A Model-based RCM Analysis Method

Zhibao Mian
College of Computer Science & Engineering
Northwest Normal University
Lanzhou, China
mianzhibao@163.com

Shuli Jia
Department of Computer Science
North Minzu University
Yinchuan, China
jiashuli@stu.nmu.edu.cn

Xiaodong Shi
Department of Computer Science
North Minzu University
Yinchuan, China
shixiaodong@stu.nmu.edu.cn

Cairong Tang
Department of Computer Science
North Minzu University
Yinchuan, China
tangcairong@stu.nmu.edu.cn

Junjie Chen
Department of Computer Science
North Minzu University
Yinchuan, China
chenjunjie@stu.nmu.edu

Yaqing Gao
Department of Computer Science
North Minzu University
Yinchuan, China
gaoyaqing@stu.nmu.edu.cn

***Abstract*—The reliability-centered maintenance (RCM) is one of the most advanced maintenance plan generating technologies for equipments. At present, the key technologies such as FMEA and FMECA supporting the RCM analysis remains in the manual stage in some enterprises. The disadvantages are time-consuming, labour-intensive and error-prone. For complex systems containing thousands of components, to achieve a fast and effective FMECA analysis is difficult. RCM should benefit from the most advanced model-based systems engineering methods. In this paper, a model-based RCM analysis framework (MRAF) is presented. Based on this framework, RCM analysts can use model-based reliability analysis techniques, such as AADL, to model the system architecture and faults information. Then the AADL-based analysis platform OSATE can be used to automatically produce the FMEA. By combining the generated FMEA with the criticality analysis (CA) technology, this paper can semi-automatically generate the FMECA for equipments and systems being analyzed by RCM.**

Keywords—Reliability-centered maintenance, AADL, MRAF, FMEA, FMECA

I. Introduction

Currently, the functions and structures of equipments are gradually developing towards the direction of complexity and diversification, which makes their dependability analysis much more difficult. It is very hard to determine which component or process that has caused the entire system to fail by traditional maintenance methods such as regular inspection. Moreover, to satisfy the dependability performance of equipments and systems in each phase of its maintenance life cycle, an automated dependability analysis is required. There is still a lack of automated, accurate, dependable, and reusable analysis methods in this field.

Some advanced maintenance methods such as Total Productive Maintenance (TPM) [1], Condition-based Maintenance (CBM) [2], Preventive Maintenance (PM) [3], Reliability Centered Maintenance (RCM) [4] are compared and analyzed in this paper (as shown in table I).

.

TABLE I. The Comparison of the TPM, CBM, PM and RCM

Name	Context	Advantage	Disadvantage
TPM	It focuses on maximizing equipment efficiency by creating the perfect relationship between employees and equipments [1].	Its teamwork could maximize efficiency of equipments.	Its maintenance content is wide; It is only carried out after equipment failing.
CBM	It obtains the equipment information by detecting the equipment; It uses this status information to make equipment maintenance requirements.	Condition monitoring could obtain equipment failure information in real time or near real time; Minimizing time spent on maintenance [2].	Condition monitoring test equipments and costs to train staff are expensive.
PM	It is regularly performed on a piece of equipment to lessen the likelihood of it failing [3].	Making planning is the biggest benefit to prevent failure.	High cost, labor-intensive, large spare parts inventory and unnecessary maintenance.
RCM	It uses failure modes and effect analysis methods based on reliability theory to determine equipment maintenance needs and methods [7]; It focuses on the critical system (the important functional system) and improves the reliability and safety of equipments.	It can avoid or reduce unnecessary maintenance work and minimize maintenance costs [6]; Selecting the most appropriate maintenance tactic (TPM, CBM, etc.) for each failure mode of a system [5].	Its seven steps are implemented separately from each other, which is inconvenient.

Following the analysis of the above methods, the RCM is selected as a maintenance technology with great economy and reliability for complex system in this paper. The traditional procedure of applying the RCM method [8] is as follows:

