Virtual Reality-based Blockchain Application for Optimized Collaborative Decisions of Modular Construction

Mohamed Assaf [1], Rafik Lemouchi[1], Mohamed Al-Hussein[1], Xinming Li[2*]

[1] Department of Civil & Environmental Engineering, University of Alberta, Edmonton, AB, Canada;
[2] Department of Mechanical Engineering, University of Alberta, Edmonton, Alberta, Canada.
massaf2@ualberta.ca, lemouchi@ualberta.ca, malhussein@ualberta.ca, xinming.li@ualberta.ca*

Abstract

Game engine technology has been studied extensively lately in offsite construction (OSC) research due to its ability to develop virtual reality (VR) environments that can be used in the decision-making process prior to the actual project implementation. However, accessing these VR scenes typically requires the participant to be in a VR lab or at least possess VR-specialized hardware and software. Also, these models are typically accessed in isolation with no real-time connectivity with other stakeholders, limiting their collaboration efficiency and questioning their applicability in real life. Thus, this study proposes a web-based multi-player framework based on game engines and blockchain technologies to promote collaborative decision-making processes in OSC projects. The developed system allows users to access a cloud simulation-optimization (SO) model to evaluate several decisions based on identified key indicators. The system is validated using an offsite construction (OSC) case study. Two scenarios are defined to ensure efficient connectivity among the OSC stakeholders and the security of the developed network.

Keywords –
Virtual Reality, Blockchain, Operation Management, Modular Construction

1 Introduction

Through the advancement of construction technologies, an expanding range of sustainable and productive construction methods are proposed, including OSC. Modular construction (MC), as a part of OSC methods, has the potential to improve the construction industry's current state, which suffers from huge delays, high injury rates, and cost overruns [6]. MC shifts almost all of the tasks from onsite to offsite factories [4]. This shift provides substantial value to the quality, sustainability, and safety in the construction industry. However, shifting activities from onsite to offsite factories constitutes a complex supply chain that includes manufacturing, logistics, and construction stages. The fragmented nature of this supply chain leads to a greater demand for collaboration among the MC stakeholders, including manufacturers, logistics companies, and onsite contractors [7]. It also calls for mutual trust and efficient information sharing among the MC stakeholders [2].

Therefore, researchers aimed to use many digital technologies to support a collaborative environment for MC stakeholders [7]. These technologies include building information technology (BIM), game engines, and virtual reality (VR). VR and game engines promote the use of static BIM 3D models by adding interactive scenes. For instance, Zhang, et al. [16] developed a virtual system to support collaborative planning for construction activities of the MC projects. Their study used game engines and BIM models to create virtual scenes. Ezzeddine and García de Soto [4] aimed to integrate different teams in MC projects using game engines and BIM models. Their model targeted design, manufacturing, and logistics stages in MC projects. Wu, et al. [13] developed an information system to share secure data through the construction stage of MC. Their study integrated the blockchain and BIM technologies with the aim of mutual trust in sharing installation data. Besides, IoT applications are also found in the literature and aim to collect near real-time data in order to enhance visualization and traceability. For instance, Wu, et al. [12] developed an information system that facilitated manufacturing management by providing real-time and traceable information flow. In their study, BIM models were integrated with IoT sensors and blockchain technology to prevent the single point of failure in the IoT networks and synchronize BIM changes across different platforms. Hussein, et al. [6] developed a decision support system to optimize the logistics and construction stages of the MC projects. Simulation techniques, namely discrete event simulation (DES) and agent-based modeling (ABM), were adopted in their study to evaluate key performance indicators through MC implementation.

Despite the contributions of these previous studies, several limitations are identified. First, the applications of game engines in developing virtual scenes were only limited to a single player (user) at a time with no real-time connectivity, questioning the application of these models in real-life practices. Second, when dealing with multi-player functions, the literature lacks a decentralized network (e.g., blockchain networks) that ensures that the virtual assets are secured and participant data are transparent. This aligns with the findings of Bao, et al. [2], who aimed to develop a multi-player safety training model. Their study recommended a decentralized VR model after pointing out that centralized models as a limitation of their study. Third, even with multi-player virtual scenes, the accessibility of the models is limited since the participant needs to be in a VR facility or lab. Hence, remote access to the developed VR models is needed.

In light of these limitations, this study proposes a multi-player collaborative decision-making framework to enhance coordination and connectivity among MC participants. The proposed system combines VR, simulation, optimization, and blockchain (VR-SO-BC) to overcome the mentioned limitations. The objectives of this study are: 1) build a cloud simulation-optimization (SO) model that provides MC participants with appropriate decisions; 2) develop virtual scenes that allow participants to try various selected decisions in VR environments; 3) develop a web-based multi-player model to enhance decision-making processes; 4) validate the proposed model based on a case study.

