Hindawi
Security and Communication Networks
Volume 2018, Article ID 5150730, 12 pages
https://doi.org/10.1155/2018/5150730

Research Article

An Open Architecture Framework for Electronic Warfare Based Approach to HLA Federate Development

HyunSeo Kang,[1] **YoonJe Sung**,[1] **HyoungJun Kwon**,[1] **SugJoon Yoon**,[1] **and SangYeong Choi**[2]

[1]*Department of Aerospace Engineering, Sejong University, Seoul, Republic of Korea*
[2]*School of Defense Science, Hansung University and Myongji University, Seoul, Republic of Korea*

Correspondence should be addressed to SugJoon Yoon; sjyoon@sejong.ac.kr

Received 17 November 2017; Accepted 14 February 2018; Published 21 March 2018

Academic Editor: Prem Mahalik

A variety of electronic warfare models are developed in the Electronic Warfare Research Center. An Open Architecture Framework for Electronic Warfare (OAFEw) has been developed for reusability of various object models participating in the electronic warfare simulation and for extensibility of the electronic warfare simulator. OAFEw is a kind of component-based software (SW) lifecycle management support framework. This OAFEw is defined by six components and ten rules. The purpose of this study is to construct a Distributed Simulation Interface Model, according to the rules of OAFEw, and create Use Case Model of OAFEw Reference Conceptual Model version 1.0. This is embodied in the OAFEw-FOM (Federate Object Model) for High-Level Architecture (HLA) based distributed simulation. Therefore, we design and implement EW real-time distributed simulation that can work with a model in C++ and MATLAB API (Application Programming Interface). In addition, OAFEw-FOM, electronic component model, and scenario of the electronic warfare domain were designed through simple scenarios for verification, and real-time distributed simulation between C++ and MATLAB was performed through OAFEw-Distributed Simulation Interface.

1. Introduction

With the advance of science and technology, the aspect of the warfare has changed the electronic warfare. Electronic warfare technology is validated through modeling and simulation (M&S) to save cost and time. M&S has been acknowledged as one of the most important systems in the defense field [1].

Electronic warfare is defined as the military action involving the use of electromagnetic energy to determine, exploit, reduce, or prevent hostile use of electromagnetic spectrum while simultaneously safeguarding one's own [2]. The three major subdivisions within electronic warfare are electronic warfare support (ES), electronic attack (EA), and electronic protection (EP) [3].

Electronic warfare modeling & simulation (EW M&S) is performed by gathering various EW equipment (object) modeling. At this time, the lack of EW M&S reusability, scalability, and interoperability has been degrading its usefulness [4]. Several studies have produced framework for M&S-based system, but they have not been dealing with EW M&S problem domain [5]. In order to solve these problems, "Open Architecture Framework for Electronic Warfare (called OAFEw)" was developed that can achieve the reuse and interoperability of EW M&S lifecycle [4]. The basic concept of OAFEw is that EW M&S software components are to be reassembled to suit the user's needs with its building blocks defined in the common Reference Conceptual Model and rules governing all stages through an EW M&S lifecycle [4].

Now, High-Level Architecture/Run-Time Infrastructure (HLA/RTI) is already widely used in the field of defense M&S. Also, related research is actively proceeding [5, 6]. In addition, the US DMSO (Defense Modeling Simulation Office) mandates that all defense M&S developed after 2000 are executive on RTI basis. However, there is no study on architecture framework and HLA/RTI distributed interface in EW M&S domain.

This paper focuses on simulation method, specially EW components interoperability and resusability technology for HLA. HLA is a high-level concept to support interoperability

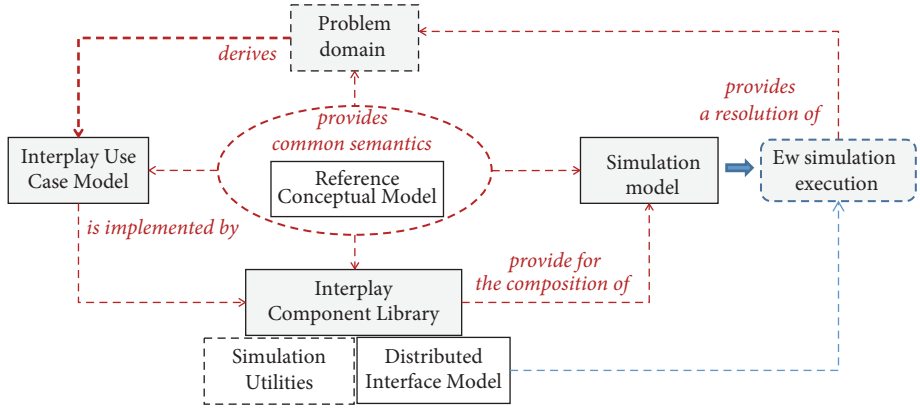

Figure 1: Constituents of OAFEw.

for distributed simulation of heterogeneous simulator as proposed by US DoD (Department of Defense) [7–9].

We present an HLA federate development method that is highly reusable and interoperable based on OAFEw Rules and OAFEw Reference Conceptual Model.

In the next section, we will review the relevant research works. And Sections 3 and 4 describe how to develop OAFEw-based HLA federate. In Section 5, we will show its implementation and show demonstration. Finally, we will have conclusions in Section 6.

2. Related Literature Review

2.1. OAFEw Basic Concept Review. An Open Architecture Framework of Electronic Warfare modeling & simulation (called OAFEw) is intended to foster easy composition of the EW simulation model and to be used for the lifecycle management of EW simulation components on the basis of the common Reference Conceptual Model so that it can achieve the reuse and interoperability of EW M&S. The basic concept of OAFEw is as follows.

Firstly, we assemble M&S components according to an EW scenario on the basis of the EW Reference Conceptual Model. Secondly, we use common templates for the component specifications that are the embodiment of elements in the EW Reference Conceptual Model. Thirdly, we keep the common rules for component assembling, its implementation, reuse, and governing all stages through an EW M&S lifecycle [4].

OAFEw comprises a Reference Conceptual Model, an Interplay Use Case Model, an Interplay Component Library, and a simulation model, which are shown in Figure 1.

The Reference Conceptual Model is a conceptual model of an EW real world in the problem domain. It is noted that a conceptual model represents the real world to be implemented in the simulation model. The Reference Conceptual Model provides common semantics to be referenced by other OAFEw constituents [4].

The Interplay Use Case Model is to model use cases within the context of EW simulation domain. The Interplay Use Case Model is associated with an EW scenario with which a certain problem should be resolved. The scenario thus can be formulated using each of the interplay use cases. The Interplay Component Library is a collection of EW simulation components implemented according to the specifications of the interplay use cases. The components in the Interplay Component Library must have both internal interfaces to the simulation utilities such as a simulation engine and external interfaces to the HLA/RTI to be compliant with the HLA of distributed simulation. Thus, there are two types of components in the Interplay Component Library. These are simulation utility components and distributed interface components. Simulation utility components are simulation execution and collection of simulation data. Distributed interface components are for the interoperability of the components residing on the other federates in the EW federation [4].