978-1-7281-8915-4/20/$31.00 ©2020 IEEE
DOI 10.1109/QRS-C51114.2020.00059

- Step1: System selection and data collection. The most critical system should be screened because the RCM analysis requires time and resources. Next, the system information is to be collected, mainly including system components information and fault data.
- Step2: System boundary definition. System boundaries and definitions lay the foundation for system selection in step1. They usually have been found in the normal course of equipment design.
- Step3: System components and functional block description. The essential details of the critical system must be identified and recorded to perform the remaining steps in a thorough and technically reasonable manner. The functional block is a top-level representation of the major functions that the system performs.
- Step4: System functions and functional fault definition. The previous steps provide the basis for effectively promoting the defined system functions in the step. A complete list of system functions is defined by RCM analysts. They also need to define the functional failures to determine how functions might be defeated.
- Step5: Generate system FMEA and FMECA. This step is to identify failure modes that could potentially generate unexpected functional failures. The failures are based on the results of step4 of RCM. RCM expert analyzes the impact of these failures on the entire system. The FMECA goes a step further, assessing the risks associated with each failure mode and then prioritizing corrective action.
- Step6: RCM logic decision analysis. The failure modes in step5 are further classified in this step. The goal of this step is to further prioritize the resources and emphasis in terms of their impact on each failure mode.
- Step7: Maintenance plan. For each of these failure modes identified in step6, this step is to ascertain a list of appropriate candidate tasks. Then the most effective task from among the competing candidates is eventually selected to formulate a maintenance plan.

In these above 7 steps, the first 4 steps are used to collect equipment's information. Step 5 comprehensively analyzes those collected information and produces important FMEA and FMECA tables. Meanwhile, FMEA and FMECA plays a key role in the logical decision step (step6) and the maintenance plan step (step7). Thus, how to generate an accurate FMEA and FMECA is crucial for RCM analysis.

Traditional FMEA and FMECA analysis methods are highly subjective. The accuracy of the analysis results highly depends on the engineers' skills. For the same system, FMEA and FMECA analyzed by different engineers may vary greatly due to differences in their knowledge and thinking approaches [9]. Meanwhile, these technologies also gradually show some shortcomings, e.g. consuming a large amount of time, manpower, material resources and are error-prone for complex systems. One of the major difficulties is that RCM needs to link and track all of the various functional failure-component combinations to generate the FMEA. For small systems, the FMEA could be produced manually. But for systems with a large number of components, it is quite hard to generate the FMEA by using manpower. The model-based system engineering (MBSE) method [11] is considered as an effective means to solve this problem. The MBSE emphasizes the establishment of a comprehensive model of system function and reliability. The model requires defining a system's constituent elements and normal operating behaviors. Then, the fault behaviors of system are described and annotated to a system model, so that FMEA and FMECA can be automatically exported.

In light of the above research, this paper proposes a model-based RCM analysis framework (MRAF). The MRAF is used to perform the RCM analysis by building a comprehensive model for a complex system. The processing is that the RCM's first four steps' information is established into a system model by using modeling languages and tools. By using the proposed MRAF, the FMEA can be automatically generated. The FMECA table can be also semi-automatically generated. These tables can be further used later for RCM logic decision analysis and maintenance plan generating.

II. Related Work

The MBSE provides supports for automated FMEA and FMECA analysis through the architecture fault and criticality modeling [25]. Reference [12] introduced the development of model-based reliability analysis techniques for complex system. In model-based system engineering, it is required to build various modeling languages to support various analysises. Researchers had compared and analyzed some famous modeling languages, including UML (unified modeling language), East-ADL (electronics architecture and software technology-architecture description language), SysML (system modeling language) and AADL (architecture analysis and design language) [14-16], and summarized their advantages and disadvantages. From the comparison of the above modeling languages, AADL's architecture and fault modeling mechanisms could promote RCM analysis and is easier to meet the modeling requirements for large complex systems. Therefore, AADL and its supporting open-source tool platform OSATE [22] are selected to demonstrate the proposed MRAF method. Systems' architecture is modeled by using AADL's specific semantics elements. The fault information (error model) of a system is described by using AADL-based error model annex EMV2 [10, 19]. Moreover, EMV2 specifies the system failure behavior and error propagation to solve the reliability aspects of the system architecture [18, 19], which is very suitable for the reliability analysis background for the proposed MRAF method.

Some research has been done for using AADL and its EMA to automatically generate FMEA [20-21]. Wang and et al. [26] proposed a reliability modeling method for the Integrated Modular Avionics System based on AADL and EMA [17]. Gu and et al. [29] developed the FMEA and CA properties into AADL's error model annex to generate the FMECA table automatically. Currently, their method has not yet been integrated into OSATE. In this paper, we adopt the CA method developed in [29] to create AADL's criticality model.

