A Quantitative FMEA Method based on Model Dynamic Simulation

Hanqing Zhou, Zhihao Wang, Haixiong Lv
Quality Engineering Technology Center, China Aeronautics Polytechnology Establishment, Beijing, China
qdzhouhanqing@126.com

Abstract. This paper presents a FMEA method based on model dynamic simulation. Firstly, the functional block diagram of the system is built according to the characteristics of the airborne system, failure rate is added to the traditional functional module and the modeling of the inner behavior of the functional module is described in detail. Secondly, the dynamic running process of the system is simulated through dynamic simulation of discrete events to find out the failure states of the system and the probability of their occurrence. Finally, FMEA can be made and form can be filled base on failure state and probability of occurrence obtained from dynamic simulation. This paper solves the problems of the traditional FMEA method which can't analyze the temporal logic analysis and the failure analysis of combined functions.

1. Introduction
For traditional FMEA, the consideration is seldom given to the dynamic time constraint relations among components. In this case, it is easy to omit certain failure states of the system or the influences of the system failure could be judged by mistake. The highly complex systems of the aerospace are mostly the real-time ones, in which the correctness of the system not only relies on its logic characteristic but also relies on its temporal characteristics [1]. Air disasters happened in recent years are more the failures caused by the fail to satisfy the temporal logic when all components interact with each other. To solve such problems, the model checking based on formal verification was initially introduced into the reliability assessment field of complicated airborne system abroad as of 2001[2][3][4]. The method of state space searching was adopted to detect whether the foregoing system model satisfied the specific nature and give out counter example when the system didn’t satisfy the nature [5][6]. However, there are following problems about the promotion of complete formal methods in China. Firstly, modeling is complicated and difficult through the method based formal verification during model checking, but it only could be applied into part of the system. Complete formal methods cannot be promoted nationwide easily. Secondly, though the model checking method ensures the model is correct, it cannot manifest the probability of failure of hardware added to the model and that of actual use.
Condition of the real system cannot be given. To solve the above-mentioned problem, it is very necessary to study FMEA through which the failure resulted by the temporal logic error during interaction of multiple components can be analyzed and complexity of modeling can also be reduced. Fig. 1 shows overview of the quantitative FMEA method.

2. Simulation Model

Build functional block diagram of the system according to the assigned layer of the system, represent one function with one block in the functional block diagram. The function can contain sub-function. According to the interdependence and hierarchical relations among the functions, the blocks can be connected with the wire without arrow to represent the physical relations among each function. The description format of the function is: “function name, function content definition and end function”.

Modeling of the inner behavior can be completed according to each block in the functional block diagram of the system to describe the inner behavior of the function represented by the block. It is necessary to build dynamic deduction and simulation model containing sate and event for accidents to describe the state, event, initial state and state transition inside system and describe them with ten-member group <C,I,S,O,R,F,E,T,L,ER> . The module containing sub-function doesn’t contain descriptors of state and related state transition, including S, R, F and T, but contains the sub-module descriptor C and descriptor ER that is used for describing the event association between parent module and sub-module. However, as for the module not containing sub-function, it doesn’t contain C and ER. The specific description methods are as below:

Step 1. As for the module containing sub-function, describe the sub-function of the function. For example, in function func1 C E1:subfunc1; E2:subfunc1; E3:subfunc1; end function, func1 contains E1, E2 and E3 of three subfunc1 types.

Step 2. Describe the input variable of the function. In ten-member group, I represents input variable. Each function module contains one or many input variables, for which the specific format is “variable name: variable type: in”. By taking input1: float: in for example, if the variable has fixed initial value, the variable type can be replaced with variable value, like input2: [0,1]: in.

Step 3. Describe the state variable of the function. S in the ten-member group represents state variable, indicating certain condition or state in life cycle of the function. State indicates the result of a series of
activities executed by the function. Each function module contains one or many state variables. When the corresponding condition is satisfied, the state can be transformed. Then, the format of the state variable is “variable name: [variable value]: s”, like state1: [working, failed, repair]: s.

Step 4. Describe the initial state of the function. R in the ten-member group represents initial state. There is only one initial state in each functional block diagram. The initial state R belongs to one kind of S, R ∈ S, indicating the state of each function when the simulation begins. The format of the initial state is “state name: =state value”, like state1: =working.

Step 5. Describe the failure state of the function. F in the ten-member group represents failure state. There is one or many failure states in each functional block diagram. Failure state S belongs to one kind of S. As for F ∈ S, it is necessary to record the time when entering the state and compute probability of failure occurrence if reaching the state during the process of simulation. Format of failure state is “state name: =state value”, like state1: =failed.

