Study on Scene Simulation Federate of Plane's RMS Requirement Determination Simulation System

Cao Jun ,Sun Yufeng ,Zhao Guangyan,Kan Linjie
School of Reliability and System Engineering
Beihang University
Beijing, China
e-mail: armybupt@yahoo.com.cn ,syf@buaa.edu.cn, zhaoguangyan@buaa.edu.cn ,kanlinjie1985@126.com

***Abstract*-Scene simulation technology has been developed into a research focus with the military simulation development. The key technologies of the scene simulation federate are discussed in this paper, and then the method of integration between HLA and Vega, and the way of keeping the integrality of scene simulation information are presented. This article outlines the development process of the scene simulation federate of plane's Reliability, Maintainability and Supportability (RMS) requirement determination simulation system based on HLA, Vega and Visual C++. The scheme of scene simulation federate is helpful for the development of similar equipment's scene simulation. The simulation results indicate that the scene simulation federate can vividly image the actual use of plane, achieve the expected functions and effects and improve the effects of plane's RMS requirement determination.**

Keywords-Scene simulation; High Level Architecture (HLA); Vega; Reliability, Maintainability and Supportability (RMS).

I. INTRODUCTION

The property of equipment's RMS is one of the important facts which affect equipment's efficiency, fighting capability, viability and Life Cycle Cost, and equipment's RMS requirement demonstration is equally important as the capability demonstration. In order to allow researchers intuitively and vividly observe the whole process of demonstration in virtual environment, scene simulation federate of plane's RMS requirement determination simulation system is developed. It can shorten the cycle of responding to the demonstration results because of the intuitive way for simulation data provided by scene simulation federate. It also can improve determination efficiency. Scene simulation technology can make users intuitively and vividly see most of object states in virtual environment, since it can create living virtual environment, comprised by visual, vocal and tactile environment. Scene simulation technology has been widely used in military simulation, virtual tourism, etc, because of good immersion, interactive and real-time characteristics[1,2].

Currently, there are many different three-dimensional visual simulation software. But since plane's RMS requirement determination simulation system has been developed based on HLA, and scene simulation federate should be good real-time and interactive, the paper develops the scene simulation federate with Vega which is a high performance visual simulation system and can be used to construct a simulation application quickly with the highest performance possible[3,4]. By discussing and bringing forward an appropriate method of integrating Vega into HLA and function of scene simulation federate, the aim of the paper is to develop scene simulation federate of plane's RMS requirement determination simulation system. In this paper, the progress of plane's RMS requirement determination simulation was displayed in three-dimensional animation by scene simulation federate based on Vega, Visual C++ and HLA, which is developed by applying Run-Time Infrastructure (RTI).

II. DESIGN OF SCENE SIMULATION OF PLANE'S RMS REQUIREMENT DETERMINATION SIMULATION SYSTEM

A. Introduction of Plane's RMS Requirement Determination Simulation System

The operating, maintaining and supporting process of plane is simulated by plane's RMS requirement determination simulation system in different condition such as task plan, maintenance system, etc. It is used to analyze affecting factors of plane's operational readiness and dependability, review the plane's RMS parameters, validate the distribution of the index, and choose a proper project[5]. It's developed based on HLA, and the structure of the plane's RMS requirement determination simulation system is shown as Figure.1.

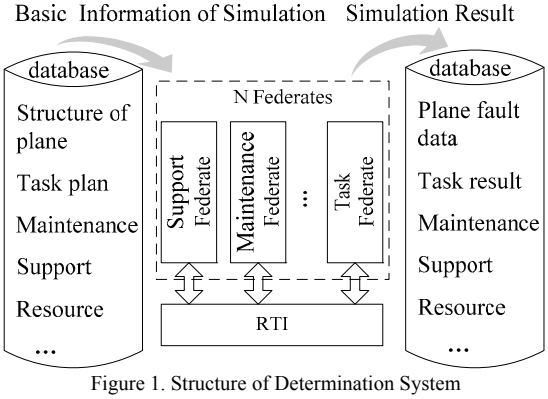

Figure 1. Structure of Determination System

B. Scheme of Scene Simulation

The primary plane's RMS requirement determination simulation system was a federation which was composed of N federates. A publish-and-subscribe interaction exits between the N federates, and the plane's RMS requirement

determination simulation is executed through federates' properties and interaction between them. Therefore, it is necessary to create a new federate, the scene simulation federate. It should have some functions which are described as follows:

(1) Getting simulation information: Receiving the primary plane's RMS requirement determination simulation federation's interaction information, subscribing some federates' object class attributes and accessing the original simulation information.

