

Software Design and Implementation of Management and Control System in GNSS Simulation Platform

Pengfei Zhang^{1,2}, Chengdong Xu^{1,2}, Chunsheng Hu² and Dan Song²

¹Key Laboratory of Dynamics and Control of Flight Vehicle, Ministry of Education

²School of Aerospace Engineering, Beijing Institute of Technology,

No.5 South Zhongguancun Street, Haidian District, Beijing, 100081, China

e-mail: successful.2008@163.com

Abstract—As the management and control center of Global Navigation Satellite System (GNSS) simulation platform, Management and Control System (MCS) is responsible for input and output interfaces of the whole platform. The design quality of MCS has a great influence on the work efficiency of the platform. In this paper, MCS software is designed and implemented. Firstly, the function requirements of each module in MCS are analyzed. Secondly, the general framework and the data interaction relations of MCS are presented. Then the key techniques in MCS are described in detail, including design of software interface, selection of models management schemes and research on visualization method. At last, MCS developed in this paper is demonstrated and tested partly. Taken as a whole, the MCS has a friendly human-computer interaction interface, complete functions and flexible operation procedure. The MCS has applied to GNSS simulation platform, and the functions of each module in MCS run smoothly.

Keywords—GNSS; Management and Control System; software design; models management; visualization

I. INTRODUCTION

Global Navigation Satellite System (GNSS) simulation platform can provide validation means and tool support for multi-GNSSs compatibility and interoperability performance evaluation, signal transmission environment research and receivers performance test [1]. It can also reduce investment in real tests and real platforms. Management Control System (MCS) is the management and control center of the platform, it is responsible for coordination and management of Signal Generation System (SGS), Space Environment Simulation System (SESS), Signal Receiving and Processing System (SRPS) and Performance Evaluation System (PES) [2]. In addition, MCS is in charge of recording, storing, displaying the simulation data and some interaction control. Meanwhile, MCS monitors the running state of instruments and equipments in the platform. Since 1980s the American GPS was put into service, domestic and abroad relevant research institutes have developed a variety of navigation simulation software, including STK software developed by American Analytical Graphics Corporation, SimGEN software developed by England Spirent Corporation and PANDA software developed by domestic Wuhan university GNSS center [3][4]. Compared with the existing achievements in related fields of domestic and abroad, MCS of GNSS

simulation platform designed in this paper has more complete functions and more convenient operations. MCS can provide extensible navigation simulation task design of multi-GNSSs and multi-scenarios, it can also establish communication connections and send simulation tasks to other systems. And then, MCS can receive the output data of each system and store them in databases. Moreover, integrated display module is designed in MCS to visualize the important data of the platform so that the simulation results can be shown more intuitively.

In this paper, the function requirements of each module in MCS are analyzed firstly, and then the general framework and the data interaction relations of MCS are presented. Next the key techniques in MCS are described in detail, including design of software interface, selection of models management schemes and research on visualization method. At last, MCS is demonstrated and tested partly.

II. REQUIREMENT ANALYSIS

MCS is responsible for the management and control of the whole GNSS simulation platform, including the design and management of simulation tasks; distribution of the tasks; receiving and storing of the simulation data; monitoring of the running state of instruments and equipments in the platform; visualization of simulation data and equipment state. In this paper, according to general requirements, MCS is divided into several modules as follows: simulation task design module, simulation task running module, data management module, models management module and integrated display module. The function requirements of each module are shown in Fig. 1.

- Simulation task design module is responsible for displaying relative operations of the task design interactively, including setting of simulation task parameters, selection of simulation models, definition of scenarios and so on.
- Simulation task running module is responsible for displaying relative operations of the task running interactively, including setting of servers' connection information, control of simulation command, monitoring the state of instruments and equipments. This module sends the corresponding data packet based on different simulation commands, and it receives the output data of different servers in a