Step 6. Describe the output variable of the function. O in the ten-member group represents output variable. Each functional module contains one or many output variables. The format of the output variable is: “variable name: variable type: out”, like output1: float: out.

Step 7. Describe the event of the function. E in the ten-member group represents event. Event can trigger state transition, which should satisfy the corresponding transition condition. When the condition is satisfied, the event trigger can drive state transition. The event in the simulation model is mainly divided into two categories, including time delay and transient event, in which the time delay is also divided into two categories namely random event (probability distribution function with parameter) and fixed time delay event while transient event is also divided into two categories immediate occurrence event and precondition event. The format of random event in delay event is “event name (delay: =probability distribution function (probability of failure lambda))”, like failure1 (delay: =exponential(1E-10)). The format of the fixed time delay event in delay event is: “event name (delay: = Dirac (delay time))”, like failure2(delay: =Dirac (2)). It is just necessary to designate event name for the transient event. The condition of precondition event is described when implementing state switching, so the format of the transient event is “event name”, like failure3.

Step 8. Describe state transition of the function. T in the ten-member group represents state transition, the relationship between two state variables, indicating the objects executes certain action in the source state and enters target state when certain specific event occurs or certain specific condition is satisfied. The format of state transition is “state name: source state value [transition condition is true] - event name > state name: =target state”, in which the transition condition is optional. When the transition condition is empty, the event occurrence will drive state transition. By taking (state1=working) -> failure -> state1: = failed for example, when failure event occurs, the state of state1 will be switched into failed from working.

Step 9. Describe the transfer function of the function. L in the ten-member group represents transfer function. Transfer function describes three kinds of logic relations, respectively:

- The logic relationship between inner output and input of the function and state variables, the left of transfer function equation is output variable while its right is the logic expression of input and state variables, like output1:= if (state1==working) then min(input1,capacity) else 0, as shown in Fig. 2 (a).
- Describe the logic relationship among functional modules at the same layer. Firstly, find out the input and output variables of two functional modules connecting the two ends, build logic relationship between output and input variables, like Func2.input1:= Func1.output, as shown in Fig. 2 (b).
- Describe the logic relationship between input and output variables of function and sub-function in the system, as shown in Fig. 2 (c).
Step 10. Build association among events in the function and sub-function. The association among events is divided into three types: synchronic association, broadcasting association and common reason association.

1) Build synchronic association among events in parent module and sub-module contained by it. Synchronic association indicates that the event should occur at the same time and the state driven by the event should also be transferred at the same time. The representation form is \( ER\{event1&event2……eventn\} \). For example, \( ER\{e1&e2\} \) indicates \( e1 \) and \( e2 \) occur at the same time. The state transition driven by \( e1 \) and \( e2 \) is executed at the same time.

2) Build broadcasting association among events in parent model and sub-module contained by it. Broadcasting association indicates at the occurrence of certain event, other events having broadcasting association with it are notified. If the transition condition of the state driven by these events, it will be executed, or not. The representation form is \( ER\{event1|event2……Event\} \).

3) Build common reason associated events in parent module and sub-module contained by it. Common reason event contains broadcasting association. Besides, multiple events with common association occur independently alike common ones. Common reason events are usually used for multiple sub-modules in parent module. The failure may be caused by their inner error and certain common reason. The representation form is \( ER\{event1, event2……Event\} \).

3. Dynamic Simulation
A discrete event dynamic simulation method is used to simulate the model and record the probability of failure. Develop a computer program to realize below the simulation algorithm. The result of failure probability is to run multiple simulations by computer program, and the simulation results are obtained on average. Failure probability is an approximation, along with the increase of the number of simulation approximate value will gradually close to the real value. The execution process of single simulation algorithm is as follows:

Step 1. Initialization.
- Set start time \( t_0 \) and end time \( t_f \) of the simulation;
- Initialize the state in each functional block diagram according to the designated value of \( R \) in ten-member of simulation model;

Step 2. Update event table, find out the event (\( E \) in the ten-member group of simulation model) satisfying the trigger condition in the system, the transition condition of state transition (\( T \) in ten-member group of simulation model) is empty or true. It is necessary to mark the events satisfying the trigger condition with NewEventList, and then take following steps:

1) Delete the event not existing in NewEventList from the event table;
2) Add the event that not existing in event list in NewEventList to the event table and mark as the new one.

Step 3. Set simulation time \( TIME=t_0 \).