The simulation model is to mimic an EW real world related to a problem domain. The simulation model is composed of components in the Interplay Component Library. The components are reused for the simulation composition associated with an experiment scenario of the problem domain [4].

2.2. OAFEw Specification Rules Review. Specification rules constrain the specification method of the OAFEw constituents such as the Reference Conceptual Model, the Interplay Use Case Model, and the Distributed Interface Model. Figure 2 shows the graphical depiction of their relationships constraining each other.

Firstly, the Reference Conceptual Model provides OAFEw constituents of a common dictionary which explains what is going on in an EW real world. The Reference Conceptual Model should be described using CML (Conceptual Modeling Language). The CML elements are shown schematically in Figure 3. The CML describes both static and dynamic properties of EW entities in a way that an EW element generates an event, then the event again activates the element behavior and alters the state of the element. Every element belongs to a category and has characteristics. Piece and game space inherit from the element, and they belong to a battle space. For the sake of the fuller modeling power of the OAFEw constituents, we extend the notion of the events

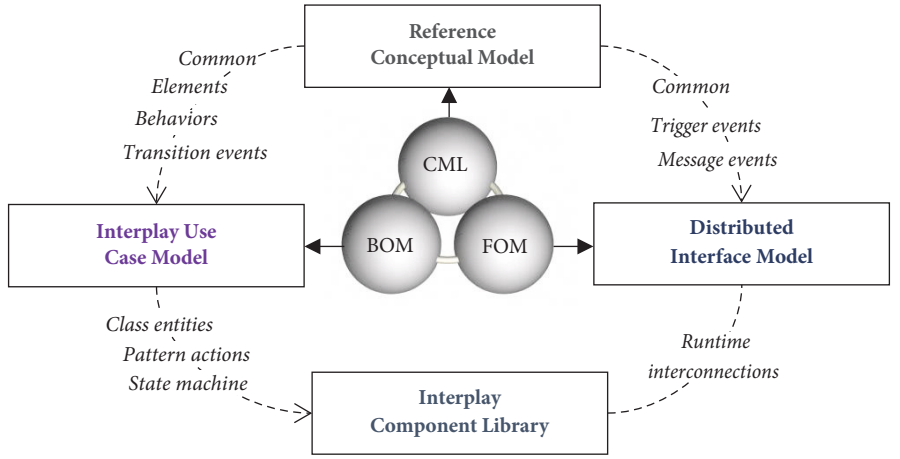

Figure 2: Specification methods.

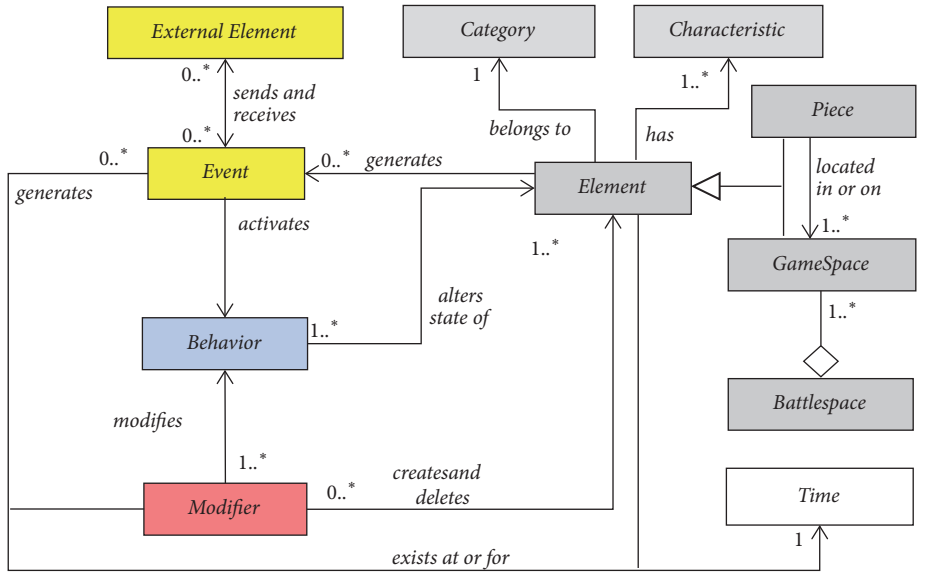

Figure 3: Conceptual Modeling Language.

to three types which are partly adapted from BOM (Base Object Model): transition event, message event, and trigger event. Transition event is an event that changes the inner state of an element. Either the message event or the trigger event is an event sending an external element. The message event designates the receiver, but the trigger event does not designate the receiver.

Secondly, the OAFEw use cases in the Interplay Use Case Model have to be captured so that the series of use cases may compose an experimental scenario for the resolution of problems. The interplay is the same notion as in the BOM standard which is a sequence of pattern-actions involving one or more conceptual entities which appeared in the EW Reference Conceptual Model. The pattern-action is a single step in a pattern of interplay that may result in a state change of a conceptual entity. Thus, each Interplay Use Case should be specified using BOM standard. Further, the specification of interplay use cases has to use the same dictionary of conceptual entities, behaviors, and events in the EW Reference Conceptual Model, because they should share the common semantics with each other. The use case specifications are then to be implemented as reusable components in the Interplay Component Library. The components provide the simulation model with reusable building blocks.

Finally, the Distributed Interface Model should be specified with the HLA Object Model Template (OMT) in order to be compliant with HLA. HLA Object and Interaction in the OMT have to be defined using either the message event or the trigger event in the conceptual model in order to share common semantics throughout all of the OAFEw constituents.

2.3. HLA/RTI Review. HLA (High-Level Architecture) is one of the first priority efforts of the US Department of Defense (DoD) as a common technical standard to promote

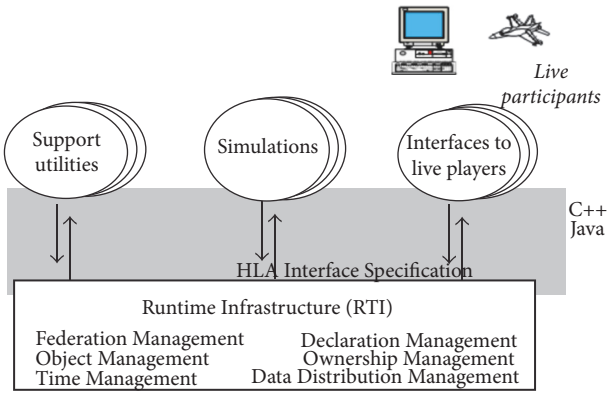

Figure 4: RTI introduction.

interoperability between all kinds of simulations in a network or distributed environment or between C4I (Command, Control, Communication, and Intelligence) systems and weapon systems and reusability of each component.

