

# The Research on the Running Management Mechanism of Simulation Model Platform

Dong YE<sup>1,2</sup>

1 .The Economic Management Academy of Beijing  
Institute of Technology  
BIT,Beijing,China

2.The National Defense University  
NDU,Beijing,China  
Yd\_km@163.com

Yonggang WANG ,Junqiang XU

The National Defense University  
NDU,Beijing,China  
wang\_yong\_gang\_z@163.com

**Abstract**---The running management component is an important part of the simulation model platform and plays an important role in the whole lifecycle of the simulation system. Based on the real application background, the paper concludes several typical mechanisms of simulation running management and describes the analysis and design methods of it.

**Keyword:** model platform, running management, design

## I. INTRODUCTION

Running management is a basic function of the model platform, which ensures the synchronization and coordinates the exchange of events within multi-federations simulation. Usually, the running management takes the hybrid control mechanism based on time advancing and discrete event trigger to ensure the logic correctness and smooth running of the whole simulation. It is the management core of the model engine, model components and the interaction between them, whose main duty is to ensure the synchronization, data consistence and logic correctness of all the simulation components.

## II. THE TYPICAL SIMULATION RUNNING MANAGEMENT POLICIES

The two basic simulation running management policies are time advancing and event scheduling. The time advancing means the simulation time moves forward in one step in each time interval, which is shown as Figure 1. Within the time advancing, the parameters of model status are updated in each time interval. In general, time advancing steps must be coordinate with the other management services of the model platform so that the simulation result is causally correct and ordered. The event scheduling means the simulation is advanced by the triggered events and the model status can be updated only within the event. As showing in Figure 2, the event scheduling is faster and more effective than time advancing mechanism in the occasion of fewer simulation events happened.

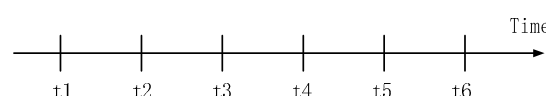


Figure 1. Continuous Time Advancing

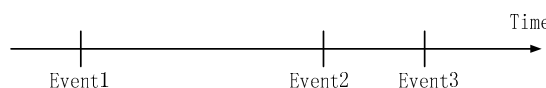


Figure 2. Discrete Event Scheduling

## III. THE ANALYSIS AND DESIGN OF RUNNING MANAGEMENT POLICIES

Considering the different characteristic movements in the hybrid simulation system, the design of the running management policy in the model platform must contain the multi-step-size management of the time advancing policy. In this policy, the actions of high-speed simulation entities (e.g. the air crafts) should be taken as the measurement basis for the simulation to calculate the basic or minimum time advancing steps. Meanwhile, the low-speed simulation entities (e.g. vehicles or personnel) take the bigger time steps to advance the simulation periodically, in which, the entities' basic behaviors can be simulated in the cumulated time intervals, and the two kinds of entities will be re-synchronized after several big time steps, which is shown in Figure 3.

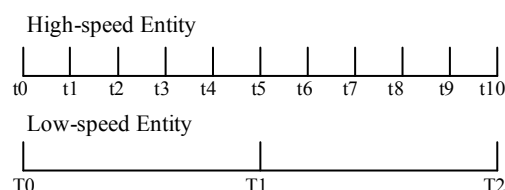


Figure3. Two kinds of time advance steps of different entities

The above-mentioned time advance policy is usually used in the training simulation program. While in the construction simulation system, the event scheduling policy should be taken to achieve the analysis conclusion in short period. In this situation, each simulation federate should provide its own time look ahead and the minimum one will be calculated as the next time stamp which the federation can

advance to. As shown in Figure 4, each axis represents the simulation time of each federate and the shadow part represents the next look ahead. According to the figure, the minimum look ahead (17) of federate 2 will be chosen and the federation can advance to time stamp (17) conservatively without any disorders.

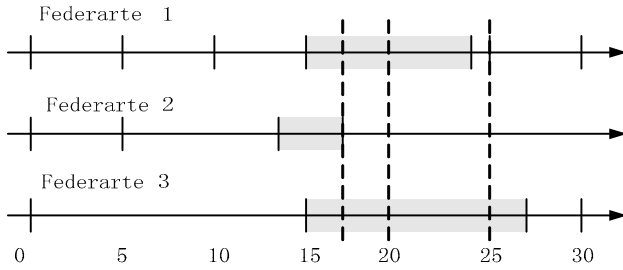


Figure4. The Conservative Time Advance Policy

#### A. The Analysis of Simulation Running Management Process

The simulation running management consists two correlated parts, which are manual management and runtime management services. The manual management means that the director and operator can input the system parameters and correct the simulation clock through the model management console to monitor the whole simulation system running status. Meanwhile, the runtime management services utilize the supporting functional services provided by the model platform, which can automatically respond to the user input and manage the running status of simulation system.