III. The Model-based RCM Analysis Framework (MRAF)

A. An overview of the Framework

This paper proposes a MRAF as shown in the righthand side of Fig. 1, which integrates the seven steps of the traditional RCM analysis procedures (as shown in the left-hand side of Fig. 1) into a comprehensive model by using the MBSE techniques [13]. The comprehensive model is divided into three phases to integrate different steps of RCM. This paper concentrates on the first phase.

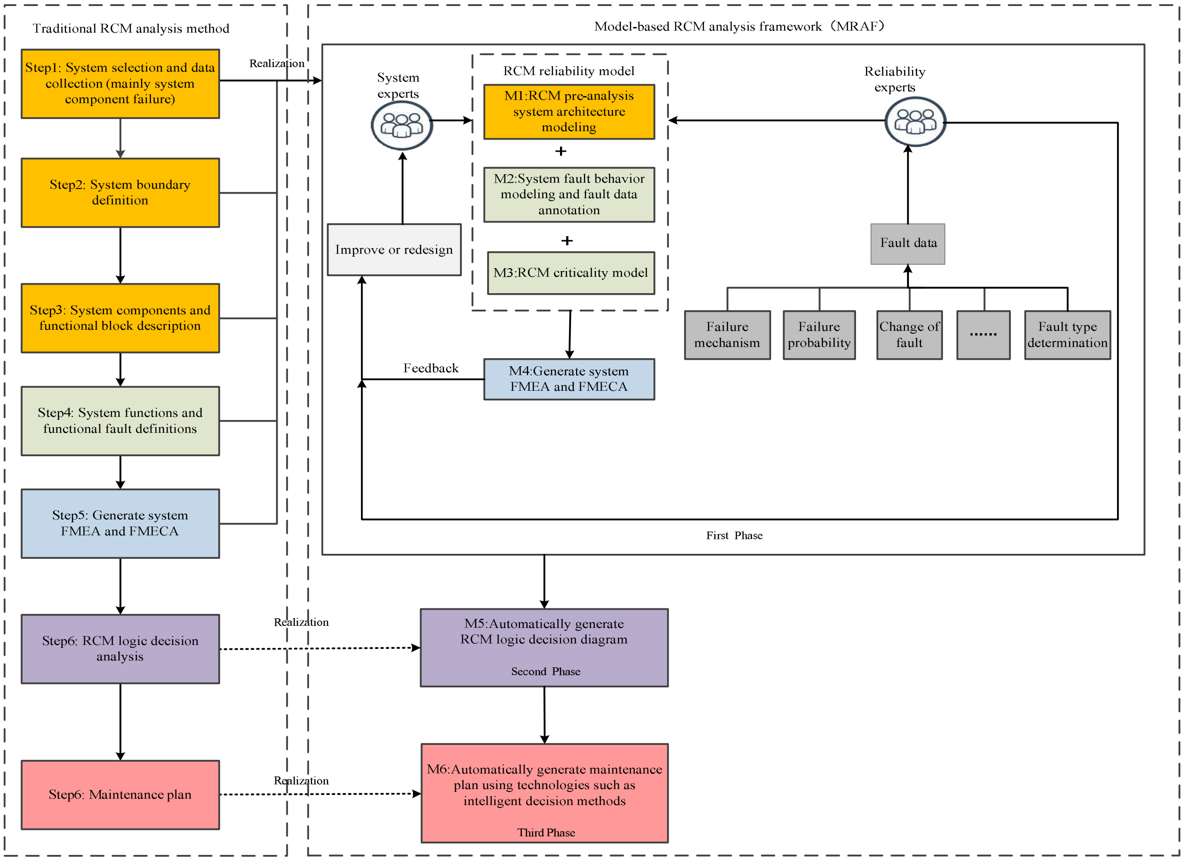

Traditional RCM analysis method
Step1: System selection and data collection (mainly system component failure)
Step2: System boundary definition
Step3: System components and functional block description
Step4: System functions and functional fault definitions
Step5: Generate system FMEA and FMECA
Step6: RCM logic decision analysis
Step6: Maintenance plan
Realization
Realization
Realization
Model-based RCM analysis framework（MRAF）
System experts
RCM reliability model
M1:RCM pre-analysis system architecture modeling
+
M2:System fault behavior modeling and fault data annotation
+
M3:RCM criticality model
Reliability experts
Fault data
Failure mechanism
Failure probability
Change of fault
......
Fault type determination
Improve or redesign
Feedback
M4:Generate system FMEA and FMECA
First Phase
M5:Automatically generate RCM logic decision diagram
Second Phase
M6:Automatically generate maintenance plan using technologies such as intelligent decision methods
Third Phase

Fig. 1. The correspondence diagram of the traditional RCM analysis method and the model-based RCM analysis framework (MRAF).

