Multifaceted Modeling and Simulation Framework for System of Systems Using HLA/RTI

Byeong Soo Kim, Chang Beom Choi and Tag Gon Kim
Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
Daejeon, KOREA
(E-mail : bskim@smslab.kaist.ac.kr, cbchoi@smslab.kaist.ac.kr, tkim@ee.kaist.ac.kr)

Keywords: System of systems, Multifaceted modeling and simulation, SES/MB, HLA/RTI

Abstract

The multifaceted system modeling method represents all the components and alternatives of a system. As one of these methods, the System Entity Structure/Model Base (SES/MB) enhances the organization of model families as well as, storing and reusing model components. However, the real world can be described not only as an individual system but also as a collection of those systems, which is called a system of systems. Because SES/MB is limited in simulating the system of systems using HLA/RTI, an extended framework is required to simulate the system of systems. This paper proposes a System of Systems Entity Structure/Federate Base (SoSES/FB) framework for simulation in a distributed environment. The proposed framework provides a library of simulators (FB) and SoSES formalism, which represents structural knowledge of a collection of simulators. It also provides a new federation synthesis process. The paper introduces a simulation of a warship's anti-missile defense system using the proposed SoSES/FB framework.

1. INTRODUCTION

A real-world system has various structures and alternatives, not a fixed structure. For example, a computer system consists of a process, main memory, secondary memory, and input-output device. The main memory consists of RAM and ROM, and the secondary memory has several alternatives like HDD and SSD. Multifaceted System Modeling is a modeling method that represents all the components and alternatives of a system [1]. One of these multifaceted modeling methods is the System Entity Structure/Model Base (SES/MB) framework [2]. The SES/MB framework constructs the structural knowledge of the system through SES formalism, and completes an executable simulator by synthesis of models in MB.

It is possible to simulate a standalone system through the SES/MB framework, but the real world cannot always be described with a standalone system. The real world is described with system of systems that is a collection of the standalone systems. For example, an independent computer system is considered a standalone system, while the system that is connected through the network is considered a system of systems. A system of systems is a collection of independent systems that pool their resources and capabilities together to create a new, more complex system that offers more functionality and performance than simply the sum of the constituent systems.

Such a system of systems usually operates in a distributed environment, so an SES/MB framework for a standalone system cannot be used to simulate a system of systems. A new, multifaceted modeling and simulation framework is needed to simulate a system of systems in a distributed environment.

Therefore, this paper proposes a System of Systems Entity Structure/Federate Base (SoSES/FB) framework based on High Level Architecture/Run Time Infrastructure (HLA/RTI) to simulate a system of systems in a multifaceted modeling simulation method. By extending the SES formalism, this paper first proposes an SoSES formalism that is suitable for the modeling of system of systems. The SoSES/FB framework contains a SoSES tree to represent structural knowledge of a system of systems and a library of simulators to complete an interoperable simulator. It also provides a new federation synthesis process by applying the SoSES/FB framework. The SoSES/FB framework increases the reconfigurability of the system of systems, and provides the multifaceted modeling and simulation in the distributed environment.

This paper is organized as follows. Section 2 presents the SES/MB framework and HLA/RTI, and Section 3 explains the proposed SoSES formalism. Section 4 describes the SoSES/FB framework based on SoSES formalism and the proposed federation synthesis process. Section 5 applies the SoSES/FB framework to the simulation of a warship's anti-missile defense simulation. Finally, Section 6 concludes this paper.

2. RELATED WORK

2.1. SES/MB

The SES/MB framework is used for the multifaceted modeling simulation of a standalone system. Figure 1 shows the outline of the SES/MB framework. The SES/MB consists of the SES Base and Model Base. The SES Base stores SES trees, which have the structural knowledge of a system and the Model Base stores real models that represent the behavioral knowledge of a system. A user can synthesize a complete simulator using the SES tree and models in each base. With the SES/MB framework, a user can formally model and develop a system through the automatic synthesis process. Also, the framework can improve the reusability of the existing models [3].