HLA consist of HLA Rule [7], Federate Interface Specification [8], and Object Model Template [9].

The reuse unit of the simulation software that provides the model implemented for the simulation is called the federate. The multiple federates in a federation communicate with each other via the RTI using a common OMT. Federation aggregates and simulates the entire environment; also each federate's information and state variable can be exchanged.

RTI is a middleware that implements the interface specification of the HLA. It provides a common service of the simulation system as shown in Figure 4 and provides declaration and management of federates.

Within the HLA, federations are comprised of federates that exchange information in the form of objects and interactions–concepts that will be explained further in this guide. The HLA is defined by three components: (1) Federation Rules, (2) the HLA Interface Specification, and (3) the Object Model Template.

The interface specification describes the six service management functions as protocols regarding the functional interface between each federate and the RTI. Each service provides the following functions.

(1) Federation management: it is a service that manages Federation Execution. It provides services such as creation, removal, subscription, withdrawal, synchronization, storage, and recovery of federations.

(2) Declaration management: it manages the subscribe-publish mechanism of shared objects between federates as a service that manages the declaration of state and information exchange between federates belonging to a federation.

(3) Object management: interaction with mock objects in federation creates and discovers messages, updates and reflects objects, deletes, and manages the function of sending and receiving interactive messages.

(4) Ownership management: it is a service related to acquiring or transferring the right to update and delete shared objects to be simulated within a federation.

(5) Time management: it is a simulation time management service that runs in the federation.

(6) Data distribution management: it is a routing service to control the amount of messages exchanged within a federation.

3. OAFEw-Based HLA Federate Development

3.1. EW Reference Conceptual Model. The first implementation of OAFEw is its Reference Conceptual Model [4]. EW Simulation Reference Conceptual Model is a reference model for the conceptual model of the reality that the electronic warfare simulation model wants to simulate. The Reference Conceptual Model also serves as a common semantic dictionary of EW components for constructing a specific conceptual model. The Reference Conceptual Model of OAFEw is defined as a model that describes the EW real world through CML.

This conceptual model includes the entire domain EW M&S. This study was based on the OAFEw conceptual model version 1.0.

Reference Conceptual Model version 1.0 (Figure 5) consists of 4 components: *ECMsystem*, *WeaponSystem*, *Platform*, and *Operator*. Each of the components has events and their behaviors. This shows and explains the process of electronic warfare simulation.

It should be defined for reusable modeling and simulation based on HLA-based federate. This can be defined through a mutual progressive use model.

The most important role of the interoperable usability model is to express the use of the EW domain. Interactive use identifies and standardizes a modeling unit called CML interplay. In addition, this interactive and progressive Use Case Model becomes a building block of the simulation scenario.

3.2. Interplay Use Case Model. OAFEw Rules 3 and 4 define that the use of the Interplay Use Case Model should be identified to be a component of the EW Simulation Specification Model. Use Case Model should be specified in accordance with the template form of the mutual progression of the Base Object Model (BOM) standard [2, 4].

Interplay Use Case Model is composed of a group of interrelated element specifying metadata information, Reference Conceptual Model information, and class information. This information is defined using HLA OMT [2] and mapping between Reference Conceptual Model element and object model elements that identify the class structure.

Use Case Model is intended to reduce the degree of coupling between HLA/RTI and federated layers of HLA/RTI-based distributed simulation. It is the process required to reuse the Federated Object Model (FOM) by providing a way for the user to combine components easily. This allows many distributed interfaces in the domain.

The completed unit BOMs are the unit elements of the FOM for HLA-based distributed simulation. Use Case Model

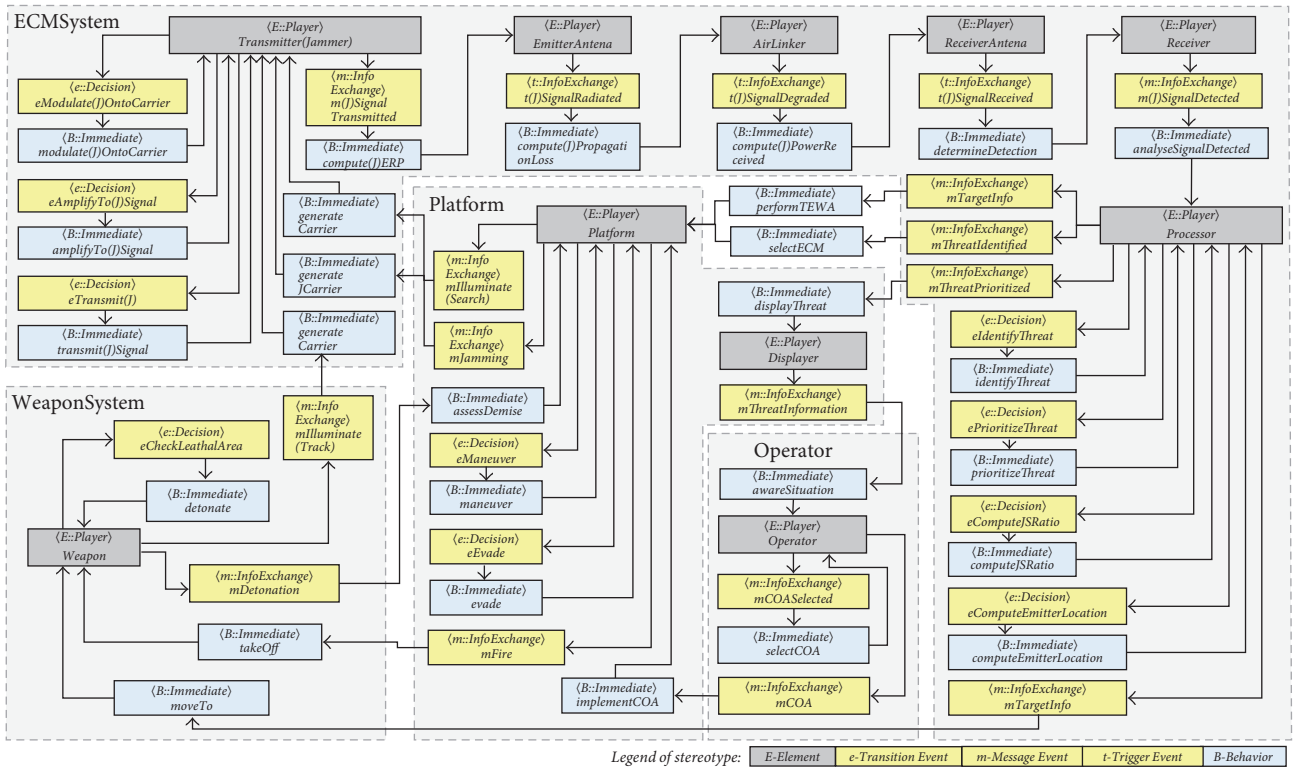

Figure 5: EW Reference Conceptual Model v1.0.

is pattern of interplay: state machine, entity type, and event type definition.

