Development of Hybrid Anti-Submarine Weapon Training Simulator
Using Component-Based Development Methodology

Yeong-cheol Kim
Changbum Ahn
PGM Technology Research Institute
Agency for Defense Development
P.O.Box 35-1, Yuseong, Daejon
305-600, Republic of Korea
+82-42-821-2463, +82-42-821-2693
ghkyc@add.re.kr, cbahn@add.re.kr

Sang Tae Lee
Dae Young Kim
M&S Center
LIGNex1 Co.,Ltd.
102-18, Mabuk-dong, Giheung-gu, Yongin-City,
Gyeonggi-do, Republic of Korea
+82-31-288-1207, +82-31-288-
lst0564@lignex1.com, dykim@lignex1.com

Keywords:
Training Simulator, Hybrid Training Architecture, Anti-Submarine Weapon, Component-Based Design

ABSTRACT: The role of modeling, simulation and analysis (MS&A) in defense systems development is continuously evolving as simulation-based design, simulation-based acquisition, and simulation-based training in Republic of Korea. MS&A is now considered as mandatory activity in the areas of the development of operational concept for the newly developed weapon systems, requirements analysis and verification, system simulations for performance evaluation, system and component design by simulation, test and evaluation as simulation, and the development of training systems throughout the life cycle of defense R&D program. In this paper, design and implementation of operator procedure training simulator for Korean vertical-launch anti-submarine missile systems are discussed. The training simulator is designed to have not only simulated equipments but also real-world fire control equipments in order to increase training fidelity by the requirements from Republic of Korean Navy. A hybrid configuration, which has simulation architecture and real-world fire control system architecture, requires unique design of software framework since two separate architectures should be integrated into a system. Systems engineering efforts which are made to develop software framework are also presented.
1. Introduction

The use of Modeling, Simulation and Analysis (MS&A) in defense systems acquisition is now being embraced by the Republic of Korea acquisition community for simulation-based design, simulation-based acquisition, and simulation-based training and experimentation. The development and use of Modeling and Simulation (M&S) are now considered as mandatory activities for the life cycle support of defense system acquisition such as in support of the development of operational concepts for future weapon systems, requirements analysis and verification, system and component design, performance evaluation, test and evaluation, and the development of training systems. It is believed that M&S provide cost and time-saving engineering methodologies, which has been proved to be true from practices in the Korean Vertical Launch Anti-Submarine Missile Systems (KVLA) development program.

KVLA is a set of anti-submarine missile and standard vertical launching system developed by Agency for Defense Development in Republic of Korea. KVLA provides long-range anti-submarine capabilities for destroyer class surface ships. In the beginning of KVLA program, the need for various M&S systems was proposed by both research and end-user groups. M&S systems required in KVLA program is shown in Table 1. Each system was developed and utilized to perform designated role.

<table>
<thead>
<tr>
<th>Required System</th>
<th>Role of M&amp;S System</th>
</tr>
</thead>
<tbody>
<tr>
<td>KVLA System</td>
<td>Develop and analyze operational concept and requirements</td>
</tr>
<tr>
<td>Simulator</td>
<td>Develop and validate missile guidance and control software</td>
</tr>
<tr>
<td></td>
<td>Support KVLA HILS &amp; MILS</td>
</tr>
<tr>
<td>FCS Development</td>
<td>Develop KVLA Fire Control System</td>
</tr>
<tr>
<td>Bench</td>
<td>Provides FCS configuration management</td>
</tr>
<tr>
<td></td>
<td>Support maintenance phase</td>
</tr>
<tr>
<td>ASW Mission</td>
<td>Develop KVLA tactics for KDX-II and KDX-III class destroyer</td>
</tr>
<tr>
<td>Planning Systems</td>
<td>Shipboard mission planning of KVLA</td>
</tr>
<tr>
<td>Training Simulator</td>
<td>Procedure training for KVLA FCS, and launcher operator</td>
</tr>
<tr>
<td></td>
<td>Team work training for KVLA, and C2 operator</td>
</tr>
</tbody>
</table>

Especially, much interest on the development of training simulator was given by Republic of Korea Navy. A great variety of user requirements was discussed in the beginning of the program. Among them, high training fidelity with the use of real-world equipment was a most difficult issue to resolve since it could cause several technical considerations.