2 Methodology

Figure 1 illustrates the architecture of the VR-SO-BC-OSC framework. This framework aims for the following: 1) provide critical decisions for the construction, logistics, and planning phases of MC projects; 2) discuss the feasibility of the chosen decisions in virtual environments; 3) provide a fully immersed VR experience for the MC participants to interact with the developed virtual models; 4) develop multi-player virtual scenes governed by blockchain network for collaborative decision making.

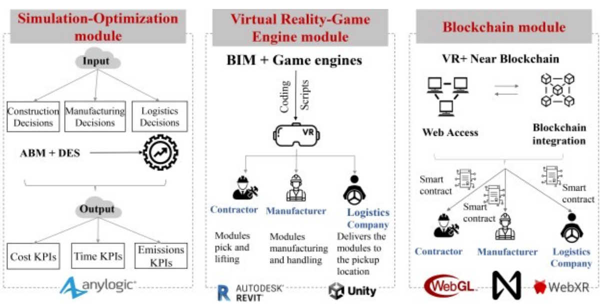

Figure 1: The VR-SO-BC-OSC architecture

The proposed framework integrates simulation and optimization techniques to estimate analytical values for the critical decisions of the OSC implementation. Even though process simulation can be integrated within the virtual game scenes, it can only represent the flow of the process with no complex interactions and dynamic behavior of the agents [9]. Therefore, the current study's authors chose to develop a hybrid simulation model for this stage that would allow us to consider dynamic behaviors and complex interactions. However, a cloud version of this model is integrated in the following steps with a game engine to assess the physical properties and feasibility of the selected decisions. The proposed hybrid simulation model combines discrete event simulation (DES) and agent-based modeling (ABM) methods to represent the overall MC supply chain activities, considering the dynamic behaviors of the included agents. Furthermore, this module also presents several key performance indicators (KPIs), such as cost, time, and produced CO_2, to measure the performance of the MC supply chain. The hybrid simulation model is then integrated with an optimization model that aims to find near-optimal decisions in each stage of the MC project to enhance the performance of the MC supply chain. The results of the proposed SO model are fed into the virtual reality/game engine module to evaluate the selected decisions in immersive virtual environments.

The virtual reality/game engine module is proposed as a continuation of the developed SO module. Its purpose is to improve decision-making by considering all of the physical properties and feasibility of the chosen decision from the previous module. The developed module integrates the merits of BIM and game engine technologies. Building a 3D BIM model forms the first step in this module, as it determines the design of all of the prefabricated modules and substructure elements. It is worth mentioning that the structural and architectural design of the MC project is beyond the scope of the current study. Game engines are then employed to create virtual environments of the developed BIM model. Virtual environments enable MC teams to visualize each process, allowing them to make proactive and informed decisions [4]. In this module, three main MC teams are integrated: manufacturing, onsite assembly, and transportation teams. Each team has its own virtual environment, and they all share a separate virtual scene. The creation of the virtual environment will be detailed in the following sections. Furthermore, the virtual scenes are developed to enable a fully immersive VR experience. Immersive VR experiences can enable participants to interact highly with virtual environments, facilitating a more efficient decision-making process [15].

The last module in the proposed methodology is the web-based VR/ blockchain module. Web-based VR allows participants to access the developed VR module via remote platforms that do not require them to be in a

VR facility or laboratory [10]. The proposed module allows participants to access the developed virtual environments in screen-based VR and immersive VR (using headsets). Further, a server connection is established between the web screens and the main designed server in the game engine. Despite the capabilities of the web-based VR models in promoting connectivity and facilitating better-informed decisions, they lack data transparency and security of the participants. This issue aligns with the results of Bao, et al. [2], who recommended integrating the web-based VR application with BC technology to ensure the persistency of the virtual scenarios, transparency of the user data, and activity tracking on the web-based VR model. Hence, a BC network is added to the developed Web-based VR models to address these limitations. The developed blockchain comprises several blocks. In which each block includes its own records, timestamps, and hash values of the previous block [11]. In addition, the BC network is supported by a smart contract to facilitate adding participants of any organization to the BC network. Each participant in the web-based network owns a non-fungible token (NFT) that incorporates a unique digital signature, defining the ownership of the virtual character in the virtual environment.

