A Study on the Test Method of Software Verification & Validation of Nuclear Power Plant

Min-seok Kim [a*], Hosun Ryu [a], Dongil Lee [a]
[a]KHNP, Central Research Institute, 70, 1312-gil, Yuseong-daero, Yuseong-gu, Daejeon, 34101, South Korea
[*]Corresponding author: minseok2@khnp.co.kr

1. Introduction

Since digital control equipment has hardware, software, communication, control and diagnostic functions, it is difficult to guarantee reliability by functional testing alone. Therefore, it is necessary to develop and reflect a method to test the performance of the digital system, and it is necessary to develop the technology to cope with the expected regulation change. In particular, software requires procedures to prevent initial failure through sufficient verification, such as software V & V, before applying to digital control equipment.

Therefore, KHNP is carrying out the project of 'Development of digital I&C equipment performance validation technology & infrastructure' to secure the performance verification technology of digital control equipment. This paper describes the automatic test case derivation of Function Block Diagram(FBD) software and the automatic test method using performance verification equipment[1].

2. Test Method

We have developed FBD Editor with the ability to automatically generate test cases and case sequences. We used this program to automatically generate test cases for the FBD of the plant control system. Then, we test the generated test cases automatically using the performance verification equipment.

2.1 Automated Test Case Generator

FBD is one of the programming languages for the Programmable Logic Controller defined in the IEC 61131-3 standard. The use of FBD programs to implement safety system such as reactor protection system has increased the importance of testing for FBD programs. Thus, research has been carried out to automatically generate test cases for FBD programming [2, 3].

Table I: Test Coverage Criteria

Basic Coverage (BC)	Test all data path in the FBD at least once
Input Condition Coverage (ICC)	Test both True and False for all binary input edges in the FBD
Complex Condition Coverage (CCC)	Test both True and False for all binary input and output edges in the FBD

In this project, the basic concept is to define the data path which is the connection of edges from input to output, and to cover the data paths existing in the FBD program more than once, using the FBD program itself as a test model. We defined the basic coverage (BC), the input condition coverage (ICC), and the complex condition coverage (CCC), which are test coverage criteria of the FBD program, and used the method of automatically generating test cases based on these coverage.

2.2 Generation of Test Case

Figure 1 shows one of the FBDs of the plant control system created by the FBD Editor. Test cases were generated to satisfy the complex condition coverage for [STRT COM] and [STP COM], which are FBD outputs.

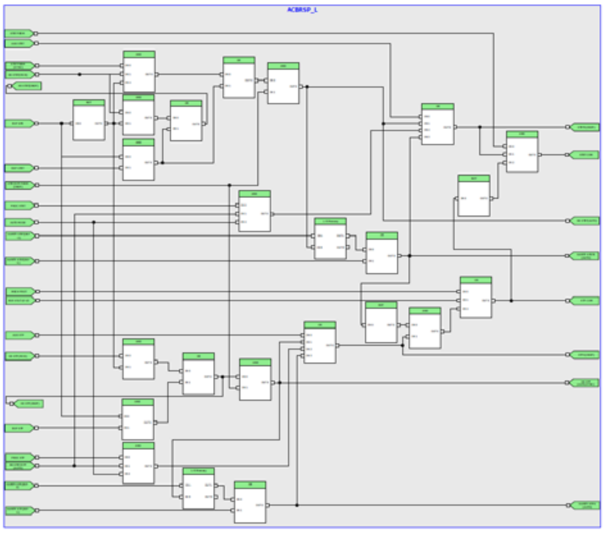

Fig. 1. Screen Capture of the FBD Editor

As a result of generating a test case that satisfies the complex condition coverage for [STRT COM] output, 104 of the total 106 test requirements were satisfied, and the overall coverage achieved 98%. There are four test cases required for this purpose.

