas		
1	Lobby	1
1	Reception	2
East Ward		
29	Dialysis Units	3 to 31
2	Isolated Units	32 to 33
1	MRP Station	34
6	Staff Station	35 to 40
1	Female Locker Room	41
1	Break Room	42
1	Housekeeper Station	43
West Ward		
24	Regular Units	44 to 67
3	Isolated Units	68 to 70
3	Staff Station	71 to 73
1	Male Locker Room	74
1	Housekeeper Station	75
Out		
1	Outside the hospital	76

The personnel within the hemodialysis unit collaborate harmoniously to facilitate the efficient functioning of the unit and prioritize the welfare of the patients. Each member fulfills distinct responsibilities, thereby making unique contributions to the various dimensions of patient care and support. Presented below are the delineated roles and actions of both the staff and patients within the hemodialysis unit, systematically indexed for reference (Table 3)

Table 3. Indexing System of Actions

Index	Actions
0	No action
Hemodialysis Assistants (HAs)	
1	Set up dialysis machines, including dialyzers and tubes
2	Sit at nursing stations when not actively preparing machines
Nurses	
3	Connect machines to patients
4	Rotate between patients during dialysis to assess blood pressure, blood flows, and circuit pressures
5	Disconnect the machines from the patients
6	Handover patient care to the next shift's nurses
7	Clean patient charts and blood pressure cuffs
8	Sit at the nursing station in proximity to their assigned patients

Other Clerical and Support Staff

9 Provide various support services

Nephrologists

10 Oversee patient care during shifts

11 Conduct rounds and provide medical expertise

Nurse Practitioners (NPs)

12 Assist nephrologists during shifts

13 May stay in an office area or leave the unit when not actively seeing patients

Nephrology Residents

14 Make rounds in the unit on behalf of the most responsible physician (MRP)

Housekeeping Staff

15 Cleanse chairs/beds, replace linens, and clean the area after patients leave

Patients

16 Lineup

17 Get screened at the reception

18 Wait for a message from an assistant declaring that the dialysis station is ready

19 Reach their assigned station

20 Get the treatment

21 Once the treatment is over, the patients return to the lobby and wait for a short time

22 Leave the environment

Patients arrive at the lobby, form a queue, undergo initial screening at the reception, and await notification from an assistant confirming the readiness of the dialysis station. Upon reaching their designated stations, patients remain in place until the treatment concludes. Following the completion of treatment, patients return to the lobby, briefly wait, and subsequently exit the facility.

In our approach, we have established a structured system for representing actions or transitions between destinations. This system employs arrays composed of four elements, which are organized as follows:

[*Agent Type, Agent Index, Next Destination, Action Type*]

To illustrate, consider the following example: [1, 1, 2, 17]. This signifies:

- Agent Type = 1 (indicating a patient)
- Agent Index = 1 (referring to the patient's unique identifier)
- Next Destination = 2 (signifying the patient's getting screened at the reception)
- Action Type = 17 (denoting the specific action of the patient getting screened at the reception).

