CHEMICAL ENGINEERING TRANSACTIONS
VOL. 33, 2013
A publication of AIDIC
The Italian Association of Chemical Engineering
Online at: www.aidic.it/cet
Guest Editors: Enrico Zio, Piero Baraldi

ISBN 978-88-95608-24-2; **ISSN** 1974-9791
DOI: 10.3303/CET1333102

Process Interaction Simulation Research of Adapting to Corrective Maintenance

Bocheng Gao*, Linhan Guo, Lin Ma, Naichao Wang

School of Reliability and System Engineering, BUAA, Beijing, China
gbc0311@163.com

Corrective maintenance process simulation is one of the most important methods for equipment supportability analysis, and corrective maintenance process simulation has a positive effect on analyzing and evaluating the unavailable time which is induced by the maintenance. Discrete event simulation is one of the most effective tools to analysis corrective maintenance process nowadays. Process Interaction method (PIM) whose modelling idea is the same as human's thinking way of natural is one of the discrete event simulation methods, and PIM is easy to be realized in the current Object-Oriented development environment, so PIM is chosen as the main algorithm used to achieve corrective maintenance process simulation. In this paper, detailed design of corrective maintenance process simulation software is achieved by using PIM. All the process class libraries and the relevant entity class libraries indispensable in simulation model are defined, and at the same time these process class libraries and entity class libraries are developed. The staple principle based on PIM about how to impel the simulation clock is illustrated. The time distribution class which is used to calculate the time of random events is achieved. At last, a simulation case is given to verify the correctness of the simulation algorithm achieved in this paper. The methodology proposed in this paper provides a new way for the implementation of the corrective maintenance process simulation, and this methodology can be more easily extended to support the concurrently maintenance activities simulation.

1. Introduction

Corrective maintenance process is composed of all activities which can make products restore to the required state after some failures happen. Those activities include some steps, for example, failure detection, isolation, decomposition, replacement, reassembling, testing, etc. By corrective maintenance process simulation, it can analyze and evaluate the unavailable time induced by the maintenance of the equipment system. At present, many scholars have studied corrective maintenance process simulation modelling, and some of them use the Monte Carlo simulation method to simulate the equipment corrective maintenance activities of the base operational units, just as Guo Linhan did in 2007. But they only give the simulation method from logic angle, they do not give a specific algorithm design from angle of software development.

Corrective maintenance process is a discrete event. Event Scheduling, Activity Scanning, Three-Phase Approach and Process Interaction method (PIM) are all the more commonly used discrete event simulation algorithms. Because of the controlling logic of Three-Phase Approach is clear and easy, some scholars use Three-Phase Approach to the discrete event simulation, just as Michael and Ricardo A. Cassel did in 1995 and 2001. But with the occurrence of PIM whose modelling idea is the same as human's thinking way of natural, and owing to the achievement of Process Interaction is easy in the current development of object-oriented environment, PIM is used more and more frequently. According to the feature of corrective maintenance, PIM can be chosen to do the corrective maintenance process simulation. Some scholars realize Process Interaction algorithm, such as Jos M.Garrido, but he does not specify how Process Interaction is applied to professional areas of technology. And it is the main technical difficulty to design discrete event simulation algorithm with specific technical background while developing the simulation

Please cite this article as: Gao B., Guo L., Ma L., Wang N., 2013, Process interaction simulation research of adapting to corrective maintenance, Chemical Engineering Transactions, 33, 607-612 DOI: 10.3303/CET1333102

platform, moreover, if this algorithm is achieved, it would be the basis for developing a more complex maintenance simulation system which is efficient and has a good real-time performance.
In this article, according to the characteristics of corrective maintenance process, detailed design of corrective maintenance process simulation software is developed by using PIM. All the process classes and the relevant entity classes which are indispensable in corrective maintenance process simulation model are defined, and these process classes and entity classes are developed. The basic rule based on PIM about how to impel the simulation clock is illustrated. The time distribution class which is used to calculate the time of random events is designed.

2. Corrective Maintenance Process Interaction Simulation Model

2.1 The definition of Process in the model

Process is composed by some events associated with certain type of activities. A process describes the logical and the temporal relationship among events and the activities included in the process. Event is instantaneous behaviour which changes systems' status; Activity is the procedure performed by the entity. The status of process can be *work*, *wait*, *ready* and *over*. When the process is in its *work* status, it can perform some actions, such as: updating a data, or signalling to another process, etc. The process in *wait* status cannot come to work until it received a wake-up signal from other process. When a process is in its *ready* status, it can work after a period of time. The process in *over* status cannot convert to other status.

