

Research Article

Simulation Testing of Maritime Cyber-Physical Systems : Application of Model-View-ViewModel

Dong-Chul Lee,¹ Kyung-Min Seo,² Hee-Mun Park,² and Byeong Soo Kim ³

¹Daewoo Shipbuilding & Marine Engineering (DSME), Siheung 15011, Republic of Korea

²Korea University of Technology and Education (KOREATECH), Cheonan 31253, Republic of Korea

³Myongji University, Seoul 03674, Republic of Korea

Correspondence should be addressed to Byeong Soo Kim; kimbs@mju.ac.kr

Received 5 August 2021; Revised 12 October 2021; Accepted 29 October 2021; Published 21 January 2022

Academic Editor: Bo Xiao

Copyright © 2022 Dong-Chul Lee et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

From the perspective of the system of systems development, system-level functional testing is required for designing subsystems. This study utilizes modeling and simulation techniques to analyze the operational behaviors of the subsystems and confirm data communication between them. The targeted system in the study is a naval combat system (NCS), which is a typical type of defense cyber-physical system (CPS). Three types of models were designed for the simulation testing of the NCS: a combat-management model for simulating the overall computational activities, physical models to confirm the intrasubsystem behaviors, and data integration models to test the intersubsystem communications. These models are realized with the Model-View-ViewModel design pattern, which strongly facilitates graphical user interfaces being decoupled from model logic and data. We consider underwater combat scenarios as an application. Six significant physical subsystems within the NCS are simulated and tested: a ship-steering system, an inertial navigation system, a global navigation satellite system, a periscope, sonar systems, and a plotting board. We expect that the proposed work will play a principal role when analyzing the behaviors and communications of defense CPSs and providing an environment for functional testing as a digital twin.

1. Introduction

A military combat system is a type of maritime cyber-physical system (CPS) [1–3] that collects physical data through sensor subsystems [4]. In turn, it makes real-time decisions by feeding the data into computing subsystems [5]. As a naval combat system (NCS) for a surface ship or underwater vessel is highly complex, its functionalities regarding sensors, computers, and communications should be tested individually in system of systems engineering [6, 7].

For the functional testing of an NCS, the following real-world requirements at the subsystem level must be considered. Firstly, most physical subsystems of an NCS are highly distributed [8]. As the subsystems operate separately, they have multiple console panels to identify and control their operations. Next, the subsystems usually have different data formats and communication protocols [9]. To communicate between the

subsystems normally, NCS requires data integration systems (DISs) to convert their input/output (I/O) data into interpretable forms. Thus, DIS provides the loose connections of subsystems, which guarantees flexible system adjustment, e.g., by allowing the addition, removal, and change of the subsystems.

Because of massive costs in resources and time, it is difficult to test an entire NCS with actual subsystems. For such cases, modeling and simulation (M&S) techniques have become good alternatives. Although CPSs containing an NCS are intrinsically complicated, well-categorized models for them are understandable, analyzable, and certifiable [10–12]. Therefore, they give helpful simulation results and insights to develop CPSs successfully [13–15]. Based on the requirements of an NCS, the functional scope of the simulation testing conducted for this study is two-fold: (1) the subsystems' operational behaviors and (2) data conversion between them.

Various M&S methods for defense CPSs have been used over the last decade. Some researchers have proposed classification methods for CPS models [16–19], and others have developed simulation systems with diverse modeling languages [20–22]. The collective contribution of these studies has been to suggest meaningful modeling methods and simulation software for defense CPSs. Despite this contribution, the methods still warrant some improvements. For example, some CPS models lack a realistic configuration of their target systems. They focus on evaluating a CPS’s integrated performance as a whole rather than analyzing the operational behaviors of subsystems. In addition, some researchers developed simulation systems with a centralized approach so that the communications between the subsystems could not be simulated or expanded. Thus, the previous techniques were not suitable for subsystem-level simulation testing.

This study presents a practical simulation approach to testing defense CPSs, specifically an NCS, at the subsystem level. We functionally classify CPS models that closely resemble the components of the real NCS. Then, we develop a distributed simulation system to help with analyzing intrasubsystem behaviors and confirm the conversion of intersubsystem data.

The designed models were implemented using the Model-View-ViewModel (MVVM) design pattern [23] in Windows Presentation Foundation (WPF). The main advantage of the MVVM pattern is to decouple the graphical user interface (GUI) significantly from the model logic and data [24]. We created GUIs for the subsystems’ console panels with extensible application markup language [25]. The logical parts of the models—e.g., the decisions, dynamics, and data conversions—are implemented in the models of the MVVM pattern. This separation between data and logic increases the transparency, modularity, and flexibility of the overall modeling, i.e., the external graphical interfaces and the internal model logic. For this reason, the MVVM design pattern has been used widely to develop software for human-computer interaction (HCI) in many industrial fields [26, 27]. To the best of our knowledge, this study is the first attempt to develop a practical implementation of the MVVM design pattern for simulating a naval CPS.

As a practical case study, we simulated underwater combat operations. The physical subsystems to be modeled in this simulation included a ship-steering system, an inertial navigation system (INS), a global positioning system (GPS), a periscope, sonar systems, and a plotting board. A DIS model was developed to convert simulated data from the INS model into real-system data comprising hexadecimal values with a predefined header. The empirical results show how the physical models behaved during the simulation and how the DIS model transformed the collected data into compatible information for system scalability. With several discussions, we expect that the developed simulation system can be utilized to test an NCS at the subsystem level successfully.

The study is organized as follows: Section 2 describes the NCS briefly to provide the background, and Section 3 analyzes previous works. Section 4 proposes the overall simulation architecture and the realization of the three model

types based on the MVVM pattern. Section 5 explains and discusses an application for naval combat simulation. Finally, Section 6 presents our conclusions.

2. Background

An NCS, as a defense CPS, evaluates real-time threat situations and assigns appropriate weapons based on complex computations. Figure 1 shows a simplified configuration of an NCS. It comprises the multiple families of physical subsystems that perform similar functions (the red part in Figure 1). For example, the navigation sensor systems containing a GPS and an INS calculate the position, orientation, and velocity of their vessel. Underwater sensor systems, as typified by sonar systems, mainly detect objects in the water, and surface sensor systems, such as radar or periscope systems, perceive objects on the water’s surface [28]. The underwater and surface sensor systems must be integrated with the navigation sensors because the accuracy of target detection requires precise self-localization. Weapon systems, e.g., missiles or torpedoes, contain independent objects launched from the vessel against offensive and defensive tactics.

As a computational part of the CPS, a combat management system (CMS) utilizes all information gathered from sensor and weapon systems [29]. It continuously performs the information-processing stages in the computational domain. For example, the CMS perceives the surroundings of its vessel and identifies approaching threats. After tracking them, it launches appropriate weapons and controls them, if necessary [30]. The CMS also contains several subsystems. As shown in the green part of Figure 1, it has multifunction consoles for HCI. The multifunction consoles display the overall tactical situation upon awareness by data integration from signal-processing units. They assist vessel operators with understanding the situation and commanding their decision-making.

An NCS configures a combat system data bus (CSDB) and a central bus network to connect all the physical and computational subsystems. Note that the physical subsystems do not plug into the CSDB directly. The subsystems within the same group link to a data-integration system (DIS), which is interconnected across the CSDB sequentially [31]. In this case, the DIS (the blue part in Figure 1) performs essential roles as a gateway [32]. As each subsystem has a different data format and communication type, the DIS converts its I/O data to compatible forms with the CSDB. It also allows new subsystems to be added to the same system group without directly affecting the CSDB. Therefore, the physical subsystems are connected through interoperability arrangements, which guarantees that they do not require strong coupling or tight integrations [33].

To summarize, the NCS is a defense CPS that has physical and computational subsystems connected via a centralized network [34]. For data conversion and system scalability, NCS also contains several DISs according to the subsystem types. In this regard, we developed three models for simulation testing: combat management (CM), physical, and DIS models.

