DEVELOPMENT OF AIR COMBAT HDEVS MODEL IMPLEMENTED IN HDEVSIM++ ENVIRONMENT

Jun Hee Lee
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology (KAIST)
291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea
the78910@kaist.ac.kr

Seon Han Choi
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology (KAIST)
291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea
gigohan01@kaist.ac.kr

Ho Dong Yoo
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology (KAIST)
291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea
hdyoo9@kaist.ac.kr

Jung Koo
HANGIL C&C Co. Ltd.
632 Seobusaet-gil, Geumcheon-gu, Seoul 08504,
Republic of Korea
kj67616@hangilcnc.com

Tag Gon Kim
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology (KAIST)
291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea
tkim@kaist.ac.kr

ABSTRACT

Being a mixture of discrete event and continuous time models, a hybrid model represents real-world dynamic systems in more realistic manner. Hybrid discrete event systems specification (HDEVS) formalism specifies a hybrid model in a modular, hierarchical form that employs discrete event systems specification (DEVS) formalism, differential equation formalism, and a formal interface. The HDEVSim++ environment implements the HDEVS formalism and an associate hybrid simulation algorithm in C++. This paper describes the development of an air combat hybrid model using the HDEVS formalism, which is implemented in the HDEVSim++ environment. The model contains a collection of fighter models, each of which consists of discrete event models for pilots and continuous models for flights. The simulation process is visualized using SIMDIS to verify and validate the trajectories for interactions between fighters, which are controlled by pilots' tactical decisions. Simulation performance is measured in execution time as the number of fighters varies.

Keywords: Hybrid System, HDEVS, HDEVSim++, Air Combat

1 INTRODUCTION

Modeling and simulation (M&S) has been used to analyze and verify tactics in various battlespace situations effectively (Seo *et al.* 2011). The analysis of air battle, which is the most important part of modern warfare, can be analyzed as an air combat model. The air battle situation is a hybrid system that consists of a discrete event system depicting various decisions and tactics and a continuous-time system that describes the missiles and fighters' movements. For example, as shown in Figure 1, the maneuvering part corresponding to the engine in the fighter aircraft is a continuous time model (CTM), which can be described by the 6-degrees-of-freedom (6-DOF) differential equation, and a pilot model that recognizes battlespace situations and determines maneuvering can be specified as a discrete event model (DEM) through discrete event systems specification (DEVS). Therefore, hybrid M&S methodology should be applied to analyze the hybrid system of air combat situations effectively.

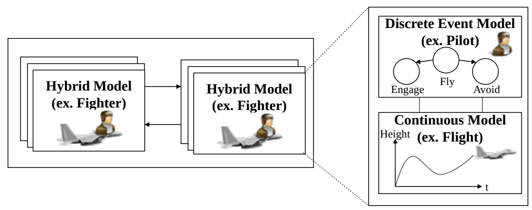

Figure 1: Example of an Air Combat Hybrid Model

Hybrid M&S methodology can be divided into three types as follows. The first is the modeling methodology integration method. It describes and implements a hybrid model through an integrated modeling formalism proposed only for hybrid systems. The second method describes each DEM and CTM for the subsystems and executes the models using the integrated simulation engine. It can apply existing modeling methodology to detailed systems. The third method's modeling mechanism is the same as the second method, and it executes each model with the separated simulation engine. In this case, an interoperation process is necessary.

This paper describes the air combat model by applying HDEVS, which is one of the second methodologies for specifying hybrid systems. HDEVS provides an environment for modeling hybrid systems in a hierarchical and modular form by adding a continuous atomic model that solves ordinary differential equations, a continuous-coupled model, and a hybrid coupled model to the existing DEVS coupled model and atomic model. In addition, as DEM and CTM can be modeled separately, reusing existing models is possible. Because the DEM handles the message and the CTM handles the signal as input and output data, it is necessary to transform the data in order to exchange information between the two kinds of models. HDEVS provides an environment to convert the data using an interface model called a converter, as shown in Figure 2.

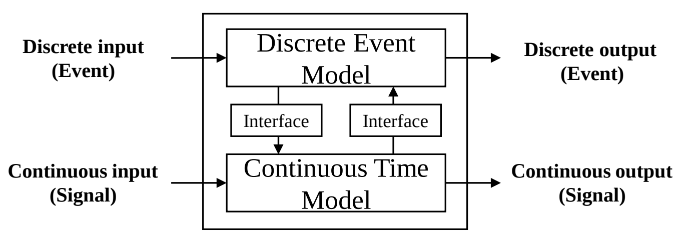

Figure 2: HDEVS Modeling Concept

The paper is organized as follows. Section 2 explains HDEVS and its environment, and Section 3 explains an HDEVS-designed air combat model and its implementation environment. Section 4 describes a case study of the air combat model, and Section 5 explains a simulation performance experiment and the results. Finally, Section 6 concludes this paper.

2 BACKGROUND

2.1 Abstraction Levels of Defense Modeling

M&S provides means to obtain data from design to operational efficiency evaluation. This series of work requires a series of M&S with varying abstracted levels of detail that are appropriate for a particular application. Table 1 shows four kinds of hierarchies of defense modeling and their characteristics. The air combat model described in this paper is a combined model of engagement and engineering level. The aircraft's maneuvering model and missile CTM correspond to engineering-level modeling, and other models correspond to engagement-level modeling.

