

A Study of Real-time Simulation Supporting Platform Based on VMIC Model Wrapped

Chenglie Du¹, Kai Yang¹, Tao You¹, Nan Jiang², Dong Zhong¹

¹School of Computer College Northwestern Polytechnical University, 710129, Xi'an, China

²Kunming Shipborne Equipment Research and Test Center, Kunming 650051, China

Abstract—This paper propose the idea of building a real-time package simulation platform in VMIC Reflective Memory network through drawing on the idea of generic HLA Federation, especially for real-time needs of semi-physical simulation. For model complexity, heterogeneity and diversity of the model, it propose wrapper solutions based on model. The wrapper model, transmitted data simulation and federate iterative program have been done analysis and research. It also designs and implements the statement management, the Federal management, object management and time management services. It verifies the platform's real-time high prosperity, through comparing the data of the platform performance and the MAK-RTI performance data.

Keywords-federation; reflective memory network; wrapper; real-time simulation.

I. INTRODUCTION

System modeling and simulation technology can be divided into modeling technology, modeling and simulation support systems technology, and simulation applications technology^[1]. The establishment of a support platform has a significant in conserve resources, improve efficiency, cost savings. At present, the field of simulation technology for the military-industrial is toward to complex, distributed, heterogeneous, and real-time direction, which makes support platform become the focus of the research. We draw on HLA / RTI and design thinking, design and develop real-time simulation platform based on VMIC Reflective Memory network^[2]. For model complexity, heterogeneity and diversity of the model, it propose wrapper solutions based on model. The wrapper model, transmitted data simulation and federate iterative program have been done analysis and research. It also designs and implements the statement management, the Federal management, object management and time management services^[3].

In this paper, combined with a combat system real-time simulation, it verifies the platform's real-time high prosperity, through comparing the data of the platform performance and the MAK-RTI performance data^[4].

II. REAL-TIME SIMULATION NETWORK BASED ON REFLECTIVE MEMORY NETWORK

In real-time network application, traditional Ethernet and Ethernet card can not meet the harsh requirement of real-time performance. So real-time optical fiber network based on virtual shared memory occurred. VMIC

company' reflection memory real-time network^[3]Rtnet(ReflectiveMemory) is one of mature product so far. Reflection memory network is a fast real-time network and it can be used to connect every kind of computers. It has many features about strict transmit certainty, predictability, high-speed, light host load, strong adaptable for hardware and software platform, reliable transmission error-correcting ability, supporting interrupt signal transmission and so on. Reflecting memory network is composed by interface board inserted in the two nodes and the fiber connecting interface board. In every interface board, there is a special storage and they are mapped to the same address space, constitutes the distributed shared-memory. Reflection memory board can be inserted into many bus's main board such as VME, PCI, CompactPCI, and so on^[5].

Real-time prosperity of the simulation system is the input control signal, the system time response characteristics can accurately reflect the time response characteristics of real system of mathematical model after inputting the same control signal. It is necessary to coordinate real-time simulation of semi-physics and virtual system, so the requirement for real-time prosperity is below a few milliseconds. In this situation, Real-Time Ethernet (RTE) or Industrial Ethernet cannot be achieved, it needs to provide 100 microsecond latency level communications networks, so it is clear that reflective memory network can meet the low latency, high demand for real-time.

III. THE LOWER SOLUTION BASED ON INTERPOLATION

A. Models integrated thinking based on wrap

In order to implement the simulation model can plug-and-play in the simulation system, the results of seamless integration, and consider the real-time network system and the maturity of the docking RTI[4, 5], we design a system with reference to the application of the RTI interface specification and the package mechanism in network environment. Application of RTI interface specification has been discussed in detail in the second chapter. The mechanism of HLA packaging applications achieves relatively complex programming. In order to facilitate the development and implementation of application systems, reducing development time and the workload, a lot of research has been done on RTI programming interface package, and researcher put forward a framework for HLA application development,

data structure templates and the concept of code templates. We take object class template release and object instance handling template, simulation tasks template as an example, discuss the HLA package^[6].

A federate simulation models program is one of the basic elements in simulation test. From point of the model view, the model is complex, heterogeneous and diverse, such as armored vehicles, bombers, aircraft carriers and so on. Therefore, a generic design suitable for all models prototype is almost impossible. Due to the heterogeneous, so it needs to package the model. Our integrated package for each model is to design a generic wrapper, as shown in Figure 1.

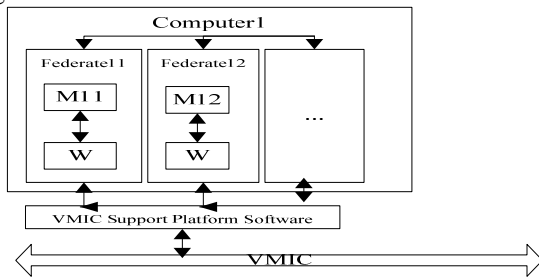


Figure 1. Model of integrated

Figure 1 indicates the relationship among the supporting platform software, wrapper (W), model (M) and network VMIC. Model 1, model 2 to model n is the dynamic link library. Wrapper is responsible for packaging these heterogeneous models, and shielding the differences between models^[7]. It provides a unified interface to greatly reduce the difficulty of the development of simulation models. So it is also increase the reusability of the software and shorten the software development cycle, meets the software "Open - Closed" principle. Wrapper is collecting model data from the reflection memory network and publishing the data that other model subscribe to memory network.