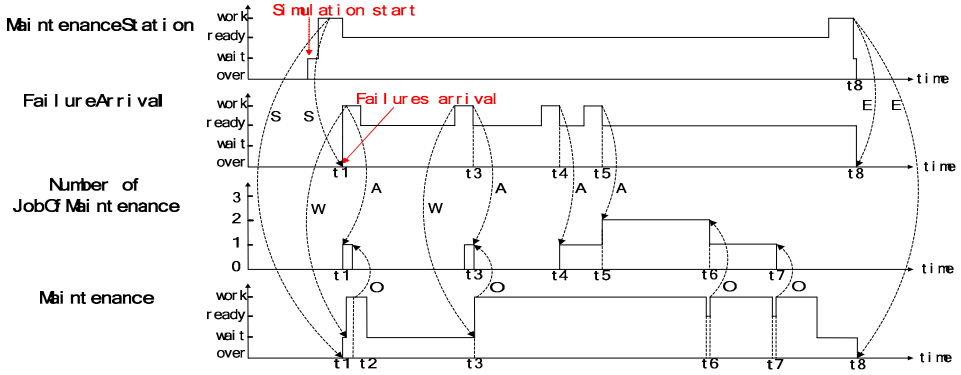

Figure 1.Corrective maintenance in a station schematic

In Figure 1, MaintenanceStation issues the starting signal and the ending signal. FailureArrival adds failure parts. Maintenance is the process of maintaining. JobOfMaintenance is the failure part.
The semaphores in figure 1: S–Process starts working. E–Process stops working. A–FailureArrival adds a failure part. O–Maintenance removes the failure part and maintains it. W–Wakes up Maintenance. As shown in figure 1, FailureArrival and Maintenance are different processes in the corrective maintenance process, and they must be defined as different processes. When more than one corrective maintenance happens, there must be a process to manage the entire process just like MaintenanceStation, so a specific process must be designed to manage the whole simulation process.

2.2 The definition of Entity in the model

Entity represents each of the identified objects in the system. As shown in figure 1, the failure part needs to be dealt by Maintenance and FailureArrival. The quantity of failure parts has a direct impact on whether Maintenance starts working, and the failure part is the main object which FailureArrival works for. So failure part must be treated as an entity class whose name is JobOfMaintenance to achieve.

2.3 Simulation parameters calculation

The i-th maintenance time is ΔT_i whose formula is as follow:

$$\Delta T_i = T_E(i) - T_A(i) \tag{1}$$

$T_A(i)$ is arrival time of the i-th maintenance, and $T_E(i)$ is ending time of the i-th maintenance. The formula of $T_A(i)$ is as follow:

$$T_A(i) = F_A^{-1}(X) \tag{2}$$

$F_A(X)$ is cumulative distribution function of the failure time, and its expression is $F_A(X)=\int_0^t f_A(T)dT$, in this expression, T is the arrival time of failure parts, and the density function of failure time is $f_A(T)$. $f_A(T)$ can be any density function. Take exponential distribution for example, $f_A(T)$ is exponential distribution, so formula of $T_A(i)$ is $T_A(i)=F_A^{-1}(X)=-\frac{1}{\lambda}\ln(X)$. λ is the arrival rate, and X is a random value in the range (0,1).

The formula of $T_E(i)$ is $T_E(i)=T_S(i)+T_M(i)$. $T_S(i)$ represents the starting time of the i-th maintenance, and $T_M(i)$ represents the lasting time of the i-th maintenance.

1). The formula to get $T_S(i)$ in simulation is as follow:

$$T_S(i)=\begin{cases}T_E(i-1), & T_E(i-1)>T_C\\ T_E(i-1) or T_C, & T_E(i-1)=T_C\\ T_C, & T_E(i-1)<T_C\end{cases} \tag{3}$$

T_C is the current time of the simulation clock, and it is a value changing with the simulation running, so the value of $T_S(i)$ will appear the following circumstances:

If $T_E(i-1)>T_C$, the (i-1)-th maintenance cannot finish before T_C, so Maintenance is in its *working* status, and the i-th maintenance cannot start working until the (i-1)-th maintenance finishes its work. In this condition, $T_S(i)=T_E(i-1)$.

If $T_E(i-1)=T_C$, the (i-1)-th maintenance just finishes its work at T_C. So $T_S(i)=T_C or T_E(i-1)$.

