Software FMEA Approach Based on Failure Modes Database

Baiqiao HUANG, Hong ZHANG, Minyan LU
Department of System Engineering
Beihang University
BeiJing, China

***Abstract*—A classification method of software failure modes based on software IPO process is presented. And then two database called general failure modes database (GFMD) and special failure modes database (SFMD) are proposed based on this classification method. Furthermore, a new approach of software FMEA which is based on GFMD and SFMD is presented. This approach which makes the analysis process of FMEA more operable and the failure modes obtained from analysis more comprehensive improves the efficiency of Software FMEA. Meanwhile GFMD and SFMD also offer a platform for the analyzers to accumulate and share their experience. The case study shows that this approach mentioned in this paper is effective in the practice.**

Keywords- Failure mode; IPO; GFMD; SFMD

I. INTRODUCTION

Software failure modes and effect analysis(SFMEA) supposes that failure modes occurred in a software module, then analyze their effects and seek their root cause, and then the corresponding measure will be taken ,so as to avoid introducing software defect and improving software reliability. Software FMEA is such an action that driven by the failure modes. So identifying the failure modes is one of the most important steps, and the quality of the software FMEA is also decided by the identified software failure modes.

Software failure modes have close relation with the feature of the software. Generally the analyzers identify the failure modes by the apperception of the software system and the communicating with the designers. When facing different pattern software, the failure modes need to be identified newly, that is time and resource consuming. And it is much dependent on the analyzer's expertise and familiarity to the analyzed software [4]. So it is necessary to collect and summarize the failure modes for different pattern software, In order to give the software FMEA some experiential instruction. In the paper [1], the author summarized the failure modes for system lever FMEA, such as "Fails to execute", "Executes incompletely", "Output incorrect" and "Incorrect timing-too early, too late, slow, etc". However it is somewhat simple. In GJB1391A, a software failure modes category list is recommended. In the list, there are some common failure modes which are classified by five categories. However it is also too simple to instruct the software FMEA. In the paper [4], the author collected the failure modes from the same pattern software or software alike. And divide these failure modes into two parts, i.e. those that may appear in general software and those that only appear in aircraft embedded software. But it does not describe more details.

This paper presents a failure modes classifying approach which bases on software IPO structure. It divides the failure modes into three parts called "Input failure", "process failure" and "Output failure". And some common features have been abstracted from the three categories, called general failure mode, and then found the GFMD. Since different pattern software has different failure modes, these failure modes which only appear in special software were called special failure modes, and divide them by software pattern. They are the member of the SFMD. A software FMEA approach bases on the two databases is presented also. In this approach ,we divide the software module into three logic structure, "input ","process " and "output", just like the failure modes category. And then find the failure modes for the object from the two databases, afterwards, the effect and root cause. The detail of founding the FMD is mentioned in part two. FMD based Software FMEA approach is mentioned in part three. In part four, there is an example of this approach, and part five is the conclusion.

II. FAILURE MODES DATABASE BASED ON IPO CLASSIFICATION

A. The IPO Structure of Software Module

Whether a whole system or a sub-function, all software products can be regarded as consisting of three logic parts of IPO (Input, Process, Output), where the Input receives the exterior inputs, the Process makes the necessary transactions and the Output transfers the results of transactions. Moreover, this is also composed by linking each part of IPO together, as shown in Figure 1. The defects introduced in any part of IPO may result in software failure. Hence, failure modes are classified by using these three parts. Accordingly, the SFMEA approach also should make the analysis with respect to the three parts of IPO in each module.

