RESEARCH ON CO-SIMULATION TASK SCHEDULING IN CLOUD SIMULATION PLATFORM

Chen Yang[a], Bo Hu Li[b], Xudong Chai[c]

[a]Beijing University of Aeronautics and Astronautics, Beijing, China
[b]Beijing University of Aeronautics and Astronautics, Beijing, China
[c]Beijing Simulation Center, Beijing, China

[a]wzhyoung@163.com, [b]bohuli@moon.bjnet.edu.cn, [c]xdchai@263.net

ABSTRACT
In order to address the problem of single collaborative simulation task scheduling, considering the characteristics of simulation resource encapsulated with virtualization technology, the author first proposed the unified model describing the co-simulation system for co-simulation task, which is the basis of task scheduling; Secondly, based on the unified model, the virtualization-based supporting system, in cloud simulation platform (CSP), of dynamic construction of the co-simulation system were introduced. Thirdly, the scheduling procedure, namely the dynamic construction of the co-simulation system, was discussed. Finally, the primary application example and conclusion was presented.

Keywords: cloud simulation platform, HLA federation, co-simulation task scheduling, virtualization technology

1. INTRODUCTION

Recently, as an effective tool for understanding and reconstruction of the objective world, M&S theory, methodology, and technology have been well developed, form its own systematic discipline, and their application area are increasingly expanded. At the same time, they are developed toward "digitization, virtualization, networking, intelligence, integration, and collaboration", which are considered as the characteristics of the modern trend.

As the application field of M&S continues to expand, the size and complexity of simulation application system are greatly increased, which poses a severe challenges to M&S technology. High level architecture provides a general framework and corresponding software engineering method "FEDEP" for developing large-scale distributed simulations, which can promote the reusability and interoperability of simulation model. However, HLA does not inherently take into account of resource management and task scheduling for co-simulation, especially the simulation resource is statically bound together with federates before simulation start, in which case the automatic scheduling is lacked. The combination of M&S and grid computing gives birth to simulation grid, which realizes the dynamic share and reusability, collaborative interoperability, dynamic optimization for simulation execution, of different resource. Simulation grid has resource management and task scheduling for certain degree, but due to the heterogeneity of OS and software environment, the large variance of performance, of nodes in simulation grid, in addition to unstableness of network, the simulation grid can not effectively and quickly execute the large-scale simulation, and the effect of co-simulation task scheduling methods is decreased severely with little advantage. Moreover, simulation grid can not show efficient support for fine granular resource (e.g. CPU, storage, software in nodes) share, multi-user, fault-tolerance, etc. For example, in simulation grid, distributed computing nodes have different kinds and versions of OS and software, which will lead to the limited nodes the task can be scheduled to, such as the federates programmed and compiled to run in Windows OS is hard to scheduled to execute in Linux servers without any adaption.

Due to the unsolved problems in simulation grid, People in Beijing Simulation Center introduce the notion of "cloud computing", and further with integration of virtualization, pervasive computing, and high performance computing technology, propose a networktized M&S platform-Cloud Simulation Platform (CSP) to enhance the ability of simulation grid. CSP employs virtualization technology to encapsulate the computing resource, software, simulation models, etc as virtualized simulation resource, masking the heterogeneity of resources. Based on encapsulated resource, virtualization technology can provide enabling technology for dynamic setting up of simulation execution environment, or even simulation system, which will benefit the co-simulation task scheduling and federates' deployment.

Co-simulation task scheduling referred to in this paper focuses on one single task scheduling, in which several subsystems collaboratively run to accomplish the task, and the scheduling here is mainly about the resource scheduling process to construction virtualization-based simulation system. The scheduling in simulation runtime is not discussed in this paper.

Considering the requirement of co-simulation task in CSP, the author first proposes the top-level description model of collaborative simulation system for task, and then the supporting system for co-simulation task scheduling, and the whole procedure of scheduling are then presented. Lastly, the conclusion and further work are given. Without special declaration, the scheduling object in this paper refers to HLA-based co-simulation.