Table 1: Abstraction Levels of Defense Modeling (Piplani, 1994)

Level of model	Force	Level of detail	Output	Model Type
Theater/Campaign	Combined	Aggregated	Campaign outcome	Discrete Event System Model
Mission/Battle	Multi-platform	Some aggregation or individual entities	Mission effectiveness	
Engagement	Some ally entities vs. some enemy entities	Detailed subsystems	System effectiveness	
Engineering	Single weapon systems	Highly detailed	Performance of systems, components	Continuous System Model

2.2 HDEVS M&S Environment

The HDEVS formalism specifies a hybrid model in a modular, hierarchical form that employs DEVS formalism, differential equation formalism, and a formal interface. The HDEVSim++ environment implements the HDEVS formalism and an associate hybrid simulation algorithm in C++. Meanwhile, HDEVS is applied in research presenting a system-of-systems (SoS) approach for a cyber-physical system as modeled formally (Lee *et al*. 2015).

2.2.1 HDEVS Formalism

Definition 1 (Kwon *et al*. 2013) is a mathematical expression for the hybrid coupled model (HCM). Each CTM, DEM, HCM, and converter model has an equivalent relationship and a coupling relationship. Also, discrete event input/output is connected to discrete event input/output, and continuous time input/output is connected to continuous time input/output. The submodels of these HCMs can be HCMs, DEMs, CTMs, or S/E (signal-to-event) and E/S (event-to-signal) converters (interface models), as mentioned.

Definition 1: Hybrid Coupled Models (HCMs)

$$HCM = < X, Y, M, IC, EIC, EOC, SELECT >$$

$$X = X_{cont} \cup X_{disc} : Set\ of\ hybrid\ inputs$$

$$Y = Y_{cont} \cup Y_{disc} : Set\ of\ hybrid\ outputs$$

$$M \subseteq CM \cup DM \cup HCM \cup SEM \cup ESM$$

$$: Set\ of\ hybrid\ components$$

Constraints:

$$EIC \subseteq (X_{disc} \times \cup_i dX_i) \cup (X_{cont} \times \cup_i cX_i)$$

$$: \text{External Input Coupling Relation}$$

$$EOC \subseteq (\cup_i dY_i \times Y_{disc}) \cup (\cup_i cY_i \times Y_{cont})$$

$$: \text{External Output Coupling Relation}$$

$$IC \subseteq (\cup_i dY_i \times \cup_j dX_j) \cup (\cup_i cY_i \times \cup_j cX_j)$$

$$: \text{Internal Coupling Relation}$$

$$Select : 2^{\{M\}} - \emptyset \rightarrow M : \text{Tie-Breaking Function}$$

Definition 2 (Kwon *et al*. 2013) is a mathematical representation of the S/E and E/S converter models. SEM represents a S/E converter model that converts analog signals into events, and ESM represents an E/S converter model that converts events into analog signals. These two converter models have different types of input and output sets, each of which converts the type of data through the conversion functions f_{SE} and g_{ES}.

Definition 2: Signal-to-Event Converter Model

$SEM = < X_{cont}, Y_{disc}, f_{SE} >$
X_{cont}: Set of Continuous Time Inputs
Y_{disc}: Set of Discrete Event Outputs

Constraint:
$f_{SE}: \Omega \rightarrow 2^{Y_{disc}}$
- Ω : Continuous Set of X_{cont}
- $2^{Y_{disc}}$: Power Set of Y_{disc}

Definition 3: Event-to-Signal Converter Model

$ESM = < X_{disc}, Y_{cont}, g_{ES} >$
X_{disc}: Set of Discrete Event Inputs
Y_{cont}: Set of Continuous Time Outputs

Constraint:
$g_{ES}: 2^{X_{disc}} \rightarrow \Omega$
- $2^{X_{disc}}$: Power Set of X_{disc}
- Ω : Continuous Set of Y_{cont}

2.2.2 HDEVSim++ Environment

HDEVSim++ is a simulation environment that runs HDEVS-based models Kwon *et al.* (Kwon *et al*. 2013, Sung and Kim, 2011) developed. The simulation engine reconstructs the integrated hybrid model as a preprocessing procedure. During that procedure, the integrated model is separated into the DEM and CTM, and the engine identifies interface models known as converter models. Then, the integrated simulation engine executes each separated model through the DEVS simulation engine and continuous simulation engine for efficiency, as shown in Figure 3.