3 VR-SO-BC-OSC Model Development

This section presents the development of the modules mentioned above. Figure 2 shows the sequential steps included in each module. The proposed model is tested on a case study of a modular bridge project. The bridge consists of forty prefabricated steel modules and has a main span of 300 meters and two side spans of 120 meters each. More details on the statistical system of this case study were presented by Yu and Chen [14]. It is worth mentioning that this case study serves as a proof of concept to show the capabilities of the proposed model.

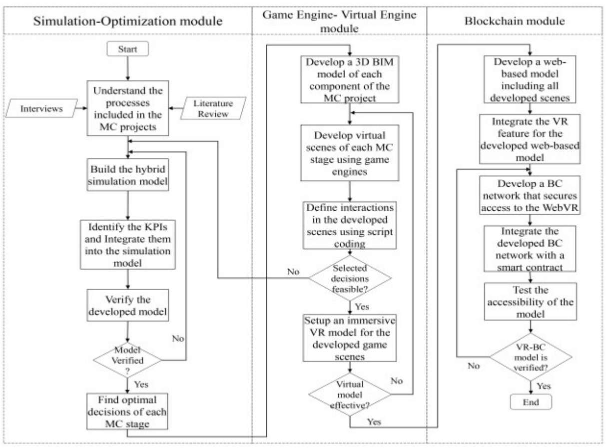

Figure 2: Steps included in the model development

3.1 Simulation-Optimization module

This subsection discusses the development of the SO model. It is worth mentioning that the SO model is an extension of the work done by Assaf, et al. [17], where the optimization of each project KPI was the main focus of that work. However, this study mainly focuses on developing a multi-user decision-making environment for collaborative decisions and participant integration.

As mentioned, a hybrid simulation model of agent-based modeling (APM) and discrete event simulation (DES) techniques is developed to consider all MC activities and possible interactions between the resources. Anylogic tool is used to develop the SO model. Figure 3 shows a sample of SO model development.

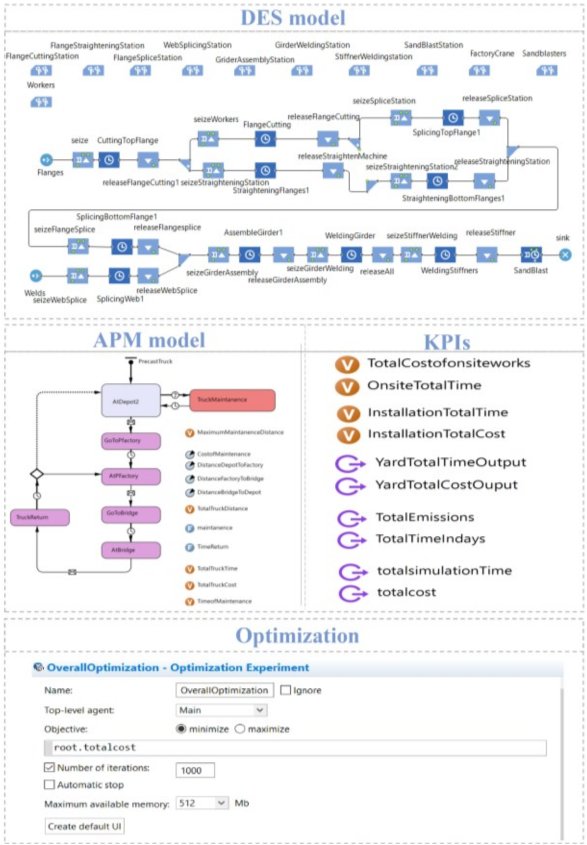

Figure 3: Overview of the developed SO model

The DES model simulates activities related to manufacturing, logistics, and onsite works. The APM supports the interactions between agents (resources) and the DES processes and between one another. Four main APM agents are considered: precast trucks, steel girders trucks, barge, and traffic agents. These agents send and receive messages to and from the DES model and among one another to update specific parameters and trigger certain behaviors. For instance, when a particular stage is reached in the DES model, a "GoToPfactory" message is sent to the precast agent to travel to the factory based on the distance specified in the model. Similarly, the traffic agent sends messages to the other agents to influence their traveling time based on the traffic conditions.

Several KPIs are calculated within the APM-DES model, including cost, time, and emissions in each MC stage. The change in these KPIs is based on several parameters in all MC stages. For instance, some of the included parameters in the onsite installation stage are the number of mobile cranes and the number of workers. These parameters influence the hybrid simulation model and, subsequently, the resulting KPIs. A sensitivity analysis is carried out to discover which of these parameters are sensitive and have a high impact on the KPIs. An optimization model is integrated into the simulation model to efficiently discover the appropriate values for these parameters. In this regard, the OptQuest optimization tool (supported by Anylogic) is used to run the optimization model. The decision variables of the model are considered based on the sensitivity analysis and include the following: No of steel trucks, No of precast trucks, Inventory capacity, Just-in-Time (JiT) application, No of production lines, No of factory workers, No of onsite mobile cranes. Constraints are defined in the optimization model based on user specifications. For instance, the user can indicate a certain budget that the cost should not exceed. This means that if this budget is exceeded, the generated solution will be infeasible.