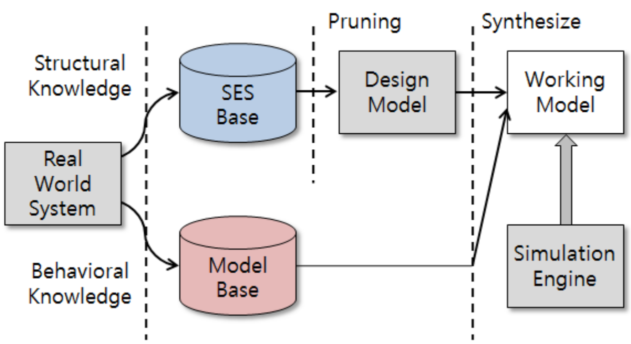


Figure 1. Outline of SES/MB framework

Several research projects related to SES/MB have been conducted over the years. One of these studies produced a knowledge-based model automatic synthesis methodology that represents SES using Scheme and uses DEVS-Scheme as a simulation environment [4]. There is database performance analysis research that represents SES using DB SQL and uses DEVSim++ as a simulation environment [5]. Also, there is a digital system design which represents SES using DB SQL and uses SystemC as a simulation environment [6].

Research related to SES/MB was performed from various angles using various languages and environments. The SES/MB framework was applied to the various industry fields [7][8].

These research studies all targeted the standalone system, while there is no study of the system of systems applying the multifaceted modeling method.

2.2. HLA/RTI

A system of systems consists of several independent systems that need to interoperate: data must be exchanged among systems, and they must be used in each system. The High Level Architecture (HLA), which is approved as the IEEE 1516 standard, is a specification for interoperation among distributed heterogeneous simulations. The HLA consists of three parts: the HLA Framework and Rules, the Federate Interface Specification and the Object Model Template (OMT). The Run-time Infrastructure (RTI) is software implementing the HLA specification.

A standalone simulator for the HLA standard is called a federate. The set of federates is called a federation in the HLA [9]. The Federation Object Model (FOM) is a collection of all shared objects and interactions defined inside a federation. The Simulation Object Model (SOM) is a list of shared objects and interactions that a federate exchanges [10]. The SOM includes exchanging data with other federates. The FOM, the set of the SOMs, constitutes federation data [11].

Many various simulators have been developed using HLA/RTI. This paper proposes an SoSES/FB framework for multifaceted modeling and simulation using the existing simulators.

3. SOSES FORMALSIM

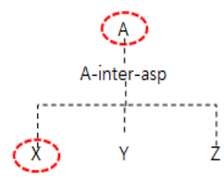
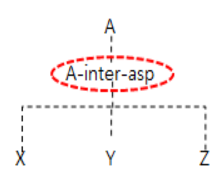
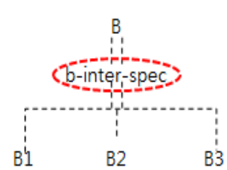
This section proposes the SoSES formalism for a system of systems, which consists of independent systems. The SoSES formalism is an extended version of SES formalism that enables it to simulate the system in a distributed environment, HLA/RTI.

The SoSES formalism is based on SES and represents the decomposition, coupling, and taxonomy information of a system of systems. The SoSES formalism consists of representation and operation, and operation consists of nodes and data.

3.1. Representation of SoSES

SoSES formalism provides three types of node to represent the structure of the system. The types of node are shown in Table 1.

Table 1. Type of SoSES Node

	Representation	Meaning
System node		A, X, Y and Z mean each system
Inter-aspect node		A is composed of X, Y and Z
Inter-specialization node		B is specialization in B1, B2 or B3

The system node corresponds to a real system, and it means a federate or a federation. A root node is a system node, which means a federation (a system of systems). An inter-aspect node represents the decomposition relationship of a system node. An inter-specialization node represents the taxonomy relationship of the system of systems. The taxonomy relationship means the alternatives of the system of systems.