978-1-4244-4905-7/09/$25.00©2009 IEEE

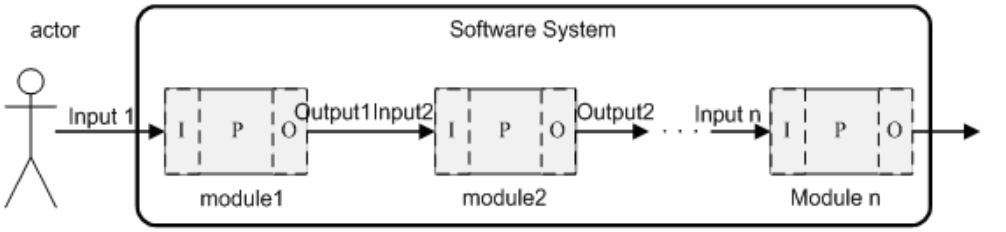

Figure 1. Software IPO structure

B. The Classification and Collection of Failure Modes

To conclude and summarize the software failure modes, failure data should be collected as many as possible first. The resource of our database mainly consists of testing records of various patterns of software products, aviation software failure cases in publications, and the deduction according to the failure cases. If the failure modes can be classified according to the location of software module IPO in which the failures are triggered, it can be divided into "Input failure modes", "process failure modes" and "Output failure modes".

One reason that the software failure modes are more complex than hardware is the software failure modes are time-dependency. Especially for embedded software, the mistake in one input or output of signal due to time sequence even makes the whole software in confusion. Therefore, for software module, it not only has the numerical and functional failure modes, but also has the time-sequence failure modes which are very significant to the FMEA analysis in embedded software. Several common time-sequence failure modes are such as the signal input/output too early, too late, overtime and frequency abnormity.

Failure mode has its generality and character. For example, there are many input patterns, such as keyboard input, hard disk file input, memory address variable input and etc. Each input pattern has its special failure modes different from the others. Therefore, if these failure modes in input process can be concluded and labeled by "keyboard input process failure modes" or "file input process failure modes", they are with respect to the special process and can be called the special failure modes. Summarizing all failure modes in each process of IPO and abstracting the common express methods of each corresponding process, the general failure modes of three processes are presented. The common failure modes are applicable for large numbers of software modules. The main advantage of the common failure modes is that if there are no appropriate special failure modes can be used to guide the analysis in software FMEA process, the failure modes of the analysis object can be determined according to the corresponding common failure modes with the characters of software modules.

C. Constituting the Failure Modes Database

The FDB can be constituted by classifying and concluding the collected failure modes data according to the classification method mentioned above. In the GFDB, the general failure modes which are abstracted from large numbers of failure modes can be divided into three categories, namely, "Input process", "Manage process" and "Output process". Furthermore, each category of the GFMD can be divided into two sub-category, namely, "data failure" and "time-sequence failure". The "Input process" failure modes of the common failure modes database are enumerated in Figure 2. The SFMD is also divided into three categories according to the three parts of IPO. Various failure modes in the special software process are collected in each category. The "file input process" and "input data error" failure modes in special failure modes database are enumerated in Figure 3. The relationship between the general failure modes and the special failure modes which can be found from the comparison of Fig.2 and Fig.3, is that one failure mode in the GFMD may be correspond to the several failure modes in the SFMD.

There are 43 general failure modes belong to three parts of IPO in GFMD. And there are 184 special failure modes belong to 15 kinds of special software modules. FMD is the conclusion of the experiences in SFMEA, and can be expanded constantly along with the accumulated analysis experiences. FMD not only provides a platform used to accumulating the experiences for analyses, but also a data bank can be referenced by the analyses who are lack of experiences.

General Failure modes	
Time-Sequence Input Failure Modes	data/signal too early
	data/signal too late
	data/signal is overtime
	data/signal is frequency abnormity
Data Input Failure Modes	data is reversed and misplace
	data is redundant
	data is deficit
	data precision error
	data is out of range
	data error format
	Right inputs are refused
	right range but wrong value

Figure 2. General Failure Mode for Input

Special failure modes	
File input failure modes	File name is wrong
	File name is invalid
	File not exist
	File is opened
	File format is wrong
	File format is invalid
	File head is error
	File ending is error
	File length is wrong
	File data is lack of some right information
	File context is blank
	File data information is wrong

