Nuclear Engineering and Technology 50 (2018) 570–581

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original Article

Development of simulation-based testing environment for safety-critical software

Sang Hun Lee [a], Seung Jun Lee [b], Jinkyun Park [c], Eun-chan Lee [d], Hyun Gook Kang [a, *]

[a] Department of Mechanical Aerospace and Nuclear Engineering, Rensselaer Polytechnic Institute (RPI), 110 8th Street, Troy, NY, 12180, USA
[b] School of Mechanical, Aerospace and Nuclear Engineering, Ulsan National Institute of Science and Technology (UNIST), 50 UNIST-gil, Ulsan, 44919, Republic of Korea
[c] Integrated Safety Assessment Division, Korea Atomic Energy Research Institute (KAERI), 111 Daedeok-daero, 989beon-gil, Yuseong-gu, Daejeon, 34057, Republic of Korea
[d] Korea Hydro & Nuclear Power Co., Ltd., 1655 Bulguk-ro, Gyeongju-si, Gyeongsangbuk-do, 38120, Republic of Korea

ARTICLE INFO

Article history:
Received 30 January 2018
Received in revised form
28 February 2018
Accepted 28 February 2018
Available online 27 March 2018

Keywords:
Digital Instrumentation and Control System
Nuclear Power Plant
Software Reliability Quantification
Software Testing

ABSTRACT

Recently, a software program has been used in nuclear power plants (NPPs) to digitalize many instrumentation and control systems. To guarantee NPP safety, the reliability of the software used in safety-critical instrumentation and control systems must be quantified and verified with proper test cases and test environment. In this study, a software testing method using a simulation-based software test bed is proposed. The test bed is developed by emulating the microprocessor architecture of the programmable logic controller used in NPP safety-critical applications and capturing its behavior at each machine instruction. The effectiveness of the proposed method is demonstrated via a case study. To represent the possible states of software input and the internal variables that contribute to generating a dedicated safety signal, the software test cases are developed in consideration of the digital characteristics of the target system and the plant dynamics. The method provides a practical way to conduct exhaustive software testing, which can prove the software to be error free and minimize the uncertainty in software reliability quantification. Compared with existing testing methods, it can effectively reduce the software testing effort by emulating the programmable logic controller behavior at the machine level.

1. Introduction

With a shift in technology to digital systems as analog systems are approaching obsolescence and because of functional advantages of digital systems, existing nuclear power plants (NPPs) have begun to replace analog instrumentation and control (I&C) systems, while new plant designs fully incorporate digital systems [1]. Compared with the analog I&C systems, the digital systems provide advanced performance in terms of accuracy and computational capabilities and have potential for improved capabilities such as fault tolerance and diagnostics [2]. However, the use of microprocessor-based digital systems in NPP safety I&C systems has triggered a big challenge in incorporating their characteristics into the probabilistic risk assessment (PRA) model of NPPs used to evaluate the digital system reliability and its risk effect on the NPP safety.

A comprehensive review of the risk issues of digital I&C systems that should be considered in the NPP PRA model has been conducted by Kang and Sung [3]. Among various issues, estimation of the software failure probability was identified as one of the important factors in terms of NPP risk, and a sensitivity study was conducted to analyze the relationship between the system reliability and the software failure probability for a typical digital reactor protection system (RPS). A report on operation and maintenance experience described how software error was a major cause of digital system failures during 1990–1993 [4]; during this time, 30 failures were caused by software error, compared with nine random component failures, among a total of 79 digital system failure events. Several reports also stated the importance of software-based errors, which are considered to be a credible source of the common-mode or common-cause failure of the digital systems [5,6], that can lead to significant safety threats of NPPs. Therefore, quantification of software reliability plays a very

* Corresponding author.
E-mail addresses: lees35@rpi.edu (S.H. Lee), sjlee420@unist.ac.kr (S.J. Lee), kshpjk@kaeri.re.kr (J. Park), eclee@khnp.co.kr (E.-c. Lee), kangh6@rpi.edu (H.G. Kang).

https://doi.org/10.1016/j.net.2018.02.007

important role in ensuring the safety of NPPs, and the verification of a very low software failure probability is crucial for the PRA of a digitalized NPP.

In response, quantitative software reliability methods such as the software reliability growth model (SRGM), Bayesian belief network (BBN) model, and test-based method have been proposed and adopted in the nuclear field. The SRGM method [7] has been widely used in the software engineering field to assess software reliability by estimating the increment of reliability as a result of fault removal over time. By applying a software reliability model and using existing software failure data to estimate its parameters, the software reliability is assessed and predicted based on extrapolation. However, the SRGM method was found to be not applicable to safety-critical software [8] because of its high sensitivity in estimating the number of faults to time-to-failure data and the rare software failure sets in NPP safety-critical applications that are developed under a strict development and verification and validation life cycle.