These three types of nodes, the "data" is needed to represent the coupling relationship of the system of systems. The data contain two pieces of information defined in HLA: object and interaction. We use the object when the data are changeable in time and maintaining the changed data requires sharing among federates. The object has attributes as details. The interaction data need not be maintained as the time goes by. The interaction has some parameters as details. In addition, the coupling relationship among federates must be specified. We designate "publish" which means input data of HLA/RTI as [P], and "subscribe" which means output data of HLA/RTI as [S]. In case both input and output data are required, we specify both [P] and [S]. For example, '[P] Object X {x1, x2, x3}' means that the object X has the attribute x1, x2, and x3, and the object will be published.

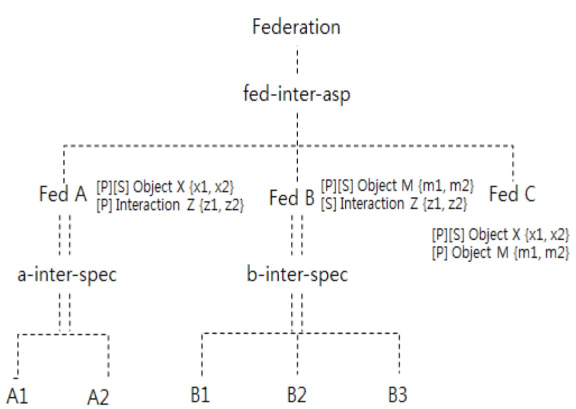


Figure 2. An example of an SoSES tree

In the SES formalism, the relationship of input–output port is specified to deliver a message. But SoSES formalism needs to specify the object besides the interaction, which means a message. An example of a tree constructed using the SoSES formalism is shown in Figure 2.

3.2. Pruning

A federate must be selected from among several alternatives after constructing a SoSES tree. It is essentially a pruning operation. Concretely, pruning is a process to select a system node among child nodes of an inter-specialization node representing taxonomy relationship.

If a user selects all the child nodes of the inter-specialization nodes, the selected child node is substituted for the parent node of the inter-specialization node. Then, the inter-specialization node and the unchosen child nodes are deleted.

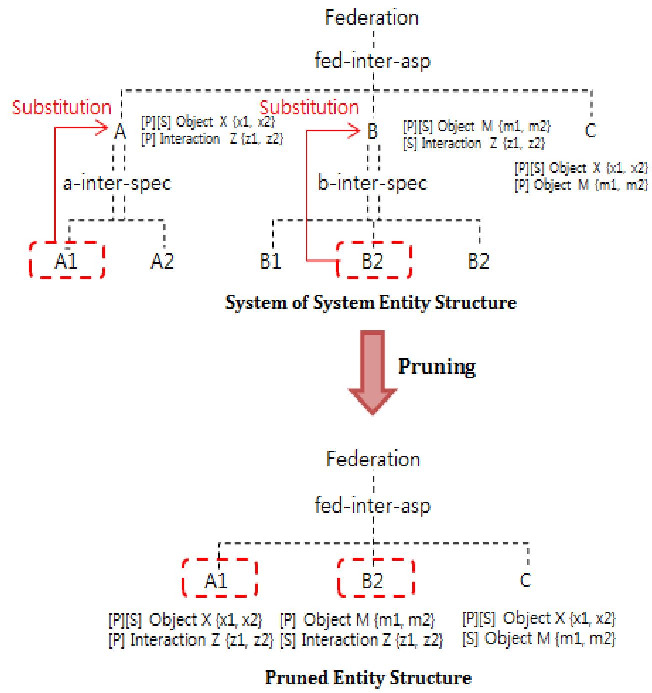


Figure 3. An example of pruning operation

An example is shown in Figure 3. In the SoSES tree, A1 and A2 are the child nodes of the a-inter-spec node, and B1, B2, and B3 are the child nodes of the b-inter-spec node. If a user selects A1 and B2, the node A is substituted into the node A1, and the node B is substituted into the node B2. The unchosen system nodes and the inter-specialization nodes are removed from the tree, and then the SoSES tree consists of system node A1, B2, C and fed-inter-asp node.