Firstly, Interface Use Case Model of class entity is the class entity component providing a mechanism for describing the types of conceptual entities used to represent "senders" and "receivers" identified within a "pattern of interplay" and carry out the role of conceptual entities identified within a state machine [2].

Secondly, pattern-action template components provide a mechanism for identifying sequences of pattern-action necessary for fulfilling a pattern of interplay, which may be represented by a BOM [11].

Finally, the state machine template component provides a mechanism for identifying the behavior states expected to be exhibited by one or more conceptual entities.

As shown in Figure 6, the OAFEw Use Case Model is the main purpose of defining the action for simulating the system and subsystem for the EW simulation.

The unit of the electronic warfare simulation scenario is called interplay. The Key Interplay required for the OAFEw-based electronic warfare simulation is defined by *Jamming*, *ReceiveSignalWithJaming*, *Receive SignalWoutJam*, and *Illumination*. Pattern-action, state machine, entity type, and event type can be defined according to SISO BOM standard for each action [11].

This Use Case Model is the basic manual for the simulation of EW domain as mentioned above, which becomes the basic unit BOM for reusable FOM for HLA federate.

3.3. Model Mapping (OAFEw-FOM). If the Use Case Model and unit BOMs are created through the OAFEw Reference Conceptual Model, after that, we can obtain FOM for HLA simulation through the developed Use Case Model of the OAFEw.

This FOM is called OAFEw-FOM.

OAFEw-FOM consists of HLA Object class and HLA Interaction. Parameter is data field of an interaction that event sent between simulation entities. Attribute is data field of an object that a collection of related data sent between simulations. HLA Object/Interaction classes are made up through interplay entity type and event type mapping of Use Case Model.

This section introduces how to develop OAFEw-FOM through Use Case Model mapping. The Use Case Model allows us to create HLA Simulation Object Model (SOM) and OAFEw-FOM using two types of entity and event mapping. Each type can be defined in the Reference Conceptual Model.

We define the event type BOM of the Use Case Model by distinguishing the "message event" and "trigger event." The reference concept model of OAFEw defines a trigger event if there is no receiver (or target) and a message event if there is a receiver (or target).

Then, event type mapping can be used to define attributes and parameters of Object or Interaction class according to whether there are messages or triggers.

The BOMs generated for each Key Interplay are used to construct a repository for HLA federate Object/Interplay

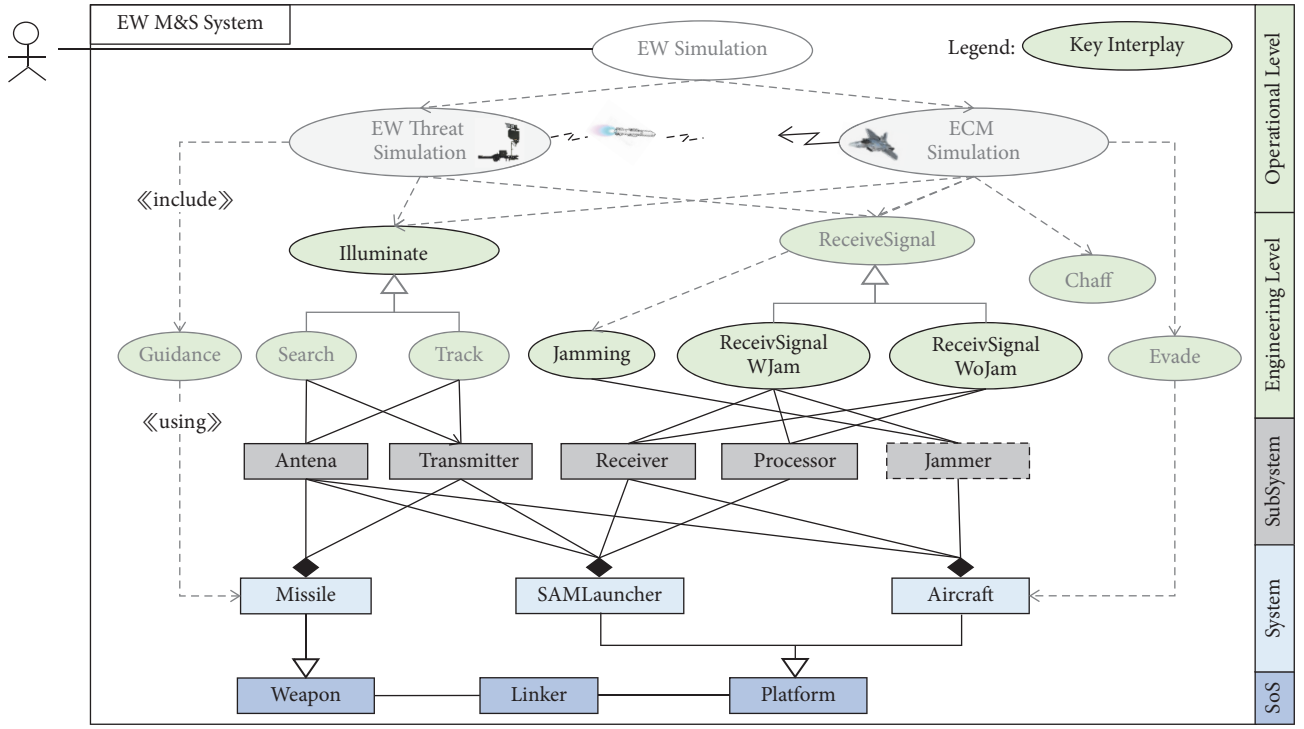

Figure 6: Interplay Use Case Model.

class and attributes/parameters through the model mapping process. This can be referred to as the OAFEw-FOM for electronic warfare simulation HLA federation. It can be used as a dictionary terminology for electronic warfare simulation federate through OAFEw-FOM. Table 1 shows the OAFEw-FOM of *Illuminate* Key Interplay, an example generated through the Reference Conceptual Model and Use Case Model.

The Use Case Model of OAFEw and the OAFEw-FOM of the Distributed Interface Model are specified according to the HLA/OMT format.

In addition, the interaction with objects of HLA is defined using message event or trigger event. Distributed simulation is performed through HLA/RTI based on the OAFEw-FOM defined above.

3.4. FED Generation. OAFEw uses Distributed Simulation Interface based on Interplay Use Case Model. HLA interface disorder occurs when federates with different FOMs want to distribute simulation in a federation. Interplay Use Case Model uses the same HLA/OMT representation format as Distributed Interface Model. The attribute and parameter, which is the data transmission method of distributed simulation through RTI, are classified into Object or Interaction class. Also HLA Object/Interaction class can be reused during the simulation lifecycle without revising.

If the HLA Object/Interaction class definition and attributes/parameters selection for EW simulation are completed through OAFEw-FOM, then FED (Federation Execution Data) file for HLA Federation Execution can be created.

