Using Middleware to Connect STK and RTI

Yang Jian
The Institute of Electromechanical Engineering and
Automation National University of Defense Technology
Changsha, China
tiptopboy@163.com

Huang Jian
The Institute of Electromechanical Engineering and
Automation National University of Defense Technology
Changsha, China
huang_jian@139.com

***Abstract*-Currently, the simulation development faces the requirement of interconnection on applications. The middleware provides a way to deal with this problem, and improve the level of interoperability between applications. STK has been used widely in simulation analysis applications in spaceflight field. In order to realize the interconnection between STK and HLA, this paper has researched on the traditional STK-RTI middleware and gives the idea of component-based middleware development to improve the reusability of middleware modules. Based on this idea, the STK-RTI middleware is developed to verify the feasibility and validity of the component-based middleware development method. This paper has also accomplished the reuse of middleware-component for different HLA simulations.**

Keywords-Middleware, STK, Module, Compatibility, Reuse, RTI, HLA

I. INTRODUCTION

Integrative joint operations require united commands and assignments in the multiple dimensions battle space. In order to search new stratagems that fit for the new environment and verify the effect of cooperation that takes on among combat units, the simulation platform based on the requirement of integrative joint operations is needed to offer a seamless environment for different applications, and this platform allows all kinds of combat unit to be integrated into simulation. The complicated simulation has brought forward high demand in reusability and interoperability on applications. It has been proved that the middleware technique can realize the intercommunion between applications with its advantage on improving the level of interoperability of the existed applications [1].

II. TECHNIQUES

A. Why need middleware

Applications that can deal with specific problems are powerless for the aim of building complicated simulations. To complete complicated and multiplicate missions, wide platforms are required. It is inefficient to develop new applications for customers and vendors, they want to make use of the existed applications. Obviously, in this way, vendors need just take a short-period on sorfware development while customers shouldn't pay too much. Therefore, integrating the existed applications into complicated simulation platform is a good choice. The new platform should sustain and manage multiple applications , and mask the difference in network and system environment. Reliable and effective data transfer and transform between applications are necessary [2]. The conflict between requirements of interoperability and differences in structure and operating environment of existed applications must be resolved.

As a technique to connect applications, middleware may be suitable to meet the requirement of building wide platforms by using the existed applications. Middleware is a type of independent system software and server program. Distributed application shares resources from different techniques by using it. Middleware locates at the top layer of the Client/Service system, manages computation resource and network communication [1]. Middleware supplies a common platform for applications to run on upper layer, it masks the complicated and difference on infrastructures. In fact, middleware doesn't only realize the interconnection but also interoperation between applications.

Middleware supports normal protocols and interfaces, which help user to develop and integrate complicated applications agilely. Since the public interface is more and more important to interoperation, middleware has been playing a crucial role in integrating different applications [3] [4].

B. component-based development pattern

The idea of component-based software development has been the driving force in the course of software development. A component is a cell, with determinate function, standard interface. A set of components may compose an application. The most attractive character of component is reusablity. Once a component has been developed by developer, it will be put in the component warehouse. When a new application is to be developed, the software developer will search the component warehouse for useful components first. Therefore, the process of software development can be simple and efficient. The level of reusability and composability of middleware may be improved by separating it into components [5]. The component-based software development idea meets with customers' and vendors' requirement on puting down expenses.

The research in this paper is about one kind of middleware, which supports interaction between STK and a set of applications, for example, RTI, GIS etc. Because of the difference on applications and the difference on development motives, the detail realization of some modules may not be the same. But the main structure and concept of middleware should not be different. it ensures that we could reuse the existed modules to make up new

Published by Atlantis Press, Paris, France.

middleware. Basing on the project requirement, this paper have designed the STK-RTI middleware in detailes.

C. *Introduction of STK*

In the multiple dimension battle field, all kinds of weapons, such as satellites, airplanes and ships should be brought into chain of command. In the field of spaceflight, the Satellite Tool Kit (STK) is an advanced system analysis application software, it is usually used to analyze complicated spaceflight, navigation, earth and sea missions [1]. It provides multiple results such as two and three dimensions visualized dynamic instance precise diagram and report to support the weapon in its life cycle. STK also has great potential on battlefield inspection, coverage analysis, and attacking effect evaluation for military satellite.

STK offers the Connect module specially to link to external applications. Through this module, the interconnection can be established between external applications and STK, which ensures reliable data transmission and interaction between two applications. The Connect module is shown as Figure 1.

As STK is more and more widely used in simulation applications, the middleware to connect STK and other applications is in great demand

III. STK-RTI MIDDLEWARE DESIGN

This paper has already described the idea of component-based development in middleware, but it just stays on conceptual level. The following paragraphs will give detail description of the application of this idea in STK-RTI middleware development.