<table>
<thead>
<tr>
<th>Table 2. Real-World Equip. vs. Simulated Equip.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of Equipment</td>
</tr>
<tr>
<td>Real-World Equipment</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Simulated Equipment</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

It might be always troublesome decision to apply real-world equipment or simulated equipment when the configuration of training simulator is discussed. In case of applying real-world equipment, training fidelity could easily acquired, whereas rising cost burden caused by purchasing high-priced component could be a serious concern to acquisition manager. Furthermore, several functionalities such as simulation control or replay could be hardly implemented when tactical computer program is installed without any modification. If these functionalities could be considered in the early stage of the development of tactical computer program, restriction on simulation control could be resolved in part.

On the other hand, training fidelity could be relatively decreased when simulated equipment is configured for training simulator. This issue could be resolved if the functionalities of real-world tactical computer program could be appropriately captured. However, risk on the unexpected efforts induced by complexity of tactical computer program should be taken by program manager. Advantage of using simulated equipment could be powerful simulation control and relatively low development cost. Functionalities of simulation control could be easily implemented with the use of simulated computer program.

These aspects should be thoroughly traded-off by systems engineer in case of the assessment of training configuration. Recently, hybrid training configuration, which real-world equipment is in case of important component of training simulator, while the rest is
configured with simulated equipment, is commonly required by military end-user group.

In this paper, hybrid training architecture utilizing common M&S resources developed by earlier M&S activities prior to the development training simulator is presented.

2. Component-Based Development

2.1 Component-Based Development

Component-Based Development (CBD), also known as Component-Based Software Engineering (CBSE), is a branch of software engineering methodologies, the priority of which is the separation of concerns in respect of the wide-ranging functionality available throughout a given software system. This practice aims to bring about an equally wide-ranging degree of benefits in both the short-term and the long-term for the software itself and the organization that sponsors it.

Components are considered to be part of the starting platform for service orientation throughout software engineering, for example Web Services, and more recently, Service-Oriented Architecture (SOA) - whereby a component is converted into a service and subsequently inherits further characteristics beyond that of an ordinary component. Another important attribute of components is that they are substitutable, so that a component could be replaced by another (at design time or run-time), if the requirements of the initial component (expressed via the interfaces) are met by the successor component. Consequently, components can be replaced with either an updated version or an alternative for example, without breaking the system in which the component operates.

As a general rule of thumb for engineers substituting components, component B can immediately replace component A, if component B provides at least what component A provided, and uses no more than what component A used. Software components often take the form of objects or collections of objects, in some binary or textual form, adhering to some interface description language so that the component may exist autonomously from other components in a computer. When a component is to be accessed or shared across execution contexts or network links, techniques such as serialization or marshalling are often employed to deliver the component to its destination.

Reusability is an important characteristic of a high quality software component. A software component should be designed and implemented so that it can be reused in many different programs. It takes significant effort and awareness to write a software component that is effectively reusable.

The component needs to be:
- fully documented
- thoroughly tested
- designed with an awareness that it will be put to unforeseen uses

In the 1960s, scientific subroutine libraries were built that were reusable in a broad array of engineering and scientific applications. Though these subroutine libraries reused well-defined algorithms in an effective manner, they had a limited domain of application. Commercial sites routinely created application programs from reusable modules written in Assembler, COBOL, PL/1 and other 2nd and third generation languages using both System and user application libraries.

Today, modern reusable components encapsulate both data structures and the algorithms that are applied to the data structures. It builds on prior theories of software objects, software architectures, software frameworks and software design patterns, and the extensive theory of object-oriented programming and the object oriented design of all these. It claims that software components, like the idea of hardware components, used for example in telecommunications, can ultimately be made interchangeable and reliable.

Development strategy based on component-based development methodology according to M&S master plan has been established. M&S framework as an infrastructure to ensure reusability and extensibility was developed in the early stage of KVLA program.
2.2 M&S Framework

In general, software framework could be defined as collections of specific classes and services which could be reused. It also provides logical architecture for the applications. An object-oriented modeling methodology was applied in order to identify common components required essentially to develop Anti-submarine Warfare (ASW) M&S applications. Functional components which were captured by use case analysis were re-arranged and implemented in a detailed design of framework.

