Collaborative Visualization of a Warfare Simulation Using a Commercial Game Engine

Hyungki Kim[1], Yuna Kang[1], Suchul Shin[1], Imkyu Kim[2], and Soonhung Han[2]

[1] Department of Mechanical Engineering, KAIST, Korea
{diskhkme,balbal86,eva317}@kaist.ac.kr
[2] Department of Ocean Systems Engineering, KAIST, Korea
{kimimgoo,shhan}@kaist.ac.kr

Abstract. The requirement about reusable 3D visualization tool was continuously raised in various industries. Especially in the defense modeling and simulation field, there are abundant researches about reusable and interoperable visualization system, since it has a critical role to the efficient decision making by offering diverse validation and analyzing process. Also to facilitate the effectiveness, many current operating systems are applying VR(Virtual Reality) and AR(Augmented Reality) technologies aggressively. In this background, we conducted the research about the design for the collaborative visualization environment for the warfare simulation through commercial game engine. We define the requirements by analyzing advantages and disadvantages of existing tools or engines like SIMDIS or Vega, and propose the methods how to utilize the functionalities of commercial game engine to satisfy the requirements. The implemented prototype offers collaborative visualization environment inside the CAVE environment, which is the facility for immersive virtual environment, by cooperating with handheld devices.

Keywords: 3D Visualization, Game Engine, Warfare Simulation, Collaborative Visualization Environment.

1 Introduction

At present, 3D visualization tools are employed both directly and indirectly in research fields that require intuitive analysis and accurate data validation. A well-known application can be found in the product design field, which moved on to the 3D CAD (computer-aided design) system from paper drawings. Moreover, in the pre-manufacturing stage, the manufacturing process can now be simulated and analyzed based on the 3D visualization environment. The process based on this type of 3D-visualization-oriented analysis and validation has better effects compared to the use of traditional values or parameter-based reports of the result [1]. The requirements of 3D visualization techniques in the defense modeling and simulation field also can be estimated, as current commercial battle lab systems actively include 3D visualization functions, as do training simulators, which inherently require 3D visualization capabilities.

R. Shumaker and S. Lackey (Eds.): VAMR 2014, Part II, LNCS 8526, pp. 390–401, 2014.

However, there are several issues that require attention when adopting 3D visualization in the defense modeling and simulation fields. The main obstacle is related to the time and cost required for development. The characteristics of defense modeling and simulation systems are such that interoperability is mandatory, as new systems are frequently developed at relatively high costs. Interoperability can suppress the duplicated costs incurred during the development, interoperation and maintenance of the system. On the other hand, interoperability is closely related to reusability, as reusability is ensured if the interoperability requirement is satisfied to a certain level.

In this research, our goal is to develop an efficient decision-making environment between experts by providing the 3D visualization result of a warfare simulation. This is termed here a collaborative visualization environment. The proposed collaborative visualization environment is a limited concept stemming from the CVE (collaborative virtual environment) in its physical space as a CAVE (cave automatic virtual environment). Because the CAVE is intended to provide an immersive environment through a high-resolution multi-channel visualization system, it provides a satisfactory user experience. However, the system is usually designed for a single user with one shared screen. Therefore, we provide a collaborative visualization environment by adopting already widespread personal devices, in this case the smartphone and tablet. The interoperability and reusability problem is addressed simultaneously with the visualization quality and collaborative issues.

2 Related Research

The related researches and cases can be divided into three categories based on the interoperability level. The researches for the first category are the system dependent development cases which is widespread method in the current operating simulators. The researches belonging to the second categories are the works based on the HLA(High Level Architecture)/RTI(Run-Time Infrastructure), which is IEEE 1516 standard. The researches for the third categories are proposing the custom data structure considering the reusability issue with performance improvement.