In Fig. 1, the first three steps of the traditional RCM corresponds to the MRAF's M1. The RCM's step4 corresponds to the M2 and M3 of the MRAF. The RCM's step5, step6 and step7 corresponds to MRAF's M4, M5 and M6 respectively.

The first phase of the MRAF includes the RCM reliability model, fault data, manpower such as the system and reliability experts. By adopting the model-based reliability analysis technology, the defined features of the first four steps of the traditional RCM are implemented in the RCM reliability model. Next, through analyzing the RCM reliability model, the FMEA and FMECA shown in the step5 of the traditional RCM method will be obtained automatically. Besides, the fault data in Fig. 1 is defined to collect critical systems' historical fault data such as failure probability. The fault data is collected and analyzed by the reliability experts for the RCM reliability model. The second phase describes that the RCM decision logic diagram could be automatically generated by adding RCM logic decision analysis information to the MRAF model. The third phase adds the maintenance information into the MRAF model by using intelligent maintenance planning technologies to automatically analyze and generate maintenance plans.

B. The Description of the First Phase of MRAF

1) RCM pre-analysis system architecture modeling: The pre-analysis system in the RCM reliability model mainly builds an RCM architecture model for the first three defined features of traditional RCM analysis. A system's architecture model is built for each component and subsystem and connections between them by using architecture modeling languages such as SysML [23] or AADL.

It is important to note that the system architecture model, could only include the lowest level of repairable and detachable components.

2) System fault behavior modeling and fault data annotation: This step mainly builds an RCM error model specifying the fault and dangerous behavior of the system. The fault information is annotated into the RCM architecture model. Annotation means associating the architecture model with corresponding fault data.

3) RCM criticality model: A criticality analysis (CA) feature of the system (shown as M3 in Fig. 1) is implemented to the system model to build an RCM criticality model. The CA [27] feature could assign criticality ratings to assets based on their potential risks. It is composed of risk priority number (RPN).

The RPN consists of occurrence probability ranking (OPR), effect severity ranking (ESR) and detection difficulty ranking (DDR). The RPN value is composed of the product of the value of OPR, ESR and DDR (range from 1 to 10) as defined in IEC 60812 [27]. The value of OPR, ESR and DDR could be obtained by combining with the experience and knowledge of experts and standards. Finally, the RPN promotes the establishment of the RCM criticality model.

4) Generate system FMEA and FMECA: The RCM architecture model, fault model and RCM criticality model are used to build the RCM reliability model. The RCM reliability model can be used to automatically generate the FMEA and FMECA reports including failure modes, failure effects and hazard analysis for the overall system. The reports can help engineers to obtain reliability defects and eliminate or control component hazards to an acceptable level. More important, in the context of this paper, advanced maintenance decisions for the system can also be obtained automatically based on those important FMEA and FMECA reports.

IV. ILLUSTRATIVE EXAMPLE

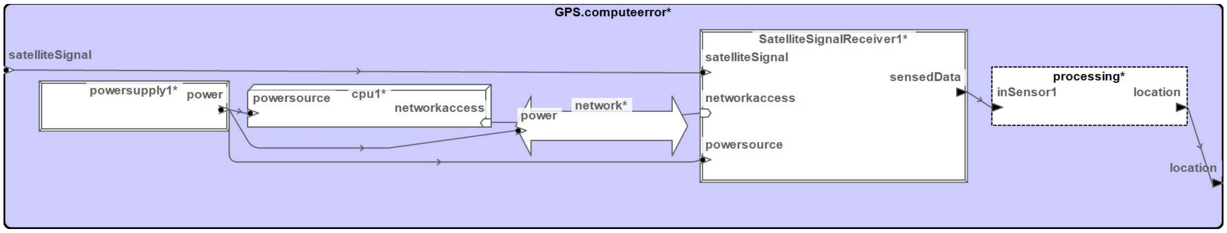
In this section, to realize the application of MARF, a GPS's important subsystem is analyzed by the MRAF. The RCM reliability model in the MRAF is implemented as an AADL reliability model. An AADL reliability model includes an AADL architecture model, an AADL error model and an AADL criticality model. These models can be modelled by using AADL's OSATE platform [22].