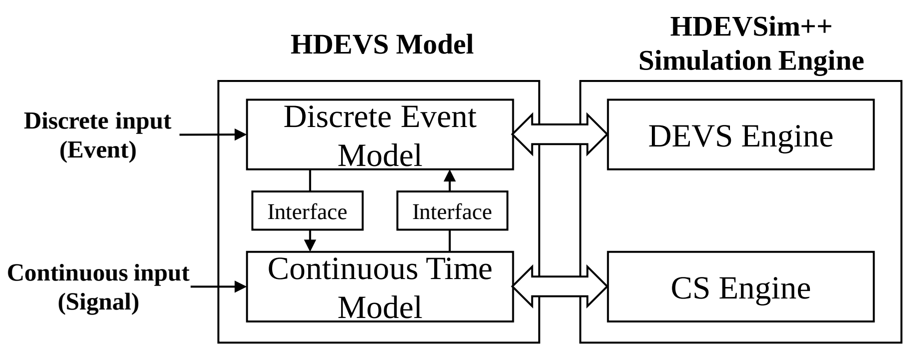

Figure 3: HDEVSim++ Simulation Concept

The class hierarchy of HDEVSim++ environment is shown in Figure 4. Every model inherits the CModel, which is a common skeleton class of models, and adds necessary functions. CHCoupled is a hybrid coupled model class, CAtomic is a discrete atomic model class, CCAtomic is a continuous atomic model class, and CAEInterface is an S/E converter model class. The dashed box in Figure 4 is the core API of each class. The CHCoupled class has an AddComponent, which registers models in the coupled model; AddCoupling, which is for coupling between DEMs; and AddCSCoupling, which is for coupling between DEMs and CTMs. CAtomic class has ExtTransFn, which is a DEVS external transition function; IntTransFn, which is a DEVS internal transition function; OutputFn, which is a DEVS output function; and TimeAdvanceFn, which is a DEVS time advance function. CCAtomic class has UpdateState, which updates the state of CTM every time step, and UpdateOutput, which generates an output based on the updated state. The CAEInterface has CheckCondition, which checks whether an event is generated from the CTM, and UpdateOutput, which creates output based on the generated event. The E/S converter model inherits the CAtomic class because it needs only ExtTransFn, which receives a message from DEM and transforms it to a signal for coupled CTM in a predefined manner. In the case of the S/E and E/S converter models, it is necessary to specify each type of model as AE or EA in a constructor.

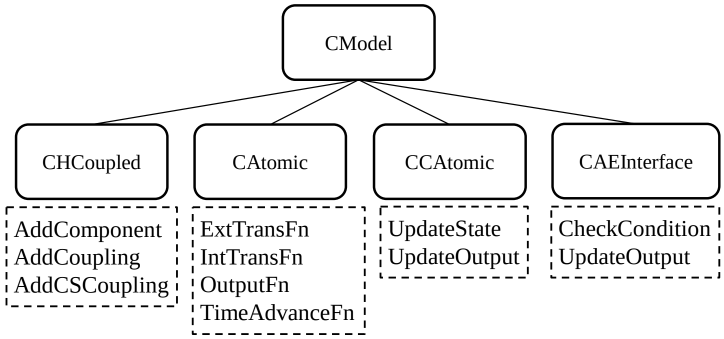

Figure 4: HDEVSim++ Model Class

3 AN AIR COMBAT HYBRID MODEL DESIGN AND IMPLEMENTATION

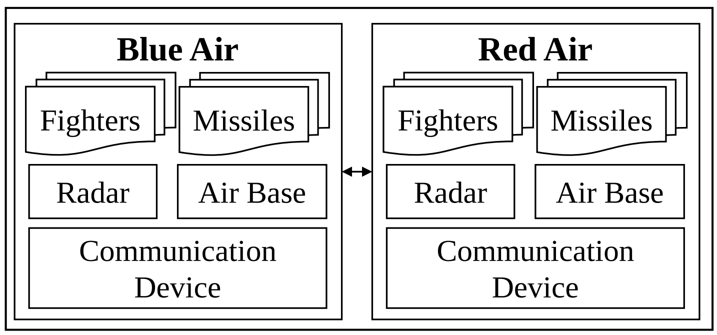

Figure 5: Air Combat Hybrid Model Structure

3.1 Modeling Overview

The elements that make up the air combat are fighter aircrafts, missiles, radars, airbase, and communication devices, as shown in Figure 5. Blue Air represents ally forces, and Red Air represents enemy forces. An example of an air-to-air combat scenario is as follows: ally and enemy fighter squadrons equipped with medium-range and short-range air-to-air missiles conduct beyond-visual-range (BVR) engagements. This includes maneuvering, detecting, and engaging fighters which are the core combat objects. The fighters' maneuvering covers taking off, ascending, flying in orbit, and route flying. In particular, tactical formation flying is essential to the squadron missions. The fighters detect opposing forces using equipped sensors and attack with air-to-air missiles after detecting them. The attacked fighters detect the missile threat and respond to enemy missiles by conducting an escape maneuver.

The fighter model is a hybrid model consisting of a 6-DOF maneuvering equation model (Choi *et al*. 2016) describing continuous maneuvers and a DEM that decides maneuvers based on given battlespace situation information. Various kinds of fighters and missiles can be simulated through external parameter changes. As missiles do not need to describe precise movement like fighters, a pseudo 6-DOF model, which is a simplified 6-DOF model is applied for missile models. Missile types include air-to-air missiles for air combat and free-fall and guided bombs for air-to-ground combat. The radar model consists of two DEMs, one of which detects enemy objects, and the other transmits the detected information. The airbase is a DEM that compiles the battlespace situation and gives orders, like sortie commands, to fighters. A communication device is a DEM that describes the communication between objects and delivers the messages after a certain delay.

HDEVS formalism is suitable for this air combat model because it can represent the hierarchical structure of those complex models shown in Figure 6. It can also specify the CTM of the fighter. Additionally, through modular design using HDEVS formalism, we can freely change the number of fighters and missiles according to the requirements. Consequently, the air combat model is effective for various battlespace situation analyses.