At first, the common approach in the various systems is developing the integrated structure with simulator and visualization module. In [2], the large scale visualization system is adopted for the digital mock-up and driving simulation of the evaluation process in the maglev business. The Ogre3D engine was used for 3D visualization in this research, based on the classification and comparison between diverse graphics engines and toolkits. In [3], the simulation architecture was proposed based on the visualization engine for the real-time visualization of the defense simulation system. Also by providing the plug-in functions to manipulate the visualization algorithm, user can customize their visualization results. In [4], the objective was similar with the [3], but they focused on the representation of the synthetic environments that can be used in defense modeling and simulation systems. In [5-7] researches, the ground and aerial warfare simulation system was developed in the integrated structure with simulator and visualization module. In [5], the XNA, which is commercial game engine, was used and in [7], X-Plane was used for the visualization. In [6], authors pointed out the problem of cost-effectiveness with the commercial visualization tools. So they developed the novel LOD controlling algorithm which was described as a key problem in the visualization of the aerial warfare simulation.

In the previous researches the visualization function was developed as a part of supporting tool. And since the visualization module was integrated with the system, it is system dependent and the redundant cost for development is unavoidable as we described in the background section.

Typically in the defense modeling and simulation field, using HLA/RTI is considered as an efficient way of solving this problem [8]. HLA/RTI is the methodology to guarantee the interoperability and improving the reusability by standardizing the configuration method of the middleware. In [9], authors pointed out the interoperability problem from using the various 3D models in the single simulation system. So the proposed the Scene Simulation Platform based on the HLA/RTI. In [10], X-Plane and Google Earth was designed to interoperate based on the HLA/RTI to enable the geospatial information on the simulator. Furthermore, the simulation result was visualized on the Google Earth environment by logging the result in KML(Keyhole Markup Language) file. In [11] and [12], HLA/RTI was aggressively adopted to enable the real-time monitoring and visualization by constructing the visualization module as a separate federate. Since all the systems mentioned above are constructed as HLA-compliant, the efficient interoperation is possible by facilitating the standard-based interoperation.

However, the discussion about the semantic interoperability is not fully investigated in the previous researches. To satisfy the semantic interoperability, the data should be exchanged in unambiguous and shared manner which can be supported by the analysis and capability of the data in the individual system. HLA/RTI can guarantee the syntactic interoperability between the systems while we can point out the lack of consideration about the semantic interoperability. For instance, Vega[13] and MeraVR[14] provides the user interface, API(Application Programming Interface) and additional package to extend the visualization system as semantic interoperable. It can be a solution of efficient visualization if there is no restriction about the target simulator/federator. However, the cost and time consumption is still considerably high for the extension of semantic interoperability.

Another way to get the semantic interoperability is giving the limitation about the data that system can handle. In [15], author proposed the Universal Heterogeneous Database Metadata Description, which enables the integrated description about the battlefield by designing the data structure which has capability of heterogeneous simulation result. In [16], XML(Extensible Markup Language schema) schema was proposed to represent the state of the object in web-based battlefield visualization. In [17], the data model for construction simulation was proposed and the result was visualized. At last, SIMDIS[18] proposed the ASI format for similar objective and developed the visualization tool for defense modeling and simulation systems.

In summary, the visualization tool which depends on the simulation system suffers from redundant cost consumption for development. There are researches based on the HAL/RTI to attempt to solve the problem but still remain the requirements about the researches considering the semantic interoperability. For now, the semantic interoperability can be achieved by limiting the capable data of system.

3 Proposed Method

3.1 Overview

In this research, we adopt an approach that limits the capable data of a system in an effort to develop a collaborative visualization environment. This approach allows the efficient development of a reusable visualization tool at the level of current technology. The data is intended to represent the result of a warfare simulation. Also, to provide an efficient collaborative visualization environment for decision-making processes between experts in a 3D visualization environment, we develop a system that is based on the CAVE system, which works with state-of-the-art visualization techniques coupled with ubiquitous devices.

To address these issues, we define two underlying currently available technologies.

1. SIMDIS data file structure
2. 3D visualization and networking technologies in commercial game engine

First, SIMDIS is a well-known analysis and display tool developed by NRL (Naval Research Laboratory). SIMDIS can be utilized for result analyses in the defense modeling and simulation field. One of its advantages is that the implementation process is not necessarily for the visualization session. The ASI data file structure is well defined and has large coverage of warfare related simulation results; therefore, simply logging or parsing the simulation result allows instant visualization. Various use cases and related research show the semantic interoperability of SIMDIS in defense modeling and simulation fields. Our research allows the data file structure of SIMDIS to achieve a certain level of semantic interoperability of the tool as well as the syntactic interoperability of existing simulators with functions that log the results with a SIMDIS data file.