![Diagram of software design process]

**Figure 1. OOM-based Software Design Process**

Object-Oriented Modeling, or OOM, is a modeling paradigm mainly used in computer programming. Prior to the rise of OOM, the dominant paradigm was functional programming, which emphasized the use of discreet reusable code blocks that could stand on their own, take variables, perform a function on them, and return values. The Object-Oriented paradigm assists the programmer to address the complexity of a problem domain by considering the problem not as a set of functions that can be performed but primarily as a set of related, interacting Objects. The modeling task then is specifying, for a specific context, those objects (or the class the objects belongs to), their respective set of properties and methods, shared by all objects members of the class. The description of these objects is called as a schema. The model description or schema may grow in complexity to require a notation. Many notations have been proposed, based on different paradigms, diverged, and converged in a more popular one known as UML.

OOM methodology could be also applied to an object-oriented software design process, which defines analysis – abstraction - modeling process by implementing a model in software domain from real-world model. Such a model describes the requirements for the software system to be realized and forms an abstraction in two ways shown as Figure 1. First, it abstracts from real world details which are not relevant for the intended software system. Second, it also abstracts from the implementation details and hence precedes the actual implementation in a programming language.

In a design of software model derived from abstraction of real-world, design pattern could be introduced as a best practice defining good solutions. A design pattern is a general reusable solution to a commonly occurring problem in software design. A design pattern is not a finished design that can be transformed directly into code. It is a description or template for how to solve a problem that can be used in many different situations. Object-oriented design patterns typically show relationships and interactions between classes or objects, without specifying the final application classes or objects that are involved. Algorithms are not thought of as design patterns, since they solve computational problems rather than design problems. Design patterns usually can speed up the development process by providing tested, proven development paradigms. Effective software design requires considering issues that may not become visible until later in the implementation. Reusing design patterns helps to prevent subtle issues that can cause major problems, and it also improves code readability for coders and architects who are familiar with the patterns. Design patterns originally grouped design patterns into the categories Creational Patterns, Structural Patterns, and Behavioral Patterns, and described them using the concepts of delegation, aggregation, and consultation. Classification of design patterns are shown in Figure 2.

![Classification of design patterns]

**Figure 2. Classification of Design Patterns**
A proposed M&S framework was designed to have three sub-frameworks; Infrastructure, Core, Accessory. The functional architecture of framework is shown in Figure 3.

Core sub-framework is composed of four functional blocks, which are presentation, simulation, model, and scenario. The presentation block which is in charge of visualization of data processed from model block supports 2-dimensional, 3-dimensional display and user control. It is possible to conduct actions, and transform state of model with physical and environmental model blocks. They provide abstract class to application developer. Model could be designed and maintained with consistency since every physical and environmental model should be implemented by inheritance from abstract classes defined in model block. Timer in simulation block provides not only wall clock time in real-time simulation, but also logical clock in logical simulation such as Monte-Carlo simulation.

Infrastructure sub-framework provides data access, communication middleware, and translator which could be referred as communication basis. Communication middleware could be defined as a programming service which has a function to arbitrate between two or groups of processes. Data access and communication middleware were designed to provide common interface functions to application developers independent of network protocols, and data types. ACE which is an open source framework was used to build platform-independent, protocol-independent programming services.
3. Proposed System Architecture

A proposed M&S framework has been released to software development groups in the form of both physical and logical packages. In the physical aspect, software framework has a form of re-distributable package in order to provide common libraries and core functions by successfully integrated into the development environments. The quality of software products based on a software framework could be maintained with the help of design patterns which provide standards of programming as a best practice. Application architecture and various types of abstracted model are provided for the purpose of supporting system design task in the logical aspect. Primitive data types and base classes are also offered to populate common data format throughout the KVLA M&S applications.

Highly reusable and reconfigurable simulation architecture for integrated M&S system was designed to support KVLA M&S activities throughout the development lifecycle. KVLA system simulator was developed as a multi-purpose M&S platform in the simulation architecture to meet various needs for M&S; system analysis supports for requirement definition, configuration management, and system design verification, Test-bed for firing control system (FCS) validation and verification, KVLA system T&E (test and evaluation) supports and combat systems integration supports, tactics development and mission planning supports and operational supports as a training simulator. Valuable resources including M&S framework, communication middleware and M&S model architecture have been successfully deployed to application developers to provide an integrated development environment in software engineering domain.

