ATLANTIS PRESS
Advances in Computer Science Research, volume 63
International Conference on Artificial Intelligence and Engineering Applications (AIEA 2016)

Safety Analysis Method for COTS Software Components in Train Control System

Jiancheng Mu[1, a], Dongmei Huang[1, b], Lianchuan Ma[1, 2, c] and Yuan Cao[1, 2, d]

[1]School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China;

[2]National Engineering Research Center of Railway Transportation Train Control System, Beijing Jiaotong University, Beijing 100044, China.

[a]jchmu@bjtu.edu.cn, [b]14120246@bjtu.edu.cn, [c]lchma@bjtu.edu.cn, [d]ychao@bjtu.edu.cn

Abstract. Commercial off-the-shelf (COTS) software and hardware components are widely used in the design of train control system. In order to satisfy the application requirements of the safety computer in train control system, it is necessary to analyze its safety properties. In this paper, a method of safety analysis for the safety computer is proposed. The safety properties of the safety computer in train control system are verified by establishing the system model of safety mechanism, and establishing a safety base in safety computer management units (SCMU), and measuring the safety of each part of the system step by step, and then establishing a safety chain. Finally, tests are carried out through a designed software fault injection tool to demonstrate the effectiveness of the proposed method.

Keywords: COTS; Safety Computer; Safety Mechanism; Safety Base; Safety Chain.

1. Introduction

As a safety critical system [1], railway transportation train control system has been widely concerned. Safety computer is an important part of the train control system, so it is required for high reliability and safety properties. The generally accepted and adopted design method of safety computer is to use commercial off-the-shelf (COTS) hardware and software components to integrate the system. Compared to the traditional design methods using special safety components and chips, this method can shorten the development cycle and reduce the cost [2].

As COTS products is a common, standard and open design in concept, it is rarely considered the special needs of safety critical areas. For commercial interests, COTS products manufacturers usually do not provide detailed design information about COTS software and hardware. In the safety analysis of the safety computer, which consists of COTS software and hardware components, the COTS software and hardware components can only be regarded as black boxes [3], which makes the safety analysis face a huge problem.

In this paper, a safety analysis method is proposed, which is suitable for the analysis of safety computer in train control system, referring to the trusted computing [4-6] in the field of information security. By ensuring the certainty and integrity of the behavior of the safety computer entities, the safety of the system is validated.

2. Model of the Safety Operation Mechanism

The safety computer contains two lines of the same host computers, which consists of three functional domains, including the general computational domain (GCD), the safety computer management domain (SCMD) and the safety input and output domain (SIOD).

The SIOD is designed based on the safety mechanism of the hardware. The GCD is implemented based on the general COTS software and hardware. As a logical computing coprocessor, the safety properties of the GCD are determined by the SCMD, as shown in Figure 1.

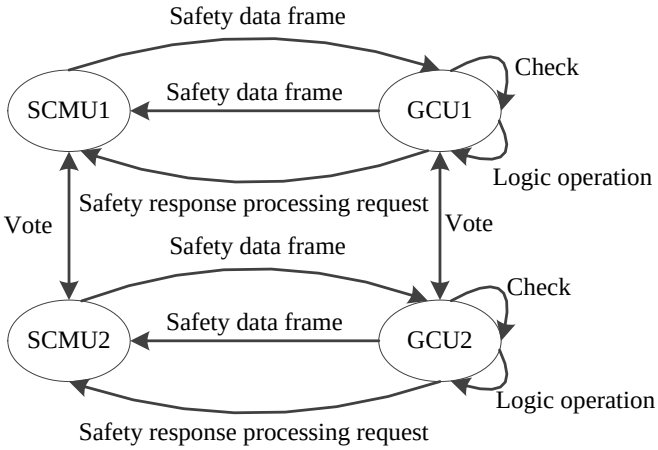

Fig. 1 The safety operation mechanism between SCMD and GCD

The SCMD sends safety data frames to the two GCUs according to the secure communication protocol. Then the two GCUs will carry on the safety decoding, and check whether the received data frames are legal. If the safety data frame is legal, the logic operation is performed; otherwise, the GCD returns the data frame with "safety response processing request" to the SCMD. After the operation, the two GCUs vote the results through the COTS Ethernet, if the vote results are consistent, the logical operations are proved correct, then the respective operation results are composed of safety data frames and returned to the SCMD to be handled. Otherwise, the "safety response processing request" is sent to the SCMD to request the safety response.

