공학석사 학위논문

Integrated Method of Analysis and Visualization Based on HLA and Its Application to Collaborative Simulation in Shipbuilding

HLA를 기반으로 한 해석–가시화 통합 방법 연구 및 조선 해양 협업 시뮬레이션에의 적용

2016년 2월

서울대학교 대학원
조선해양공학과
이 성 (LI XING)

Abstract

Integrated Method of Analysis and Visualization Based on HLA and Its Application to Collaborative Simulation in Shipbuilding

During ship construction and offshore installation, more than two cranes are required in many cases. Even using one crane, some signalmen also need to assist the crane operator. And it is noticeable that the probability of accidents is considerably high due to operators' collaboration. In order to prevent potential risks and ensure safety of operation process, simulation technology is widely applied. In this study, collaborative simulation is developed which allows several workers to conduct the same operation simultaneously in visual environment and then investigate potential safety risks.

The current study can be summarized into four parts. Firstly, in order to describe the crane and block as real as possible, and make it as working in the actual operation site, VR (virtual reality) technology that could improve the sense of reality and immersion is studied. Secondly, because the cranes and block in virtual environment should move like a real movement, research on physics analysis technology based on multibody system dynamic has been done. Thirdly, to control the cranes in the simulation, workers who operate the crane need a controller. Therefore, a scenario generator is developed which can convert the signals from the controllers to the input datum for VR and analysis. Finally, in order to effectively integrate the VR technology, multibody dynamic system technology and controller as well as consider of interoperability and reuse, this study proposes an integrated simulation interface based on the High Level Architecture.

This study utilizes developed collaborative simulation, which can be applied in simulating block turn-over (rotating 90 or 180 degrees) operation and topside module installation. Each simulation allows four operators to operate under the same virtual environment simultaneously and during this process, some danger may occur due to operators' mistakes. This study makes contributions in simulating potential outcomes caused by operators' operations, and collecting detailed data for further investigation. This study can be applied in ensuring safety in complicated scenarios, training operators and many other aspects.

Keywords: HLA, Virtual Reality, Collaborative Simulation, Shipbuilding, Offshore Installation

Student number: 2014-22145

Contents

Figures

Tables

1. Introduction

1.1. Background of this study

In shipyard, a ship is constructed by joining its parts on the dock after dividing it into a number of blocks and assembling them. Recently, many shipyards produce the blocks into large sizes. These blocks are called mega blocks and the majority of them approximately weigh from 1000 to 3000 tons [1]. If the mega blocks are to be lifted, more than two cranes are required in many cases. Figure 1-1 shows an example of a ship block turn-over operation using one floating crane and two crawler cranes. However, it is noticeable that the probability of accident occurance is considerably high due to crane operators' collaboration. In order to evaluate potential risks and ensure safety of operation process, it is essential to simulate the crane operation before it is proceeded. However, most of the simulations are conducted by fixed scenarios as shown in Figure 1-2, which neglect the impact of multiple worker's operations.

On the other hand, to make offshore structure installment easy, the offshore structures are usually divided into several modules, and topside modules weighing from 3000 to over 20000 tons have been installed on offshore structure [2]. During the installation process, large offshore cranes and barge ship need to work together. Figure 1-3 shows a topside module installation using dual crane rig and barge ship. We can do simulation using physics-based analysis program by fixed scenarios as Figure 1-4 shows, but still, risks that may be caused by workers' operations are hard to be simulated or evaluated.

Therefore, this study proposes the collaborative simulation, which takes worker's operations in real-time into account.

Figure 1-1 Block turn-over using 1 floating crane and 2 crawler cranes

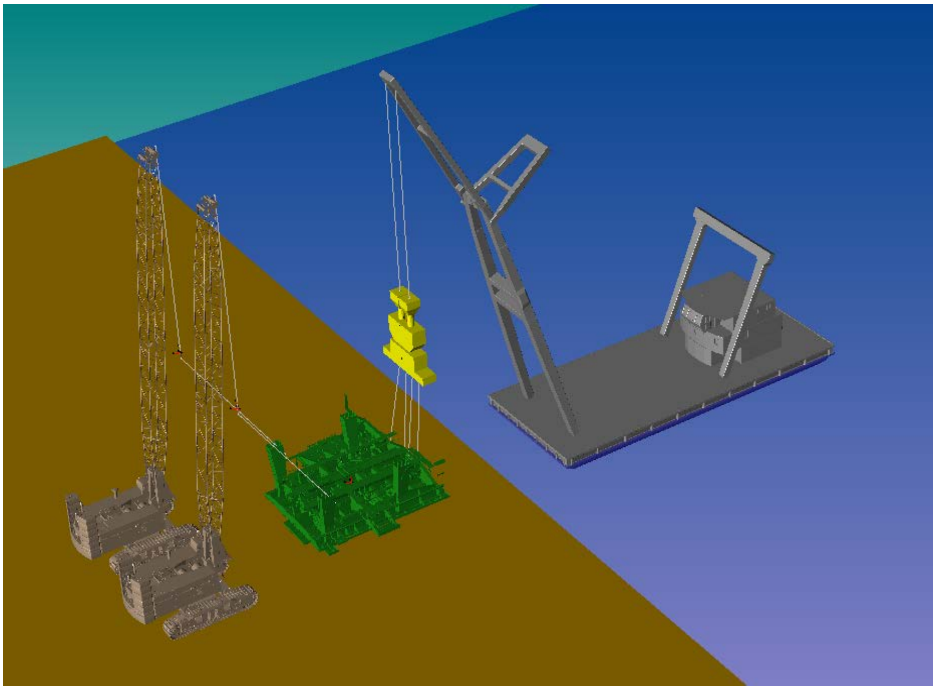


Figure 1-2 Block turn-over simulation by fixed scenarios

Figure 1-3 Topside module installation using dual crane rig and barge ship

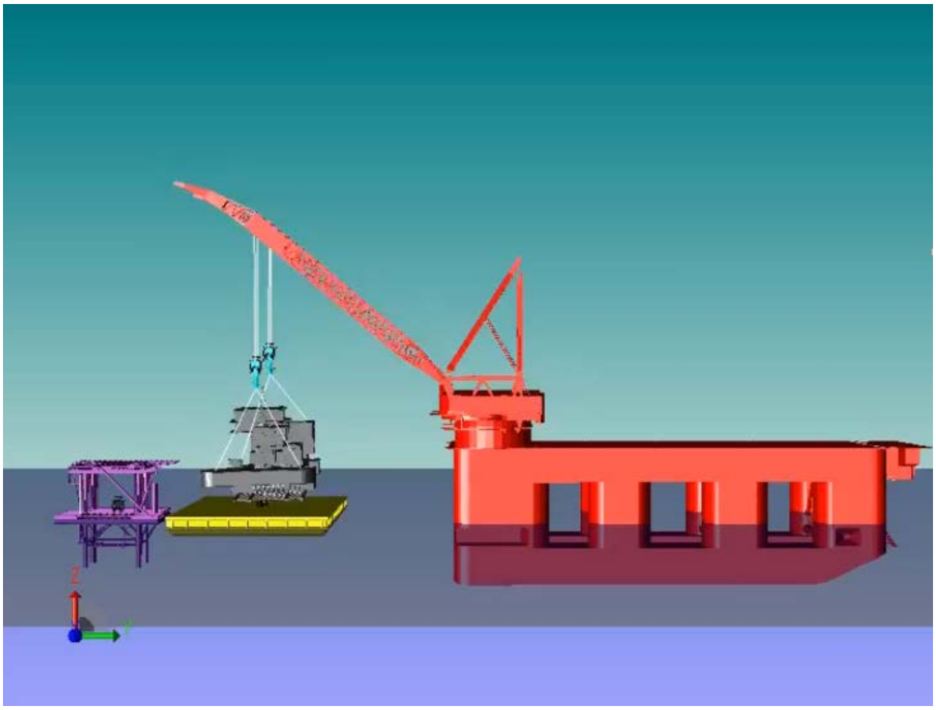


Figure 1-4 Topside installation simulation by fixed scenarios

1.2. Four technologies for collaborative simulation

To implement collaborative simulation, four technologies are required as follow.

First, workers who operate the crane need a controller, such as joystick, control lever, button etc., which is a new introduction comparing with most preciously developed simulation programs that conduct simulation with fixed scenarios. At the same time, different signals from operators are generated through the scenario generator, which can be read for other simulation components.

Second, when the controller sends signals, the cranes, wire ropes, blocks or topside modules should move like a real movement or follow the Newton's laws of motion. We need calculate the motion of them and detect collision in real-time, and the objects in the simulation will be influenced by varied external forces, such as wind, current, control, etc. In addition, since the floating crane, dual crane rig and the barge ship float on water, the forces acting on floater such as hydrostatic and hydrodynamic, also need to be considered. Thus, a physics based analysis program serves as the solution to these problems.

Third, while the workers operate the crane or ship during simulation, they should see the scene as they do in reality. To fulfill this, lifelike three-dimensional models including block model, crane models and other necessary models are imported to make the scenes more vivid. Besides, some natural effects were added in order to improve immersion effect of virtual environment. For example, when simulating the dual crane rig operating at sea, ocean, sky, sunshine, shallow and other natural effects are inserted to make the simulation more like real world. Therefore, virtual reality technology is an important part of collaborative simulation.

As mentioned before, to implement collaborative simulation, a controller, VR and physics based analysis program are demanded. However, how to connect three of them together effectively is a crutial topic of this study. For example, how to deliver the motion received from the analysis component to the VR component in real-time? Due to the VR component and analysis program developed in different languages, how to define the data types and connection rules? With such a real-time simulation, how to manage the time synchronization? Therefore, the fourth technology is to integrate all the distribution simulation components together effectively based on the high level architecture.

1.3. Related works

Bo et al. [3] developed an airport ground service visualization collaborative simulation program using three-dimensional virtual reality editing software VirTools based on HLA (High Level Architecture). Civil airport service includes many kinds of services, and in this work, the collaborative simulation mainly focuses on lift car terminal, baggage car terminal and roaming terminal. The car terminal and baggage terminal can control their movement, while the roaming terminal can control aircrafts and other ground service vehicles. In order to integrate three of them, the simulation members release and order data by RTI (Run Time Infrastructure) system. However, the simulation has not taken physics analysis into account.

Figure 1-5 Airport environment models [3]

Li et al. [4] developed an integrated platform for collaborative virtual maintenance operation and training of complex equipment. For reusability and extensibility, it is also designed and developed based on HLA. The simulation support platform is designed with the combined hierarchical structure and modularized components to support the interactive communication of heterogeneous data and information. The operators use the data glove and motion tracking system to transfer their operations. Maintenance operator module and models module are connected by RTI. Because this simulation purpose is used for training and it can track operator’s motion directly, physics analysis is not considered.

Figure 1-6 Real-time interactive control for collaborative operation and motion simulation [4]

Belmonte et al. [5] studied how to manage virtual reality devices as federate resources in a virtual world on the basis of HLA. This approach has been applied as a framework to build simulators for training workers in civil engineering. All control devices in the simulation are virtual, and there is no physical device to control the underwater robot. The trainee wears a pair of data gloves with infrared markers to control the robot. In this work, both physical analysis and collaboration have not been considered.

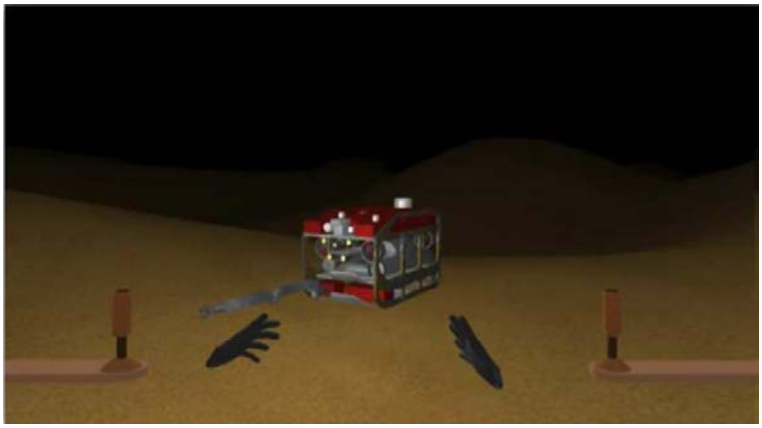


Figure 1-7 3D robot model [5]

Cha et al. [6] developed a virtual reality based fire training simulator, which provides a range of second-hand experience to the general public or inexperienced firefighters or commanders so that they can make prompt decisions as well as safe and organized responses in actual fire situations and thereby enhance human safety. In order to effectively achieve this training goal, it is crucial to reliably express fire dynamics as realistic graphics with the help of CFD (computational fluid dynamics), which is widely used to precisely predict the behaviors of fluid phenomena. Even this work has considered virtual reality technology and physics-based analysis, but collaboration and the HLA have not been included.

Figure 1-8 Fire training experience using immersive devices [6]

Ueng et al. [7], Tozzi et al. [8] and Longo et al. [9]'s works are applied to ship, naval and marine field respectively which is similar to the current study. In Ueng's work, they propose a simulation system for ship motions. By using the program, users can simulate motions of ships under the influences of waves, wind, currents, and the internal forces of the ships. This work connects VR and analysis directly but it is not a collaborative simulation.

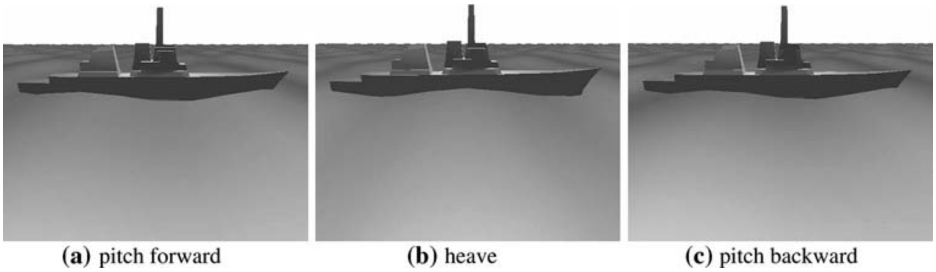


Figure 1-9 Pitch and heave motions of a war ship [7]

Figure 1-10 SAND and 3D visualization system [8]

Figure 1-11 The ship bridge replica and the visualization area [9]

In Tozzi [8] and Longo [9]'s works, physics-based analysis and VR are included, and they also use the HLA to connect every simulation component. But still, two studies have not considered collaborative operation.

1.3.1. Summary of the realted works and this study

Table 1-1 Summary of the related works and this study

	VR	Analysis	Collaborative simulation	Integrated method	Purpose
Ueng et al. [7]	O	O	X	Direct	Ship motion analysis
Belmonte et al. [5]	O	X	X	HLA	Simulation of under-water robot
Bo et al. [3]	O	X	O	HLA	Simulation of civil aircraft ground services
Cha et al. [6]	O	O	X	Direct	Training in fire drill
Li et al. [4]	O	X	O	HLA	Collaborative maintenance training of complex equipment
Longo et al. [9]	O	O	X	HLA	Training in marine ports
Tozzi et al. [8]	O	O	X	HLA	Evaluation of naval operations
This study	**O**	**O**	**O**	**HLA**	**Simulation of shipbuilding and offshore installation**

1.4. Overview of this study

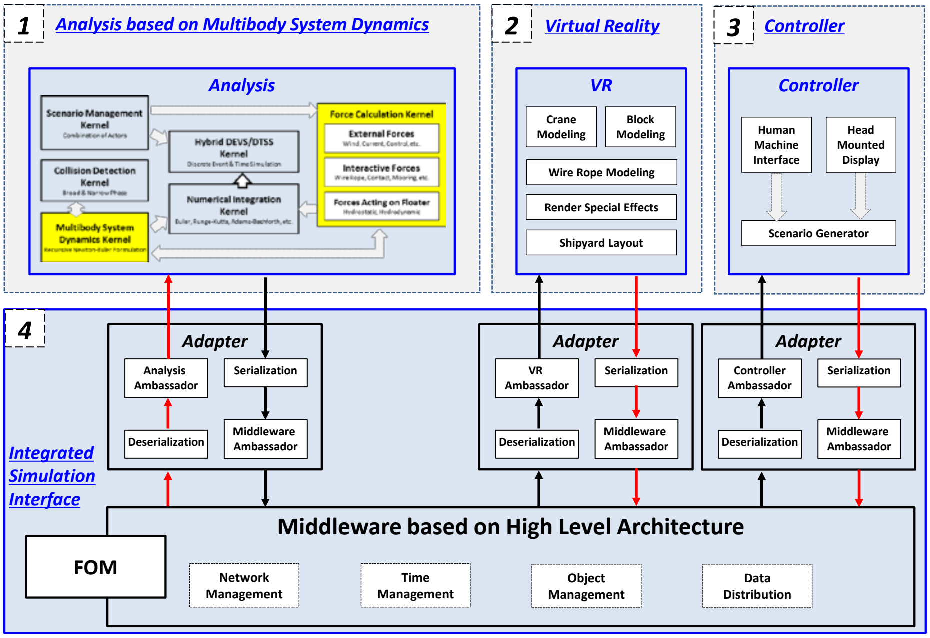


Figure 1-12 Overview of this study

Figure 1-12 shows the conceptual structure of this current study, which demonstrates the collaborative simulation system with four major techniques.

Firstly, for analyzing the motion in reality, such as dynamic motion of floating crane and block, wire rope tension, collision detection, and so on. This study uses the in-house physics-based analysis program (Figure 1-12 (1)) based on MBD (multibody dynamic system), which is a system that allows multiple bodies to be connected together with joints or springs. Additionally, this program composes of scenario management kernel, collision detection kernel, multibody system dynamic kernel, numerical integration kernel, hybrid DEVS/DTSS kernel and force calculation kernel. The program completes all the physics analyses.

Secondly, Figure 1-12 (2) shows a part of models which are imported from ship CAD for providing highly immersion for operators. And this study employs the professional 3D computer graphics program 3DMAX to model and uses the cross-platform game engine Unity3D to create virtual scenes. During analysis process, the models applied are simplified and idealized with the purpose of increasing the speed of analysis. On the other hand, while the models are used for VR, more realistic effects can be produced by touch-up, texture mapping, and rendering.

Thirdly, as shown in Figure 1-12 (3), for delivering the operating signals from operators to other simulation components, this study develops a scenario generator and utilizes the signals from HMD (human machine interface) and HMI (head mounted display) to generate various scenarios for virtual reality and analysis.

Fourthly, integrated simulation interface is developed based on the HLA (High Level Architecture) to effectively integrate all the simulation components together as well as consider of interoperability and reuse as shown in Figure 1-12 (4).

2. Analysis based on multibody system dynamic

2.1. Introduction to multibody system

Multibody system is defined as a collection of interconnected rigid bodies, which can move relative to one another, consistent with joints that limit relative motion of pairs of bodies. Forces and moments are caused by compliant elements like spring, dampers, tires, shock absorbers, actuators, or rods and other elements, that give rise to reaction forces and moments. The compliant elements themselves are generally assumed to be massless. The forces they exert on bodies of the system depend upon the relative position and velocity of the bodies [10]. To summarize, Multibody system is a system in which multiple bodies are connected together with joints or spring.

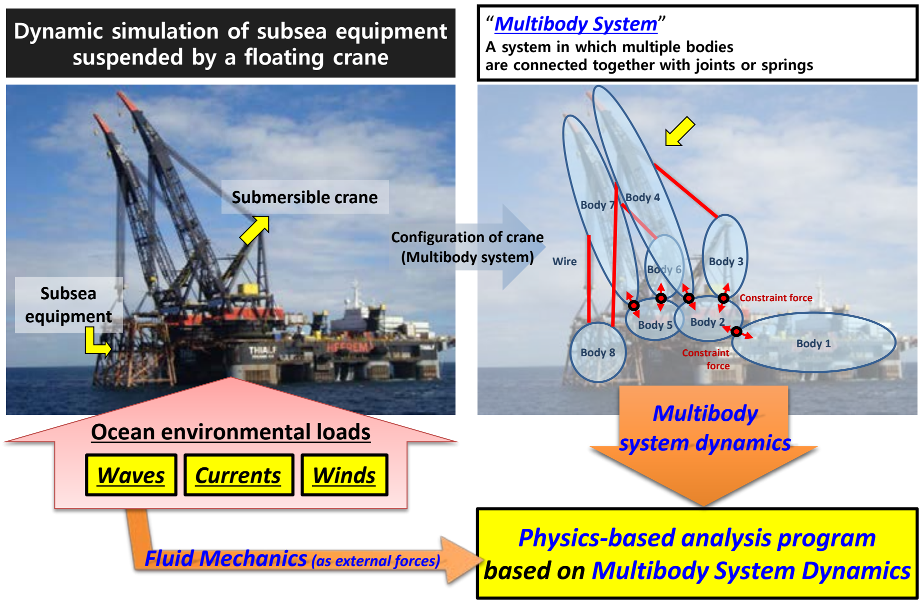


Figure 2-1 Concept of multibody system

Figure 2-1 displays a multibody system. It is a dynamic simulation of subsea equipment suspended by a dual crane rig. When evaluate dynamic response of subsea equipment, the dual crane rig should be modeled as a multibody system. In this system, eight bodies are connected together with joints or springs.

2.2. Configuration for the implementation of physics-based analysis program

In this study, physics-based analysis program, which has been developed for five years in-house, is used for motion analyzing. And the configuration for the implementation of this program is shown in Figure 2-2.

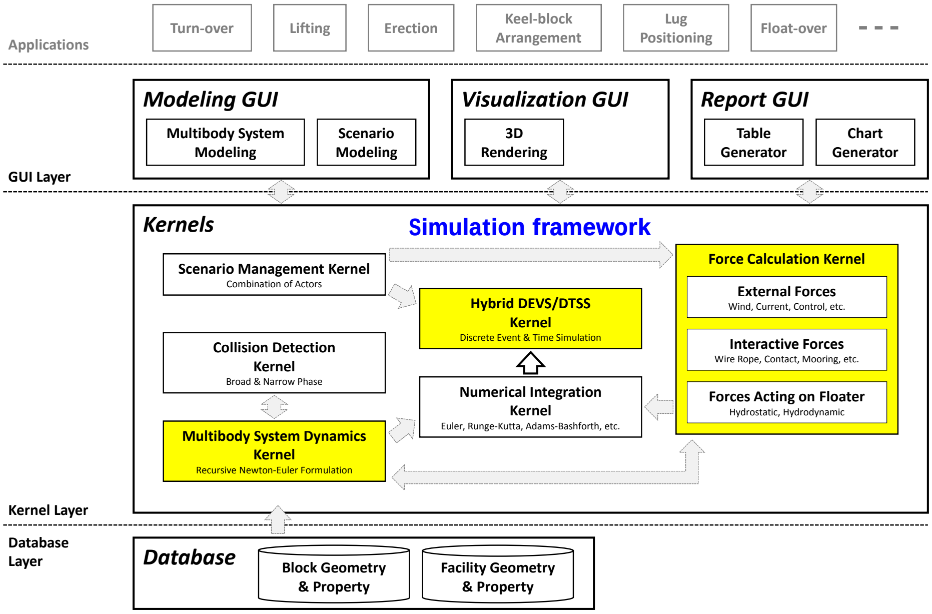


Figure 2-2 Configuration for the implementation of physics-based analysis program

The developed program has three layers: the Database, Kernels, and Graphic User Interface (GUI). The core function of the kernels layer is to simulate the operations in shipyards and offshore industries. It has six components: the multibody system dynamics kernel, force calculation kernel, numerical analysis kernel, hybrid discrete event system specification (DEVS)/discrete time system specification (DTSS) kernel, scenario management kernel, and collision detection kernel. The GUI layer is based on these kernels and supports the users' simple simulations from various cases in shipyards or offshore industries. In addition, the geometric properties of the block and facilities in shipyards are needed to configure the simulation for the production of ships and offshore plants, so these are managed in the database layer.