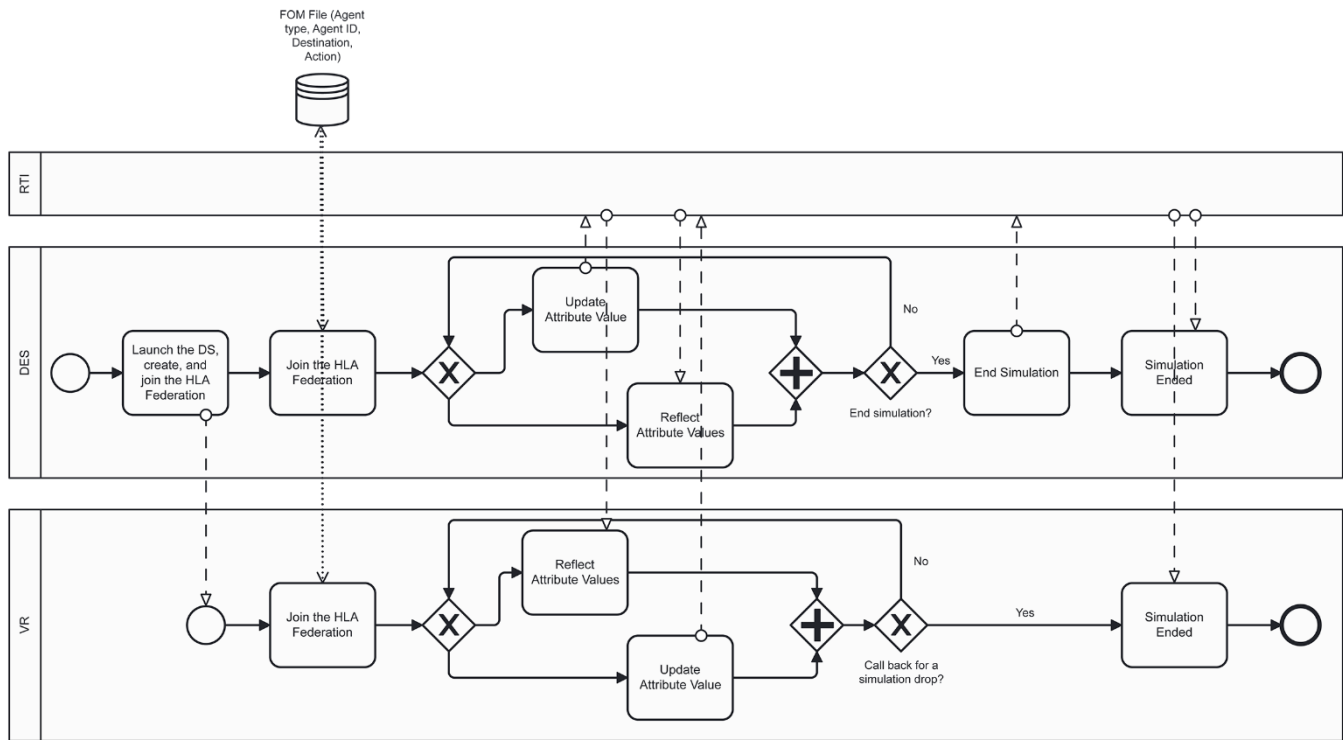


Figure 1. Model Architecture

This indexing system allows for easy retrieval of array elements based on their respective index numbers. The typical sequence of actions for a patient is shown in Table 4.

Table 4. Sequence of Actions for a Patient

Actions	Sequence
Get screened at the reception	[1, 1, 2, 17]
Wait for a message from an assistant declaring that the dialysis station is ready	[1, 1, 0, 18]
Reach their assigned station	[1, 1, 3, 19]
Get the treatment	[1, 1, 0, 20]
Once the treatment is over, the patients return to the lobby and wait for a short time	[1, 1, 1, 21]
Leave the environment	[1, 1, 76, 22]

The integration of JaamSim and Unity, two robust yet distinct technologies, has been achieved by leveraging the HLA standard. On one hand, JaamSim operates as a Java-based engine, where models are intricately connected to Java code. Conversely, Unity primarily relies on a C++ foundation, with code implementations predominantly in C#, JavaScript/UnityScript, and, on occasion, Boo.

To facilitate communication between two distinct components, or federates, characterized by their differences, an HLA interface has been meticulously configured. This interface serves as an intermediary between the RTI and the HLA federate, bridging the gap and enabling seamless interaction and data exchange even with their inherent heterogeneity.

The global system architecture of the DS system is depicted in Figure 1, providing a comprehensive overview. An HLA federation consists of multiple federates, a Runtime Infrastructure (RTI), and a Federated Object Model (FOM). The RTI offers a standardized suite of services for data exchange, synchronization, and collaborative management, whereas the FOM delineates the objects, classes, interactions, and parameters employed for communication.

In this architectural framework, the DES component used in the JaamSim platform seamlessly interacts with the VR environment constructed using Unity 3D.