3.2 HDEVS Air Combat Hybrid Modeling

Figure 6 shows the HDEVS air combat hybrid model based on the overall modeling structure. For simplicity, we assume that there is one fighter aircraft and one missile. Blue Air, shown in Figure 6-(a), is an upper HCM that encompasses all blue forces. It consists of a BAFtr (fighter HCM), a BAMsl (missile HCM), a TACC_MCRC (ground radar base HCM), a communication device (discrete event atomic model (DAM)) and an air base (HCM). The battle participation elements similar to the actual battlespace situation briefly described in Section 3.1 are communicating with each other.

Among them, we describe in more detail the BAFtr, which is composed of DEMs, CTM, and converter models. It is shown in Figure 6-(b). It consists of a BAPilot (HCM) that acts as a pilot, a BSensor (DAM) that acts as an aircraft sensor, and a BCSManeuver (continuous time atomic model (CAM)), which is a maneuvering part of the fighter. This BCSManeuver model is the engine of the fighter aircraft, which runs every fast cycle (0.01sec) and updates the fighter's position. It is the same as Choi *et al.*'s model (Choi *et al*. 2016). The BCSManeuver is a CTM, and the rest of the models are DEMs; therefore, the type of data between two kinds of models are different. A BESConverter and BSEConverter are used to convert the type of data between the DEMs and CTMs. The E/S converter converts DEM data to CTM data, and the S/E converter converts CTM data to DEM data.

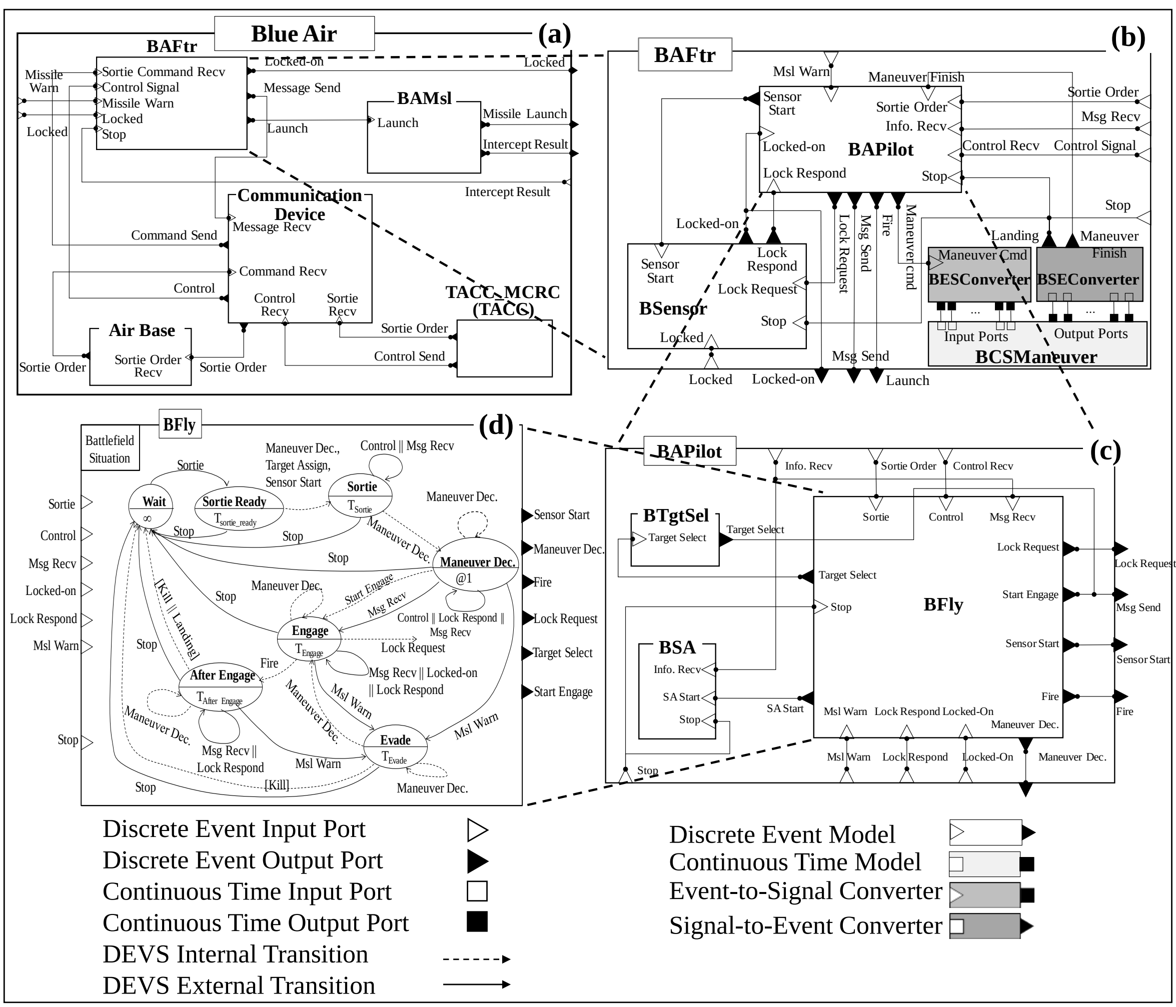

Figure 6: HDEVS Air Combat Hybrid Model

It seems complicated to separate the entire model into DEMs and CTMs connecting with S/E and E/S converter models, although it has a benefit in terms of scheduling. For instance, in the case of route flight situations, usually, pilot DEMs must check as to whether they arrive at the waypoint periodically. However, if this methodology is applied, CTM and S/E converters check whenever they execute on behalf of pilot DEMs, and S/E converters send an event message when they arrive. Consequently, the DEM does not need to be scheduled frequently. Based on the HDEVS environment, the BAFtr model can be represented with HDEVS formalism as follows.