As mentioned before, the program has six core kernels, and the function and role of each kernel are as follows.

2.2.1. Multibody system dynamics kernel

The crane systems in shipyards are all multibody systems in which multiple rigid bodies are joined together. The computer methods used in the automated dynamic analysis of multibody systems that consist of rigid bodies are generally classified into two main approaches. In the first approach, the configuration of the system is identified using a set of Cartesian coordinates that describe the locations and orientations of the bodies. In the second approach, relative joint variables are used to formulate a minimum set of differential equations of motion, and two types of formulation use relative joint variables: embedding formulation and recursive formulation.

The dynamics of a rigid body system are described according to the system's equations of motion, which specify the relationship between the forces that act on the system and the accelerations they produce.

2.2.2. Force calculation kernel

The force calculation kernel calculates the forces that act on the multibody system. Some force act on a rigid body of the system, and other force calculation kernel has three kinds of forces: external force, interactive forces, and forces that act on a floater such as a ship or an offshore plant.

External forces that act on floater such as a ship or an offshore plant. External forces act on a body or joint. Some forces such as wind and current forces are caused by the environment, and other forces, such as the controlling force, are planned by an engineer.

Interactive forces act on two or more bodies by interconnecting them with a certain mechanical system (for example, a wire rope or a spring) or by colliding them. A tension force caused by a wire rope, a contact force caused by colliding two rigid bodies, and a mooring force are examples of interactive forces. Moreover, a load balancing system, called an “equalizer” in shipyards, is also considered as a type of this kernel. To apply equivalent tensions to all wires, the length of each wire is controlled in proportion to the tension that acts on it after classifying it as an equalizer.

The forces that act on a floater are only existing forces on the domain in ships and offshore plants. There are two kinds of these forces: hydrostatic and hydrodynamic forces. The hydrostatic force applied to a floating body is calculated while considering the body’s instantaneous position. The hydrodynamic force is calculated in the time domain using the 3D Rankine panel method. The hydrodynamic force exerted on a floating body can be determined by integrating the pressure over the wet surface.

2.2.3. Numerical analysis kernel

To simulate and analyze the dynamic phenomena in the shipbuilding process for each time unit, a numerical analysis kernel was developed in this program. Because of some strict cases of simulation for the production of ships and offshore plants, the numerical analysis kernel provides various numerical integration methods such as the Euler method, Runge-Kutta method, Adams-Bashforth method, and Hilber-

Hughes-Taylor method. The Runge-Kutta method is one of the best solutions for numerical integration, but the target block in shipyards is much heavier than the spring coefficient of wire ropes. Thus, the numerical integration of the operation in shipyards is often unstable when the Runge–Kutta method is used. Because of this problem, this system supports other numerical integration methods for the achievement of numerical stability, and the user can choose one method depending on his or her case. Using the kernel, the integrator integrates the equations of motion and calculates the position and velocity of each body in the given systems at each time unit.

2.2.4. Hybrid DEVS/DTSS kernel

There are two kinds of simulation. A simulation in which the state of a model changes by means of any event is called ‘‘discrete event simulation.’’ The discrete event simulation processes events, which change the state variables of a model, in the order in which they occur.

A simulation that calculates the state of a model at each unit time is called ‘‘discrete time simulation.’’ It is mostly used to analyze dynamic or mechanical systems because it calculates the state of a model at each time unit.

2.2.5. Scenario management kernel

The scenario management kernel can help configure a scenario of heavy load operations in shipyards or offshore industries. By analyzing a sequence of motions in a multibody system, the scenario can be specified by the combination of unit actions. A unit action is user-defined input that acts on subcomponents of a multibody system.

2.2.6. Collision detection kernel

To minimize the risk during the production of ships and offshore plants, the interferences among the bodies and wire ropes must be checked. One of the functions of the collision detection kernel is to detect their collision with other bodies and wire ropes during the simulation.

2.3. Procedure for solving equations of motion in physics-based analysis program

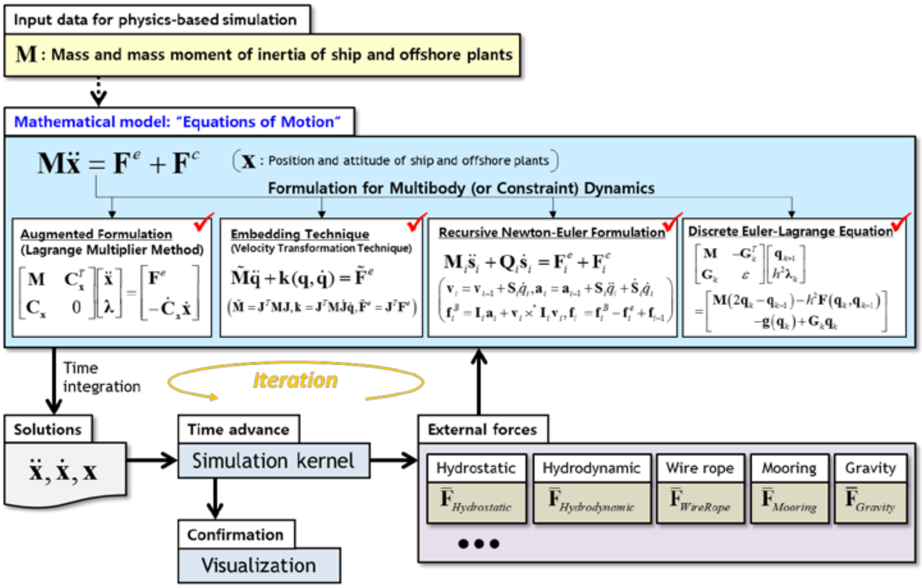


Figure 2-3 Procedure for solving equations of motion in physics-based analysis program

In this study, in order to analyze the motions of multibody system, the equations of motion have to be solved. The procedure for solving equations of motion in physics-based analysis program is shown in Figure 2-3. First, the input data, such as the mass and mass moment of inertia of ship and offshore plants are determined. This information is entered into the equation of motion which is based on multibody system dynamic. The hydrostatic force, the hydrodynamic force, the wire rope force, the mooring force, and gravity are calculated and entered into the equation of motion. After the acceleration is obtained from the equation of motion, the velocity and the position is visualized. The simulation kernel advances the simulation time and enters the next simulation time into the equation of motion and the external forces. In this way, simulation is performed by iteration.

2.4. Modeling method in physics-based analysis program

The basic models applied in this study come from shipbuilding CAD, 3D MAX or model website. Both analysis program and VR, same models are used with same layout. However， in analysis program, considering the speed of analysis, only the models which are required to analyze their motion have been used. And also the models applied are simplified and idealized to increase the speed of analysis. On the other hand, while the models are used for VR, more realistic effects can be produced by touch-up, texture mapping, and rendering.

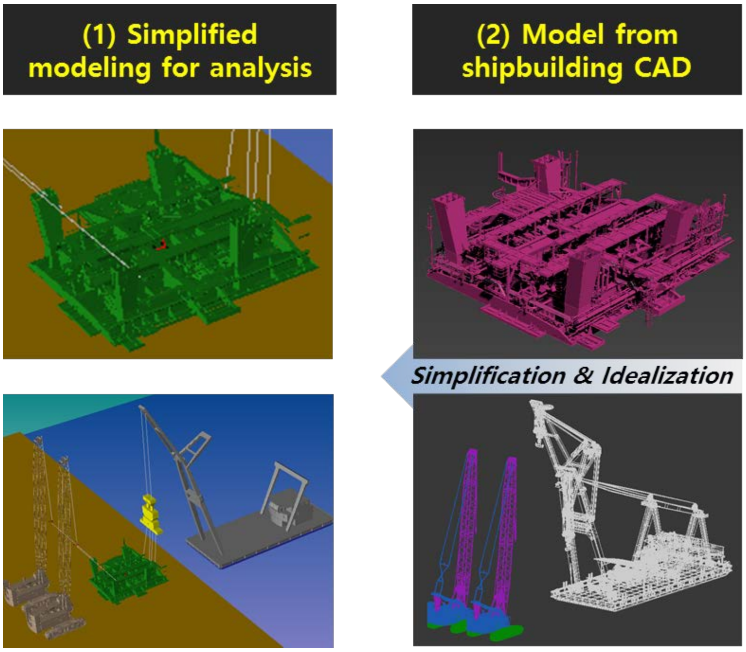


Figure 2-4 Simplification and idealization models for analysis & model from shipbuilding CAD

Figure 2-4 (1) suggests the models after simplification and idealization, while Figure 2-4 (2) shows the basic models from shipbuilding CAD. Before the simplification, block and cranes have sufficient details, such as the details of the pipeline, the topside structure of the floating crane and so on. 3DMAX or other 3D computer graphics program are applied to simplify models. Some objects or meshes can be removed and the model format should be exported to STL files which can be imported into the analysis program.

In the application of the ship block turn-over operation, the block, the wire rope, the floating crane, the crawler cranes and the hooks are required to be analyzed and the motions or informations of them should be delievered to other VR simulation components. The analysis of land and the ocean is also necessary. For example, collision between land, crawler cranes and block should be detected and the fluid forces of the ocean will act on the floating crane. However, during simulation, VR simulation components do not demand additional requirement of analyzing the land and ocean information, because the result of collision detection or external force has covered how land and ocean act on these objects, such as the crawler cranes and the floating crane in this case.

3. Visualization for virtual reality

3.1. Introduction to virtual reality

Although the analysis program can elicit visual output, the effect is not realistic enough to simulate real operation environment. In order to make immersive visual effect, the same layout is applied by both analysis program and VR simulation components. To be specific, the analysis component is for analyzing and outputting results, and the VR simulation components, which receive results from analysis component, are responsible for creating realistic visual effects in real-time.

In this study, visualization is an ensential part of collaborative simulation, during simulation, operators should see the sense which can bring them reality and immersion of virtual environment. The recent, groundbreaking development of virtual reality (VR) technologies have given rise to computer-human interaction technology that allows actual users to participate in a virtual world reproduced by computers.

Virtual reality, which refers to as immersive multimedia or computer-simulated reality, replicates an environment that simulates a physical presence in places in the real world or an imagined world, allowing the user to interact in that world. Virtual realities artificially create sensory experiences, which can include sight, hearing, touch, and smell. Most up-to-date virtual realities are displayed either on a computer screen or with special stereoscopic displays, and some simulations include additional sensory information and focus on real sound through speakers or headphones targeted towards VR users. Some advanced haptic systems now include tactile information, generally known as force feedback in medical, gaming and military applications. Furthermore, virtual reality covers remote communication environments which provide virtual presence of users with the concepts of telepresence and telexistence or a virtual artifact (VA) either through the use of standard input devices such as a keyboard and mouse, or through multimodal devices such as a wired glove or omnidirectional treadmills. The simulated environment can be similar to the real world in order to create a lifelike experience—for example, in simulations for pilot or combat training—or it can differ significantly from reality, such as in VR games [12].

Recently, groundbreaking development of virtual reality (VR) technologies has given rise to computer–human interaction technology that allows actual users to participate in a virtual world reproduced by computers. This highlights the need for VR-based training simulators enabling safe, convenient, and planned repetitive training. Training simulators, in general, should have basic functions such as scenario generation and control, virtual reality content representing actual situations, immersive virtual reality interface devices, and the capability of evaluating the training process and results. In particular, realistic virtual reality content should be expressed on a real-time basis under the given training scenarios depending on the locations and viewpoints of users' choice. The content is offered to trainees via immersive interface devices such as a HMD (Head-Mounted Display), large screen, and tracker to maximize the sense of presence and interactivity; trainee responses and mission fulfillment data are recorded and analyzed for overall training evaluation [6].

3.2. Modeling method of virtual reality

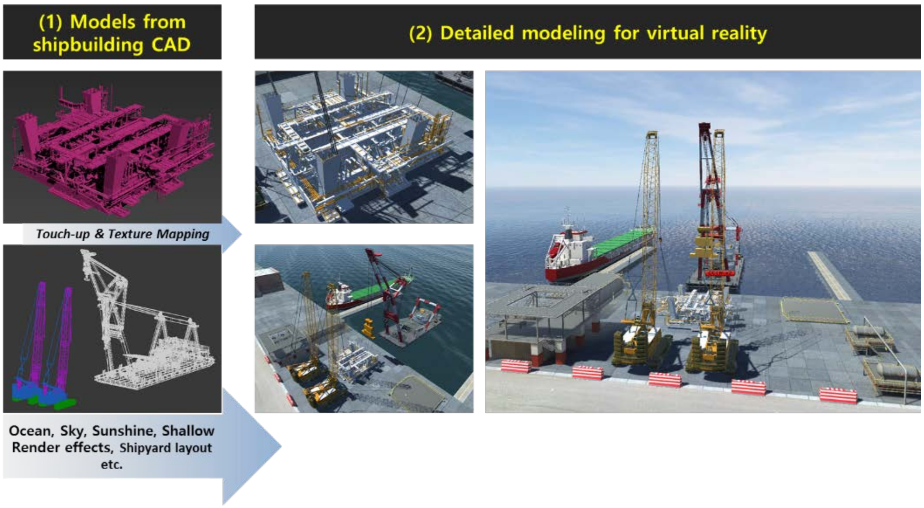


Figure 3-1 Models from shipbuilding CAD & detailed modeling for VR

Figure 3-1 (1) shows the basic models from shipbuilding CAD. Figure 3-1 (2) shows the detailed modeling for VR. In order to create a more realistic operation scene in the virtual environment, touch-up and texture mapping are applied by using 3DMAX or Unity3D. Moreover, to establish the operation scenes as real as possible, an addition of ocean, sky, sunshine, shadows, and other render effects have been added, and the shipyard has been laid out by setting other equipments, buildings and so on.

4. Controller for interactive user input

In this study, two kinds of hardwares are used for controller simulation component. The cranes, ships or wire ropes in virtual environment are controlled using joystick, the positional tracking of HMD (Head Mounted Display), which can track the real head movements of the operators.

4.1. Hardware parameters

4.1.1. Joystick

Figure 4-1 Logitech Extreme™ 3D Pro joystick

Figure 4-1 shows the joystick used in this study and the model is Logitech Extreme™ 3D Pro. It has 12 programmable buttons, 8-way hat switch and rapid-fire trigger.

4.1.2. Head mounted display

Figure 4-2 Oculus Rift Development Kit 2 and positional tracking

Figure 4-2 shows the HMD used in this study and the model is Development Kit 2 (DK2). The DK2 consists of two parts, which are the OLED display that is 1920 pixels wide × 1080 pixels high (960 × 1080 per eye) with a 100° field of view, and the positional tracking that can track the operators' real head movements. To be noticed, the positional tracking is a component of the controller while the OLED display is just for displaying 3D scenes.

4.2. Controller for scenario generation

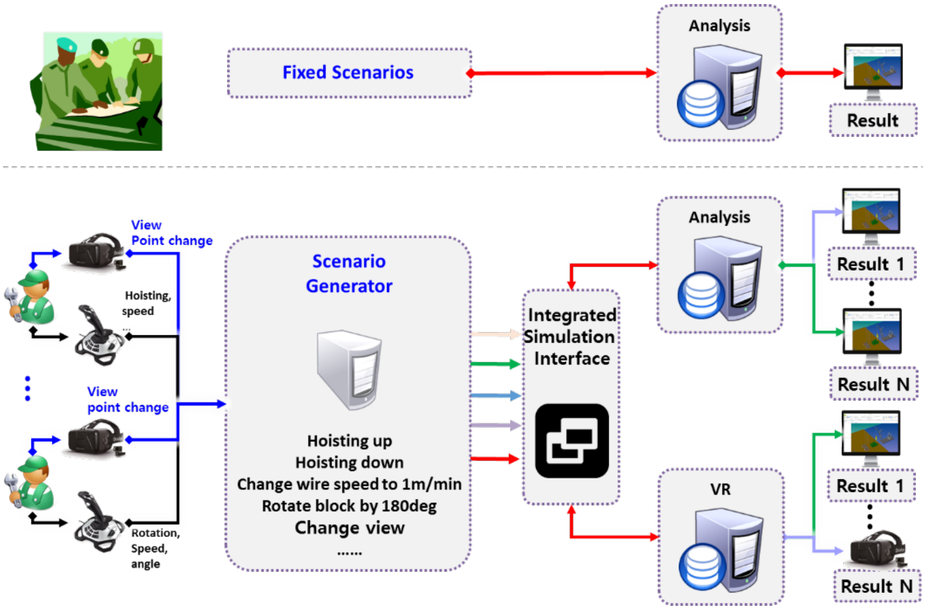


Figure 4-3 Controller for scenario generation

Usually, the results from analysis programs will be the same because the simulation is conducted with fixed scenarios. However, in this study, joystick is used to control the crane and other objects, and the operators are able to observe the scenes from every angle with the HMD (Head Mounted Display), which is used for tracking view angle signals and displaying the operator scenes. At the same time, the signals from joystick or motion tracking of HMD cannot be directly recognized by analysis component or VR simulation components. To deal with this problem, a scenario generator that allows different signals from operators to be generated in different scenarios in real-time is developed, and the generated scenarios can be read for other simulation components.

For example, when pushing the joystick, the signal generated from the joystick is a number between -1 to 1 in accordance with direction and force difference of the operation. Then the scenario generator will read this signal and change wire speed or other properties consequently.

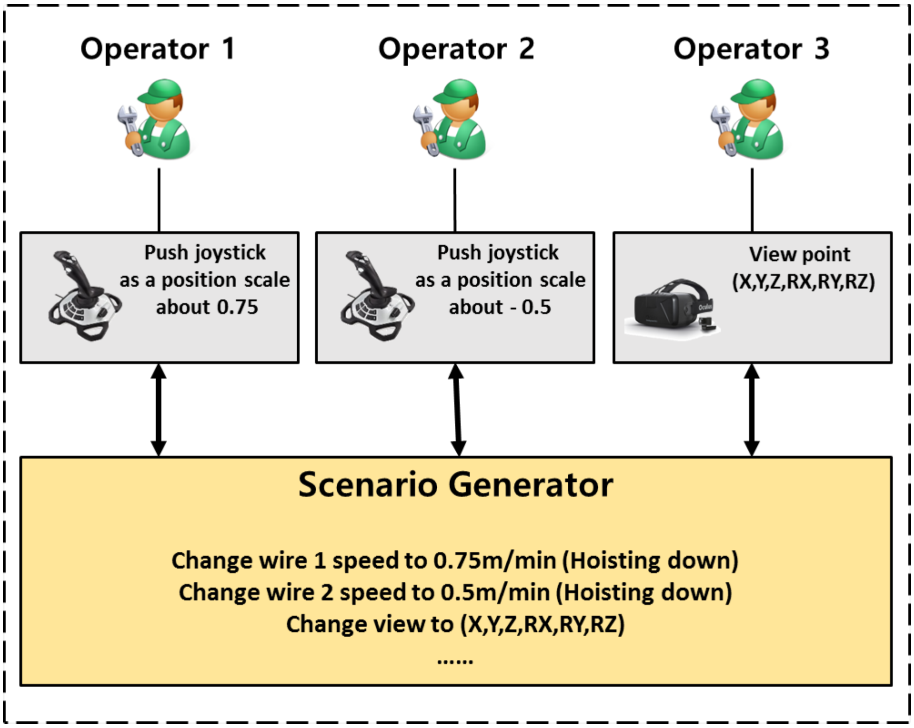


Figure 4-4 Example for scenario generator

When operator 1 push the joystick which generates a position scale about 0.75, operator 2 push the joystick which generates a position scale about -0.5, and the positional tracking of HMD detected the view point (X,Y,Z,RX,RY,RZ), the scenario generator can generated scenarios as shown in Figure 4-4.

By using controller, HMD, scenario generator, and the operators will feel like operating the real crane in the virtual environment.

5. Integrated simulation interface based on high level architecture

5.1. Necessity of HLA

In this study, collaborative simulation consists of three simulation components: controller, analysis program and VR. Since at least two workers are required, every worker has two simulation components which are controller and VR. Simulation requires a minimum of five components and the exchange of information between five of them in this case as shown in Figure 5-1.

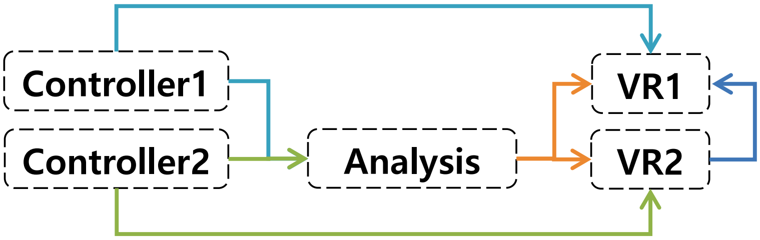


Figure 5-1 Information exchange by two workers in collaborative simulation

When one more worker is added, two simulation components should be added and many items have to be changed as shown in Figure 5-2. So if more than one workers are added, it will be very complicated and inefficient. Basically, network is required because simulation components are participated in distributed environment. The analysis simulation component and VR simulation component are developed in different languages or program, therefore, data types and connection rules also should be well defined.

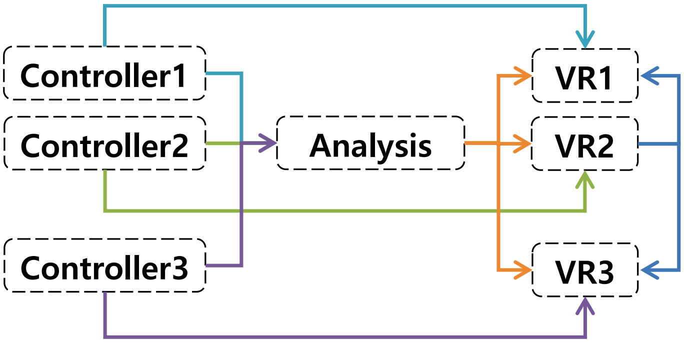


Figure 5-2 Connection between a new worker and existing simulation components

To solve these problems, not only should all the distribution simulation components be integrated together effectively, interoperability and reuse should be considered for the collaborative simulation. An integrated simulation interface consisting of an integrated simulation middleware and an adapter can be used, as shown in Figure 5-3.

