

# Study on Data Collection System in Joint Warfare Material Support Simulation System Based on HLA

Junbin Shi, Hao Xing, Xiaotao Bai

Changchun Institute of Engineering Technology, Changchun, Jilin, 130117, China

**Keywords:** Joint Warfare Material; Support Simulation System; Data Collection System

**Abstract.** Joint Warfare Material Support Simulation System (JWMSSS), which is a large-scale hybrid complex system, will produce a great deal of data in its running process. These data are of various types and have complex structures which make the collection of simulation data quite difficult. According to the features of JWMSSS, the following factors such as minimizing influences of Data Collection System on JWMSSS, collecting as more public data and appointed private data as possible and logging in and reading data rapidly as well are considered in designing Data Collection System (DCS). Correspondingly, the paper also put forward means for accomplishing DCS from the designing of the architecture of the DCS, the generality of the DCS modules and the simulation database.

## Introduction

With its development as the basic warfare form of informational local war, the joint warfare relies increasingly on the materiel support and makes it the key factor for the victory or defeat of the joint warfare. Accordingly, it's vital to apply HLA simulation technology to construct distributed joint warfare materiel support simulation system, imitate the activity and process of the materiel support, and analyze its outcome in the construction and development of the JWMSSS. In the process, the collection of simulation data is the key procedure in the application of JWMSSS.

JWMSSS, which is a large-scale hybrid complex system, will produce a great deal of data in its running process. These data are of various types and have complex structures which make the collection of simulation data quite difficult, restrict the further development of the simulation system scale and influence the performance of the support system. Consequently, the following factors are mainly considered in designing the data collection system: to minimize computing the framework of the support simulation system and the influence of the network burden to the data collection system; to collect public data and appointed private data as more as possible; to storage and read collected data rapidly. Based on the characteristics of the JWMSSS, furthermore, the paper proposes the corresponding solutions to accomplish the data collection system from the perspective of design of the architecture of the DCS, the generality of the DCS modules and the database.

## Design of the generality of DCS

The generality of DCS was put much more concern in designing it that the system could be applied to JWMSSS as well as other simulation systems. The key point in the generality accomplishment is to reconcile the conflict between the independency of some simulation application and the dependency of some simulation process. From the perspective of technology, the independency of some simulation application refers to the non-subscription for the relevant information for simulation application in the federation in the system design and the dependency of some simulation application refers to the subscription for certain information for simulation application in the system running.

Traditionally, two ways are available for implementing the generality of the system. One is using the different functions provided by MOM, the other is using dynamic compilation.

The generality accomplishment by means of using MOM will sacrifice some functions of DCS and acquire limited information due to the restriction for the function of MOM. There are two problems to solve by using dynamic compilation to accomplish the generality of DCS. One is that

the workload of dynamic compilation is great and the time consumption is much when especially compiling large-scale federation and the complicated compiling process should be repeated once any modification of the FOM was made. The other is the difficulty to accomplish the generality for no interface to other systems. Based on the above problems, the method for the design of the generality of DCS is proposed as follows.

The design for the generality of DCS is achieved together by three mutual related modules. The structure is as follows.

#### (1) Configuration module

The running process of the configuration module is to extract the necessary information from FOM/SOM, its functions mainly include:

- Read OMT files, illustrate object classes and interaction classes information of the simulation application in tree-structure. The user determines object classes and interaction classes to be subscribed according to actual requirement and save the data structure.
- Set the simulation running environment, e.g., the federation name, simulation time step- length, time advance policy, etc.
- Select out all those complicated and enumerating data structures.
- Save the object classes and interaction classes to be subscribed and the configuration of the running environment as well as the data structure as XML format to create configuration files for dynamic subscription module and unpack modules.

In configuration module, users subscribe necessary object and interaction classes can concentrate on the professional knowledge and decrease the network and RTI burden and improve the running efficiency. In addition, configuration module saves the complicated and enumerating data structure as character strings in the simulation application that helps confirm the data structure and identify and parse the data dynamically by comparing data-type names in later unpacking process .

#### (2) Dynamic subscription module

Dynamic subscription module reads configuration files at first and obtains the subscribed object class names and its attributes, interaction class names and its parameters, then it acquires handles of object classes and its attributes, interactions and its parameters using RTI agent functions. Finally, it subscribes the necessary object classes and interaction classes using RTI functions based on the preceding acquired handles.

The dynamic subscription is fully demonstrated in the process, which the object classes and interaction classes subscribed by DCS are variable according to the information provided by the different configuration files. So the generality of simulation application is accomplished from this perspective.

#### (3) Unpacking module

Instances of object classes and interaction classes may be discovered and their attributes and parameters may be reflected in the simulation process after the subscription. Since no attribute and parameter definitions are available, the data must be unpacked by comparing the data-type names.

Before unpacking, the initialization work must be well done. The work includes two aspects of content. First is to name every object class (interaction class) and all their attributes (parameters). To make sure every datum has a unique name, the following name principle should be employed: to name it by attribute name if its attribute value is a simple data type, by attribute name and domain name of complicated data type if its attribute value is a complicated data type and add other domain names till every domain be a simple data type if there are other complicated data type. Furthermore, the gap between data must be concerned in the unpacking process to guarantee validity of the data.

