Transactions of the Korean Nuclear Society Spring Meeting
Jeju, Korea, May 22, 2009

A Study on the Test Coverage for Software Validation in Nuclear I&C System

Hyung Tae Kim, Young Doo Kang, Choong Heui Jeong, Dae Il Kim
Instrumentation & Control Division, Korea Institute of Nuclear Safety, 19 Guseong-dong, Yuseong, Daejeon, Korea,
kshape@kins.re.kr, k407kyd@kins.re.kr, k148jch@kins.re.kr, dikim@kins.re.kr

1. Introduction

Digital technology has proliferated in several commercial and safety-critical application areas. However, digital systems are also seen as having issues associated with their use in nuclear power plants (NPPs). Due to increased system complexity, it reduces the likelihood that the system can be exhaustively tested, therefore making it difficult to prove the system safety. These issues include software validation. It is difficult to asses how much every software feature is covered by a test case when testing during software development. Therefore we need a means of test coverage to verify software and support demonstration of absence of unintended functions. The objective of this study is to propose a guideline of test coverage in NPPs.

2. Test Coverage

Test Coverage is a measure of how much testing can be done given a set of test cases [5]. Test coverage is any measure of completeness with respect to either a requirement (requirement-based) or the code's design / implementation criterion (code-based), such as the verification of use cases (requirement-based) or execution of all lines of code (code-based). Basic measures of test (or code) coverage are as follows [1,3,4,5]:

- Statement Coverage
- Decision Coverage (Branch Coverage)
- Modified Condition/Decision Coverage (MC/DC)

In this section, code coverage and coverage analysis will be introduced.

2.1 Statement Coverage

To achieve statement coverage, every executable statement in the program is invoked at least once during software testing. Achieving statement coverage shows that all code statements are reachable.

2.2 Branch Coverage

Branch coverage reports whether boolean expressions in control structures are evaluated to both true and false values by the test cases. This coverage is also known as decision coverage.

2.3 Modified Condition/Decision Coverage (MC/DC)

The MC/DC criterion enhances the condition/ decision coverage criterion by requiring that each condition be shown to independently affect the outcome of the decision [5]. The independence requirement ensures that the effect of each condition is tested relative to the other conditions. However, achieving MC/DC requires more thoughtful selection of the test cases and, in general, a minimum of *n*+1 test cases for a decision with *n* inputs.

2.4 Coverage Analysis

Generally, in order to provide evidence that software was verified to the degree required for the applicable software level, requirements coverage analysis and structural coverage analysis are performed [4,5]. Requirement coverage analysis determines how well the requirements-based testing verified the implementation of the software requirements, and establishes traceability between the software requirements and the test cases. Structural coverage analysis determines how much of the code structure was executed by the requirements-based tests, and establishes traceability between the code structure and test case.

3. Analysis of Related Standards

In this section, IEEE Std. 1008, IEC 60880 and DO-178B will be analyzed focusing on test coverage.

3.1 IEEE Std. 1008-1997

IEEE Std. 1008 defines an integrated approach to systematic and documented unit testing. The approach uses unit design and unit implementation information, in addition to unit requirements, to determine the completeness of the testing [1]. This standard describes test coverage requirements criteria in the following.

Test plan should be documented including the areas to be covered by the unit test set and the degree of coverage required for each area. When testing a unit during software development, every software feature must be covered by a test case or an approved exception. When testing a unit implemented with a procedural language during software development, every instruction that can be reached and executed must be covered by a test case or an approved exception. This means that statement coverage should be reached.

In annex A9 and A10, stronger code-based requirements and code coverage tools was addressed. Based on the criticality of the unit or a shortage of unit requirement and design information, the code-based coverage requirement could be strengthened. One option is to strengthen the requirement from instruction coverage to branch coverage. An automated means of recording the coverage of source code during unit test execution is highly recommended.