3.2 Virtual Reality-Game engine module

This subsection discusses the development of the game scenes that will be used to connect various MC stakeholders together for collaborative decision-making. Further, a cloud version of the developed hybrid SO model by Anylogic is accessible in these scenes to allow users to examine several KPIs based on their input. Two main software tools are used in developing the virtual scenes: Autodesk Revit and Unity 3D. Revit tool (Version 2023) is used to create the 3D BIM models of all of the elements in the MC project. In the BIM models, the needed information (such as module ID and location) is included to be later used in the virtual scenes. The developed BIM models are then exported in FBX format, which is compatible with the Unity3D tool. It is worth mentioning that the developed virtual scenes extend virtual scenes presented by the study authors in Assaf et al. [1]. The Unity3D tool (version 2022.3.11.f1) is used to develop four main scenes, as shown in Figure 4. The first scene, shown in Figure 4(a), is an exploring scene that is accessible to all participants and allows them to do the following tasks: 1) access the Anylogic cloud model to test various decisions, 2) explore the chosen decisions in an interactive virtual environment, 3) review modules details and installation instruction, 4) connect with other stakeholders in the virtual environment for collaborative decision-making process. The second scene is the onsite scene, as shown in Figure 4(b).

The scene offers the onsite team the following functions: 1) real-time assessment of various KPIs, such as time and CO_2 emissions, 2) real-time assessment of the clearance between the prefabricated component and the surroundings, and 3) connect the logistics team with the onsite team.

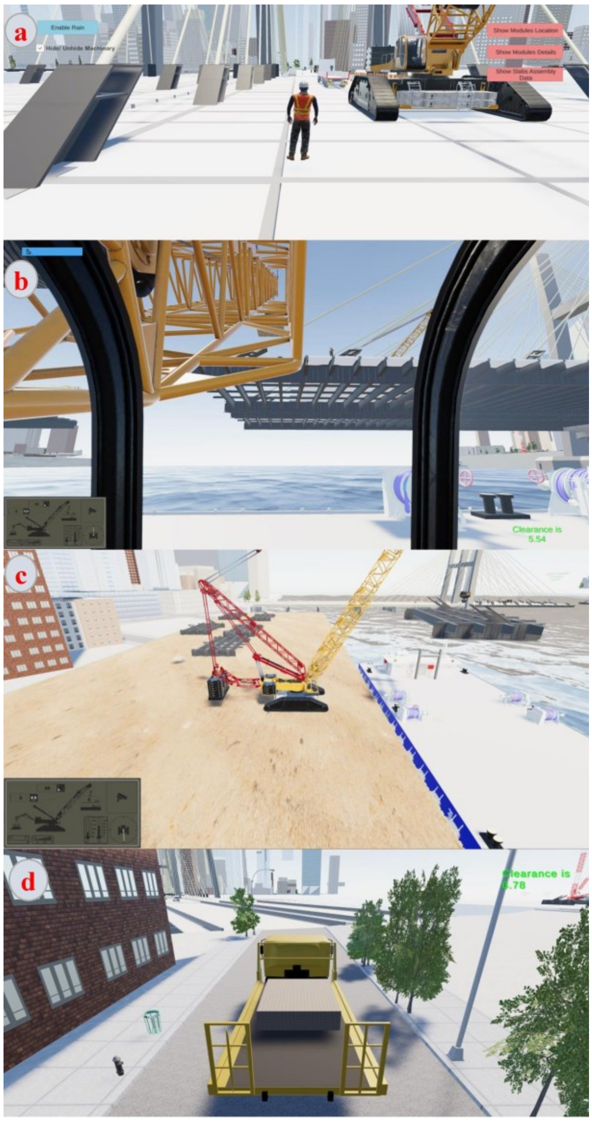

Figure 4: Developed virtual scenes in Unity Editor

The third developed scene is the steel manufacturing scene, which is displayed in Figure 4(c). The manufacturing team can perform the following tasks: 1) check the time needed for loading and producing CO2 emissions in real-time, and 2) evaluate the selected number of modules' storage levels. The last scene is the logistics scene, as shown in Figure 4(d). The transportation team is able to explore the following features: 1) navigate through city roads leading to the project, 2) assess in real time the clearance of the modules for each chosen road, 3) test the accessibility to the pickup locations, and 4) evaluate the selected safety precautions when maneuvering to the pickup locations. It is worth mentioning that the user is able to navigate the Anylogic cloud model form within Unity Editor to try different scenarios. They can then test the selected decisions in the developed virtual interactive scenes. It is also worth mentioning that users can access virtual scenes in a fully immersive mode. Steam VR plugin is added to Unity3D, and the immersive experience is set using the following hardware tools: Varjo XR headset, HTC Vive controllers, and HTC Vive base stations.