The unpacking procedure must undergo two steps in simulation process. First is to record the discovered instances in discovered-instance call-back functions of the federation agent RTI. The instances information should be stored in data arrays and all attribute names of the instance should be constructed an attribute table based on above name rules according to its object class and its data structure to provide storage space for the next parsed data. Next is to unpack the data in reflected-attribute call-back functions of the federation agent RTI. All necessary information to unpack can be acquired by the above configuration module, dynamic subscription module and

pretreatment of complicated data structure.

### **Designing the DCS architecture**

In the HLA, the federation can define the format of information to exchange between federates optionally. This makes the distributed simulation environment more flexible and at the same time the flexibility makes collecting data more complex. It's impracticable to connect a concealed data logger to the network to recognize and record all simulation exchanged information and storage it in a log file. Therefore, three types of data collection projects, such as centralized, separated and distributed data logger, are adopted in the HLA.

#### **1) Centralized data logger**

Centralized data logger is to select a data logger in the simulation system and attach it to any position of the network as long as it subscribes data sufficiently. As the data are centralized, it's easy to analyze simulation data and recall the simulation process, but a great deal of bandwidth resource will be occupied because all updating any entity attribute and sending interaction must flow into the logger across routers in the process of simulation. This is prone to be the bottleneck of the data collection and cause the lost of entity attribute value and the interaction parameters value in the large-scale WAN simulation. Therefore, this mode is only fit for the distributed simulation of small-scale, unitary bus or circular networks.

#### **2) Separated data logger**

Separated data logger is to attach rational amount of loggers to proper position in the network. In the distinct segmental interactive simulation network, several loggers are considered to use for the layout and subscription of the simulation entity attributes and interactions to make sure that the simulation data would be transferred to the closest logger with the lowest cost of network burden and bandwidth resource. So this project can sustain the large-scale distributed interactive simulation. At the same time, all these loggers may increase the data streams and repeatedly record the same data or omit some valuable data due to different loggers subscribing the same data repeatedly. Furthermore, as the collected data are distributed, it's difficult to analyze and recall the simulation data.

#### **3) Distributed data logging**

No special loggers are used in the distributed data memory and the simulator itself would serve to record data. All those data could be completely recorded during the inflow and outflow process of all simulation entities data (updated attributes and interaction parameters). With the data type mainly determined by the amount of simulators, having no additional data transferred and network burden, this means of data collection will definitely take up the time and space resources of the simulator, which will greatly decrease the performance of simulation entities and increase the difficulty for data analysis and replay of the simulation process due to the collected data distributing in respective simulators.

From the above analysis, we can conclude that separated data logger which may slightly increase the network burden without any influence for the performance of simulators can make the data collection comparatively concentrated. Consequently, this project would be applied to determining the architecture of data collection system in JWMSSS.

### **Design the temporal database for distributed simulation**

The temporal information of data is crucial in data collection process and has an important status in the simulation application. But the available temporal database technology was designed for special domain so far as the historical relational database model or bi-temporal database model is concerned and neither can be fit to collecting the simulation system data. Therefore, based upon the historical relational database model, we propose the simulation temporal database model for collecting real-time distributed simulation data by expanding the traditional relational database model.

This is a temporal database model. It can record and manage states of respective attributes of the

entities in the life cycle of the simulation process. The stamp information used in the model manages the life cycle states of attributes of entities impliedly, which is the states information of entity attributes in some stamp duration  $[t_1, t_2]$ . It is expanding the traditional relational database and a snapshot of the simulation system at the stamp point, which is a traditional relational database, can be obtained when the life cycle  $[t_1, t_2]$ , shortens to some stamp point. The operation of RDB such as selection, mapping, connection, can be expanded correspondingly in this model.

## Conclusion

Based on data analyses of JWMSSS, the paper designed the DCS and proposed the corresponding projects from the architecture, generality and database. In decreasing network burden in maximum and solving the bandwidth, the DCS can support large-scale distributed interactive simulation for JWMS and accomplish the generality in other different simulation federations. Although the simulation temporal database applied in DCS can achieve the immediate data collection, real-time query and efficient replay of the data of the large-scale distributed simulation system, the advantage of DCS in the postmortem data analyses is not so obvious and needs further research.

## References

- [1] Wu Zebing, Li Weiqing, Jiang Xiajun and Wu Huizhong. "Method for Distributed Simulation Data Collection Based on Temporal Database," Journal of System Simulation.Vol.18 No.5.pp.1217-1221, May, 2006.
- [2] Ju Rusheng, Qiao Haiquan, Qiu Xiaogang and Huang Kedi. "Research on Design and Application of HLA Simulation Result Database," Journal of System Simulation.Vol.18 No.2.pp.327-330, Feb, 2006.
- [3] Ju Rusheng, Qiao Haiquan, Chen Shaoqing and Huang Kedi. "A Method for Creating Database Relational Tables Based on OMT Files in HLA," Computer Simulation. Vol.23 No.5.pp.119-122, May, 2006.
- [4] Jiang Xiajun, Li Weiqing and Wu Huizhong. "The Research of Data Collection Technology in Advanced Distributed Simulation," Journal of System Simulation.Vol.16 No.8.pp.1758-1761, Aug, 2006.
- [5] Zhang Ke, Zhang Xinyu, Ju Rushen and Qiu Xiaogang. "Solution of Data Collection in HLA," Journal of System Simulation.Vol.16 No.12.pp.2725-2728, Dec, 2004.