A. AADL Architecture Modeling for the GPS.computeerror System

A complex GPS system can be divided into several subsystems in terms of functional independence. A system selected should be critical depending on its effect on operations, its previous costs of repair, its frequency of failures and time leading to downtime. Therefore, the GPS.computeerror system, one of the important GPS subsystems [28], is chosen to illustrate the MRAF method. We adjusted the GPS.computeerror system for the convenience of demonstrating our method. This system is mainly used to calculate possible errors in positioning data. The system's software, hardware architecture and their corresponding functions are described by using AADL concepts of components and connections.

Fig. 2 shows the top-level architectural model of the GPS.computeerror system. In the top of Fig. 2, an AADL architecture model of the system is built on the OSATE platform by using its graphical modeling approach. The corresponding AADL text description is shown in the bottom of Fig. 2. The system mainly has the following components: power supply (component type device), satellite signal receivers (device), CPU (processor), processing (process) and network (bus). Each system or task has its corresponding input and output data and events, which is implemented through the concept of ports, such as satelliteSignal (abstract port), networkaccess (access port), senseData (data port). The functional requirements of the system (data exchange) are realized through connections such as sattoSatelliteSignalReceiver1 (line 13 in Fig. 2). An abstract processing is used to describe one of the system tasks and is achieved by using AADL's process component type.

The AADL architecture model of the GPS.computeerrorsystem is created by the above modeling work. This model is the implementation of the RCM pre-analysis system architecture modeling in the first phase of the MRAF method.



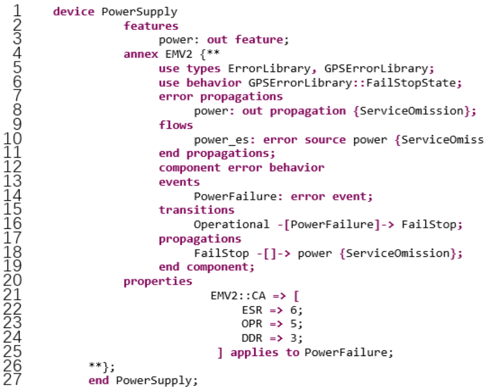
```
1 system implementation GPS.computeerror
2   features
3                       satelliteSignal: in feature;
4                       location: out data port;
5           subcomponents
6                       SatelliteSignalReceiver1: device GPSParts::sensor;
7                       processing: process GPSParts::GPSProcessing;
8                       cpu1: processor GPSHardwareParts::CPU;
9                       network: bus GPSHardwareParts::Network;
10                      powersupply1: device GPSHardwareParts::PowerSupply;
11          connections
12                      -- logical connections
13                      sattoSatelliteSignalReceiver1: feature satelliteSignal -> SatelliteSignalReceiver1.satelliteSignal;
14                      s1toproc: port SatelliteSignalReceiver1.sensedData -> processing.inSensor1;
15                      proctoext: port processing.location -> location;
16                     -- physical network connections
17                      s1tonetwork: bus access network <-> SatelliteSignalReceiver1.networkaccess;
18                      cputonetwork: bus access cpu1.networkaccess <-> network;
19                      -- power connections
20                      powertos1: feature powersupply1.power -> SatelliteSignalReceiver1.powersource;
21                      powertocpu: feature powersupply1.power -> cpu1.powersource;
22                      powertonetwork: feature powersupply1.power -> network.power;
```

Fig. 2. The AADL architecture model and its corresponding text description for the GPS.computeerror system [28].

B. AADL Error Model for the GPS.computeerror

After completing the AADL architecture modeling, it needs to build a corresponding error model, i.e. to annotate each component with an error behavior information for the system. To illustrate the AADL error model construction process, the component powersupply1 is used as an example (as shown in Fig. 3). Other components use the similar fault modeling method. The error model includes the declarations of error events, error propagations and error flows.

In Fig. 3, The PowerSupply defines a component error behavior to declare the error event ‘PowerFailure’. After occurring the ‘PowerFailure’ error event the state of the device will change its ‘operational’ state to the ‘FailStop’ state. Then, the device will propagate the error type ‘power{ServiceOmission}’ through the three connections declarations (from line 20 to 22 in Fig. 2).



```
1    device PowerSupply
2            features
3                power: out feature;
4            annex EMV2 {**
5                use types ErrorLibrary, GPSErrorLibrary;
6                use behavior GPSErrorLibrary::FailStopState;
7                error propagations
8                    power: out propagation {ServiceOmission};
9                flows
10                   power_es: error source power {ServiceOmission};
11               end propagations;
12               component error behavior
13               events
14                   PowerFailure: error event;
15               transitions
16                   Operational -[PowerFailure]-> FailStop;
17               propagations
18                   FailStop -[]-> power {ServiceOmission};
19               end component;
20           properties
21                     EMV2::CA => [
22                         ESR => 6;
23                         OPR => 5;
24                         DDR => 3;
25                      ] applies to PowerFailure;
26       **};
27       end PowerSupply;
```

Fig. 3. The error model for the powersupply1 component in the GPS.computeerror system.