If $T_E(i-1)<T_C$, the (i-1)-th maintenance is in its *waiting* status. It can start working if there is a failure part arrivals, so the i-th maintenance starts working at T_C, the starting time of maintenance $T_S(i)=T_C$.

2). The formula of $T_M(i)$ which is used to represent the lasting time of maintenance is similar to the formula of $T_A(i)$, and the formula is as follow:

$$T_M(i)=F_M^{-1}(Y) \tag{4}$$

$F_M(Y)$ is cumulative distribution function of maintenance time, and Y is a random value in the range (0,1).

3). According to the calculation above, ΔT_i (i=1… N) which can represent the maintenance time of per simulation, so the average maintenance time ΔT_x can be described in the follow formula:

$$\Delta T_x=\sum_{i=1}^{N}\Delta T_i \Big/ N \tag{5}$$

3. Corrective maintenance process algorithm design based on Process Interaction

3.1 The work process of simulation

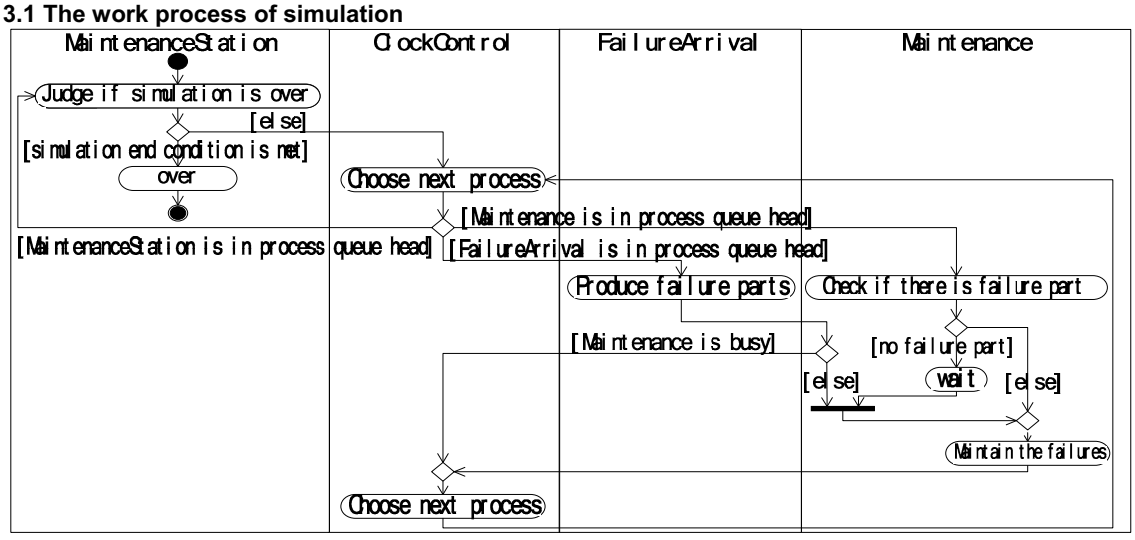

Figure 2. Activity diagram of corrective maintenance simulation process

Figure 2 is corrective maintenance process simulation activity diagram. This figure shows the whole program to implement the simulation of corrective maintenance process. The relationship between the various activities and the overall work processes of the simulation are shown in Figure 2.

3.2 Entity class –JobOfMaintenance design

Entity class whose name is JobOfMaintenance represents failure parts. JobOfMaintenance is a very important class in corrective maintenance process simulation algorithm. Maintenance and FailureArrival do a series of interaction around this class. The properties of this class are altered by Maintenance and FailureArrival, and the properties are used to calculate simulation data. Such as ResponseTime and ArrivalTime, and so on.

3.3 Time distribution class design

In this class, some methods which are used to calculate the inverse function of commonly used random distribution function are built, and the returned values of these methods represent the occurrence time of random events. Such as, ExponentialStream(EVariable) is the method whose occurrence time distribution of random events is exponentially distributed, and EVariable is the efficiency value of the work; ServiceTime(StartNO) is the method whose returned value which is got randomly is the service time, and StartNO is the maximum value of the service time.

3.4 Clock controlling class - ClockControl design

ClockControl class is a core class in the entire simulation. The process classes impel simulation clock by using ClockControl. All the processes are chronologically according to their wakeuptime arranged in a process list whose name is ReadyList. ClockControl uses ReadyList to determine which process is next.
In this class, the key method *Clock()* use the local variables to get current process, and it compares current process with the first process in ReadyList. If they are the same process, current process continues running. If not, current process stops running and its status is changed to *ready*.