3.3. Data Synthesis

Data synthesis is an operation about the coupling relationship of system of systems. After pruning, the data of selected federates is combined by data synthesis algorithm.

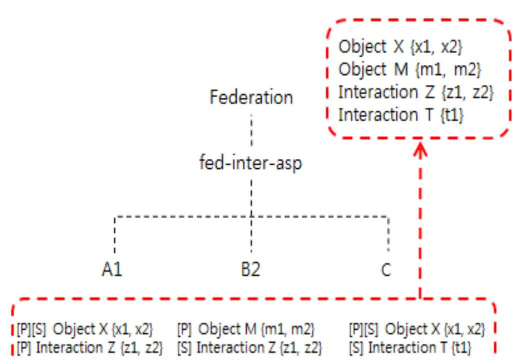


Figure 4. An example of data synthesis

Figure 4 shows the concept of the data synthesis. Data of the system nodes A1, B2, and C are combined. At this time, overlapping data are treated as one datum. The overlapped data should have the same data type (object and interaction) and same data name. Each object (or interaction) should have same attribute (or parameter) name and number.
The data synthesis algorithm also checks whether the data are published or subscribed. When heterogeneous simulators interoperate with each other, data exchange is required between federates. If there are no data to exchange between federates, there is no necessity to interoperate. In other word, if a federate publishes data, there must exist a federate that subscribes to the data. If there is at least one publish-subscribe pair, then the process will progress correctly. If these conditions are satisfied, FOM is generated. Algorithm 1 shows the data synthesis algorithm.

```
Input : ESi (Node of System of Systems Entity Structure)
Output : Synthesized FOM
STACK is initially empty stack
push(STACK, ESi)
while STACK is not empty then
    ESj = pop(STACK)
    Pes = ESj.getPublishSet()
    Ses = ESj.getSubscribeSet()
    for each element of Pes Pi ∈ Pes
        for each element of Pubi ∈ Pub
            for each element of Pubi Pj ∈ Pubi
                check structure of Pi and Pj
                if they are identical
                    insert Pi into Pub
                else
                    error
            end for
        end for
    end for
    for each element of Ses Si ∈ Ses
        for each element of Subi ∈ Sub
            for each element of Subi Sj ∈ Subi
                check structure of Si and Sj
                if they are identical
                    insert Si into Sub
                else
                    error
            end for
        end for
    end for
    for each element of Pub Pubi ∈ Pub
        for each element of Sub Subi ∈ Sub i≠j
            for each element of Pubi Pk ∈ Pubi
                for each element of Subj Sl ∈ Subj
                    check structure of Pk and Sl
                    if they are identical
                        insert Pk into MPubi
                        insert Sl into MSubj
                    end if
                end for
            end for
        end for
    end for
    if MPubi and MSubj is empty
        error
    else
        for each element of Pub Pubi ∈ Pub
            for each element of Sub Subi ∈ Sub
end while
```

Algorithm 1. Data Synthesis Algorithm

4. SOSES/FB FRAMEWORK

The SoSES/FB framework shown in Figure 5 consists of a SoSES Base and a Federate Base. The SoSES Base stores the SoSES tree and Federate Base stores HLA-compliant federates. The SoSES/FB framework also contains an SoSES manager, federate manager, and synthesizer.