(2) Rendering scene: Analyzing simulation information and rendering scene.

(3) Displaying basic information: Displaying each piece of basic simulation information in the form of text in order to make researchers watch and understand easily.

The scene simulation federate initializes the system with the original simulation information at first. Secondly it receives the interaction information and subscribing some federates' object class attributes based on HLA framework, and finally it displays the simulation process by rendering the three-dimensional scene. The scene simulation federate structure is shown in the Figure.2.

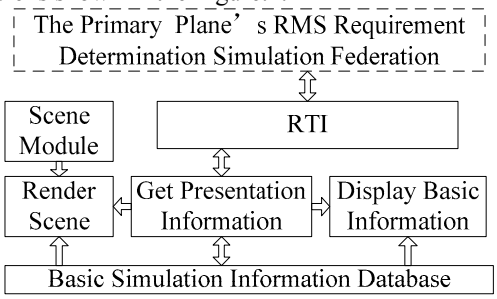

Figure 2. Structure of Scene Simulation Federate

C. Design of Integrating Vega into HLA

HLA technology is based on kinds of TRI which provides functions to implement HLA rules for achieving network communication transmission services and management of simulation operation[6]. And using Vega technology could vividly show the simulation process. Actually, HLA and Vega have several characteristics in common: supporting the MFC library functions, operating by activating independent simulation thread, configuring system setting before running the simulation thread, controlling the simulation circulation by specific strategy, etc. Therefore, HLA and Vega can be integrated in the software platform according to their common characteristics.

The scene simulation federate should activate two threads. One is HLA thread that manages the simulation control and the data interacted with other federates. The other one is Vega thread which shows the simulation information received from HLA thread by rendering three-dimensional scene. While the simulation information queue is empty in HLA thread, the scene simulation federate is allowed to advance the time, otherwise it is not allowed to send time advance request. The design of integrating Vega into HLA is shown in the Figure.3.

D. Integrality of Simulation Information

Sometimes, the interaction information received by the scene simulation federate can not describe the current scene, mainly due to the following:

(1) Interactions and attributes of objects are mainly used to execute mutual operation, in which one federate may send message and attribute update to other federates. In order to reduce the traffic flow, part of the events and their attributes are not published.

(2) Most of the simulation information is expressed in form of ID for object uniqueness and storage convenience in simulation process. This information is not convenient for researchers to analyze.

(3) The primary plane's RMS requirement determination simulation federation was not changed in order to keep its independence.

Therefore, it is necessary to analyze the information and get detailed information via accessing some relative tables in database by searching ID for keeping the simulation information integrated and intuitive.

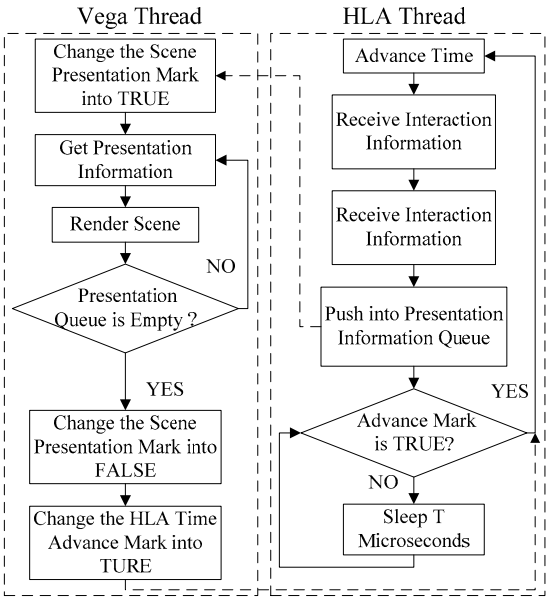

Figure 3. Method of Integrating Vega into HLA

III. DEVELOPMENT OF THE SCENE SIMULATION FEDERATE

A. Design of Object/Interaction Class

Federate achieve interactive operation and information traffic with other federates by publishing itself object attributes and interaction messages, and subscribing object attributes and interaction messages which it requires. The interactive information includes whether the operation accomplishes or not, the results, the impacts on other federates, etc. These operation information and object class attributes describe the whole process of plane's determination simulation. The process includes task flow, fault information, maintenance work, support work and so on. The scene simulation federate should subscribe this information and perform them in the form of three-