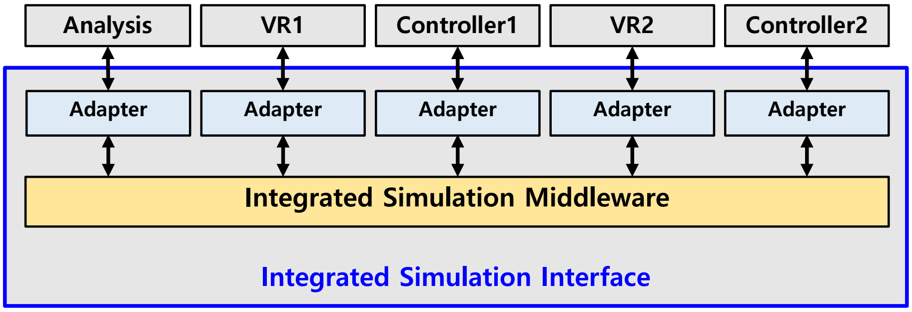


Figure 5-3 Integrated simulation interface having the integrated simulation middleware and the adapters

The simulation components are connected with integrated simulation middleware rather than connecting with each other directly and transferring data from one to another. It also synchronizes the overall simulation time even though it is not in the same computer. The adapter is plugged in each simulation component in order to be connected with the middleware. If new simulation components are added, the only work needed is to build connection between the new components and the middleware, as shown in Figure 5-4. It is obviously more efficient contrasting with the example in Figure 5-2.

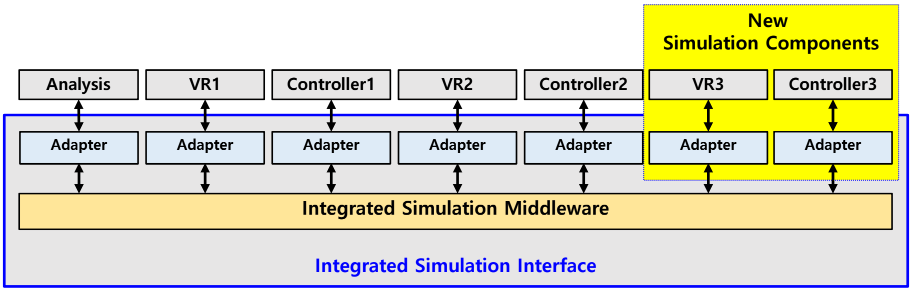


Figure 5-4 Addition of new simulation components by using the integrated simulation interface

This explains the proposal of the integrated simulation interface in this study, and HLA is such a general-purpose architecture for distributed computer simulation systems which can solve these problems.

5.2. Technical overview of HLA

HLA (High Level Architecture) is an open international standard, developed by the Simulation Interoperability Standards Organization (SISO) and published by IEEE. HLA is a standard document that describes the components of HLA and what interfaces and properties they must have. Anyone can develop any software component of HLA. Implementations of the different components are available among commercial companies, governments, academia, and open source developers. Two of the most common cases for HLA are training and analysis, other cases using HLA include test, engineering, and concept development [15].

5.2.1. Interoperability and reuse

HLA is an architecture that enables several simulation systems to work together. This is called interoperability, a term that covers more than just sending and receiving data. The systems need to work together in such a way that they can achieve an overarching goal by exchanging services. There are two reasons that need to connect several simulation systems. First, an organization may already have a number of simulations systems that need to be used together with newly acquired simulations or simulations from other organizations. Second, there may be a requirement to simulate a “bigger picture”, where models form different organizations interact. Experts form different fields need to contribute different models. In many cases it would also be a monumental task to build one big system that covers everything compared to connecting several different simulations.

One important principle of HLA is to create a federation of systems. Most simulation systems themselves are highly useful. With HLA, we can combine them to simulate more complex scenarios and chains of events. HLA enables us to reuse different systems in new combinations [15].

5.2.2. Important concepts of HLA

The Runtime Infrastructure (RTI). It is a piece of software that provides the HLA services. One of them is to send the right data to the right receiver.

The Federate. It is a system connecting to the RTI, typically a simulator. Each federate can model any number of objects in a simulation. It can, for example, model one aircraft or hundreds of aircrafts. Other examples of federates are general tools like data loggers or 3D visualizers.

The Federation. It covers all the federates, the RTI that they connect to and the FOM that they use. This is the group of systems that interoperate.

The Federation Object Model (FOM). It is a file that contains a description of the data exchange in the federation, for example the objects and the interactions that will be exchanged. This can be seen as the language of the federation.

The Simulation Object model (SOM). It is very similar to a FOM. It is based on the same format, the HLA object model template. It describes what information one particular federate can publish and subscribe.

5.2.3. HLA components

The HLA is defined by three components:

(1) HLA rules

The HLA provides a set of ten rules that together ensure the proper interaction of federates in a federation and define the responsibilities of federates and federations.

The rules of federations are as follow:

1) Federations shall have an HLA FOM, documented in accordance with the HLA OMT.

2) In a federation, all simulation-associated object instance representation shall be in the federates, not in the RTI.

3) During a federation execution, all exchange of FOM data among joined federates shall occur via the RTI.

4) During a federation execution, joined federates shall interact with the RTI in accordance with the HLA interface specification.

5) During a federation execution, an instance attribute shall be owned by at most one joined federate at any given time.

The rules of federates are as follow:

6) Federates shall have an HLA SOM, documented in accordance with the HLA OMT.

7) Federates shall be able to update and/or reflect any instance attributes and send and/or receive interactions, as specified in their SOMs.

8) Federates shall be able to transfer and/or accept ownership of instance attributes dynamically during a federation execution, as specified in their SOMs.

9) Federates shall be able to vary the conditions (e.g., thresholds) under which they provide updates of instance attributes, as specified in their SOMs.

10) Federates shall be able to manage local time in a way that will allow them to

coordinate data exchange with other members of a federation.

(2) Object model template (OMT)

The OMT is a necessary basis for reuse and forms a documentation standard describing the data used by a particular model.

(3) Interface specification

The interface specification addresses interoperability and describes a generic communications interface that allows simulation models to be connected and coordinated. Although the HLA is an architecture, not software, use of runtime infrastructure (RTI) software is required to support operations of a federation execution. The RTI software provides a set of services, as defined by the federate interface specification, used by federates to coordinate operations and data exchange during a runtime execution.

5.2.4. Management areas of RTI

The federate interface specification defines the functional interface between federates and the RTI. The RTI provides services to join federates in a way that is analogous to how a distributed operating system provides services to applications. The services are arranged into seven basic groups.

(1) Federation management

These services allow for the coordination of federation-wide activities throughout the life of a federation execution. Such services include federation execution creation and destruction, federate application joining and resigning, federation synchronization points, and save and restore operations.

(1) Declaration management

These services allow joined federates to specify the types of data they will supply to, or receive from, the federation execution. This process is done via a set of publication and subscription services along with some related services.

(2) Object management

These services support the life-cycle activities of the objects and interactions used by the joined federates of a federation execution. These services provide for registering and discovering object instances, updating and reflecting the instance attributes associated with these object instances, deleting or removing object instances as well as sending and receiving interactions and other related services.

(3) Ownership management

These services are used to establish a specific joined federate's privilege to provide values for an object instance attribute as well as to facilitate dynamic transfer of this privilege (ownership) to other joined federates during a federation execution.

(4) Time management

These services allow joined federates to operate with a logical concept of time and to jointly maintain a distributed virtual clock. These services support discrete event simulations and assurance of causal ordering among events.

(5) Data distribution management

These services allow joined federates to further specify the distribution conditions (beyond those provided via Declaration Management services) for the specific data they send or ask to receive during a federation execution. The RTI uses this information to route data from producers to consumers in a more tailored manner.

(6) Support services.

This group includes miscellaneous services utilized by joined federates for performing such actions as name-to-handle and handle-to-name transformations, the setting of advisory switches, region manipulations, and RTI startup and shutdown.

5.3. Runtime infrastructure (RTI)

5.3.1. Process for selecting RTI

As mentioned before, RTI is a piece of software that provides HLA services. It is important to consider the price and whether the development environment supports the C# language when selecting the RTI for the collaborative simulation. Most of the RTI support C, C++ or Java, only very few RTI support C#, but few people use them. Not only RTI but also an object model editor is required, which can help us to develop the FOM and SOM correctly. Unfortunately, almost no RTI can meet the requirement. But if using RACON, SimGe and Portico together, the problem can be solved.

5.3.2. RACoN

The RTI abstraction component for .NET (RACoN) is an open-source library that deals with the HLA RTI level communication in order to access the federation-wide execution. RTI is middleware broker software that manages the federation execution and object exchange through a federation execution. The RACoN provides the .NET wrapper classes for the RTI and RTI-specific data structures. As it offers more maintainable, robust, and portable methods, developing an abstraction layer (wrapper) over RTI is a popular approach in many HLA-based projects.

Currently, the main limitations of RACoN are as follows. RACoN supports only HLA13 standard and it does not support the full HLA13 federate interface services. RACoN supports DMSO RTI 1.3NGv6 API and Portico 2.0.1 x86 API. This imposes the installation of DMSO 1.3NG or Portico libraries prior to RACoN.

5.3.3. Simulation Generator (SimGe)

In order to develop the collaborative simulation based on HLA, the FOM and SOM are also required to develop. The FOM is a file that contains a description of the data exchange in the federation, while the SOM is very similar to FOM which is based on the HLA object model template.

The RACoN developer also provides a HLA object model editor, simulation design and development environment, as well as a code generator that is intended to generate code automatically for HLA-based distributed simulations. SimGe includes an object model editor (OME), a report generator, and a code generator. SimGe OME allows the user to manage the object model and enables the creation and modification of HLA object model template and object models and the import and export of the HLA related files (i.e. Federation Execution Data (FED), Federation Document Data (FDD)). The code generator automatically generates code for the target platform RACoN.

FED/FDD is a skeleton and an agreement among federates what to share within a federation. The federates are responsible to implement the required data structures to represent this agreement. When a federate discovers an object, what it will do with it totally depends on the implementation of the federate.

5.3.4. Portico

Portico is an open source, cross-platform, fully supported HLA RTI implementation. Designed with modularity and flexibility in mind, Portico is a production-grade RTI for the Simulation and Training Community [18].

Even RACoN supports DMSO RTI 1.3NGv6 API and Portico 2.0.1 x86 API. During using DMSO RTI, some exceptions and errors occurred, and the RtiExec, an RTI executable process, must run during each simulation. However, for the Portico, it no longer run a central RTI, but all federates operate in a peer-to-peer manner so that there is no need to start an RTI. Therefore, Portico is used in this study.

5.4. Case study: chat federate application

The subsequent sections explain how to develop a federate applicaition by using the RACoN, SimGe, and Portico in detail, accompanied by an example, the chatting federation is implemented using RACoN library and it is provided by RACoN.

Chatting is an HLA based distributed interactive application that provides basic chatting functionalities such as selecting a nickname, entering a chat room, and so on. By using the chat client, a federate application, one can exchange messages with his/her friends in chat rooms. Before entering a chat room, he/she has to pick up a unique nickname. The "chat client" provides a graphical user interface for the user interaction and deals with the RTI communication. The conceptual view of the application is presented in Figure 5-5. In this case, the whole chatting is the federation and the federates are chat client A and B.

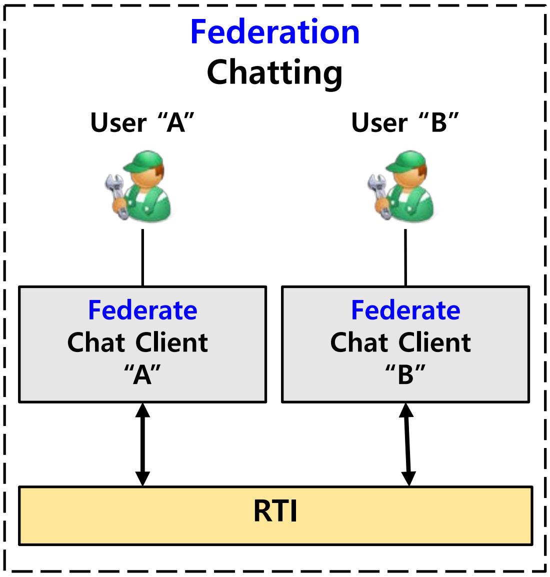


Figure 5-5 Chat application conceptual view

5.4.1. Object model

The first thing to do is to develop the SOM and FOM. The object classes are presented in Table 5-1, and the interaction class and its parameters are presented in Table 5-2.

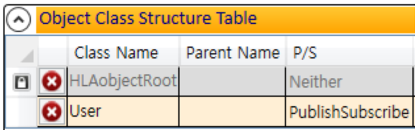
Table 5-1 The object classes in chatting federation

Object Class	Attributes	Type
User	Nickname	String
	Status	Enum{READY,INCHAT}

Table 5-2 The interaction classes in chatting federation

Interaction class	Parameters	Type
Chat	Sender	String
	Message	String
	TimeStamp	DateTime

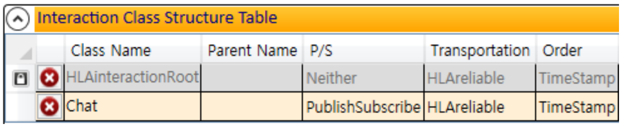
In SimGe, the object class can be edited as shown in Figure 5-6, and Publish/Subscribe should be defined, in this case, the user class is publish and subscribe. Figure 5-7 shows the interaction class and the class name is Chat, both user object class and chat interaction class are a root class, the parent name will be set to root by default. The user object class attributes and data type are shown in Figure 5-8, in this table, the hierarchical relationships should be defined, because the attributes nickname and status have an affiliation with object class user. Therefore, the object column in the attribute table is set to user. Similarly, the interaction class has an affiliation with parameters (i.e. Sender, Message and TimeStamp), so the interaction column in parameter table as shown in Figure 5-9 are set to chat. Figure 5-10 and Figure 5-11 shows the data types are created which are used by attribute table and parameter table.



Object Class Structure Table

Class Name	Parent Name	P/S
HLAobjectRoot		Neither
User		PublishSubscribe

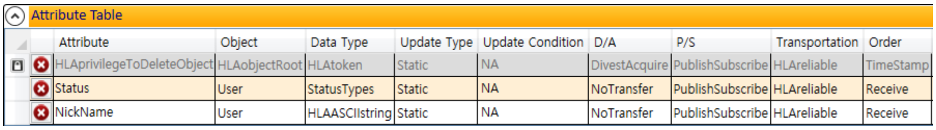
Figure 5-6 Object class structure table in SimGe



Interaction Class Structure Table

Class Name	Parent Name	P/S	Transportation	Order
HLAinteractionRoot		Neither	HLAreliable	TimeStamp
Chat		PublishSubscribe	HLAreliable	TimeStamp

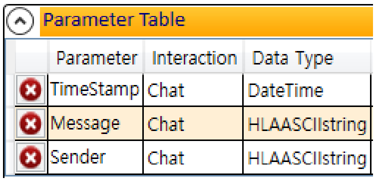
Figure 5-7 Interaction class structure table in SimGe



Attribute Table

Attribute	Object	Data Type	Update Type	Update Condition	D/A	P/S	Transportation	Order
HLAprivilegeToDeleteObject	HLAobjectRoot	HLAtoken	Static	NA	DivestAcquire	PublishSubscribe	HLAreliable	TimeStamp
Status	User	StatusTypes	Static	NA	NoTransfer	PublishSubscribe	HLAreliable	Receive
NickName	User	HLAASCIIstring	Static	NA	NoTransfer	PublishSubscribe	HLAreliable	Receive

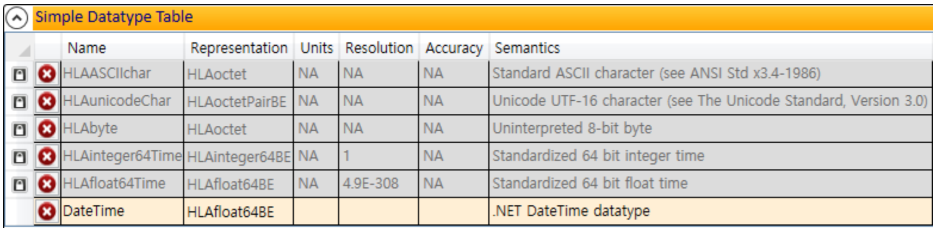
Figure 5-8 Attribute table in SimGe



Parameter Table

Parameter	Interaction	Data Type
TimeStamp	Chat	DateTime
Message	Chat	HLAASCIIstring
Sender	Chat	HLAASCIIstring

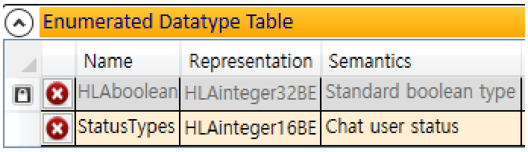
Figure 5-9 Parameter table in SimGe



Simple Datatype Table

Name	Representation	Units	Resolution	Accuracy	Semantics
HLAASCIIchar	HLAoctet	NA	NA	NA	Standard ASCII character (see ANSI Std x3.4-1986)
HLAunicodeChar	HLAoctetPairBE	NA	NA	NA	Unicode UTF-16 character (see The Unicode Standard, Version 3.0)
HLAbyte	HLAoctet	NA	NA	NA	Uninterpreted 8-bit byte
HLAinteger64Time	HLAinteger64BE	NA	1	NA	Standardized 64 bit integer time
HLAfloat64Time	HLAfloat64BE	NA	4.9E-308	NA	Standardized 64 bit float time
DateTime	HLAfloat64BE				.NET DateTime datatype

Figure 5-10 Datatype table in SimGe



Enumerated Datatype Table

Name	Representation	Semantics
HLAboolean	HLAinteger32BE	Standard boolean type
StatusTypes	HLAinteger16BE	Chat user status

Figure 5-11 Enumerated datatype table in SimGe

At the same time, the FED (HLA 1.3) or FDD (HLA 1516-2010) file should be exported by SimGe.

5.4.2. The class structure

The chatting federate application static structure is depicted in Figure 5-12. The federate SOM class (federateSom) and related interaction (CChatIC) and object classes (CUserOC) are generated using SimGe. Moreover, the federate class (CChatFd) is also generated as a partial class to enable the user to customize the generated code. SimGe also can automatically generate the code for the object model and federate callbacks.

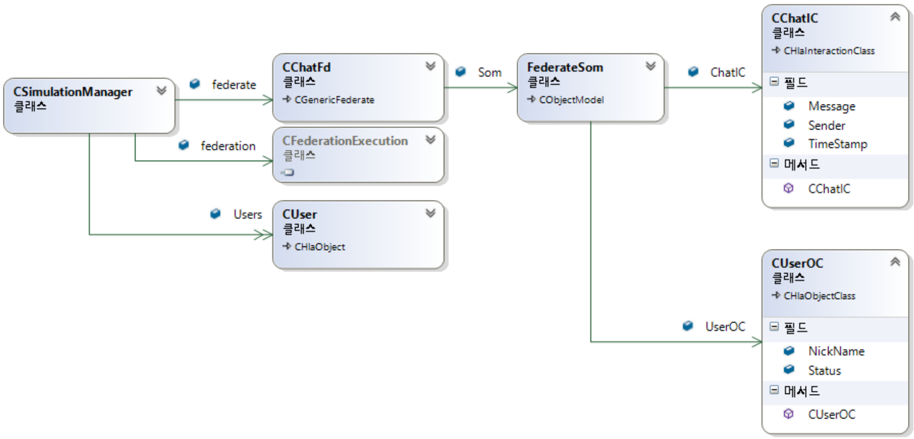


Figure 5-12 Class diagram of chatting federate application

5.4.3. Implementation

(1) Federation management

As shown in Table A-1, federation management includes the necessary services such as creating, joining, resigning, and destroying a federation execution. In this case, the data and information in a FOM that are used by the Create Federation Execution service and successive Join Federation Execution service invocations to configure the federation execution. When a federate leaves a federation, the Resign Federation Execution service terminates a federate’s participation in this federation. If the last federate needs to leave the federation, it can use the Destroy Federation Execution service to terminate an executing federation.

(2) Object management

Object management services are presented in Table A-3, after successful initialization of the federation execution, each objet need to register by using Register Object Instance service. The updates of the values of the HLA object attributes are sent to other federates when a value of an attribute is changed. To inform the change, Update Attribute Values service is defined by updating the name and status for the user. In order to send an interaction which are sender, message and timestamp in the chatting federation, Send Interaction service is used.

(3) Federate ambassador events

Federate ambassador events represent the RTI-initiated service callbacks such as Discover Object instance service and Remove Object instance. And all the federate ambassador event handlers begin with FbAmb_ prefix in RACoN API. In order to receive interactions from other chat client, Receive Interaction service is used to receive nickname, message and date time. Similarly, Reflect Attribute Values is used as interaction reception which includes nickname and status. When a request is received for object update, someone waits from us to provide updates for a specific object we owned. First, we find the object and its attributes for which update is requested, and then using Provide Attribute Value Update service to provide the updates.

5.4.4. Portico RTI initialization data

To connect chat client in different computers, related configuration should be set by using RTI.rid file. This file contains all the configuration options for Portico. By the way, the Portico RTI.rid file can not be used in DMSO RTI environment. On the section 4.5 in RTI.rid file which is JGroups Bind Address should be set to the same address. And all the federates should use the same file that has changed.