The two system management methods, which are mentioned above, can be used together to meet the needs of different kinds of applications, including single-platform pattern and distributed pattern. When all the model components are deployed and running on a single platform, which is known as single-platform pattern, the components' running management and scheduling are in the care of one specified component of platform which will schedule the running according to the scheduling table of federation. In another case, the components are deployed on the distributed hosts, which are following a specified pattern in the network. One of the components will be appointed as the main console and schedule all the other federates' running in the federation basing on the system's scheduling table in the shared database. Meanwhile, the running scheduling in each host is in the charge of scheduling component, which is under the control of main console and accords to the scheduling messages and specified scheduling table of local host. These two types of scheduling methods make up of the dual-level scheduling mechanism.

#### B. The Design of Simulation Running Management Process

When the simulation clock is moving forward, the model engine schedules all the components' running and monitors the running status of them. No matter whether all the components can finish the calculation in one time step and reach the synchronization status or some of them can't

accomplish in time, the platform's management component will send feedback to the main console to facilitate monitoring. There are two types of synchronization functions in the running management. One is the synchronization between the platform and the main console, and the other is the synchronization within all the components in the platform. The running management process of the platform is showing in the following figure 5:

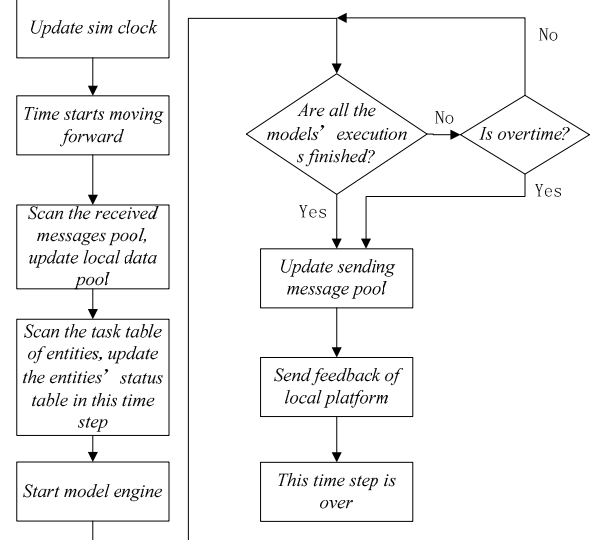


Figure5. The Flow Chart of Running Management of Platform

The whole process of the running management of platform can be divided into following several steps:

Step 1: Synchronize data. When the simulation clock is updated, the platform starts moving forward in one time step. It can receive the instruction commands, event messages and entities' data which are received by the received data pool in the beginning of step, and also can update the local data pool.

Step 2: Update the model status table in this time step. According to the task table, event message table and the task type and model mapping table, the platform will figure out the models which should execute in this step and insert the model info into the model status table.

Step 3: Start the model engine and schedule the model running. The model engine and model will insert the model running status into the model status table. When all the models' execution is finished in this step or the synchronization is overtime, the platform will end the model running in this time step.

Step 4: The running management program will update the sending data pool with the entities info and events info which are updated by local model in this time step. This facilitates the data exchanging through the synthesized data link to the other platforms which are in need of these data.

Step 5: The feedback messages of step forward is accomplished or is overtime are sent to the central controlling server for monitoring or handling.

### C. The Management and Scheduling of Models

The scheduling of models will be executed in the following steps:

- The first step is to ensure that the data exchange interface of platform is valid. Including the task data of execution in this time step, the model component data and etc.
- The second step is to initialize the local model running status table in the model engine with the received data. The initialization and monitoring of model running are also accomplished in this step.
- The third step is to create the monitoring thread to monitor the model running status in real time.
- The forth step is to invoke the correlative model components and execute the simulation, which is guided of the model running status table.
- The fifth step is to wait for the returning message of the monitoring thread. The main thread will return after the finishing of all the models' execution, no matter they quit the execution in normal or informal status. Then, the simulation in this time step is over after the finish of model engine and the main thread's execution.

The monitoring thread is in charge of the monitoring of all the models' running status. It polls the model running status table periodically and records the execution time of each simulation thread. When the overtime status of thread is found, the monitor thread will shut down the thread by force to ensure the whole simulation system can run without any interruption. When all the models' execution is finished, the monitor thread will send a message to inform the main thread and quit.

The simulation thread is in charge of the loading and initialization of the models. First of all, the simulation thread receives the data from the main thread to find out the exact tasks to be executed and the model component to be loaded. Secondly, the simulation thread will judge the type of model components. If they are composite components, the simulation thread will create the component object directly, achieve the data exchange interface of the platform and invoke the interface to run the simulation.