Step 4. If \( TIME\geq t_f \) or event table is empty, transfer to step 9, otherwise execute step 5.

Step 5. Calculate time for occurrence of new event in the event list, the time for occurrence of event is equal to the sum of the system’s current time and event delay time. The specific calculating way: as for the transient event, when the transition condition is satisfied immediately and the delay event is 0, the occurrence time is the system’s current time. As for the fixed time delay event in time delay event, the occurrence time is the sum of system’s current time and delay time. As for the random event in time delay
event, it is necessary to sample the probability of occurrence. When sampling, hypothesize the probability of occurrence of the event obeys the even distribution of \([0, 1]\); obtain the specific numerical value \(p \in [0,1]\) of \(p\) through random sampling, and then solve the delay time of the event according to different distributions. By taking exponential distribution for example, the distribution function of exponential distribution is \(p = 1 - e^{-\lambda t}\). After \(p\) is obtained, the numerical value of \(t\) can be inferred. The specific formula is 
\[
\lambda \left( -\log(1 - p) \right)^\frac{1}{\beta},
\]
inferred like this for other distributions. The execution time of the event can be obtained by adding the system’s current time and \(t\).

Step 6. Arrange the events in event list according to the occurrence time, by extracting the events with the earliest occurrence time from the event list, the system’s simulation time will be deducted to the occurrence time of the event, setting \(\text{TIME} = \text{tEvent}\).

Step 7. Event occurrence and drive state transition, judge whether the target state is the failure state; if it is failure state, calculate the probability of the occurrence of failure according to the system’s current time. The probability of occurrence is \(1/\text{TIME}\), and record the result in the simulation result. The result can be specifically saved with two-dimensional vector. The dimension is event name and current serial number of simulation times.

Step 8. Calculate change in input and output of the system triggered by the state transition and change in true or false values of conditions for state transition, return and execute step 2.

Step 9. Simulation end.

4. Quantitative FMEA

Implement failure mode and influence analysis on failure state and probability of occurrence obtained based on dynamic deduction and simulation of events, fill in FMEA analysis form. The detailed process is as bellow:

1) Implement failure recognition of inner failure state of the function discovered during process of the simulation and describe the corresponding failure mode of the failure state. If the failure state could cause the occurrence of failure mode, add new record in FMEA table, fill in the code, product or function sign, function and failure model in FMEA form, and take the following steps and fill in other items in FMEA form, or proceed with the next failure state until all failure state are handled.

2) According to the failure mode, combine the inner failure state of task profile and function and determine the current task stage of the system and working mode and fill it in FMEA form.

3) According to the failure mode, analyze the failure cause and fill it in FMEA form.

4) According to the failure mode, analyze the failure influence, including local influence, influence of higher layer and final influence, and fill in FMEA form.

5) According to the analysis result of failure influence, determine the severity category of the failure mode, and fill in FMEA form.

6) According to the analysis result of failure mode reason and failure influence, fill in failure detection methods one by one and fill in FMEA form.

7) According to result of failure influence and detection, analyze and adopt remedial measures and fill in FMEA form.

Obtain the corresponding probability value of failure state calculated during the process of simulation, and fill in the column of failure rate in FMEA form.

5. Experiment

Fig. 3 is a simplified aircraft fuel supply system. ATA_XX_Main_Pump is the main pump, ATA_XX_Secondary_Pump is a cold backup of the auxiliary pump, ATA_XX_Detector used to detect the main pump is working correctly, failure occurs when the main pump, send instructions to the auxiliary pump, and then auxiliary pump start-up. ATA_XX_Switch_Valve is a switching valve for switching fuel
oil supply according to the main auxiliary pump. ATA_XX_TANK is a storage tank that is used to store fuel and fuel the aircraft during the fuel supply outage.

ATA_XX_Detector contains an event failure, obeying the exponential distribution, with a probability of 1E-6. ATA_XX_Main_Pump contains an event failure, obeying the exponential distribution, with a probability of 1E-6. ATA_XX_Secondary_Pump contains two events, one of which is Failure, obeying exponential distribution, the probability is 1E-6. The other is event start, obeying the Dirac distribution, with a delay of 0.5 seconds. This means

![Diagram of aircraft fuel supply system](image)

Figure 3. A simplified aircraft fuel supply system.

