

Service-oriented SMP2 Modeling and Simulation for GNSS

<http://dx.doi.org/10.3991/ijoe.v9iS7.3186>

H.T. Hou¹, F. Xie², C. Wang¹ and W.X. Zhang¹

¹National University of Defense Technology, Changsha, China

²Tsinghua University, Beijing, China

Abstract—We propose a service-oriented modeling and simulation (M&S) method based on simulation model portability 2 (SMP2) to simulate the performance of the Global Navigation Satellite System (GNSS). The proposed method can be utilized to construct large complex simulation systems, such as GNSS, to determine the reusability, composability, and interoperability of heterogeneous simulation resources. A service-oriented SMP2 M&S framework with its conceptual framework, system architecture, and system engineering process is first introduced. The development and composition of the SMP2 model as well as the schedule method are then discussed in detail. Lastly, a service-oriented SMP2 M&S environment is designed and implemented, and the regional dilution of precision performance of the GALILEO Navigation Satellite System in China is analyzed based on the architecture and method.

Index Terms—service-oriented architecture (SOA), simulation model portability 2 (SMP2), distributed simulation, parallel simulation.

I. INTRODUCTION

The complication and bulkiness of real systems are challenges confronting the modeling and simulation (M&S) method. The Global Navigation Satellite System (GNSS) performs all-weather positioning and velocity measurement at proper timing for occupants of the Earth and near-Earth space users. GNSS is also widely utilized for both military and civil purposes. GNSS is a space engineering system with a large-scale and complex structure. Given the multi-organizational development of the complex features of navigation systems in various fields, the key concern in reducing the difficulty of system development and simplifying the research cycle in multi-domain modeling is which method to reuse and integrate in heterogeneous simulation models in different fields.

M&S has evolved into networkilization, virtualization, intelligentization, cooperativization, and generalization^[1]. Good M&S specification and system development paradigms are the basis of the success of large complex simulation system applications.

Simulation model portability 2 (SMP2)^[2-4], which was advanced by the Europe Space Office in 2004, is superior over other M&S specifications in terms of domain independence, platform independence, and simulation model framework description. The overall support process for simulation development and application promotes model friendliness, efficient simulation, convenient integration, and model reusability^[5]. The results of our research on the application of SMP2 reveal that several key application techniques can improve the popularity of

SMP2^[6-12]. These techniques include visualized model design, automated model assemblage, multi-simulation schedule, dynamic model instantiation, distributed simulation, parallel simulation, and so on^[5].

Service-oriented architecture (SOA), which integrates existing services to achieve the functionalities required to build the application, is an ideal distributed software development paradigm^[13]. The simulation components of SOA are loosely coupled. These components can be identified to create a simulation application. The components can be deployed to a heterogeneous platform where they communicate via standard protocol, such as XML and SOAP.

This paper presents a service-oriented SMP2 M&S method. Below are the other contributions of this paper.

1. A service-oriented SMP2 M&S framework is established. In this framework, the concepts and mechanisms of SOA and SMP2 are integrated into a service-oriented SMP2 simulation conceptual framework. A distributed and parallel simulation system architecture is presented based on the conceptual framework. A system engineering process is provided for the construction of service-oriented SMP2 simulation systems.

2. A model-driven SMP2 model service development method is introduced. This method describes how SMP2 model services are deployed and registered and provides the information required in the process.

3. A SMP2 model service composition method is introduced. This method employs a proxy to create local and remote models. Four proxy-based model compositions are designed to establish the method.

4. A parallel schedule method for the SMP2 model service is introduced. This method redefines the schedulable task and its schedule process for the parallel execution of service-oriented SMP2 simulation.

II. SERVICE-ORIENTED SMP2 M&S FRAMEWORK

A. Conceptual framework

SMP2 is a component-based M&S standard. The components of simulation systems are of three types, namely, simulator component, simulation service component, and model component. Detailed information on the concept, function, and role of the components can be found in the reference^[1].

Traditional simulation services consist of simulation components or subsystems that provide simulation functions to the simulation application in the form of runtime infrastructure, simulation data logging, two- or three-dimensional display, and result analysis. These traditional simulation services can be deployed as services

in an SOA environment. Deploying the models as services to support the construction of simulation systems is important. Fig. 1 provides the conceptual framework of service-oriented SMP2.