On the other hand, SIMDIS lacks functions for game-like scene generation, unlike other visualization toolkits such as Vega or MetaVR, as SIMDIS focuses more on objective analysis. In our research to meet the needs of game-like scene generation efficiently, we decided to use a commercial game engine. Current commercial game engines are capable of relatively high performance for 3D visualization, and the networking functions of a game engine can be employed to construct a collaborative environment.

3.2 Hardware Configuration

The collaborative visualization environment proposed in our research is based on the CAVE facility. The iCAVE facility at KAIST was built to provide an immersive virtual environment with a resolution of 6400x1920 pixels in a field-of-view angle of 120 using a seven-channel display on a cylindrical screen. The scene for each channel is controlled by single multi-channel client run on a desktop PC, and the main controller PC is set to manipulate the entire visualization system. Furthermore, personal handheld devices cooperate with the main controller to provide domain-specific data to individual users. The overall hardware configuration is illustrated in Fig. 1. The main controller PC and multi-channel clients generate the scene by means of distributed visualization with the master-slave concept. Handheld devices are connected to the main controller PC to synchronize the visualization time with the generated scene on the shared screen.

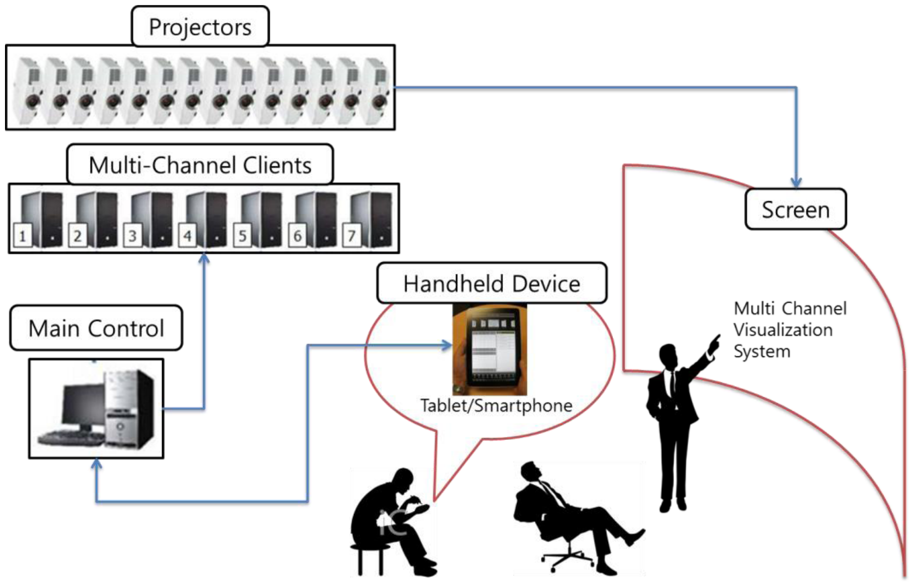

Fig. 1. Overall hardware configuration

3.3 Module Design

The entire system is designed to have a modularized structure for easy maintenance. Each module performs independent functions, and the data exchanges are accomplished through an interface that is de-signed specifically for this research. Therefore, if an update is required in the future, there is an advantage to using this type of modular design, as easily changing an individual module is all that is required.

The modules are divided into the data processing module, the weapon system visualization module, the terrain/environment visualization module, the animation module, the graph plot module, the user inter-face module and the multi-channel module. The weapon system visualization module and the terrain/environment visualization model also have an interface between their own 3D model databases. The layer structured module is illustrated in Fig.2. As noted in section 3.1, the component functions in a commercial game engine consist of upper-level visualization modules designed as part of this research. These modules run on each hardware platform to offer collaborative visualization