3.5 Process class design

3.5.1. The base class of process – SeriesActivate

SeriesActivate class is the base class of all the process classes. It contains all the properties describing the feature of the process and some public methods. For example, wakeuptime used to sign the time at which the process starts working; Passivity signing that the process is in *waiting* status; Activate() making the process start working; Run() function in which the process executes its work.

3.5.2. Maintenance class

The primary objective of Maintenance is to simulate the maintenance activity in corrective maintenance process. Based on this objective, the Run() function of Maintenance is designed as Figure 3.

```
public override void Run()  // the activities of the process class run in Run() function
{double ActiveStart, ActiveEnd;  // starting time and ending time of maintaining failure parts
for (;;){
 working = true;  //property of Maintenance, when it is true, Maintenance is working
 while(!MaintenanceStation.JobOfMaintenanceQueue.IsEmpty())//if failure part queue is not empty, start maintaining
     Maintaining(time) //record the starting time of per maintenance,calculate the ending time of per maintenance,impel simulation clock
 working = false; // property of Maintenance, when it is true, Maintenance is waiting
 Cancel();//stop current process, impel simulation clock}
}
```

Figure 3. The Run() function of Maintenance process

3.5.3. FailureArrival process class

FailureArrival simulates the process by which the failure parts are carried. In figure 4 the flow diagram of FailureArrival Run() function is shown, and the dashed arrow sign that the time of activity is discontinuous.

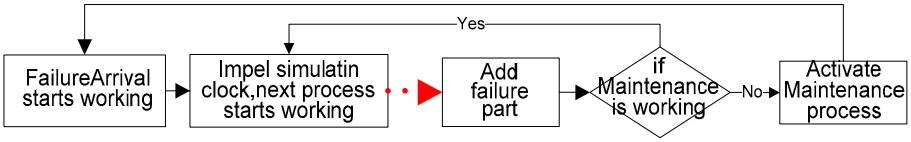

Figure 4. The flow diagram of FailureArrival Run()

3.5.4. MaintenanceStation process class

MaintenanceStation controls the whole process of simulation, and it sends the starting and the ending signal of simulation. Firstly, MaintenanceStation sets the time to check if the simulation should be over, if it is the time to end the simulation, it ends the simulation. If not, ClockControl class is used to impel the simulation clock, and corresponding process is activated, then the simulation can run until the objective of simulation is achieved. In this way, the whole process of simulation is controlled by MaintenanceStation.

4. Simulation Case

4.1 Model of the simulation case and the input data

In reality, because of the limitation of the models which can be analyzed by using the queuing theory, so it cannot get the result by calculating for some complex models. But according to the actual requirement, the time distribution in the algorithm is uncertainty, so the algorithm for simulation models should be cable of simulating any distribution.

To verify this simulation algorithm has high accuracy, the following case whose result could be got by using the queuing theory is proposed.

The case model describes the maintenance process of pumps used in a production line in a chemical factory. The fault time of pumps is exponentially distributed, and the failure rate of the pumps is λ_{FA}. There is one repairman in the chemical factory, and the time this repairman needs to maintain the pumps follows exponential distribution with the parameter μ_M.

After analysis, the model is M/M/1 queuing model, the formula to calculate the mean maintenance time of this model is as follow. In the formula, W is the mean maintenance time of the pumps.

$$W = \frac{1/\mu}{1-\lambda/\mu} \tag{6}$$

The input parameter table of the simulation algorithm is shown as table 1. In table 1 N_{over} is the least number of fault pumps which have been maintained.

Table 1.Input parameters form of corrective maintenance process simulation

Simulation Data	Input parameter		
Simulation No.	μ_M (hour)	λ_{FA} (hour)	N_{over}
1	6	15	4000
2	5	20	5000
3	8	24	7000
4	10	20	10000
5	15	20	100000

4.2 Comparison and analysis of output data and the validate data

Comparing the result of simulation algorithm with the queuing theory analytical calculation result, table 2 is got. ΔT_x is the mean maintenance time of the simulation algorithm, and N is the maintenance times in this table. W is the mean maintenance time calculated by the queuing theory.

