A DDS-based Distributed Simulation for Anti-air Missile Systems

Dohyung Kim[1], Hyun-Shik Oh[1] and Seong Wook Hwang[2]
[1]*Modeling and Simulation Technology Division, Agency for Defense Development, Bugyuseong daero 488 beon gil, Yuseong, Daejeon, 305-152, Republic of Korea*
[2]*M&S Department I, SIMNET Co., Ltd., 892-9 Jijok-dong, Yuseong, Daejeon, 305-330, Republic of Korea*

Keywords: Air Defense Engagement Simulation, Data Distribution Service (DDS), AddSIM, AddSIM-DDS, Component-based Distributed Simulation Environment, High-level Architecture (HLA).

Abstract: This paper introduces the development of a distributed air-defense engagement simulation model based on data distribution service (DDS). To design and develop effectively, system developers need a high-resolution engagement simulation including complex engineering-level models and operational scenario models. Increasing the resolution of the model results in the growing model's complexity which requires greater resources than that of a single computer. We tried to build a distributed engagement model using AddSIM-DDS which combines the Advanced distributed simulation environment (AddSIM) and DDS. We describe an air-defense scenario, overall structure of the model, and simulation construction on distributed nodes. We also define several DDS topic types for interoperation. Finally, we provide the results that show the validity and effectiveness of the DDS-based distributed simulation.

1 INTRODUCTION

In the defense acquisition domain, modeling and simulation is playing increasingly important roles in entire acquisition life-cycle phases. Especially an engagement simulation model gives a very useful tool for system development engineers by helping to derive detailed specification from high-level requirements. It also supports system verification and validation (V&V) through the model-based virtual experiments which evaluate system performance and effectiveness.

For the effective development of a new anti-air missile system, an air-defense engagement simulation is required, which can cover engagement and engineering levels in terms of the abstraction levels of defense modeling and simulation (M&S). It should capture scenarios including strategy and tactics at the engagement-level which are related with functions such as surveillance, threat evaluation and weapon allocation (TEWA), firing and tracking, and flying formation and evasion (Choi and Wijesekera, 2000). Likewise, it should also consider the dynamics of individual weapon systems at the engineering-level. Therefore it can be developed using a hybrid modeling approach of continuous-time and discrete event models.

For the analysis of the system developer's perspective, system developers require a higher resolution engagement simulation model than a conventional generic engagement model which stays at the system level of granularity. So the hybrid model for an air-defense engagement simulation should be able to utilize high-resolution engineering-level models directly. However, there is a problem that increased model resolution results in the growing complexity. For example, the missile model of the air-defense engagement simulation may consist of many different sub-system models which consist of multiple component models again. Each component model calculates its dynamics, update state variables, and interacts with other models on the time-scale varying from millisecond to microsecond. Accordingly, it becomes more complex and difficult to simulate than the engagement-level model alone.

On a single computer resource, it is difficult to run the high-resolution engagement simulation model which contains a number of high-resolution engineering models. It may fail to meet user requirements for execution time or it may not be executed at all due to insufficient memory. Addressing these issues, parallel and distributed simulation method has been used to simulate large and complex models (Fujimoto, 2000).

Some studies have focused on High Level Architecture (HLA)-based interoperation of

Kim, D., Oh, H-S. and Hwang, S.
A DDS-based Distributed Simulation for Anti-air Missile Systems.
DOI: 10.5220/0005980402700276
In *Proceedings of the 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2016)*, pages 270-276
ISBN: 978-989-758-199-1

engineering models to build a high-fidelity engagement simulation (Sung, Hong, and Kim, 2009; Hong et al., 2011). HLA was developed to unify various distributed simulation approaches and to define a general purpose architecture for distributed computer simulation systems. Now it has emerged as a widely adopted middleware standard for interoperation simulation. In the HLA-based interoperation approaches, distributed engineering-level models and discrete event models representing such things as command and control (C2), join the federation for the upper level engagement simulation. In addition, there have been several studies related HLA-based interoperation between high-resolution models such as virtual simulators.

However, HLA was not designed to support the low latency data sharing which is required as the simulation complexity and scale increases (Andrew, 2014; Zheng et al., 2009) The HLA Runtime Infrastructure (RTI), a software implementation of the Interface Specification of HLA, are not enough for massive distributed simulation federations with frequently exchanging data (Lopez, 2011). In order to address these challenges, AddSIM-DDS was developed which is the engagement simulation environment based on data distribution service (DDS) for distributed systems (Kim et al., 2014). The Object Management Group (OMG) DDS is a standard specification for data-centric publish and subscribe communications. DDS provides high performance with low latency and various qualities of service capabilities (OMG, 2007).