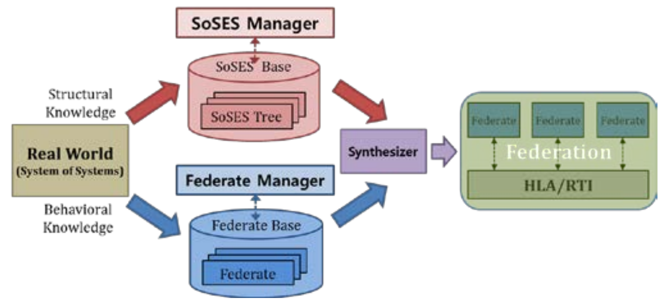


Figure 5. Outline of SoSES/FB framework

The SoSES Base stores SoSES trees constructed from SoSES formalism and the Pruned Entity Structure (PES) tree after the pruning operation. The SoSES Base has the information of federates, while the Federate Base has executable HLA-compliant federates and SOM data (Figure 6). When a federation is synthesized after the pruning operation, system nodes refer to node names and load executable federates in the Federate Base.

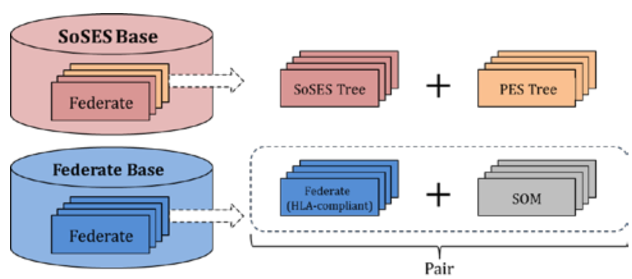


Figure 6. SoSES Base and Federate Base

A SoSES manager constructs and manages a SoSES tree. The SoSES manager contains a tree manager for managing a tree structure, a pruner for pruning operations, and a data synthesizer for data synthesis. An SoSES tree can be constructed by receiving a federate list from the Federate Base. Also, we can use an SoSES tree constructed by domain experts from SoSES Base. The SoSES manager judges the suitability of the SoSES tree based on SoSES formalism. A pruner receives an SoSES tree from the SoSES manager, and performs a pruning operation. Then a data synthesizer makes a FOM using a data synthesis algorithm.

A federate manager manages federates in the Federate Base. The federate manager sends the federate information in the Federate Base to the synthesizer. Alternatively, the federate manager receives the result of the SoSES manager (federate information and FOM) and then executes each federate based on federate information.

4.1. Federate Synthesis Process

A new federation synthesis process based on FEDEP (Federation Development and Execution Process) [12] is shown in Figure 7. This paper proposes a process applying only the federation implementation and execution level of FEDEP using an SoSES/FB framework.

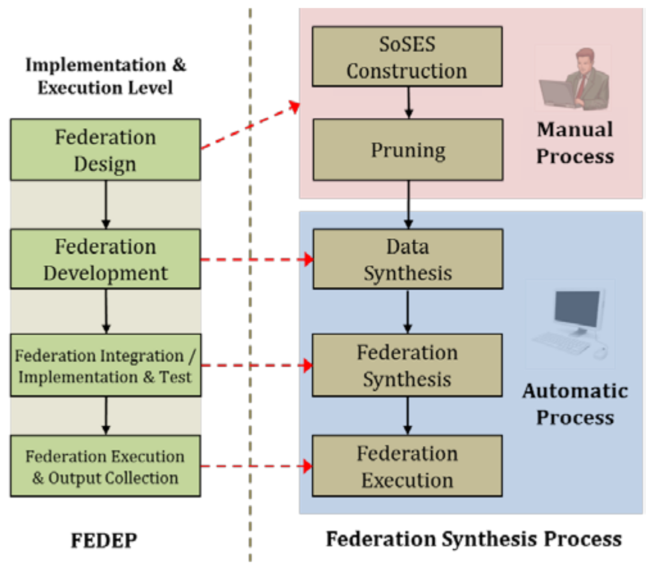


Figure 7. Federation synthesis process

After defining the federation objective and analyzing the federation concept, a federation design level is required. In the federation design level achieved passively by a user, we can construct and prune the SoSES depending on the objectives and requirements provided by the federation analysis. The SoSES construction is developed directly by a user or selected from the SoSES Base. The finished SoSES tree from this process provides the information about alternatives, composition, and coupling.