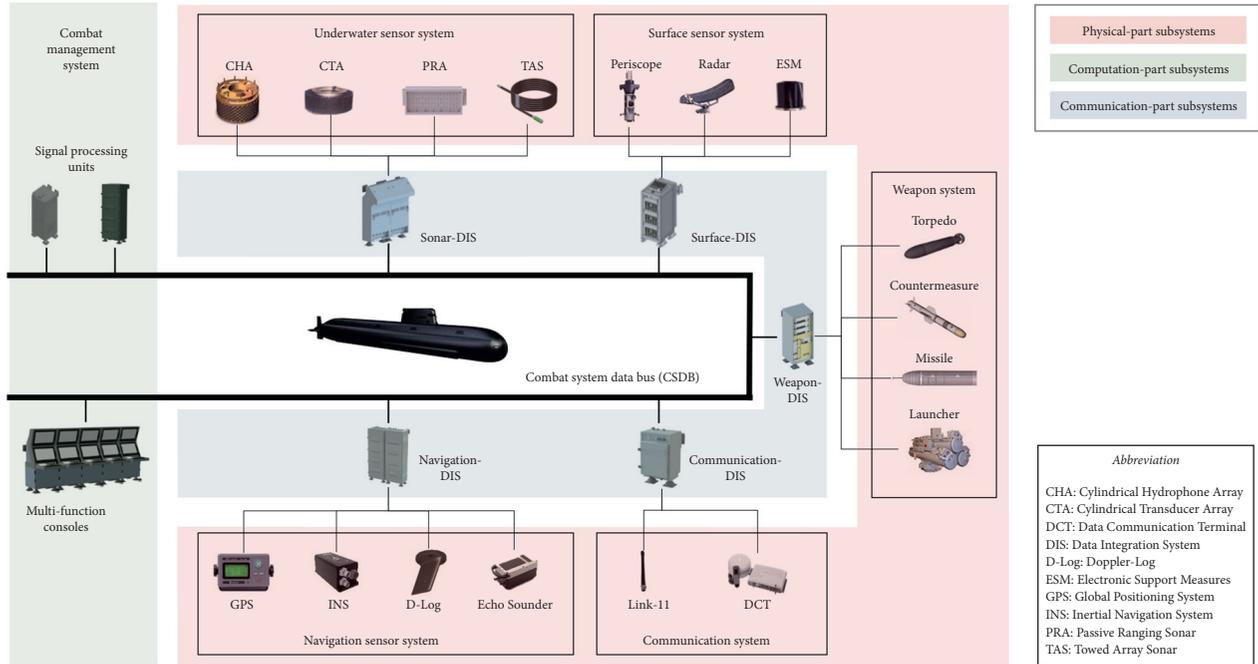


FIGURE 1: Simplified configuration of the naval combat system classified into three components: physical, computation, and communication subsystems.

3. Literature Review

Simulation testing utilizes the computerized models of a real system to evaluate the corresponding system's functionality under a given set of conditions [35]. Over the last decade, various studies have been conducted to provide indications for the simulation testing of defense CPSs. In this section, we categorize them by aspects of their modeling and simulation, as summarized in Table 1.

From a modeling perspective, several classification methods have been proposed for defense CPS models. For example, Sung and Kim [16] developed a collaborative modeling methodology for a domain-specific discrete event-simulation system, which categorizes models as discrete event-level models or behavioral-level models. Li et al. [17] similarly explained two-level modeling (physical and decision modeling) for defense CPSs. They focused on decision modeling rather than physical modeling because combat effectiveness is closely related to the tactics of the defense CPS under certain conditions. Zhu et al. [18] introduced domain-specific metamodeling with two orthogonal dimensions. Ontological modeling locates a model element from a domain-definition perspective, and linguistic modeling is concerned with modeling language definition. Park et al. [19] split a defense system into two parts for CPS modeling: core parts representing the system's physical and performance characteristics and shell parts for its operational attributes.

Although the above modeling methods provide transparent classification systems, they focused more on analyzing the integrated functionalities of the target systems. They tend to explain a holistic view of their modeling rather than the modeling itself. Thus, they have a weakness when

evaluating the functionalities of subsystems, such as navigation, radar, and sonar systems. In addition, as the methods were developed to enhance general modeling indices, e.g., model composability and reusability, the authors were not interested in basing the model components on a realistic system configuration.

Simulation studies have been conducted to represent defense CPS models graphically and analyze them. Squarcia et al. [20] developed a numerical simulation tool named Integrated Combat Systems Simulation (ICS-Sim), which allows the integration of simulation models to analyze and evaluate systems' performance for studying innovative algorithms. As these studies were fundamentally developed with a centralized simulation environment, it is difficult to apply them to geographically dispersed simulation systems with a common simulation platform.

Similar to our simulation approach, some researchers focused on distributed simulation with realistic components of defense CPSs. For example, Xu et al. [21] federated one computational simulation system at the engagement level and three physical simulation systems at the engineering level. Cheng et al. [22] also developed a distributed simulation system containing a general simulation platform, several observation systems, and a navigation system. They focused on how to construct interoperable simulations with legacy systems and achieve data flows according to real-time constraints. Despite their contributions, however, these works do not provide behaviors or communication data at the subsystem level.

In recent years, our research team has carried out various M&S studies for defense CPSs. For successful model-based systems engineering (MBSE), we have studied M&S activities across all phases of MBSE. For example, we have

TABLE 1: Related works for modeling and simulation studies of defense cyber-physical systems.

Previous work	Motivation	Method
Sung and Kim [14–16]	A collaborative modeling process was proposed. It formally defines the roles and responsibilities of domain engineers, M&S engineers, and platform engineers.	The models can be switched at the run-time or the load-time of a simulation run by the dynamically linked library technology.
Li et al. [15–17]	Domain-specific modeling containing physical and intelligent aspects was developed.	Code generation mechanisms transform domain-specific decision models to python code.
Zhu et al. [16–18]	The proposed two-level metamodeling was based on petri nets to support continuous state transitions and event triggering.	MetaDepth was used for supporting the two-level textual modeling and implementing the deep semantics.
Park et al. [17–19]	A defense CPS model is classified into core and shell parts. The two parts are specified with the DEVS (discrete event systems specification) formalism.	The Delta3D engine was used to represent combat entities graphically within a centralized simulation environment.
Squarcia et al. [18–20]	A multiscale method was applied for modeling and simulation of systems/subsystems with a different degree of detail and accuracy.	An integrated simulation environment was developed for the performance evaluation of entire naval CPSs.
Xu et al. [19–21]	The proposed warship model contains one computational model for overall tactical behaviors and three engineering models regarding weapons.	A specific software-hardware environment was required to interconnect legacy systems.
Cheng et al. [20–22]	The proposed model for a CPS contains navigation, radar, and measurement submodels.	Three key technologies are resolved in their simulation system: general simulation platform, distributed simulation, and data exchange.

conducted the requirement analyses of next-generation systems [36] and created several modeling methods for CPSs, such as conceptual modeling [37], discrete event modeling [38], multifidelity modeling [39], and distributed simulation for heterogeneous systems [40]. Based on this methodological background, this study develops a practical simulation system with CPS models that are more realistic. The critical points for this study are (1) how to evaluate behaviors of subsystem models graphically for performance analysis and (2) how to convert their communication data for system scalability.

4. Proposed Work

This section, firstly, proposes the overall simulation architecture for testing the naval CPSs. Then, we explain the three main components: the CM, physical, and DIS simulators.

4.1. Overall System Architecture. Simulation testing is the process of designing and creating a computerized model of an engineered system for conducting tests to evaluate the behavior of the corresponding real system under a given set of conditions [35]. The proposed CPS simulation confirms intrasystem behavior and intersystem communication, which is required for simulation testing. To this end, the simulation architecture in Figure 2 utilizes three types of simulators: a CM simulator for overall simulation activities and physical and DIS simulators to achieve the above-stated requirements.

The CM simulator supports the continuous execution of the information-processing stages in the tactical domain and builds a situational picture of the ship’s surroundings [29]. To simulate the overall actions of all the combat entities, it contains simulation engines and model constructions, which will be explained in the following subsection. After

simulating the main activities, the simulator sends the simulated data to the relevant physical simulators.

The physical simulators realize their dynamics and behaviors by imitating the console panels of the corresponding system. As the real NCS has various types of physical subsystems, we selected several subsystems developed in this study, as shown in Table 2. The vessels usually use multiple navigation systems, which provides greater accuracy than possible when using any single system. For example, physical simulators control the propulsion, check their own behaviors with navigation simulators, detect threats with sonar and periscope simulators, and check the surroundings with a plotting board simulator. In this study, a periscope simulator was developed for virtual-constructive simulation, which will be explained in the discussion section.

Finally, the DIS simulator performs a gateway role. It links the existing physical simulators to external CPSs or their simulators by generating compatible I/O data on both sides. In other words, it stores received simulation data, converts them to real-world data, and sends the converted data externally.