dimensional scene. So, the scene simulation federate do not need object class, and its interaction class comprises of simulation control class, task unit class, task section class, unit fault class, maintenance class, support class and resource class. These interaction classes' attributes are subscribing, that is, federate can subscribe interaction class and use the attributes, but it is not required to operate the object class affected by the interaction. Part of interaction attributes are shown in Table1.

B. Design of Visual Presentation Module

Visual presentation performs each piece of presentation information by rendering three-dimensional scene, and it is mainly composed by five classes based on Visual C++: analyzing presentation information class, task unit presentation class, maintenance presentation class, support presentation class and fault presentation class.

Analyzing presentation information class analyzes the interaction information received by federate, and the analyzing flow is shown in Figure. 4. Task unit presentation class performs four states of plane's common tasks which are start, finish, failure and canceled. Maintenance presentation class describes the four states (start, finish, failure and waiting) of kinds of plane maintenance works. Support presentation class shows the information of the subscribing, traffic and spare meeting situation of spare and support resource. And fault presentation class marks plane's ordinary fault, critical fault and other faults, and displays them.

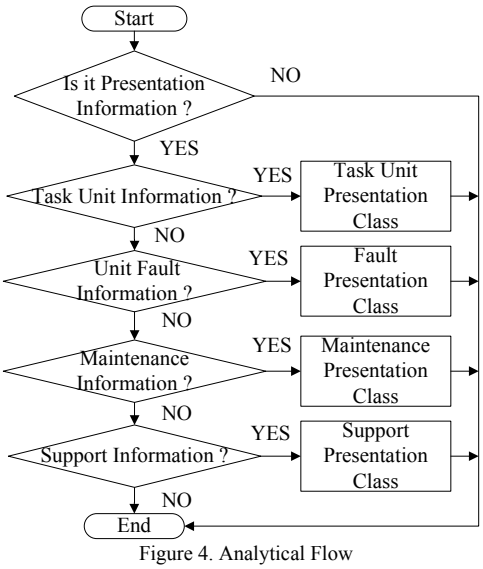

Figure 4. Analytical Flow

C. Achievement of the Scene Simulation Federate

Scene simulation federate receives plane's task, maintenance, support and fault events, analyzes and displays them in form of three-dimensional scene based on HLA which is supported by RTI platform. The simulation flow is shown in Figure. 5.