JaamSim assumes a pivotal role in this integrated system, it publishes four critical elements for each agent within the simulation: "Agent Type," "Agent Index," "Next Destination," and "Action Type." Simultaneously, Unity, as part of the system, subscribes to these variables which allows real-time tracing of agents, enabling precise monitoring of their movements and interactions within the simulated healthcare unit then publishes the achievement of these actions, awaiting instructions for new actions. Leveraging the publish/subscribe mechanism of the HLA standard, JaamSim transmits the necessary information to the RTI, which ensures that all subscribed components receive specific data at synchronized intervals. Unity follows a similar process, publishing pertinent information to the RTI for dissemination.

In HLA, ownership management plays a pivotal role in the seamless operation of distributed simulations. It pertains to the regulation and allocation of ownership rights over specific attributes of individual object instances among federates, which are the distinct simulation participants. *Ownership Rights* cover a federate's authority to govern specific facets of a simulation object, such as altering its state or attributes. These rights prescribe which federate holds the privilege to revise or handle designated objects within the simulation. *Ownership Transfer* mechanisms are integrated into HLA to enable the dynamic transfer of object ownership among federates. Knowing that at any time, only one federate is allowed to update a given attribute of a given object instance.

In this context, an *attribute* (e.g., destination) refers to a property or characteristic of an *object* (e.g., nurse) in the simulation, while an *object instance* represents a unique occurrence of that object (e.g., nurse number 5).

The nurse actions need to be coordinated between two federates, JaamSim and Unity 3D, for managing the "action" attribute of a nurse object instance:

- **Initial Ownership Assignment:** At the start of the distributed simulation (DS), the "action" attribute's ownership of a nurse object instance is assigned to the JaamSim federate. This means JaamSim exclusively possesses the right to update and control the nurse's actions.
- **Publishing and Ownership Release:** JaamSim updates the "action" attribute to reflect the nurse's actions and publishes the changes. After publication, JaamSim releases its ownership of the attribute.
- **Ownership Transfer:** Subsequently, the Unity 3D federate subscribes to the "action" attribute, aiming to synchronize and visualize the nurse's actions in the 3D environment. Unity 3D acquires ownership of the attribute afterwards. This ensures that Unity 3D can control the attribute during its turn.
- **Updates and Publishing:** Unity 3D updates the "action" attribute as required for visualization purposes and publishes these updates. Once the updates are published, Unity 3D releases its ownership.

This cycle continues as needed within the distributed simulation, with federates taking turns to own, update, publish, and release the "action" attribute. Ownership management in this context ensures that only one federate has control over the attribute at any given time, preventing conflicts and maintaining data consistency. It enables efficient coordination and synchronization of nurse actions between JaamSim and Unity 3D, facilitating a realistic and interactive DS experience.

The essence of ownership management lies in ensuring that only one federate is permitted to update

a given attribute of a particular object instance in the course of the simulation. This exclusivity minimizes conflicts, maintains data consistency, and enables effective synchronization, preventing multiple federates from attempting simultaneous modifications that could lead to inconsistent or erroneous simulation outcomes.

3. Results and Discussion

In this research, we have effectively addressed a spectrum of challenges associated with the integration of DES into VR environments. These challenges encompassed the elimination of duplicative tasks and discrepancies in replicating physical architectures between VR and DES.

Also, significant reductions in the volume of exchanged messages within the simulation environment have been achieved through our proposed improvements. To illustrate, in the case of a patient spending an average of four hours in the hemodialysis unit per treatment, the conventional approach required the transmission of a staggering $4 \times 60 \times 60 \times 24 = 345,600$ position (x,y) messages across the DS infrastructure to ensure seamless, uninterrupted movement within the VR environment. In contrast, with the implementation of our novel architecture, each patient now generates a mere 16 sequence updates, facilitating communication between JaamSim and Unity 3D. This transformative architectural shift has resulted in a 99.9954% reduction in message volume for a single patient. Notably, this substantial decrease in message traffic is also consistently observed across all staff members, demonstrating the remarkable efficiency gains achieved through our approach.

