The study of integrating HLA concept for refreshing the antiquated Training Simulation System

LIN HUI
Dept. of Innovative Information and Technology
Tamkang University, Lanyang Campus
Yilan County, Taiwan
amar0627@gmail.com

JI-JEN WU
Dept. of Computer Science and Information Engineering
Tamkang University,
Taipei, Taiwan
wujj5770@yahoo.com.tw

YU-JU LI
Dept. of Computer Science and Information Engineering
Tamkang University,
Taipei, Taiwan
kk5599@msn.com

***Abstract*—System upgrade is an issue involving risk, cost and expanding performance. With advantage of powerful computer, this paper will focus on presenting a way to upgrade an aging Training Simulation System (TSS) by attaching HLA feature in order to have interoperability and reusability for being part of large-scale military joint warfare exercise simulation.**

Keywords : DIS , HLA, RTI

I. Introduction

The aging Training Simulation System (TSS) is hard to interact with the other simulation system such as joint warfare gaming system, tactical simulation systems or simulators that would not allow TSS to be part of large-scale joint warfare simulations. The objective of this paper is to analyze TSS for its original design concept, framework and communication method as well as how to move High Level Architecture (HLA) into TSS as HLA-based TSS. It will allow HLA-based TSS interacting with the systems on both homogeneous and heterogeneous. The standards of Distributed Interactive Simulation (DIS) and HLA originated from Defense Modeling and Simulation Office (DMSO) of U.S. DOD. HLA specification was officially defined as IEEE1516 standard in Sep. 2000. With a standard framework, these standards can be wildly used by the current military systems and make the interactions among different simulation systems possible. This paper starts with tracing TSS's framework, the way of internal message interchanged and then to analyze HLA framework for transforming purpose. Based new system flow upon six services of HLA Run Time Infrastructure (RTI) with the requirement of reserving TSS fundamental tactical functions, upgraded TSS can be more flexible and versatile.

II. High Level Architecture

As predecessor of HLA, DIS was designed to support loosely coupled training exercises on local area networks but unable to support large exercises on wide area networks for its high latency in WAN applications that makes it impossible to support close interactions in a large-scale exercises. Another DIS defect is that simulations are not repeatable because of causality and latency being not managed. HLA is the sophisticate version of DIS with the advantage of interoperability and reusability that makes large-scale exercises carried out in different simulation systems issue solved easily.

A. HLA standards

There are three categories of standards in IEEE1516 series which are HLA rules, Federate Interface Specification and Object Model Template. The HLA rules specify the relationship of federation and federate, the policy and agreement set for the purpose of completely interacted. Federate Interface Specification defines the functions and services provided by RTI. It includes six management modules and series of functions, and it ensures that each Federate provides callback function. Object Model Template (OMT) defines the data format of all the simulation exercises and their participating components in HLA. There are two categories in OMT: Federation Object Model (FOM) and Simulation Object Model (SOM). The Object Model can describe a single federate in HLA, in the meantime it creates a HLA SOM; it can also describes federation produced by multiple interacted federates and creates a FOM. The main purpose of OMT is to upgrade the compatibility of simulation systems and the reusability of component in each system.

B. Run Time Infrastructure (RTI)

RTI provides a series of software services to support the operation and data exchange among federates and these services satisfy the function modules which are needed when interactions take place among federates in the HLA environment. The main purpose of RTI provides a standardized interface service for executing programs of exercise.

RTI provides six management modules which allow functional callback among federates, in case to complete the service and ordering of whole system procedure. The six modules are federation management, declaration management, object management, ownership management, time management and data distribution management.

III. TSS Analysis

System initialization of the original TSS is at the beginning of execution for ensuring the system consistency. There are two stages in the initialization, hardware and exercise. The purpose of hardware initialization is to ensure all computers used in the exercise started normally, otherwise the report is made and the abnormal computer would be excluded in exercise. The purpose of initialization is a preparation for the exercise.

TSS has two parts: the server end and client ends. In system initialization phase, in terms of hardware, these two ends would execute error detection, data interchange and communication setup.

When all client ends' computers are set, all needed items in exercise including choose exercise name, trainee allocation, communication plan, voice communication allocation, plotting boards, electronic display units and Stand Alone Communication Unit (SACU) allocation would be selected.