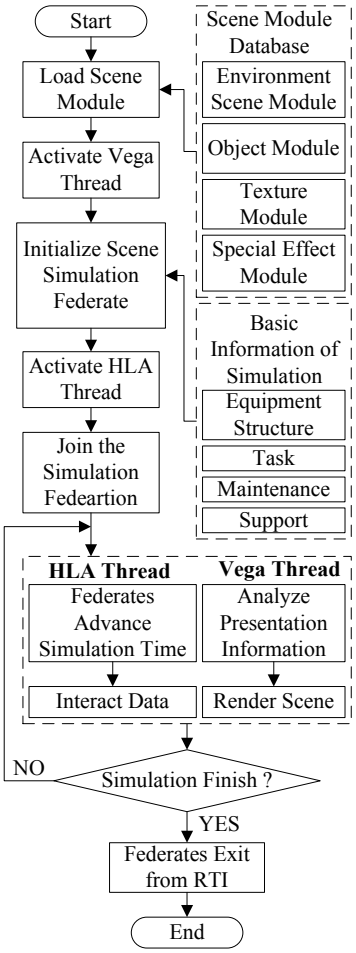

Figure 5. Simulation Flow

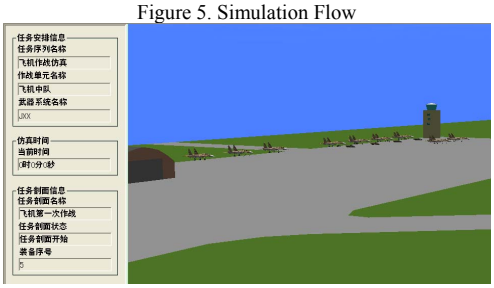

Figure 6. Interface of Simulation Software

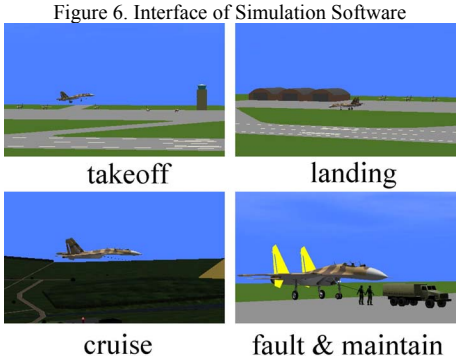

Figure 7. Scene Simulation Effects

The flow of simulation is shown as follows:

(1) Load scene modules. First, call Vega functions to initialize the system and create the shared memory area and semaphore area. Next, load the environment scene modules, object modules, texture modules and special effect modules. Finally, conFigureure system related parameters.

(2) Initialize the information of federate. Access the database to finish initializing the structure of plane, task information, maintenance information, support information and so on.

(3) Join the federation. Scene simulation federate join the federation to contact with other federates by RTI.

(4) Advance the simulation time. Federate would accept unified manage of RTI tool after completing interacting data and updating attributes, and advance simulation time to the next time point.

(5) Interact data. Scene simulation federate subscribe and receive other federates' interaction information, which includes task, maintenance, support and fault information.

(6) Analyze presentation information. Vega thread analyzes presentation information after HLA thread received the interaction information. The analyzing logic is shown in Figure.4.

(7) Render scene. Render scene by the graphics rendering function of Vega after step 6 which would fix interaction information on scene presentation instance.

(8) Simulation circulation. Execute step 4 to step 7 in cycle until the simulation ended.

(9) Exit from RTI. Scene simulation federate exits from federation and the process of plane's RMS requirement determination simulation.

Scene simulation federate is developed based on HLA, Vega and Visual C++. It achieves the function demand as the design process in Figure.5 and the effects is shown in Figure.6 and Figure.7.

IV. SUMMARIES

The paper discusses the function requirements of scene simulation federate based on analyzing the plane's RMS requirement determination simulation system. The way of integrating Vega into HLA and keeping integrality of simulation information are put forward in the paper according to the key technologies of scene simulation federate. The development scheme based on HLA, Vega and Visual C++ is designed and developed based on plane's RMS requirement determination simulation system. This simulation system has great practical value because of its high fidelity and immersion, providing an intuitive and vivid way for watchers, and improving the effects of the determination simulation. It is useful for improving system expansibility, reusability and modifiability with using object-oriented design method and it is helpful to develop visual simulation system of similar equipment.

REFERENCES

[1] PENG Liang, HUANG Xin-han. Simulation System of Cruise Missile Using VC and Vega Prime [J]. Journal of Central South University, 2004,41(1):219-224.

[2] LIN Xin, SONG Yan, WANG Xing-ren. A Distributed Flight Simulation System Based on HLA [J]. Journal of System Simulation, 2004:2751-2753.

[3] Multigen Paradigm Inc. Vega programmer's guide [M]. Dallas: Multigen Paradigm Inc., 2001.

[4] Multigen Paradigm Inc. Lynx user's guide [M]. Dallas: Multigen Paradigm Inc., 2001.

[5] Wang Xiao, Zhao Guangyan, Sun Yufeng. Plane's RMS Requirement Demonstration Simulation Based on HLA[C]//2010 3rd International Conference on Computer and Electrical Engineering. Sichuan, Chengdu, China, 2010:243-246.

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

TABLE I. PART OF INTERACTION ATTRIBUTES

Interaction Class	Interaction Attributes
Simulation_Control	ControlType
	ReceiverID
Task_Unit	MessageType
	TaskUnitName
	TaskUnitID
	TaskUnitStartTime
	TaskUnitRunTime
	DelayTime
	TaskUnitDurationTime
	BattleUnitID
	TaskSectionID
	ReceiverID
	SenderID
Unit_Fault	EquipmentID
	UnitUniqueID
	UnitFailTime