B. Platform system architecture

Simulation support platform architecture is show in Figure 2:

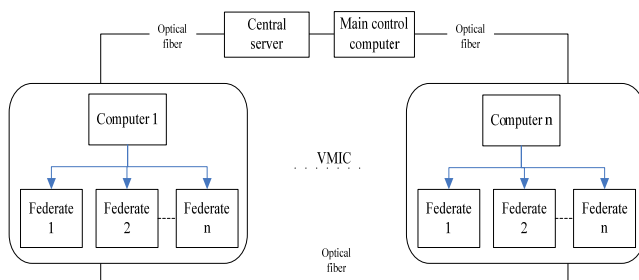


Figure 2. System architecture

Each node has its own simulation server in VMIC network, called sub-server, responsible for their own Federal member's requests service. There is a central server in Real-time network, which responsible for unified

management of all sub-servers, and overall request for federal operations^[8].

1) Master control computer: It controls federal and federal members join the system. First of all, the master control computer distributes the model and configuration file-related to each sub-server. It sends to central server and then central server specific distributes to every sub-system and control the operation of the process simulation. Master control computer communicate with the central server through the SOCKET.

2) Center Server: Main function is to manage time, federal, data distribution and object management services with sub-sever, and receive host commands at the same time. It is an important position in the entire system and manages the unified scheduling management.

3) Computer n: Synergies to achieve a variety of management services. At the same time, it manages assigned federal members, including the configuration of the Federation members, the operation of the Federal members and receives commands from Central server.

IV. IMPLEMENT OF SIMULATION SUPPORT PLATFORM

A. Function design

It provides the following services:

1) Declaration management services: It is establishing a relationship which is published and subscribe between announced members of the Federation, and this relationship can be changed dynamically. This service can reduce the delay and the amount of data in networks during the simulation process.

2) Federation management services: This function is implemented by the host computer and the central server. When there is a new Federation to join the system, after configuration in the master control computer and registration in the central server, then central server can manage the federate, including the federate members join in and out.

3) Object Management Services: This function is implemented by the central server and all sub-servers. After publishing the object of each federate member, only the federate who subscribed the object will receive the data of the object. According to the object management mechanism, federate member can automatically update their own attribute in the running process and subscribed federate will get the updated attributed values. Federate member will send the interactive class event; the federate who subscribe the event will receive the interactive event.

4) Time management services: It is assure the order of all federate members simulation event as the same order in real world and coordinate the activities of the Federal members. The real-time simulation or ratio real-time simulation does not affect simulation results.

B. Implementation process

1) Implement of declaration management: It uses UML configuration file to implement. The example of UML file:

```

<Model instance_name="FederateA"
model_name="FederateA.dll"
init_file="FederateA.txt"><objectclass><name>missile</
name><attribute>position</attribute></objectclass><inter
catclass><name>explode</name><attribute>energy</attri
bute></intercatclass></Model>

```

In the xml configuration file, it gives a list of each model and its corresponding interactive object class publishes and subscribes relationship. After getting the xml file, it store the result of the analysis file in a linked list structure, and can dynamically change publish and subscribe relationship.

2) Federation management services implements. In the central server, the main process creates a federal management thread; the thread is always running, and waiting the registration signal of the master control computer. A new member of the federal need to join, it is configured by the master computer, and then handed to the thread. Federal management thread defines a linked list structure, called FederationList, it dynamic manages federal and federation to join and exit. Federal or federation has a unique ID, and have their own address space.

3) Object management implement is show in the figure 3:

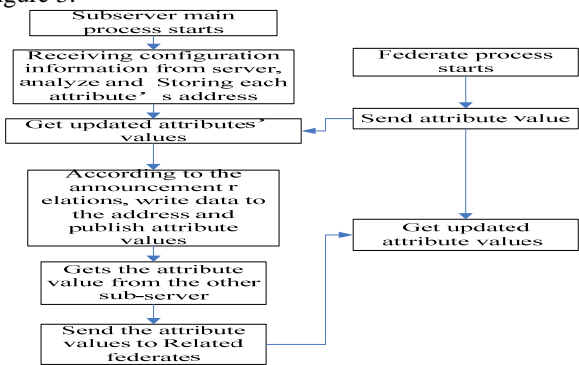


Figure 3. Object Management Service Flow chart

4) Time management service implements

In the central server, a time management services will start automatically as system clock, and send time data to all the running federation members through VMIC communication mechanism. There is a time management thread in every federate program. It obtains the time data to determine whether to advance according to the cycle of federate member. At the same time, it sends the next step time data to the time management program in central server. The time management program receives the data and select least time as the next step promoting time.