The system starts exercise initialization after all needed items selected that is for letting all participated consoles, i.e. computers, be in the same environment for their simulated training. At this stage, main computer (server end) and local computers (client end) would have a series of actions of preparation including database establishment, multicast group setup, exercise scenario preparation, communication device preparation and status transition. In this process, any computer failing for its initialization would be offline and excluded in the exercise.

When system has initialization done, the exercise starts. When exercise is about to finish, the 'end' button is selected on the exercise screen of each computer. System transmits finish message to server so that the client computer can logout. There is a series of weapon deployment, detection message transmitting etc. in between system start and system end. The training simulation exercise flow chart is as Figure 1.

Message interchange in TSS is by the mechanism of Reliable Multicast Process Group (RMPG) which embedded in each computer for managing message interchanges in group of computers. Application User Interface (API) offered by RMPG provides the service such as communication establishment, maintenance, and connectivity. TSS is steady because of the functions offered by RMPG supporting the huge amount of message interchanges.

The message interchange in TSS is sent via UNIX Domain Socket. According to the different destination of message, the program will store messages into Global Queue or Local Queue. Global Queue would send message to other groups; Local Queue would send message to its own group only. Before message sending away, system will transform it into PDU format.

IV. UPGRADE TSS WITH HLA ARCHITECTURE

Figure 2 is the system architecture of the original TSS. It is a Client-Server architecture consists of two parts: Main Computer (server end) and Local computer (client end). Both use socket as information communication.

The Main Computer part includes TTC_MANAGER, APP_PROC and exercise main database. TCC_MANAGER is responsible for the check of all computers status and helping to send out messages. APP_PROC manages the tactical services and sends the messages to Local Computer. All data will be stored in main database.

Local Computer consists of MFK, User Interface, Local_APP and local database. MFK is a touch panel for user to give instruction, such as selecting exercise rate, weapon etc. Local_APP is responsible for all local tactical services and sending messages to Main Computer. Local Computer stores the data in local database. User Interface displays all exercise situations on user's control panel.

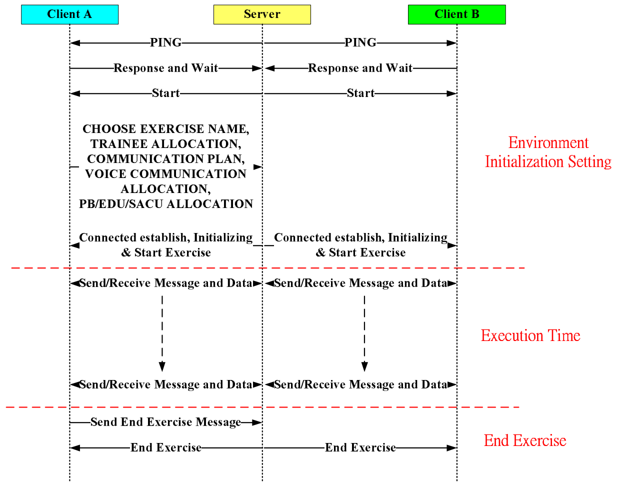

Figure 1. System Flow Chart

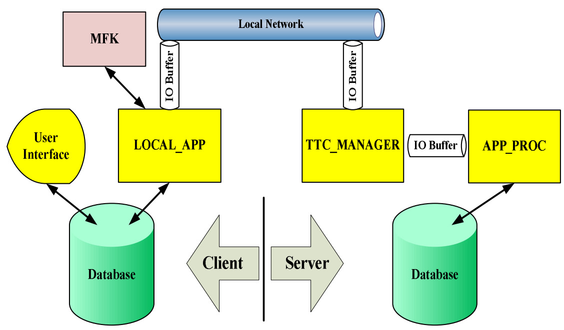

Figure 2. Original System Architecture

Since the original TSS framework is less capable with Ada computer language and hardware being out of date and impossible maintained, this paper suggests the way of upgrade is to make a transition from the original architecture into HLA. HLA is the most common used military framework and it can combine different manned simulator, computer and utility in case to upgrade the range of simulation. Hence, all HLA architecture simulation systems can perform integrated joint simulation country wide.