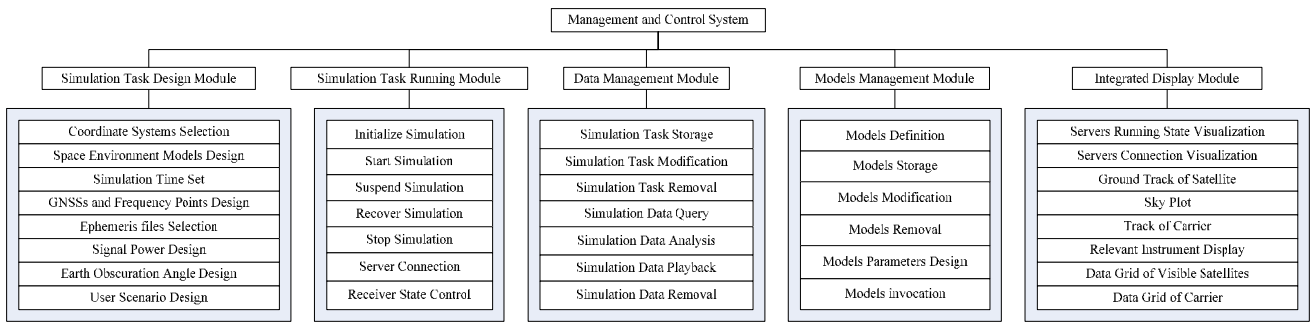


Figure 1. Function requirements of each module in MCS.

thread respectively as long as there is any one server sends data to MCS.

- Data management module takes charge of establishing databases to store relevant personnel information, simulation tasks and simulation data. The module should provide the ability of calling back, modifying and deleting the information which are stored in these databases. The function of simulation data playback can also be achieved.
- Models management module should establish a database to store simulation models, and it should also achieve the function of defining models, adding models, modifying models and deleting models. Meanwhile, it is in charge of calling back and displaying the simulation models existing in the database.
- Integrated display module is responsible for visualization of simulation data, including subpoint track diagram of satellites, sky plot, vehicle trajectory diagram, table of visible satellites information and so on. And what's more, it also takes charge of displaying the state of each instrument and equipment, including the running state of them and the servers' connection information.

III. GENERAL FRAMEWORK DESIGN OF MCS

According to function requirements of MCS and its each module, the general framework of MCS is designed as shown in Fig. 2.

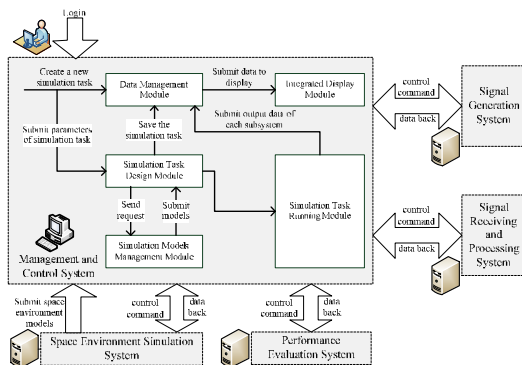


Figure 2. General framework of MCS.

In Figure 2, the call relations among modules in MCS and data transfer relations among MCS and other systems are shown. The workflow of MCS is as follows:

- The operator logs in MCS, the login information of the operator is stored in data management module by MCS;
- The operator creates a new simulation task or opens an existing simulation task, the simulation task information (including task name, task storage path, task description and so on) is stored in data management module by MCS;
- The operator adds simulation models which are commonly used and some other models which are provided by SESS into MCS, then the models and their essential information are stored in simulation models management module;
- For different tasks, the operator sets different simulation parameters in simulation task design module, and then these parameters will be stored in a xml document which represents the simulation task created or opened before;
- The operator connects MCS and SGS, SESS, SRPS, PES respectively in simulation task running module through internet according to software interface which will be introduced in the fourth part. Then MCS sends the corresponding task to SGS, SESS, SRPS and PES respectively. When MCS receives response from each system, MCS starts to send relevant control command. Each system executes the command which they have received from MCS, and then they return the output data (including the running state of system and simulation results) to simulation task running module in MCS. MCS submits these data to data management module for storing and calling;
- Integrated display module calls the data stored in data management module to analyze and visualize.

IV. KEY TECHNIQUES IN MCS DESIGN

A. Design of Software Interface

Design of software interface mainly involves information interaction mode between MCS and other systems. The information which needs to interact is divided into state information and data information. The interaction of state

information uses the User Datagram Protocol (UDP), and the interaction of data information uses Transmission Control Protocol (TCP) [5]. UDP is a connectionless protocol, it doesn't need to establish a connection before sending data. The state information needs to upload at any time without request and broadcast to the whole network. When the state of a system changes, other systems need to monitor it and take corresponding measures timely. So the interaction of state information uses UDP. TCP is a reliable and connection-oriented protocol. That is, a logical connection should be established before data transmission. So the interaction of data information uses TCP so that the transmission is more specific and more reliable [6]. In order to manage conveniently, this paper takes MCS as a client and other systems as servers. The data service process between a client and a server is shown in Fig. 3.