Now we will show how to create a FED file based on Use Case Model. We will describe the HLA OMT to generate the FED file. An OMT is a form for describing a simulation object. All federates of the simulation belonging to the federation are represented according to this template and exchange information and state variable. OMT has to be defined using either the message event or the trigger event in the conceptual model in order to share common semantics throughout all of the OAFEw constituents.

HLA OMT are classified into three types.

First, there is only one SOM per federation, and it describes simulation information. It defines an interpersistent message of the federate and an interaction message. This is achieved through the OMT Model Mapping process using the Use Case Model. For example, characteristics of the fighter, radar, and missile platform corresponding to the system level platform on the electronic warfare simulation can be HLA attributed to an object class. Also event triggers such as *fire* and *explosion* are designed as HLA parameters of Interaction class.

Next, OAFEw-FOM is implemented by describing the exchange attribute/parameter information of federates involved in federation, where there is one OAFEw-FOM per federation. The RTI federate exchanges the state and the information of each federate through the publish-subscribe method, and it describes the declaration about the attribute/parameter sending, receiving, and saving.

Finally, the Management Object Model (MOM) identifies all the attributes and interaction message information present in the federation, it describes the information, and it also shows the state of the federation itself.

The HLA 1.3 version and the 1516 version OMT have essentially the same functionality, but the OMT format is

Table 1: Illumination interplay OAFEw-FOM.

Name	HLA Object/Interaction class	Entity type Characteristics	HLA attribute/parameters
Transmitter Entity	HLAobjectClass. ECMSystem. Transmiter	ID	Transmitter.ID
		Type	Transmitter.RDType
		Location	Transmitter.Location
		Frequency	Transmitter.Frequency
		EmitterPattern	Transmitter.EmitterPattern
		TransmiterPower	Transmitter.TransmiterPower
		TargetDistance	Transmitter.TargetDistance
Emitter Antenna Entity	HLAobjectClass. ECMSystem. Tr_Antenna	ID	EmitterAntenna.ID
		Type	EmitterAntenna.RDType
		Location	EmitterAntenna.Location
		Frequency	EmitterAntenna.Frequency
		EmitterPattern	EmitterAntenna.EmitterPattern
		TransmiterPower	EmitterAntenna.TransmiterPower
		TargetDistance	EmitterAntenna.TargetDistance
AirLinker Entity	HLAobjectClass. ECMSystem. AirLinker	AirLinker	AirLinker

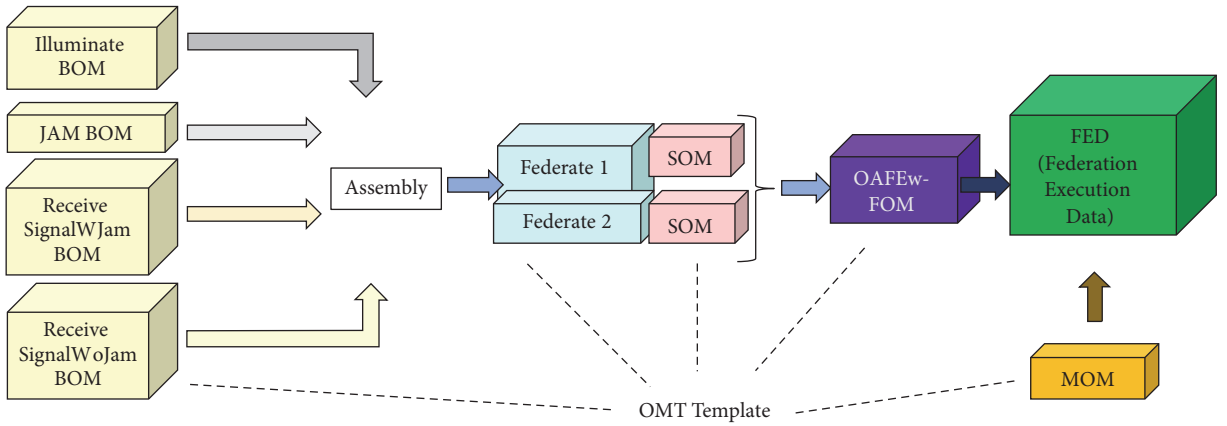

Figure 7: FED generation process.

slightly different. In HLA 1.3, Federation Execution Data (FED) files are used in the standard format and can be extended to fed, .mtl extensions. The 1516 OMT standard format is an FDD (FOM Document Data) file that can be extended with the .xml, .fdd extension. In this study, OMT is designed by defining the class and the attribute values of object based on OAFEw-FOM as FED file format of HLA1.3 version as shown in Figure 7. The information and attributes of federates are described in SOM, and the OAFEw-FOM is configured by combining these SOMs.

After that, OMT was designed and implemented by integrating FOM and MOM and implementing FED file which is the Federation Execution Data.

In this way, it is possible to create a federation with the Federation Execution File based on the OAFEw Reference Conceptual Model terminology dictionary and the FED file generated through the OAFEw-FOM.

4. OAFEw-Based Distributed Interface Model

4.1. Federate Ambassador Design. After designing the FED to define the simulation, we design and implement the simulation management service function for the electronic warfare simulator. The detailed design items of the management service for implementing the component-based interoperability device of the electronic warfare environment are as follows. Figure 8 shows the OAFEw-based HLA/RTI distributed interface flow cycle.

First, we create a Federate Ambassador through the RTI library and get the managed service to be called through the RTI. Then we implement the *RTIAmbassador* requesting that the service was enabled through the Federate Ambassador implemented. The main routine of HLA Distributed Interface for the electronic warfare simulation was designed by the *RTIAmbassador.*