5.5. Framework of collaborative simulation

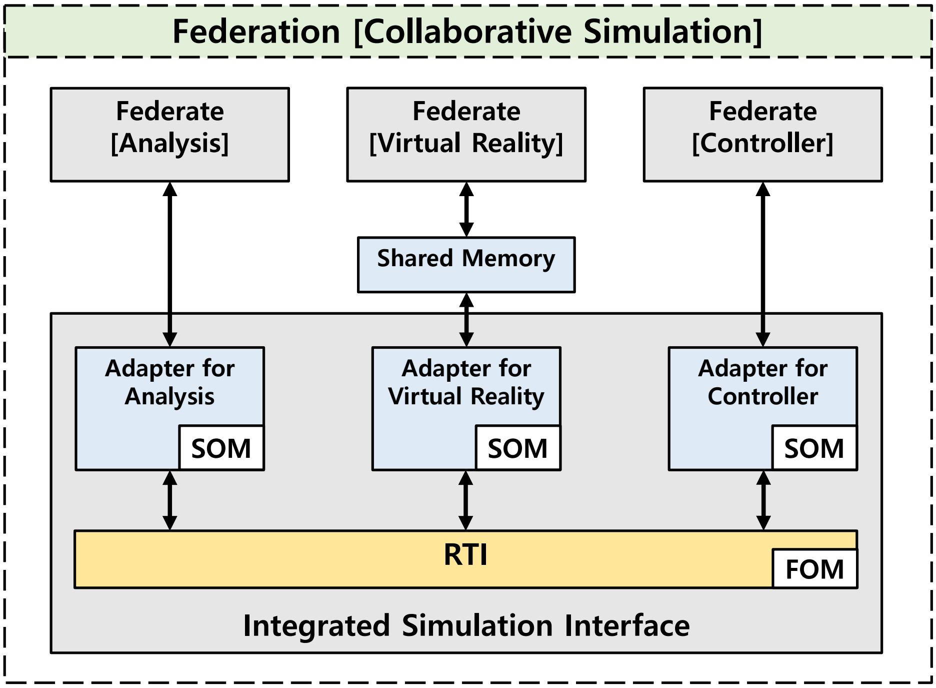


Figure 5-13 Framework for collaborative simulation

Figure 5-13 shows the framework for collaborative simulation in this study. The collaborative simulation can be defined as HLA federation, and the three HLA federates are analysis, VR and controller. The integrated simulation interface is developed based on HLA to effectively integrate all federates together. Every federate has its own adapter and it is developed based on RTI abstraction component for .NET (RACoN). As mentioned before, RACoN is developed using .NET 4.0 runtime while the Unity3D works on Mono runtime which equivalent to .NET 3.5, therefore both of them can not connect directly. To solve this problem, a shared memory dynamic link library is introduced between VR federate and VR adapter, which helps them to share memory. When the data is written into the memory, VR will read them automatically. As each federate has their adapter, each adapter as well has an own SOM which can be generated by SimGe and it includes the HLA object classes, interaction class and routing spaces. Similarly, the FOM describe the shared object, attribute and interaction for the whole federation and it passed to the RTI by means of a FED/FDD file and it can be generated by SimGe. In this chapter, it will focus on the details of the intergrated simulation interface.

5.6. Integrated method of analysis, VR and controller

5.6.1. FOM design for block lifting example

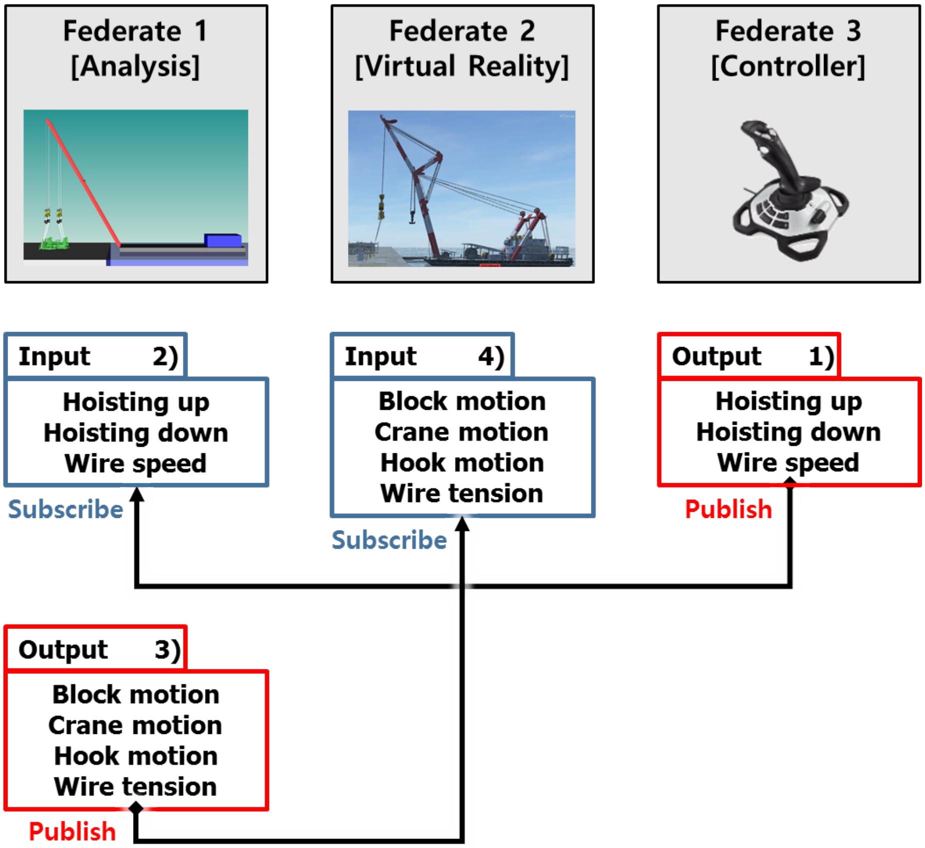


Figure 5-14 FOM design for block lifting example by using one floating crane

FOM is designed as shown in Figure 5-14, and an example of block lifting by using floating crane will be taken to introduce the FOM in detail. FOM is made up with three federates: Analysis, VR and controller and subscription is introduced to improve the efficiency of data delivery. Since what controller federate publish is needed by analysis federate and what analysis federate publish is needed by VR federate, analysis and VR federates subscribe controller outputs and analysis outputs respectively. Then when controller publish new outputs, the data will only be delivered to federate 1 instead of federate 2. Similarly, when analysis publish new outputs, federate 2 will be the only receiver. In this way, data delivery among federates becomes more efficient and faster.

5.6.2. Adapter design for block lifting example

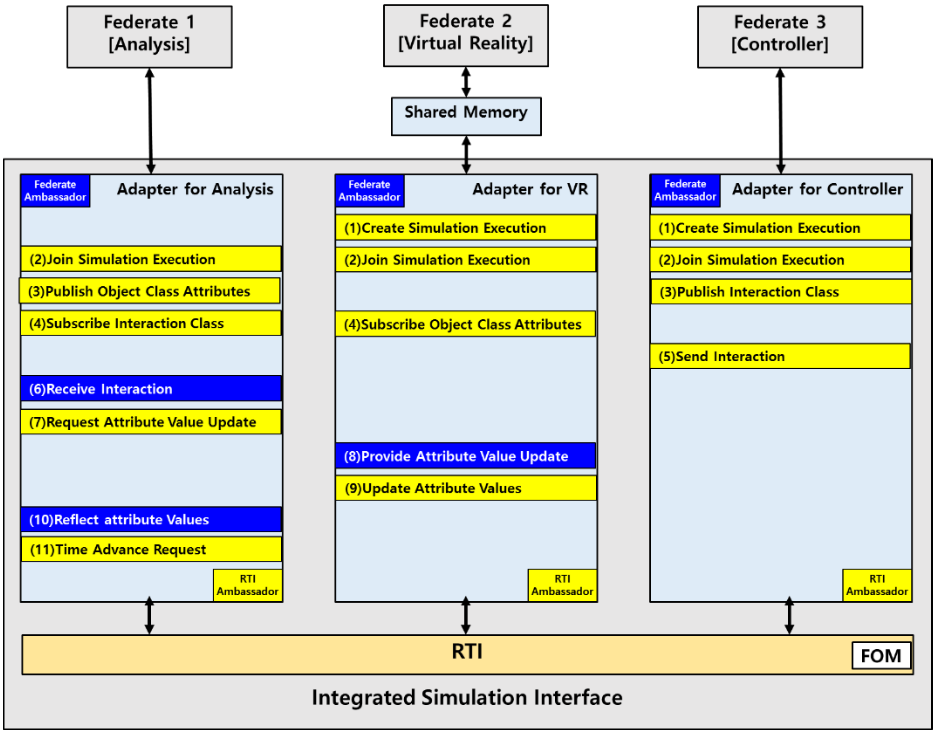


Figure 5-15 Adapter design for block lifting example by using one floating crane

Figure 5-15 shows how adapter is developed. In the case of block lifting by using one floating crane, federate 3 creates simulation and then all the federates join in and start publishing their interaction class, object class and also subscribe the data they require. When controller send interactions, for example hoisting up/down in this case, this interaction will be delivered to the analysis federate who subscribe its outputs. Then analysis starts, requests attribute value update, and provides attribute value update to VR. When value is updated in VR, analysis federate will receive a response and time advance request.

5.6.3. Data transform procedure by using integrated simulation interface

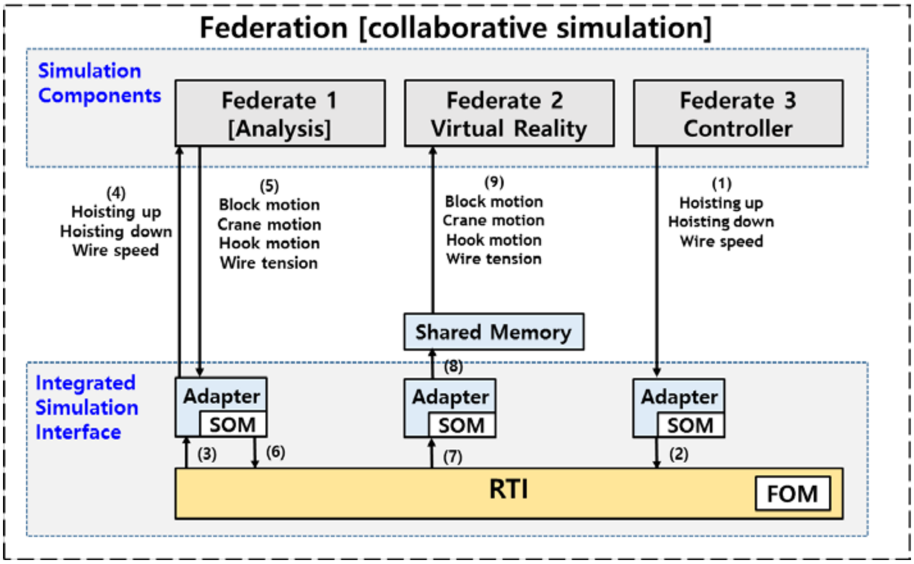


Figure 5-16 Data transform procedure by using integrated simulation interface

Figure 5-16 shows how inputs and outputs of federates exchange date via integrated simulation interface. Firstly, Signals generated by controller are delivered to RTI through adapters (1, 2). With FOM, RTI is able to recognize that the signals are to be sent to federate 1 and deliver the signals through adapters of federate 1 (3, 4). And results of analysis federate will be conveyed to RTI (5, 6). FOM is able to recognize that federate 2 has subscribed analysis outputs and then write the data into memory of the computer which accommodates federate 2 (7, 8). When shared memory notice the entrance of data, the data will be automatically delivered to VR (9).

6. Application examples of “collaborative simulation in shipbuilding and offshore installation”

In this study, collaborative simulation can be summarized into four parts.

Firstly, in order to describe the crane and the block as real as possible, and make operators feel like working in the actual operation site, VR technology that could improve the sense of reality and immersion is studied. This study applies Unity3D to develop the collaborative simulation scenes, and immersive interface devices such as HMD to present three-dimensional scenes.

Secondly, because the cranes and block in virtual environment should move like a real movement, research on physics analysis technology based on multibody system dynamic has been done. By using physics-based analysis program, which has been developed for five years in-house, the motion analysis results can be presented in real-time. Through the integrated simulation interface, the analysis results can be synchronized with VR.

Thirdly, to control the cranes in the simulation, workers who operate the crane need a controller. Therefore, a scenario generator is developed which can convert the signals from the controllers to the input datum for VR and analysis. In addition, a part of the HMD, positional tracking, can track the real head movements of the operators to assist the creation of stunning visual experience.

Finally, in order to effectively integrate the VR, analysis and controller as well as consider of interoperability and reuse, this study developed an integrated simulation interface based on the HLA.

By these four techniques, this study utilizes developed collaborative simulation, which can be applied in simulating block turn-over operation and topside module installation. Each simulation allows four operators to operate under the same virtual environment simultaneously to implement collaborative operation.

6.1. Application to “Block turn-over operation”

6.1.1. Introduction to block turn-over operation

In shipyard, a ship is constructed by joining its parts on the dock after dividing it into a number of blocks and assembling them. In some cases of block production, blocks should be constructed under the turn-over condition because it is easier to assemble the blocks under the turn-over condition than under the normal condition. Thus, many blocks are assembled in the turn-over condition and then get reversed during the block erection. This is called a turn-over process [1].

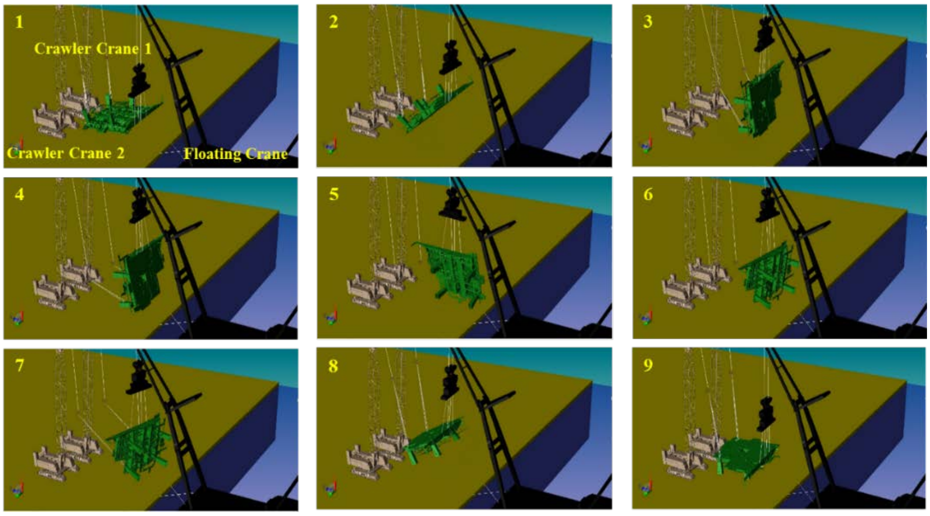


Figure 6-1 Block turn-over operation process

Figure 6-1 shows the ship block turn-over operation conducted in order to erect a block weighing about 350 tons using one floating crane and two crawler cranes. In this case, the floating crane is a piece of equipment that can pull up heavy blocks on seawater by buoyancy with a maximum weight of 1800 tons, and two crawler cranes can lift as heavy as 900 tons. The main dimensions of block and cranes are as shown in Table 6-1. During the turn-over process, the block is lifted vertically in stage 1, 2 and 3, and then get rotated 180 degrees in stage 4, 5 and 6, and finally, pulled down horizontally in stage 7,8, and 9. The detailed process are described in Table 6-2.

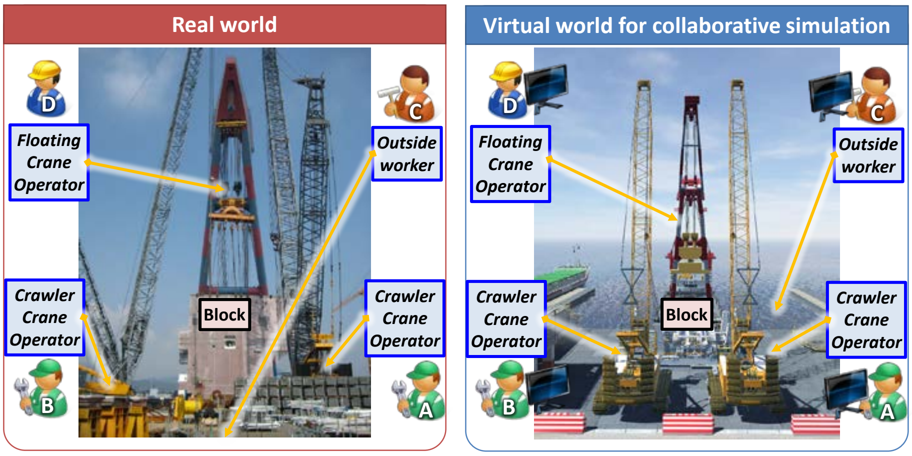
Table 6-1 Main Dimensions of floating crane, crawler crane and block

	Floating crane	**Crawler crane**	**Block**
Capacity or Weight	1,800 ton (C)	900 ton (C)	350 ton (W)
Main Dimensions	62m X 27m X 71m	21m X 17m X 65m	34m X 27m X 11m

Table 6-2 Operation procedure of block turn-over operation

Stage	**Crawler Crane 1**	**Crawler Crane 1**	**Floating Crane**
1	Hoist up	Hoist up	Hoist up
2	Stop	Stop	Hoist up
3	Hoist down	Hoist down	Hoist up
4	Hoist down	Hoist down	Stop
5	Disconnection	Disconnection	180deg Rotation
6	Stop	Stop	Stop
7	Connection	Connection	Stop
8	Hoist up	Hoist up	Hoist down
9	Hoist down	Hoist down	Hoist down

6.1.2. Collaborative simulation for block turn-over operation



Totally four workers should be included in the same virtual world.

Figure 6-2 Collaborative simulation for block turn-over operation

Figure 6-2 shows the collaborative simulation for block turn-over operation in real world and virtual world.

In real world, to complete block turn-over operation, a minimum of four operators is required, who are one floating crane operator, two crawler crane operators and one outside worker. When operators operate their crane, they cannot see the whole scene because the block is so large and the all operators are operating in the crane inside. Thus, outsider workers are necessarily demanded to observe the whole situation and instruct the whole operation via interphones. Usually, more than one outsider work participates in the operation in actual practice, but this study only considers operations with one outsider worker for simplicity.

In the virtual world for collaboration, in order to closely mimic the operation scene, the aforementioned four technologies are well applied. By using VR, the virtual operation scene can be created in a way close to the real world. Through the physics analysis, the cranes, wires, hooks and blocks can move like a real movement. Further mover, the analysis results can be used for the operation evaluation. The controllers and scenario generators can make the operation more immersive, and the integrated simulation interface ensures that all simulation components integrate together effectively.

6.1.3. Simulation components for block turn-over operation

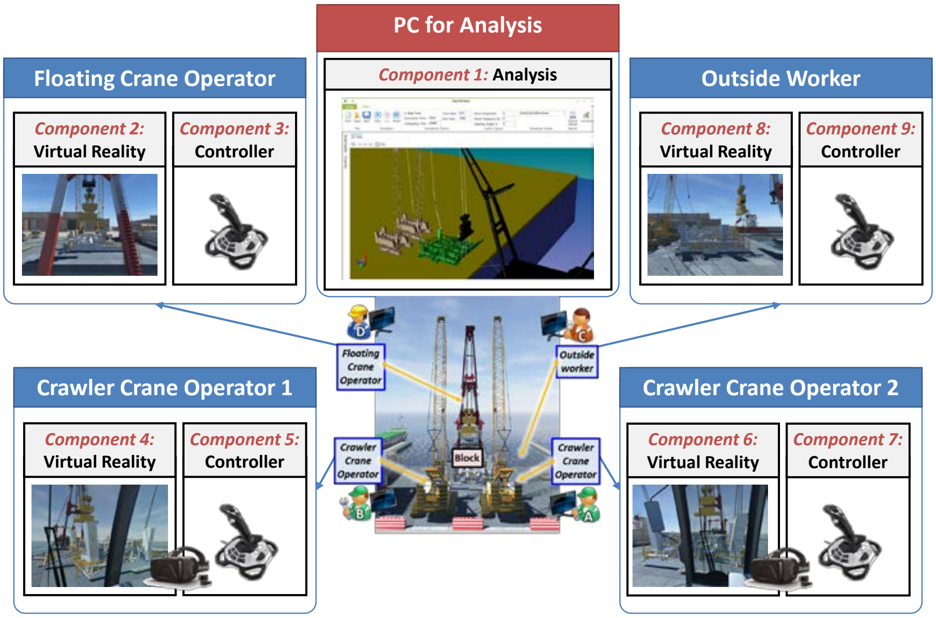
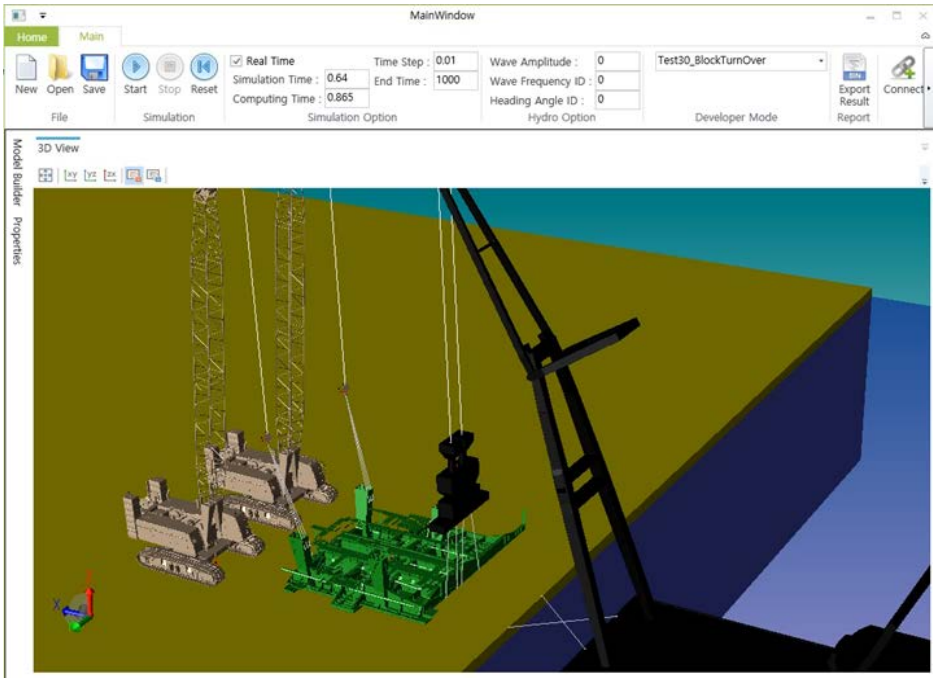


Figure 6-3 Simulation components for block turn-over operation

In this application, there are four operators, each operator has two simulation components, which are VR scene and controller in a computer, and all the simulation components share one analysis component. The analysis component can run in four operators’ computer or an independent computer. In this case, because two crawler crane operators are very close to the block, if the view of them is fixed, the scene they see will be very limited and it is inconvenient for the operator during the operation. Therefore, the HMD is utilized so that the positional tracking can make a trace of the operators' real head movements, and the movement can be converted to signal which allows operators to change their views to see the situation.