This paper introduces a DDS-based distributed simulation approach for anti-air missile systems. We made a high-resolution air-defense engagement model which can be utilized in performance prediction and evaluation of the missile system. We developed the hybrid model for the systems involved in the scenario and constructed a DDS-based distributed simulation using AddSIM-DDS. Our experiments and results show the validity and effectiveness of the DDS-based distributed simulation. Especially, we compared the results of a distributed simulation based on two different types of middleware-DDS and HLA/RTI.

2 BACKGROUND

In this study, we developed a DDS-based distributed engagement model using AddSIM-DDS. Before explaining the air-defense simulation model, this section describes DDS, AddSIM, and AddSIM-DDS, which are the background of our study.

2.1 Data Distribution Service

DDS is a functional specification to efficiently delivery data across distributed systems in publish-subscribe manner. OMG approved DDS as a machine-to-machine middleware standard since 2003. DDS aims to enable scalable, real-time, dependable, high-performance and interoperable data exchanges between applications. By these advantages, DDS has widely used in military and commercial area.

To send and receive data, events, and commands among the nodes, publisher nodes create "topics" and publish the data. DDS delivers the data to subscribers that declare an interest in that topic. The subscriber catches and uses the data. With the key benefit, the application and the DDS communication part can be decoupled; the application can be developed without determining who should receive the messages, where recipients are located, what happens if messages cannot be delivered.

Additionally, DDS allows the user to specify Quality of Service (QoS) parameters to configure discovery and behavior mechanisms. DDS simplifies distributed applications and encourages modular, well-structured programs.

In spite of its benefits, DDS has some limitations to apply to distributed simulation area which needs more requirements like federation save/restore and synchronization. Joshi et al. tried to overcome those limitations by making an equivalent to HLA-like federation or time management service (Joshi and Castellote, 2006). Nextel Aerospace Defense & Security (NADS) also focused on HLA architecture migrating to new architecture by fusing DDS middleware (Lopez and Martin 2011). They used the DDS standard as default for messaging, while the middleware object model was based on HLA metadata.

However, previous studies have some limitations. They did not fully consider building a distributed high-resolution engagement model using a reusable component-based simulation environment. As mentioned above, system developers want the execution of a high resolution engagement model within reasonable time limits. So it has been required to develop a component-based distributed simulation infrastructure which can simulate a large and complex engagement model effectively.

2.2 AddSIM and AddSIM-DDS

AddSIM is an engagement simulation environment for composing and reconfiguring weapon system models, in plug-and-play way (Oh et al., 2014). AddSIM aims to integrate the models which were developed and used during each weapon system development phase. AddSIM users can make and simulate their models in synthetic battle-fields for weapon system effectiveness analysis.

Figure 1 shows the operational concept of AddSIM. AddSIM is installed on a local computer. Users can develop models, setup simulation scenarios, execute, and get the results with GUI. Developed models are saved in the repository. AddSIM can accommodate models on remote computer. AddSIM provides environmental services (terrain, atmosphere, and maritime), spatial service, journaling/logging service.

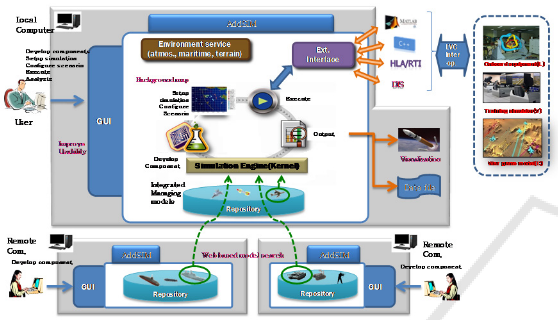

Figure 1: Operational concept of AddSIM.

A top-level component in AddSIM is called a player and it describes the behavior of a single weapon system. About distributed case, AddSIM supports integrating remote players (weapon systems) into the master simulation. The Adaptive Communication Environment (ACE) Object Request Broker (ORB), in short TAO is used for communicating data in Figure 2. AddSIM performs the simulation using distributed objects rather than executing a distributed simulation in a parallel manner.

However, the distributed object simulation of such schemes is relatively slow to communicate data because of its tightly coupled character. Also, in simulation cases of many loosely coupled participants, it seems to be needed that distributed simulation techniques, which have distributed simulation engines and communicate essential data for interoperation.