That the secondary pump has time delay after receiving the start working instruction. The ATA_XX_Switch_Valve contains an event Failure, obeying the exponential distribution, with a probability of 1E-6. The ATA_XX_TANK contains two events, Filling_up, which obeys the Dirac distribution, with a delay of 0.5 seconds, indicating that the fuel filled container takes 0.5 seconds. Another event is Emptying, which obeys the Dirac distribution, and the time delay is 0.3 seconds, indicating that it takes 0.3 seconds to run out of fuel in the tank. All the events are shown in Tab. I.

Using the method mentioned above, the system is modeled and the results obtained by dynamic simulation are shown in the Tab. II, where the total time of dynamic simulation is 10 hours. The failure of the system is ATA_XX_TANK fuel exhaustion, which causes the aircraft fuel to be interrupted. All the cases of ATA_XX_TANK fuel exhaustion are given in the table, including the event sequence before the failure state and the probability of occurrence.

| Tab.1 EVENT DETAILS |
|---------------------|
| Event | Distribution pattern | λ | Delay |
| ATA_XX_Detector.failure | Exponential | 1E-6 |
| ATA_XX_Main_Pump.failure | Exponential | 1E-6 |
| ATA_XX_Secondary_Pump.Failure | Exponential | 1E-6 |
| ATA_XX_Secondary_Pump.starting | Dirac | 0.5 |
| ATA_XX_Switch_valve.Failure | Exponential | 1E-6 |
| ATA_XX_TANK.Emptying | Dirac | 0.3 |
| ATA_XX_TANK.Filling_up | Dirac | 0.1 |

| Tab.2 SIMULATION RESULTS |
|---------------------------|
| Event Sequence | Probability |
| ATA_XX_Switch_valve.Failure -> ATA_XX_TANK.Emptying | 9.999950000172397E-6 |
| ATA_XX_Main_Pump.failure -> ATA_XX_TANK.Emptying -> ATA_XX_Secondary_Pump.starting -> ATA_XX_TANK.Filling_up | 9.999950000172397E-6 |
ATA_XX_Main_Pump.failure
-> ATA_XX_Detector.failure
-> ATA_XX_TANK.Emptying

ATA_XX_Main_Pump.failure
-> ATA_XX_TANK.Emptying
-> ATA_XX_Switch_valve.Failure
-> ATA_XX_Secondary_Pump.starting

ATA_XX_Main_Pump.failure
-> ATA_XX_TANK.Emptying
-> ATA_XX_Secondary_Pump.Failure

ATA_XX_Main_Pump.failure
-> ATA_XX_TANK.Emptying
-> ATA_XX_Secondary_Pump.starting
-> ATA_XX_Switch_valve.Failure

ATA_XX_Detector.failure
-> ATA_XX_Main_Pump.failure
-> ATA_XX_TANK.Emptying

Based on the above results and system reliability requirements, quantitative FMEA analysis can be carried out and FMEA forms can be filled out. The probability of failure is the probability of event sequence.

6. Conclusion
In this work we have presented a quantitative FMEA method, which enables verification and safety assessment of models using dynamic simulation techniques. We use the ten tuple simulation model to describe the internal behavior of the system, and simulate the operation of the system based on discrete event simulation. And then, quantitative FMEA analysis is performed based on simulation results. Finally, we have experimentally demonstrated the feasibility of the approach by evaluating the method on a simplified industrial case. As part of our future work, we plan to improve the system simulation performance by introducing new compilation methods.

References
[1] John Rushby. "Formalism in Safety Cases," Making Systems Safer. London: Springer-Verlag London Limited, 2010: 3-17.
[2] M Bozzano, A Villafiorita, O Akerlund, P Bieber, C Bougnol, et al. "ESACS: an Integrated Methodology for Design and Safety Analysis of Complex Systems," European Safety and Reliability Conference (ESREL). Toulouse: Balkema publisher, 2003: 203-221.
[3] Anjali Joshi, Michael W. Whalen, and Mats P.E. "Heimdahl. Model-Based Safety Analysis Final Report." NASA Contractor Report NASA/CR-2006-213953, 2006.
[4] B Meenakshi, K Das Barman, KG Babu, K Sehgal. "Formal safety analysis of mode transitions in aircraft flight control system," In: AIAA/IEEE Digital Avionics Systems Conference - Proceedings, 2007: 2C11-2C111.
[5] Legay Axel, Delahaye Benoit, Bensalem Saddek. "Statistical model checking: An overview. Runtime Verification," First International Conference, RV 2010, Proceedings, v 6418 LNCS, 2010: 122-135.
[6] Huth Michael. "Some current topics in model checking," International Journal on Software Tools for Technology Transfer, 2007, 9(1): 25-36.