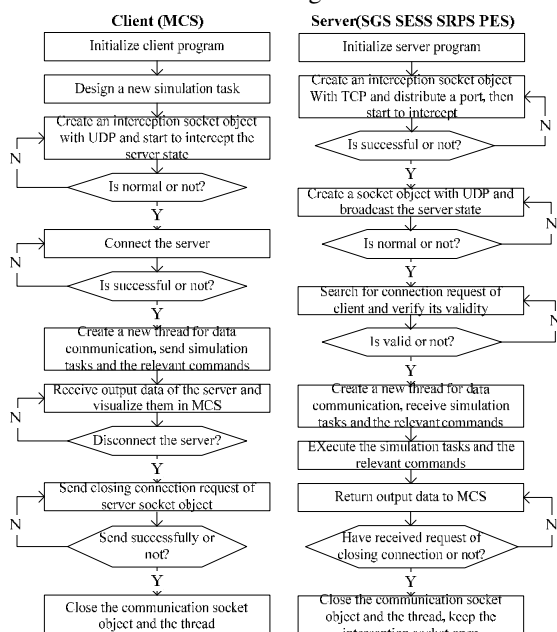


Figure 3. Data service process between a client and a server.

The server broadcasts its state with UDP, when the server state received by the client with UDP is normal, the client connects the server with TCP. Then the client creates a new thread for data communication and sends a simulation task or relevant commands to the server, the server creates a new thread to receive the simulation task or the relevant commands at the same time. Next, the server executes the commands and returns output data to the client, the client receives the data and visualizes them in the integrated display module. The transmission data packet format between the client and the server is designed as Table I.

TABLE I. THE TRANSMISSION DATA PACKET FORMAT

Data Items	Child Items	Length
Header	Mark	4 bytes
	Category	1 byte
Body	Information	Variable-length
Footer	Check Code	4 bytes

In Table I, the category is the unique identification of a data packet. In other words, which data packet is received can be identified by 1 byte category in this data packet.

B. Selection of Models Management Schemes

There are mainly two kinds of models management schemes in domestic and abroad simulation analysis software at present, they are centralized management and decentralized management respectively [7]. For the former, the simulation models are put together in the client for centralized management. When designing the simulation task, the models will be called directly or sent to the server to call. This scheme can prevent the models from modifying at will, it is convenient to control and manage the corresponding version [8]. Simulation models can be sent to any one server so that the server scalability can be increased. However, if the simulation models need to be installed, the operator will install the models in the server manually, thus the simulation task will not be run automatically. For the latter, the simulation models are put in every server. When designing the simulation task, simulation task running module calls the models in each server directly. This scheme decreases the transmission data quantity and simplifies the running process. However, the scheme has an against influence on the scalability of the server.

The simulation models in MCS contain both models which are needed to install (executable program .exe model) and models which are not needed to install (dll model and matlab model). So the mixed management scheme which combined with centralized and decentralized management is taken. For a model which is needed to install, decentralized management scheme is selected. The model should be installed in the specific server at first, then only the general information should be written to the database. When the simulation task is running, the specific server is requested to connect according to the recorded information in database, and then the model is called in the server. For a model which is not needed to install, centralized management scheme is selected. Besides the general information, the model file and other supporting files should be written in the database. When the simulation task is running, the model can be sent to any one server which has been connected as needed.

C. Research on Visualization Method

The integrated display module in MCS is responsible for visualization of data provided by data management module. The data are classified to visualize in different styles, such as visualization of control flow and data flow, subpoint track diagram of satellites, sky plot, vehicle trajectory diagram, table display of observation data and other graphs. In this paper, Windows Presentation Foundation (WPF) applications are selected as a tool of interface development. WPF is an exhibition layer development framework based on XML, .NET Framework and vector drawing technology. A new language of eXtensible Application Markup Language (XAML) is used to develop the interface so that the background program can be apart from the interface well. Thus MCS development will be more flexible [9]. According to the characteristics of the visualized data and the

application background of MCS, two plugins Bing Maps and Develop Express are selected. The two plugins can make the interface more beautiful and make the visualized content more intuitive. As to subpoint track diagram of satellites and vehicle trajectory diagram, a traditional visualized method is to select a picture as a map. But the method cannot be zoomed, and it is not flexible. In this paper, a plugin Bing Maps is chosen. The Bing Maps WPF Control Software Development Kit (SDK) can let developers integrate Bing Maps into rich WPF applications. Because it provides the binaries and programming reference for the Bing Maps WPF Control. Shapes, media and pushpins can be added to the map, thus the subpoint track diagram of satellites and vehicle trajectory diagram can be visualized intuitively [10]. The panels visualization and the overall interface style choose a plugin Developer Express WPF Controls as a development tool. It has rich functions and is easy to use.