The conceptual framework of service-oriented SMP2 has three layers: service/resource layer, information/management layer, and tool/support layer. The service/resource layer consists of model services and the kernel of service-oriented SMP2 simulation. The information/management layer is a universal description, discovery and integration protocol (UDDI); this layer facilitates model discovery and selection. The information/management layer guarantees the efficiency and quality of simulation systems. The tool/support layer provides tools to support users as they perform the entire simulation process from model design to result analysis. The service-oriented SMP2 M&S activity has three actors; these actors interact with the three layers of the conceptual framework of service-oriented SMP2. These actors are shown in Fig. 2.

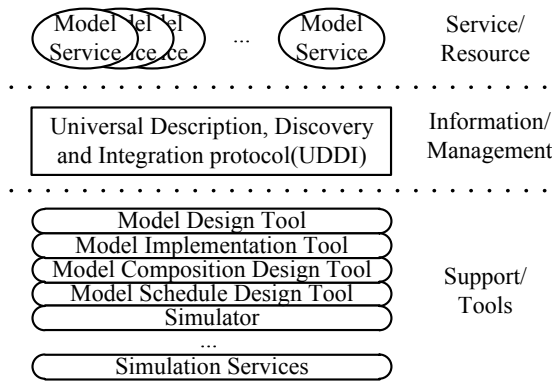


Figure 1. Conceptual framework of service-oriented SMP2.

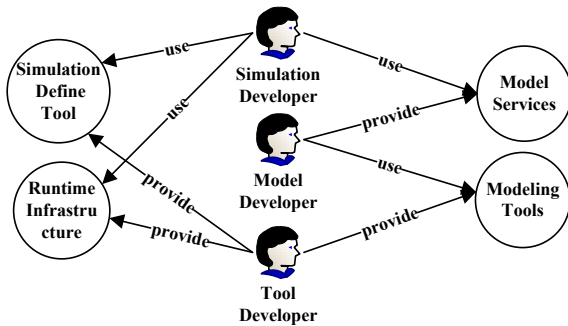


Figure 2. Actors in the service-oriented SMP2 M&S activity.

B. System Architecture

The top level of SMP2 simulation has three levels. These levels represent the different abstract levels, including real world, platform independent model (PIM), and platform specific model (PSM).

The first level is real world. This level is related to the system to be modeled. Important system engineering data are related to model reuse in this level; however, SMP2 does not provide the modeling concept and tools for this level. United Modeling Language(UML) or System Modeling Language(SysML) is utilized in this level.

The second level is PIM level. This level corresponds to SMP2 catalogue, assembly, and schedule. These three define model design, composition, and schedule

information, respectively. The use of assembly and schedule to achieve a dynamic simulation configuration through XML files is optional in this level.

The third level is PSM level, which is the implementation level of SMP2 models. The SMP2 standard provides the mapping method from the SMP2 models to the C++ code and supports the auto-generation of codes. The model deposits (*.dll, *.lib) are integrated into the simulation software after the compilation of the codes. The original SMP2 simulation is end at this stage; however, in a service-oriented SMP2 simulation, the model deposits (*.dll, *.lib) are deployed as services.

A service-oriented distributed and parallel SMP2 simulation system architecture is propounded based on the conceptual framework and top-level architecture of the service-oriented SMP2 simulation system (Fig. 3). The simulation system consists of three parts: simulation nodes, M&S clients, and the simulation information center.

The simulation nodes contain the simulation model, simulator, local simulation service, and simulation coordination service. The initialization, schedule, interaction, and destruction of local models are controlled by the simulator; the local simulation service system controls the simulation logs, events, and time. Simulation control, time synchronization, and the message generated by the different nodes are managed by the simulation coordination service.

M&S clients include modeling and simulation clients. Model developers utilize the modeling client to design, implement, and deploy the model service to the simulation node and to register the model service to the simulation information center. Simulation developers utilize the simulation client to inquire about the model services, define model composition and schedule, and initiate the simulation.