$$
\begin{aligned}
&HCM = < X, Y, M, IC, EIC, EOC, SELECT > \\
&\quad X = X_{cont} \cup X_{disc} : \textit{Set of hybrid inputs} \\
&\quad\quad -X_{cont} = \{\emptyset\} \\
&\quad\quad -X_{disc} = \{\text{MslWarn, SortieOrder, MsgRecv, ControlSignal, Locked, Stop}\} \\
&\quad Y = Y_{cont} \cup Y_{disc} : \textit{Set of hybrid outputs} \\
&\quad\quad -Y_{cont} = \{\emptyset\} \\
&\quad\quad -Y_{disc} = \{\text{Locked}-\text{on, MsgSend, Launch}\} \\
&\quad M \subseteq CM \cup DM \cup HCM \cup SEM \cup ESM : \textit{Set of hybrid components} \\
&\quad\quad -\text{CM} = \{\text{CS Maneuver}\} \\
&\quad\quad -\text{DM} = \{\text{BSensor}\} \\
&\quad\quad -\text{H}CM = \{\text{BAPilot}\} \\
&\quad\quad -\text{SEM} = \{\text{BSEConverter}\} \\
&\quad\quad -\text{ESM} = \{\text{BESConverter}\}
\end{aligned}
$$

$$
\begin{aligned}
&\textit{Constraints:} \\
&\quad EIC \subseteq (X_{disc} \times \cup_i dX_i) \cup (X_{cont} \times \cup_i cX_i) : \text{External Input Coupling Relation} \\
&\quad\quad = \left\{ \begin{array}{c}
(BAFtr.MslWarn\ , BAPilot.MslWarn), \\
(BAFtr.SortieOrder, BAPilot.SortieOrder), \\
(BAFtr.MsgRecv, BAPilot.InfoRecv), \\
(BAFtr.ControlSignal, BAPilot.ControlRecv), \\
(BAFtr.Stop, BAPilot.Stop), \\
(BAFtr.Stop, BSensor.Stop)
\end{array} \right\} \\
&\quad EOC \subseteq (\cup_i dY_i \times Y_{disc}) \cup (\cup_i cY_i \times Y_{cont}) : \text{External Output Coupling Relation} \\
&\quad\quad = \left\{ \begin{array}{c}
(BAPilot.MsgSend, BAFtr.MsgSend), \\
(BAPilot.Fire, BAFtr.Launch), \\
(BSensor.LockedOn, BAFtr.LockedOn)
\end{array} \right\}
\end{aligned}
$$

$$
\begin{aligned}
&\quad IC \subseteq (\cup_i dY_i \times \cup_j dX_j) \cup (\cup_i cY_i \times \cup_j cX_j) : \text{Internal Coupling Relation} \\
&\quad\quad = \left\{ \begin{array}{c}
(BAPilot.SensorStart, BSensor.SensorStart), \\
(BAPilot.LockRequest, BSensor.LockRequest), \\
(BAPilot.ManeuverCmd, BESConverter.ManeuverCmd) \\
(BSensor.LockedOn, BApilot.LockedOn), \\
(Bsensor.LockRespond, BAPilot.LockRespond), \\
(BESConverter.Out, BCSManeuver.In), \\
(BCSManeuver.Out, BSEConverter.In), \\
(BSEConverter.Landing, BAPilot.Stop), \\
(BSEConverter.ManeuverFinish, BAPilot.ManeuverFinishIn)
\end{array} \right\}
\end{aligned}
$$

The BAPilot (Pilot HCM), which acts as a pilot, is shown in Figure 6-(c). It consists of a BFly (DAM), which is a mental model of a pilot, and a BSA (DAM), which is the situation-determination part of a pilot. In addition, only the leader of a squadron has a BTgtSel (DAM), which selects and assigns targets to wingmen.

The pilot mental model BFly (DAM), which is the core part of the BAPilot, is shown in Figure 6-(d). It has the states of Wait, Sortie_Ready, Sortie, Maneuver Decision, Engage, After_Engage, and Evade; makes complex maneuvering decisions through input messages and state-variable values in a specific state; and sends the determined control to the BCSManeuver CAM through a BESConverter to perform corresponding maneuvering. When BFly receives the maneuvering a completed message delivered from the BSEConverter, it makes another maneuvering decision based on that state.

3.3 Model Implementation in HDEVSim ++ Environment

Figure 7 shows an example of one-to-one mapping of a source code and a design diagram of the sensor DAM written in C++ language in an HDEVSim++ environment. It specifies a coupling relationship of input/output ports, an external transition function, an internal transition function, and an output function as an example. The AddInPort is a function for adding input ports, and the AddOutPort is a function for adding output ports. The ExtTransFn is the external transition function, which handles received messages and changes states based on the message; the IntTransFn is the internal transition function, which changes state by internal event; and the OutputFn is the out function, which generates an output according to the state. For reference, in this example, the IntTransFn and OutputFn are executed in same time. Because the design document and source code are matched in a one-to-one relationship, the source code becomes easy to understand and maintain.