V. IMPLEMENTATION AND TEST OF MCS

MCS chooses high performance computer which installs Windows XP or a newer operating system as hardware platform, and it chooses C# in visual studio 2010 as software development platform. The calculation processing capacity of high performance computer and the stability of operating system are better used. Meanwhile, as a completely object-oriented language, C# synthesizes the characteristics of structure abstract and function abstract so that it is convenient for modular design of MCS. Also, the language can achieve the scalability and polymorphism of software and it is easy to modify. So it can satisfy the requirements of MCS development completely.

MCS is implemented through development of each module in this paper. The main software interface is shown in Fig. 4.

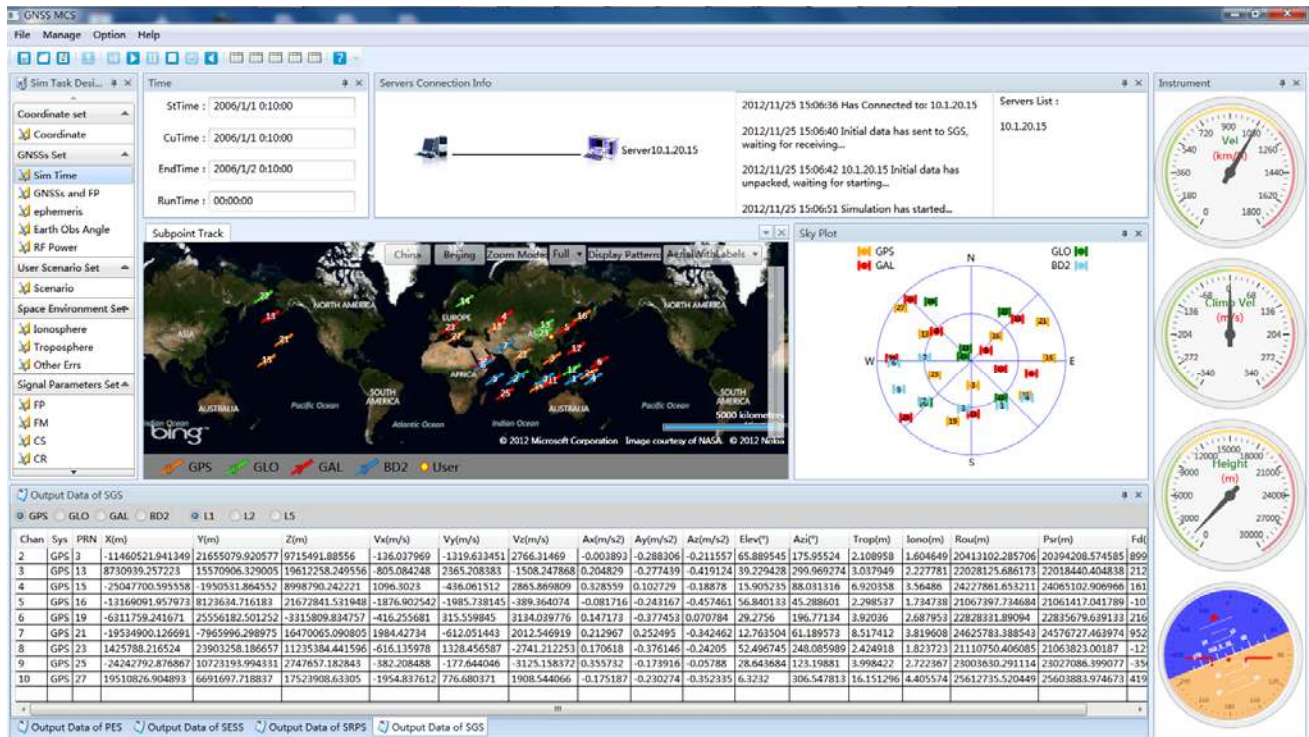


Figure 4. The main software interface.