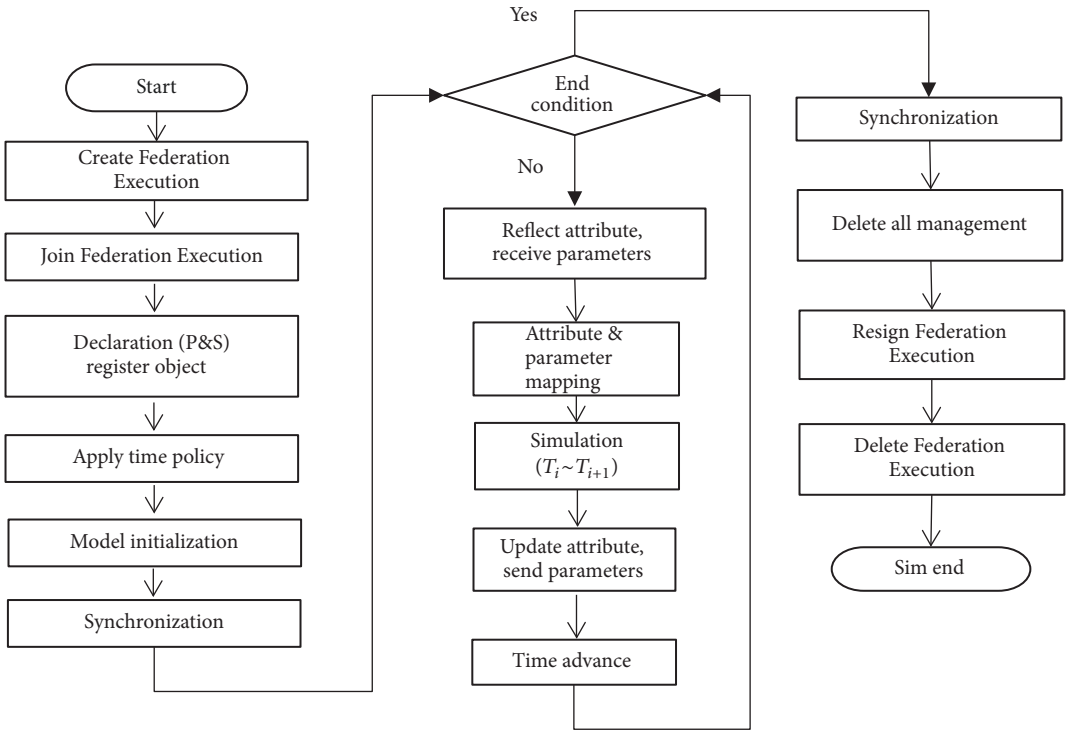

Figure 8: HLA/RTI interface flow chart.

```
// Create the federation
   createFedEx(rtiAmb,
   federationName, federationFile);
// Create the federation
   createFedEx(rtiAmb,
   federationName,federationFile);
// Main Iteration
  while (1)
   {
// Attribute Handle Map Initialize
       theAttrHandleMap.clear();
       theClassHandle =
rtiAmb.getObjectClassHandle("EW_Dat");
// Attribute Mapping
     theAttrNameHandleMap.insert
//Attribute Update
     rtiAmb->updateAttributeValues(
     theObjectHandle,
     *(theAmbData.attrValues),
     (*currentTime),
     tag.c_str());
```

Algorithm 1: OAFEw-FOM based interface module pseudocode.

Subsequently, federates join the federation using the *joinFederateExecution* of the implemented *FederateAmbassador*.

Afterward, if the computation of the models is done up to the end condition, the computing result of the models is transmitted at the different federates. At this time, the result value is exchanged through the *RTIAmbassador*'s Handle in the publish-subscribe manner.

Handle values consist of a key and values are mapped to each instance, so that the key and value can be transmitted together. At this time, if we use the normal *Map* container provided in Visual Studio, there is a problem that the duplicate key value is not recognized as object handles. In order to solve this problem, we implemented *Multimap* container as shown in Algorithm 1 so that even if multiple attributes have overlapping values, we can construct an ordered pair to form handles.

When the message handle to be published is configured, the request for message transmission and time progression is made. At the same time the RTI is called back through the *RTIAmbassador* in accordance with the time progress of the federate managing the simulation time.

One cycle of the computation model is done by updating the attributes with time advance permission. The model calculation is repeated until the simulation end condition.

In order to simulate electronic warfare through Distributed Interface Model, the component and model of the electronic warfare platform are made in dll format, and the variable region values to be used corresponding to OAFEw-FOM are generated.

After that, we compiled the dll file in the Distributed Interface Model and created a real-time distributed simulation.

4.2. Time Management for Real-Time Simulation. In a distributed simulation, each federate has different operating

Table 2: HLA federate time management policy [10].

	Time Regulating	Time Constrained
Time Regulating	(i) It affects other federates' time (ii) It synchronizes the full time progress of the federation with this federation	(i) Other federates affect and are affected by time progress
Time Constrained	(i) Other federates affect and are affected by time progress	(i) This federate is under the influence of regulating federate (ii) This federate can be observer (management)

systems and protocols. If the logical time synchronization of each federate in the real-time and scenario-based simulation is not performed, the causal relation of the logical time is erroneous and the simulation result having the reliability cannot be derived [9].

The federate can request time progress via the RTI. In this paper, we implement the time management service provided by the RTI in the user API (Application Programming Interface) to construct the electronic warfare simulation environment, thereby preventing the error of the logical time of the federates participating in the federation.

The RTI provides two federate time management policies: Time Constrained and Time Regulating. Time Constrained and Time Regulating policy is affected by the time of its simulation time advance from the time of the other federates (Table 2).

In case of Time-Constrained Federate, RTI provides support for receiving different federate Time Stamp Order (TSO) messages. So, in order to prevent causality errors, the simulation time is advanced to the LBTS (Low Boundary Time Stamp) [12].

On the other hand, when Time Regulating policy is applied, RTI reports its TSO message to other federates. Each federate can use all two policies, only one, and none at all. As shown in Table 2, federate can use time management policies to manage the time of federation and total federation.

The designed OAFEw-based Distributed Interface Model implements these two policies as individual functions so that the user can selectively apply the functions through the user interface module. When the user sets the usage policy for the time management of the federate, the federate time policy is applied via the RTI at the start of the simulation. Algorithm 2 shows the pseudocode for the time-step progress request and approval of the federate using time management policy.

When the visual system of the EW simulation is added, it is appropriate to simulate the logical time progressing in a time-step manner at regular time intervals.

Thus, using the time synchronization management of the simulation, all events sent to the other federates at the current time step are calculated and advanced the next time step.

```
//become time constrained, regulating
    rtiAmb->enableTimeRegulation
    ((*currentTime), (*lookAhead));
    rtiAmb->enableTimeConstrained();
//Time Advance
  if ( theAmbData.isRegulating )
    (*currentRTITime)+=(*oneTimeStep);
    (*currentTime)+=(*oneTimeStep);
     rtiAmb->timeAdvanceRequest
     (*currentRTITime);
     theAmbData.timeAdvanced = false;
```

Algorithm 2: Time management pseudocode.

5. OAFEw-Based Distributed Interface Case Study

In order to verify the usefulness of the OAFEw-based Distributed Interface Model proposed in this study, a simple real-time simulated heterogeneous electronic domain as shown in Figure 9 was performed.

The engineering level EW model of OAFEw Use Case Model is implemented by C++ language. In addition, visual system federate and simulation control station federate are implemented using MATLAB/Simulink. Each of the C++ and MATLAB/Simulink federates contains HLA/RTI service and constitutes one federation with the same FED file.

An engagement scenario for an electronic warfare simulation is that friendly fighters penetrate the enemy radar base and explode and return. If a fighter is detected on an enemy search radar, a guided missile is fired and the fighter is tracked. At this time, the fighter emits a jamming signal. The simulation is terminated when the fighter is blown away or escapes for more than a certain amount of time. And the jamming effectiveness depends on the survival rate of the fighter.

The EW Simulation Specification Model is constructed by Key Interplay BOMs as shown in Figure 10. The models designed and implemented are stored and managed by the OAFEw Model Component Library.