Real-world C2 network was used without modification in the network architecture in order that the interface of KVLA and SLTS (Ship-Launched Torpedo System) should be tested, and verified. For the purpose of extensibility of system simulator to training simulator or land-based T&E support system, simulation network is added in the architecture. SLTS control panel and its launcher systems were also configured in order to simulate shipboard anti-submarine weapons, which are KVLA and SLTS, at the same time. The application architecture of KVLA system simulator was designed to have three independent layers based on M&S framework. All of simulation applications should have three blocks of layers composed of user interface, model, and network interface. This 3-tier architecture guaranteed the minimum change of software blocks in the case of the modification of user interfaces and network interface.

![Figure 7. Simulation Architecture of KVLA M&S](image)
3.1 Hybrid Training Architecture

KVLA training simulator was designed to have hybrid configuration, which has both real-world FCS equipments and simulated equipments in order to meet the requirement of Republic of Korea Navy. This hybrid training architecture requires design of software framework since two separate architectures should be integrated into a system. Newly configured training equipments were designed to be integrated into a training architecture on a basis of KVLA M&S architecture acquired from the previous research. A hybrid training architecture has four unique layers; training control layer, arbitrator layer, real-world equipment layer, and simulated equipment layer.

Design of training control computer (TCC) in training control layer is inherited from simulation controller of KVLA system simulator. A great amount of software requirements is re-used except for newly defined training-related requirements.

Arbitrator layer plays an important role in hybrid training architecture since all of data populated from two different networks should be managed with combat system simulator (CSS) and system link computer (SLC). CSS was designed to be positioned in the middle of simulation and C2 network to arbitrate between two networks. The information reported from KVLA and SLTS is synthesized in CSS in order to transmit it to TCC since CSS is an only route from KVLA and SLTS to simulation network. SLC was designed to efficiently control simulated equipments newly configured for training simulator. Modification of simulation architecture could be minimized with the introduction of SLC in a design of hybrid training architecture.

Three real-world equipments, which are KVLA missile control console (KMCC), vertical launch control console, and ship-launched torpedo control panel (SLTCP), was used without any modification in order to be used for training simulator. An embedded training facility was designed to be activated in a training environment.

![Figure 8. Hybrid Training Architecture](image-url)
3.2 Software Architecture

3.2.1 Training Control Computer

Training Control Computer (TCC) performs the function of editing simulation scenario, controlling entire simulation phase, monitoring and recording simulation data. TCC was designed to have four functional blocks of process which are Front-end, logger, EDMS and back-end as shown in Figure 9. Front-end of TCC has four functional components; simulation control, scenario manager, communication middleware module. First, Scenario manager module generates the scenario file. Generated scenario file has an extension XML standard. Simulation control module controls the entire behavior of simulation including individual states. TCC EDMS has object modeling, communication middleware module. Object modeling module is simply modeling submarine object, anti-submarine object, and missile object. In addition to add other object model easily. TCC Logger has Analysis, logging, communication middleware module. All of applications were designed to send their internal logging data to TCC logger after simulation. In case of real-world KMCC software, network drive was used in order to acquire internal data recording since tactical software was used without any modifications. Analysis module in TCC logger synthesizes data by various way of analysis. TCC back-end has a function to send simulation control command, and receive simulation status from other applications through message distributor. If TCC front-end has to send one message to four equipments, TCC front-end just needs to send it to message distributor with routing table.

3.2.2 Combat System Simulator

Combat System Simulator (CSS) has a 3-tier based structural design for multi-purpose use. One of the most important aspects of 3-tier architecture is that system with multiple layers could be fractionalized structurally. This advantage could be taken from software architect in a design phase in order to define unique role of each layer, and also from software developer in implementation or maintenance phase since it is easy to make modification of interface between layer, or modification of functions in each layer with no impacts on interface.