Upgrading original TSS with HLA-based feature can benefit the joint exercise because of being interoperable. There are also several advantages with HLA framework: first, it provides an interface for all computers through network that increases the interaction between users and data acquirer (including system designer and analysts). Second, the modeling and simulation environment is under open system framework, adopting reusable components to construct that maximizes utility rate and flexibility. Third, HLA framework integrates all simulators and simulation systems that expand the capability in using simulations in joint exercises. Forth, the RTI mechanism reduces the data quantity on network for interchanges that is superior to the use of PDU format in TSS. Fifth, the modular process would make system easier in the function of edit, increase, delete and with system's extensibility Sixth, DIS does not take care of the issue of time management that would have chaos occurred in the order of data receiving. HLA comes with complete time managing functional services that can avoid the time ordering problem.

Figure 3 is the overview of HLA-based architecture for upgrade purpose. TSS architecture maintains, but TTC_MANAGER is replaced by FedExec of HLA. The in-use function of six service modules in RTI is able to support the most requirements. The comparison of TTC_MANAGER and FedExec is showed on Table I.

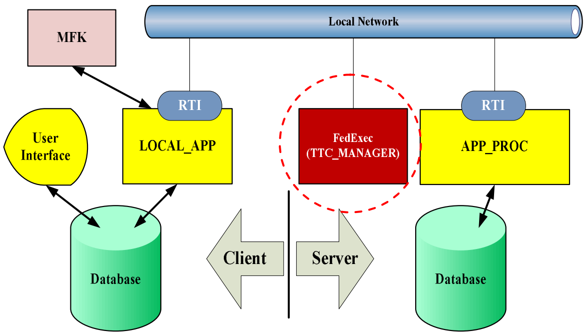

Figure 3. HLA Architecture

TABLE I. TTC_MANAGER VS. FEDEXEC

TTC_Manager in TSS	FedExec in HLA
• Checking all participating computers' status • Handling additional computer to join • Message interchange • Exercise recovery	• Joining and leaving federate in federation management • Helps message interchange in RTI • Restored function in federation management • Time management in RTI

Two major improvements need to be done in order to make HLA-based architecture embedded in TSS possible. The first improvement is to modify the communication achieving by letting RTI manage all message interchange that allows tactical module to concentrate on its tactics design and tactical planning without the concerns of data transfer issue. The second is to modify Local_APP and APP_PROC that will base HLA architecture upon the actions on current APP module with addition, modification and delete. Details would be as follows.

A. The first improvement

In terms of communications between Main Computer and Local Computer, under TSS architecture, before message send-out, it needs to transform the send-out message into PDU packets and then saved in buffer. As for receiving message, the PDU packets would be received and save into buffer. However, the PDU format is not precise and the data is highly overlapped with most of data remaining unchanged but is still packaged in PDU packet that would waste bandwidth resource.

However, when under HLA architecture, this communications only need message conform to FOM, then the message can be sent out through RTI, other than that, it won't send any unchanged messages. The RTI declaration mechanism handles all the transmission; the users won't need to know where the packet is sent to. They only need to create the useful data and let RTI handle the rest of it.

B. The sceond improvement

This is to modify APP module. Figure 4 depicts the contrast of APP module in TSS and HLA. The main modification is to change the APP from original TSS into HLA architecture.

In TSS APP module, the send-out from client A to client B must by server, therefore some APPs would appear at both of main computer and local computer, such as Dynamics, MSG_OUT, EX_MNG and Detection, etc. In fact, the function of APPs are not the same at both main and local computer, e.g. at main computer, the tactical APP is simply responsible for receiving and send-out the message to the clients but it's not the case at local computer that would make mistakes that is ambiguous.