4.2. Combat Management Simulator Development. Figure 3 shows the configuration of the CM simulator. It consists of simulation engines and a simulation model, which means that models can be developed independently of a simulation engine and interfaced with the simulator using an API.

The simulator layer contains platform-level models. The submarine model has various submodels, as well as submarine and target models. The combat-management simulator enables the users to interact with and control the physical systems for the large-scale systems [42]. In the model layer, the space module simulates the physical signals to be transferred within the environment. The acoustic

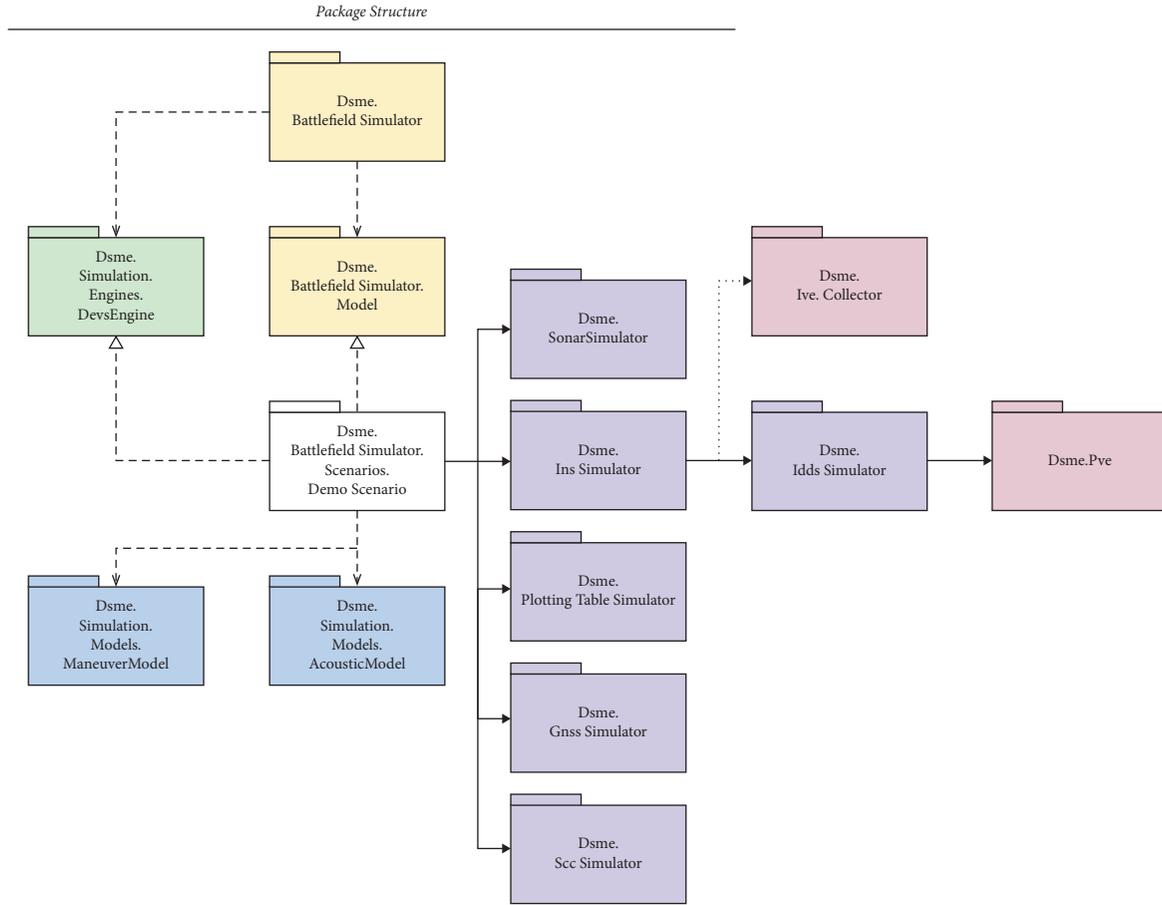


FIGURE 2: An overall simulation architecture for testing cyber-physical systems.

TABLE 2: Physical subsystems targeted in this study.

Physical subsystems	System group	Subsystem role
Ship steering system	Propulsion system	It controls positional and postural information.
GPS	Navigation sensor system	It detects and calculates positional and postural data of own ship on the water.
INS	Navigation sensor system	It detects and calculates positional and postural data of own ship underwater.
Sonar system	Underwater sensor system	It detects and identifies underwater and surface ship threats.
Plotting board	Surveillance system	It checks overall situations with battlefield condition.
Periscope	Surface sensor system	It checks sea-level situations.

module simulates the acoustic signals generated from the engines and propellers of the combat entities.

It is a closed-loop system that repeatedly performs object detection and data processing, makes command decisions, and determines how to respond to actions by appropriately using composite equipment over the course of a war. Each submodel is designed by the system taxonomy. In this study, we used a discrete event system and a discrete time system. For example, the data from the sensor and weapon models are fed into a control system simulation that models the sequence of events, from detection of the threat by various sensors to its engagement by different weapons.

Figure 4 shows the class diagram for the combat-management simulator to implement the concept of Figure 3. To resolve this issue from the software side, we use the MVVM design pattern in the WPF framework. We divide

the class diagram into View, ViewModel, and Model horizontally according to the MVVM pattern.

In Model, the combat-management simulator conducts a simulation algorithm and model logic. In View, the user can configure the scenario. The battlefield simulator helps the user to configure new scenarios. In the scenario, the user sets the model structure and values of the model's attributes, observes the simulation's trajectories, and monitors the specified attributes. The user handles the simulation object's spatial information, such as its position and speed. Mutual influences among the subsystems and the nonindependent behavior of the subsystems are evaluated from the battlefield simulator. The combat-management simulator has references for specific scenarios. It only uses interface classes. Thus, it is not affected by changes to the scenario. The modeler specifies which model attributes can be changed in

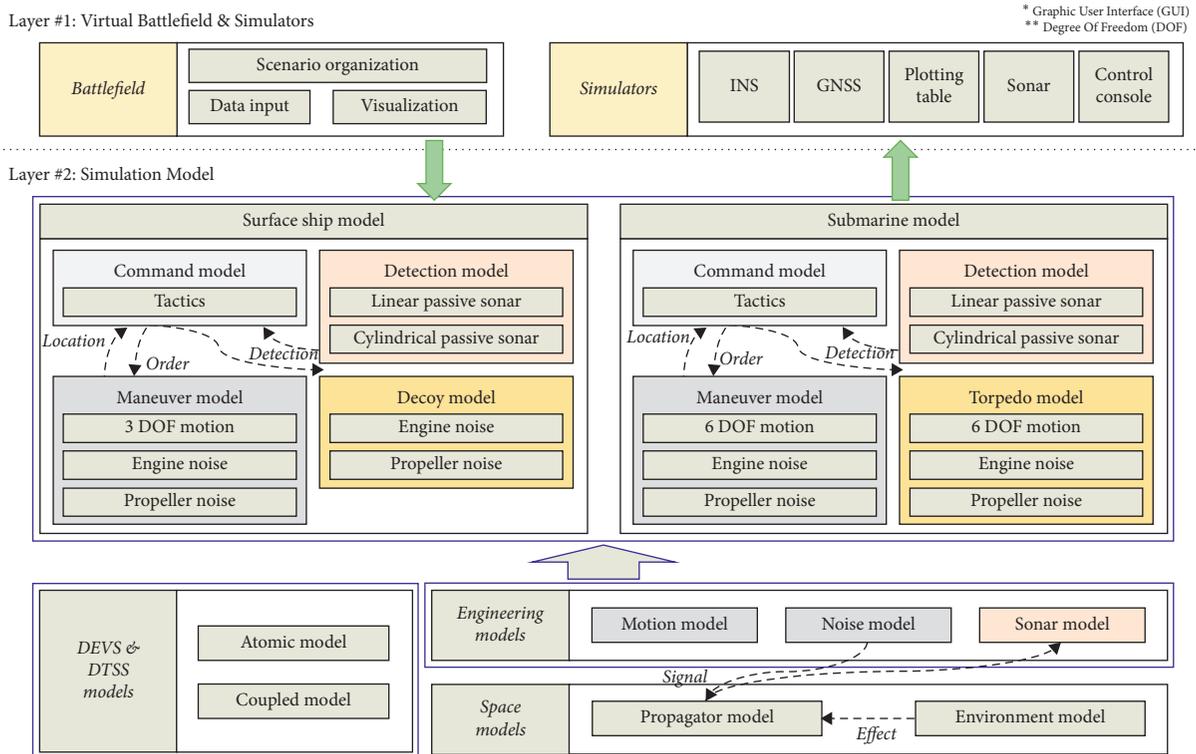


FIGURE 3: Simulation architecture of combat management simulator [41].