### D. The Monitoring of Models' Running Status

In the process of the platform's execution, many model components are loaded to accomplish the battle simulation task. So, it is very necessary to find out an effective way to monitor and manage these components. To achieve these goals, the model platform sets up the model running status info table in the data pool and provide the graphical user interface for the user to monitor and control the model components in an intuitionistic and convenient way.

The monitoring of model running status is an important guarantee for a robust model platform, and is also the basis of synchronization and controlling of model components in it. The core of the running fundamental infrastructure is the model controlling logical network, which consists of five main parts, including model, running management, model engine, monitoring and running status table. The relationship within these parts is shown as following figure 6.

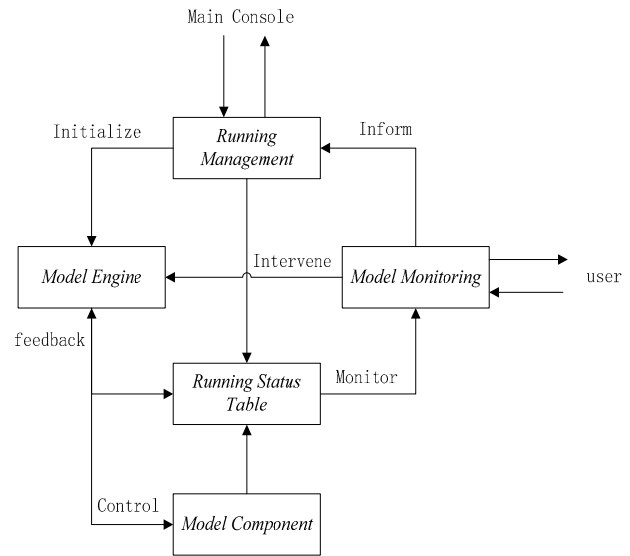


Figure6. The Logical Network of Models' Running Status Monitoring and Management

When the model platform receives the control message from the main console or the intervening command from the GUI, the running management program will scan the task list and find out all the tasks should be executed in this time step. Then, it will scan the mapping table of task type and model, insert the models and entities' info into the model running status table and set the initial status of all the models to "waiting for execution". After that, the running management program will start the model engine and the model running status monitoring program. According to the model status table, the model engine will start the threads separately for each model. The status of successfully started model will be set to "running", while the status of "starts error" will be set to the unsuccessful counterpart. In the process of the models' running, if the model can accomplish its execution, the correlative status in the model running status table will be set to "execution accomplished". In the case of failure or delay within the time step, the running management program will set the status to "execution delayed". The user can guide the model's execution through the GUI and the model monitoring program will send the user's intervening messages to the model engine, which will take care of the correlative response. After the finish of all the models' execution in this time step, the model monitoring program will inform the running management program about the execution accomplishment. Otherwise, the execution result will be sent to the main thread by the running management program after the finish of this time step.

## IV. CONCLUSION

Running management is a core function of simulation model platform, which ensures the running, scheduling and management of simulation. The running management part takes care of the simulation's initialization, synchronization, command translation, data organization and the driving of model engine. A set of effective running management

policies are the basis and insurance of the simulation system's success.

#### REFERENCES

- [1] Upton, D. M., The management of manufacturing flexibility. California Management Review, pp. 72- 89, 1994.
- [2] Weiping Wang, Qun Li, Yifan Zhu, Feng Yang, The Theory and Application of Flexible Simulation [M]. the Publishing Company of University of National Defense and Technology, Chang Sha, 2003.
- [3] Fuqing Yang, Hong Mei, Keqin Li, Software Reuse and Software Component Technology [J], Acta Electronica Sinica, 27(2) : pp. 68-75, 1999.
- [4] Gang Xiao, Jun Xu, Yuanming Zhang, Fei Gao, Huiqiang Lv, Research on Software Component-based Domain-specific Framework on Software Configuration Model Basis [J], Computer Application and Software, 25(5) : pp. 90-92, 2008
- [5] Xiaofeng Hu, Guangya Si, Lin Wu, Guochun Zhang, the Theory of War Gaming(Vol.1&2) [M], the Publishing Company of National Defense University, 2004
- [6] OMG. MDA Guide Version 1.0.1. June, 2003.
- [7] Yaping Ma, Xiuluo Liu, Yuan Li, the Battle Simulation System (Vol.1&2) [M], the Publishing Company of National Defense University, 2005
- [8] Lihua Ma, the Research of Component Developing Method and Application Based on MDA [D], the master degree thesis of Chang Chun University of Science and Technology, 2007.