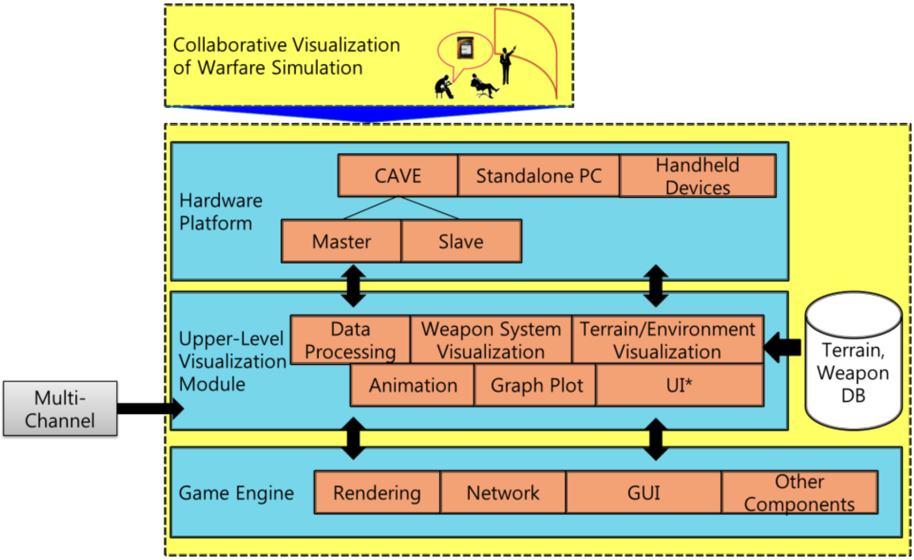

Fig. 2. Layer structure of the system and modules

3.4 Development Environment

With the designed module, a comparative study was carried out to determine the best possible development environment to use. The comparison factors are divided depending on the visualization, user interface, and networking aspects, and the compatibility. Each factor is compared in terms of its performance and efficiency during the development process. Compatibility pertains to the handheld devices, which are based on the Android OS for smartphones. Finally, the cost is also an important factor for development, as our system is based on the CAVE environment, which in some cases requires a separate license for each client PC. The result is illustrated in Table 1.

First, SIMDIS is a complete tool package which lacks extensibility compared to visualization engines, thus making multi-channel visualization difficult to achieve. OSG (OpenSceneGraph), Ogre3D and Delta3D are open-source rendering/game engines which have similar characteristics apart from their detailed features, such that Ogre3D is weaker at geospatial data handling and Delta3D supports HLA/RTI related features. Vega and Unity3D are high-level engines which have greater functionality than others. Vega offers various functions in the form of plug-in packages, but the cost is not negligible. On the other hand, Unity3D is one of the most actively applied engines in the current game market, and the range of application is expanding to the engineering and science fields. In particular, extensibility and compatibility to diverse platforms is considered to be a major advantage of this engine despite its relatively low cost. This advantage can lower the cost and shorten the time of development. In our research, the Unity3D engine was utilized for good cooperation with handheld devices in a collaborative visualization environment given its sufficient 3D visualization quality.

Table 1. Comparison of the development environments

		SIMDIS	VR-Vantage	OSG	Ogre3D	Delta3D	Vega	Unity 3D
Commercial		N	Y	N	N	N	Y	Y
Engine Level		X*	X*	△	△	△O	O	O
Visualization	Performance	△	O	O	O	O	O	O
	Implementation Efficiency	△	O	X	X	△	O	O
	GIS Support	△	O	△	X	△	△	X
UI	Scalability	X	△	O	O	O	△	O
	Implementation Efficiency	X	O	△	△	△	O	O
Network	Performance	△	O	△	△	O	O	O
	Implementation Efficiency	O	O	△	△	△	O	O
Compati-bility	Coverage	X	△	△	△	△	△	O

4 Module Development

In Unity3D, the application is constructed by setting the component functions in the scene graph nodes, known as GameObject. The components, including the rendering, networking and particle effects and the encapsulating scripts, readily enable efficient development. Thus, the upper-level module consists of the set of scripts, and the scripts control component functions and external libraries simultaneously.

4.1 Data Processing Module

The data processing module passes the data through the interface to the other modules after processing the result of the simulation data and stores it in the defined data model. In this research, we utilize the data format of SIMDIS to define the data model for defense modeling and for the simulations. The data model is a class of model which includes the overall scenario information (reference coordinates, reference time), the platform information (platform ID, classification and the name of the 3D model), the platform data (position, velocity and orientation along the simulation time), among other data. The core function of the data processing module is the parsing of the data from the result of the simulation into the class to hand over the data to other modules if the proper requirements are detected.