The middleware consists of five modules, the Connect and Interact module is designed to interact with STK； the Fed Agent module can establish connection with RTI； the Load and Mapping module loads and parses xml file， mapping different data structure； The Manage Agent module manages the process of simulation；The Control and Display module is used to interact with users. The structure of STK-RTI middleware is shown as figure2.

Each of these five modules realizes different functions of the STK-RTI middleware. Except the Control and Display module, the others four modules are designed as DLLs, they are independent of each other, and provide standard interfaces. The Manage Agent module takes up the core place; it is used as the manager of middleware.

A. *Connect and Interact module*

This module is of great importance because it directly connects with STK. The STK/Connect module provides powerful interface for the third party application. By using Connect module, middleware can establish and cut off connection with STK, send commands to STK to set instance information and get all the object information from STK. In order to store instance-data from STK temporarily, this module should contain a data-pool. After all the instances are obtained， the instances information will be sent to Data Agent module. The inner structure of the module is shown as fig 3.

B. *Fed Agent module*

Because the middleware is to connect STK and RTI, so the STK-RTI middleware should be able to join the federation to interact with RTI.

This module acts as an agent of STK to join the federation; it should be able to interact with other HLA federates. Therefore, this module should has the ability that common federates should have, including linking with RTI, advancing the simulation time, publishing the object and its attribute, sending interaction command, receiving the message from other federates and interacting command.

To discuss this module clearly, this paper needs a detail introduction of HLA, but this is not the main point of this paper, so we just have a briefly description on its function.

C. *Load and Mapping module*

This module is the most significant part in implementing the function of middleware. The data from STK is meaningless if they are not processed to fit the application that connects with STK. This module provides a mechanism to mapping different data structures.

If we compare the middleware with cakes, the others modules are the processes of preparing eggs, flour, sugar and so on. The mapping module is as important as the process of turning these materials into delicious cakes. This module defines the data that are used to intercommunicate between two applications, and gets the mapping mechanism between applications by parsing external xml file; this xml file includes all the mapping information.

This paper realizes the mapping mechanism by building a common data pool. It stores data that are used to interact between two applications; two labels are used to identify each group of data. For the data coming from RTI, their first identifier are the designations in HLA and the second are their titles in STK. The two identifiers’ mapping relation is defined and stored after the first identifier. In this way, the middleware can get the data’s designation from both the source-application and the target-application and mapping them accurately.

D. *Manage Agent module*

If we compare developing middleware with making cakes, the Manage module is as important as the cook, who knows all the flow clearly. Though each module has its own functions, without manage module, the target is still unachievable. These modules need to work together accordance with the procedures, which is defined in the Manage Agent module. This module manages the whole process of simulation, such as the mechanism of simulation time advance. This module should also have the ability to receive kinds of control commands from users. The inner structure of this module is shown as figure 5.

A crucial task of this module is to manage the simulation time advance. On one hand, RTI will transfer the simulation time advance of the federation to middleware. On the other hand the STK/ Scenario runs based on the set time step. It is necessary to implement the logical time of STK synchronization. In the middleware, we get the start time by parsing the xml file, thereafter middleware requests time advance from RTI to update simulation time that used to drive the STK Scenario and interact with RTI accurately.

E. *Control and Display module*

For RTI, The STK-RTI middleware exists as an agent of federate. For the sake of convenient control and observing the real time situation, the control and display module is designed to receive user's commands and display the real time data that comes from the manager module.

STK-RTI middleware is a special application of middleware; with its own characters. Generally, the middleware connecting STK and external application is consisted of these above five modules. For different applications, the functions of corresponding modules may have differences. For example, if we want to build middleware to connect STK and GIS, our purpose is to display the situation of the battle field with two dimensions visualized dynamic graphic, then the control and display module should be put emphasis on.

After completing these five modules and compose them into a whole. Then the middleware is finally completed. Each one of these modules is unattached with the others, they have specific functions with steady interfaces. These modules reflect the ideas of component-based and hiberarchy development. The designed middleware has achieved the objectives we set initially: building a bridge between STK and RTI to make a wide simulation platform. In this new platform, the STK joines in the RTI as a federate, satellite and airplane can interoperate with other weapon cells with no otherness.

IV. SUMMARIES

Composable middleware can meet the requirement of interconnection between applications and provide seamless complicated simulation platform, which accommodate the trend to improve the level of interoperability between applications. Further more, the level of reusability and composability can also be improved by utilizing the idea of component-based development. STK has been used widely nowadays in multi-dimension war game because of its abilities in simulation analyse. Therefore, it is no doubt that the middleware connecting STK and other application may have enormous market in the future.