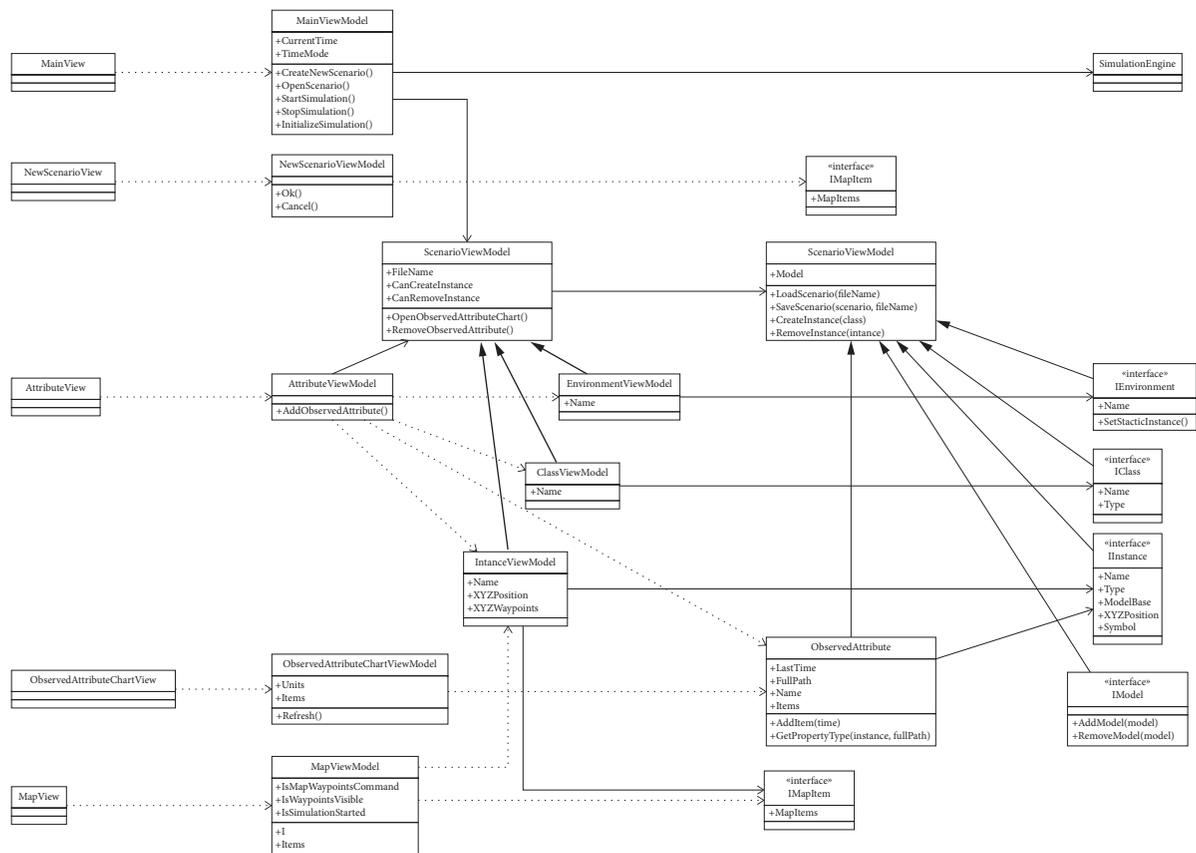


FIGURE 4: Simplified class diagram of the combat management simulator.

the view and specifies values for those attributes. It is desirable to allow a modeler to change the model's structure in view. The software architecture for the battlefield simulator encapsulates (hides from the outside) their implementations (i.e., their internal behaviors) while providing similar interfaces (i.e., access methods).

4.3. Physical Simulators Development. The physical simulators provide physical subsystem aids and data and meteorological information for physical stabilization. Figure 5 shows the underwater sensor simulator in Table 2. Figure 5(a) shows that, firstly, the targets generate noise, and the noise radiates in all directions. The radiated noise propagates far away, undergoing the loss of acoustic energy and addition to background noise. Once the radiated noise arrives at sonars, the sonars analyze the radiated noise. Finally, sonar operators determine whether the noise represents targets (Figure 5(b)).

The sonar equation systematically estimates the expected signal-to-noise ratios for sonar systems. The signal-to-noise ratio determines whether a sonar will detect a signal in the presence of background noise in the ocean. It considers the source level, sound spreading, sound absorption, reflection losses, ambient noise, and receiver characteristics. Such signal-to-noise can be expressed as a passive sonar equation.

$$SE = SL - PL - NL - BW + AG - DT. \quad (1)$$

The first term, source level (SL), is the magnitude of the radiated noise, whose major sources are engines and propellers. The characteristics of the radiated noise vary, depending on the type and operational conditions of the naval ship, which becomes the clue for the classification of targets. Propagation loss (PL) refers to the loss of acoustic energy that the radiated noise undergoes when propagating. Noise level (NL) is the magnitude of the background noise. Bandwidth (BW) and array gain (AG) are related to signal processing. Sonars analyze the radiated noise by signal processing, and the results of the signal processing are the magnitudes of the radiated noise according to bearings and frequencies. Finally, detection threshold (DT) is the standard for determining whether there are targets.

4.4. DIS Simulators Development. The combat system links together various network topologies, treated as subnetworks, in a larger topology. In this case, the DIS is designed to act as a proxy for combat systems residing on the central data bus, which enhances interface compatibility. The DIS simulator verifies that the realized simulated model accurately reflects the authentic behavior of the real system to be tested. The primary role of the DIS simulator is to convert simulation data into a middleware data format.

5. Application: Naval Combat Simulation

This section introduces a practical application to maritime warfare. The developed simulation system was used to test the behaviors of the NCS's subsystems and verify data conversion in a specific subsystem.

5.1. Simulation Scenario. Modern warfare has two conflicting goals, depending on the side: (1) how the attacker can hit targets accurately, and (2) how the defenders can incapacitate the attackers effectively [36]. For example, during a maritime engagement between a submarine and a surface ship, the submarine fires a torpedo, which sends out sounds and seeks the surface target with echoes. In contrast, the surface ship employs countermeasures to deceive the threat, i.e., the torpedo, and implements evasive maneuvering [36].

We assumed underwater warfare in this application: a friendly submarine attacks a hostile surface ship with a torpedo. Simulation models for the submarine were built based on the proposed CPS modeling. The CPS models specify when the submarine would launch a torpedo and how it would control the torpedo to hit the target. The simulation models for the surface ship also contained when and how to launch various types of countermeasures according to the defensive tactics used. The simulation would terminate when the surface ship has gone down or when the torpedo is discharged.

5.2. Simulation Environment. We developed three types of CPS simulators for the submarine. Five physical simulators, as shown in Figure 6, were distributed across four desktops. The CM and DIS simulators were set up on the same desktop as that of the plotting board simulator. They interacted with the developed physical simulators. For example, the CM simulator computed the overall actions of all of the combat objects. The submarine's actions were decided based on the sonar, GPS, INS, ship steering, and plotting board simulators. The DIS simulator transformed simulation data from the INS simulator into real-world data.

The simulators were developed in .NET framework 4.7 using the language C#. We utilized WCF to realize service-oriented communications between them. As a real INS has a serial interface to connect to external systems, the INS and DIS simulators were connected based on RS-232 serial communications. To retain the consistency of the GUI with the MVVM design pattern, we used DevExpress, which provides best-in-class user-interface controls for WinForms, ASP.NET, and WPF [43].

5.3. Simulation Execution. Figures 7–10 show the GUI results of the simulators developed in this application. Figure 7(a) shows a screenshot of the CM simulator, which comprises of five submenus. The main menu (Part I in Figure 7(a)) helps the user to control the overall simulation settings. A user can manage the simulation scenarios, such as by creating, saving, and opening them. When a simulation scenario is chosen, the user constructs the simulation models and configures their variables and parameters via Parts II and III. After completing the configuration, the user can start the simulation and pause, resume, and stop it, if necessary.