4.2 Weapon System Visualization Module

The weapon system visualization module manages a range of saved data in the data processing module, such as scenarios, platform information and the weapon model database. It also loads 3D model data for visualization depending on the scenario. To generate game-like scene, the visualization module can generate particle effects such as smoke and flames on the 3D model. In Fig. 3, left side figure shows a visualization result on Unity3D through the weapon system visualization module.

4.3 Terrain/Environment visualization Module

The terrain visualization module performs the loading of the terrain model near the referenced coordinate system of the scenario from the terrain database. In addition, this module performs the rendering of the sea and atmospheric environment to construct the overall environment corresponding to the scenario. The terrain database includes significant geographic information because the 3D terrain polygon models created through the pre-processing of a DEM (digital elevation map) and satellite images contain latitude and longitude information. An ocean surface model is also generated according to the camera projection matrix to create an un-bounded ocean surface, and the clouds are created using a volume model to create a realistic and atmospheric scene. Additionally the underwater effect using particles and the terrain using observed bathymetry data is implemented to render the underwater view. In Fig. 3, right side figure shows a visualized environment which encompasses the use of terrain near Ulleung Island of Korea.

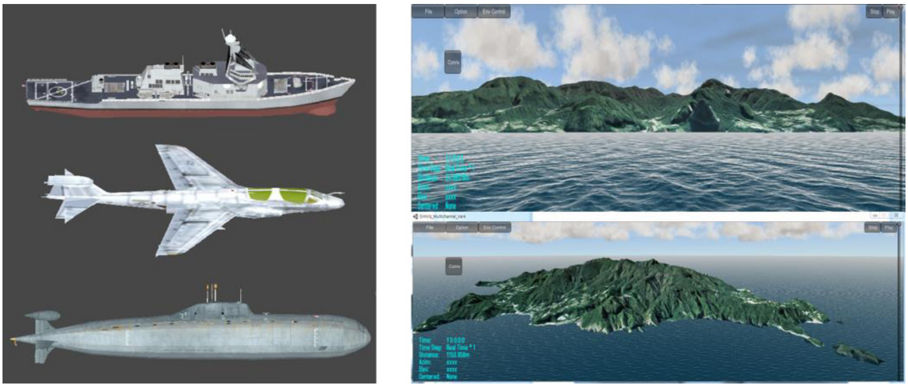

Fig. 3. Weapon system and environments around Ulleung Island visualized with Unity3D

4.4 Animation Module

The animation module manages changes of the position and orientation of the weapon system in the scenario according to the simulation time. The execution mode of the application can be divided into the real-time visualization and the after-action review steps. In the after-action review mode, visualization is performed after the parsing of the simulation log saved in the ASI file format from the completed simulation in the data processing module, making the visualization time independent on the simulation time. However, in the real-time visualization mode, the visualization time is dependent on the simulation time given that the simulation and the visualization are performed at the same time. The visualization time proceeds according to the frame update at a certain interval internally, and the position and orientation of the platform are updated with the most current data at that time. Additionally, because the data update rate should be held at 60Hz to ensure the production of smooth animations in applications such as a game, the position and orientation of the platform are updated using linear interpolation even if the simulation data does not exist at some times in the after-action review mode. This module allows the creation of smooth animations from simulation data which has irregular time intervals generated by discrete event simulations.

4.5 Multi-channel Module

The multi-channel module is developed using a type of master-slave model to visualize the entire scene in the iCAVE facility. The master-slave model is one way to realize distributed visualization, as it only transfers the data for state synchronization and runs the same applications on all node PCs. This model has advantages when used for large-scale visualization, as the network bandwidth requirement is relatively low [18].

Each multi-channel module recognizes the role of the master or the slave relative to themselves from the external initialization file at the very beginning. If the node is a slave, the view frustum of the camera of the master node is divided by seven and only renders the scene for each respective assigned camera region. The transferred data can be divided into the command information that is the one-time events, and streaming information which needs continuous transfer. The streaming information is transferred at 60Hz from master to slave and the command information is transferred immediately if the command occurs in the master. According to the information transfer, the same scene can be visualized in a multi-channel environment with the scene generated in the master. The visualized result in the iCAVE environment is illustrated in Fig. 4.

