Front. Electr. Electron. Eng. China 2007, 2(1): 45–48
DOI 10.1007/s11460-007-0008-9

RESEARCH ARTICLE

WANG Jiangyun, WANG Xingren

Research on system architecture of modeling and simulation

Abstract The modeling and simulation (M&S) architecture describes and defines the relationship between the different parts of a simulation. The simulation system architecture and simulation support platform architecture are discussed separately. The simulation support platform architecture consists of the management layer, the resource layer, the communication layer, the application layer and the infrastructure layer. The best way is to design and realize the M&S collaborative environment for simulation support platform in the resource-communication-application three-dimension space.

Keywords modeling and simulation, system architecture, simulation technology, simulation support platform

1 Modeling and simulation (M&S)

The modeling and simulation (M&S) technology has been widely used in the full lifecycle of industrial products, including requirement analysis, plan testing, design, production, training etc. At the same time, the M&S has also been used in non-engineering domains such as traffic management, economic analysis, and military training. The M&S has become the third method used to know and rebuild the world after theoretical research and experimental research. The essence of the M&S is to study the actual or imaginary objects through reflecting objects in an actual environment into a virtual environment generated by computers.

The model is the basis of simulation. It is the physical, mathematical or logical description of a system, an entity, an environment, a phenomenon and a process. Simply put, the model is one kind of computing function, and the simulation is the execution process of the model in computers [1].

Generally speaking, the development process of the simulation system includes six steps.

Step 1 Analyzing the question of the objects according to the aim and the requirement;

Step 2 Building the concept model of the objects;

Step 3 Designing the system architecture;

Step 4 Modeling the system, sub-system, and components;

Step 5 Running the model and the simulation system to get the results;

Step 6 Verifying and correcting the model and simulation system according to the simulation results.

The core of the model and the simulation system is question→system→model and their decomposition and composition.

The M&S is a series of activities. A support platform, or a support environment, is needed to complete these activities. The simulation system and the simulation support platform have different concepts. The simulation system is a self-existent system and aims at the application requirement. The simulation platform is common, resource sharing. In the simulation platform, the simulation system could be developed, built and executed.

The M&S architecture describes the relationship of the composed parts. The architecture is propitious of deploying resources, accomplishing the message interaction among systems, and fulfilling data management and time management.

The M&S architecture uses a layered, modularized and networked method to achieve geographical distribution, resource sharing, time consistency, interoperability and reusability [2].

The simulation system is a model composed of many models, and is a system composed of many systems. The scale of the simulation system and the simulation model may be large or small. However, we should adopt the layered structure while designing the simulation system: complex system→system→sub-system→component→sub-component, or multi-federation→federation→federate→entity→component→sub-component, or system→sub-system→model→module→sub-module. Generally, the federate corresponds to the simulation system. Sometimes, the federation corresponds to the simulation system.

The information interacting in the simulation system is shown in Fig. 1. Interoperability is mainly reflected in the

Translated from *System Simulation Technology*, 2006, 2(2): 63–68
[译自: 系统仿真技术]

WANG Jiangyun (✉), WANG Xingren
School of Automated Science and Electrical Engineering, Beijing University of Astronautics and Aeronautics, Beijing 100083, China
E-mail: wangjiangyun@buaa.edu.cn