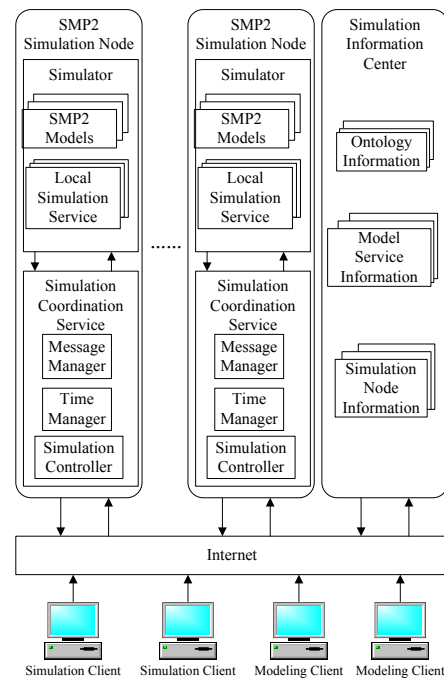


Figure 3. Service-oriented SMP2 simulation system architecture.

The simulation information center (SIC) is an extended UDDI that manages the information on all the simulation

resources to support model discovery and selection. The information includes simulation ontology, model service, and simulation node information.

The simulation client in the service-oriented SMP2 simulation sends an inquiry to SIC to obtain the simulation models. Model composition and schedule are designed and passed to the simulation controller interface service of each SMP2 simulation node. The simulation nodes then initialize the simulation models and perform the simulation.

C. System Engineering Process

The system engineering process provides instructions on how to construct service-oriented SMP2 simulation systems. The main process is shown in Fig. 4, and the steps are detailed below.

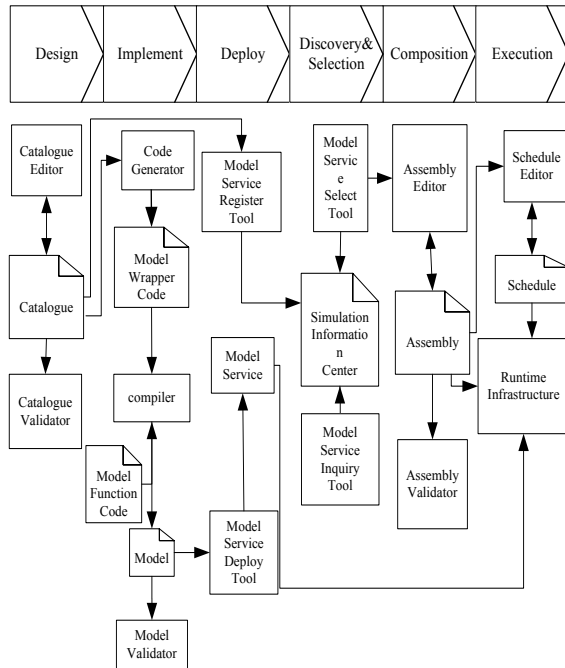


Figure 4. System engineering process of service-oriented SMP2 M&S.

(a) Simulation model design. Simulation model design (catalogue) and validation can be completed by relevant tools. The model developer employs these tools to define the model's attributes, structure, and functions.

(b) Simulation model implementation. The model developer develops the model with function codes after catalogue is transformed to model wrapper code by the code generator. The developer compiles the codes and tests the model through the compiler and model validator.

(c) Simulation model deployment. The implementation model can be deployed to the simulation nodes as a model service. Registration in the simulation information center should be conducted in terms of model design files and relevant information, which can be auto-generated.

(d) Simulation model discovery and selection. The simulation developer obtains information on the model services from the simulation information center. The model services of interest are then selected and download related catalogue files.

(e) Simulation model composition. The simulation developer defines the instances and relationships in the simulation application and validates them through relevant

tools after all the catalogue files that might be useful in the simulation application are downloaded.

(f) Simulation execution. The simulation developer indicates how and when the simulation tasks will be scheduled (transfer and trigger in SMP2) and then validates the tasks through relevant tools. The simulation developer finally executes the simulation through the runtime infrastructure (simulator and simulation services) with the assembly and schedule files.

III. MODEL DEVELOPMENT METHOD UNDER SOA

The model-driven develop paradigm allows the model developer to focus on the model functions, which describe the characteristics of real systems without considering the implementation details of the simulation. The model-driven develop paradigm and auto-mapping technique are also utilized in service-oriented SMP2 M&S to reduce the difficulty of system development. Fig. 5 provides a detailed model develop process under SOA.