The system component failure and impact information are annotated into its AADL architecture model to create an AADL error model so as to specify the fault and dangerous behavior of the system. This work is the implementation of the system fault behavior modeling and fault data annotation in the first phase of the MRAF method.

C. AADL criticality model for the GPS.computeerror

In the bottom of Fig 3, a CA property is defined and annotated to EMV2. According to the criticality analysis (CA) standard and the experience of RCM experts, the effect severity ranking (ESR), occurrence probability ranking (OPR), and detection difficulty ranking (DDR) grades (from line 20 to 25 in Fig. 3) for the error event ‘PowerFailure’ defined for the component powersupply1 are annotated to the AADL model. The annotation of CA property for each component is used to create an AADL criticality model. The combination of AADL architecture model, AADL error model and criticality model is called AADL reliability model. This relates to the concrete implementation of the RCM reliability model.

D. Generate system FMEA and FMECA based on AADL

The previous steps have developed an AADL reliability model including the first four defined features of RCM. AADL then uses this reliability model to obtain the FMEA and FMECA. By using the ‘Analyze Fault Impact’ command on the OSATE platform, the FMEA report for the above built AADL reliability model can be automatically generated as shown in table II. By combining the generated FMEA with the criticality analysis (CA) technology, the paper can semi-automatically produce the FMECA for the GPS.computeerror system as shown in table II.

TABLE II. THE FMECA REPORT FOR THE GPS.COMPUTEERROR SYSTEM

FMECA												
FMEA									***CA***			
No.	***Component***	***Initial Failure Mode***	***1st Level Effect***	***Failure Mode***	***2nd Level Effect***	***Failure Mode***	***3rd Level Effect***	***...***	***OPR***	***ESR***	***DDR***	***RPN***
1	SatelliteSignalReceiver1	{SensorFailure}	{ServiceOmission} sensedData -> processing:inSensor1	processing{ServiceOmission}	{ServiceOmission}location -> GPS_computeerror_Instance: location [External Effect]			...	6	3	5	90
2	network	{NetworkFailure}	{ServiceOmission} bindings -> [No Binding]					...	7	3	5	105
3	network	{NetworkFailure}	{ServiceOmission} access -> SatelliteSignalReceiver1:networkaccess	SatelliteSignalReceiver1{ServiceOmission}	{ServiceOmission} sensedData -> processing:inSensor1	processing {ServiceOmission}	{ServiceOmission} location -> GPS_computeerror_Instance:location [External Effect]	...				
4	network	{NetworkFailure}	{ServiceOmission} access -> cpu1:networkaccess	cpu1{ServiceOmission} [All Out Props]	{ServiceOmission} networkaccess -> network:access	network {ServiceOmission}	{ServiceOmission} bindings -> [Propagation Cycle]	...				
5	cpu1	{CPUFailure}	{ServiceOmission} bindings -> [No Binding]					...	7	5	3	105
6	cpu1	{CPUFailure}	{ValueError} bindings -> [No Binding]					...				
7	processing	error event computeError	{InaccurateData} location -> GPS_computeerror_Instance:location [External Effect]					...	5	3	3	45
8	processing	{LowPrecisionData}	{LowPrecisionData} location -> GPS_computeerror_Instance:location [External Effect]					...	7	5	2	70
9	powersupply1	error event PowerFailure	{ServiceOmission} power -> SatelliteSignalReceiver1:powersource	SatelliteSignalReceiver1{ServiceOmission}	{ServiceOmission} sensedData -> processing:inSensor1	processing{ServiceOmission}	{ServiceOmission} location -> GPS_computeerror_Instance:location [External Effect]	...	5	6	3	90
10	powersupply1	error event PowerFailure	{ServiceOmission} power -> cpu1:powersource	cpu1{ServiceOmission}[Unhandled Failure Effect]				...				
11	powersupply1	error event PowerFailure	{ServiceOmission} power -> network:power	network{ServiceOmission}	{ServiceOmission} bindings -> [No Binding]			...				