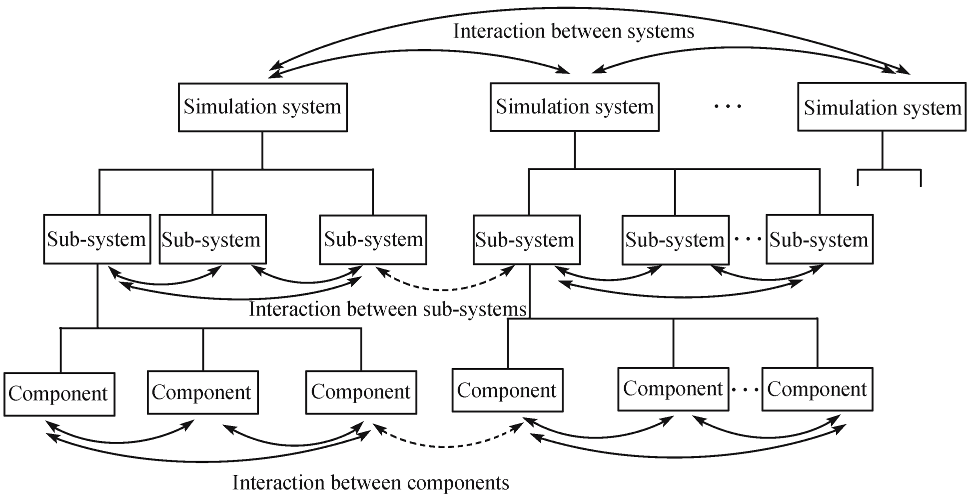

Fig. 1 Information interaction among simulation systems

data being transmitted among the components in the same layer, such as the data transmission among the simulation systems, the data transmission among the simulation sub-systems in the same simulation system, and the data transmission among the components in the same simulation sub-system. The sub-systems in the different simulation system should complete the data interacting by the upper layer. This is one of the principles when designing simulation system architecture [3].

2 The architecture of simulation system

The architecture of a simple simulation system is simple and composed of a simulation management system, sub-systems, and modules. It also includes the simulation record/display, data collection and evaluation, etc [4].

The architecture of a complex distributed simulation system is shown in Fig. 2.

The architecture uses an HLA/RTI framework and includes multiple federates (federate 1, 2,…, *n*). Federates include entities, such as planes, missiles, tanks, and ships. The SNE system describes the time-space attributes of a synthetic natural environment (terrain, atmosphere, ocean, space). The C^4I system describes the information communication, command and control. It could be real devices or a simulation system. The CGF system generates forces and the scenario editor generates a campaign plan. The database provides all kinds of data. The simulation results could be recorded, analyzed and evaluated.

The complex distributed simulation system should solve the following technologies.

2.1 Data management

In the process of simulation, every entity will generate much data dynamically, such as plane (entity) posture, location and velocity, etc. A great amount of data should be transmitted through the network, which may cause data jam and time delay. However, not every system, or sub-system, or entity, needs all the data. Their requirements will change with time.

Data management solves the mode and time of data transmission from the source (publisher) to the destination (receiver). Data transmission has many modes.

1) Broadcast: transmit all the data from publisher to all nodes in the simulation system. This mode will form a bottleneck;

2) Unicast: transmit data from one publisher to one receiver;

3) Multicast: transmit data from one publisher to multi-receiver.

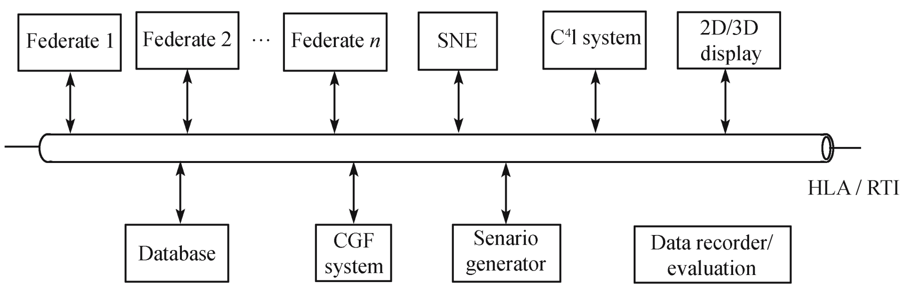

Fig. 2 Architecture of a complex distributed simulation system

Multicast is the best mode in a complex distributed simulation system. It forms some communication team according to the publisher and subscriber. It groups nodes that require the same information into one communication team. The following are the principles of grouping communication team:

1) Region distribution: according to the distance of the entities. Because the sensors have a definite measure range, and the attack has a definite range. Only those entities which are in the same range could interact with each other, so these entities should belong to the same communication team;

2) Force deploy: for example, ship doesn't engage with tank. According to the force type, we could divide them into different communication team;

3) Interesting factor: when two entities have the same interesting message, they could belong to the same communication team.

2.2 Time management

The simulation system must have time management mechanism. Time management solves the advancement mechanism of the simulation.

1) The logical sequence of events;

2) Real time;

3) Time synchronization of multi-nodes.