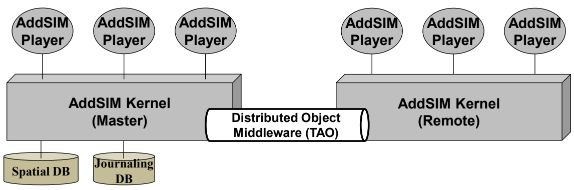

Figure 2: Distributed simulation concept of AddSIM.

Based on these ideas, AddSIM-DDS is developed. DDS middleware is added to the communication layer of the original AddSIM and some modification of Graphical User Interface (GUI) and Kernel layer for accommodate the DDS. Figure 3 shows the distributed simulation concept using AddSIM-DDS.

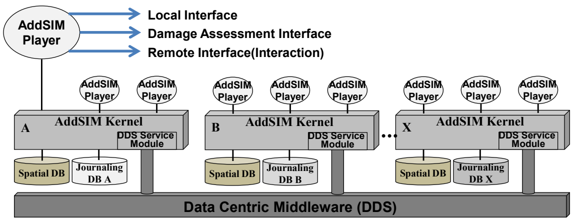

Figure 3: Distributed simulation concept using AddSIM-DDS.

In this point, we have to synchronize the spatial database because all spatial AddSIM players, almost physical combat weapon systems, shall share same spatial data for engagement. AddSIM-DDS is designed to synchronize the all spatial DB's instantly using DDS and not to journal DB's. In addition, distributed AddSIM-DDS nodes are synchronized to correctly execute the entire simulation without temporal causality errors.

3 METHOD

This section describes modeled scenario, overall structure of the model, and the way for simulation construction on distributed nodes. We utilize AddSIM-DDS to develop the system models participating in air-defense operations and construct federation.

3.1 Modeled Scenario

Our model includes enemy and friend entities with focus on a ground-to-air engagement situation. Especially, for the purpose of performance prediction and evaluation of our defensive systems, we focused on air defense operations area. With the presented scenario, we can analyze relationships between performance variables of defensive systems and operational effectiveness. For example, we analyzed how the delay of C2 network affects the final miss distance, the closest distance between aircraft and missile.

The concept of the scenario is described in Figure 4. When a fleet of enemy aircrafts are coming into our defense area, our multi-function radar detects and tracks them. After analyzing and data fusing, radar sends fire commands to a launcher with the target information. Then, fire commands are transmitted to a launcher across C2 network with some delays. After receiving the information, the launcher calculates launch direction and the midcourse way-point of inertial navigation guidance. After all the necessary information for launch is injected to a missile, the launcher sends a launch signal to the missile. The fired missile flies to a pre-determined way-point using inertial navigation guidance, and switches to homing guidance flight with a seeker. Enemy aircrafts warned on an incoming missile attack can do an evasive flight according to their operational concepts.

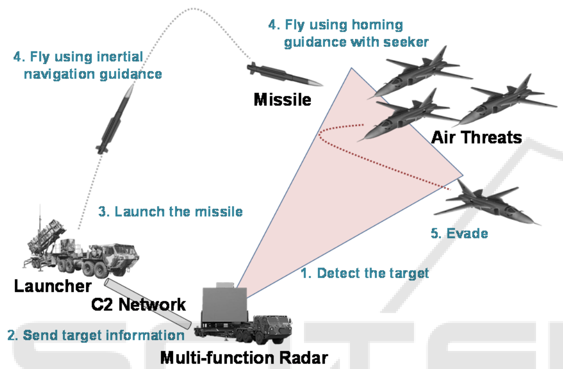

Figure 4: A brief scenario of air-defense engagement.

3.2 Simulation Construction

We designed and implemented simulation models to represent the entities described in Section 3.1. Our distributed air-defense engagement model (ADEM) includes continuous-time models representing dynamics of continuous systems such as multi-function radar (MFR), missiles (MSL), launchers (LCR), and air threats (ATS). It also includes discrete event models which describe the logic of engagement control systems (ECS), embedded control systems in missiles, and overall controls of simulation scenarios.

We used AddSIM to develop the engagement model. System-level entities in the scenario are modeled as AddSIM players. AddSIM players may have many functional and physical sub-components hierarchically, which represents behavior or physical structure of the system. Hierarchical structure of our simulation model is illustrated in Figure 5. Detailed behavior and mathematical formulations are based on the previous research (Oh and Kim, 2012). In this study, we focused on the construction of a DDS-based distributed simulation using AddSIM-DDS.