Next is the pruning process. We can select the federate from among several federates depending on the federation objective, and then the pruning proceeds by means of the pruning algorithm. From the pruning process, we can get the information of the required federates and the data. In the data synthesis process corresponding to the federation development process of FEDEP, we can automatically construct a FOM required by the federation. There is no need to develop each federate on the basis of existing HLA-compliant federates. Resulting from the data synthesis process, the interoperation possibility is checked by a data synthesis algorithm, and a FED file is generated.

The next level is the federation synthesis and execution process. After the previous processes, the synthesizer sends the federate information and the FOM to the federate manager, connecting through the network. At the same time, the synthesizer gives an order to the federate manager to execute the given federate. Then, each federate manager executes the federate in Federate Base. After that, the synthesizer gives orders to create, join, and start the federation. Consequently each federate starts the simulation automatically.

5. CASE STUDY

This section presents an example of modeling and simulation using a proposed SoSES/FB framework. The process follows for a proposed federation synthesis process. The warship anti-missile defense simulation developed to use in a distributed environment is used to apply the SoSES/FB framework. All federates of the warship simulation are HLA-compliant.

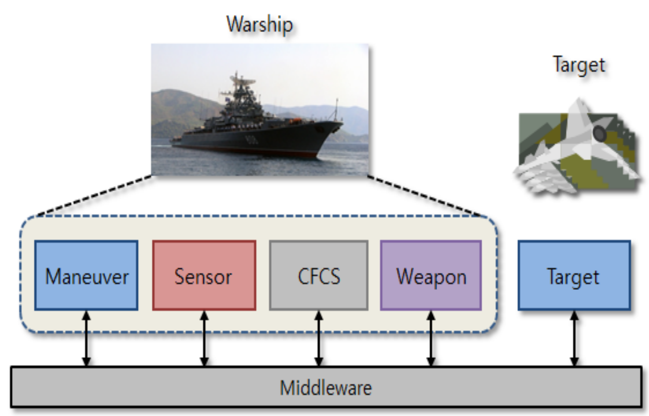


Figure 8. Scenario of warship simulation

The scenario of warship simulation is as follows (shown in Figure 8). A target is flying toward the warship. Radar in the warship pinpoints the location of the target and warship, and delivers it to the warship. The warship judges whether it will attack the target according to the distance between them and orders to attack the target. If the warship attacks the target, it is decided whether the target is shot down. From this simulation, we get the survival ratio of the warship [14]. The warship simulator consists of Warship and Target. And the warship consists of Maneuver, Sensor, Command and Fire Control System (CFCS), and Weapon federates.

5.1. Federate Synthesis Process

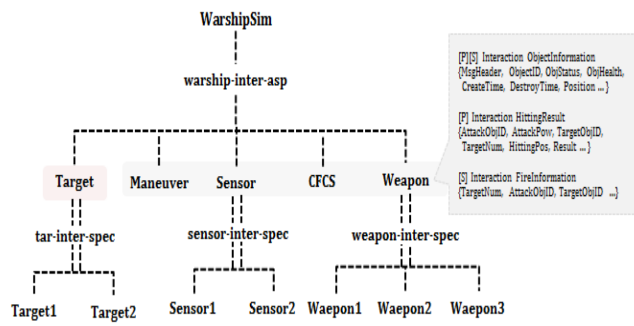


Figure 9. SoSES tree of warship simulation

In the warship simulation, each federate has several alternatives. For example, there are various sensor federates because there are various sensing algorithms and inner structures. As shown in Figure 9, the compositions, alternatives, and couplings of the warship simulation are represented by SoSES formalism.