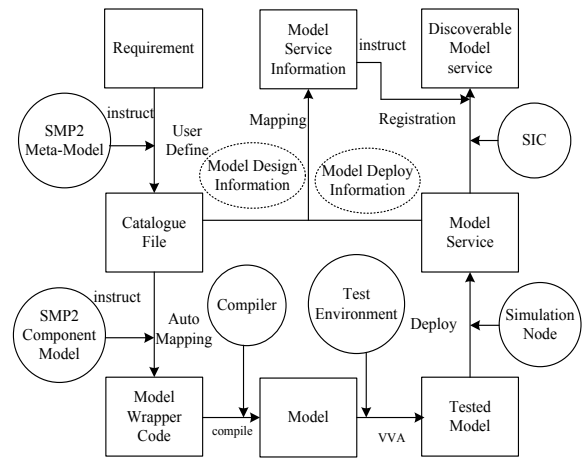


Figure 5. Model develop process under SOA.

Model deployment and registration are two key steps to share resources across the Internet and achieve "simulation in need," which is different from the original SMP2 simulation. Model deploy involves the uploading of the model to the simulation nodes. When the simulation nodes receive a request, they initialize, schedule, and finalize the model instance, which is the basis of the simulation. Model registration involves the uploading of the model service information to the SIC; hence, other users can find and use the registered model service.

Two types of information must be uploaded, namely, function information and service information. Function information contains the details about the model's variables, properties, operations, entry points, event sinks, and event sources. Service information contains the details about the mode's domain ontology, metadata, application information, behavior information, function semantics, and quality information.

IV. MODEL COMPOSITION METHOD UNDER SOA

The simulation models in the service-oriented SMP2 simulation are distributed in the different simulation nodes in the form of service. Two model compositions exist. The first is the composition of models inside a simulation node, which can be achieved in a traditional manner. The other is the composition of models with different nodes. A

proxy-based simulation model service composition method is proposed at this stage to establish the service-oriented distributed model composition.

A model proxy is utilized to localize the remote model service. Every model proxy stands for a remote model. A model proxy should have an interface similar to that of the remote model and should employ the Web service interface to initialize the corresponding interface functions. A complete model instance tree is generated in every simulation node. The local models are directly maintained in the memory in the form of model instance, whereas the remote models are maintained in the memory in the form of model proxy. The model proxies in the proxy-based composition should perform the functions of the remote models and map those functions to the remote models. These functions include interface calls and event and data stream reception. Entry point calls and event and data stream sending and delivery should be scheduled by the simulator in the simulation node in which the remote models reside.

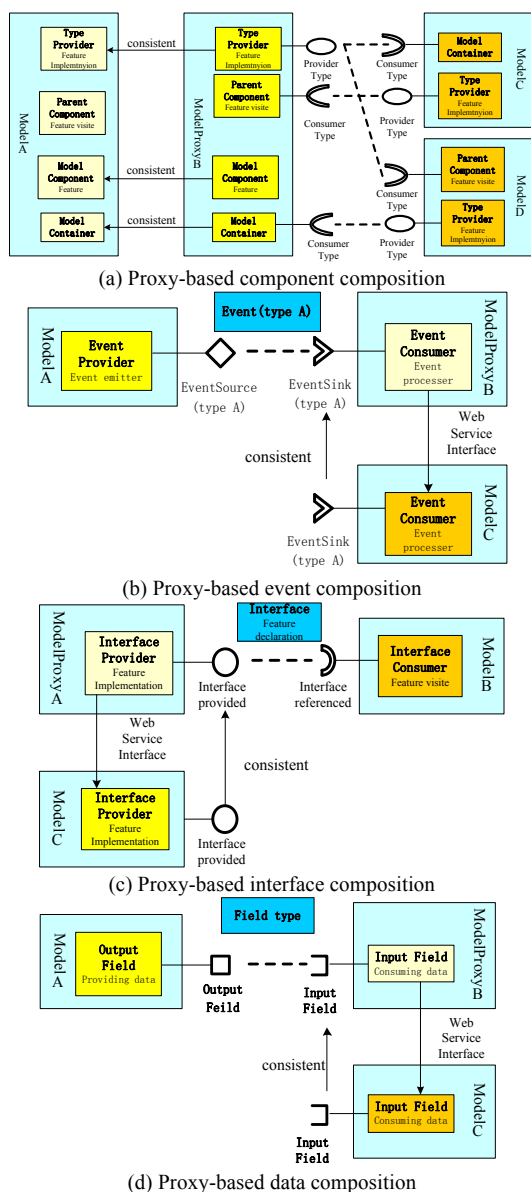


Figure 6. Proxy-based SMP2 model service composition.