Part IV is the main screen to visualize the current simulation situation. It provides all of the trajectories of the combat objects, which are represented with North Atlantic Treaty Organization joint military symbols. In this

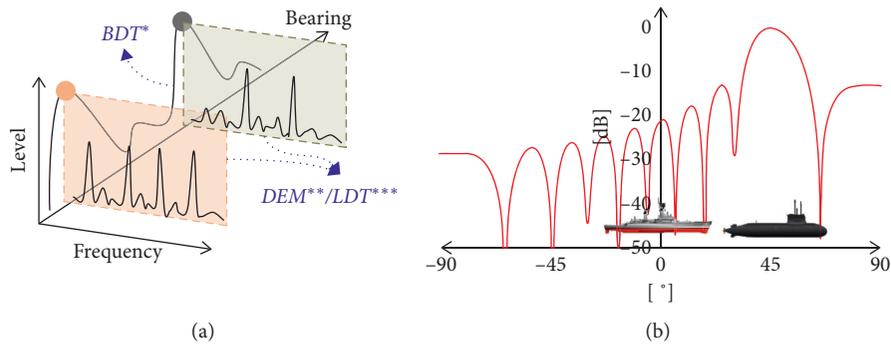


FIGURE 5: Sonar simulator descriptions: (a) analysis of the radiated noise; (b) sonar operation based on radiated noise.

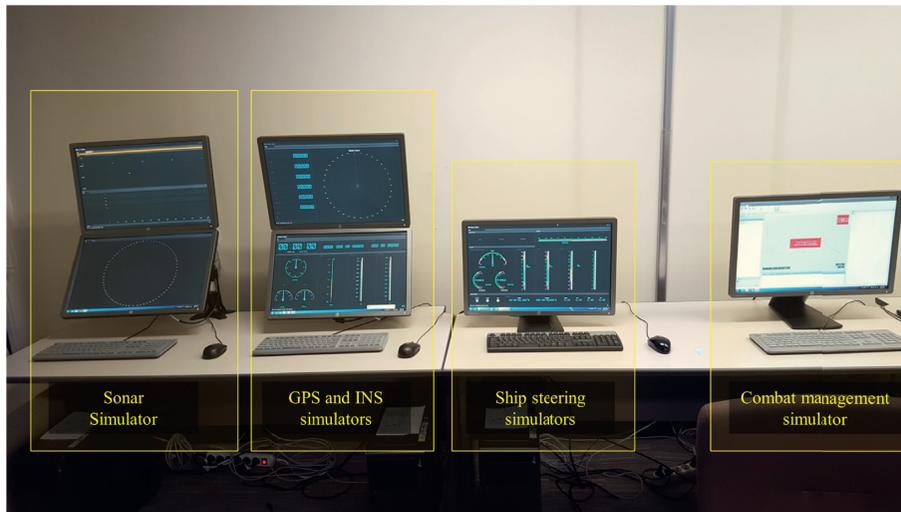


FIGURE 6: Simulation architecture of combat management simulator.

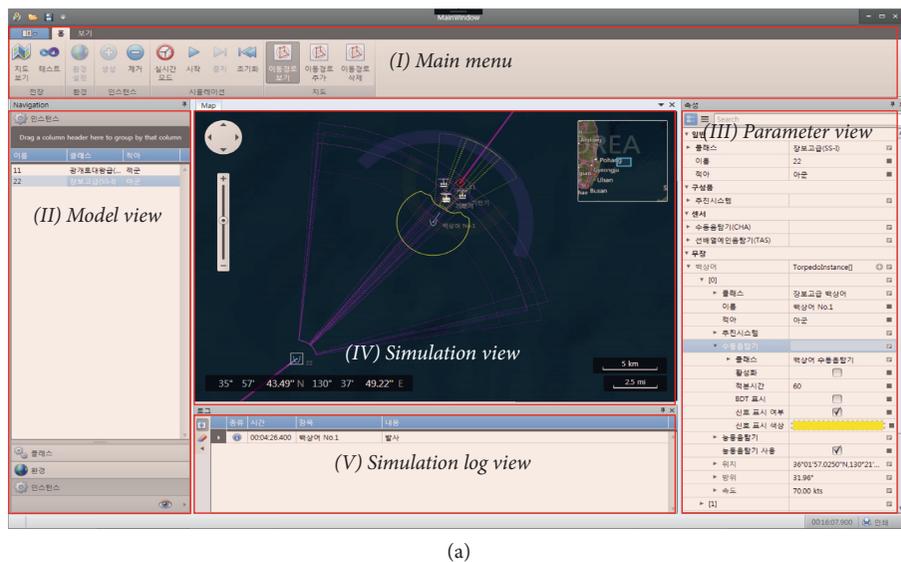
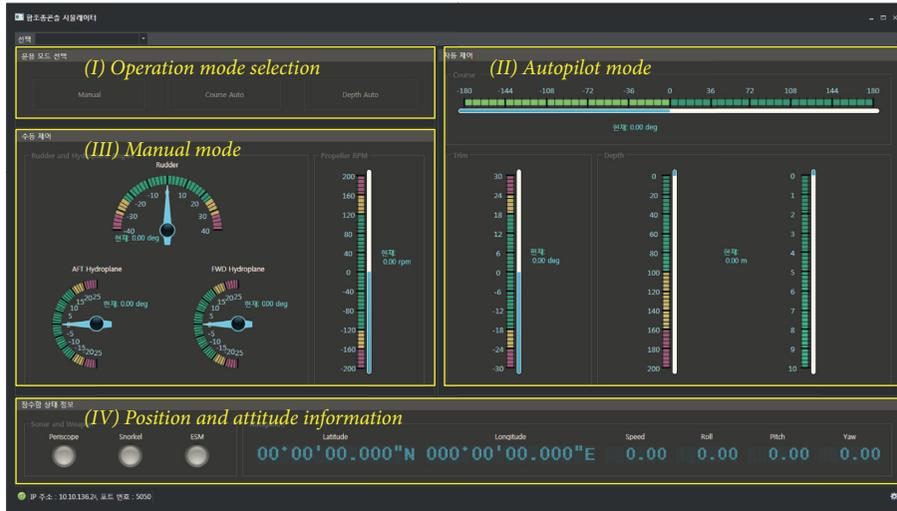
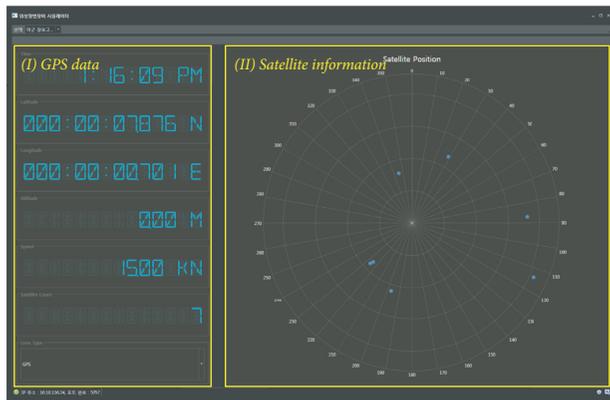


FIGURE 7: Continued.



(b)

FIGURE 7: Graphical user interfaces (GUIs) of the developed simulators: (a) CM simulator; (b) ship steering simulator.

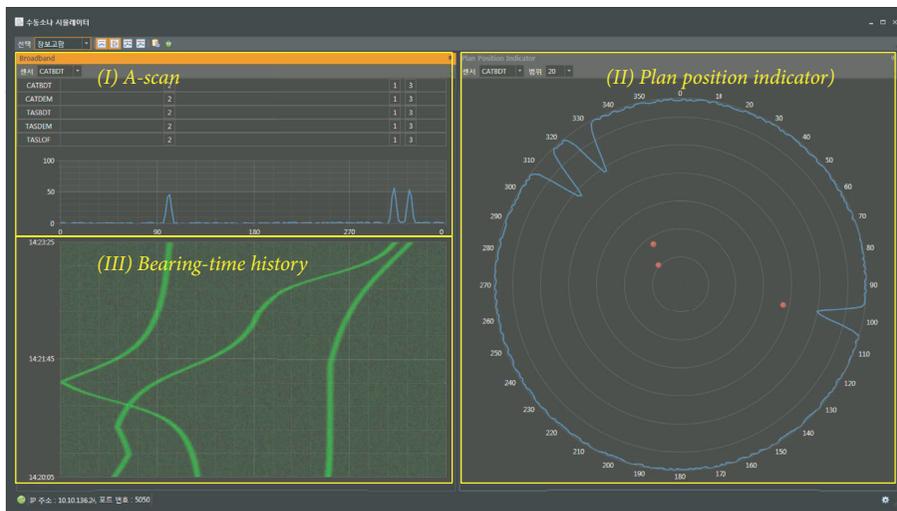


(a)