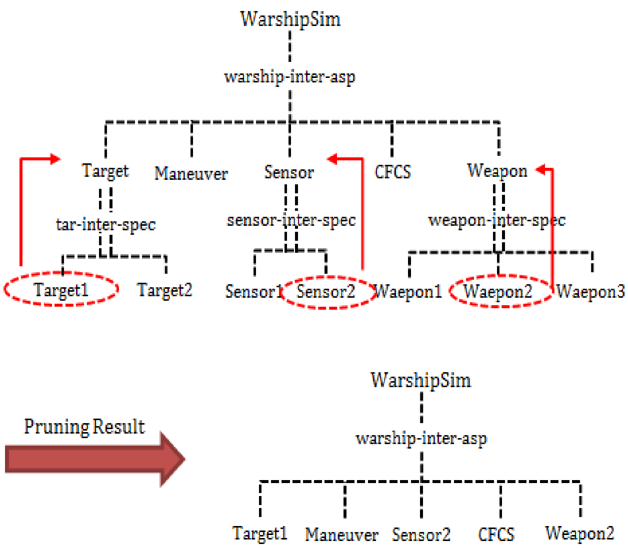


Figure 10. Pruning operation of warship simulation

After constructing the SoSES tree, a user selects a node among several alternatives (pruning operation). In this section, we select Target1, Sensor2, and Weapon2, and we get the PES shown in Figure 10. Each federate datum is merged into a FOM using the data synthesis algorithm after pruning. Figure 11 shows the data synthesis process of the warship simulation. Then, the synthesizer sends the federate information and the FOM of the warship simulation to the federate managers. Each federate manager executes the corresponding federate. Then, the synthesizer gives orders to create, join, and start the federation (shown in Figure 12). Consequently, the warship simulation is executed, and we can get the survival ratio of the warship.

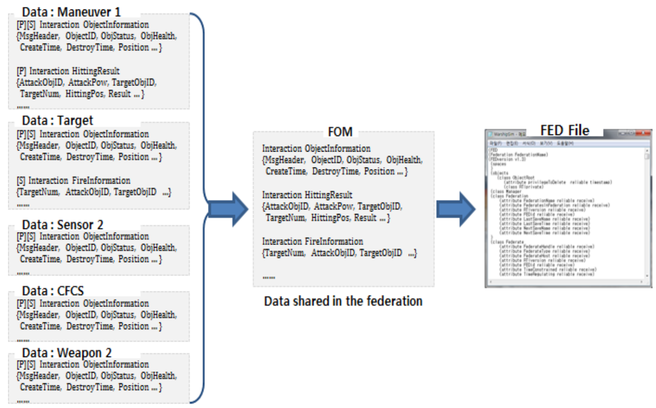


Figure 11. Result of data synthesis process

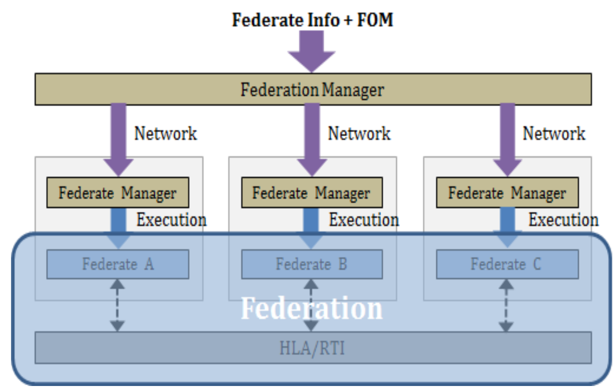


Figure 12. Federation synthesis

6. CONCLUSIONS

This paper proposes an SoSES/FB framework for simulations using HLA/RTI. The existing SES/MB framework cannot simulate a system of systems because it is suitable only for a standalone system. So, there is a need for a new framework to simulate a system of systems using HLA/RTI. This paper proposes SoSES formalism to complete the SoSES/FB framework and proposes an SoSES Base and Federate Base to store structural and behavioral knowledge of the system of systems. Using this SoSES/FB framework, this paper also proposes a new federation synthesis process.

By applying the SoSES/FB framework to the warship simulation, we know that this simulation behaves similarly to a federation that is synthesized, following FEDEP. The SoSES/FB framework provides user-friendliness by automating the federation synthesis process. The SoSES formalism increases the efficiency in managing the structure of the federate, and it is easy to simulate a system of systems which has several alternatives by changing the structure. In this paper, the SoSES/FB framework uses existing HLA-compliant federates, but it gives a restriction to use other federates. Further work should consider the automatic creation of interfaces of federates.