As shown in Figure 11, the electronic warfare simulation can be constructed by applying the interoperability device, and the field environment can be simulated and visualized by the interoperability with the HLA/RTI.

This result verified that real-time simulations could be easily performed without expensive and time-consuming reassembly, as is the design purpose of the OAFEw.

6. Conclusions

The various research models developed in the Defense Electronics Specialization Research Center project are modeled according to the environment of development programs of different developers. Projects developers require simulation

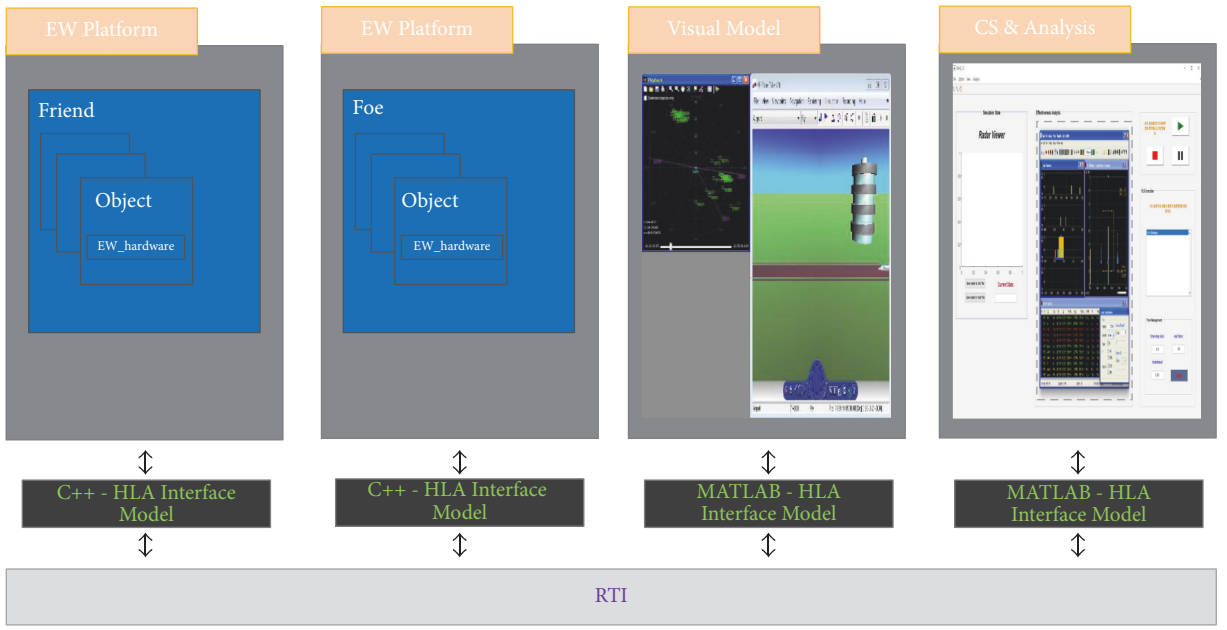

Figure 9: HLA-based C++-MATLAB EW distributed simulation.

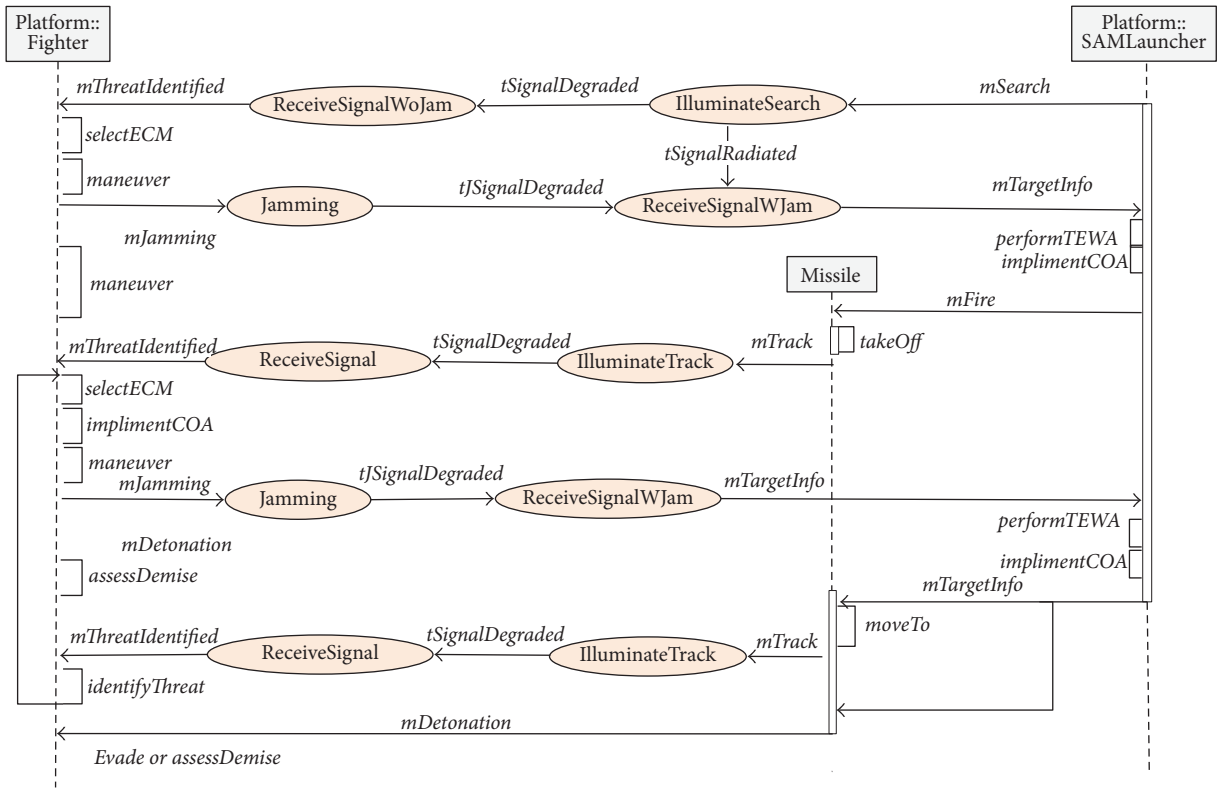

Figure 10: Simulation Specification Model.

models to run in real-time use development tools such as MATLAB or Visual Studio to model and test dynamic systems.

However, in order to simulate a model developed in a different program, the developer needs to modify it. To solve this problem, this study introduces HLA/RTI federate development method based on the OAFEw.