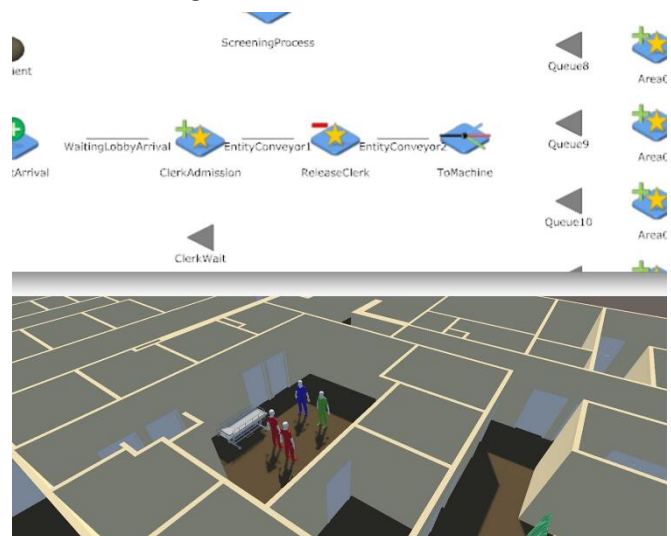


Figure 2. JaamSim and Unity 3D illustration

4. Conclusions

In this paper, we have introduced an approach aimed at optimizing the integration of DES within VR environments as illustrated in Figure 2. Our proposed framework is designed to enhance effectiveness and efficiency by adhering to standardized practices. Through the allocation of ownership rights, the HLA serves as a DS standard that streamlines data exchange, fosters scalability, and ensures orderly execution within extensive, distributed simulations.

As a future direction, we contemplate the addition of a *human-in-the-loop* layer, a pivotal augmentation that involves human interaction and empowers users to influence the outcomes of events or processes. This expansion holds promise, particularly in the realm of training and immersive experiences, further enriching the capabilities and applicability of our framework.

Acknowledgments

We would like to express our gratitude to the ANR (Agence Nationale de la Recherche) for funding this research under the DemoES project. Additionally, we extend our appreciation to the Faculty of Management, Economics, and Science (FGES) at Lille Catholic University for their financial support.

