Research and Implementation of Distributed Simulation System Based on HLA

Weidong Zhang(✉), Yun Xiao, and Ni Lei

Wuhan Maritime Communication Research Institute, Wuhan 430079, China
67599891@qq.com, 63937073@qq.com, 57512385@qq.com

Abstract. Distributed simulation technology based on HLA is an important direction of simulation development. Based on the HLA technology, this paper put forward an idea about establishing Federation between two different simulating platforms. We developed an interface protocol to communicate with HLA service for Federate, and several network protocols on EXata to simulate the networking process. Finally, the EXata which is good at protocol modeling and the Eagle which is good at movement modeling could interactive through a HLA interface and join in a Federation. In this way, the collaborative simulation of two platforms is realized.

Keywords: HLA · Distributed simulation · EXata

1 Introduction

Distributed simulation is the product of the combination of simulation technology and network technology development [1]. For a large network or a target network with distribution requirements, the distributed simulation method can be used. In this way, the target network can be distributed to multiple computers or simulators, with each one implementing parts of the functions for the network, and we can finally realize interactive dynamic simulation through distributed technologies.

Distributed simulation technology originated from the SIMNET research project which is developed by Defense Advanced Research Projects Agency (DARPA) and the U.S. army in 1983. The United States department of defense modeling and simulation office (DMSO) which is responsible for the simulation in military field put forward the concept of high-level architecture (HLA) in March 1995, and released the HLA specification in August 1996. HLA is a new framework of simulation technology, which defines the relationship between each part of the simulation function, rather than a series of data exchange standards. It defines the rules of the whole process from a higher level of simulation development, modeling and designing.

This paper introduces the basic concept of HLA technology, and realizes a collaborative simulation of two simulation platforms which are good at different simulation functions based on a Federation.

G. Sun and S. Liu (Eds.): ADHIP 2017, LNICST 219, pp. 109–116, 2018.
https://doi.org/10.1007/978-3-319-73317-3_14

2 Distributed Simulation System Based on HLA

The HLA standard defines the concepts of Federation and Federate. A simulation system combined with simulation subsystems is defined as Federation, and every simulation subsystem is defined as Federate. The core of HLA standard is the HLA rules, Object Model Template (OMT), and interface specification.

According to the HLA rules, all the Federation and Federates should provide their object model following the OMT that are Federation Object Model (FOM) and Federates' Simulation Object Model (SOM). The function of OMT is to provide a standard and documented form which can describe the information of the Federation and Federates' object model. As parts of modeling and simulation repository, both FOM and SOM were stored in the according database and can be used or reused during the executing process [2–6].

2.1 System Composition

Based on the HLA distributed simulation technology [7], this paper describe how to develop HLA interface and functions for EXata (a communication simulating software) and Eagle (a simulation modeling software), in order to make a Federation and realize a system through object type releasing and reflecting method. The system designed as below (Fig. 1).

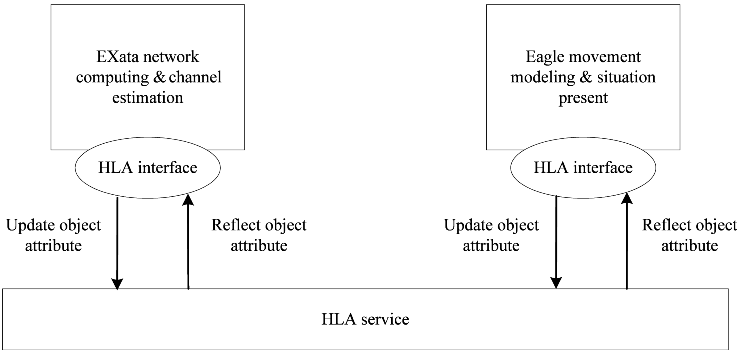

Fig. 1. System composition.

2.2 Functions of Subsystems

EXata mainly completes the work of protocol modeling and course calculating, with its functions as below:

(1) Build up a communication network in the simulate network environment based on a standard or self-defined protocol model, and simulate the whole process of the dynamic executing.
(2) Build up an application modeling based on a currency application model.