A high-level architecture of CSS is designed to have three layers as shown in Figure 10 which are human-computer interface (HCI), model, and communication layers. Communication middleware in communication layer which could be referred as infrastructure of software is designed to conduct not only all of message transmission between UDP/TCP data from network and model layer but also scheduling and management of data transmission. Model layer has a function to process every data from HCI and communication layer. Actions created by CSS operator, and internal states processed by model layer are displayed in HCI layers with proper graphical user interfaces.

Middleware interface manager which exists between model and HCI layer distribute all of messages coming from other equipments through communication middleware. HCI interface manager between HCI and model layer performs a role of transmitting information processed in model layer to HCI layer. Operator inputs and HCI information are also transmitted through HCI interface manager. This design enables independency of the model on HCI, and communication layer.
Graphical user interface consists of Labeled Panel Display (LPD), Soft Key Panel (SKP), and Tote. These are managed by HCI manager. Graphical user interface is shown in Figure 11. Safe queue was designed to increase the reliability of communication between layers. Deadlock during communication could be prevented by the use of safe queue since each process should not send any data directly to other process. Data transmission between processes could be managed by safe queue. Track, and target management model are located in the model layer. Engagement models for KVLA/SLTS have increased their fidelity throughout the lifecycle of KVLA program. Communication and data access middleware exist in communication layer as a data communication service. Functional architecture of CSS is shown in Figure 11.

3.2.3 System Link Computer

System Link Computer (SLC) performs the function of merging I/O simulation data and real data. Control panel module is controlled the switch operation and display ramp by state monitoring panel simulator, remote control panel simulator, cell safety enable switch simulator.

The System link computer (SLC) is connected to the real-world Vertical Launch Control Console (VLCC) by LAN. In Connected it uses Dummy Hub. It is not directed to SLC and KMCC. A KVLA missile control console (KMCC) and VLCC have communicated with LAN. The training simulator needs to sending Engagement message and receiving Engagement message from the real-equipments. Accordingly the system link computer catches to sending/receiving Engagement messages from the real-world equipments. It used the packet capture method. The method is WinPcap. It is open library. WinPcap is the industry-standard tool for link-layer network access in Windows environments: it allows applications to capture and transmit network packets bypassing the protocol stack, and has additional useful features, including kernel-level packet filtering, a network statistics engine and support for remote packet capture.

The simulation network is connected to training control computer, anti submarine warfare combat system simulator, state monitoring panel simulator, remote control panel simulator, cell safety enable switch simulator. The training simulator is communicated LAN (TCP/IP) with using the common HUB. The real tactical network is connected to KVLA missile control console, vertical launch control console, ship launched torpedo system. The real-world equipments are communicated LAN (UDP) with using the Dummy HUB.

![Figure 11. Functional Architecture of CSS](image)

![Figure 12. Functional Architecture of SLC](image)

![Figure 13. Network Connectivity Diagram](image)
4. Implementation of Training Simulator

The training simulator consists of several simulated equipments and real-world equipments. The real-world equipments are consisted of the KMCC, VLCC, and SLTS. The simulated equipments have TCC, CCS, SMP, RCP, and CSES. The linked equipment has SLC.

A training control computer has several of functions. System Control function controls Simulation equipments totally that remote power on/off, check connected simulation equipments status. Train Scenario is generated to scenario file and do delete, edit, save, open. Operation control is distributed to simulation equipment to scenario files that control simulation start/stop/resume. And it is the correspondence of the real-world equipment mode and the simulation equipment mode. Object simulation is moved to submarine object, anti-submarine object, and missile object. Situation observation is managed by generated the train event in training, operation time, 2D Map and object list. Training result is saved by the engagement event that displayed missile object event and printed the train result. GUI emulator education is not operated the real-world equipment. Accordingly GUI emulator education uses equipment switch control, displays ramps.

Combat system simulator function is simulated by anti-submarine warfare. CSS is managed by track and target. System link computer function controls I/O simulation data and merges real data and simulation data. The real-world equipment is expensive. And it is difficult for the test, using real-world equipment. The Simulator equipments price is cheaper than the real-world equipments price. Therefore, the Simulator equipments are made in SMP, RCP, and CSES. The simulator equipments control switches operation and display ramp. The KVLA Missile control console controls target engagement. The vertical launch control console controls launch procedure. Ship launched torpedo system controls torpedo engagement.