The technique used to develop middleware is agile to develop complicated simulations. Middleware gets the mapping information from xml file, and we don't have to define the mapping relation as we are about to run the middleware. In this way the mistake from users can be avoided. Furthermore, as each part of the middleware is encapsulated in DLL, we can reuse these DLLs as we want. Hence, we need't waste resources to develop new applications.

REFERENCES

[1] Study on the Design and Implement Technology of the STK-RTI Middleware. JIANG Li-li, Wang Da, HUANG Jian, Qiu Xiao-gang, HUANG Ke-di. Systems Modeling and Simulation[C]. Part 2, Part 14, 284-288, 2007

[2] Middleware : A Model for Distributed System Services. Philip A. Bernstein. COMMUNICATIONS OF THE ACM February[C] 1996/Vol. 39, No. 2 87

[3] An Execution and Transaction Model for Active, Rule-Based Component Integration Middleware. Ying Jin, Susan D. Urban, Amy Sundermier, and Suzanne W. Dietrich. engineering and deployment of cooperative information systems[C]:Beijing, 17-20 September 2002

[4] An Architecture for Next Generation Middleware. G.S. Blair, G. Coulson, P. Robin, M. Papathomas. Middleware '98 Proceedings of the IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing[C]. Springer-Verlag London.UK.1998

[5] Interoperability in the Next Generation Training Systems. Per M. Gustavsson. 2009 Fall Simulation Interoperability Workshop[C] Orlando, Florida, September 2009

[6] The Levels of Conceptual Interoperability Model. Dr. Andreas Tolk, James A, Muguira. 2003 Fall Simulation Interoperability Workshop [C].Orlando, Florida, September 2003

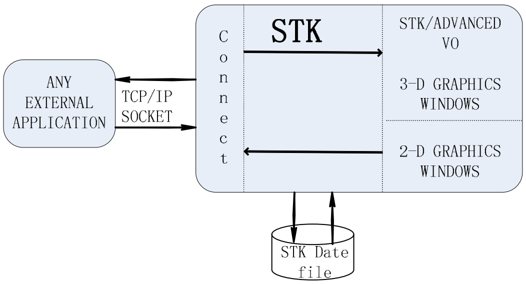

Figure 1 Function of STK/connect module

Published by Atlantis Press, Paris, France.

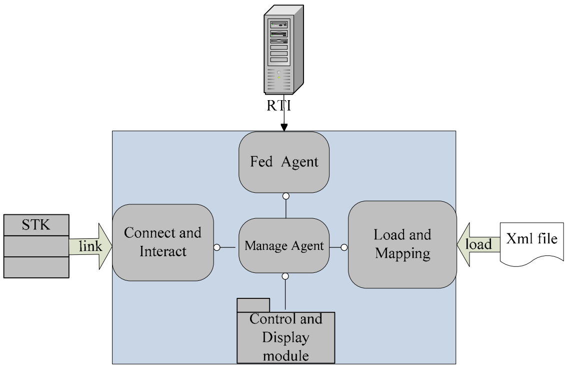

Figure 2 Structure of STK-RTI middleware

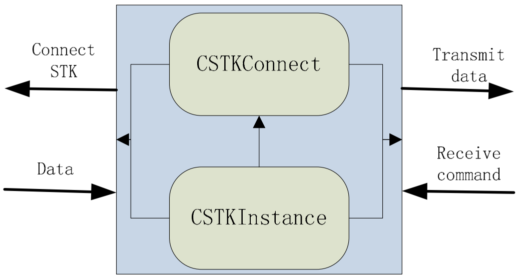

Figure 3 Inner structure of connect and interact with STK module

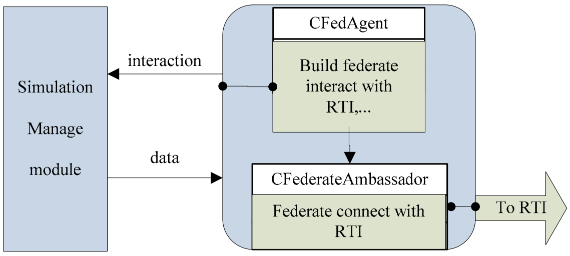

Figure 4 Structure of Fed Agent

<table>
<tr><td colspan="3">SimManager module</td></tr>
<tr><td colspan="2">sent data from data pool to FedAgent module</td><td rowspan="4">Manage the process of simulation</td></tr>
<tr><td colspan="2">Load the type and attribute of HLA object</td></tr>
<tr><td>Put The data received from STK into data pool</td><td rowspan="2">Interact with STK</td></tr>
<tr><td>Send command to STK connect module</td></tr>
</table>

Figure 5 Functions of SimManager module

Published by Atlantis Press, Paris, France.