3.3 Multi-Player blockchain module

This subsection discusses the implementation of the web-based virtual model that offers multi-player functions and is supported by blockchain integration. As mentioned before, this module aims to provide a collaborative decision-making environment to test the previously agreed decisions. A multi-player function is added to the Unity editor after developing the virtual scenes using BIM and game engines. Fishnet is a network solution by Unity3D that provides many multi-player functions and supports web-based integration [5]. Fishnet invokes network behavior in the virtual scenes, allowing clients and the server to join the same virtual scenes with real-time connections. Therefore, C# scripts that control all of the desired networking interactions are updated to support multi-player features. Thus, the networking behavior is integrated into Unity Editor using Fishnet functions. The network manager is a vital component in establishing the connection between the server and clients [5]. A player, e.g., a truck, is spawned in the network manager component and is synchronized among the server and clients. Spawning points can be defined to specify the location of networked objects.

To provide authority on who can manipulate the players, a function is added to each network object to only allow clients to control their own players. A canvas is also predefined in the network to support establishing the server and adding clients. Furthermore, the multi-player model also supports broadcasting functions to provide communications among objects without the need for network components. For instance, to connect the logistics team and onsite team, the truck (network object) is added to the network manager, while the crane is configured using broadcast functions. This way, the movement of both crane and truck is synchronized among the server and clients. Furthermore, real-time connection and bi-directional data are configured on the multi-player model. For instance, the onsite team and logistics team can view real-time evaluation of different KPIs, such as clearance checks and produced emissions. This connectivity and information sharing are tested through chat functions between the server and clients.

Despite the capabilities of the multi-player model to provide collaborative decision-making among project teams, it is centralized in a way that the participants need to be in a VR facility to try the developed scenes. Thus, web functions and decentralized networks are discussed in this subsection to overcome this issue. WebGL is applied in this part to deploy the web-based VR model. WebGL is a JavaScript API that deploys 3D and 2D virtual scenes within any compatible web browser [3]. Unity 3D supports WebGL and can directly build WebGL extensions. However, mobile browsers are not supported by Unity WebGL, which is a limitation since most people prefer touch-based interactions [8]. Unity 3D supports WebXR, allowing users to access the web-based model in immersive environments using headsets.

As discussed in the methodology section, the blockchain network is integrated with the developed model to provide the following functions: access control for the stakeholders, decentralized networks that prevent a single point of failure, and record all of the activities on the web-based model. Among many blockchain platforms, this study focuses on NEAR protocol integration for the following reasons: 1) NEAR is a layer one blockchain network without dependency on other chains [11]; 2) NEAR protocol is developed to have a low transaction cost, making it suitable for networks that have rapid transactions; 3) it is compatible with Unity 3D and allows the users to develop their own smart contract to support the interaction with the blockchain network. Therefore, the NEAR protocol is selected and added to Unity 3D virtual scenes to be later included in the WebGL building process.

A sample of how the smart contract supports the developed blockchain network is shown in Figure 5. One of the prerequisites to establishing the NEAR blockchain is the NEAR wallet. The wallet contains a set of Non-Fungible Tokens (NFTs) that are controlled by a smart contract. The tokens, in this case, are considered the participants in the network. Figure 5(a) shows adding a participant to the blockchain network using the smart contract integration. The information needed when adding a new token (participant) includes the following: token ID, title, description, and image. Figure 5(b) shows how the smart contract secures any activities on the blockchain network with hash value and timestamp to be immutable and permanently saved. As discussed, the NEAR blockchain is integrated with the Unity 3D platform, meaning a smart contract must be assigned to the editor. Figure 5(c) shows how the smart contract is included in the C# scripts to be later on the part of the WebGL build. Figure 5(d) shows the newly added token to the NEAR wallet.