2. THE TOP-LEVEL DESCRIPTION MODEL FOR CO-SIMULATION SYSTEM

Co-simulation system is the execution entity for fulfilling the co-simulation task. The efficiency of its running directly determines the effect of simulation, which is the reason of co-simulation task scheduling. For the purpose of high efficient scheduling, we first build the unified top-level model for describing co-simulation system. The model contains the essential properties of co-simulation system, which determines the constitution of federation and interoperability between federates. The model provides solid foundations for dynamic construction of federation, which is the process of the scheduling.

Firstly, HLA-based co-simulation system can be described as follows by unified model:

Definition 1. *Co-simulation system = <Fed, Intera, Env, Comm, Comp, Num, StpCrit>*

Definition 2. $Fed=<FM_i \mid 0< i \leq Num>$ is defined as the model of federation, where *FM=<DomainOnt, FuncDesc>* is defined as the model for describing federates. And $|Fed|=Num$ defines the number of federates in federation. *DomainOnt* is domain ontology-based description of federate. *FuncDesc* denotes the semantic description of federate.

Definition 3. $Intera=<Src, Dest, Info \mid Src, Dest \in Fed>$ is defined as the model for interactions between federates. In which *Src* refers to source federate generating information, *Dest* denotes the destination federate for interactive information, and *Info* is the corresponding information to be exchanged.

Definition 4. $Env=<EnvDepend(FM_i) \mid 0< i \leq Num >$ defines the running environment of each federate, including the requirement of OS and software sets. $EnvDepend(FM_i)=<OS, SoftSet>$, in which *OS* refers to the type of operating system such as Windows XP, and *SoftSet* denotes the software list。

Definition 5. $Comm=<N_{net}(FM_i, FM_j) \mid 0< i, j \leq Num, i\neq j>$ defines the interaction requirement for communication capability of network between federates.

Definition 6. $Comp=<N_{compute}(FM_i) \mid 0< i \leq Num>$ defines the computation requirement of each federate.

Definition 7. *StopCrit* is defined as the condition for stopping simulation running. It can be the times of simulation execution, or the time of simulation running, or the bound of variables in simulation.

In fact, the process of dynamic construction of virtualization-based co-simulation system is the realization of the co-simulation task scheduling process. So, the supporting system for scheduling is given as follows.