6.1.4. Modeling result for analysis



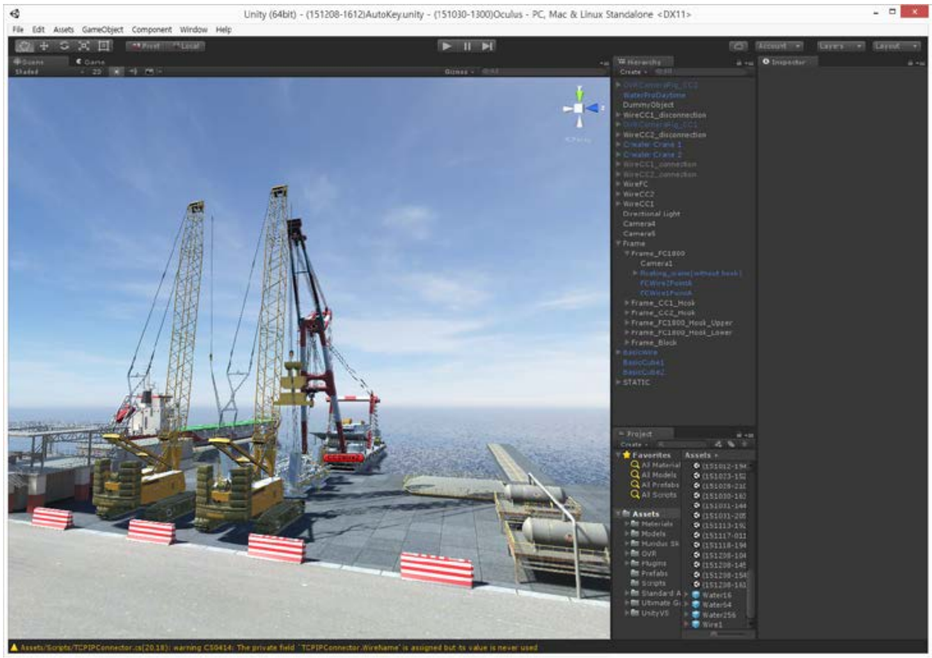
☑ Real Time
Simulation Time : 0.64
Computing Time : 0.865
Time Step : 0.01
End Time : 1000
Wave Amplitude : 0
Wave Frequency ID : 0
Heading Angle ID : 0
Test30_BlockTurnOver

Figure 6-4 Block turn-over operation modeling result for analysis

Figure 6-4 shows the modeling result of block turn-over operation. Only models participating in analysis and calculation are involved. The brown box represents the land, which supports collision detection with the objects on the land while the blue box stands for the seawater, where external forces, such as hydrostatic and hydrodynamic forces, have an effect. As the figure shows, two 900-ton cranes are settled on the land while the 1800-ton floating crane is tied by moorings, floating on the sea. Besides, the blocks are connected with three cranes by hook-attached wires so that they can be hoisted up and down.

To specify the simulation process. Before simulation, the mass and mass moment of inertia of floating crane, crawler cranes, block, hooks and wire rope data should be determined, and the input data from controller will be processed in the scenario management kernel. Then when the simulation officially begins, the hydrostatic force, the hydrodynamic force, the wire rope force, the mooring force, and gravity enter into the equation of motion, be calculated and then be reflected. In addition, such results will also be transferred to other simulation components in real time.

6.1.5. Modeling result for VR



```
Assets/Scripts/TCPIPConnector.cs(20,18): warning CS0414: The private field `TCPIPConnector.WireName' is assigned but its value is never used
```

Figure 6-5 Block turn-over operation modeling result for virtual reality

Figure 6-5 shows the modeling result of block turn-over operation by Unity3D. Unlike analysis program, not only models participating in analysis and calculation are enrolled, visual effects has been added in order to make the operation scene as immersive as possible. For example, effects of seawater, land, sky, sunshine, and shadows are settled. Moreover, in order to make the scene like a shipyard, inner wall, barriers and other equipments has been imported. The floating crane, crawler cranes, blocks and hooks employed in the VR is of the same scale with those in analysis programs, in which such components are simplified for analytical convenience. That is to say, the center point, initial position, rotation and other data of all the models in VR maintains the same with that in the analysis program, but only touch-up and texture mapping are applied for the immersion purpose.

The seawater effect in VR scene is provided by the built-in water resource in Unity3D and it is notable that such effects are independent of the sea force that acts on floating crane and is analyzed in the analysis program.

The wire ropes used in this study are attained from the Ultimate Rope Editor in Unity3D asset store. Each rope has its start and end points; however, information regarding this can not be exported by analysis program. To tackle this problem, barely visible game objects are set at the connection points between wires and hooks, cranes and blocks so that the start and end points of each rope can be attained. These game objects are installed in the directories of hooks, cranes and blocks respectively and when the root directory changes, the game objects changes accordingly. For example, if changes in hook’s position and rotation occur, game objects (equals to the start point and end point of the wire rope) in the directory of the certain hook will change in accordance so that the wire can be guaranteed to be connected with hooks, cranes and blocks all the time.

Besides, four cameras corresponding to respective views of four operators are located at four seats and four exe files will be exported for operators’ usage.

6.1.6. Inputs and outputs of simulation components

Table 6-3 Inputs and outputs of simulation components / federates

Operator/PC	Simulation Components / Federates	Input	Output
PC for analysis	Federate 1 Analysis	Hoisting up Hoisting down Wire speed Connection Disconnection Rotation	Block motion Crane motion Hook motion Wire tension
Floating Crane Operator	Federate 2 VR 1 (Display)	Hoisting up Hoisting down Wire speed	
	Federate 3 Controller 1 (Joystick)		Hoisting up(J) Hoisting down(J) Wire speed(J)
Crawler Crane Operator 1	Federate 4 VR 2 (Display / HMD)	Block motion Crane motion Hook motion Wire tension View Point	
	Federate 5 Controller 2 (Joystick + Positional Tracking)		Hoisting up(J) Hoisting down(J) Wire speed(J) View Point(PT)
Crawler Crane Operator 2	Federate 6 VR 3 (Display / HMD)	Block motion Crane motion Hook motion Wire tension View Point	
	Federate 7 Controller 3 (Joystick + Positional Tracking)		Hoisting up(J) Hoisting down(J) Wire speed(J) View Point(PT)
Outside Worker	Federate 8 VR 4 (Display)	Block motion Crane motion Hook motion Wire tension	
	Federate 9 Controller 4 (Joystick)		Connection(J) Disconnection(J) Rotation(J)

As chapter 7.1.3 shows, the simulation involves nine components and the federation consists of nine federates. The inputs and outputs among federates are demonstrated in the Table 6-3, and how data transform among these federates will be introduced in the following chapter.

6.1.7. Data transform procedure by using integrated simulation interface

Figure 6-6 and Figure 6-7 demonstrate how inputs and outputs of federates exchange date via integrated simulation interface.

Figure 6-6 indicates as follows.

(1) Controllers of floating crane (federate 3), crawler crane 1 (federate 5), crawler crane 2 (federate 7) send out signals to hoist up.

(2) The signals are delivered to RTI by respective adapter.

(3) With FOM, RTI is able to recognize that the data has been subscribed by federate 1, thus deliver data to analysis (federate 1).

(4) When the data goes through the adapters during analysis, scenario generator will transform the data into scenarios that can be understood, such as changing floating crane wire speed to 1m/min.

(5) The results include block motion, floating crane motion, crawler crane 1 motion, crawler crane 1 motion, floating crane hook motion, crawler crane 1 hook motion, crawler crane 1 hook motion and wire tension.

(6) With FOM, RTI is able to recognize that the data has been subscribed by federate 2, 4, 6 and 8.

(7) Then data will be delivered to computers which embodies federate 2, 4, 6, 8 respectively through each corresponding federates.

(8) Data from RTI will be written into memories of the computers that accommodate federate 2, 4, 6, and 8.

(9) When the shared memory notices the entrance of data, it automatically conveys the data to federate 2, 4, 6 and 8 and the movements of each objects will be reflected on each screen.

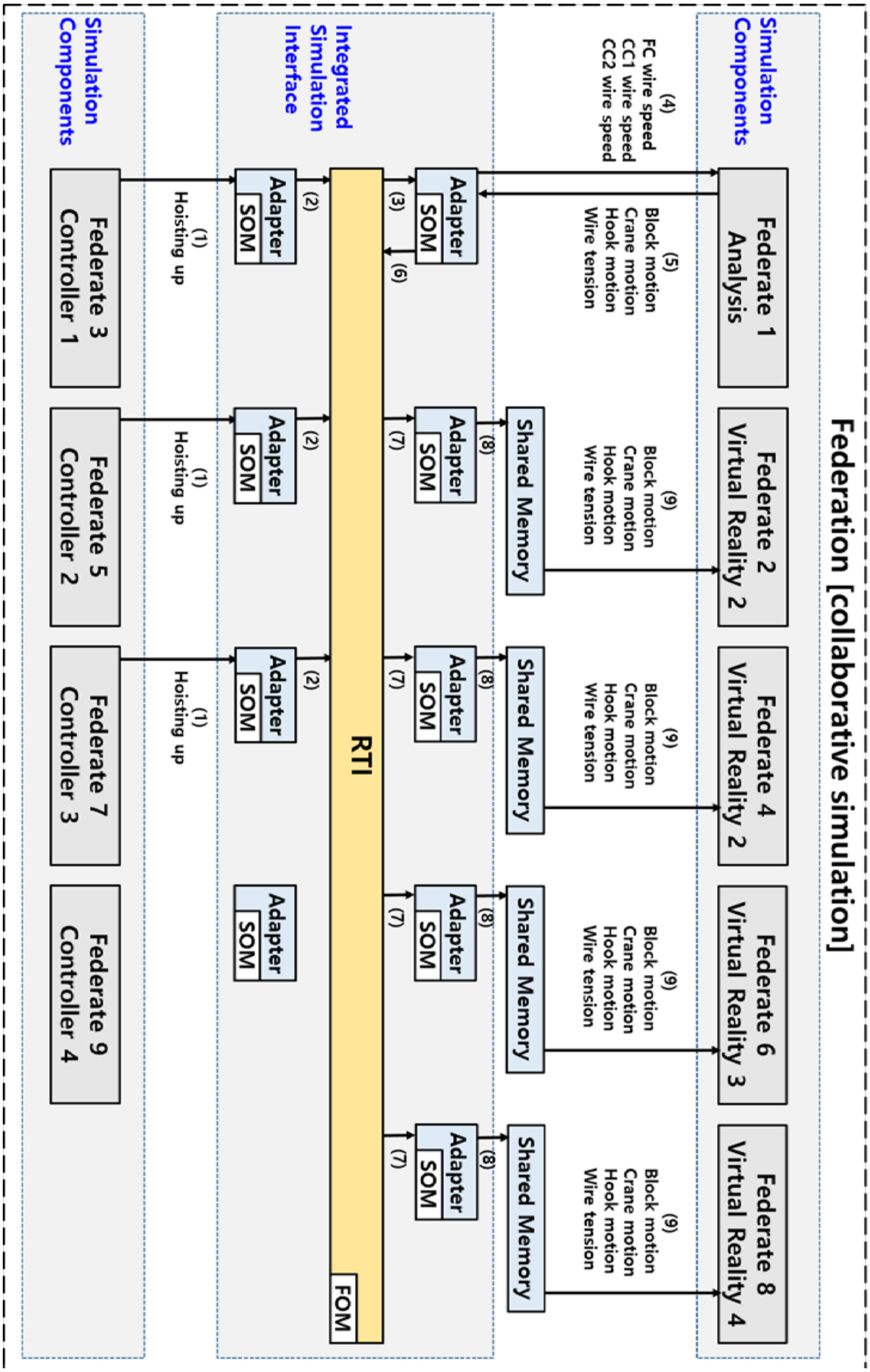


Figure 6-6 Data transform procedure by using integrated simulation interface

Figure 6-7 indicates that,

(1) When federate 5 and federate 6 change the view.

(2) Data will be conveyed to RTI through adapters

(3) With FOM, RTI is able to recognize that the destination of the date is federate 4 and 6.

(4) Data will be written into the memories of computer which accommodates federate 4 and 6.

(5) When the shared memory notice the entrance of data, it automatically convey the data to federate 4 and 6.

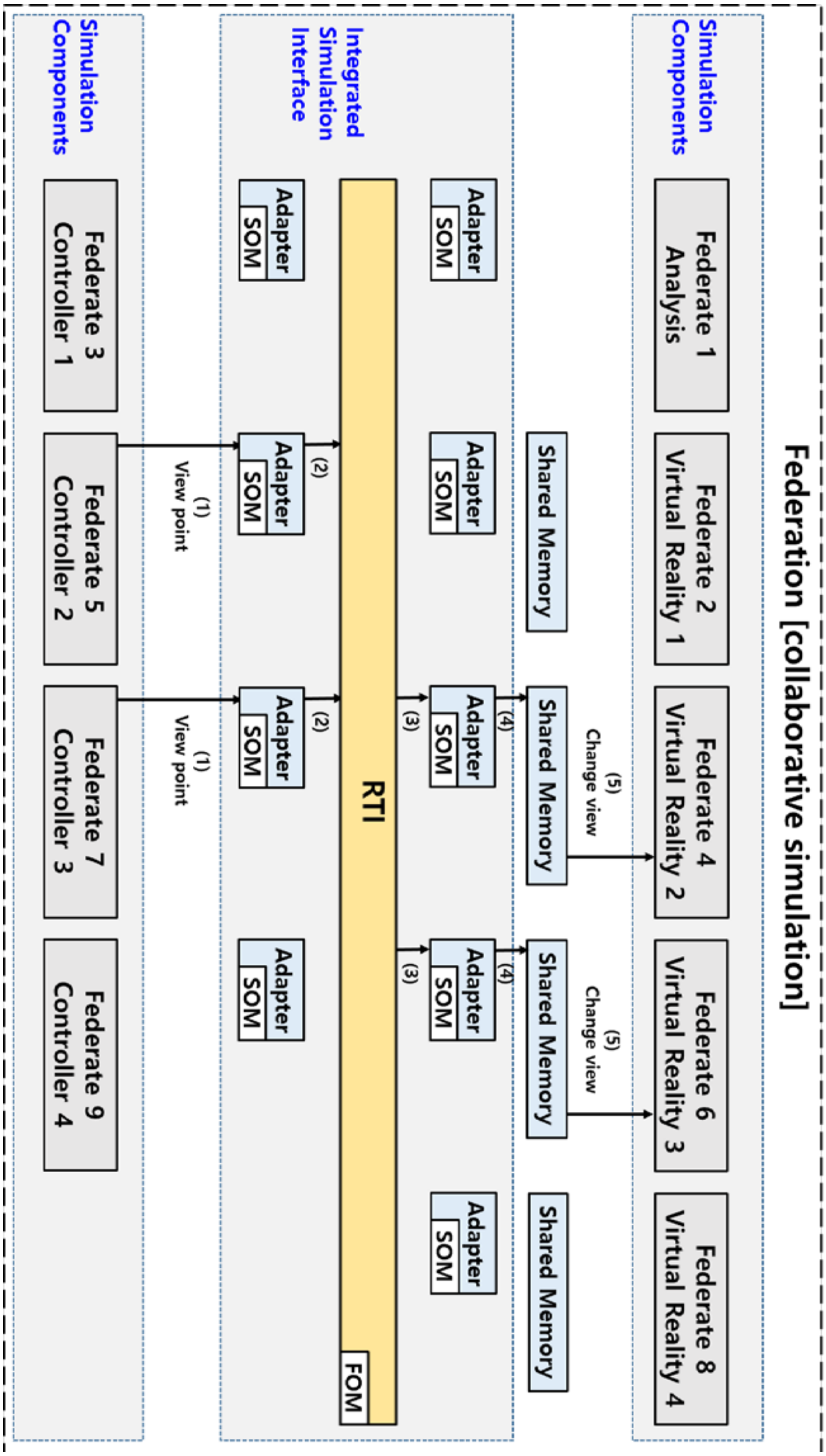


Figure 6-7 Data transform procedure by using integrated simulation interface

6.1.8. Prototype simulator based on the integrated simulation interface

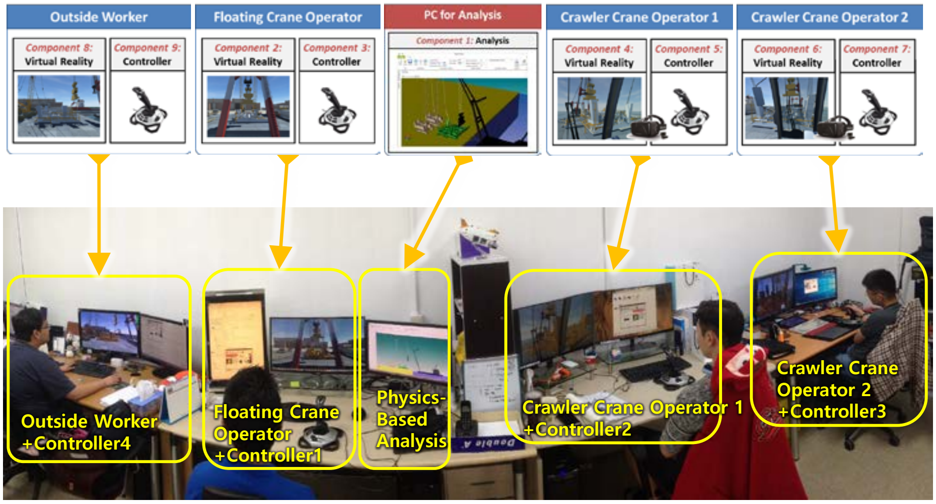


Figure 6-8 Prototype simulator based on the integrated simulation interface

Figure 6-8 shows how the simulation is conducted. There are four workers with four computers. From left to right, VR and controller of outside worker are in the first computer, VR and controller of floating crane as well as analysis are in the second computer, and the VR and controller of two crawler cranes are in the next two computers. With integrated simulation interface, four workers are able to accomplish block turn-over operation in simulation.

6.1.9. Collaborative simulation results of the prototype simulator

Figures from 7-9 to 7-16 demonstrate four operators' views during simulation.

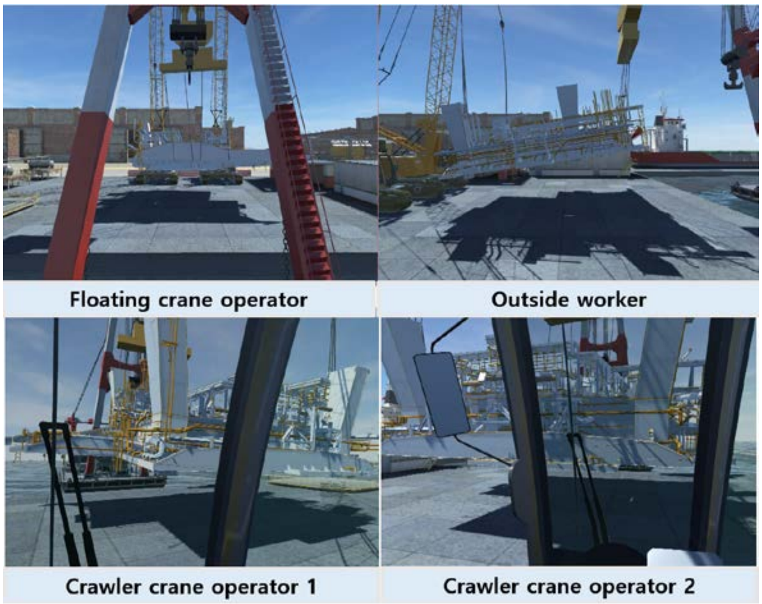


Figure 6-9 Four operators' views during simulation (1/8)

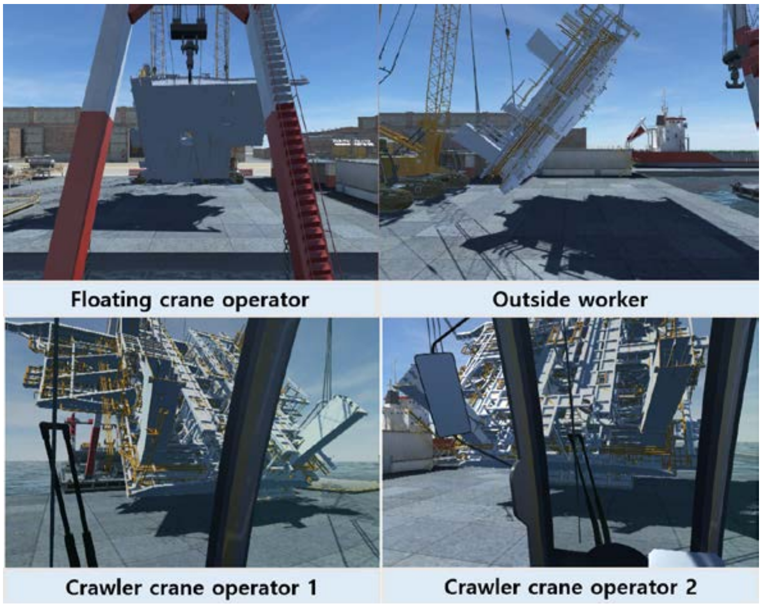


Figure 6-10 Four operators' views during simulation (2/8)

Figure 6-11 Four operators’ views during simulation (3/8)

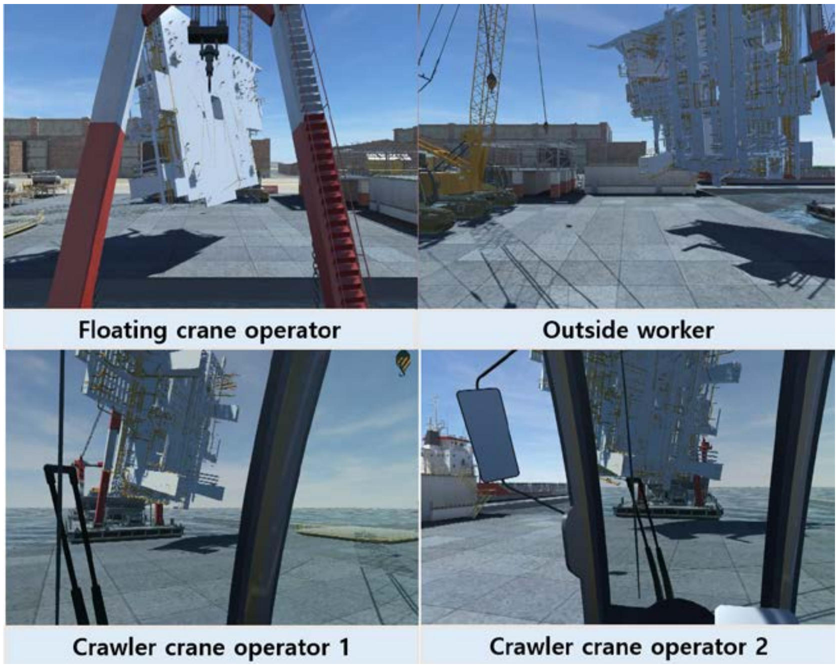


Figure 6-12 Four operators’ views during simulation (4/8)

Figure 6-13 Four operators' views during simulation (5/8)

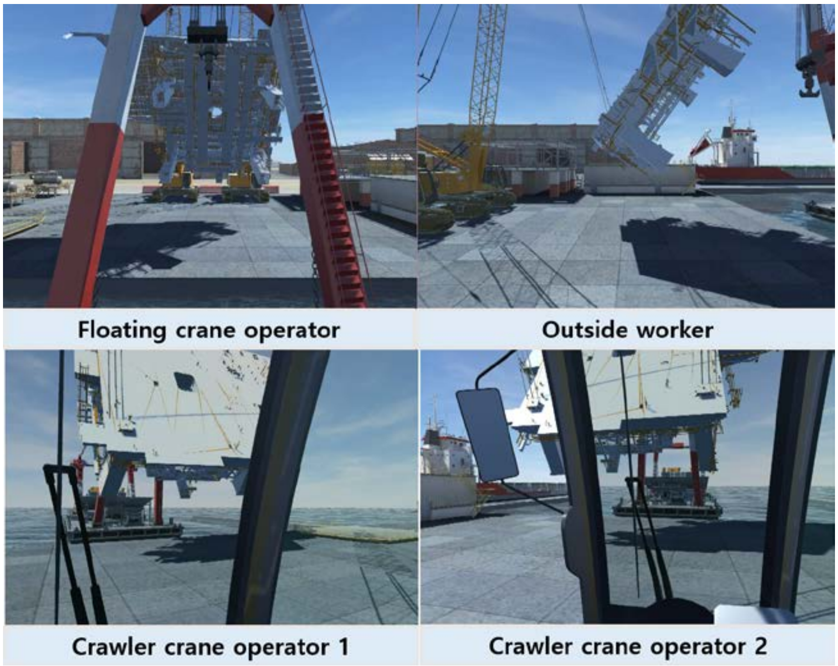


Figure 6-14 Four operators' views during simulation (6/8)

Figure 6-15 Four operators’ views during simulation (7/8)

Figure 6-16 Four operators’ views during simulation (8/8)

6.1.10. Discussion on collaborative simulation results (wire tension)

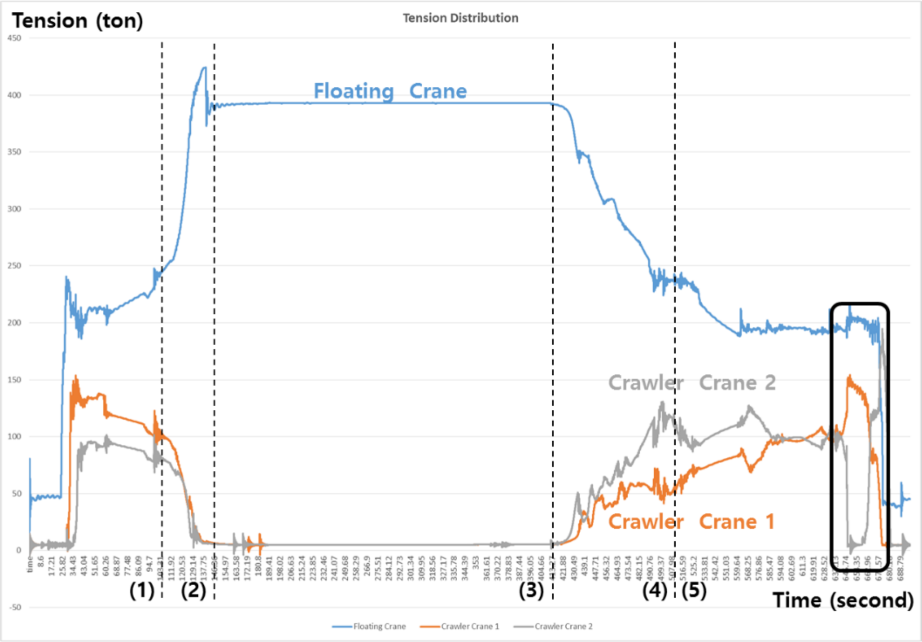


Figure 6-17 Tension distribution of collaborative simulation for block turn-over operation

Figure 6-17 shows the tension distribution of wires of floating crane, crawler crane 1, crawler crane 2 as time goes by.