Taking a sever SGS as an example, part functions of MCS are tested. Firstly, the operator logins MCS and creates a new task through simulation task module shown in the left column of Fig. 4. Secondly, MCS starts to intercept the server state, MCS will connect to SGS if the state of SGS which is broadcasted is normal. Thirdly, MCS sends the task to SGS through the network and sends “Start” command to SGS. SGS starts to simulate and sends the output data to MCS, MCS receives the output data to store in the database and visualize them in integrated display module. Some test results can be shown in Fig. 4. Vehicle trajectory can be drawn according to the vehicle data in all output data

returned by SGS. Vehicle trajectory program is shown in Fig. 5.



Figure 5. Vehicle trajectory program.

In the above test, only part of the MCS functions is tested. Due to the limitation of the paper length, other functions of MCS cannot be tested one by one. Fortunately, MCS designed by this paper has been used to GNSS simulation platform.

VI. CONCLUSION

The function requirements of each module in MCS of GNSS simulation platform are analyzed in this paper. Then the general framework is come up with and the data interaction relations of MCS are introduced. Next, the key techniques in MCS design are described in detail, including design of software interface, selection of models management schemes and research on visualization method. MCS designed and implemented in this paper is demonstrated and tested partly at the end of the paper. Taken as a whole, MCS designed by this paper has a friendly human-computer interaction interface, complete functions and flexible operation procedure. In the models management module, the mixed management scheme which combined with centralized and decentralized management is selected so that both the advantages are taken. In particular, in the integrated display module, two plugins Bing Maps and Develop Express are used so that the visualization results are more beautiful and the visual feeling is more real.

ACKNOWLEDGMENTS

This work was supported by the National High-Tech. R&D Program, China (No.2011AA120505) and the National Natural Science Foundation, China (No.61173077).

REFERENCES

- [1] Yue Wang, Minjian Zhao, Jie Zhong, Liyan Li, and Xuanxuan Lv, "Design and implementation of programmable multi-mode GNSS signal simulator," International Conference on Communication Technology Proceedings (ICCT), IEEE Press, Nov. 2010, pp. 865-868.
- [2] S.Lim, D. W. Lim, M. Liu, S. W. Moon, C. Park, and S. J. Lee, "Design of a Software-based Multi-Channel GNSS IF Signal Generator," International Conference on Control, Automation and Systems, Oct. 2008, vol. S1-4, pp. 652-656.
- [3] Pengfei Zhang, Chengdong Xu, Chunsheng Hu, Ye Chen, "Coordinate transformations in satellite navigation systems," International Conference on Electronic Engineering, Communication and Management (EECM2011), Springer Verlag, Dec. 2011, pp. 249-257.
- [4] Pengfei Zhang, Chengdong Xu, Chunsheng Hu, Ye Chen, "Time Scales and Time Transformations among Satellite Navigation Systems," The 3rd China Satellite Navigation Conference (CSNC2012), Springer Verlag, May. 2012, pp. 491-502.
- [5] Ian Sommerville, Software Engineering, 9th. ed., St Andrews: Addison-Wesley, 2011, pp.95-135.
- [6] Hua Cai, Qile Zhao, Hanrong Sun, and Zhigang Hu, "GNSS real-time data quality control," Geomatics and Information Science of Wuhan University, vol. 36, Jul. 2011, pp. 820-824.
- [7] Wenguang Li, Tao Xing, and Guangshuai Wang, "Software Design and Implementation of Automatic Mass GPS Receiver Quality Inspection System," International Conference on Energy and Environmental Science (ICEES), Oct. 2011, vol. 11, doi:10.1016/j.egypro.2011.10.268.
- [8] L. G. Arvanitis, B. Ramachandran, D. P. Brackett, H. Abd-EI Rasol, and X. S. Du, "Multiresource inventories incorporating GIS, GPS and database management systems: a conceptual model," Computers and Electronics in Agriculture, vol. 28, Aug. 2000, pp. 89-100, doi: 10.1016/S0168-1699(00)00124-1.
- [9] Matthew MacDonald, Pro WPF in C# 2008: Windows Presentation Foundation with .NET 3.5, 2nd. ed., CA: Apress L. P., 2008, pp. 64-500.
- [10] Ivanov Rosen, "Real-time GPS track simplification algorithm for outdoor navigation of visually impaired," Journal of Network and Computer Applications, vol. 35, Sep. 2012, pp. 1559-1567, doi: 10.1016/j.jnca.2012.02.002.