(3) Capable for HLA interface, and capable for building Federation, joining Federation, subscribing and releasing.
(4) Update and synchronize of position based on node positon data from HLA service through the process of simulation.
(5) Release the application data from HLA service through the process of simulation.
(6) Compute the network protocol process based on the node positon data, and release the node connection data from HLA service.

Eagle mainly completes the work of opera movement modeling and result rendering, with its functions as below:

(1) Capable for node position and movement modeling.
(2) Capable for HLA interface, and capable for building Federation, joining Federation, subscribing and releasing.
(3) Release the node positon data from HLA service.
(4) Reflect and appear the node connection state from HLA service through the process of simulation.
(5) Reflect and appear the application data from HLA service through the process of simulation.

3 System Implementation

3.1 Runtime Environment

The distributed simulation system adopted MAKRti ver. 4.1.1f as the operation support environment. After MAKRti is installed successfully, there would be a *MAK_RTIDIR* variable added with its value equaling to the installation path of RTI software. Meanwhile, variable *PATH* will include the path to RTI's lib folder and bin folder. Also we need to modify the value of "*setqb RTI_useRtiExec*" from 0 to 1 in the *rid.mtl*, or there would be a mistake when operating.

3.2 Federation Design

This system is developed based on the standard HLA1.3 interface protocol. EXata and Eagle join the same Federation with a name as "*DISP-SYS*", following by the file as "*DISP-STRIVE-RPR.fed*". Besides, every Federate should have its own Federation name as the flag to join the Federation.

3.3 Federation File

The federation file "*DISP-STRIVE-RPR.fed*" in this system is modified based on the basic FOM file form. In the file, we defined the object class and interaction class both are one-way inheritance tree, with the root class of the object class is *ObjectRoot*, and the root class of the interaction class is *InteractionRoot*. A complete name of a class is composed by the name of the root through to the class, separated by "." in the middle.

In addition, the federal file has some necessary object classes for management issues, such as *privilegeToDeleteObject*, etc.

We use some defined classes and create new object class, object class attributes, and interaction classes that we need as below:

(1) ObjectRoot.BaseEntity.WorldLocation
(2) Object-Root.BaseEntity.PhysicalEntity.DamageState
(3) ObjectRoot.BaseEntity.Buffer

3.4 HLA Workflow

We developed a workflow for Federate as shown in the following Fig. 2.

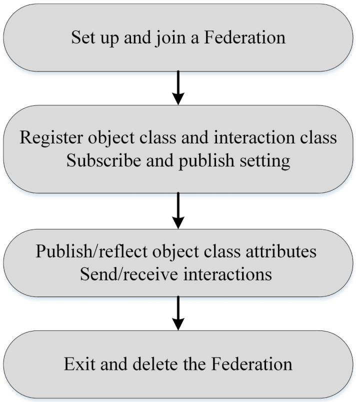

Fig. 2. HLA workflow.

Before the Federation starts running, Federate need to have the function of setting up the HLA configuration, including setting the Federation name, Federation file name, Federate name, and other required files which can be read when the system starts running. At the same time, before the Federate begins to run, it needs to start the RTI software.

After the Federate starts, one of them is supposed to invoke the RTI service "*createFederationExecution*" to establish the Federation, and other Federate will find that the Federation has been established and would not establish the Federation again. Establishing a Federation requires inputting the common Federation file name and the path to the Federation file. Every Federate needs to invoke the RTI service "*joinFederation-Execution*" to join the Federation. After joining the Federation successfully, you can see the established Federation name in the RTI software and the membership of the Federation.

Thereafter, Federate can register their associated object classes, object class attributes, and interaction classes, subscribe to related object class attributes or interaction classes as planning, and publish or send attributes or interaction classes with a publishing authority. Federate gets the object class handle from the RTI service "*getObjectClassHandle*", the attribute handle from "*getAttribute-Handle*", and the interaction classes handle from "*getInteractionClassHandle*". Federate subscribes from the RTI service "*subscribeObjectClassAttributes*" and "*subscribe-InteractionClass*". At last, Federate publish from the RTI service "*publishObjectClass*" and "*publishInteractionClass*". After Federate publish to RTI with corresponding attributes and interaction, RTI will inform the Federate who had subscribed the very attributes. The Federate can receive the message through a callback function. Federate can update object class attributes by invoking RTI service "*updateAttributeValues*", and send interaction by "send-Interaction".