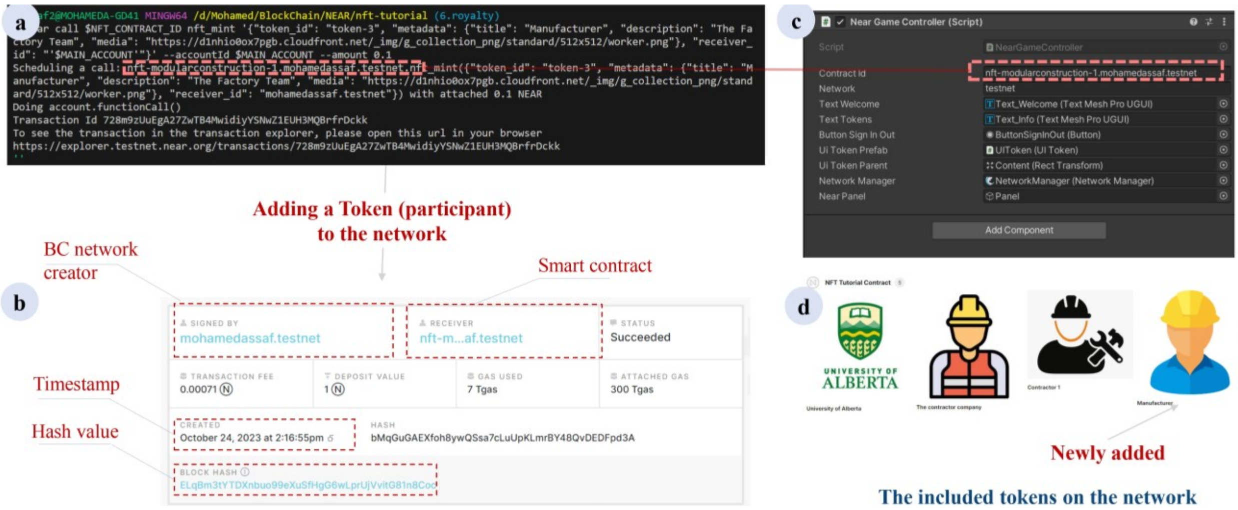

Figure 5: Smart contract integration in the blockchain network and within Unity

Another notable feature of integrating NEAR protocol in Unity Editor is assigning a name header for each participant in the multi-player environment. The information of the tokens (participants) is retrieved from the blockchain network and assigned in the virtual scenes. Table 1 shows a sample of the algorithm used in doing so. As shown in the table, a remote procedure call (RPC) is used to send information across the network. This information is synchronized for both clients and server screens. A NEAR canvas in Unity editor is also added to facilitate the connection to the network and selection of the participants. When a certain token is clicked, the mentioned algorithm is triggered, and the participant information is assigned.

Table 1: synchronizing participant's information

Sample Algorithm: Retrieving Participant information to be added to the virtual scenes

```
public class Player: NetworkBehaviour
{
   [SerializeField] private TMPro.TextMeshProUGUI
_ParticipantUsername;
   public static string localPaticipantName;
   [SyncVar(OnChange =
nameof(OnParticipantNameChanged))]
   public string username;
   private void OnParticipantNameChanged (string
oldValue, string newValue, bool isServer)
   {
      _ParticipantUsername.text = newValue; }
   public override void OnStartClient(){
      base.OnStartClient();
      if(!IsOwner) return;
      ServerSetName(localPaticipantName);}
   [ServerRpc]
   private void ServerSetName(string name)
{username = name;}
```

4 Results and discussions

This section provides a discussion of the results of the proposed framework. The section will focus on two main validation cases: 1) connecting various participants in the exploring scene and 2) connecting the onsite team and logistics team in the onsite scene. Figure 6 shows a screenshot of the accessed model through the editor and web. The right-hand side of the figure shows a participant accessing the multi-player model using the Unity Editor. Establishing the server on the editor is essential to enable other clients to enter the multi-player network. The left-hand side shows a client joining the network from a web window. It is worth mentioning that both participants can do the same functions. For instance, participants can access a cloud version of the Anylogic model to try different scenarios and evaluate the project KPIs. The Anylogic button will only appear when the participant successfully signs into the blockchain network. When the first user enables the network server, they must sign to the blockchain and select the corresponding NFT. The RPC discussed earlier will assign and reflect the participant's name with the potential clients. The client will be asked to join the blockchain network and select the appropriate NFT. The smart contract in the blockchain will record these activities on the network and assign them hash values.

The real-time connectivity between the two participants and the efficiency of the established network is tested as follows. First, the movements of the two characters (with the assigned names) are tracked in the two windows to test the efficiency of the network manager in synchronizing the activities in both scenes.