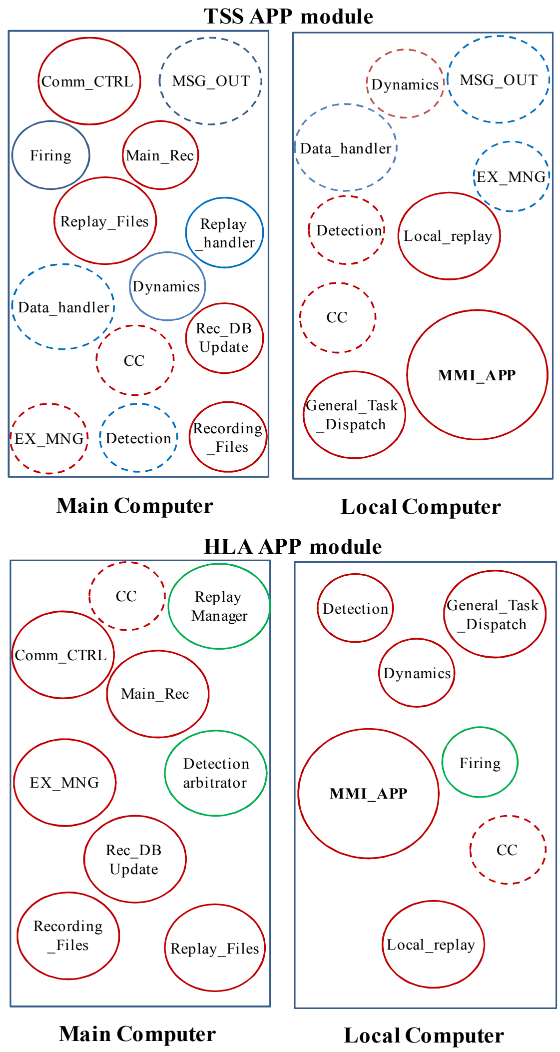

Figure 4. The contrast of APP in TSS and HLA (dash line indicates the same APP at both computer)

In HLA APP module, all message interchange is taken care by RTI instead of server. Thus, the main computer under HLA architecture has much simpler APP. The planning for Local Computer and Main Computer in HLA is clear with APP functions more distinct. The Local Computer focus on tactical execution such as ship maneuvering, weapon launching, radar detection, user interface and playback; the Main Computer manages mission assignment, exercise arbitration and playback control. By means of modular, the partitions of system structure have their clear border with a better division of work, i.e. we simplified the TSS APP module. Future system program modifying, maintenance and update will be easier. And it avoids the problem of over-duplicated program codes.

V. THE EXECUTION PROCESS IN HLA

HLA provides complete functions in support of system basic design. The system flow of HLA has with three steps: Exercise Initialization Setting, Execution Time, and End Exercise. These three steps are responsible for the initialization, execution and exit of the exercise.

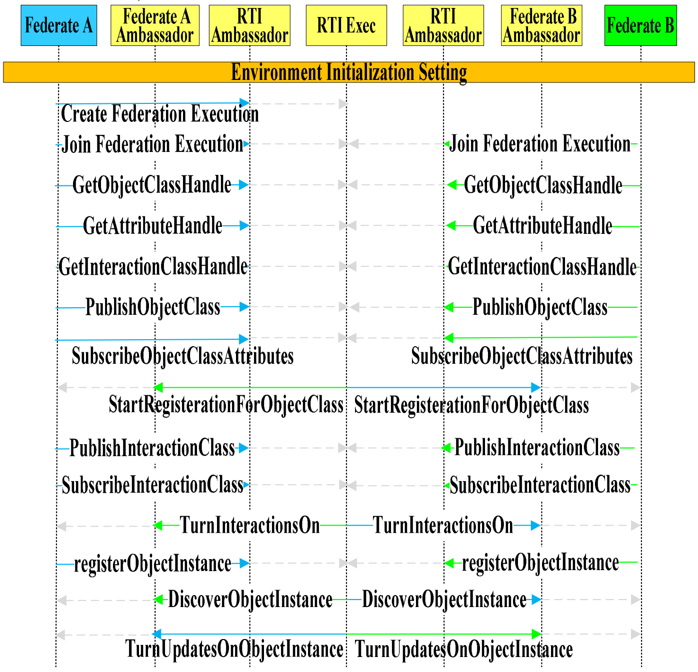

Figure 5. Exercise Initialization Setting

Exercise Initialization Setting manages joining federates, creation and data distribution of federation. A federate creates the first federation, then the following federates can join in directly. Once federate join the federation, it requires to publish its information to RTI such as objects ownership, interaction property that declares for the available information. Then RTI will transfer to other Federates which subscribed information from RTI. When simulation started, RTI will inform those subscribers (federates) for the update the latest information. Exercise Initialization Setting process shows as Figure 5.