RTI has developed the callback function prototype which should be implemented by Federate themselves, including "*discoverObjectInstance*", "*receiveInteraction*", and so on. In this research, we use simulation node's flag as the object handle, the object class which can update the attribute as the object class handle. Both the object handle and the object class handle can be gotten from RTI through the callback function "*discoverObjectInstance*". Federate members use this handles to build an information library. In the same way, attribute handle and its value can be gotten through the callback function "*reflectAttributeValues*", with which information Federate can acquire the update message of the responsive attribute.

Eventually, a Federate can stop running or exit the Federation from calling the RTI service "*resignFederationExecution*". A Federate can delete the Federation by calling the RTI service "*destroyFederationExecution*", considering there is no other Federate still in the very Federation.

4 Experiments

4.1 Setup and Modeling

We developed a GUI for HLA interface setup, Table 1 show the content of the key parameters which we use in this system. We can set Federation name and Federation file path as below. The file should be edited before building up the Federation. For the EXata software, we gave it a Federate name as "EXata". Similarly, we name the Eagle software as "OSTW_Client" sharing the common Federation name and file.

Figure 3 shows that five simulation nodes were settled on a scene in EXata. The node 1, 2, 4 in one network and node 2, 3, 4, 5 in the other. You can see that a business

Table 1. HLA interface parameters on EXata.

Parameter	Content	Remark
Federation name	DSIP-SYS	Same with Eagle
Federation file path	.fed file storage path	Same file with Eagle
Federate name	EXata	Eagle: OSTW_Client
Entities file path	.hla-entities storage path	
Radios file path	.hla-radios storage path	
Network file path	.hla-network storage path	

flow is configured from node 1 to node 5. Correspondingly, 5 nodes are arranged on the Eagle with node 2 and node 4 settling to move in the opposite directions.

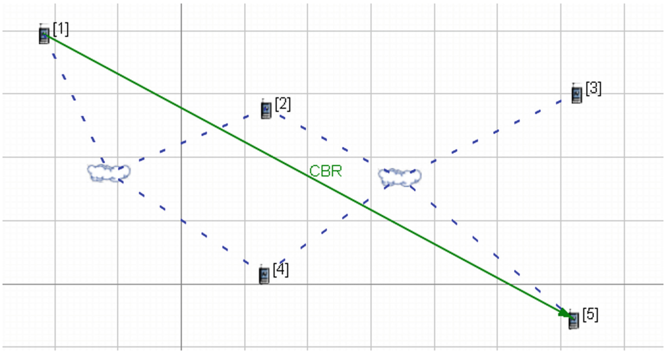

Fig. 3. Network configuration in EXata.

4.2 Simulation Operation

Firstly, we started the RTI software on the both hosts. After executing EXata and Eagle, we can find out that they could join the Federation “DSIP-SYS” successfully. In the running process, the logical connection relationship between the nodes is displayed on Eagle. This connection information was come from EXata through the RTI service. In this way, Eagle makes a synchronized presentation as the results of network protocol calculation on EXata.

Secondly, we started the business flow on EXata. Eagle also got the information through RTI and rendered it on its interface, as shown in the following Fig. 4.

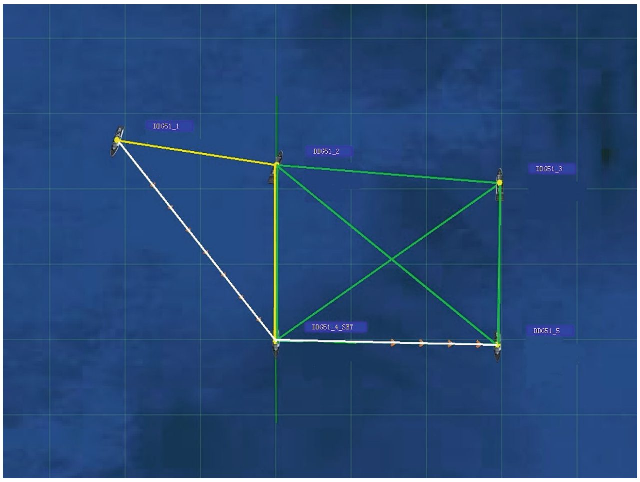

Fig. 4. Sending and receiving business display on Eagle when running.