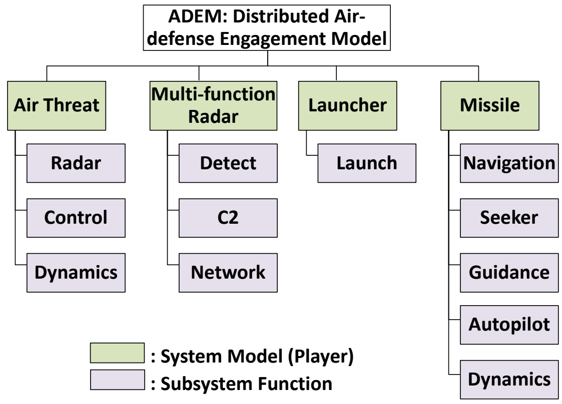

Figure 5: Hierarchical structure of ADEM.

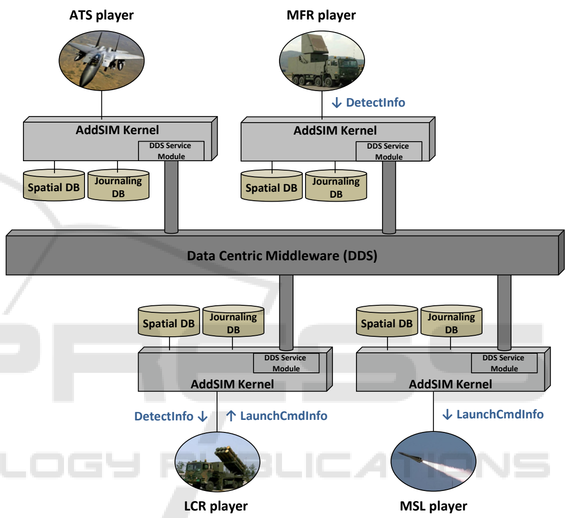

Figure 6: Structure of a distributed simulation.

The entire federation of ADEM consists of four AddSIM-DDS federates corresponding to AddSIM players. Figure 6 shows the distributed structure of ADEM simulation. All information exchanged in the AddSIM-DDS federation is divided into two categories. One is an instantaneous message for interrupt or notification, the other is persistent object data throughout the simulation. This corresponds to an interaction and object class in HLA, and an interface and spatial data in AddSIM. As described Section 2.2, every distributed AddSIM node has its own spatial database and contents of the database are synchronized by distributing updated data through DDS middleware. This spatial database plays an environmental model in the domain of an agent-based simulation.

Table 1: Topics for data exchange in ADEM federation.

Topic	Attribute	Data type	Description	QoS
DetectInfo	Detect_time	STime*	Detected time (s)	TOPIC_QOS_DEFAULT KEEP_ALL_HISTORY_QOS BY_SOURCE_TIMESTAMP_DESTINATIONORDER_QOS VOLATILE_DURABILITY_QOS RELIABLE_RELIABILITY_QOS
	Lockon	Bool	Lock on flag	
	Target_ID	Integer	Target identifier (enumeration)	
	Target_type	Integer	Target type (enumeration)	
	Target_pos[3]	Double[3]	Target position (East, North, Up) (m)	
	Target_vel	Double	Target velocity (m/s)	
	Target_azim	Double	Target azimuth (rad)	
LaunchCmd Info	Launch_cmd	Bool	Launch command flag	
	Waypoint_pos[3]	Double[3]	Waypoint for inertial navigation (East, North, Up) (m)	
SpatialInfo	Simulation_time	STime	Simulation timestamp	
	Object_ID	String	Object identifier	
	Object_parent_ID	String	Parent identifier	
	Object_Type	Integer	Object type	
	IFF	Integer	Identification friend or foe (enumeration)	
	RCS	Double	Radar cross-section (m^2)	
	Damage_state	Integer	Damage state	
	Object_state	Integer	Object state	
	Object_pos[3]	Double[3]	Object position (Latitude, longitude, and altitude)	
	Object_orient[3]	Double[3]	Object attitude (Yaw, pitch, and roll)	
	Object_LineVel[3]	Double[3]	Linear velocity in the ENU	
	Object_LineAcc[3]	Double[3]	Linear acceleration in the ENU	
	Object_RotVel[3]	Double[3]	Rotational velocity (rad/s)	
	Object_RotAcc[3]	Double[3]	Rotational acceleration (rad/s^2)	
	AdditionalState[10]	Double[10]	Additional state variables	

*STime is a structure defined in AddSIM to represent the simulation time without floating-point error.