The report is generated by tracing a fault occurrence from its error sources through the error flows within components and propagation paths between components. The error flows are determined by error flow declarations. The propagation paths are determined by AADL connection declarations. It traces a failure from its error source or error event. Each effect column indicates the path of the outgoing propagation of one component to another component being affected by the propagation. The resulting failure mode of the receiving component is indicated in the next column. The trace terminates as an external effect, that is, impact to the operational context of the top-level system, as being masked, or a number of other indicators.

In table II, the SatelliteSignalReceiver1's initial failure mode shows that the component as a failure source occurs the fault '{SensorFailure}'. The fault propagates to the processing directly connected to it and thus is the 1st level effect. Then, the effect causes the processing occurs the failure 'processing{ServiceOmission}' and has a second level effect. The network's error '{NetworkFailure}' impact terminates as a 'No Binding', that is, the outgoing propagation is for a binding point, but the binding has not been specified yet. This network's error has three propagation paths. The third propagation path (No.4) propagates the fault process with a fault mode 'All Out Props'. This indicates the situation when an incoming propagation is mapped to all outgoing propagations. At last, the fault impact terminates as a 'Propagation Cycle'. It means the impact trace reaches an element in the trace that has previously propagated the same error type on the same outgoing propagation point.

The powersupply1's (No.10) initial failure mode shows that the component occurs a failure event 'PowerFailure'. Its error impact terminates as an 'Unhandled Failure Effect'. It means an incoming failure effect that is not handled as sink or by an outgoing error propagation, i.e., the incoming propagated error type is not listed in any error paths or outgoing error propagations.

For the CA in table II, the network and cpu1 have the largest RPN number. This means that relevant work should be done to focus on solving the components' problems when formulating the maintenance plan of the system. The influence factors of their failure modes should be reduced.

The FMECA and FMEA provide an effective and scientific basis for further faults classifying of RCM's logical decision (step6) and formulating maintenance plan (step7). Some suggestions for the formulation of maintenance programs are given. For instance, if the failed component has a low reliability and leads to multiple repeated reaction failures. Then, maintenance personnel can carry out detailed inspections one by one and also replace those components with higher reliability if it is necessary. Meanwhile, some auxiliary detection methods and equipments can be utilized for troubleshooting and maintenance [24]. According to FMEA, FMECA reports and an acceptable maintenance level of the enterprise, the investment in maintenance time and cost for each failure mode can also be analyzed and considered.

V. Conclusions and Future Work

This paper presents a model-based RCM analysis method. This method standardizes the traditional RCM analysis process, improves the operability and implementation rate of the RCM analysis, and provides support for the further development of RCM theory. The paper has implemented the first five steps of the traditional RCM analysis procedures to the proposed MRAF model by using AADL. After analyzing the model, the FMEA table, one of the significant basis for RCM analysis, has been obtained automatically. The FMECA table has been obtained semi-automatically by combining the generated FMEA with CA. The method is illustrated with an example system modeled in AADL. The hardware and software of the system have been modeled in the form of an AADL reliability model by using AADL architecture description and error model description in the OSATE platform. By analyzing the reliability model, the FMEA and FMECA tables are produced effectively. These tables are utilized to carry out further RCM decision analysis.

In the future work, on the one hand, by using the AADL's extension mechanism, an FMECA plug-in will be developed based on the OSATE platform to generate the FMECA automatically. On the other hand, by using AADL's extension mechanism, we will continue to extend our MRAF model to implement the rest work (step6 and step7 as shown in Fig. 1) in the RCM analysis procedure.

Acknowledgment

This work was supported by the Ningxia Key Research and Development Plan Project (No.2018BEE03019), the Natural Science Foundation of Ningxia Province of China (No.2019AAC03120), the 2020 Scientific Research Ability Promotion Program for Young Teachers from the Northwest Normal University (No.NWNU-LKQN2020-15), the Research start-up Project from the Northwest Normal University.

References

[1] N. Habidin, S. Hashim, N. Fuzi, and M. Salleh, Total productive maintenance, kaizen event, and performance. The International Journal of Quality & Reliability Management, Vol. 35(9), 2018, pp. 1853-1867.

[2] T. Hiruta, T. Uchida, S. Yuda, and Y. Umeda, A design method of the data analytics process for condition-based maintenance. CIRP Annals - Manufacturing Technology, Vol. 68(1), 2019.

[3] N. Yang, Research on preventive maintenance strategy for the task-based system. Beijing Jiaotong University, 2019.

[4] J. Liu, Reliability-centered maintenance implementation analysis and research in the chemical industry, Tsinghua University, 2016.