3. THE VIRTUALIZATION-BASED SUPPORTING SYSTEM FOR DYNAMIC CONSTRUCTION OF CO-SIMULATION SYSTEM

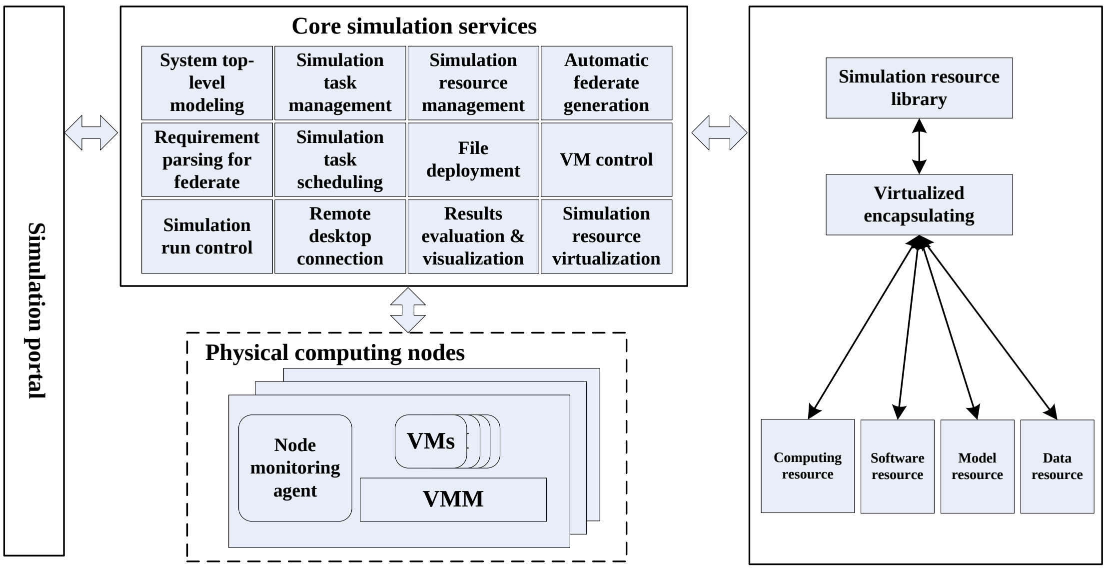

Figure 1: the framework of the supporting system

CSP employs virtualization technology to encapsulate the computing resource, software, simulation models, etc as virtualized simulation resource, which is stored in simulation resource library. Using encapsulated resource, based on the state information of each physical computing node collected by node monitoring agent, the core simulation services in CSP can support the dynamic construction of co-simulation system.

CSP together with the virtualized simulation resource constitute Cloud Simulation System (CSS). The supporting system in CSS includes the logical functional modules shown in Figure 1. It consists of three main parts: simulation portal, core simulation services, and simulation resource library.

- Simulation portal

Simulation portal is the entrance point of simulation activities for users. It supports the simulation activities based on internet or desktop, in which users can submit co-simulation task, acquire the results, etc.

- Core simulation services

Core simulation services are composed of services: system top-level modeling, simulation task management, simulation resource management, automatic federate generation, requirement parsing for federate, simulation task scheduling, file deployment, Virtual machine control, simulation run control, remote desktop connection, results evaluation and visualization, simulation resource virtualization, etc.

System top-level modeling service can provide the top-level modeling service for users based on internet or desktop, support to decompose the complex multidisciplinary tight coupled system into several sub-systems, and help to describe the models of federation and interactions between federates.

Simulation task management service enables the management of co-simulation task. Specifically, the services can support distributed simulation workers to accomplish the simulation task collaboratively.

Simulation resource management services support semantic-based simulation resource searching services, and also the downloading, uploading, revising, deleting, registering, publishing service, etc of simulation resource with certain permission.

Automatic federate generation services can generate the framework of federate according to the description model of federation. Moreover, it provides user-friendly interface to customize the framework, then user can finish functional entity development in the federate on the basis of the framework.

Requirement parsing for federate services can parse and acquire the configuration file of the federate, which includes the semantic description of the federate running environment.

Simulation task scheduling services provide the function to monitor the running state of physical machine, and use certain method to choose suitable physical computing nodes for the execution of virtual machines.

File deployment services give support to remotely deploy simulation related files to specified file path of the computer with certain IP address. These files together with the running environment constitute the executable simulation system.

VM control services lend support to the operation of the virtual machine: start, shutting down, suspending, and resuming the virtual machine. Further, it provides the user to access the virtual machines remotely by desktop connection for performing simulation activities.

Simulation run control services support the execution management of simulation federation, which includes creating federation execution, unified start of federate execution, monitoring federates' state, synchronizing the logical time of federate. And the function of pausing, resuming, resigning, destroying federation execution, etc is also supported.

Simulation resource virtualization services include services of creating different kinds of virtualized simulation resource. Here, virtualized simulation resource mainly refers to virtual machines, into which the simulation resources are encapsulated.

- Simulation resource library

Simulation resource library is employed to store and manage the virtualized simulation resource, especially the management of VM pools, such as the searching of suitable VMs in library with semantic information.

4. THE PROCEDURE FOR THE DYNAMIC CONSTRUCTION OF CO-SIMULATION SYSTEM

The dynamic construction of co-simulation system is composed of three stages: system modeling stage, simulation model development stage, and simulation model deployment and assembling stage. Strictly the former two stages should not be included. But the former two stages are the basis for the scheduling, and for the aim of easy understanding, we list them out. The three stages will specify the content of each factor in unified model of simulation system. We do not pay much attention to the details of realization, and just focus on the principles in the procedure of federation dynamic construction.

The steps are as follows:

- System top-level modeling

Through the requirement analysis of the simulation task, the decomposition of complex research object is a must. The simulation system can be decomposed into several sub-systems in different domain. The decomposition principle is like this: after decomposition, the sub-systems each should have tight coupling inside and loose coupling with entities outside in order to wipe the bottleneck of communication brought by the irrational decomposition.