4.6 UI Module

The UI module enables the user to control and manipulate the visualization system using GUIs such as buttons or scroll bars. For instance, the input of a scenario file, termination of the application, and environment control and manipulation of the camera position are processed by means of user input commands. The UI module then transfers this information to the appropriate module through the interface.

4.7 Graph Plot Module

The graph plot module is a separate module which represents detailed values of the data of the position and orientation of individual platforms in the scenario. The graph

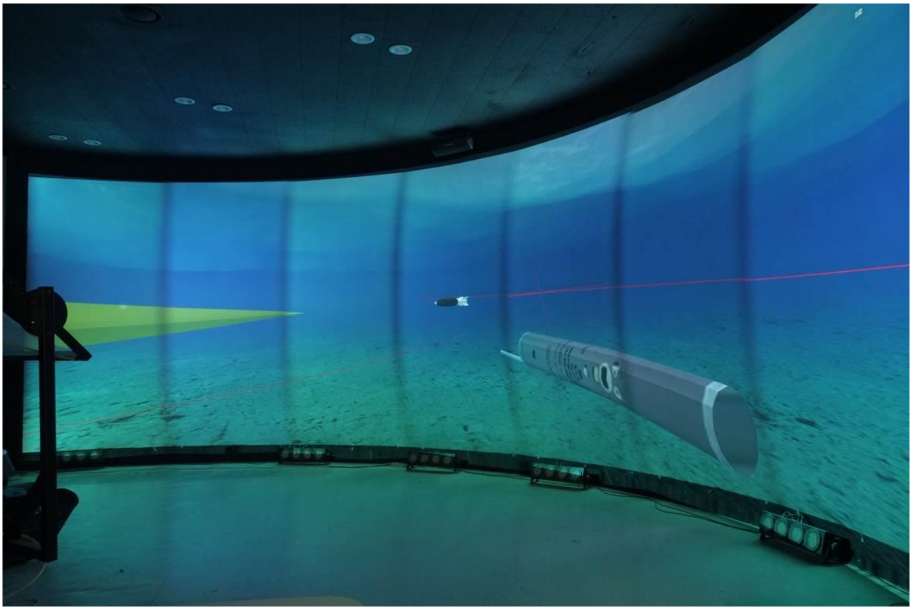

Fig. 4. Visualization result in iCAVE at KAIST

plot module is developed based on handheld devices, as the interests of individual experts can differ. Thus, to realize an immersive multi-channel environment and provide expert-specific information simultaneously, we provide detailed information with personal devices. In a situation in which divers (the experts in this case) are in the decision-making process, displaying the detailed values of each platform on a single screen is not efficient. In our study, handheld devices run the graph plot module, the data processing module, the simplified UI module and the multichannel modules for the synchronization of the visualization times so that detailed values can be observed on a smartphone.

5 Result

The entire system is implemented with the designed modules and operated with one master PC, seven slave PCs and two handheld devices, which are both smartphones running the Android OS. For the test scenario, a decoy operation scenario which includes the maneuvering of a submarine and a decoy and a battleship with the movement of its torpedoes is included. A surface-to-air and surface-to-surface missile operation scenario based on an engineering-level model was also tested. Each scenario includes three to six weapon systems and, for the decoy operation scenario, two additional 3D models were visualized to represent the detection range of the torpedoes.

The visualization was successfully done for run-time visualization and after an action review. However, in the orbit camera mode, there were shaking effects in the scene. The orbit camera mode keeps the observed platform always at the center of the camera such that if the position of the platform is updated frequently, any small synchronization difference between the animation module and the multi-channel module generates defects. This problem can be solved if the slave itself calculates the camera position by changing its updating method for the camera position as the command information.

6 Conclusion

In this research, we proposed a method to develop a collaborative visualization environment for a warfare simulation which involves the design of the required modules and the functions of the modules, and we implemented it using a commercial game engine. Regarding the interoperability of the visualization environment, we focused on semantic interoperability with the use of an existing data model stemming from a frequently utilized tool in the defense modeling and simulation field. In addition, game-like scene generation is achieved at a relatively low cost via an appropriate commercial game engine. Finally, the utility of the system is enhanced using handheld devices to provide expert-specific information.