V. PERFORMANCE TEST

A. Time promoting request test

We have built simulation models to test the size of the average advance time in a variety of load cases, and compare the test data in MAK-RTI. Simulation models are following: Assume that the total number of the federate is n, the order is 1,2, ..., n; each Federation i publishes a different class C (i), the size is M bytes, then all the federate will be publish different type class (C (1), C (2), ..., C (n)); if n is 8, we take it as an example, it has 8 publish class. Numbered 1-4 in Federation members released their own category, a total of four categories; No. 5-8 members of the Federation must be subscribe C (1), C (2), C (3), C (4). We will test the advance time in different data size, and do it repeatedly. P4 1.5 or more computer nodes connected through the VMIC-5565, use the Windows XP operating system, there is a central server, with the other two machines A and B as a sub-server, to run the Federal members respectively.

The test results of Simulation model in the simulation based on the generic federal support platform and the MAK-RTI environment are shown in Figure 4. It is seen that the system advance real-time prosperity is higher than the time-MAK-RTI real-time under the same conditions.

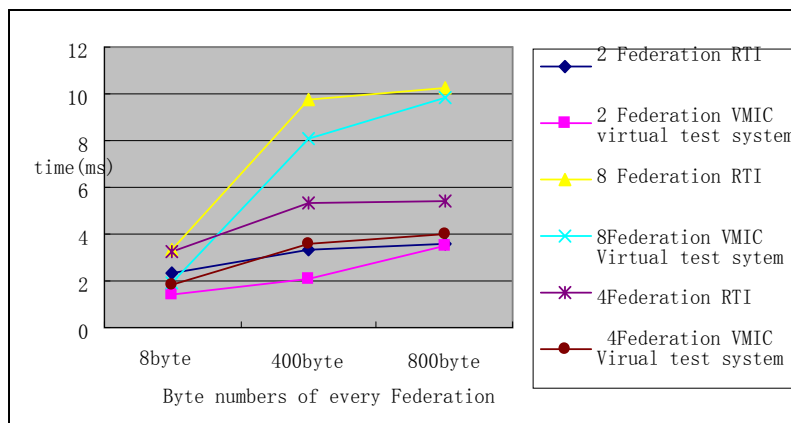


Figure 4. Time progresses comparison chart

B. Test the delay of attribute update

We test the delay value of object and interactive attribute update of the MAK-RTI and the simulation based on the generic federal support platform under the same condition. The results are shown in Table 1.

TABLE I. DIFFERENT PLATFORMS PROPERTY DELAYED CONTRAST

Attribute size (byte)	16	32	64	128	256	512	1024
Simulation platform	0.177	0.244	0.269	0.335	0.395	0.421	0.465
Mak-RTI	0.336	0.398	0.449	0.493	0.556	0.521	0.668

We can see from Table 1, the delay of RTI properties is almost double the support platform in the situation that object class and interactive class properties has equal size.

This shows that two key indicators improved using the method which proposed in this paper. It provides a relatively weak bounds for building simulation system architecture and easy to design. Also real-time prosperity of supporting platform makes it possible can achieve the semi-physics simulation.

VI. CONCLUDING REMARKS

Supporting platform for distributed simulation is an important basis of computer simulation. In VMIC environment, we design a new idea in the light of HLA’s federation thinking. Based on this line of thinking, we design a simulation platform, and achieve declaration management, federal management, object management and time management services. We also test the method combined with examples of the test environment and

verify it is reasonable. It is need to consider how to make the environment has better flexibility and stability, as well as the package technology of semi-physical model in the further research.

VII. ACKNOWLEDGEMENTS

This work is supported by Aviation Science Fund(20135553034)and NPU Foundation for Fundamental Research.

REFERENCES

- [1] Li Bohu, Chen Zongji. Modeling&Simulation in China–Present and Future[A]. Proceedings of Asian Conference on System Simulation and Scientific Simulation Conference/the 5th International Computing, [C]. Shanghai, 2002.
- [2] Claxton J D, Cavoli C, Johnson C. Test and evaluation management guide (Fifth edition)[M]. Defense Acquisition University Press, 2005.
- [3] Milan Jovanovic, Veljko Milutinovic. An overview of reflectivememory systems [J]. IEEE Concurrency (S1092-3063), 1999, 7(2):56-64.
- [4] Liao Ying, Liang Jiahong and so on. Real-time simulation theory and supporting technology,[M].National Defense Science and Technology University Press, 2002
- [5] Huang Xiaodong, He You, Jiang Benqing. The research and test of RTI real-time and information throughput,[J]. Journal of System Simulation, 2003, 15(4):516-518.
- [6] B Logan , G Theodoropolous1 The distributed simulation of multi-agent systems1 Proceedings of the IEEE , 2001 , 89 (2) : 174-185.
- [7] Banks J, Carson J S, Nelson B L. DM Nicol, Discrete-Event System Simulation[M]. Prentice Hall, 2000.
- [8] Wang L, Ling M, Zeng Q, et al. Application of RTX in semi-physical simulation for laser-guided bomb[J]. Infrared and Laser Engineering, 2006, 35(1): 78.