Transactions of the Korean Nuclear Society Autumn Meeting
Gyeongju, Korea, October 24-25, 2013

Evaluation of Fault Detection Coverage of Digital I&C Systems

Seung Jun Lee, Wondea Jung
Korea Atomic Energy Research Institute
1045 Daedeok-daero, Yuseong-gu, Daejeon, 305-353, Korea
sjlee@kaeri.re.kr

1. Introduction

One of the important features of digital instrumentation and control (I&C) systems is the fault-tolerant technique. Fault-tolerance is the system's capability to help the system perform correctly the specific required functions in spite of the presence of faults. In the fault tolerance evaluation, fault detection coverage is a crucial factor. The fault detection coverage is the ability to detect errors that are caused by faults in a system. If faults are not detected by a certain detection algorithm, the system could be in failure. Evaluating the fault detection coverage of the fault-tolerant technique is important for the safety analysis of digital systems [1].

Digital I&C systems have more various fault-tolerant techniques than conventional analog I&C systems. Even though these fault-tolerant techniques are designed to ensure and improve the safety of systems, the effects of them have not been properly considered yet in most system probabilistic safety assessment (PSA) models. There have been several researches into the reliability of digital systems [1]-[4]. However, systematical frameworks or reasonable models to obtain the reliability of digital systems by considering the effects of fault-tolerant techniques have not been proposed. Therefore, it is necessary to develop an evaluation method reflecting the features of digital I&C systems.

2. Fault-Tolerant Techniques in Digital I&C Systems

A fault-tolerant technique cannot detect all possible faults in a system but detect some faults in a certain range. In digital I&C systems, therefore, multiple fault-tolerant techniques are implemented simultaneously at each level of the system's hierarchy for higher system reliability such as component-level fault detection algorithms (e.g., memory checksum, watchdog timer for detecting microprocessor halt), board-level self-diagnostics (e.g., loop back check for input and output module), and system-level error detection mechanisms (e.g., automatic periodic test, state comparison algorithm of redundant modules). Each fault-tolerant technique has different inspection period from real-time monitoring to monthly testing. The range covered by each fault-tolerant technique is also different. A fault occurred in a system might be detected by one or more fault-tolerant techniques. Even though the fault is not detected by the fault-tolerant technique implemented in lower level of system, it could be detected by higher level fault-tolerant technique in the system.

To evaluate the fault detection coverage of a system, different inspection range and period of each fault-tolerant technique should be considered and duplicated effect of fault-tolerant techniques caused by multiple fault-tolerant techniques should be eliminated [1].

3. Fault Detection Coverage Quantification by Fault Injection Experiments

For identifying the exact fault detection coverage, it is the best way to simulate all the possible faults physically by hardware implemented fault injections. However, it is difficult to simulate all the faults using hardware-implemented fault injection techniques because it requires expensive hardware and some faults cannot be controlled and limited in the complexity of the system [5]. Therefore, the limited hardware-implemented fault injection technique, in which faults can be injected only to memory and register, was used. The fault injection experiment in this work is based on the assumption that all faults in a system are reflected on the faults in memory and register because a fault should affect the memory or register related to the output variables in order to generate a wrong output.

For more realistic evaluation, the prototype of digital I&C systems (ATIP: Automatic Test Interface Processor, BP: Bistable Processor) which are actually adopted in a digitalized nuclear power plant were used for the fault injection experiment [6]. The fault injection experiment for fault detection coverage quantification was performed based on the following three steps.

- Fault type identification
- Memory map development
- Fault injection experiment

3.1 Fault Type Identification

A fault in a digital I&C system is categorized into seven types according to its effect, as shown in Table I.

Table I. Fault types

	Changed and Used			Unused or unchanged
	Correct output	Wrong output	No output	
Detected	O	O	O	O
Undetected	O	X	X	

'O' indicates the faults which do not cause unsafe status. If a memory bit is not affected by a fault or is not used, then there will be no effect on the system. If a correct output is generated in spite of the fault, then the system will work correctly. Even though an incorrect output is generated by the fault, the system will result in safe status if the fault is detected. The faults represented by 'X' are dangerous faults which cause abnormal behavior of a system. When there is an undetected fault causing wrong or no output of a system, the system might work abnormally.