Figure 3. General Failure Mode for Input

III. THE SFMEA APPROACH BASES ON THE FAIURE MODES DATABASE

The implementation steps of SFMEA are presented in GJB1391A, literature [3] and [5]. clipping these steps and combining with the FMD based on the IPO classification, a SFMEA approach bases on the FMD is proposed. The results of case study show that this approach is much operable and effective. The detailed implementation steps of this approach are introduced below:

1) familiarizing software system, and illustrating the function hiberarchy picture

Before analysis, the analysts need to understand the software function and structure. And illustrate the function hiberarchy picture. The function hiberarchy picture is the basis for deciding analysis range and first analysis level, and analyzing the effect of failure modes. Familiarizing the software characteristic is the preparative action for identifying the failure modes. The way for this is to communicate with the designer and read software document.

2) Define the analysis rule

The analysis rule is the criterion and restriction to the whole analysis actions, which defines the analyses process and restricts the casualness of the analysis actions, and makes the analysis results of the same analysis object among different analyses groups won't have large differences. The common rules include the restriction rule to the analysis range, the method used to decide the first analysis layer, the definition of severity level and etc.

Software FMEA is called the reliability design and analysis approach and only used for the analysis of the safety-critical software or the safety-critical modules in software for it is time-consuming and effort-consuming. Hence, software FMEA is not an ergodic analysis. The range of the software FMEA actions this time is determined by combing the evaluation of the critical degree of each module with the time resource and staffs in the analysis group.

For FMEA is an analysis approach from bottom to top, thus the initial analysis layer should be determined. If the granularity of layer selection is quite large, the analysis result is limited. Then, the appropriate initial analysis layer should be selected and the selection rule also should be determined before the FMEA.

The severity level is the evaluation of the influence consequence which is obtained from FMEA in the analysis process, and provides the reference for the decision on the implementation of the improvement measures. The determination of the severity level should be decided by the experts who are familiar with the system. The severity is determined by the final influence of failure modes on the system.

3) Decide the analysis range and the first analysis layer

According to the rules referred in the upper chapter, and combining with the object software actual conditions, the analysis range and the first analysis layer should be decided. It is recommended that when software FMEA carried out in the requirement phase, make the least function module found in the requirement specification as the first analysis layer, namely make the leaf node in the function hiberarchy picture as the first analysis layer. When in the design phase, make the least design module found in the design document as the first analysis layer.

4) Divide the object into IPO structure, and identify the failure modes according to the FMD

After determining the analysis object, the next key action is to identify the object's failure modes. According to the IPO classification database, first divide the software module into IPO logic structure, and then find the corresponding failure mode from the database for each process. The principia of using the database is that if there are corresponding special failure modes in the SFMD, then pick out them as the candidate failure modes. If not, then use the GFMD as a reference. And identify the failure modes According to the feature of the software. It should be noted that the FMD is just the experience anciently, the failure modes in the database not always very suitable for the object. So it is necessary to filtrate or make some change. And failure modes selected from the database is not completely, when doing software FMEA, the analyzer should add new failure modes according to the actually conditions.

Dividing the software module into three logic parts makes the software FMEA more operable, especially for complex system. Identifying the failure modes from the FMD, not only speed up the analysis progress, but also utilize the experience in the past. So it raises the efficiency of the software FMEA.

5) Analysis the effects, seek the root courses, find the measure, and fill in the FMEA list

After identifying the failure modes, the next step is to analyze its effect on each layer, such as local layer, next high layer and system layer. Give a severity level according to the severity level identifying list. Then find out the cause of the

failure and the measure to avoid or reduce its serious effect. And at last fill in the software FMEA list.

IV. CASE STUDY

In this section, a case study is presented to illuminate the detailed analysis process of the SFMEA approach based on FMD presented in this paper.

The DPS control is required in a software system. The devices of DPS should do the self-test before DPS are operated in which DPS loads a section program from hard disk. Comparing the results of the self-test with the oracle, the self-test process is perfect if the results are right. According to the analysis, the layer in which the DPS self-test exists can be regarded as the first analysis layer of FMEA. Now, a case study of software FMEA is implemented on the DPS self-test for advanced analysis. The following analysis process is belonging to the fourth step and the fifth step mention in section 3.