Using the system modeling services in simulation portal, based on graphical interface, users finish the decomposition of system, and build the description model of interactions between sub-systems. Then, according to certain semantic template, users should accomplish the description of federates (semantic-based domain ontology and function description). Users referred here are mainly chief simulation technology officers. Then, they create the co-simulation task by simulation task management services, with which they upload the files generated in system modeling process.

In this step, the factors *Fed*, *Intera*, *Num*, and *StpCrit* in unified description model *are* instantiated.

- Searching for simulation models based on the description of federates

System high-level modeling gives the description of different domain federates, including ontology based domain description and function description. Using simulation task management services, simulation practitioners in different domains first get the federate description. Then according to the description, practitioners search for the professional models of entities using simulation resource management services (or develop professional models on their own). The models are downloaded. However, this is not enough, because normally, the professional models cannot be used directly in HLA-based collaborative simulation.

- Automatic federate generation services for HLA-based federate development

The professional practitioners employ the system top-level model to generate the framework of each federate using automatic federate generation services. Then based on the federate framework and the professional models, the complete federate which is ready for simulation execution can be developed. Finally, the professional practitioners build the requirement description of federate in these aspects: the running environment, computation and communication requirement, by referring to the description of professional models.

In this step, the factors *Env*, *Comm*, and *Comp* in unified description model *are* instantiated.

- Submitting the co-simulation task

The practitioners who are responsible for the subsystem development, upload federates and description files to corresponding containers in simulation task management services. The top-level model of co-simulation system, federate models and respective description models together constitute the basis of co-simulation system. After checking that all the subsystems have been well finished, the chief simulation technology officers will submit the simulation task using simulation task management services.

- Parsing the description files of the simulation task

The requirement parsing for federate services can acquire the running environment of each federate from the configuration files of respective federate, in which the requirement of executing federate is contained, namely:

$EnvDepend(FM_i) = <OS, SoftSet>$, where FM_i denotes federate, *OS* refers to the type of operating system, *SoftSet* represents the software list needed。For the requirement of the whole federation, *Env* contains the requirement information of all federates.

The running environment information of each federate is used to searching for virtual machine image for each federate.

- Searching for the virtual machine image based on the information of federate running environment

The virtual machine image pool, which is a part of simulation resource library, is a collection of virtual machine images. The virtual machine image pool is located on shared storage. Users can create virtual machine images via simulation resource virtualization services, and upload them to the virtual machine image pool, at the same time, register and publish the semantic description of them via simulation resource management services. Each virtual machine image is a kind of resource that can be shared and reused with different running environment. So, a large number of virtual machine images in the pool can meet the demand of most users. If not, users can create their own ones based on their special requirement.

Simulation resource management services can search virtual machine pools for suitable one in accordance to the semantic description of each federate' requirement. If there exist more than one virtual machines in the result, then users can click the virtual machines listed to check the detailed information to choose the most suitable ones.

- Start virtual machines on suitable physical computing nodes

The simulation scheduling services support the state monitoring service of physical computing nodes, such as the CPU utilization, the available Memory, and the bandwidth, the network delay between them. Further, these services can help gather the statistics of historic monitoring information and forecast the load of each physical computing node.

In order to realize the load forecasting function, the exponential smoothing algorithm is employed to forecast the load of physical computing nodes, then select suitable nodes to start virtual machines on. The simulation task scheduling services can optimally choose several physical nodes using different algorithms to guarantee enough computing and fast communication capability. Or users can search for physical nodes according to their configuration description, check the historical statistics of monitoring information of them, and then choose several ones manually.

VM control services can support the control function of virtual machines on physical nodes, for example start, powering off, suspending, and resuming virtual machines. VM control services first automatically configure the CPU, memory, storage, and network of virtual machines. And then start the virtual machines in chosen physical nodes.

- Remote connection to get the desktop of virtual machines

After the construction of virtual computing environment, the following demands show that the remote desktop connection services are needed.