(1) Stage (1) is the time for hoisting up

(2) Stage (2) is the time when floating crane is being hoisted up, and crawler crane 1 and 2 staying still.

(3) Stage (3) is for block rotation

(4) Stage (4) is the time when floating crane are being hoisted down, and crawler cranes are being hoisted up.

(5) Stage (5) is the time when all the cranes are being hoisted down.

In the last stage, the tension of crawler crane 2 drops dramatically. This is because the worker operating the crawler crane 2 hoists down the wire faster than the worker operating crawler crane 1 does while the block is going down.

The situation in detail is showed in Figure 6-18. Due to the operater’s mistake, the wire connected to crawler crane 2 is almost loose without stress while the wire connected to the crane 1 will be overloaded. Such dangerous mistake can be detected by the collaborative simulation developed by this current study and be prohibited in advance so that the safty of the real operation will be ensured.

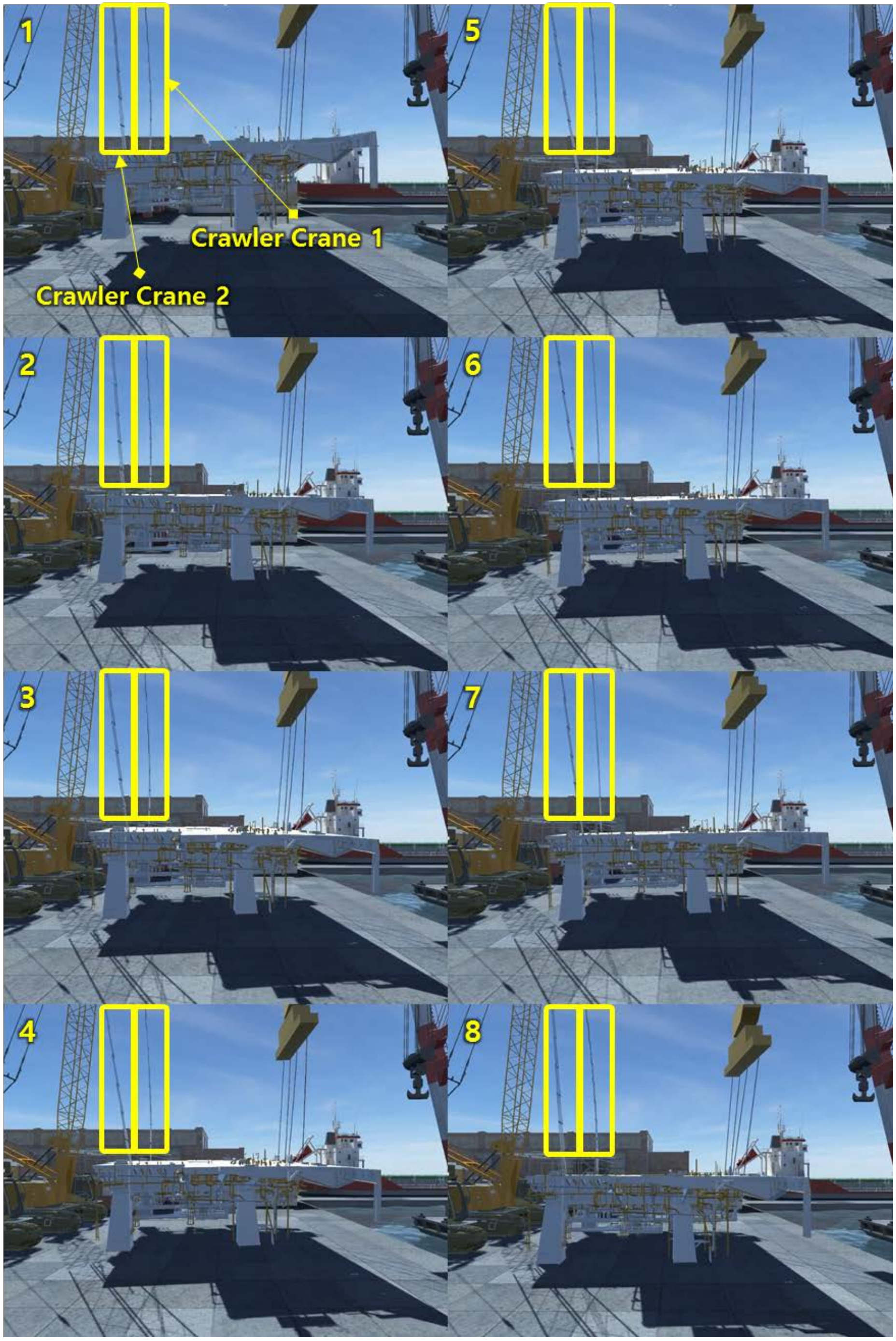


Figure 6-18 Mistake operation for collabrotive simulation for block turn-over operation

6.2. Application to “Topside module installation”

6.2.1. Introduction to topside module installation

To make offshore structure installment easy, the offshore is usually divided into several modules, and topside modules weighing from 3000 to over 20000 tons have been installed on offshore structure [2]. Thus, during the installation process, large offshore cranes and barge ship need to work together.

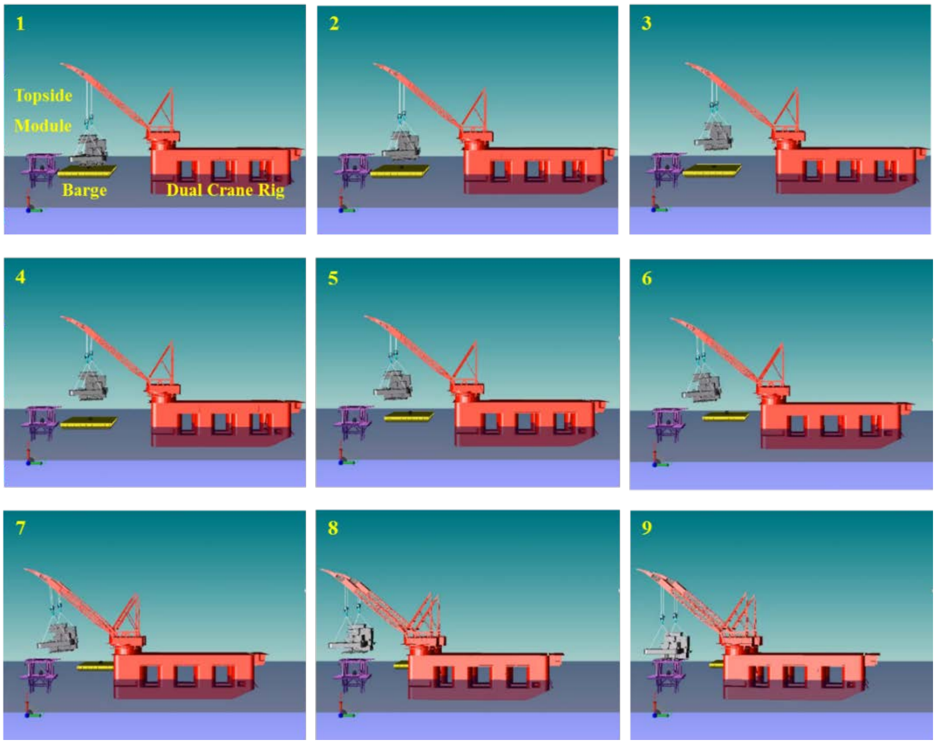


Figure 6-19 Topside module installation process

Figure 7-19 demonstrates how topside module is installed in usual cases. In this case, a dual crane rig weighing 14200 tons are used to install a topside module weighing 5000 tons. The dual crane rig allows two cranes to operate simultaneously. The main dimensions of dual crane rig and topside module are as shown in Table

6-4.

From stage 1 to stage 3, topside module is lifted from the barge ship until it reaches certain height and then the barge ship leaves. From stage 4 to stage 7 is the time when dual crane rig is moving forward. When the topside module reaches certain height, two cranes on the dual crane rig will start going down. Finally, the topside module will be installed on the offshore structure in stage 8 and 9.

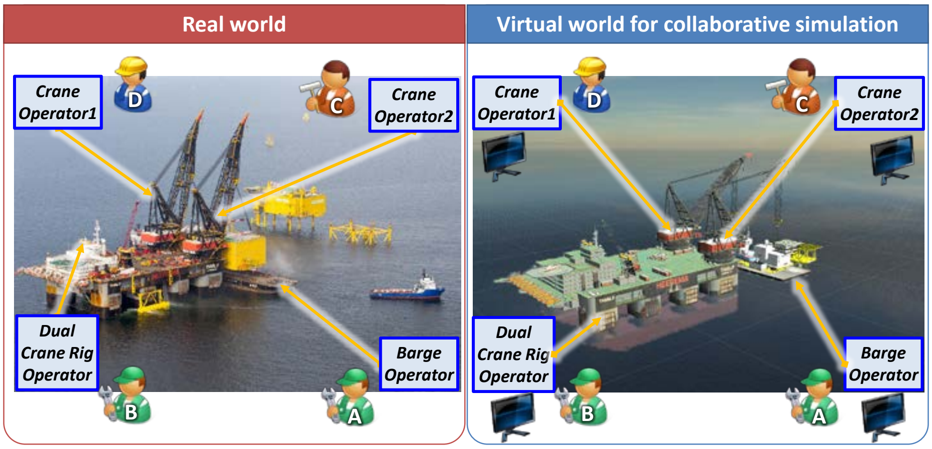
Table 6-4 Main dimensions of dual crane rig and topside module

	Dual crane rig (DCR)	**Topside module**
Capacity or Weight	14,200 ton (C)	5,000 ton (W)
Main Dimensions	165m X 88m X 50m 11.8~31.6m (Draft)	83m X 32m X 39m

Table 6-5 Operation procedure of topside module installation

Stage	**Dual Crane Rig (DCR)**	**DCR Crane 1**	**DCR Crane 2**	**Barge**
1	Stop	Hoisting up	Hoisting up	Stop
2	Stop	Hoisting up	Hoisting up	Stop
3	Stop	Hoisting up	Hoisting up	Going forward
4	Stop	Hoisting up	Hoisting up	Going forward
5	Going forward	Stop	Stop	Going forward
6	Going forward	Stop	Stop	Going forward
7	Going forward	Stop	Stop	Stop
8	Stop	Hoisting down	Hoisting down	Stop
9	Stop	Hoisting down	Hoisting down	Stop

6.2.2. Collaborative simulation for block turn-over operation



Totally four workers should be included in the same virtual world.

Figure 6-20 Collaborative simulation for topside module installation

Figure 7-20 shows the collaborative simulation for topside module installation in real world and virtual world.

In the real world, at least four operators are required to complete topside module installation, who includes one dual crane rig operator, two DCR crane operators and one barge operator. Similar with block turn-over operation, topside module operation also requires outside worker to monitor the whole process because the view of the DCR crane operator is blocked by the topside module. Besides, the outside worker serves as the barge ship operator at the same time in this case.

The main purpose of this study is to improve the simulation to be as close to real operation as possible. To fulfill this, four techniques were utilized. More immersive visual effects are realized with the help of VR, and a variety of external forces are taken into account and reflection of the movements of objects including the dual crane rig, wires, topside module, barge, and hook are accomplished with physical analysis. In addition, controller and scenario generator improve the sense of reality when operators are operating the crane or barges. Furthermore, all the simulation components are effectively combined together via integrated simulation interface so that four operators are able to conduct topside module installation simultaneously in virtual world.

6.2.3. Simulation components for block turn-over operation

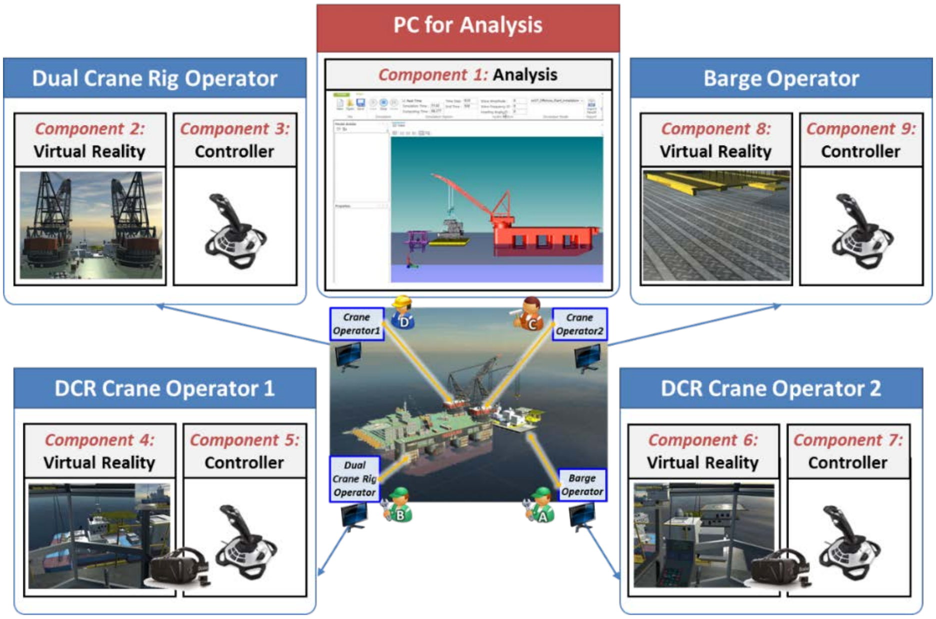
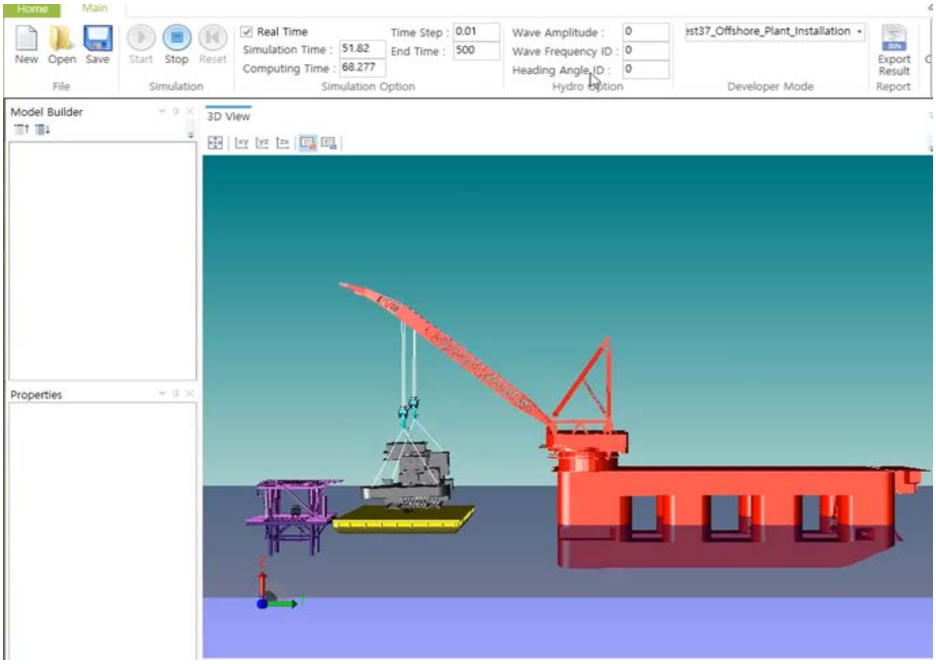


Figure 6-21 Simulation components for topside module installation

In this application, there are four operators, and each operator has two simulation components, which are VR scene and controller in a computer, and all the simulation components share one analysis component. The analysis component can run in four operators' computer or an independent one.

Since the views of crane operators are partially blocked by the topside module, barge operator need to be responsible for observing the whole situation and monitor other operators. On the other hand, crane operators can change their view angles while positional tracking on the HMD will track actual movements of each operator and generate certain signals.

6.2.4. Modeling result for analysis



☑ Real Time
Simulation Time : 51.82
Computing Time : 68.277
Time Step : 0.01
End Time : 500
Wave Amplitude : 0
Wave Frequency ID : 0
Heading Angle ID : 0
st37_Offshore_Plant_Installation

Figure 6-22 Topside module installation modeling result for analysis

Figure 7-22 shows the modeling result of topside module installation. All the visible models in the analysis programs are to be analyzed and calculated. The blue box represents seawater which provides buoyancy and other external forces form the sea, while the purple model stands for the offshore structure where topside module will be installed. The yellow one is the barge ship which is responsible for shipping topside modules and the red one stands for the dual crane rig, where an object with a maximum weight of 14200 ton can be lifted. Here in this study, the topside module weighs 5000 tons.

In order to fasten the analysis, most models have been simplified. Before simulation, the mass and mass moment of inertia of dual crane rig, topside module, barge ship, hooks and wire rope should be determined, and input data from controller will be processed in the scenario management kernel. When simulation starts, the hydrostatic force, the hydrodynamic force, the wire rope force, the mooring force, and gravity will be entered into the equation of motion, be calculated and then be reflected. The results will be transferred to other simulation component in real time.

6.2.5. Modeling result for VR

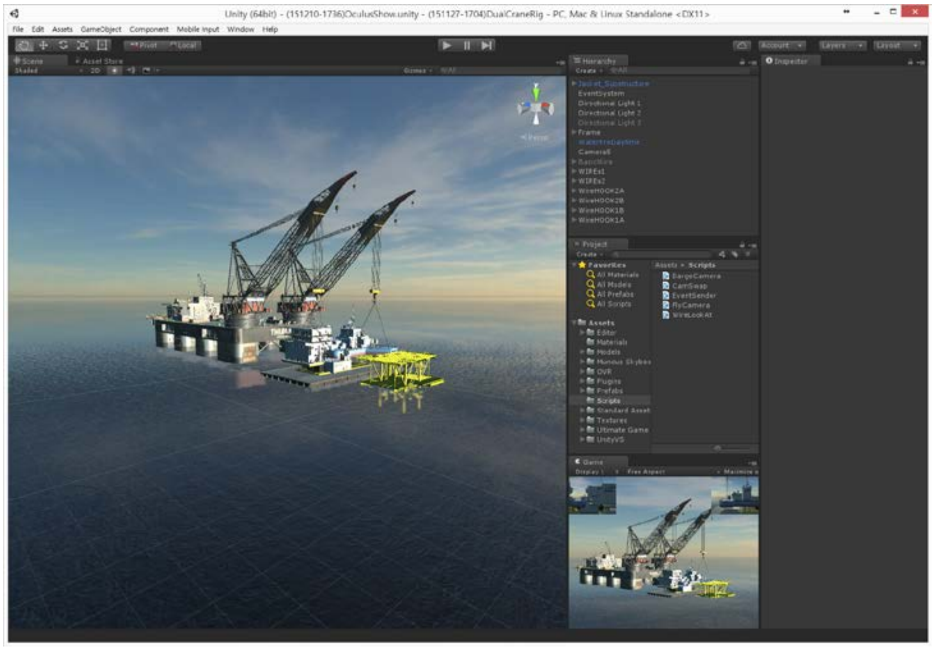


Figure 6-23 Topside module installation modeling result for virtual reality

Figure 7-23 shows the modeling result of topside module installation by Unity3D. Many special effects are added apart from those models in calculation, for instance, the seawater, sky, sunshine and shadow and so on. Additionally, contrasting to models analysis programs, those in VR are more immersive and closer to reality after texture mapping and touch-ups.

Like block turn-over operation, same initial position and rotation are applied in both analysis and VR, and all the models possess same scale and center. Therefore, all objects in both analysis programs and VR can keep the same movements. Similarly, the seawater in VR is only for visualization, independent of the seawater in analysis part.