[5] Y. Wu, X. Jia, L. Wen, W. Song, and C. Guo, Review on the Development and Application of Reliability-centric Maintenance (RCM). Journal of Ordnance Engineering College, Vol. 28 (04), 2016, pp.13-21.

[6] C. Luo, Research on RCM-based Gantry Crane Maintenance Strategy. Southeast University, 2018.

[7] J. Moubray, Reliability-centered Maintenance (2nd Ed.). Oxford: Butterworth-Heinemann, UK. 1997.

[8] I. Afefy, Reliability-Centered Maintenance Methodology and Application: A Case Study. Engineering. Vol. 2(11), 2010, pp.863-873.

[9] C. Spreafico, D. Russo, and C. Rizzi, A state-of-the-art review of FMEA/FMECA including patents. Computer Science Review. 2017.

[10] B. Larson, J. Hatcliff, K. Fowler, and J. Delange, Illustrating the AADL error modeling annex (v.2) using a simple safety-critical medical device. 2013, 33(3).

[11] I. Scheeren and C. Pereira, Combining Model-Based Systems Engineering, Simulation and Domain Engineering in the Development of Industrial Automation Systems: Industrial Case Study. 2014 IEEE 17th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, Reno, NV, 2014, pp. 40-47.

[12] Y. Hu, R. Wang, X. Wang, and Y. Fu, Review on the development of model-based analysis technology for safety and reliability of complex systems. Journal of Aeronautics, 2019.

[13] P. Feiler. AADL and model-based engineering. ACM, 2014.

[14] CMU/SEI, Architecture Analysis & Design Language (AADL). http://www.aadl.info/. (accessed May.25.2019).

[15] P. Feiler and D. Gluch, Model-Based Engineering with AADL: An Introduction to the SAE Architecture Analysis & Design Language. 2013.

[16] A. Johnsen and K. Lundqvist, Developing Dependable Software-Intensive Systems: AADL vs. EAST-ADL. In Proceedings of 16th

Ada-Europe International Conference on Reliable Software Technologies. Reliable Software Technologies - Ada-Europe, Edinburgh UK, 2011.

[17] SAE, SAE-AS5506/1A, Architecture Analysis and Design Language (AADL) Annex Volume 1: Annex E: Error Model Annex. 2015.

[18] P. Feiler, Model-based validation of safety-critical embedded systems. In Proceedings of Aerospace Conference. IEEE, 2010.

[19] P. Feiler, Architecture Analysis and Design Language(AADL) Annex Volume 3: Annex E: Error Model V2 Annex. Number SAE AS5506/3 (Draft) in SAE Aerospace Standard, 2013.

[20] J. Delange, P. Feiler, D. Gluch and J. Hudak, AADL fault modeling and analysis within an ARP4761 safety assessment. 2014.

[21] Y. Li, L. Nan, and X. Long, Discussion on Reliability Modeling of Embedded System Based on AADL. Computer Technology and Development. 2015, pp.234-236.

[22] CMU/SEI, Open Source AADL Tool Environment (OSATE). http://osate.org/. (accessed May.25.2019).

[23] L. Wu, Y. Yan, F. Gao, X. Chen, and C. Nie, Research on Modeling and Verification Methods for Embedded Software Systems Based on SysML. 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), Sofia, Bulgaria, 2019, pp. 150-157

[24] Q. Luo, RCM-based equipment maintenance decision-making method and its application research. Zhejiang University of Technology, 2016.

[25] S. Sharvia, S. Kabir, M. Walker, and Y. Papadopoulos. Model-based dependability analysis: State-of-the-art, challenges, and future outlook. in Software Quality Assurance: Elsevier, 2016, pp. 251-278.

[26] P. Wang, C. Zhao, and F. Yan, Research on the Reliability Analysis of the Integrated Modular Avionics System Based on the AADL Error Model, International Journal of Aerospace Engineering, 2018.

[27] IEC 60812, Analysis techniques for system reliability-Procedure for failure mode and effects analysis (FMEA). IEC (Intern. Elect. Commission), 2006.

[28] P. Feiler and others, examples / SafetyTutorial. https: //github.com/osate/examples/tree/master/SafetyTutorial. (accessed May.25.2019).

[29] B. Gu, Y. Dong, and X. Wei, A Qualitative Safety Analysis Method for AADL Model. 2014 IEEE Eighth International Conference on Software Security and Reliability-Companion, San Francisco, CA, 2014, pp. 213-217.