3.2 IEC 60880-2006

IEC 60880 describes test coverage requirements criteria as follows [3]. The tests performed should extensively exercise the software. Among the criteria required in the plan, test coverage criteria should be considered of prime importance. The verification plan shall identify any objective evidence required to confirm the extent of testing. For that purpose, the test coverage criteria chosen according to the design (see Annex E) shall be justified and documented. In Annex E.4.2.2 Path Testing, coverage criteria such as statement, branch and path coverage, etc., were addressed.

3.3 RTCA/DO-178B

The RTCA/DO-178B is the primary means used by aviation software developers to obtain Federal Aviation Administration (FAA) approval of airborne computer software [4]. The objectives of the software life cycle processes applicable to a given piece of software are based on the software level determined by the system safety assessment. To accommodate different criticality environments, DO-178B created five software levels (A, B, C, D, E) which are based on the potential of the software to cause safety-related failures identified in the system safety assessment. For software level A, it describes that requirement, statement, branch, MC/DC coverage should be achieved.

4. Case Study of Test Coverage in Korea Nuclear I&C System

In this section, we review Core Protection Calculation System (CPCS) for SHIN-KORI 1,2, and POSAFE-Q PLC software for nuclear I&C system.

4.1 CPCS for SHIN-KORI 1,2

In CPCS software for SHIN-KORI 1,2, module test cases were developed based on inputs calculated to exercise the branches in the CPCS C code. An automated test coverage tool, LDRA, was used to demonstrate branch coverage. Requirement coverage analysis was performed by Requirement Traceability Matrix (RTM). CPCS software achieved requirement, statement and branch coverage.

4.2 RTOS of POSAFE-Q PLC

RTOS embedded in safety grade PLC, POSAFE-Q, was developed by the Korea Nuclear I&C System (KNICS) project. RTOS software achieved requirement coverage and branch coverage. It was verified by RTM and an automated tool.

5. Regulatory Position on Test Coverage of Nuclear I&C Systems

Basically, requirement coverage should be achieved regardless of software level. Also, test (or code) coverage should be applicable to software in nuclear I&C systems. In addition to requirement coverage, code coverage requirement should be achieved according to software level. Because studies on MC/DC provided the rationale of applicability of it, we will adopt MC/DC coverage [4,5,6]. Table I shows the proposed test coverage requirements according to software level [2]. In order to provide evidence that software was verified to the degree required for the applicable software level, requirements coverage analysis and structural coverage analysis should be performed.

Table I: Test Coverage Requirements

S/W Level	Test Coverage Requirements
Safety Critical	MC/DC coverage Branch coverage Statement coverage Requirement coverage
IEEE 1012 SIL4	
Safety Related	Branch coverage Statement coverage Requirement coverage
IEEE 1012 SIL3, SIL2	
Non-Safety	Requirement coverage
IEEE 1012 SIL1	

6. Conclusions

IEEE Std. 1008, IEC 60880 and DO-178B recommended that test coverage of software be achieved and its analysis be performed according to software level. Basically, requirement coverage should be achieved and additional code-based coverage requirements can be required according to software level. In addition, requirements coverage analysis and structural coverage analysis should be performed. In future works, we will develop a detailed guideline of test coverage in NPPs.

REFERENCES

[1] IEEE Std.1008-1997, “Software Unit Testing”.
[2] IEEE Std.1012-2004, “Software Verification and Validation” .
[3] IEC 60880-2006, “Software aspects for computer-based systems performing category A functions”.
[4] RTCA/DO-178B, “Software considerations in airborne systems and equipment certification”.
[5] NASA/TM-2001-210876, “A Practical Tutorial on Modified Condition/Decision Coverage”.
[6] Dupuy, A. and Leveson, N. “An empirical evaluation of the MC/DC coverage criterion on the HETE-2 satellite software”, Proceedings of the Digital Aviation Systems Conference (DASC), Philadelphia, USA, October 2000.