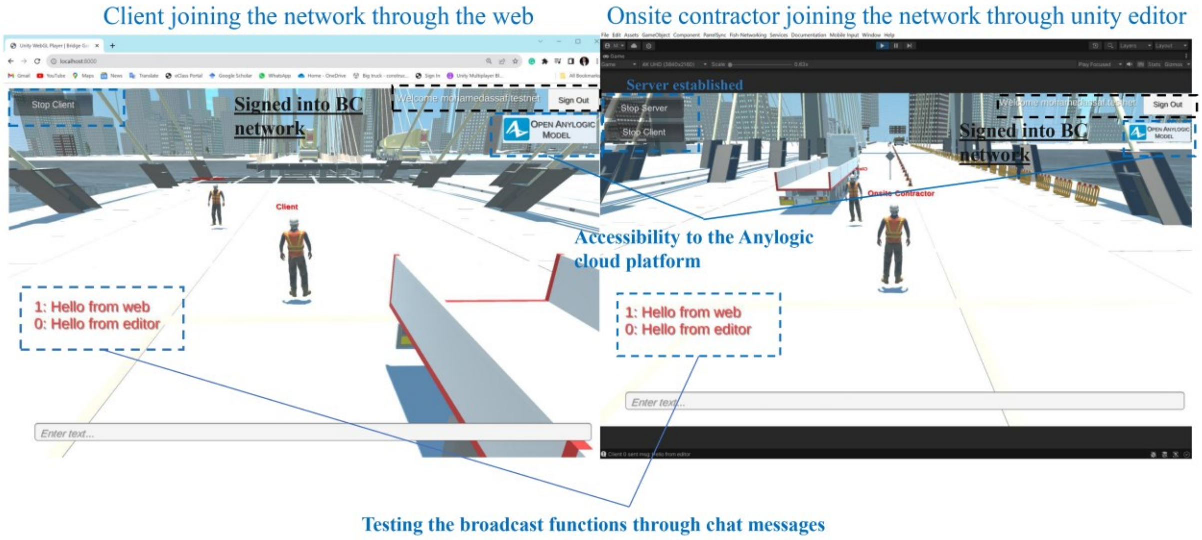

Figure 6: Real-time connectivity between different stakeholders in the web-based model

Second, the broadcast functions are tested through the chat messages between the server and the clients. These tests varied the system's abilities in providing real-time connection among various stakeholders in the developed virtual scenes. When the participant accesses the cloud version of the Anylogic model, they manipulate the sensitive parameters and have an evaluation of the discussed KPIs. Figure 7 shows a sample of the features provided by the cloud SO model. For instance, the figure shows the inventory level and the produced modules changing over time based on the participant's specific set of inputs. In addition, it also shows the modules' installation cost and the onsite work cost over time.

Figure 7: Feature samples provided by the SO model

Another testing scenario is the connectivity between the onsite team and the logistics team. In this case, a truck carrying precast elements is navigating through the city roads to reach the pickup point. The network manager is configured to a new player, which is the truck. Hence, the movement of the truck is synchronized in all of the screens. When the server is initiated, the client (logistics team) can sign into the blockchain network and join the game scene. The client then can start navigating the truck to pick up points with several metrics evaluated in real time. Some of these metrics are module clearance checks and traveled distance. The onsite team, on the other hand, can view the movement of the truck and control the movement of the onsite mobile crane. The broadcasting function is used to reflect the movement of the crane in the logistics team's view. The connections between the users are also tested using chat messages. The representative of the onsite team and the onsite team can then make collaborative decisions in a virtual replica of the MC project. Further, both teams can also access the SO cloud model to further test several decisions, such as the number of onsite cranes, number of trucks, and number of workers on site, and evaluate these decisions based on several KPIs, including produced emissions, cost, and duration. It is also worth mentioning that there is no limitation on the number of users, which means there can be more than one representative from each team in the multi-player VR model.

5 Conclusion

This study proposes a virtual reality-based blockchain model that enables multi-player networking in the developed virtual scenes. The model also supports access using web browsers to overcome the issue of limited accessibility to the VR scenes. The integration of the blockchain in the multi-player virtual environment enables a decentralized network and provides security and data transparency. It also enables the recording of all of the activities on the VR web-based model to promote trustworthiness among MC stakeholders. The model also provides participants with access to a cloud SO model that allows them to try different options and have a real-time evaluation based on a set of identified KPIs.