1) Before the simulation system running, special requirement leads to changing the configuration of simulation software, or even the OS.

2) According to the demand of interactions between users and simulation system, users need the desktop of virtual machines to implement more sophisticated control on simulation process.

3) In order to guarantee the correctness of simulation results, not only the state monitoring of simulation system is need, but also whether the exceptions are thrown should be pay attention to, because some kinds of exceptions will not lead to the

crackdown of simulation system, but will affect the correctness of simulation results.

The remote desktop connection services can give support to getting the virtual desktop of virtual machines. After the start of virtual machines, based on these services, users can use the virtual desktop just like it is the local desktop of the physical machine. In other words, users can start and configure simulation software, and build the running environment of federates. Users can also check whether federates have thrown exceptions.

- Deploying model files and start the execution of simulation system to accomplish the dynamic construction of simulation system

File deployment services can support deployment of files to specified path of machines with designated IP address. Simulation system consists of simulation models and simulation running environment. After the dynamic construction of simulation running environment, the only need is to deploy and configure simulation models (federates). Then the simulation run control services are utilized to create federation execution, and then start federates deployed in virtual machines to join the federation execution. Other simulation run control services such as monitoring the state of federation, control the execution of federation are also provided to uniformly perform the simulation experiment.

5. APPLICATION EXAMPLE

The aforementioned virtualization-based dynamic construction of co-simulation system has been primarily applied in multidisciplinary virtual prototype, large-scale system collaborative simulation, and high performance simulation areas. This paper presented an application example of aircraft landing gear virtual prototype collaborative simulation system. The main steps are as follows:

- System top-level modeling

Using desktop virtualization technology, the virtual user interface of software can be acquired through select the desired software, which is shown in Figure 2. Simulation practitioners first get the requirement of simulation task, and then build the system top-level description model. Specifically, the aircraft landing gear simulation system can be decomposed into undercarriage control model, undercarriage multi-body dynamics model, undercarriage hydraulics model, etc, as shown in Figure 3. Each model here is one federate in the federation. And the interactions between these subsystems are built. Then the HLA FOM files and the ontology-based and function description files of professional models are generated. Simulation practitioners use the simulation task scheduling services to create the simulation task by submitting the system top-level description files, as shown in Figure 4.

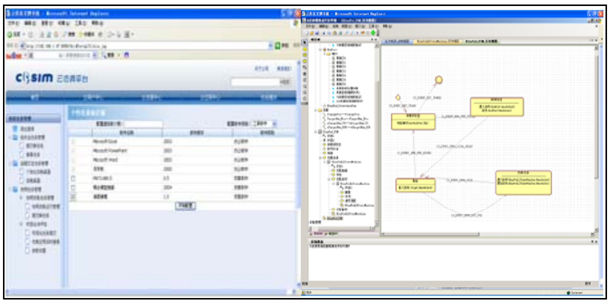

Figure 2: system top-level modeling

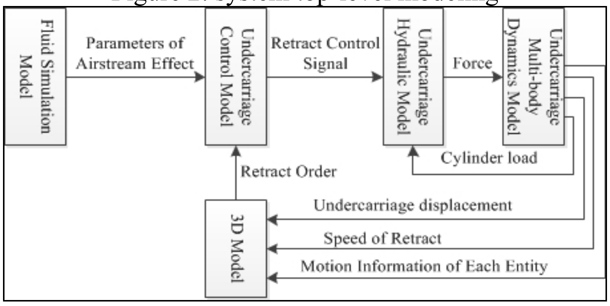

Figure 3: the decomposition of aircraft landing gear simulation system

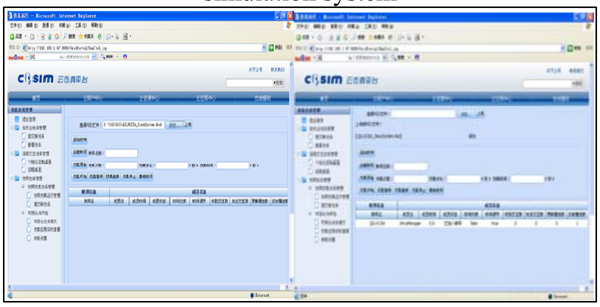

Figure 4: creation of co-simulation task

- Completing the development of federates on the basis of searching for domain simulation models