The OAFEw is a simulation lifecycle management tool that guarantees the reusability and interoperability of models developed for electronic warfare simulation. If an OAFEw

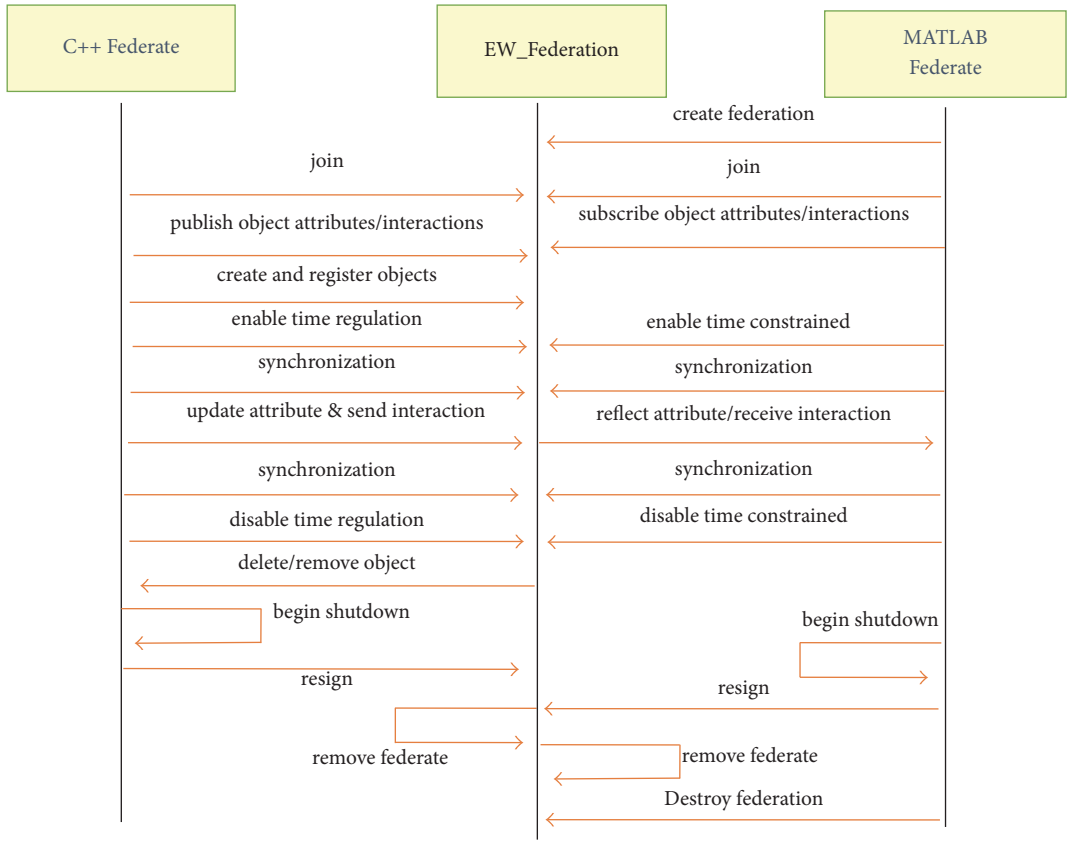

Figure 11: Simulation HLA service diagram.

method is adapted at EW M&S development, real-time distributed simulation of an EW model is facilitated. EW scenarios can be created by reassembling the Key Interplay of the Use Case Model of the OAFEw. Also, if the model is constructed through the OAFEw-Distributed Simulation Interface, there is an advantage that the EW simulation can perform without being restricted to languages such as C/C ++ and MATLAB/Simulink.

We propose the OFEW-FOM based on OAFEw Reference Conceptual Model v1.0 for many developers who want to use the OAFEw's Distributed Interface simulation. The OAFEw-FOM is useful for mapping EW model state machine and input/output variables to HLA attribute/parameter. Federate exchange information OAFEw-FOM also allows developers to set HLA attributes/parameters via a EW common term.

The developed OAFEw-based Distributed Interface Model introduced the application of time management service for reliable simulation between federates. Through this, various models of electronic equipment developed in the future can be simulated in real-time regardless of the model development environment. In addition, for the reusability of the Distributed Interface Model itself, the Interface Model was designed and proposed by structuring it as a user interface module, a HLA service interface module, and an OAFEw-FOM module.

The proposed frameworks and interfaces model were tested for the feasibility and validity through the EW participation scenarios and the EW models.

In the future, it is also necessary to address the problem of constantly requiring code modifications in response to FED changes resulting from updates to the OAFEw Reference Conceptual Model. To do this, we need a code generator that can automatically implement the OAFEw-FOM endecoding module of the proposed framework.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by Agency for Defense Development (ADD) and Defense Acquisition Program Administration (DAPA) through the Gwangju Institute of Science Technology (GIST) of The Electronic Warfare Research Center.

References

[1] J. S. Dahmann, R. M. Fujimoto, and R. M. Weatherly, "DoD high level architecture: An update," in *Proceedings of the 1998 Winter Simulation Conference, WSC. Part 1 (of 2)*, pp. 797–804, December 1998.

[2] S. K. Gupta, G. Siva Prasad, and J. Nanda Kishore, "Electronic warfare simulation-based on service oriented architecture," *Defence Science Journal*, vol. 62, no. 4, pp. 219–222, 2012.

[3] *Electronic warfare in the information age*, Artech House, 1999.

[4] S. Y. Choi, H. S. Kang, H. J. Kyeon, and S. J. Yoon, "An Open Architecture Framework for the Electronic Warfare Modeling Simulation," in *Proceedings of the European Simulation and Modeling Conference (ESM 2017)*, pp. 278–282, October 2017.

[5] M. Ficco, G. Avolio, F. Palmieri, and A. Castiglione, "An HLA-based framework for simulation of large-scale critical systems," *Concurrency and Computation Practice and Experience*, vol. 28, no. 2, pp. 400–419, 2016.

[6] B. Moller, A. Garro, A. Falcone, E. Z. Crues, and D. E. Dexter, "On the execution control of HLA federations using the SISO space reference FOM," in *Proceedings of the 2017 IEEE/ACM 21st International Symposium on Distributed Simulation and Real-Time Applications (DS-RT)*, pp. 1–8, Rome, October 2017.

[7] *IEEE, IEEE Standard for Modeling and Simulation(MS) High Level Architecture(HLA)-Framework and Rules , IEEE Std.1516*, 2000.

[8] *IEEE, IEEE Standard for Modeling and Simulation(MS) High Level Architecture(HLA)-Federate Interface Specification , IEEE Std.1516.1*, 2000.

[9] IEEE., "Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Object Model Template (OMT)," *2*, 2000, IEEE Std.1516.

[10] R. M. Fujimoto, "Time Management in The High Level Architecture," *Simulation*, vol. 71, no. 6, pp. 388–400, 1998.

[11] *SISO(Simulation Interoperability Standard Organization, Base Object Model(BOM) Template Specification , SIOS-STD-003*, March 2006.

[12] R. Fujimoto, A. Malik, and A. Park, "Parallel and distributed simulation in the cloud," *SCS MS Magazine*, vol. 3, p. 10, 2010.