Firstly, the input, manage process and output of the analysis object are defined respectively.

Input: file loading, the oracle;

Manage process: the implementation of self-test, achieving the self-test results.

Output: the conclusion is proposed by comparing the self-test result with the oracle.

And then, the appropriate failure modes are determined for each process according to the FMD based on IPO classification. The input includes the file loading and the oracle, where the file loading can find the appropriate reference from the input failure modes in the SFMD, shown in Fig.3. It should be noted that, not all the failure modes with respect to file input are effective for this analysis. According to the characters of the analysis software product, the self-test program file is a short segment of assembler which is saved in the installation directory of software in terms of the txt file. Two most possible failure modes are selected from the failure modes database in Fig.3 for FMEA. The oracle is a global variable saved in memory. Because there are no corresponding failure modes in the special failure modes database, the value error of the oracle is selected as the failure mode for FMEA according to the input general failure modes in Fig.2. Consequently, the failure modes in input process of the DPS self-test is obtained and shown in Fig. 4. Finally, based on these failure modes, the effect and cause of these failure modes are analyzed for the improvement measures and completing the form of SFMEA.

Module	Failure Mode	Cause	Failure Effect			Severity	Measure
			Local Effect	NHL Effect	System Effect		
DPS Self-test	File Content Error	importing error when amend the file	DPS Self-test Error	Software System Self-test Error	Result in Experiment delay	ii	Leave a file copy and validate Each Other
	File Not Exit	(1) file has been removed (2) absolute path of the file used in the program	DPS Self-test Exit	System Self-test Can not get DPS' s state	Result in Experiment delay	ii	(1) give out warning when file not exit (2) avoid absolute path in the program
	Oracle Error	File changed but not the Oracle accordingly	DPS Self-test alarm	System Self-test Can not get DPS' s state in dead	Result in Experiment delay	ii	Design a way in the software to amend the Oracle
	.....	.....	.....	.....	.....	.....	.....

Figure 4. DPS Self-Test Input Process Software FMEA

Furthermore, the SFMEA approach based on FMD presented in this paper has applied on two kinds of software products for FMEA in the requirement and design phase. The results of application show that this approach can advance the efficiency and operability of software FMEA. Due to the limited space, the detailed description is not discussed here.

V. CONCLUSION

Failure modes database provides a useful platform which can be used to accumulate analysis experience for software FMEA analysis staffs. Especially, because the pattern of software which is developed in one company is a little single, failure modes which are accumulated more and more have became the powerful support for advancing software reliability. Meanwhile, the SFMEA approach based on FMD not only makes the analysis process more operable, but also determines the failure mode of analysis object more quickly, so as to highly improve the efficiency of software FMEA. Furthermore, the case study also shows that, the SFMEA approach based on failure modes database in this paper is available.

REFERENCES

[1] P.L. Goddard, Software FMEA Techniques. Annual Reliability and Maintainability Symposium, 2000.

[2] J.B. Bowles, and C. Wan, "Software Failure Modes and Effects Analysis For a Small Embedded Control System," Proceedings Annual Reliability and Maintainability symposium, 2001.

[3] N. Ozarin, "Failure Modes and Effects Analysis during Design of Computer Software," RAMS, 2004.

[4] W. Dong, J. Wang, C.Z. Zhao, X. Zhang, and J. Tian, "Automating software FMEA via formal analysis of dependence relations," Annual IEEE International Computer Software and Application Conference, 2008.

[5] N. Ozarin, "The Role of Software Failure Modes and Effects Analysis for Interfaces in Safety-and Mission-Critical Systems," IEEE International Systems Conference, 2008.

[6] GJB1394A Failure Modes , Effect ,and Criticality Analysis.

[7] B. Wu, and R.Z. Tang, "Study on software FMEA technology," Mechanical and Electrical Engineering Magazine, 2004, Vol.21, No.3.