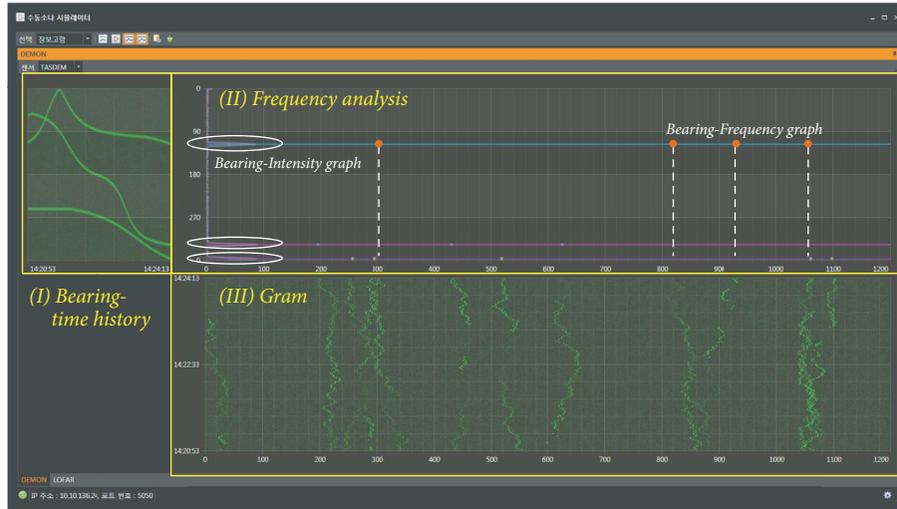
(b)

FIGURE 8: GUIs of the navigation simulators: (a) global positioning system (GPS) simulator; (b) inertial navigation system (INS) simulator.



(a)

FIGURE 9: Continued.



(b)

FIGURE 9: GUIs of the sonar simulator with two views: (a) broadband view; (b) narrowband view.

screenshot, a friendly submarine fires a torpedo at multiple threats. The sensing ranges of the submarine and the torpedo are displayed as purple and yellow fan-shaped sectors, respectively. Part V shows specific events during the simulation, e.g., threat detection, target evaluation, weapon assignment, and weapon launch. As the combat-management simulator provides an integrated situation, the individual behaviors are verified by the physical simulators.

Figure 7(b) is a screenshot of the ship-steering simulator, which shows the six degrees of freedom for the vessel, i.e., latitude, longitude, depth, heading, pitch, and roll. The simulator was developed with autopilot and manual operation modes. The user can select the mode from part I, and the selected mode is displayed and managed via parts II and III. For example, in the manual mode, the user can control a pair of hydroplanes at the forward and afterward sides to change the vessel's horizontal direction and two rudders mounted in the vertical plane to change the lateral movement. The current positional and attitude information is displayed in part IV. All of the views in this simulator are displayed as bar charts.

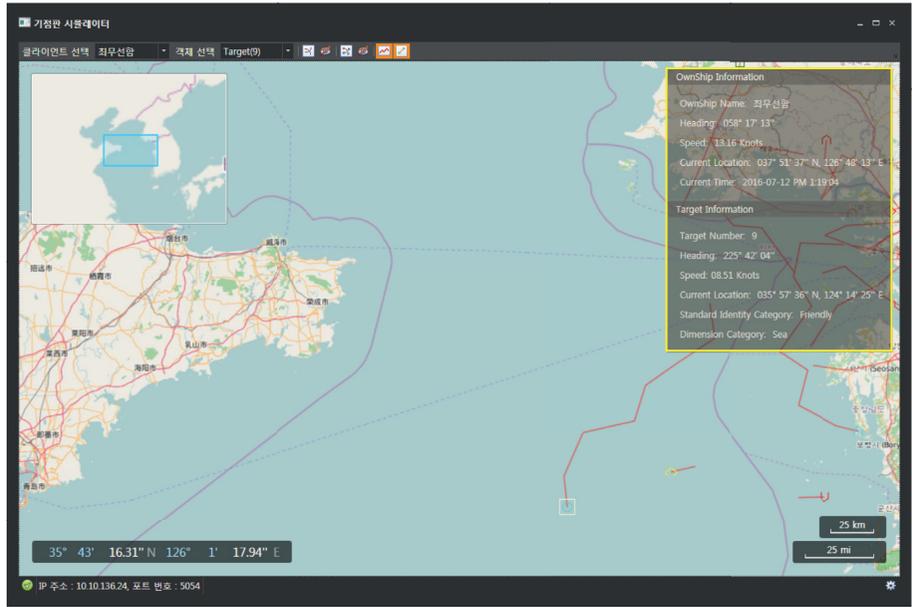
Figure 8 shows the results from the two navigation sensor simulators. The GPS simulator in Figure 8(a) provides positional data based on satellite information. Part I displays the current latitude, longitude, altitude, and speed with text, and part II illustrates the number of satellites needed to calculate the position. In this application, the navigation data are processed with the global marine navigation protocol, which is a combined electrical and data specification for communication between marine navigation equipment [44].

When the vessel is operated underwater, it cannot receive a GPS signal; therefore, the INS mainly is activated, instead of the GPS simulator. In contrast to the GPS simulator, the INS simulator in Figure 8(b) provides textual and graphical information for the vessel's position and altitude (Parts II and III). The INS simulator uses the GPS signal to calibrate navigation inaccuracies when the GPS signal is

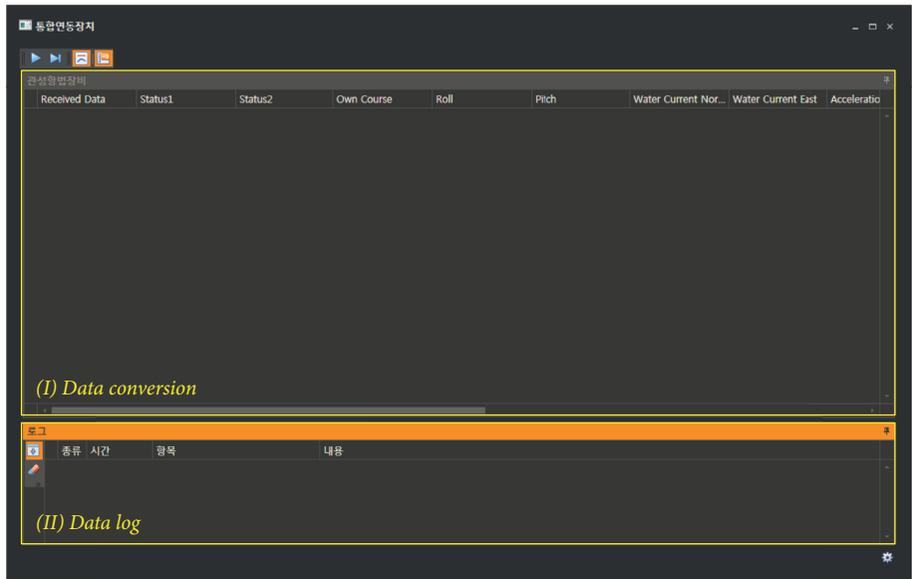
available. Thus, through Part I, the user can choose whether to use the GPS signal while checking the current simulation time.

Figure 9 provides the screenshots of the sonar simulator, which displays all of the undersea-detection information as broadband and narrowband views. The broadband view in Figure 9(a) contains three subviews: the bearing diagram, plan-position indicator, and bearing time history. Firstly, the bearing diagram displays the amplitude envelope curve of the signal according to the directed sensors. If a target is detected in a specific sensor, the target is displayed as a raised section and assigned a specific number. Next, the plan-position indicator view (part II) is a type of radar display using a polar coordinate system. In this view, our vessel is in the center, and the detected targets are marked as red points. The view displays the targets' sound levels with a crushed blue circle. In parts I and II of Figure 9(a), three targets—numbered 1, 2, and 3—were detected at about 100° , 310° , and 330° angles, respectively, with the 310° target having the highest sound level. Finally, the bearing time history (part III), referred to as a waterfall diagram, displays the output of the beamforming processor by time. The newest information is at the top of part III, which is matched with the bearing diagram.