Domain simulation practitioners can obtain the system top-level description model files via the simulation task management services. The suitable models are listed by searching the domain models in simulation resource library. Then according to the detailed semantic description of listed domain models, domain simulation practitioners download the ones which meet the requirement. With the help of automatic federate generation services, all federates are then developed using these domain models. Finally, referring to domain model descriptions of required running environment, computing and communication capability, the requirement of federates in co-simulation system are demonstrated as follows in table 1.

Table 1: The requirement of federates

No.	Simulation model (federate)	Softwa-re Enviro-nment	Comp-uting Enviro-nment	Perform-ance Demand
1	Fluid Simulation Model	Fluent	Redhat	CPU 8core,

				memory 4GB
2	Undercarriage Control Model	Matlab	Windo-ws	CPU 1core, memory 1GB
3	Undercarriage Hydraulic Model	Easy5	Windo-ws	CPU 1core, memory 1GB
4	Undercarriage Multi-body Dynamics Model	Adams	Windo-ws	CPU 1core, memory 1GB
5	3D Model	CATIA	Windo-ws	CPU 2core, memory 2GB

In which, Gigabit Ethernet and 80GB storage space is fixed for each node.

- Submitting the simulation task and constructing the simulation running environment

Federates developed by domain simulation practitioners are submitted using simulation task management services. The chief simulation technology officers submit the simulation task. The supporting system for dynamic construction of co-simulation system in CSP will parsing the requirement information of each federate, search for five suitable virtual machines images (realized using Xen middleware), in which different simulation software are included. CSP will select five physical computing nodes simply using weight sorting method fusing the factors of computation and communication capabilities. Then the virtual machine images are deployed to physical computing nodes according to the same weight sorting method and started with the demand configuration of federates. Figure 5 is the interface of simulation task submitting and dynamic building of simulation running environment.

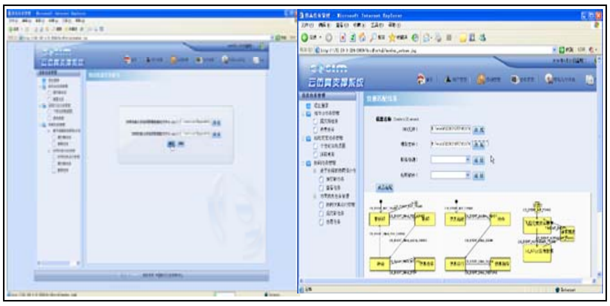

Figure 5: submitting simulation task and dynamic building of simulation running environment

- The dynamic construction of co-simulation system

After the creation of simulation running environment, the chief simulation technology officers can acquire the virtual desktop of virtual machines by remote desktop connection services. The virtual desktop of virtual machine with "matlab" software inside is shown in Figure 6. The files of each federate are then deployed to certain path in virtual machines. With the help of remote virtual desktop, users can configure OS and software. Here we just let the local RTI component point to central component with certain IP address. Finally, simulation run control services are used to start the co-simulation system and monitoring its running state, as shown in Figure 7.