SMP2 simulation models have four traditional compositions: component composition, interface composition, event composition, and data composition. The proxy-based SMP2 model service composition modes are shown in Fig. 6.

V. MODEL SCHEDULE METHOD UNDER SOA

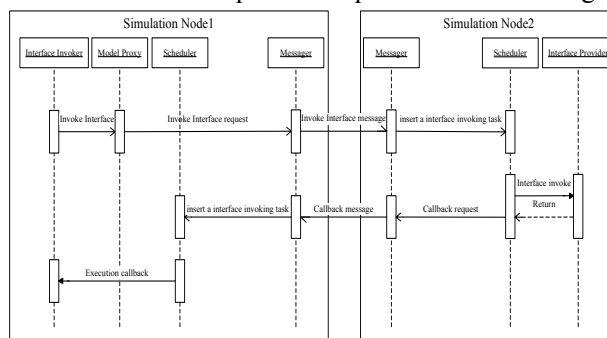
SMP2 simulation systems based on SOA contain many simulation nodes in their network. Every node contains a simulator and simulation services to manage and schedule simulation models.

SMP2 simulations involve four kinds of scheduled tasks: interface-invoking task, event-sending task, entry point task, and field transfer task. Parallel schedules occur only between different simulation servers and not inside each simulation node in service-oriented SMP2 simulation. A parallel schedule mainly occurs among the different model instances in the different nodes. The schedule of entry point tasks is not related to the other simulation nodes; thus, a parallel schedule only occurs in the three schedule tasks.

The three schedulable tasks are redefined below. The asynchronous schedule process is defined and shown in Fig. 7.

The following three peculiarities of the interface-invoking task are not suitable for parallel schedule: 1) out of the management of simulator, 2) out of the simulation time stamp and occurring instantly, and 3) synchronically invoked. Some improvements are therefore suggested to acquire a distributed parallel interface invoke among the model instances. The suggestions are as follows: 1) defining the three stages of interface invoking: request, process, and callback, 2) assigning a time stamp to the process and callback, and 3) defining the callback function.

The following two peculiarities of the event-sending task are not suitable for parallel schedule: 1) out of the management of the simulation and 2) out of the simulation time stamp and operates instantly. Some improvements are suggested to achieve distributed parallel event sending among the model instances. The suggestions are as follows: 1) defining the two stages of event sending: request and process and 2) assigning a simulation time stamp to the request of event sending.



(a) Asynchronous schedule process of the interface-invoking task

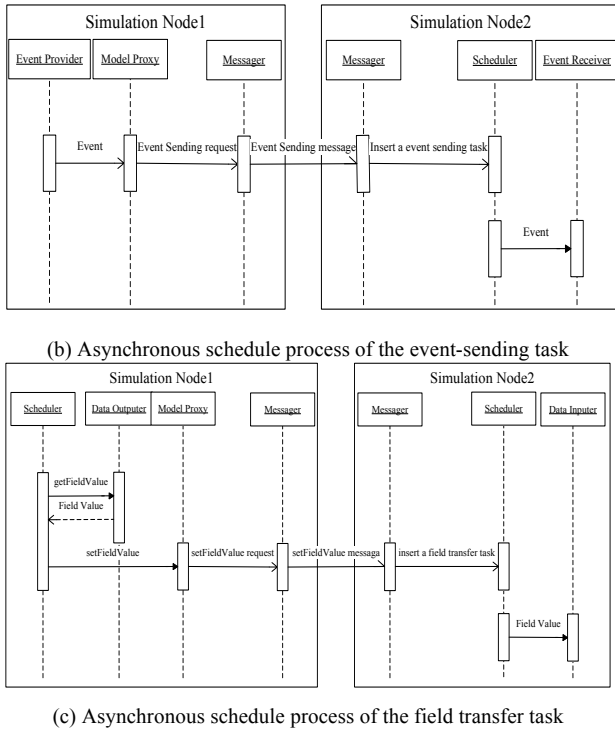


Figure 7. Asynchronous schedule processes.