Besides, barely visible game objects are set at the connection points between wires and hooks, cranes and blocks so that the start and end points of each rope can be attained. For example, if changes in hook’s position and rotation occur, game objects (equals to the start point and end point of the wire rope) in the directory of the certain hook will change in accordance so that the wire can be guaranteed to be connected with hooks, cranes and blocks all the time.

6.2.6. Inputs and outputs of simulation components

Table 6-6 Inputs and outputs of simulation components / federates

Operator / PC	**Simulation Components / Federates**	**Input**	**Output**
PC for Analysis	Federate 1 Analysis	Hoisting up Hoisting down Wire speed Going forward Go back Rotation	Topside motion DCR motion Hook motion Barge motion Wire tension
Dual Crane Rig Operator	Federate 2 VR 1 (Display)	Topside motion DCR motion Hook motion Barge motion Wire tension	
	Federate 3 Controller 1 (Joystick)		Going forward(J) Go back(J) Rotation(J)
DCR Crane Operator 1	Federate 4 VR 2 (Display / HMD)	Topside motion DCR motion Hook motion Barge motion Wire tension View Point	
	Federate 5 Controller 2 (Joystick + Positional Tracking)		Hoisting up(J) Hoisting down(J) Wire speed(J) View Point(PT)
DCR Crane Operator 2	Federate 6 VR 3 (Display / HMD)	Topside motion DCR motion Hook motion Barge motion Wire tension View Point	
	Federate 7 Controller 3 (Joystick + Positional Tracking)		Hoisting up(J) Hoisting down(J) Wire speed(J) View Point(PT)
Barge Operator	Federate 8 VR 4 (Display)	Topside motion DCR motion Hook motion Barge motion Wire tension View Point	
	Federate 9 Controller 4 (Joystick)		Going forward(J) Going back(J) Rotation(J) View Point(J)

The simulation involves nine components and the federation consists of nine federates. The inputs and outputs among federates are demonstrated in the table 7-3, and how data transform among these federates will be introduced in the following chapter.

6.2.7. Data transform procedure by using integrated simulation interface

Figure 7-24 and figure 7-25 demonstrate how inputs and outputs of federates exchange date via integrated simulation interface.

Figure 7-24 indicates as follows.

(1) Controllers of DCR crane1 (federate 5) and DCR crane1 (federate 7) send out signals to hoist up.

(2) The signals are delivered to RTI through respective adapter.

(3) With FOM, RTI is able to recognize that the data has been subscribed by federate1, thus deliver data to analysis (federate 1).

(4) When the data goes through the adapters during analysis, scenario generator will transform the data into scenarios that can be understood, such as changing DCR crane 1 wire speed to 1m/min.

(5) The results include topside module motion, dual crane rig motion, DCR crane 1 hook motion，DCR crane 2 hook motion，barge motion and wire tension.

(6) With FOM, RTI is able to recognize that the data has been subscribed by

federate 2, 4, 6 and 8.

(7) Then data will be delivered to computers which accommodate federate 2, 4, 6, 8 respectively through each corresponding federates.

(8) Data from RTI will be written into memories of the computers that accommodate federate 2, 4, 6 and 8.

(9) When the shared memory notices the entrance of data, it will automatically convey the data to federate 2, 4, 6 and 8 and the movements of each objects will be reflected on each screen.

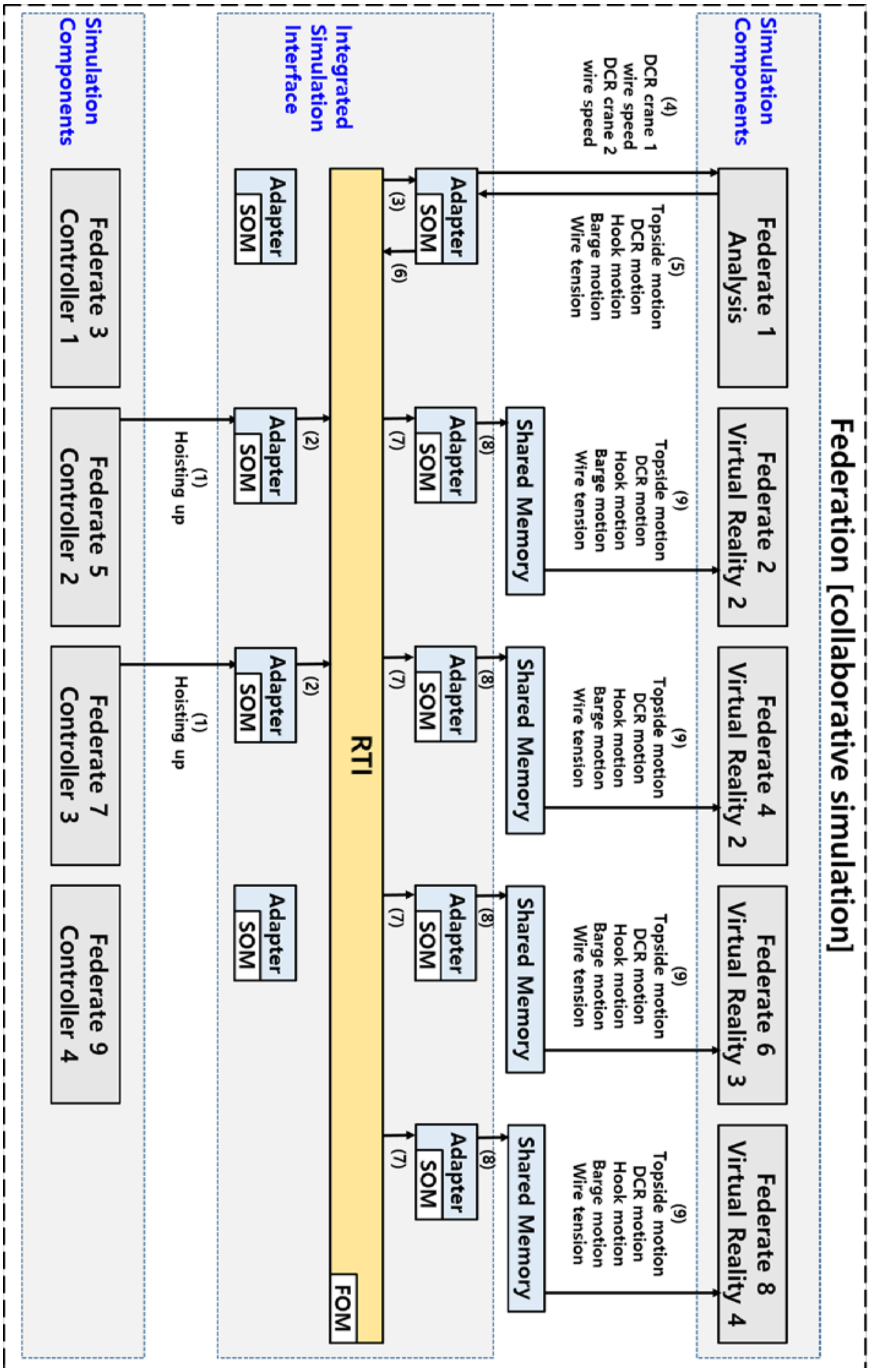


Figure 6-24 Data transform procedure by using integrated simulation interface

Figure 7-25 indicates that,

(1) When federate 5 and federate 6 change the view.

(2) Data will be converted to RTI by adapters.

(3) With FOM, RTI is able to recognize that the destination of the data is federate 4 and 6.

(4) Data will be written into the memories of computer which accommodates federate 4 and 6.

(5) With the shared memory notice the entrance of data, it automatically convery the date to federate 4 and 6.

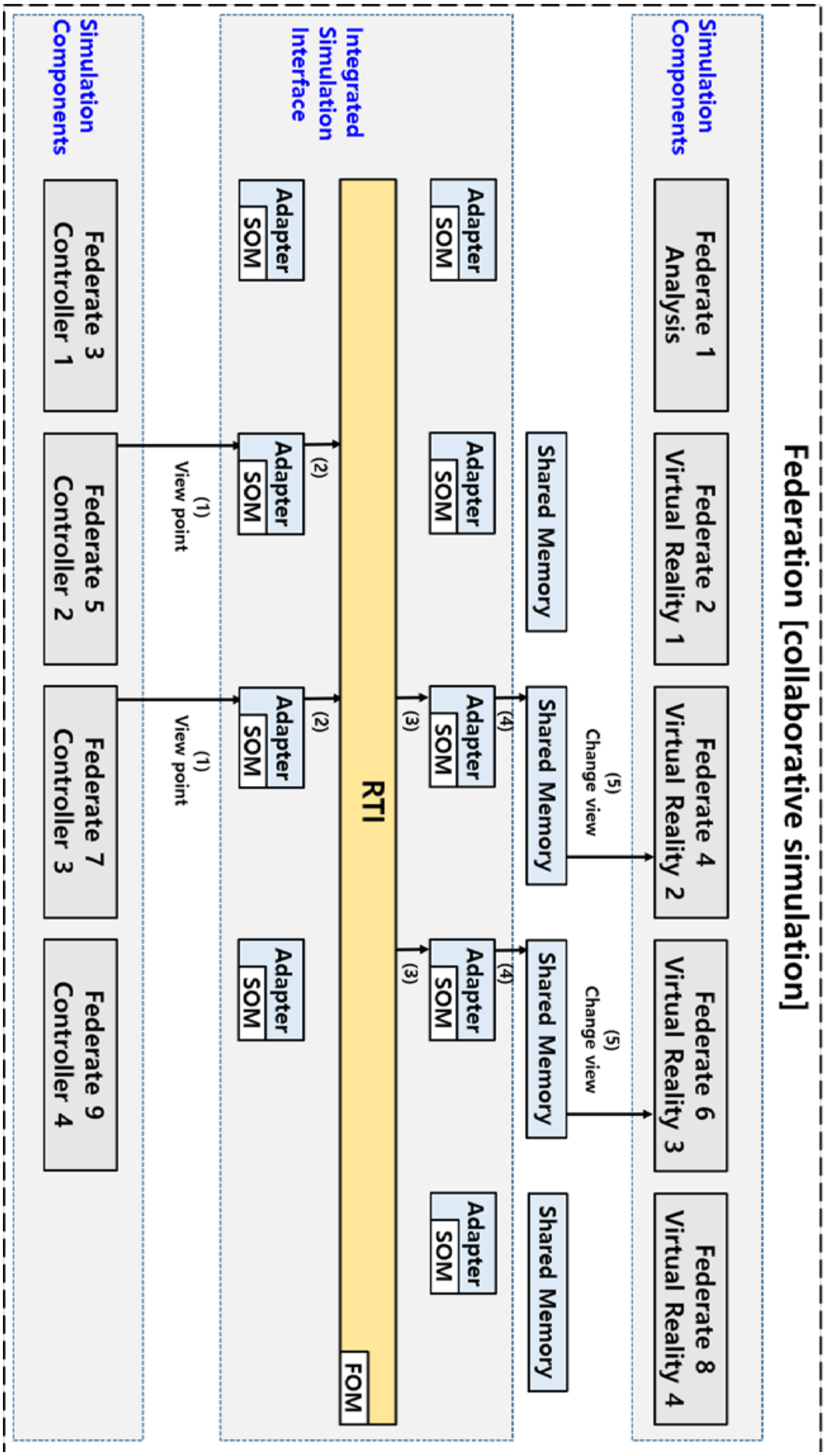


Figure 6-25 Data transform procedure by using integrated simulation interface

6.2.8. Collaborative simulation results of the prototype simulator

Figures from 7-26 to 7-30 demonstrate four operators’ views during simulation.

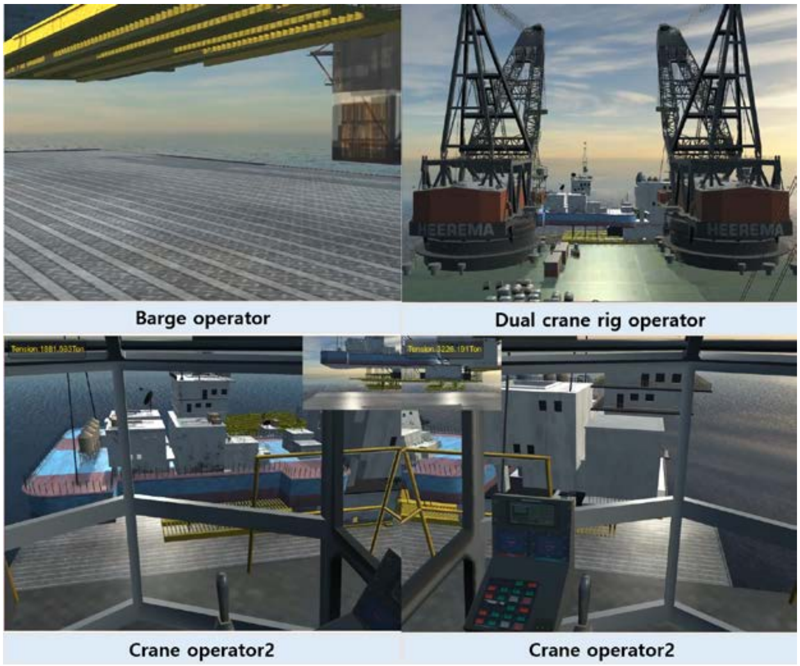


Figure 6-26 Four operators’ views during simulation (1/5)

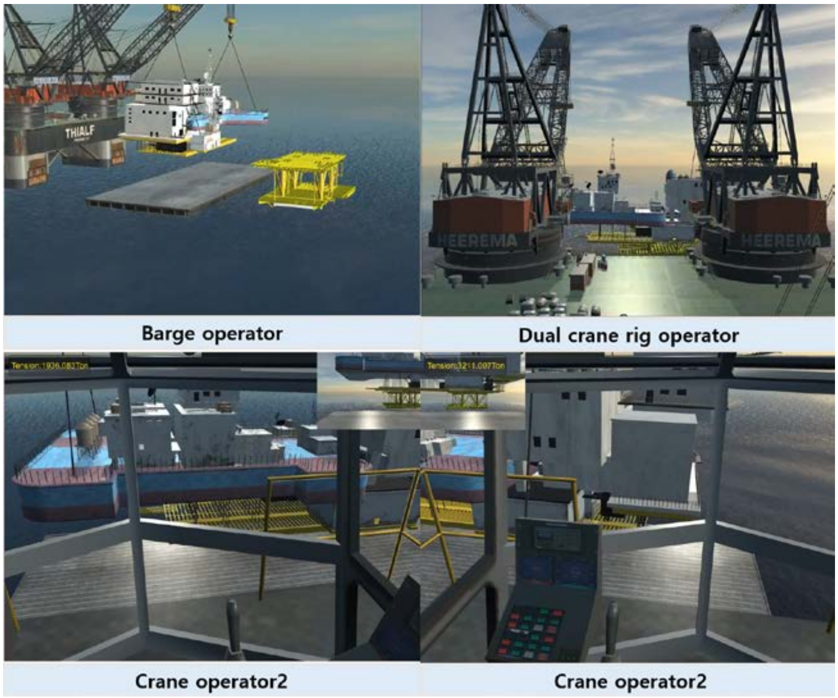


Figure 6-27 Four operators’ views during simulation (2/5)

Figure 6-28 Four operators’ views during simulation (3/5)

Figure 6-29 Four operators' views during simulation (4/5)

Figure 6-30 Four operators' views during simulation (5/5)

6.2.9. Discussion on collaborative simulation results (wire tension)

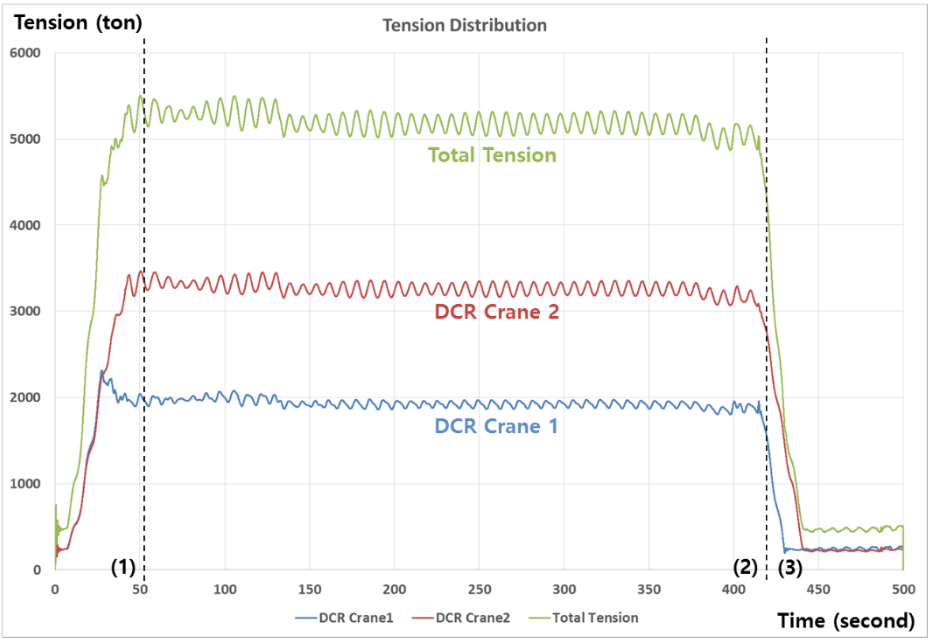


Figure 6-31 Tension distribution of collaborative simulation for topside module installation

Figure 6-31 shows the tension distribution of wires of DCR crane 1 and DCR crane 2 as time goes by. Stage (1) is the time for DCR crane 1 and 2 to be hoist up. Stage (2) is the time when DCR is moving forward and DCR crane1 and DCR 2 stay still. Stage (3) is the time for DCR crane 1 and 2 to hoist down.

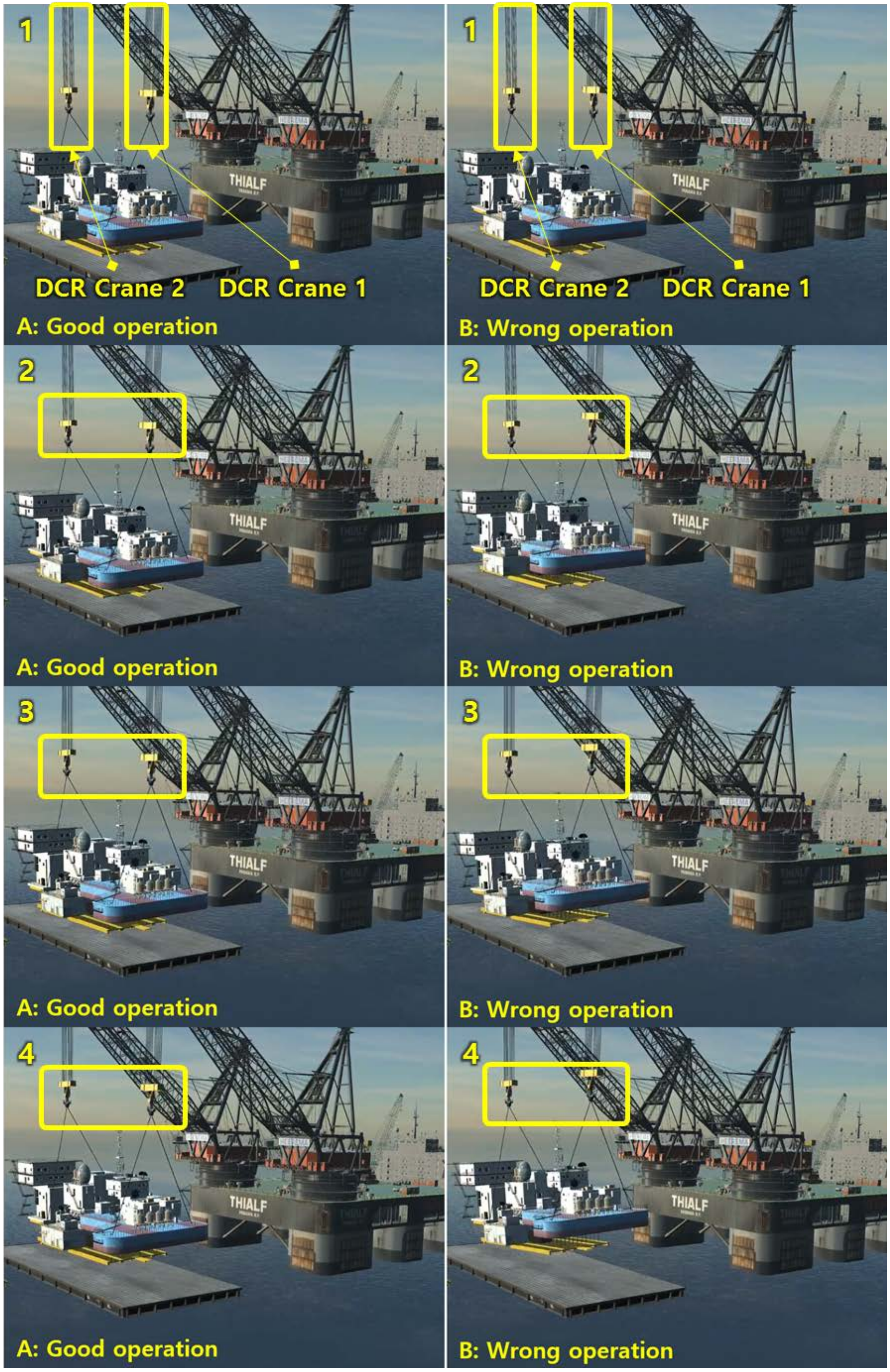


Figure 6-32 Comparison of good operation (A) and wrong operation (B)

During simulation, the centre of gravity of the object to be hoisted up is a must for consideration. In this case, the centre of gravity of topside module is more on the side where DCR crane 2 is located, so the tension of is accordingly greater. If two operators lift the module with the same speed, as shown in Figure 6-32 (B), the module will be inclined to one side or even worse, which is extremely dangerous. Therefore, the right operation should consider the location of gravity center and assign different speed to different cranes. In this case, DCR crane 1 should be lifted more slowly to maintain the topside module horizontal.