The execution process of the simulation computer is serial and periodic. The periodic time is called the frame time. The computing step is equal to the frame time in a mathematical simulation. When hardware is in the loop or human is in the loop, we must consider the clock or manage time advancement.

The following three phases must be finished in one frame time.

1) Collect and input data;

2) Calculate model and data;

3) Publish and output data.

Figure 3 shows the contents and sequence in one frame time.

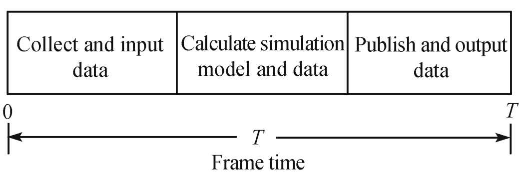

Fig. 3 Contents and sequence in one frame time

The core of time management is to computer the frame time T. There are three methods.

1) Every event and every parameter have the time stamp in the simulation process. Only these events and parameters which have the same time stamp could be disposed in one time phase;

2) The correct causality is insured. When finished with all the events and parameters, the computer begins the next frame time. Every frame time is not equal;

3) The real time is insured. We decide the frame time according to the feature of the simulation object.

3 The architecture of the simulation support platform

The simulation support platform is used to investigate, design, develop and execute the simulation systems. The simulation support platform uses a layered architecture. The structure of the simulation support platform is divided into five layers: the management layer, the application layer, the communication layer, the resource layer, and the infrastructure layer.

1) Management layer: manages the simulation support platform;

2) Application layer: designs the application systems according to the different goal and requirement;

3) Communication layer: solves the information/data interaction among multi-computer and multi-nodes, especially the large scale geographically distributed simulation system;

4) Resource layer: includes models, data, programs, tools, software, and specifications to support the development of simulation systems;

5) Infrastructure layer: includes the basic hardware and software, such as computer, network, operating system.

Figure 4 shows the architecture of the simulation support platform.

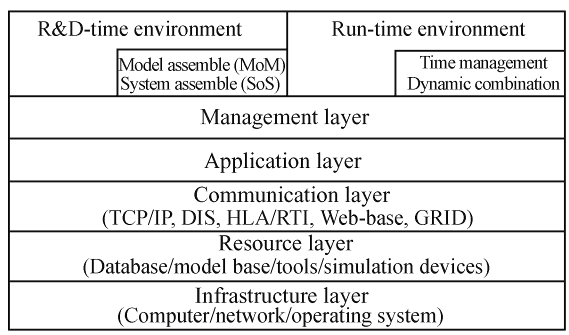

Fig. 4 The layered architecture of simulation support platform

From the view of the users, the simulation support platform includes the R&D-time environment and run-time environment. Users could analyze, model, develop, and design a simulation system in R&D-time environment and execute and manage the simulation system in run-time environment.

The resource layer, communication layer and application layer are three stanchions of the simulation support platform. In the R&D-time environment, we think of the application-resource two-dimension plane. That is, we use resources to construct the application system according to the application type, domain, destination and requirement. In run-time environment, we think of application-communication two-dimension plane. In resource-communication two-dimension plane, we think of a geographically distributed recourse to generate, store and modify.

4 Conclusions

The M&S technology is different from the science of computing and business management. It is characterized by the dynamic, interactive, distributed, and consistent features. The M&S architecture should adopt a layered, modularized and networked structure to achieve geographical distribution, resource sharing, time consistency, interoperability and reusability. In application-resource-communication three-dimensional space, we build the simulation support environment which could provide all the M&S services to study, design, develop and execute simulation systems.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant No. 60404016).

References

1. Wang Xingren. Research and discussion on technologies of modeling and simulation. Journal of System Simulation, 2004, 16(9): 1896−1897 (in Chinese)
2. Marko H. On decomposition in model development. In: Proceedings of the Simulation Interoperability Workshop, 03E-SIW-016, 2003
3. Jenny U, Farshad M, Rassul A. Collaborative modeling and simulation in a distributed environment. In: Proceedings of the Simulation Interoperability Workshop, 03E-SIW-061, 2003
4. Tae-Dong Lee, Chang-Hoon Kim, Chang-Sung Jeong. A framework for dynamic federation execution environment. In: Proceedings of the Simulation Interoperability Workshop, 03S-SIW-038. 2003