3. Safety Analysis Method for Safety Computer

3.1 Safety Base

The Safety base includes a safety measurement base, a safety storage base and a safety report base as shown in Figure 2. The safety base contains a configuration register for storing safety measurement values which can not only represent the safety of the current program entities but also represent the initial sequence of the system. In addition, the log technology is used to record the events in the process of safety measurement (including the content of the measurement, the timing of the measurement and the abnormal situation) to provide a basis for the system's fault detection and maintenance.

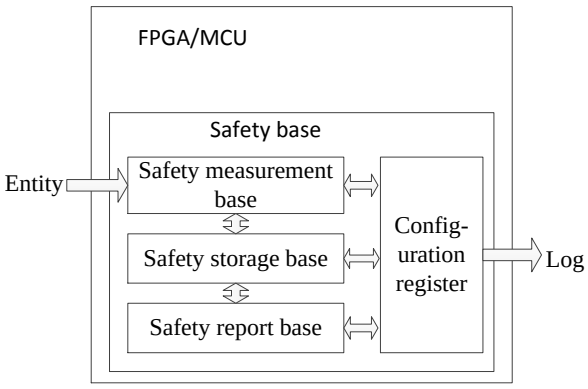

Fig. 2 The safety base

The working mechanism of the safety base is as follows:

(1) When the system starts, the safety base measures the initial sequence, records the initial sequence and the safety measurement result to the configuration register, and form the system startup log.

(2) In the process of the system operation, the safety base measures each part of the system, reports to the SCMD for processing once the danger is found, and form the log of the dangerous events during the measurement process.

3.2 Safety Measurement

The object of the measurement is the current executed program codes called the entity. This paper mainly checks the Hash value of the data to measure the safety of the data due to the safety of the various parts of the system is not easy to measure directly. An abstract value is calculated for the program entity that is about to be executed, and is saved in the configuration register. The abstract value is a hash value using the hash algorithm SHA-1 calculation, and it can represent a particular system state.

Cascading the existing value and the new measurement value to generate a calculated hash value as the new safety measurement value, and the measurement rule is found in the formula (1).

$$\text{New } PCR_{i+1} = \text{HASH}(\text{Old } PCR_i \parallel \text{New Value}) \quad (1)$$

All predictable program entities are considered as safety evidence defined as a safety record (SR) in the safety storage base. If an entity belongs to the SR, it is determined to be safe; if not, it is determined to be dangerous.

3.3 Construction of the Safety Chain

The starting point of the safety chain is the safety base, and the safety computer system is divided into several operation layers in hierarchy of hardware and software. When the measurements are performed from the low level nodes to the high level nodes, the high level nodes are proved to be in a safe state, and the safety of the nodes can be extended from the low levels to the high levels. In the safety chain structure, the safety base is established in the SCMU as the safety detection unit and the controller of the whole system as shown in Figure 3.