Out of the simulation time stamp and operates instantly is the main peculiarity of the field transfer task that is not suitable for parallel schedule. Some improvements are suggested to achieve a distributed parallel field transfer among the model instances. The suggestions are as follows: 1) defining the two stages of field transfer: output and input and 2) assigning a simulation time stamp to the request of field input.

VI. EXAMPLE

An environment was designed and implemented to support the proposed service-oriented SMP2 M&S. The environment consists of four subsystems and a series of tools. Fig. 8 shows the structure of the prototype environment.

GNSS consists of a satellite subsystem, rocket system, operation control system, user subsystem, launching site subsystem, and TT&C subsystem. Service volume simulation (SVS) is mainly utilized for navigation constellation plan evaluation and navigation performance analysis and validation to support the system integration, functional verification, performance validation, and analysis of GNSS. We study the simulated performance of the GALILEO system through the aforementioned method. Based on the navigation performance indices, we define 50 E - 150 E, 0 N - 60 N as the analysis region and 5×5 as the grid resolution. The analysis considers visibility, geometry, coverage, grid dilution of precision (DOP), NSP, and other factors. We employ a Keplerian two-body orbital model as the satellite orbit determination algorithm. The simulation cycle is a constellation regression cycle. The simulation step lasts for 300 seconds, and the algorithms involved in the model include GPS and GALILEO.

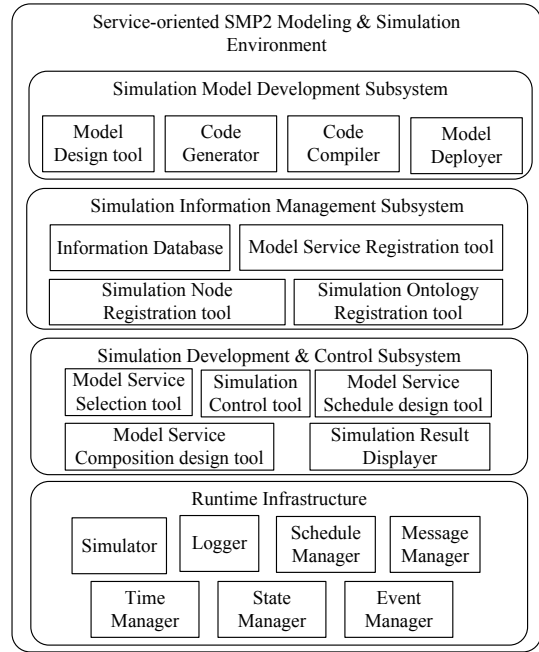


Figure 8. Structure of the service-oriented SMP2 M&S environment.

Model design, code generation, deployment, selection, composition, and schedule design are presented in Fig. 9. The DOP analysis curve, which is a part of the result of the navigation system volume simulation, is also provided in the figure.

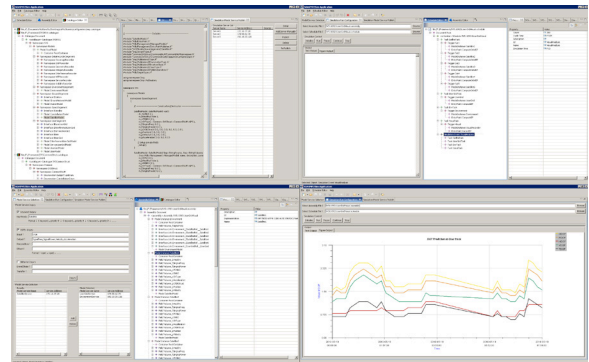


Figure 9. Navigation system volume simulation and result of grid dilution of precision (DOP) analysis.

VII. CONCLUSION

A service-oriented SMP2 M&S method for GNSS was proposed in this paper. The separation of the model, simulator, and simulation function in SMP2 and the loose-couple of SOA made the service-oriented SMP2 simulation system extensive, agile, efficient, and facilitative.