Acknowledgement

This work was partially supported by Defense Acquisition Program Administration and Agency for Defense Development under the contract. (UD110006MD)

References

[1] Bernard P. Zeigler. 1984.Multifacetted Modelling and Discrete Event Simulation. ACADEMIC PRESS.

[2] T. G. KIM, C. LEE, E. R. CHRISTENSEN and B. P. ZEIGLER. 1990. "System Entity Structuring and Model Base Management," IEEE Transactions on Systems, Man and Cybernetics, Vol. 20, No. 5,1013-1024.

[3] Wan Bok Lee. 1994. "Development of The Multifacetted System Modelling / Simulation Environment," Master Thesis, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea.

[4] T. G. Kim and B. P. Zeigler. 1989. "A Knowledge-Based Environment for Investigating Multicomputer Architectures," Journal of Information and Software Technology, Vol. 31, No. 10,512-520.

[5] Hyu Chan Park, Wan Bok Lee and Tag Gon Kim. 1997. "RASES: A Database Supported Framework for Structured Model Base Management," Simulation Practice and Theory, Vol. 5, No. 4, 289-313.

[6] Jung K. Kim and Tag G. Kim. 2006. "A Plan-Generation-Evaluation Framework for Design Space Exploration of Digital Systems Design," IEICE Transactions on Fundamentals of Electronics, Vol. E89-A, No. 3, pp. 772-781.

[7] Mittal S., Mak E., and Nutaro J.J. 2006. “DEVS-Based Dynamic Model Reconfiguration and Simulation Control in the Enhanced DoDAF Design Process,” Journal of Defense Modeling and Simulation, Vol. 3, No. 4.

[8] Saehoon Cheon, Doohwan Kim and Bernard P. Zeigler. 2008. “System Entity Structure for XML Meta Data Modeling: Application to the US Climate Normals,” 17th International Conference on Software Engineering and Data Engineering, Los Angeles, CA, USA.

[9] IEEE Std. 1516-2000 2000. IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules, IEEE Computer Society.

[10] IEEE Std. 1516-2000 2001. IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Object Model Template (OMT), IEEE Computer Society.

[11] Tag Gon Kim. 2011. IE 801 Lecture Note, Industrial & Systems Engineering, KAIST, http://sim.kaist.ac.kr/.

[12] IEEE Std. 1516.3-2003 2003. IEEE Recommended Practice for High Level Architecture (HLA) Federation Development and Execution Process (FEDEP), IEEE Computer Society.

[13] Bernard P. Zeigler, Herbert Praehofer and Tag Gon Kim. 2000. Theory of Modeling and Simulation. ACADEMIC PRESS.

[14] Chang Ho Sung, Jeong Hee Hong and Tag Gon Kim. 2009. "Interoperation of DEVS Models and Differential Equation Models using HLA/RTI: Hybrid Simulation of Engineering and Engagement Level Models," 2009 Spring Simulation MultiConf., San Diego, CA, USA.

Biography

Byeong Soo Kim is a Ph D. candidate at the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST). His research interests include methodology for M&S of discrete event systems (DEVS) and multifaceted modeling simulation.

Chang Beom Choi is a Ph D. candidate at the Department of Electrical Engineering Korea Advanced Institute of Science and Technology (KAIST). His research interests include discrete event systems (DEVS) and verification methodology.

Tag Gon Kim is a Professor at the Department of Electrical Engineering Korea Advanced Institute of Science and Technology (KAIST). He was the Editor-In-Chief for Simulation: Transactions for Society for Computer Modeling and Simulation International (SCS). He is a co-author of the text book, Theory of Modeling and Simulation, Academic Press, 2000. He published about 200 papers in M&S theory and practice in international journals and conference proceedings. He is very active in defense modeling and simulation in Korea.