Although the broadband view finds the targets' bearings and sound levels, it does not indicate which types of targets have been detected. The target type is specified by the sounds emitted and can be identified from the narrowband view. Figure 9(b) shows two narrowband views for the frequency analysis (II) and gram (III). The frequency analysis, which contains bearing-intensity and bearing-frequency graphs, provides all of the targets' detected sound frequencies. The frequency signals represent vital information about the targets' outward shapes and operations. Similar to the bearing time history in the broadband view, the gram (part III) in the narrowband view displays several sound beams simultaneously for time-series analysis. As shown in



(a)



(b)

FIGURE 10: GUIs of the developed simulators: (a) plotting board simulator; (b) DIS simulator.

Figure 9(b), twelve beam arrays for three targets were detected and temporarily faded away.

In Figure 10(a), the plotting-board simulator displays various types of information from the platforms currently in operation. The main view shows the overall trajectories of the submarine, surface ship, and torpedoes with different-colored lines. The right part of the main view offers two types of information: (1) numerical values such as latitude, longitude, heading, and speed and (2) status information containing the targets' categories and identification. Finally, Figure 10(b) is a screenshot of the DIS simulator, which provides the real-world data to be converted from simulation data. Thus, it provides increased transparency for the data set, to both the existing and connected systems.

5.4. Discussion. The simulation experiments show functional testing results at the subsystem level. The CM simulator simulates real-world combat scenarios, computes overall activity, and sends simulated data to the relevant physical simulators. The five physical simulators display specific behaviors via the user interface, and the DIS simulator shows the data communication between the simulators.

Here, we discuss two further applications for the developed simulators. First, the simulators were utilized to analyze effectiveness and performance. The ultimate goal of NCS is to maximize fighting and defensive strengths based on optimal control of sensor and weapon systems. The NCS should be optimized, and performance should be evaluated

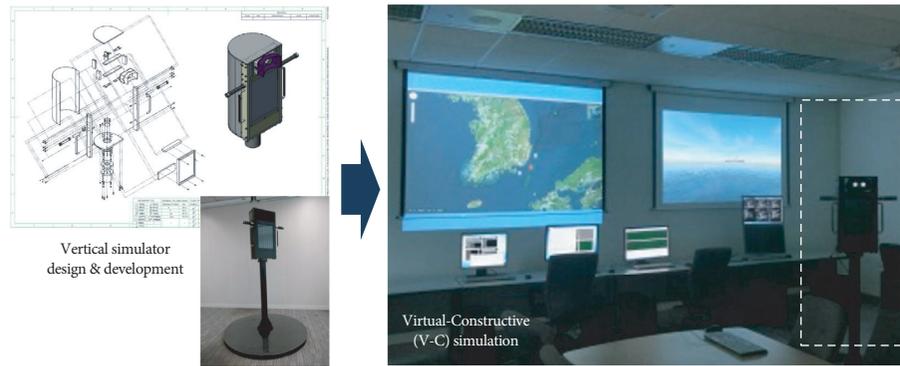


FIGURE 11: Application of virtual-constructive simulation with periscope simulator.

before a real war is conducted, to measure the system's reliability [42]. Such early feedback during the system-design phase improves to the product's quality by allowing a new understanding of emergent behavior. The proposed simulation can reduce the time it takes to introduce new products, as has been very effectively demonstrated in the automotive industry [41].

Next, the present study supports a testing environment for other studies. It is common practice in military simulations to federate systems at different levels so that they can interoperate to provide their functionalities in a new context. MBSE, a paradigm in which the model becomes the actual software, provides not only a safe and efficient testbed but also provides an immersive environment in which CPSs can learn to adapt and optimize their behavior [45–47]. A build-and-test approach is insufficient because the complexity, cost, and design times become more challenging [48].

Figure 11 shows virtual and constructive simulations, a further application for the developed system [49, 50]. We developed a periscope simulator with HCI interaction. The operator, in turn, interacts with the system or performs these activities, linked with the help of specific HCI panels. The DIS simulator in this advanced simulation produces new real-system data from received simulation data, which extends to live-virtual-constructive simulation.

6. Conclusion

In this study, we propose a layered simulation architecture for functional testing that differentiates intrasubsystem behaviors and intersubsystem communications. We designed and modularized three types of simulators. The CM simulator simulates real-world combat scenarios and sends simulated data to the relevant physical simulators. With the simulated data, each physical simulator realizes its dynamics and behaviors by imitating the console panels of its corresponding system. Finally, the DIS simulator performs a gateway role. It links the existing physical models to external CPSs or their models by generating compatible I/O data on both sides.

The proposed *s* increases the automation's transparency, both to the designer and to the user, by separating the domain processing needed to build the system view from the processing. The developed software provides parallel management of the external interfaces to the combat system and

internal interfaces between subsystems within the combat system for simulation testing.

Specifically, we discussed several findings, including the balancing of physical and computational abilities as well as the importance of information technology and the statistical trends between the models' inputs and outputs. Models enable simulation and analysis, which can result in earlier identification of design defects than prototyping can. MBSE is now a critical enabler for managing the complexity when developing complex modern systems like CPSs. We believe that the proposed systems also will allow developers and the research community, in general, to better understand open problems and their impact on the broad applicability of model-based design technologies.

Abbreviations

AG:	Array gain
BW:	Bandwidth
CM:	Combat management
CMS:	Combat management system
CPS:	Cyber-physical system
CSDB:	Combat system data bus
DIS:	Data-integration system
DT:	Detection threshold
GPS:	Global positioning system
GUI:	Graphical user interface
HCI:	Human-computer interaction
NL:	Noise level
I/O:	Input/output
ICS-Sim:	Integrated combat systems simulation
INS:	Inertial navigation system
M&S:	Modeling and simulation
MBSE:	Model-based systems engineering
MVVM:	Model-View-ViewModel
NCS:	Naval combat system
PL:	Propagation loss
SE:	Signal excess
SL:	Source level
WPF:	Windows Presentation Foundation.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by 2020 Research Fund of Myongji University.