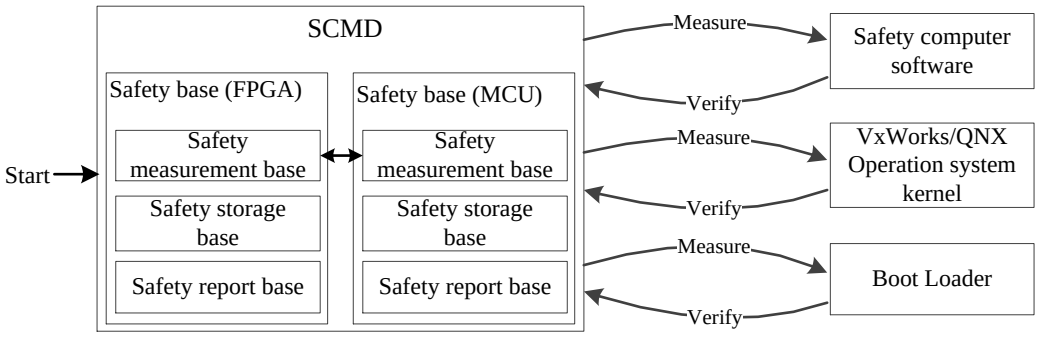

Fig. 3 The safety chain

The Safety base in the SCMD completely achieves the safety measurement base, the safety storage base and the safety report base. Firstly, we measure the boot loader, then the VxWorks and QNX real-time operation system kernel, and finally the software running in the operating system. The safety base verifies each part of the system. Only entities that measured and verified to be safe can be executed in the safety computer.

4. Test Result

This paper adopts the software fault injection technology [7] [8] to test the safety computer. We choose a small SHELL program with independent functions as the target system to perform fault injection. The times of the target system runs depends on the number of fault models in the fault injection script, for each fault model will cause the target program to run once. We carried out 4 quantization experiments, and summarized a large number of experimental test results, and got the response of the safety computer system when the fault injection tests were performed. The results are shown in Table 1.

Table 1 Fault injection test results

Number	Test times	Successful detection	Probability of success/%
1	30	29	96.7
2	70	68	97.1
3	120	116	96.7
4	200	194	97

The test results show that in variable experiments, safety base in the safety computer system can find faults timely by checking the hash value, and make corresponding response to the faults. That is, once a fault is found, the system can immediately start the safety response reaction. Through the method of safety chain, it can be effective to verify the safety properties of the system.

5. Summary

Based on the establishment of the system's safety operation mechanism model, a safety analysis method for safety computer is proposed. Through the construction of the safety chain, the safety is extended to all parts of the system, and can effectively ensure the safety of the whole system. For the COTS components in the system, it is not necessary to know the composition of its internal hardware or codes. Therefore, this method is applicable to the safety analysis of the safety computer with COTS components.

Acknowledgements

In this paper, the research was sponsored by the Beijing Laboratory of Urban Railway Transportation.

References

[1] Wu Jian, Xu Zhongwei, Yu Gang. Reliability Analysis of Safety Critical System Based on Improved DFTA [J]. Computer Engineering, 2009, 15: 117-120.

[2] M Morisio, N Sunderhaft. Commercial-Off-The-Shelf (COTS): A Survey[R]. Data & Analysis Center for Software.2000.

[3] M Nicolaidis. Fail-Safe Interfaces for VLSI: Theoretical Foundations and Implementation [J]. IEEE Transactions on Computers, 1998, 47(1):62-77.

[4] Ye Chaolong, Yang Zhigang, Tan Tonghe. Complete Trusted Chain Model of Mobile Terminal Based on Trusted Computing [J]. National Conference of Information Security Level Protection Technology, 2014.

[5] Chen Shuyi, Wen Yingyou, Zhao Hong. Design of a Mobile Platform Based on Trusted Computing [J]. Northeastern University Journal of Natural Sciences, 2008, 29(8):1096—1099.

[6] Yang Bei, Hao Zhenqiang, Fu Xiangping. Dynamic Integrity Measurement Model Based on Trusted Computing [J]. Computer Engineering,2012, 38(2): 78—81.

[7] J Arlat, Y Crouzet, J Karlsson. Comparison of Physical and Software-Implemented Fault Injection Techniques [J]. IEEE Transactions on Computers, 2003, 52(9): 1115-1133.

[8] Barenghi A, Breveglieri L, Koren I, et al. Fault Injection Attacks on Cryptographic Devices: Theory, Practice, and Countermeasures [J]. Proceedings of the IEEE, 2012, 100(11):3056-3076.