References

- Aziz, H. A. (2018). Virtual reality programs applications in healthcare. *Journal of Health & Medical Informatics*, 9(1;1000305):1-3. <https://doi.org/10.4172/2157-7420.1000305>
- Chou, H.-Y., Chen, S.-C., Yen, T.-H., and Han, H.-M. (2020). Effect of a virtual reality-based exercise program on fatigue in hospitalized Taiwanese end-stage renal disease patients undergoing hemodialysis. *Clinical Nursing Research*, 29(6):368-374. <https://doi.org/10.1177/1054773818788511>
- Dailey-Hebert, A., Simmons Estes, J., and Choi, D. H. (2021). This history and evolution of virtual reality. In: Choi, D. H., Dailey-Hebert, A., and Simmons Estes, J. (eds.), *Current and Prospective Applications of Virtual Reality in Higher Education*. IGI Global, 1-20.
- Dingel, J., Garlan, D., and Damon, C. (2002). Bridging the HLA: Problems and solutions. *Proceedings of the Sixth IEEE International Workshop on Distributed Simulation and Real-Time Applications*, Fort Worth, TX, USA, October 2002, 33-42. <https://doi.org/10.1109/DISRTA.2002.1166886>
- Fishman, G. S. (2001). *Discrete-Event Simulation: Modeling, Programming, and Analysis*. Springer. <https://doi.org/10.1007/978-1-4757-3552-9>
- Goti, A. (2010). *Discrete Event Simulations*. IntechOpen.
- IEEE (2023). *Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)—Framework and Rules* (n.d.). Retrieved September 10, 2023, from <https://ieeexplore.ieee.org/document/STD96061>
- Jerald, J., Giokaris, P., Woodall, D., Hartholt, A., Chandak, A., and Kuntz, S. (2014). Developing virtual reality applications with Unity. *2014 IEEE Virtual Reality (VR)*, Minneapolis, MN, USA, March/April 2014, 1-3. <https://doi.org/10.1109/VR.2014.6802117>
- King, D. H. and Harrison, H. S. (2013). Open-source simulation software “JaamSim.” *2013 Winter Simulations Conference (WSC)*, Washington, DC, USA, December 2013, 2163-2171. <https://doi.org/10.1109/WSC.2013.6721593>
- Kuang, Y. and Bai, X. (2018). The research of virtual reality scene modeling based on Unity 3D. *2018 13th International Conference on Computer Science & Education (ICCSE)*, Colombo, Sri Lanka, August 2018, 1-3. <https://doi.org/10.1109/ICCSE.2018.8468687>
- Possik, J., Asgary, A., Solis, A. O., Zacharewicz, G., Shafiee, M. A., Najafabadi, M. M., Nadri, N., Guimaraes, A., Iranfar, H., Ma, P., Lee, C. M., Tofighi, M., Aarabi, M., Gorecki, S., and Wu, J. (2023). An agent-based modeling and virtual reality application using distributed simulation: Case of a COVID-19 intensive care unit. *IEEE Transactions on Engineering Management*, 70(8):2931-2943. <https://doi.org/10.1109/TEM.2022.3195813>
- Possik, J., Azar, D., Solis, A. O., Asgary, A., Zacharewicz, G., Karami, A., Tofighi, M., Najafabadi, M., Shafiee, M. A., Merchant, A. A., Aarabi, M., and Wu, J. (2022). A distributed digital twin implementation of a hemodialysis unit aimed at helping prevent the spread of the Omicron COVID-19 variant. *2022 IEEE/ACM 26th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, Alès, France, September 2022, 168-174. <https://doi.org/10.1109/DS-RT55542.2022.9932047>
- Possik, J., Gorecki, S., Asgary, A., Solis, A. O., Zacharewicz, G., Tofighi, M., Shafiee, M. A., Merchant, A. A., Aarabi, M., Guimaraes, A., and Nadri, N. (2021). A distributed simulation approach to integrate AnyLogic and Unity for virtual reality applications: Case of COVID-19 modelling and training in a dialysis unit. *2021 IEEE/ACM 25th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, virtual conference, September 2021, 1-7. <https://doi.org/10.1109/DS-RT52167.2021.9576149>
- Qian, J., McDonough, D. J., and Gao, Z. (2020). The Effectiveness of virtual reality exercise on individual’s physiological, psychological and rehabilitative outcomes: A systematic review. *International Journal of Environmental Research and Public Health*, 17(11;4133):1-17. <https://doi.org/10.3390/ijerph17114133>
- Robinson, S. (2005). Discrete-event simulation: From the pioneers to the present, what next? *Journal of the Operational Research Society*, 56(6):619-629.

<https://doi.org/10.1057/palgrave.jors.2601864>

- Straßburger, S., Hamm, A., Schmidgall, G., and Haasis, S. (2002). Using HLA ownership management in distributed material flow simulations. *Proceedings of the 2002 European Simulation Interoperability Workshop*, London, England, June 2002, 1-7.
- Strassburger, S., Schulze, T., and Fujimoto, R. (2008). Future trends in distributed simulation and distributed virtual environments: Results of a peer study. *2008 Winter Simulation Conference*, Miami, FL, USA, 777-785. <https://doi.org/10.1109/WSC.2008.4736140>
- Topçu, O., Durak, U., Oğuztüzün, H., and Yilmaz, L. (2016). *Distributed Simulation: A Model Driven Engineering Approach*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-03050-0>
- Zacharewicz, G., Frydman, C., and Giambiasi, N. (2008). G-DEVS/HLA environment for distributed simulations of workflows. *Simulation*, 84(5):197-213. <https://doi.org/10.1177/0037549708092833>