For future works, the issue of HLA/RTI compliancy can be considered in an effort to improve the interoperability level by referring, for instance, to RPR-FOM. Moreover, the functionality of the handheld devices can be expanded to provide a more efficient collaborative environment for the decision-making process of the experts.

Acknowledgements. This work was supported by the Human Resources Development program(No. 20134030200300) of the Korea Institute of Energy Technology Evaluation and Planning(KETEP) grant funded by the Korea government Ministry of Trade, Industry and Energy and Development of Integration and Automation Technology for Nuclear Plant Life-cycle Management grant funded by the Korea government Ministry of Knowledge Economy (2011T100200145).

References

1. Delmia, `http://www.3ds.com/ko/products-services/delmia/`
2. Cha, M., Lee, J., Park, S., Park, D., Shin, B.: A development of maglev design review system using real-time visualization technology. In: Proceedings of the Society of CAD/CAM Conference (2010)
3. Hang, Q., Leiting, C.: Real-Time Virtual Military Simulation System. In: Proceedings of the 2009 First IEEE International Conference on Information Science and Engineering, pp. 1391–1394 (2009)
4. Hong, J., Park, Y., Park, S.C., Kwon, Y.: A Technology on the Framework Design of Virtual based on the Synthetic Environment Test for Analyzing Effectiveness of the Weapon Sys-tems of Underwater Engagement Model. Journal of the Korea Society for Simulation 19(4), 291–299 (2010)

5. Baki, K., Erkan, B.: Using Lanchester combat models to aid battlefield visualization. In: Internaltional Conference on Computer Science and Information Technology, pp. 290–292 (2009)
6. Kim, C., Park, Y., Lee, J., Kim, M., Reu, T.: A Study of the UML modeling and simulation for an analysis and design of the reconnaissance UAV system. J. of the Korean Society for Aeronautical and Space Sciences 36(11), 1112–1120 (2008)
7. Kim, S., Choi, J., Kim, C., Lim, C.: 3D Visualization for Situational Awareness of Air Force Operations. Jounal of KIISE 32(5,6), 314–323 (2005)
8. IEEE.: IEEE standard for modeling and simulation (M&S) high level architecture (HLA) - federate interface specification. Std. 1516.1 (2000)
9. Gang, C., Shang, X., Guan, Q.J., Quan, D.: Scene Simulation Platform Based on Data Fusion of Multiple Format 3D Models. In: International Conference on Computer Modeling and Simulation, pp. 342–346 (2009)
10. Martin, A.: Using the HLA, Physical Modeling and Google Earth for Simulating Air Transport Systems Environmental Impact. In: Simulation Interoperability Workshop (2009)
11. Hwam, W.K., Chung, Y., Choi, J., Park, S.C.: A Study on Implementation of Monitoring Sys-tem of Distributed Simulation for Underwater Warfare. Journal of the Korea Society for Simulation 22(2), 73–83 (2013)
12. Hur, P., Han, S.: Internet-Based Visualization of Underwater Vehicle Simulation using X3D. In: Proceedings of the 2006 Korea Society for Simulation Conference, pp. 48–58 (2006)
13. Vega, http://www.presagis.com/products_services/products/modeling-simulation/visualization/vega_prime/
14. MetaVR., http://www.metavr.com/
15. Wu, S.: Generic Battlefield Situation Dynamic Visualization Solution Based on Distributed Heterogeneous Database. In: 2012 Fourth International Conference on Computational and Information Sciences (ICCIS), August 17-19, pp. 1364–1367 (2012)
16. Koyuncu, B., Bostanci, E.: Using Web Services to Support Battlefield Visualization and Tactical Decision Making. In: International Conference on Computational Intelligence, Modelling and Simulation, CSSim 2009, September 7-9, pp. 138–141 (2009)
17. Kamat, V.R., Martinez, J.C.: 3D visualization of simulated construction operations. In: Proceedings of the Simulation Conference, vol. 2, pp. 1933–1937 (Winter 2000)
18. SIMDIS, https://simdis.nrl.navy.mil/