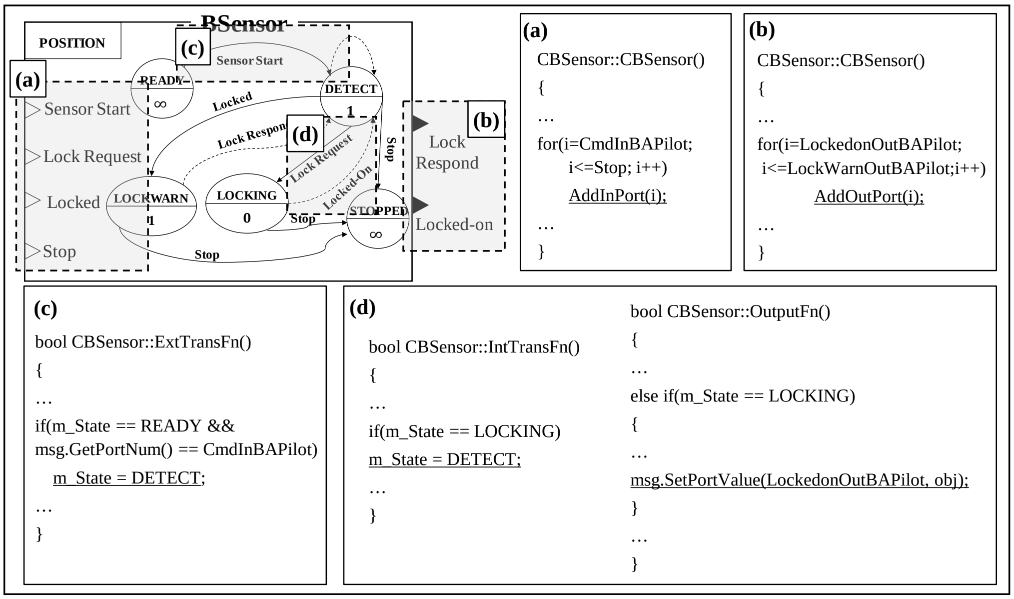

Figure 7: Implementation in HDEVSim++ Environment

Meanwhile, the HDEVS air combat model inherits the classes described in Section 2.2.2 and Figure 4 to utilize HDEVSim++ environment. For example, the BAFtr (HCM), BFly (DAM), BCSManeuver (CAM), and BSEConverter (S/E Converter) models inherit CHCoupled, CAtomic, CCAtomic, and CAEInterface classes in the order named.

The overall structure of the implemented system is shown in Figure 8. HDEVS air combat model parameters are set by external parameters. Then, a hybrid simulation engine executes the HDEVS air combat model. During the model's execution, SIMDIS visualizes the simulation results in real-time by receiving location data from fighter and missile models. SIMDIS is an analysis and display tool provided by the US Naval Research Lab (U.S. Naval Research Laboratory, 2015). By visualizing simulation results in real-time, we can verify the simulation process. Also, the model can be varied using an external parameter–setting methodology.

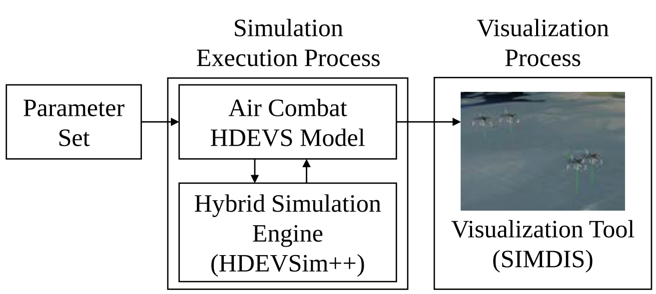

Figure 8: Implemented System

4 SIMULATION OF THE AIR COMBAT MODEL

Using the method described in Section 3, we have developed an air combat model that encompasses air-to-air, air-to-ground, and surface-to-air combat situations. Table 2 shows the strike package participating in the air combat model and its respective roles. For simplicity, only ally forces are explained.

Table 2: Strike Package Roles

Mission Roles	Actions
Strategic Attack (2F-15K)	Taking off 5 minutes before Fighter Sweep Forces taking off
	Launch SLAM-ER on two ground targets at a specific orbit point
Fighter Sweep (2F-16C/D)	Taking off 2 minutes before INT Forces taking off, flying along the path of INT Forces (flying at 20,000ft)
	Fly in the orbit point and return
INT (4F-16C/D)	Route flight and tactical attack
	#3/4 flights take off one minute after #1/2 and maintain 8NM distance with forward aircrafts
Escort (2F-15K)	Taking off one minute after INT forces, maintain distance with INT forces 8NM while fly along path of INT forces
	Fly in the orbit point and return
Air Control (E-737) Electronic Warfare (EA-6B)	Taking off and circular flying at specific location

The mission scenario of the above strike package is represented graphically, as shown in Figure 9.