This means that both the TURE / FALSE of the binary input condition of all data paths affecting the generation of [STRT COM] output in the FBD and the number of cases that satisfy TRUE / FALSE of all binary outputs in the corresponding data path are 106. In addition, it is possible to create test cases satisfying 104 of these conditions, which means that four test case sets are required.

Table II: Test Coverage Ratio

Output	Test Requirements	CCC Coverage
STRT COM	104/106	98%
STP COM	56/58	96%

2.3 Verification of FBD Program

Figure 2 shows the experimental setup for verifying the FBD program. The left side of figure 2 shows the real control facility where the FBD program to be verified is installed, and the right side is a performance verification equipment designed to automatically HILS (Hardware In the Loop Simulation) the generated test case.

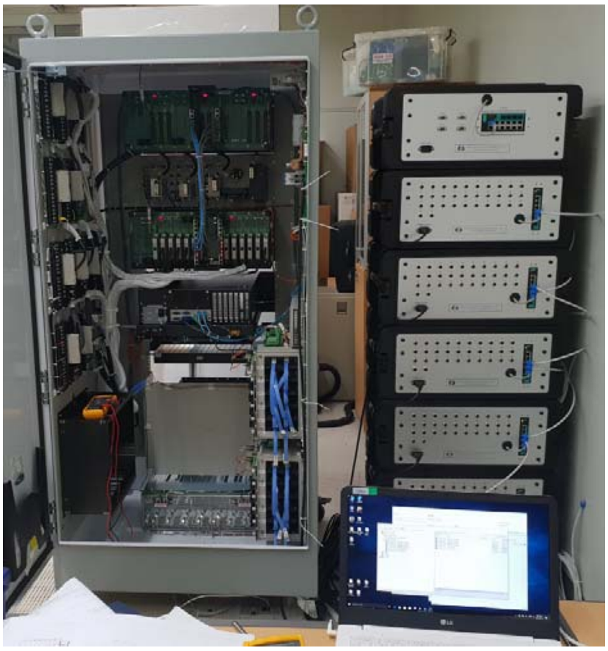

Fig. 2. The verification target equipment (left) and the performance verification equipment (right)

As a result of testing the test case with the performance verification equipment, it was confirmed that the output value is the same as the expected output.

3. Conclusions

We have developed a program that automatically generates structural test cases from FBD, the PLC design language, and applied them to real systems. A test case generated automatically through FBD can yield a test case for all possible paths in the software.

However, when the Timer Block is applied, it is necessary to test over several scan cycles to verify 100% of the logic. If the maximum delay time of the Timer Block is long, the required test cycle increases.

Also, in the case of a function block having an internal memory state, sequence testing is required to additionally generate a condition for the internal memory state to reach a specific value. At this time, a sequence test case is generated while unwinding the scan cycle of the FBD. Even if the test cycle increases infinitely, the internal memory state value may not be specified. This is because it is impossible to set the initial value of the internal memory state in the PLC. Therefore, additional research is needed to solve the problem.

However, structural testing through FBD can reduce human errors by creating test cases in a systematic and quantitative way based on a logic structure. It is also possible to test all the applied logic in addition to the specified functions.

In software V & V, the requirement for structural coverage verification is increasing. Therefore, if the automatic test case generation program and the automatic test method using the performance verification equipment are used, the utilization in the software V & V will be improved in the future.

REFERENCES

[1] H. Ryu, H. Kim, J. Kim and K. Lee, “Development of Control Verification Simulator for the Controller”, CICS’18, pp. 259, 2018.

[2] E. Jee, D. Shin, S. Cha, J.-S. Lee and D.-H. Bae, “Automated test case generation for FBD programs implementing reactor protection system software”, Software Testing, Verification and Reliability, Vol. 24, No. 8, pp. 608-628, 2014.

[3] J. Song, E. Jee and D.-H. Bae, “FBD Tester 2.0: Automated test sequence generation for FBD programs with internal memory states”, Science of Computer Programming, Vol. 163, No. 1, pp. 115-137, 2018.