Table 2: Topic for time synchronization in AddSIM-DDS.

Topic	Attribute	Data type	Description	QoS
TimeInfo	Publisher_ID	String	Publisher identifier	Same as in Table 1
	Executer_ID	String	Executer identifier	
	Time_stamp	STime	Federate time	
	Time_type	Integer	Time category (enumeration)	

We defined two types of message topics – ‘DetectInfo’ and ‘LaunchCmdInfo’. As described in Figure 6, allied force federates exchange them for launch information exchange. While these topics should be defined and exchanged explicitly by model developers, topics for spatial information exchange and time synchronization are created and exchanged autonomously by AddSIM-DDS. They are ‘SpatialInfo’ and ‘TimeInfo’ topics. Table 1 and 2 show DDS topics and their attributes which are defined in ADEM federation.

Table 3: Publish-subscribe relationship between federates.

Topic	ATS	MFR	LCR	MSL
DetectInfo	-	P	S	-
LaunchCmdInfo	-	-	P	S
SpatialInfo	P/S	P/S	P/S	P/S
TimeInfo	P/S	P/S	P/S	P/S

Table 3 shows publish and subscribe relations between federates.

4 EXPERIMENTS AND RESULTS

We installed four AddSIM-DDS federates on physically distributed computers in the laboratory. After multiple simulations, we confirmed that the result of the distributed federated simulation are same as the result on a single AddSIM node. Figure 7 shows the result of the distributed air-defense simulation in which number of enemy aircrafts and our anti-air missiles are involved. It depicts the same trajectories as the results of sequential simulation on a single node. Furthermore we verified all results quantitatively by comparing the simulated log.

Therefore, we can simulate a large and complex air-defense engagement model effectively in a distributed environment, without worries of insufficient memory or long simulation execution times.

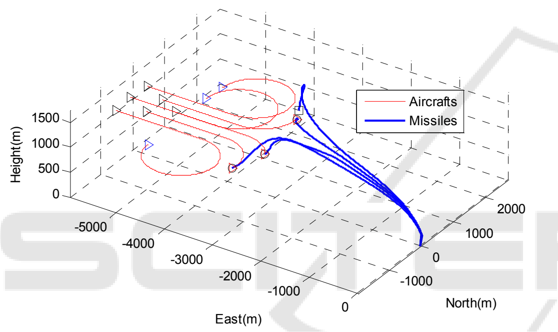

Figure 7: Simulation result of the distributed air-defense simulation.

AddSIM provides some external interfaces for interoperation with legacy models. A HLA/RTI interface is one of them (Kim, Oh, and Hwang, 2013). AddSIM also provides DDS connectivity via the kernel (Kim et al., 2014). Although AddSIM utilizes the two middleware services in a different architectural way, we were very curious on the performance difference between the two services.

For comparison, we built another set of ADEM which are the same as the previous model except for using HLA instead of DDS. In order to observe the change in the total simulation time in accordance with increasing traffic, we changed number of updates per unit simulation time (1/timestep). The comparison result is shown in Figure 8. X axis shows the number of updates per unit simulation time. We gave the seven changes from 10 to 10,000. The data size of each update are 1,192 bytes on the average - SpatialInfo (266 bytes) and TimeInfo (32 bytes). Sometimes additional traffic are generated - DetectInfo (22 bytes) and LaunchCmdInfo (14 bytes), but the impact on the overall traffic is relatively small.

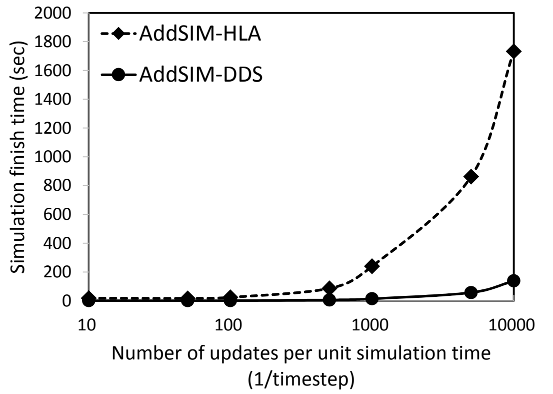

Figure 8: Simulation finish time vs. number of updates in AddSIM-DDS and -HLA.