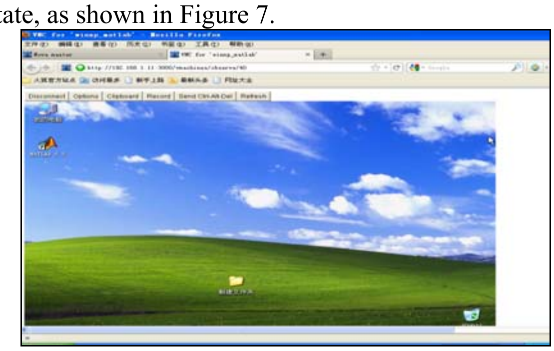

Figure 6: remote virtual desktop

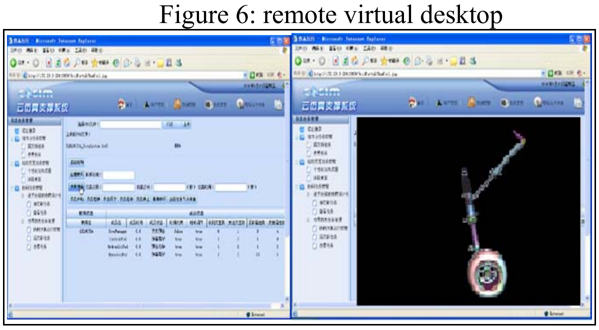

Figure 7: monitoring and controlling the co-simulation system

6. CONCLUSION AND FURTHER WORK

Through introducing virtualization technology, the author presents the supporting system and procedure for simulation task scheduling, and the primary application example. The primary application shows that: virtualization-based dynamic construction of federation can address the constrained scheduling problem caused by the tight coupling between the simulation system and physical computing resources, which will make the co-simulation task automatically scheduled to certain degree.

Future work includes as follows:

1) Further research on the semantic-based unified description model and share mechanism of virtualized simulation resource.

2) Further research on high performance collaborative simulation technology to support high efficient execution of simulation system.

3) Further research on the scheduling method when facing a large number of co-simulation tasks.

ACKNOWLEDGMENTS

This paper is supported by National Defense Key Lab, the National 973 Plan (2007CB310900) and National Defense Pre-research Foundation of China. And authors should like to express the sincere thanks to all colleagues for their help and valuable contribution.

REFERENCES

Bo Hu Li, et al., 2009. A network modeling and simulation platform based on the concept of cloud computing --- "Cloud Simulation Platform", *Journal of System Simulation*, 12 (17), 5292-5299.

Hai Jin et al., 2008. *Computing system virtualization---theory and application* (in Chinese). Tsinghua press.

Bo Hu Li, Xudong Chai, Baocun Hou, 2009. Cloud Simulation Platform. *Proccedings of the 2009 International Summer Simulation Conference*, Turkish, Istanbul.

Z. Li, W. Cai, S. J. Turner, and K. Pan, 2007. Federate migration in a service oriented HLA RTI, *Procs of international Symposium on Distributed Simulation and Real-Time Applications*, 113-121

Bo Hu Li, Xudong Chai, Yanqiang Di, et al., 2005. Research on service oriented simulation grid. *International Symposium on Autonomous Decentralized Systems (ISADS 05)*, pp. 7 – 14.

Bo Hu Li, Xudong Chai, Baocun Hou, 2006. Research and Application on CoSim (Collaborative Simulation) Grid. *The Proceeding of MS-MTSA'06*.

Bo Hu Li, Xudong Chai, Wenhai Zhu, 2004. Some focusing points in development of modern modeling and simulation technology. *Journal of System Simulation*, 16 (9), 1871-1878(in Chinese).

IEEE, 2000. Standard 1516 (HLA Rules), 1516.1 (Federate Interface Specification) and 1516.2 (Object Model Template).

S.A.Herrod, 2006. The Future of Virtualization Technology, *Keynotes of ISCA 2006*, http:// www.ece.neu.edu / conf / isca2006 / docs /Herrod-keynote.pdf.

AUTHORS BIOGRAPHY

Chen Yang was born in 1987. He received his B.S. degree in Beihang University. He is currently a Ph.D. candidate of Beihang University, Beijing, China. Research focuses on advanced distributed simulation, distributed computing, HLA-RTI, theory and practice of modeling, etc.

Xudong Chai was born in 1969. He is a researcher and deputy director at Beijing Simulation Center of Second Academy of Aerospace Science & Industry Co. and council members of Chinese System Simulation Association and National Standardization Technical Committee. His research interests include automatic control and simulation.

Bo Hu Li was born in 1938. He is a professor at School of Automatic Science and Electrical Engineering, BeiHang University, and Chinese Academy of Engineering, and the chief editor of "Int. J. Modeling, Simulation, and Scientific Computing". His research interests include multi-disciplinary virtual prototype, intelligent distributed simulation and cloud manufacturing.