References

- [1] Z. Liu, D.-s. Yang, D. Wen, W.-M. Zhang, and W. Mao, "Cyber-physical-social systems for command and control," *IEEE Intelligent Systems*, vol. 26, no. 4, pp. 92–96, 2011.
- [2] V. Gunes, S. Peter, T. Givargis, and F. Vahid, "A survey on concepts, applications, and challenges in cyber-physical systems," *KSII Transactions on Internet & Information Systems*, vol. 8, no. 12, 2014.
- [3] Y. Liu, Y. Peng, B. Wang, S. Yao, and Z. Liu, "Review on cyber-physical systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 1, pp. 27–40, 2017.
- [4] P. Helle, I. Giblett, and P. Levier, "An integrated executable architecture framework for System of Systems development," *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, vol. 10, no. 4, pp. 435–448, 2013.
- [5] B. Ge, K. W. Hipel, K. Yang, and Y. Chen, "A novel executable modeling approach for system-of-systems architecture," *IEEE Systems Journal*, vol. 8, no. 1, pp. 4–13, 2014.
- [6] N. Guertin, B. Womble, P. Bruhns, D. Schmidt, A. Porter, and B. Antypas, "Management strategies for software infrastructure in large-scale cyber-physical systems for the US Navy," *Cutter IT Journal*, vol. 28, no. 5, 2015.
- [7] W. Kang, K. Kapitanova, and S. H. Son, "RDDS: a real-time data distribution service for cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 2, pp. 393–405, 2012.
- [8] S.-C. Shin, J.-G. Shin, and D.-K. Oh, "Development of data analysis tool for combat system integration," *International Journal of Naval Architecture and Ocean Engineering*, vol. 5, no. 1, pp. 147–160, 2013.
- [9] K.-M. Seo and K.-P. Park, "Interface Data Modeling to Detect and Diagnose Intersystem Faults for Designing and Integrating System of Systems," *Complexity*, vol. 2018, Article ID 7081501, 7081501.21 pages, 2018.
- [10] I. Akkaya, P. Derler, S. Emoto, and E. A. Lee, "Systems engineering for industrial cyber-physical systems using aspects," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 997–1012, 2016.
- [11] B. Xiong, B. Li, R. Fan, Q. Zhou, and W. Li, "Modeling and Simulation for Effectiveness Evaluation of Dynamic Discrete Military Supply Chain Networks," *Complexity*, vol. 2017, Article ID 6052037, 6052037.9 pages, 2017.
- [12] P. J. Nicholas and D. L. Alderson, "Fast design of wireless mesh networks to defend against worst-case jamming," *Military Operations Research*, vol. 23, no. 3, pp. 5–20, 2018.
- [13] P. Derler, E. A. Lee, and A. S. Vincentelli, "Modeling cyber-physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 13–28, 2012.
- [14] Y. Gao, J. Cheng, J. Huang, and L. Quan, "Simulation analysis and experiment of variable-displacement asymmetric axial piston pump," *Applied Sciences*, vol. 7, no. 4, p. 328, 2017.
- [15] B. G. Kang, K.-M. Seo, and T. G. Kim, "Model-based design of defense cyber-physical systems to analyze mission effectiveness and network performance," *IEEE Access*, vol. 7, Article ID 42063, 2019.
- [16] C. Sung and T. G. Kim, "Collaborative modeling process for development of domain-specific discrete event simulation systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 532–546, 2012.
- [17] X. Li, Y. Lei, H. Vangheluwe, W. Wang, and Q. Li, "Domain-specific decision modelling and statistical analysis for combat system effectiveness simulation," *Journal of Statistical Computation and Simulation*, vol. 84, no. 6, pp. 1261–1279, 2014.
- [18] Z. Zhu, Y. Lei, A. Alshareef, H. Sarjoughian, and Y. Zhu, "Domain specific MetaModeling for deep semantic composability," *IEEE Access*, vol. 6, Article ID 18276, 2018.
- [19] S. C. Park, Y. Kwon, K. Seong, and J. Pyun, "Simulation framework for small scale engagement," *Computers & Industrial Engineering*, vol. 59, no. 3, pp. 463–472, 2010.
- [20] E. Squarcia, V. Gaeta, and F. Ciambra, "4.2. 1 Multi-scale performance evaluation in Naval Combat Systems domain," in *Proceedings of the INCOSE International Symposium*, Wiley Online Library, Chicago, Illinois, USA, July 2010.
- [21] X. Xu, G. Li, M. Yang, R. Ju, and K. Huang, "Towards the integration of engagement and engineering-level simulation over real-time and heterogeneous systems," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 12, pp. 3390–3408, 2016.
- [22] H.-H. Cheng, Q. Liu, J. Liu, D. Wang, J. Zhou, and J. Li, "Design of key technologies for simulation training system of the multi-element observation system," in *Proceedings of the International Conference on Computer, Electronics and Communication Engineering*, Sanya, China, June 2017.
- [23] X. Li, D. Chang, H. Pen, X. Zhang, Y. Liu, and Y. Yao, "Application of MVVM design pattern in MES," in *Proceedings of the Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, 2015 IEEE International Conference on, June 2015.
- [24] A. Weil, *Learn WPF MVVM-XAML, C# and the MVVM Pattern*, Lulu.com, North Carolina, NC, USA, 2016.
- [25] F. Jarnjak, "Flexible GUI in robotics applications using windows presentation foundation framework and model view ViewModel pattern," in *Proceedings of the 4th International Conference on New Trends in Information Science and Service Science*, Gyeongju, Korea (South), May 2010.
- [26] D. Shan, Dong, Jun et al., "Application OF contemporary programming techniques IN health monitoring software for suspension bridges," *NED University Journal of Research*, vol. 12, no. 2, 2015.
- [27] E. Salvador, S. Tidon, and B. E. V. Comendador, "A configurable real-time auto-locked software for electronics company (RT-ASEC)," *Journal of Software*, vol. 10, no. 10, pp. 1170–1175, 2015.
- [28] A. K. K, J. George, J. George, and V. P. Jagathyraj, "Requirements analysis of an integrated sonar suite for surface ships: systems engineering perspective," *Defence Science Journal*, vol. 64, no. 4, pp. 366–370, 2014.
- [29] T. Kim, D. Lee, and S. Lee, "Hierarchical multilevel on/off source traffic modeling for a warship combat system," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 2, pp. 226–242, 2014.
- [30] H. F. R. Arciszewski, T. E. De Greef, and J. H. Van Delft, "Adaptive automation in a naval combat management system," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 39, no. 6, pp. 1188–1199, 2009.
- [31] K.-M. Seo, K.-P. Park, and B.-J. Lee, "Achieving data interoperability of communication interfaces for combat system engineering," *IEEE Access*, vol. 5, Article ID 17938, 2017.

- [32] M. R. Khaefi, J.-Y. Im, and D.-S. Kim, "An efficient DDS node discovery scheme for naval combat system," in *Proceedings of the Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference on*, September 2015.
- [33] M. L. Butterfield, J. S. Pearlman, and S. C. Vickroy, "A system-of-systems engineering GEOSS: architectural approach," *IEEE Systems Journal*, vol. 2, no. 3, pp. 321–332, 2008.
- [34] I. Stojmenovic, "Machine-to-machine communications with in-network data aggregation, processing, and actuation for large-scale cyber-physical systems," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 122–128, 2014.
- [35] A. Engel, *Verification, validation and testing of engineered systems*, John Wiley & Sons, Hoboken, NJ, USA, 2010.
- [36] J. H. Kim, K. M. Seo, T. E. Lee, and B. W. Choi, "Achieving new insights into combat engagement analysis via simulation-based sequential experimentation," *Military Operations Research*, vol. 23, no. 4, pp. 51–80, 2018.
- [37] K.-M. Seo, W. Hong, and T. G. Kim, "Enhancing model composability and reusability for entity-level combat simulation: a conceptual modeling approach," *SIMULATION*, vol. 93, no. 10, pp. 825–840, 2017.
- [38] K.-M. Seo, C. Choi, T. G. Kim, and J. H. Kim, "DEVS-based combat modeling for engagement-level simulation," *SIMULATION*, vol. 90, no. 7, pp. 759–781, 2014.
- [39] S. H. Choi, K.-M. Seo, and T. G. Kim, "Accelerated simulation of discrete event dynamic systems via a multi-fidelity modeling framework," *Applied Sciences*, vol. 7, no. 10, 1056.1 pages, 2017.
- [40] B. G. Kang, K.-M. Seo, and T. G. Kim, "Communication analysis of network-centric warfare via transformation of system of systems model into integrated system model using neural network," *Complexity*, vol. 2018, Article ID 6201356, 6201356.16 pages, 2018.
- [41] D.-H. Jeong, M.-I. Roh, S.-H. Ham, and C.-Y. Lee, "Performance analyses of naval ships based on engineering level of simulation at the initial design stage," *International Journal of Naval Architecture and Ocean Engineering*, vol. 9, no. 4, pp. 446–459, 2017.
- [42] T. J. Paciencia, D. J. Richmond, J. J. Schumacher, and W. L. Troy, "A framework and system for theater air attrition modeling," *Military Operations Research*, vol. 23, no. 2, pp. 41–60, 2018.
- [43] P. T. Kimmel, *Professional DevExpress ASP. NET Controls*, John Wiley & Sons, Hoboken, NJ, USA, 2010.
- [44] B. Park, J. Lee, Y. Kim, H. Yun, and C. Kee, "DGPS enhancement to GPS NMEA output data: DGPS by correction projection to position-domain," *Journal of Navigation*, vol. 66, no. 2, pp. 249–264, 2013.
- [45] B. Zeigler, S. Mittal, and M. Traore, "MBSE with/out simulation: state of the art and way forward," *Systems*, vol. 6, no. 4, p. 40, 2018.
- [46] S. Mittal and J. L. R. Martin, *Netcentric System of Systems Engineering with DEVS Unified Process*, CRC Press, Florida, FD, USA, 2016.
- [47] H. Lu and H. You, "Roadmap modeling and assessment approach for defense technology system of systems," *Applied Sciences*, vol. 8, no. 6, p. 908, 2018.
- [48] T. A. Bapty, J. Scott, S. Neema, and R. Owens, "Integrated modeling and simulation for cyberphysical systems extending multi-domain M&S to the design community," in *Proceedings of the Symposium on Model-Driven Approaches for Simulation Engineering*, Virginia, VA, USA, April 2017.
- [49] W. I. Jung, L. Lowe, L. Rabelo, G. Lee, and O. Kwon, "A methodology on guiding effectiveness-focused training of the weapon operator using big data and VC simulations," *SAE International Journal of Aerospace*, vol. 10, no. 2, pp. 57–67, 2017.
- [50] Q. Qi and F. Tao, "Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison," *IEEE Access*, vol. 6, pp. 3585–3593, 2018.