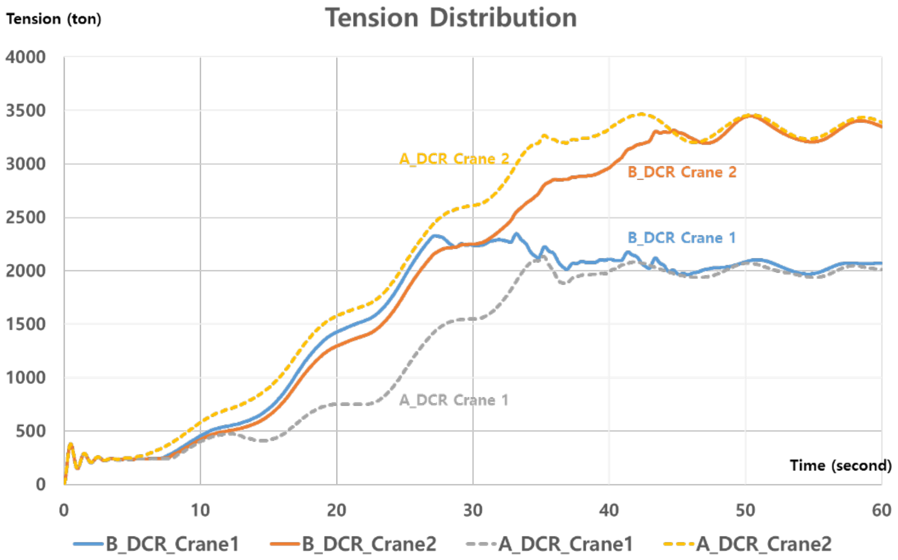


Figure 6-33 Tension distribution of good operation (A) and wrong operation (B)

The gap in tensions exists from the beginning in good operation, as the yellow and grey curves indicate in Figure 6-33, while the the difference between tensions of both cranes keeps minimum in wrong operation, which will cause dangerous situation as suggested in Figure 6-32 (B). Fortunately, such danger can be detected by collaborative simulation and be precautioned in real operation.

7. Conclusions

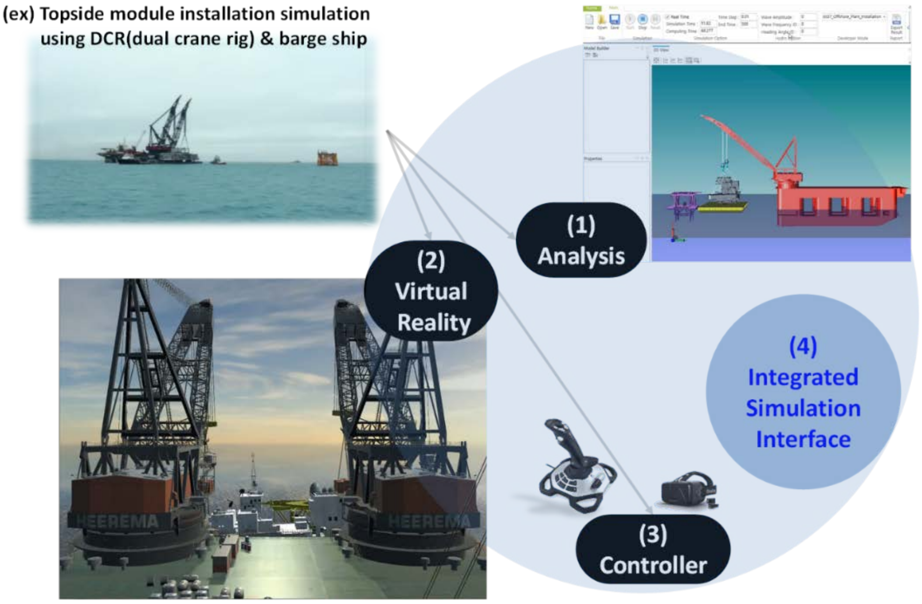


Figure 7-1 Research Overview

Multiple operators work together to accomplish ship construction and offshore installation. This current study develops a collaborative simulation to assist collaborative operation to be conducted safely and efficiently. To summarize the conclusions of this study, four main techniques are utilized to form the collaborative simulation.

1) Models are built and initial data are set in the in-house physics analysis program based on multibody system dynamic, which makes motion analysis possible. Additionally, all the models can be ensured to have same layout with VR so that the motion data can be better presented by VR.

2) VR is taken fully advantage to make every scene as close to real operation as

possible and different views are created to restore each operators’ perspectives.

3) Scenario generator is developed so that signals from controller can be delivered to analysis simulation component or VR simulation component in an accurate and efficient way.

4) Integrated simulation interface is developed with RACoN, Portico, SimGe based on HLA, which ensures communication among federates. Additionally, each federate is facilitated with an adapter, with which management serveices provided by RTI will be used and federation can be well managed.

This collaborative simulation developed by this study can be applied in many territories in shipbuilding, such as block turn-over operation and topside module installation, and so on. Each simulation allows four operators to operate under the same virtual environment simultaneously and potential risks and danger caused by operators’ mistakes can be detected during simulation. This study makes contributions in simulating potential outcomes resulting from operators’ operational mistakes, and collecting detailed data for further investigation. This study will be helpful in ensuring safety in complicated scenarios, training operators and many other aspects.

References

[1] Cha, J. H., Roh, M. I., & Lee, K. Y., 2010. “Integrated simulation framework for the process planning of ships and offshore structures”. Robotics and Computer-Integrated Manufacturing, 26(5), 430-453.

[2] Lee, J. H., 2011. “Analytical Research of Topside Installation in Mating phase with Crane Vessel. The Korean Society of Ocean Engineers”, 25(4), 1-6.

[3] Bo, W., Hu, L., Yibo, Z., & Yijie, S., 2011. “HLA based collaborative simulation of civil aircraft ground services.” In Information Computing and Applications, Springer Berlin Heidelberg, pp. 734-741

[4] Li, X., Gao, Q., Zhang, Z., & Huang, X., 2012. “Collaborative virtual maintenance training system of complex equipment based on immersive virtual reality environment.” Assembly Automation, 32(1), 72-85.

[5] Belmonte, Ó., Castañeda, M., Fernández, D., Gil, J., Aguado, S., Varella, E., ... & Segarra, J., 2010. “Federate resource management in a Distributed Virtual Environment.” Future Generation Computer Systems, 26(3), 308-317.

[6] Cha, M., Han, S., Lee, J., & Choi, B., 2012. “A virtual reality based fire training simulator integrated with fire dynamics data.” Fire safety journal, 50, 12-24.

[7] Ueng, S. K., Lin, D., & Liu, C. H., 2008. “A ship motion simulation system.” Virtual reality, 12(1), 65-76.

[8] Tozzi, D., Valdenazzi, F., & Zini, A., 2014. “Use of HLA Federation for the Evaluation of Naval Operations in Ship Design.” In Modelling and Simulation for Autonomous Systems: First International Workshop, Vol. 8906, p. 138.

[9] Longo, F., Chiurco, A., Musmanno, R., & Nicoletti, L., 2014. “Operative and procedural cooperative training in marine ports.” Journal of Computational Science.

[10] Von Schwerin, R., 1999. Multibody system simulation: numerical methods, algorithms, and software (Vol. 7). Springer Science & Business Media.

[11] Ha, S., Ku, N. K., Roh, M. I., & Hwang, H. J., 2015. “Multibody system dynamics simulator for process simulation of ships and offshore plants in shipyards.” Advances in Engineering Software, 85, 12-25.

[12] WikiHannibal, 2016. Virtual reality, Wikipedia Homepage

(https://en.wikipedia.org/wiki/Virtual_reality)

[13] Unity Technologies, 2016, Unity Official Homepage

(http://unity3d.com/unity)

[14] Unity Technologies, 2016, Unity Manual Official Homepage

(http://docs.unity3d.com/Manual/index.html)

[15] Pitch Technologies, 2016, “HLA Tutorial”, Pitch Official Homepage

(http://www.pitch.se/hlatutorial)

[16] Simulation Interoperability Standards Committee, 2010, “IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)– Framework and Rules,” IEEE Std 1516-2010 (Revision of IEEE Std 1516-2000), pp. 1 – 38.

[17] Topçu, O., 2015, RACoN Programmer's Guide v 0.0. 1.3.

[18] Calytrix Technologies Official Homepage, 2016

(http://www.porticoproject.org/)

APPENDICES

A. Implemented services of RACoN

In this section, the event handlers of RACoN API is summaried. The HLA-1516.2000(IEEE 1516.1 2010) provides 167 supported services but RACoN just implemented a part of them.

Table A-1 Federation management services

SERVICE	**RACON API**
Create Federation Execution	bool InitializeFederation(CFederationExecution fedexec)
Destroy Federation Execution	`bool FinalizeFederation(CFederationExecution fedexec, ResignAction action)` bool FinalizeFederation(CFederationExecution fedexec)
Join Federation Execution	bool InitializeFederation(CFederationExecution fedexec)
Resign Federation Execution	bool FinalizeFederation(CFederationExecution fedexec)
Request Federation Save	save as soon as possible void RequestFederationSave(String _Label) save at the specified time void RequestFederationSave(String label, Double time)
Initiate Federate Save	FdAmb_InitiateFederateSaveHandler
Federate Save Begun	void FederateSaveBegun()
Federate Save Complete	void FederateSaveComplete(bool _IsCompleted)
Federation Saved	FdAmb_FederationSaved
Request Federation Restore	void RequestFederationRestore(String _Label)
Confirm Federation Restoration Request	FdAmb_FederationRestorationRequesConfirmedtHandler
Federation Restore Begun	FdAmb_FederationRestoreBegun
Initiate Federate Restore	FdAmb_InitiateFederateRestoreHandler
Federate Restore Complete	void FederateRestoreComplete(bool _IsCompleted)
Federation Restored	FdAmb_FederationRestored

Table A-2 Declaration management services

SERVICE	**RACON API**
Publish Object Class Attributes	bool InitializeFederation(CFederationExecution fedexec)
Publish Interaction Class	bool InitializeFederation(CFederationExecution fedexec)

Subscribe Object Class Attributes	// Implicitly called in:
	bool InitializeFederation(CFederationExecution fedexec) // Subscribe OC - With its all subscribable attributes bool SubscribeObjectClass(CHlaObjectClass oc)
	// Subscribe OC - With some selected attributes
	bool SubscribeObjectClass(CHlaObjectClass oc, List<RTILayer.CHlaObjectAttribute> attributes, Boolean active = true)

Unsubscribe Object Class Attributes	// For all attributes of an object class bool UnsubscribeObjectClass(CHlaObjectClass oc)
Subscribe Interaction Class	bool InitializeFederation(CFederationExecution fedexec)
Unsubscribe Interaction Class	bool UnsubscribeInteractionClass(CHlaInteractionClass ic)
Start Registration For Object Class	FdAmb_StartRegistrationForObjectClassAdvisedHandler
Stop Registration For Object Class	FdAmb_StopRegistrationForObjectClassAdvisedHandler
Turn Interactions On	FdAmb_TurnInteractionsOnAdvisedHandler
Turn Interactions Off	FdAmb_TurnInteractionsOffAdvisedHandler

Table A-3 Object management services

SERVICE	**RACON API**
Register Object Instance	bool RegisterHlaObject(RTILayer.CHlaObject _obj)
Discover Object Instance	FdAmb_ObjectDiscoveredHandler
Update Attribute Values	bool UpdateAttributeValues(RTILayer.CHlaObject _obj) EventRetractionHandleWrapper UpdateAttributeValues(RTILayer.CHlaObject theObject, Double timestamp)
Reflect Attribute Values	FdAmb_ObjectAttributesReflectedHandler
Send Interaction	bool SendInteraction(RTILayer.CHlaInteraction theInteraction) // Send Interaction with timestamp EventRetractionHandleWrapper SendInteraction(RTILayer.CHlaInteraction theInteraction, Double timestamp)
Receive Interaction	FdAmb_InteractionReceivedHandler

Delete Object Instance	// Delete Object Instance bool deleteObjectInstance(CHlaObject obj)
	// Schedule to delete Object Instance at a specified time
	EventRetractionHandleWrapper deleteObjectInstance(CHlaObject obj, Double time)

Remove Object Instance	FdAmb_ObjectRemovedHandler
Request Attribute Value Update	bool RequestAttributeUpdate(RTILayer.CHlaObjectClass theClass)
Provide Attribute Value Update	FdAmb_AttributeValueUpdateRequestedHandler

Table A-4 Ownership management services

SERVICE	**RACON API**
Unconditional Attribute Ownership Divestiture	bool UnconditionalAttributeOwnershipDivestiture(RTILayer.CHlaObject _obj, RTILayer.RaconAttributeSet _set)
Negotiated Attribute Ownership Divestiture	bool NegotiatedAttributeOwnershipDivestiture(RTILayer.CHlaObject _obj, RTILayer.RaconAttributeSet _set) bool NegotiatedAttributeOwnershipDivestiture(RTILayer.CHlaObject _ob)
Request Attribute Ownership Assumption	FdAmb_AttributeOwnershipAssumptionRequested
Attribute Ownership Acquisition Notification	FdAmb_AttributeOwnershipAcquisitionNotified
Attribute Ownership Acquisition	bool AttributeOwnershipAcquisition(RTILayer.CHlaObject _obj, RTILayer.RaconAttributeSet _set) bool AttributeOwnershipAcquisition(RTILayer.CHlaObject _obj)
Attribute Ownership Acquisition If Available	bool AttributeOwnershipAcquisitionIfAvailable(RTILayer.CHlaObject _obj, RTILayer.RaconAttributeSet _set)
Attribute Ownership Unavailable	FdAmb_AttributeOwnershipUnavailable
Cancel Negotiated Attribute Ownership Divestiture	bool CancelNegotiatedAttributeOwnershipDivestiture(RTILayer.CHlaObject _obj, RTILayer.RaconAttributeSet _set)
Cancel Attribute Ownership Acquisition	bool CancelAttributeOwnershipAcquisition(RTILayer.CHlaObject _obj, RTILayer.RaconAttributeSet _set)
Confirm Attribute Ownership Acquisition Cancellation	FdAmb_AttributeOwnershipAcquisitionCancellationConfirmed
Query Attribute Ownership	bool QueryAttributeOwnership(RTILayer.CHlaObject _obj, RTILayer.CHlaObjectAttribute _attribute)
Inform Attribute Ownership	FdAmb_AttributeOwnershipInformed
Is Attribute Owned By Federate	bool IsAttributeOwnedByFederate(RTILayer.CHlaObject _obj, RTILayer.CHlaObjectAttribute _attribute)

Table A-5 Time management services

SERVICE	**RACON API**
Change Attribute Order Type	bool ChangeAttributeOrderType(RTILayer.CHlaObject theObject, List<RTILayer.CHlaObjectAttribute> attributes, OrderType type)
Change Interaction Order Type	bool ChangeInteractionOrderType(CHlaInteractionClass ic, OrderType type)
Disable Asynchronous Delivery	bool DisableAsyncDelivery()
Disable Time Constrained	bool DisableTimeConstrained()
Disable Time Regulation	bool DisableTimeRegulation()
Enable Asynchronous Delivery	bool EnableAsyncDelivery()
Enable Time Constrained	bool EnableTimeConstrained()
Enable Time Regulation	bool EnableTimeRegulation(Double federateTime, Double lookahead) Parameters are in seconds.
Flush Queue Request	bool FlushQueueRequest(Double time)
Modify Lookahead	bool ModifyLookahead(Double time)
Next Event Request	bool NextEventRequest(Double time)
Next Event Request Available	bool NextEventRequestAvailable(Double time)
Query Federate Time	Double QueryFederateTime()
Query LBTS	Double QueryLBTS()
Query Lookahead	Double QueryLookahead()
Query Min Next Event Time	Double QueryMinNextEventTime()
Retract	bool Retract(EventRetractionHandleWrapper retraction)
Time Advance Request	bool TimeAdvanceRequest(Double time)
Time Advance Request Available	bool TimeAdvanceRequestAvailable(Double time)
Request Retraction	FdAmb_RequestRetraction
Time Advance Grant	FdAmb_TimeAdvanceGrant
Time Constrained Enabled	FdAmb_TimeConstrainedEnabled
Time Regulation Enabled	FdAmb_TimeRegulationEnabled

Table A-6 Data distribution management Services

SERVICE	**RACON API**
Create Region	Implicitly called in: bool InitializeFederation(CFederationExecution fedexec)
Commit Region Modifications	Implicitly called in: bool InitializeFederation(CFederationExecution fedexec)
Delete Region	bool DeleteRegion(CHlaRegion region)
Register Object Instance With Regions	Registering object with a specific region with all its publishable attributes: bool RegisterHlaObject(RTILayer.CHlaObject theObject, CHlaRegion region) Registering object with a specific region with some of its publishable attributes: bool RegisterHlaObject(RTILayer.CHlaObject theObject, List<RTILayer.CHlaObjectAttribute> attributes, CHlaRegion region)
Associate Regions For Updates	bool AssociateRegionForUpdates(CHlaRegion region, CHlaObject hlaObject, List<CHlaObjectAttribute> attributes)
Unassociate Regions For Updates	bool UnassociateRegionForUpdates(CHlaRegion region, CHlaObject hlaObject)
Subscribe Object Class Attributes With Regions	bool subscribeObjectClassAttributesWithRegion(CHlaObjectClass oc, List<RTILayer.CHlaObjectAttribute> attributes, CHlaRegion region, Boolean active = true)
Unsubscribe Object Class With Regions	bool UnsubscribeObjectClass(CHlaObjectClass oc, CHlaRegion region)
Subscribe Interaction Class With Regions	bool SubscribeInteractionClass(CHlaInteractionClass ic, CHlaRegion region)
Unsubscribe Interaction Class With Regions	bool UnsubscribeInteractionClass(CHlaInteractionClass ic, CHlaRegion region)
Send Interaction With Regions	bool SendInteraction(RTILayer.CHlaInteraction theInteraction, CHlaRegion region)
Request Attribute Value Update With Regions	bool RequestClassAttributeValueUpdateWithRegion(RTILayer.CHlaObjectClass theClass, List<CHlaObjectAttribute> attributes, CHlaRegion region)

Table A-7 Support serivices

SERVICE	**RACON API**
Enable Object Class Relevance Advisory Switch	bool EnableObjectClassRelevanceAdvisorySwitch()
Disable Object Class Relevance Advisory Switch	bool DisableObjectClassRelevanceAdvisorySwitch()
Enable Attribute Relevance Advisory Switch	bool EnableAttributeRelevanceAdvisorySwitch()
Disable Attribute Relevance Advisory Switch	bool DisableAttributeRelevanceAdvisorySwitch()
Enable Attribute Scope Advisory Switch	bool EnableAttributeScopeAdvisorySwitch()
Disable Attribute Scope Advisory Switch	bool DisableAttributeScopeAdvisorySwitch()
Enable Interaction Relevance Advisory Switch	bool EnableInteractionRelevanceAdvisorySwitch()
Disable Interaction Relevance Advisory Switch	bool DisableInteractionRelevanceAdvisorySwitch()
Evoke Callback	public void Tick()

B. Prerequisites for RACoN

RACoN is complied in Visual Studio 2013 and it is decomposed into two projects, one for each dynamic library. (i.e. RACoN.dll and RACoN.RtiLayerHla13.dll). Software requirement are presented in Table B-1.

Table B-1 Software requirements

CATEGORY	COMPONENT
Operating System	**Windows 8 or later**
Runtime Environment	**Microsoft .NET Framework 4.0 Client Profile Package** **Microsoft Visual C++ Redistributable Packages for Visual Studio 2013**
HLA RTI	**DMSO RTI 1.3NGv6 or Portico 2.0.1 x86**

The RACoN down link: https://sites.google.com/site/okantopcu/racon

C. Portico environment configuration

Before setting the environment configuration, make sure the portico-2.0.1-win32.exe have installed which can be downloaded from the link as follow:

http://www.porticoproject.org/comingsoon/

Select “Properties” by left-clicking “Computer” icon and then click to “Advanced system settings”. There, click “Environment Variables” button.

First, add new two environment variables to system variables as shown in Table C-1.

Table C-1 RTI environment variables

VARIABLES NAME	RTI ENVIRONMENT VARIABLES
RTI_HOME	Portico Home Directory (typically C:\Program Files (x86)\Portico\portico-2.0.1)
JAVA_HOME	JRE Home Directory (e.g., C:\Program Files (x86)\Portico\portico-2.0.1\jre)

And than, add the following statements to the existing PATH variable:

C:\Program Files (x86)\Portico\portico-2.0.1\bin\vc10 for RTI dll files, where vc10 stands for Visual C++ 10 or later.

C:\Program Files (x86)\Portico\portico-2.0.1\jre\bin\client for jvm.dll.

초록

HLA를 기반으로 한 해석-가시화 통합 방법 연구 및 조선 해양 협업 시뮬레이션에의 적용

선박 건조 및 해양 설치 과정에서 한 대의 크레인이 아니라 다수의 크레인이 사용되는 경우가 많고, 한 대의 크레인이 사용될 때에도 이를 지원하기 위해 여러 명의 신호수가 크레인 작업자의 작업을 돕는다. 크레인 작업은 다수의 작업자에 의해 이루어지는 만큼 사고가 발생할 수가 있으며, 이를 사전에 검토하기 위해 시뮬레이션 기술을 활용할 수 있다. 이를 위해서는 여러 작업자가 동시에 접속하여 같은 작업을 수행할 수 있는 협업 시뮬레이션 기술의 개발이 필요하고, 본 논문에서는 이에 대해 연구하였다.

본 논문에서 수행한 연구 내용은 아래와 같이 4 가지로 요약될 수 있다.

첫 번째는 크레인과 블록을 사실적으로 묘사함으로써 작업자들로 하여금 실제 현장에서 작업하는 느낌을 갖도록 하기 위해, 현실감과 몰입감을 높일 수 있는 가상현실 (VR: Virtual Reality) 기술에 대해 연구하였다.

두 번째는 크레인과 블록이 물리 법칙에 따라 실제와 유사하게 움직이도록 하기 위해, 다물체 동역학 (multibody system dynamics) 기반의 해석 기술을 연구하였다.

세 번째는 작업자가 실제 크레인을 조작하는 것처럼 시뮬레이션 과정에 조종기 (controller)를 사용할 수 있도록 하고, 조종기에서 나오는 신호를 VR 과 해석의 입력 정보로 자동 변환하는 시나리오 생성기 (scenario generator)를 개발하였다.

마지막으로 가상현실 기술, 다물체 동역학 기반의 해석 기술, 그리고 조종기를 효과적으로 통합하기 위한 인터페이스를 개발하였다. 본 연구에서는 분산 시뮬레이션을 위한 표준 구조인 HLA (High Level Architecture)를 사용하여 세 가지 기술을 효과적으로 통합하였다.

본 연구에서 개발한 협업 시뮬레이션 기술을 활용하여 선박의 생산 과정 중 하나인 블록 턴오버 작업 (블록을 90 도 또는 180 도로 뒤집는 작업) 및 해양 상부 (topside) 모듈 설치 작업에 적용해 보았다. 각각의 작업에서 4 명의 작업자가 실시간으로 참여하여 동일한 환경 하에서 작업을 수행하였고, 작업자의 조작 실수 등과 같은 변수에 따라서 위험한 상황이 연출되기도 하였다. 본 연구를 활용함으로써 작업자의 조작에 의한 영향을 시뮬레이션에 실시간으로 반영함으로써 보다 상세한 결과 검토가 가능하며, 이를 통해 복잡한 시나리오에 대한 안전성 확인, 작업자의 교육 등 다방면에서의 활용을 기대할 수 있을 것이다.

Keywords: HLA, Virtual Reality, Collaborative Simulation, Shipbuilding, Offshore Installation

Student number: 2014-22145