The multi-player networking functions were tested in this study using two main cases: 1) connecting various participants in the exploring scene and 2) connecting the onsite team and logistics team in the onsite scene. Remote procedure calls and broadcasting functions were used to test the connectivity among participants in the multi-player environment. The accessibility of the integrated cloud SO model in the web-based model was also tested and proved the model's efficiency in providing users with an assessment of various decisions. Some limitations of the current work are worth mentioning, including the following: 1) the SO cloud model has no direct connection with the blockchain network. However, both of them are integrated into the VR web-based model; 2) the scheduling data is subjected to the user input. Future work will target integrating IoT sensors to facilitate the flow of near real-time data for better data traceability and visualization.

References

[1] M. Assaf, R. Lemouchi, M. Al-Hussein, X. Li, A collaborative planning model for offsite construction based on virtual reality and game engines, Proceedings, Conference on Construction Applications of Virtual Reality, Nov. 13–16, 2023, Florence, Italy, 2023.

[2] L. Bao, S.V.-T. Tran, T.L. Nguyen, H.C. Pham, D. Lee, C. Park, Cross-platform virtual reality for real-time construction safety training using immersive web and industry foundation classes, Automation in Construction 143 (2022) 104565.

[3] M. Chessa, C. Bassano, F. Solari, A WebGL Virtual Reality Exergame for Assessing the Cognitive Capabilities of Elderly People: A Study About Digital Autonomy for Web-Based Applications, ICPR International Workshops and Challenges, Springer International Publishing, pp. 163-170.

[4] A. Ezzeddine, B. García de Soto, Connecting teams in modular construction projects using game engine technology, Automation in Construction 132 (2021) 103887.

[5] Fishnet (2019). "FishNet - Unity Networking Evolved."<https://fishnetworking.gitbook.io/docs>.

[6] M. Hussein, A. Darko, A.E.E. Eltoukhy, T. Zayed, Sustainable Logistics Planning in Modular Integrated Construction Using Multimethod Simulation and Taguchi Approach, Journal of Construction Engineering and Management 148 (6) (2022) 04022022.

[7] Y. Jiang, X. Liu, Z. Wang, M. Li, R.Y. Zhong, G.Q. Huang, Blockchain-enabled digital twin collaboration platform for fit-out operations in modular integrated construction, Automation in Construction 148 (2023) 104747.

[8] A.E. Martis, C. Bassano, F. Solari, M. Chessa, Going to a Virtual Supermarket: Comparison of Different Techniques for Interacting in a Serious Game for the Assessment of the Cognitive Status, in: S. Battiato, G.M. Farinella, M. Leo, G. Gallo (Eds.), ICIAP 2017, Springer International Publishing, Cham, 2017, pp. 281-289.

[9] J. Possik, S. Gorecki, A. Asgary, A.O. Solis, G. Zacharewicz, M. Tofighi, M.A. Shafiee, A.A. Merchant, M. Aarabi, A. Guimaraes, N. Nadri, 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, 2021.

[10] P.W.S. Putcher, J.C. Roberts, P.D. Ritsos, Immersive Analytics with WebVR and Google Cardboard, 2016.

[11] Snehlata, P. Shukla, A.K. Singh, S. Tiwari, Rishabh, V.K. Dwivedi, An Intelligent Blockchain-Oriented Digital Voting System Using NEAR Protocol, SN Computer Science 4 (5) (2023) 643.

[12] L. Wu, W. Lu, R. Zhao, J. Xu, X. Li, F. Xue, Using Blockchain to Improve Information Sharing Accuracy in the Onsite Assembly of Modular Construction, Journal of Management in Engineering 38 (3) (2022) 04022014.

[13] Wu, L., Lu, W., Xue, F., Li, X., Zhao, R., & Tang, M. (2022). Linking permissioned blockchain to Internet of Things (IoT)-BIM platform for off-site production management in modular construction. Computers in Industry, 135, 103573.

[14] Yu, X. and D. Chen, Design and construction of the Tahya Misr cable-stayed bridge in Cairo, Egypt. Proceedings of the Institution of Civil Engineers - Civil Engineering, 2020. 174(2): p. 79-88.

[15] Z. Zhang, W. Pan, Virtual reality supported interactive tower crane layout planning for high-rise modular integrated construction, Automation in Construction 130 (2021) 103854.

[16] Z. Zhang, M.O. Wong, W. Pan, Virtual reality enhanced multi-role collaboration in crane-lift training for modular construction, Automation in Construction 150 (2023) 104848.

[17] M. Assaf, S. Assaf, W. Correa, R. Lemouchi and Y. Mohamed, "A Hybrid Simulation-Based Optimization Framework For Managing Modular Bridge Construction Projects: A Cable-Stayed Bridge Case Study," 2023 Winter Simulation Conference (WSC), San Antonio, TX, USA, 2023, pp. 3094-3105