Further research involving the system dynamic evolvement method is important to improve the simulation system's reliability and flexibility. Reliability allows the simulation system to select a substitutable service or migrate the original service to ensure the successful execution of the simulation when the simulation component services encounter an error or when the application requirements change. Flexibility allows the simulation system to change its structure or members (such as load balance) according to the network states to meet the requirement.

SPECIAL FOCUS PAPER
SERVICE-ORIENTED SMP2 MODELING AND SIMULATION FOR GNSS

ACKNOWLEDGMENT

The authors would like to thank the reviewers of this paper for their valuable comments and advice. The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

REFERENCE

- [1] B. H. Li, X. D. Chai. Some focusing points in development of modern modeling and simulation technology. *Journal of System Simulation*, 2004, 16(9): 1871—1878(in Chinese).
- [2] European Space Agency. SMP 2.0 Handbook Issue 1 Revision 2. EGOS-SIM-GEN-TN-0099, 2005.10.
- [3] European Space Agency. SMP 2.0 Metamodel Issue 1 Revision 2. EGOS-SIM-GEN-TN-0100, 2005.10.
- [4] European Space Agency. SMP 2.0 Component Model Issue 1 Revision 2. EGOS-SIM-GEN- TN-0101, 2005.10.
- [5] Y. L. Lei, N. L. Su, J. J. Li, et al. A New simulation model representation specification SMP2 and its key application techniques. *System Engineering Theory & Practice*, 2010, 30(5): 899~ 908. (in Chinese)
- [6] Q. Li, C. Wang, Y. F. Zhu, et al. Research on MDA Based Simulation Model Development and Integration Methodology. *Journal of System Simulation*, 2007, 19(02): 272-276. (in Chinese)
- [7] Q. Li, C. Wang, W. P. Wang, et al. Design and Implementation on Simulation Engine Compliant with SMP2.0. *Journal of System Simulation*, 2008, 20(24): 6622-6626. (in Chinese)
- [8] N. L. Su, H. T. Hou, Q. Li, et al. Simulation Analysis of Satellites' Regional Visibility Based on SMP2. *Journal of System Simulation*, 2008, 20(9) : 2253-2256. (in Chinese)
- [9] Y. L. Lei, W. Zhang, X. Zhao, et al. Research of SMP2-based Missile Countermine Simulation System. *Journal of System Simulation*, 2009, 21(14): 4312-4316. (in Chinese)
- [10] H. T. Hou, Y. Liu, Q. Li, et al. Research on Model Integration of Navigation System Service Performance Simulation. *Journal of System Simulation*, 2009, 21(6):1763-1768. (in Chinese)
- [11] X. Zhao, Y. L. Lei, Q. Li, et al. Design and Implementation of Scenarios Edit System for SMP2-based Simulation of Missiles

Attack-defense, *Journal of System Simulation*, 2009, 21(11): 3238-3240. (in Chinese)

- [12] N. L. Su, H. T. Hou, F. Yang, et al. Optimistic Parallel Discrete Event Simulation Based on Multi-core Platform and its Performance Analysis. *International Workshop on Multi-Core Computing Systems*, Fukuoka, Japan, March 16-19, 2009, IEEE CS Press.
- [13] E. Thomas. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR. 2005.08.

AUTHORS

H.T. Hou is now with the College of Information Systems and Management, National University of Defense Technology, Changsha 410073 China. His current research interests include flexible simulation theory, virtual simulation, distributed and parallel simulation, satellite navigation system simulation and system of systems engineering. (e-mail: houthongtao@nudt.edu.cn)

F. Xie is now with the Institute of Nuclear and New Energy Technology, Tsinghua University , Beijing 100084 China. (e-mail: xf@tsinghua.edu.cn)

C. Wang is with the Electrical Engineering Department, Wuhan 430035 China, on leave from the College of Information Systems and Management, National University of Defense Technology, Changsha. (e-mail: wangchaonudt@gmail.com)

W.X. Zhang is now with the College of Information Systems and Management, National University of Defense Technology, Changsha. (e-mail: zhangwangxun2010@163.com)

This work was supported in part by the National Science Foundation of China (60974074,61074107). Submitted, January 17, 2013. Published as resubmitted by the authors on May 16, 2013.