4.1 Training Control Computer

Multiple windows with graphical user interfaces were designed to give the right of control, monitor, and analysis to TCC operator. Tactical situation view and Engagement information view are shown in Figure 15 and Figure 16.

Figure 14. 3-Dimensional View of KVLA Training Simulator
TCC includes tactical view and Engagement information view. The tactical view is displayed by received train scenario from training control computer. And display scenario information. Engagement view is displayed by system states and KVLA/SLTS engagement Information. Once simulation has been stated, simulation data such as equipment status, engagement status, and object data simulated from EDMS.

4.2 Combat System Simulator

Combat systems simulator (CSS) was designed to be positioned in the middle of simulation and C2 network to arbitrate between two networks. The information reported from KVLA and SLTS is synthesized in CSS in order to transmit it to simulation controller because CSS is an only route from KVLA and SLTS to simulation network. However, the data displayed in simulation controller is a summary of status which is processed in CSS. Internal data recording facilities equipped in KVLA and SLTS equipments are used for further analysis. System configuration using two kinds of isolated networks is thought to be a best solution to maximize system extensibility.

4.3 System Link Computer

Training control computer (TCC) sends the training message to system link computer. The system link computer received the training message; the vertical launch control console resends the training message. A system link computer changed the training messages format to the real-world message formats. The real-world message changed format frequently. And it is difficult of testing the real-world equipment. Besides the real-world software does not change interface for the training software. Because of real-world software tests the training simulator software same development time. It changed SLC application development method.
The system link computer (SLC) captures the real-world messages interface with the vertical launch control console and Korea missile control console. A system link computer does not connect the vertical launch control console directly. Accordingly the system link computer is partially restricting by the real-world message format. Besides the real-world software does not need to change interface for the training simulator software. That reason it changed the system link computer software.

5. Conclusion

A wide variety of M&S systems are required throughout the lifecycle of the program especially for the weapon systems with newly developed operational concepts. M&S activities in the field of development of operational concept, requirements analysis and verification, system simulations for performance evaluation, system and component design by simulation, test and evaluation as simulation, and the development of training systems should be accompanied with the support of proper M&S applications.

In this paper, training architecture based on the M&S infrastructure which has been developed according to M&S master plan of KVLA program was presented. It was designed to have the form of hybrid structure in order to meet the requirements from end-user group. Implementation of training simulator with the help of hybrid training simulator was also discussed. Unique design of system interface between real-world tactical network and simulation network successfully supports the development of training simulator.

6. Acknowledgements

This study was supported by Agency for Defense Development in the part of Korean Vertical Launch Anti-Submarine Missile Systems Development Program.

7. References

[4] Freeman, Eric, Elisabeth Freeman, Kathy Sierra, and Bert Bates, “Head First Design Patterns”, O'Reilly Media, 2004
Author Biographies

YEONG-CHEOL KIM is a senior research engineer at PGM Technology Research Institute in Agency for Defense Development. He is in charge of the development of M&S systems for Korean Vertical Launch Anti-Submarine Missile Systems. He has many years of professional experiences for large-scale software development in the field of aerospace systems. His domain of interests is the integration of M&S functionality into real world systems based on open standards.

CHANG-BUM AHN is a principal research engineer at PGM Technology Research Institute in Agency for Defense Development. He is responsible for the system integration of Korean Vertical Launch Anti-Submarine Missile Systems. He has over 20 years of professional experiences in the field of modeling and simulation for underwater weapon systems. He also served as a team leader for the system integration and M&S in Korean Ship-Launched Torpedo development program. His research interest is in M&S based system engineering.

SANG-TAE LEE is junior research engineer at LIGNex1, Korea since 2004. He received a B.S of Computer Science in Kon-Kuk University. He studying Modeling and Simulation and made Korea-Missile Control Console GUI. Now, he is developing the Chelmae-II Training Simulator at M&S Center.

DAE-YOUNG KIM is a senior researcher at LIGNex1, Korea since 1998. He graduated with a M.S. in Control Engineering from Ajou University in 1998. He is currently the system engineer for the Chelmae-II System Simulator, the Chelmae-II Training Simulator and the Chelmae-II Testbed at M&S Center. The primary focus of his career has been M&S.