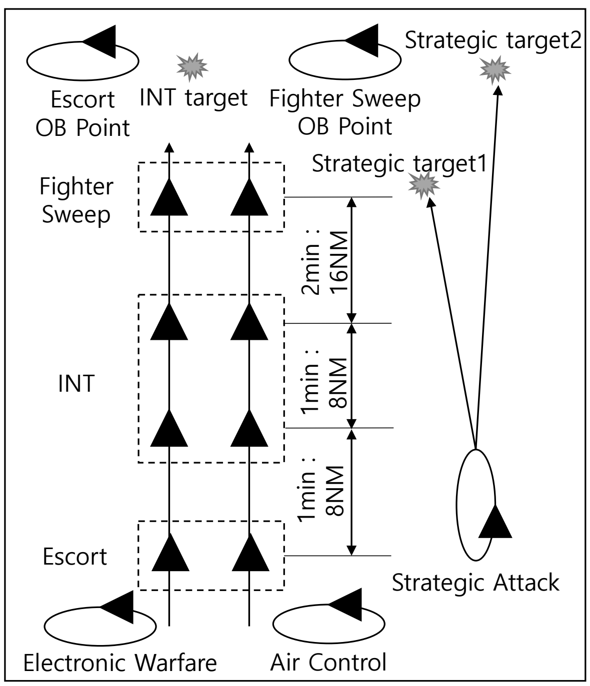

Figure 9: Mission Scenario

Figure 10 shows the battlespace situation, which is a basic execution environment of the air combat model that models the air battle scenarios of the strike package described above. The overall attack scenario is as follows: the air controller aircraft and electronic warfare aircraft take off for the first time and fly to the specific orbit point. Then, the tactical attack forces take off and strike the two ground targets with longrange missiles. Then, the fighter sweep forces, INT forces, and escort forces take off in the order shown in Figure 9. The fighter sweep forces engage with enemy fighters. After they engage, they correspond to the enemy's surface-to-air missile threat. If they succeed at evading, they fly to the specific orbit point to cover the rear squadron. Then, INT forces drop bombs at the ground target and return to base. The escort forces cover the INT forces during route flying. The path assumes a route flight from the Cheong-Ju airport to the North Korean Sun Palace (ground target) and then back to the Cheong-Ju airport, as shown in Figure 10.

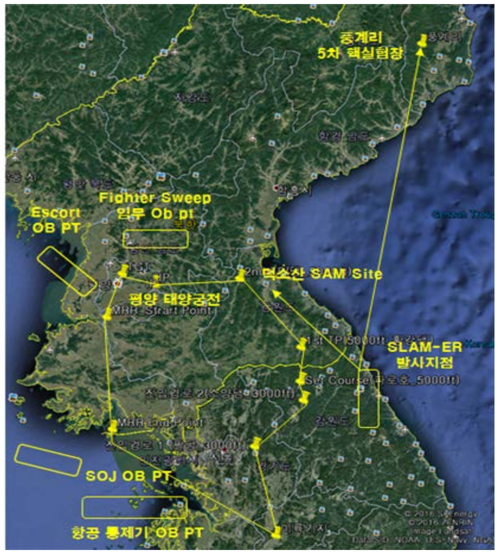

Figure 10: Common Operational Picture (COP)

Figure 11 shows the screen in which the simulation is taking place in SIMDIS under the system environment, as shown in Figure 8. When an aircraft's maneuvering CTMs perform an execution, they send position information to SIMDIS so that SIMDIS can construct the graphical information of the aircraft models. Consequently, we can visually verify the simulation process and results in real-time.

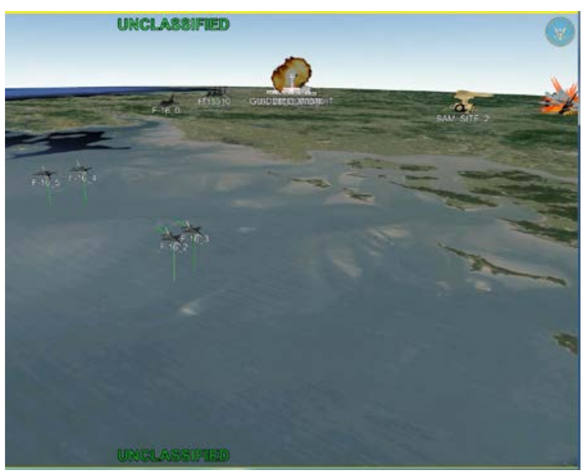

Figure 11: Visualization with SIMDIS

5 PERFORMANCE MEASUREMENT

To check if our model can be executed efficiently, an experiment is conducted on a simplified model. The simplified model contains only route flying fighters, and the route is same as that described in Figure 10. In this performance experiment, we repeatedly measured the whole simulation runtime without real-time visualization while gradually augmenting the number of fighters from 100 to 1,000 to determine the trend of growth at run time according to the number of fighter aircrafts. The environment for testing the simulation performance is as follows: OS: Windows 10, CPU: i7-6700, 3.4 GHz, RAM: 16 GB DDR4. The simulation run time estimation results are shown in Figure 12. For reference, the route-flight scenario used in the experiment is performed for 3,772 seconds (more than 1 hour) in the real-world. Even if the number of fighters increases up to 800, simulation execution is performed at a higher speed than real-world time. In other words, real-time simulation is possible. The simulation time increases linearly with the number of fighters. Therefore, even though the number of fighters increases, decent performance is guaranteed.