Table 2.Output data and validate the data form of corrective maintenance process simulation

Simulation Data	Output parameter		Queuing theory analytical data
Simulation No.	N	ΔT_x (hour)	W (hour)
1	4000	**9.85**	**10.00**
2	5000	**6.55**	**6.66**
3	7000	**11.80**	**12.00**
4	10000	**19.85**	**20**
5	100000	**60.30**	**60.00**

Comparing the result of simulation algorithm with the queuing theory analytical calculation result, Figure 5 is got. The comparison between ΔT_x and W is shown in Figure 5. The percentage of errors of corresponding data can also be seen in this figure. In Figure 5, it is manifest to see that there is little difference between the simulation results and queuing theory results. And by gradually increasing the

times of simulation, the errors between the output data and the calculated values become smaller and smaller, so the correctness of the algorithm is guaranteed and credible.

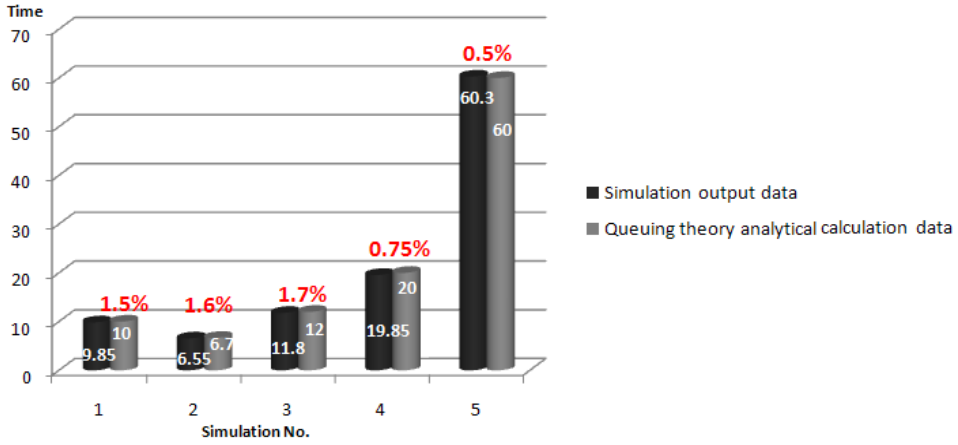

Figure 5. Comparison diagram between Simulation data and Calculation data

5. Conclusion

This paper introduces that how to complete corrective maintenance process simulation algorithm based on Process Interaction, and it designs the function of the main classes, the properties and the methods included in the classes in detail. In this paper, relevant processes and entities which are in corrective maintenance process simulation model are defined, and the functions of the processes and entities are analyzed. The Process classes and the Entity class needed in corrective maintenance process simulation are developed. Basic principle based on Process Interaction about how to impel the simulation clock is illustrated. The time parameters involved in corrective maintenance process are calculated. The whole algorithm provides a theoretical and application support to develop a more complex corrective maintenance simulation system which is efficient and has a good real-time performance in the future.

In this article, a simulation algorithm based on Process Interaction is achieved to solve a queuing model, and the correctness of its results is validated by the queuing theory. What is more, this algorithm could solve some more complex problems which could not be solved by queuing theory. In reality, some actual situations have the characteristic of Queuing network, and this model could not be calculated by using the queuing theory formula, however the simulation algorithm above could be modified to solve those problems which are more complex.

Acknowledgement

This research is supported by the National Natural Science Foundation of China (Grant No.70901004, N0. 71171008, and No.61104132).

References

Cassel Ricardo A, Michael, 2001, Distributed discrete event simulation using the three-phase approach and Java, Simulation Practice and Theory, 8, 491-507.

Claudio Gariazzo, Armando Pelliccioni, Paolo Bragatto, 2012, Simulation of Accidental Release by Means of two Different Modeling Approaches, Chemical Engineering Transactions, 26, 555-560.

Guo.Q, Liu.J.G, Chen.X.W, 2006, Optimization Model and Simulation of the Queuing System with Quick pass, Proceedings of the 6th World Congress on Intelligent Control and Automation.

Guo. L. H, Kang R, 2007, Base operational unit corrective maintenance process modelling and simulation, Journal of Beijing University of Aeronautics and Astronautics, 33, 27-31.

Jos M. Garrido, 2001, Object-oriented discrete-event simulation with Java : a practical introduction, Kluwer Academic / Plenum Publishers.

Pidd Michael, 1995, Object-Orientation, Discrete Simulation and the Three-Phase Approach, Journal of the Operational Research Society, 46, 362-374.

Sheldon M, Ross, 2011, Introduction to Probability Models, Beijing Posts& Telecom Press, 371-424.