3.2 Memory Map Development

Because a bit is binary, two kinds of faults (stuck-at-0 and stuck-at-1) can be injected. The fault injection experiment for every single bit requires much time because of huge memory bits. For example, totally about two million experiments are necessary for the only memory of BP operating system (OS) code. In order to reduce the number of fault injection experiments, the memory area for BP was analyzed and a memory use profile was developed before the experiment. From the analysis, followings were observed.

- The memory area assigned to BP was not fully used. The analysis result showed that 52% of the assigned memory was practically used and 48% was unused. The faults in the unused memory area do not have any effects, so that fault injections on this area are meaningless.
- One assembler line consists of 32 bits. Depending on the operator (mnemonic) of an assembler line, necessary bits vary from 8 to 32 bits. This means that some area in the used memory does not have any meaning.
- It was observed that 64% have 0 bit, 23% have 1 bit, and 13% have no meaning in the used memory area. While the memory for variables is continuously changed, the memory of BP code is not changed after loaded. Because stuck-at-0 faults do not change the bits which are already 0 and stuck-at-1 faults do not change the bits with 1 in the BP code memory area, we can identify the memory area which is not affected by injected faults without experiments.

3.3 Fault Injection Experiment

From the fault injection experiment, the probability of each fault type can be estimated and the fault detection coverage can be evaluated.

Three fault-tolerant techniques were considered: OSD (On-line Status Diagnostics), CSD (Component Self Diagnostics), and APT (Automatic Periodic Test). In this work, only 2 bits in each assembler line (the first and last bits) was examined for the used memory area, as a feasibility study. As shown in Fig. 1, the fault detection coverage of each fault-tolerant technique and whole fault detection coverage can be defined.

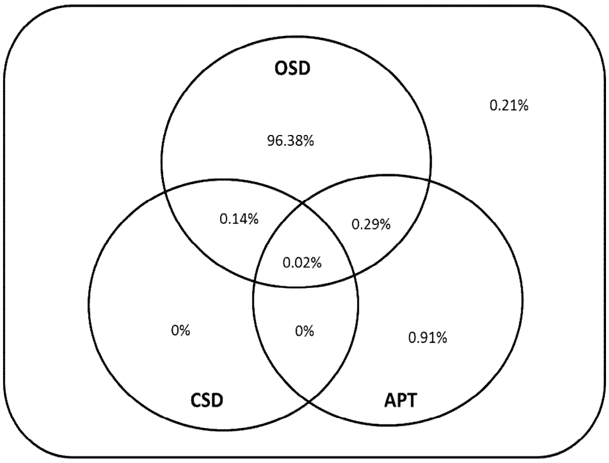

Fig. 1. Fault detection coverage

4. Conclusions

The evaluation method for fault detection coverage of digital I&C systems was proposed in this work. The proposed method quantifies the fault detection coverage based on the fault injection experiment. Even though there are several limitations of the fault injection experiment such as fault injection into only memory and register, the method has an advantage of that it is possible to observe the actual system behavior against faults in the system. More accurate system reliability evaluation of digital I&C systems can be expected through the experiment result.

REFERENCES

[1] S. J. Lee, J. G. Choi, H. G. Kang, S. C. Jang, Reliability Assessment Method for NPP digital I&C Systems considering the Effect of Automatic Periodic Tests, Annals of Nuclear Energy, Vol. 37, p. 1527-1533, 2010.

[2] H. G. Kang, M. C. Kim, S. J. Lee, H. J. Lee, H. S. Eom, J. G. Choi, and S. C. Jang, An overview of risk quantification issues of digitalized nuclear power plants using static fault tree, Nuclear Engineering and Technology, Vol. 41, p. 849-858, 2009.

[3] H. G. Kang and T. Sung, A quantitative study on important factors of the PSA of safety-critical digital systems, Nuclear Engineering and Technology, Vol. 33, p. 596-604, 2001.

[4] H. G. Kang, T. Sung, An analysis of safety-critical digital systems for risk-informed design, Reliability Engineering and System Safety, Vol. 78, p. 307-14, 2002.

[5] S. J. Kim, P. H. Seong, J. S. Lee, M. C. Kim, H. G. Kang, and S. C. Jang, A method for evaluating fault coverage using simulated fault injection for digitalized systems in nuclear power plants, Reliability Engineering and System Safety, Vol. 91, p. 614-623, 2006.

[6] S. Hur, and et al., 2007. The automatic test features of the IDiPS reactor protection system. KNS spring conference, 2007, Korea.