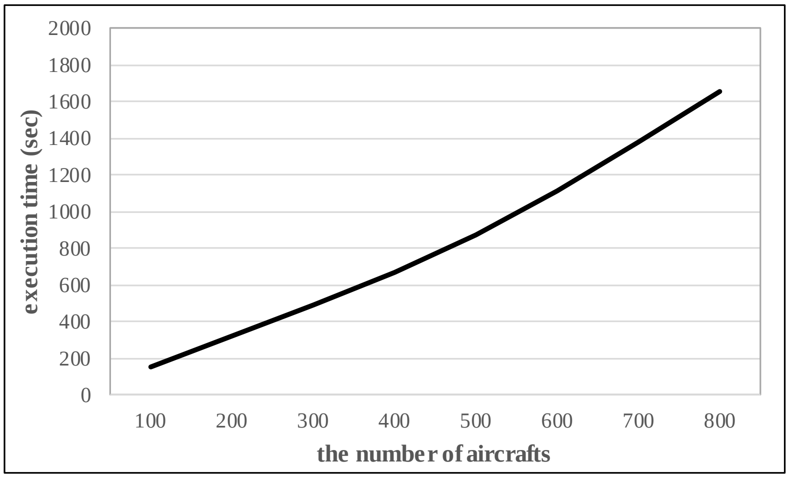

Figure 12: Experiment Results of Air Combat Model Execution

6 CONCLUSION

This paper shared an experience of developing an air combat hybrid model that contains a collection of fighter models, missile models, radar models, airbase models, and communication models. Also, it described a hybrid modeling design using the HDEVS formalism, and explained the implementation and simulation of the model in an HDEVSim++ environment. To verify and validate simulation execution and trajectories for interactions between fighters visually, SIMDIS is applied as a visualization tool. The experiment of simulating an air combat model in certain scenarios has shown that simulation runtime increases linearly with the number of aircrafts.

ACKNOWLEDGMENTS

This work was supported by South Korea Air Force under the project titled "System development for analyzing the effectiveness of aircraft's weapons".

REFERENCES

Choi, S. H., Lee, J. H., Lee, S. H., Yoo, H. D., Koo, J., and Kim, T. G. 2016. "6-dof aircraft simulation model capable of handling maneuver events (WIP)". In *Proceedings of the Summer Computer Simulation Conference*, Montreal, QC, Canada, pp. 427-432.

Kwon, S. J., Sung, C., Song, H. S., and Kim, T. G. 2013. "Integrated hybrid systems modeling and simulation methodology based on HDEVS formalism". In *Proceedings of the Summer Computer Simulation Conference*. Toronto, Canada.

Lee, K. H., Hong, J. H., and Kim. T. G. 2015. "System of systems approach to formal modeling of cps for simulation-based analysis". *ETRI Journal* vol. 37, no. 1, pp. 175-185.

Piplani, L. K. 1994. Systems Acquisition Manager's Guide for the use of Models and Simulations. *DEFENSE SYSTEMS MANAGEMENT COLL FORT BELVOIR VA.*

Seo, K. M., Song, H. S., Kwon, S. J., and Kim, T. G. 2011. "Measurement of Effectiveness for an Anti-torpedo Combat System Using a Discrete Event Systems Specification-based Underwater Warfare Simulator". *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology* vol. 8, no. 3, pp. 157-171.

Sung, C., and Kim, T. G. 2011. "Framework for simulation of hybrid systems: Interoperation of discrete event and continuous simulators using hla/rti". In *Proceedings of the IEEE Workshop on Principles of Advanced and Distributed Simulation. IEEE Computer Society.*

U.S. Naval Research Laboratory. 2015. SIMDIS User's Manual, Available from: https://simdis.nrl.navy.mil

AUTHOR BIOGRAPHIES

JUN HEE LEE received his B. S. in Computer Science and Electronic Engineering from Handong Global University in 2016. He is currently a M.S. student in the Department of Electrical Engineering at Korea Advanced Institute of Science and Technology (KAIST). His research interests include hybrid systems modeling and simulation. His email address is the78910@kaist.ac.kr.

SEON HAN CHOI received the B.S. and M.S. degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2012 and 2014. He is currently pursuing the Ph.D. degree in electrical engineering at KAIST. His research interest includes modeling & simulation of discrete-event system, efficient optimization. His email address is gigohan01@kaist.ac.kr.

HO DONG YOO is a Research Fellow of Department of Electrical and Electronic Engineering at Korea Advanced Institute of Science and Technology (KAIST). He holds a PhD in Computer Science from KAIST. His research interests include system modeling and simulation, especially in defense domain. His email address is hdyoo9@kaist.ac.kr.

JUNG KOO received his Ph.D. in NCW Engineering from Ajou University, and currently is the Chief of R&D Center of Hangil C&C Company. He is also in charge of developing the simulation logic of the air operations analysis model for the ROKAF. His email address is kj67616@hangilcnc.com.

TAG GON KIM is a Professor of the Department of Electrical Engineering at Korea Advanced Institute of Science and Technology (KAIST). He holds a Ph.D. in computer engineering from University of Arizona. His research interests include methodological aspects of systems modeling simulation and defense modeling and simulation. He is a fellow of SCS. His email address is tkim@kaist.ac.kr.