We employed OpenDDS Ver 3.8 as the DDS middleware and MÄK RTI Ver 4.1 as the HLA/RTI middleware. All experiments were performed on four distributed computers in the laboratory. Each computer has Intel® Core™ i7 3.40GHz processor and 8GB RAM. They are running on Microsoft Windows 7 Ultimate and communicating via a 1GB Ethernet network.

Although there are some constraints and assumptions, we could check performance degradation on the simulation finish time with AddSIM-HLA. More importantly, the performance degrades significantly after a certain point of traffic. Therefore, AddSIM-DDS is more effective to simulate a large and complex air-defense engagement model in a distributed environment.

These results are limited to the case of AddSIM-DDS and –HLA, and cannot be generalized to DDS and HLA. As mentioned, this experiment includes constraints and assumptions such as each middleware in a different architectural way, ignoring various parameter setting of middleware, and not performing the statistical analyses.

5 CONCLUSIONS

We introduced a DDS-based distributed engagement simulation approach for the development of air-defense guided weapons systems. We constructed the distributed simulation using AddSIM-DDS federates. DDS gives a powerful communication infrastructure with its real time performance, high rate messaging, and various QoS capabilities. And it gives a great synergy when combined with component-based high resolution simulation environment-AddSIM.

The presented approach can be applied effectively to a large and complex engagement simulation in which a number of high-resolution engineering models participate. Our experiment results show the validity and effectiveness of the DDS-based distributed simulation. However, the results of our experiments are limited to the case of AddSIM-DDS and –HLA. And we need to expend more effort on the statistical analyses and application to other complex engagement simulation models continuously.

REFERENCES

Andrew Foster. Prismtech. 2014. *Using DDS for scalable, high performance, real-time data sharing in next generation Modeling & Simulation systems* (DDS-Modeling-Simulation-WP-050914).

Choi, S. Y., & Wijesekera, D. 2000. The DADSim air defense simulation environment. In *High Assurance Systems Engineering, 2000, Fifth IEEE International Symposim on. HASE 2000* (pp. 75-82). IEEE.

Fujimoto, R. M. 2000. *Parallel and distributed simulation systems* (Vol. 300). New York: Wiley.

Hong, J. H., Seo, K. M., Seok, M. G., & Kim, T. G. 2011. Interoperation between engagement-and engineering-level models for effectiveness analyses.*The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, *8*(3), 143-155.

Joshi, R., & Castellote, G. P. 2006. A comparison and mapping of data distribution service and high-level architecture. *Technology, The Netherlands. His research interests include parallel and distributed computing, component based architectures, and embedded systems*.

Kim, D. H., Oh, H. S., & Hwang, S. W. 2013. Integrating legacy simulation models into component-based weapon system simulation environment. In *Proceedings of the 2013 Summer Computer Simulation Conference* (p. 26). Society for Modeling & Simulation International.

Kim, D., Paek, O., Lee, T., Park, S., & Bae, H. 2014. A DDS-based distributed simulation approach for engineering-level models. In *Simulation Conference (WSC), 2014 Winter* (pp. 2919-2930). IEEE.

Lopez-Rodriguez, J. M., Martin, R., & Jimenez, P. 2011. How to Develop True Distributed Real Time Simulations? Mixing IEEE HLA and OMG DDS standards. *Nextel Aerospace Defense & Security (NADS), http://simware. es*.

Oh, H. S., & Kim, D. H. 2012. Generic simulation models to evaluate integrated simulation environment. In *Advanced Methods, Techniques, and Applications in Modeling and Simulation* (pp. 413-424). Springer Japan.

Oh, H. S., Park, S., Kim, H. J., Lee, T., Lee, S., Kim, D., ... & Park, J. H. 2014. AddSIM: A new Korean engagement simulation environment using high resolution models. In *Simulation Conference (WSC), 2014 Winter* (pp. 2942-2953). IEEE.

OMG (Object Management Group). 2007. Data Distribution Service for Real-time Systems Specification Version 1.2. http://www.omg.org/spec/DDS/1.2/PDF/formal-07-01-01.pdf.

Sung, C. H., Hong, J. H., & Kim, T. G. 2009. Interoperation of DEVS models and differential equation models using HLA/RTI: hybrid simulation of engineering and engagement level models. In *Proceedings of the 2009 Spring Simulation Multiconference* (p. 150). Society for Computer Simulation International.