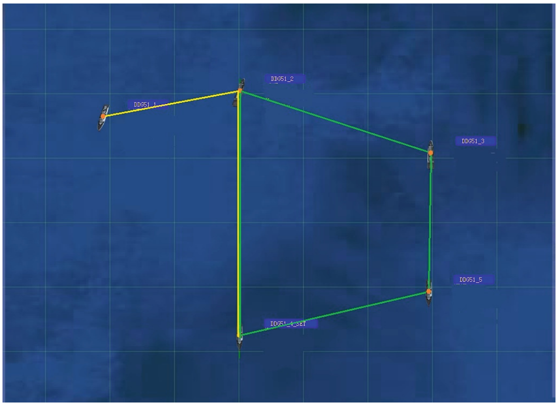

Fig. 5. Display of node moving on Eagle.

When the nodes continue moving as planned in Eagle (as shown in Fig. 5). The corresponding nodes in the EXata also move to the same new location as in Eagle (as shown in Fig. 6). The changed connection information from EXata was published which realized the synchronous change of logical connection on Eagle.

4.3 Result Analysis

In this experiment, we can figure out that developed HLA interface protocol could be used in many different simulation platforms. We made some required functions in HLA service and the simulation platforms to adapt to the needs of users. Our research is an attempt to the use of distributed simulation based on HLA technology, and in the future we would develop more flexible interface protocol to apply to more occasions.

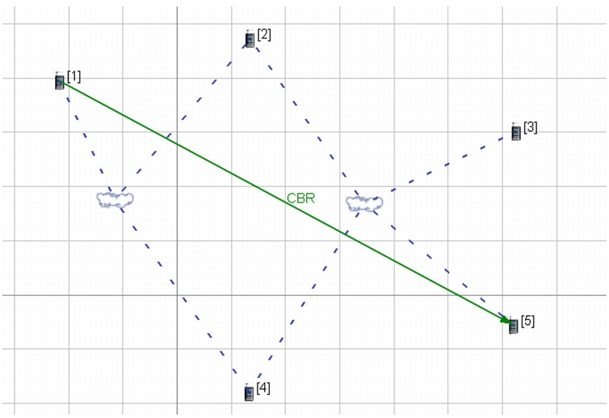

Fig. 6. Nodes synchronize motion on EXata.

5 Summary

The distributed simulation technology conforms to the development trend of the simulation activity from centralized type to distributed interactive type, which has a great significance for the research, design and verification of solving the problem in the complex and integrated system. This technology provides the possibility to realize interoperate between different simulation platforms, and makes the way of simulation activities more flexible [8–10].

References

1. Tian, Z.: Research of sub-water weapon simulation technology based on HLA, NWPU (2004)
2. Kuhl, F., Wealtherly, R., Dahmann, J.: Creating Computer Simulation Systems: An Introduction to the High Level Architecture. Pearson Education, Inc., London (2000)
3. IEEE Std 1516-2000: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Framework and Rules
4. IEEE Std 1516-2000: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Object Model Template (OMT) Specification
5. IEEE Std 1516-2000: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Federate Interface Specification
6. Bo, H.J., Zheng, Q.D., Li, O.L., Xing, S.: Research on the development of distributed simulation application system based on HLA. J. Syst. Simul. **12**, 481–483 (2000)
7. Chen, X., Ni, Z.: Research on upgrading of submarine distributed combat simulation system based on HLA. J. Syst. Simul. **19** (2007)
8. Xiao, T., Fan, W.: HLA based integrated platform for collaborative design, simulation and optimization. J. Syst. Simul. **13**, 3542–3547 (2008)
9. Zhao, G.: Research on the key technology of fully mechanized coal face safety production of virtual reality system. Xian university of science and technology (2013)
10. Chen, X., Xiong, G., Guo, B., Zhang, H.: Research on multi-disciplinary modeling based on HLA. J. Syst. Simul. **11** (2003)