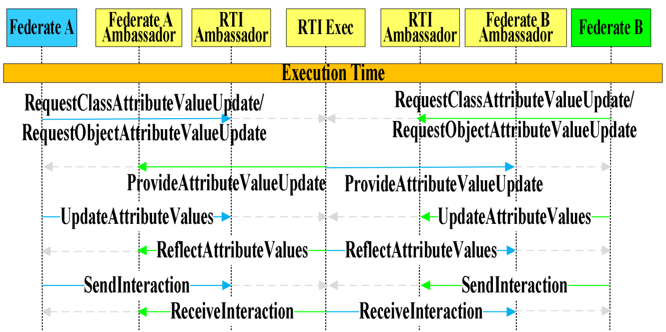

Figure 6. Execution Time

Execution time manages the updated information and transferring ownership. Object information is kept updating during exercise, therefore federate needs to instantaneously ask for renewing process from RTI timely. Transferring ownership from any federate requires to rely on RTI's functions for transferring. Execution Time process shows as Figure 6.

As simulation comes to an end, each federate needs to delete the objects it distributed and notifies the subscribers to remove the given information. Next is to exit from the federation where the federate joined and the first federate is responsible for delete the federation it created. End Exercise process shows as Figure 7.

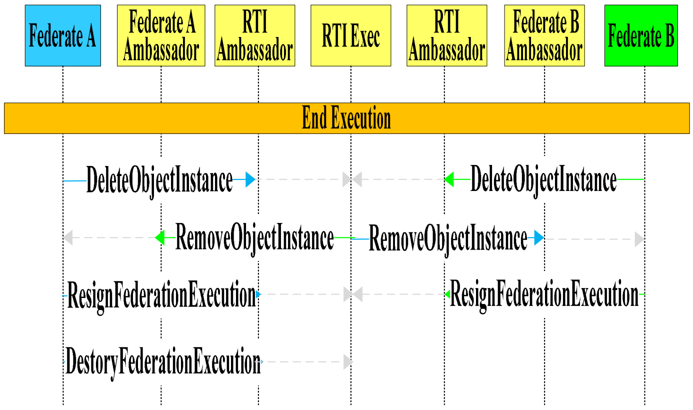

Figure 7. End Exercise

VI. CONCLUSION AND FUTURE WORK

The purpose of this paper is to upgrade the aging Tactical Simulation System for not only having the original tactical functions remained but expanding TSS's coordinated capability with other simulation systems. The concept of design for transforming TSS into HLA architecture which is proved feasible on new hardware. Other than that, this updated system is very easy to add new weapon systems by using tactical module of local computer. The limitations of DIS are also resolved. Future study will focus on time management, and the compatibility among different simulations' RTI.

REFERENCES

[1] Chih-Chen Chiu, "A Methodology for PIM Modeling", 2007.

[2] Frederick K., Richard W., and Judith D., Creating computer simulation systems : An Introduction to the High Level Architecture, Prentice Hall,2000

[3] IEEE Std 1516(2000),IEEE standard for modeling and simulation(M&S) high level architecture(HLA)-framework and rules.

[4] IEEE Std 1516.1(2000),IEEE standard for modeling and simulation(M&S) high level architecture(HLA)-Federate Interface Specification.

[5] IEEE Std 1516.2(2000),IEEE standard for modeling and simulation(M&S) high level architecture(HLA)-Object Model Template(OMT) Specification.

[6] J. Y. C Raczy, G Tan, SC Tay, R Ayani, "Adaptive data distribution management for HLA RTI," in Proceedings of 2002 European Simulation Interoperability, 2002.

[7] R.M. Fujimoto , Distributed simulation systems, Winter Simulation Conference, December 2003, pp. 124-134

[8] Thomas A. Pender, Unified Modeling Language, DrMaster Press Co., Ltd, 2003

[9] Wei-Chyuan Wang, "Design of RTI-compatible interactive visual simulation system ",1997