The BBN method has also been extensively applied to estimate the software reliability of NPP safety systems [9,10]. The method models and aggregates disparate information about the software, such as software failure data and the quality of software life cycle activities. However, the limitations of the BBN method in quantifying the software reliability include the need to develop a credible BBN model, which requires identification of a complete and independent set of software attributes and the qualification of experts to estimate model parameters and qualitative evidence. Owing to those limitations, the uncertainty in the estimated software residual faults and failure probability from the BBN model may be very large, which makes it difficult to verify the very low failure probability of 10^{-4} to 10^{-5} required for safety-critical safety integrity level (SIL) level 4 software [11].

The test-based approach is another method that can be used to assess the reliability of NPP safety-critical software; this method applies standard statistical methods to the results of software testing, in a manner similar to that in which the reliability of hardware components is analyzed [12]. The studies relevant to the test-based approach conducted in the nuclear field are mainly divided into two testing methods: 1) black-box testing methods [13–15] and 2) white-box testing methods [16,17]. The black-box testing methods consider a software program as a black box, take random samples from its input space, determine if the outputs are correct, and use the results for statistical analyses to estimate the software reliability. However, because the black-box testing methods are conducted without knowledge on the program's internal logic or structure, the limitations of black-box testing include limited coverage and completeness of the test cases [18]. On the other hand, the white-box testing methods have an advantage in that they take into consideration the internal structures of the software; so, the tests are performed to ensure that certain parts of the software are functioning correctly, with full coverage. However, because the white-box testing methods aim to test all possible paths and nodes of the software, the number of tests that must be carried out for exhaustive testing is often very large [17] when the operational profile of the software encountered in an actual use is neglected. Therefore, an efficient and effective software testing framework for the safety-critical software used in NPP digital I&C systems must be developed to prove the correctness of the software and further quantify the software reliability based on software test results.

The objective of this study is to develop a simulation-based software test bed for white-box testing of NPP safety-critical software. The test bed is developed by emulating the microprocessor architecture of a safety-critical programmable logic controller (PLC) used in an NPP digital I&C system and capturing its behavior at each machine instruction line while the software executes its dedicated safety function. The effectiveness of the proposed software testing framework is demonstrated with the safety-critical trip logic software of a fully Integrated Digital Protection System-Reactor Protection System (IDiPS-RPS), developed under the Korea Nuclear Instrumentation & Control Systems (KNICS) project [19]. Given specific software input and internal states, the proposed method can effectively reduce software testing efforts by emulating the software behavior at a machine language level; this is in contrast to existing black-box testing, which uses trajectory inputs for software testing. The test results of safety-critical software from the suggested method can be used to support the software reliability quantification of NPP digital I&C systems and can be applied to the PRA of an NPP to analyze the effect of software failure on the digital system availability or the NPP risk.

2. Target system

In this section, an overview of the NPP safety-critical digital I&C system in which the test bed is developed is provided. The basic architecture and operation mechanism of the safety-grade PLC used in the target system are reflected in the test bed.

2.1. IDiPS-RPS configuration

The IDiPS-RPS is a digitalized RPS developed in the KNICS project for newly constructed NPPs and for upgrading existing analog-based RPS [19]. It has the same function as an analog RPS to automatically generate a reactor trip signal and engineered safety feature actuation signals whenever demand comes. Fig. 1 illustrates the architecture of the IDiPS-RPS, which has four redundant channels of processors for its dedicated safety functions.

As a part of the IDiPS-RPS, the bistable processors (BPs) determine the trip state by comparing the process variables measured from the plant sensors with the predefined pretrip or trip setpoints; coincidence processors (CPs) generate a final hardware-actuating trip signal by voting logic. The processors are configured based on the safety-grade PLC platform (POSAFE-Q) [20], and the function of each processor is implemented as software in the PLC platform.

2.2. POSAFE-Q architecture

The POSAFE-Q consists of various modules, such as a processor module, communication module, and I/O module [21]. The processor module consists of a TI C32 digital signal processor, central processing unit (CPU), and various types of memory, such as flash memory and static random access memory (SRAM). The application programs in the IDiPS-RPS, such as BP trip logic and CP voting logic, are downloaded into the memory embedded within the processor module. The application software is developed based on function block diagram and ladder diagram (FBD/LD) programming. In the implementation, the FBD/LD programs are compiled to machine instruction codes, which are loaded into the PLC memory area and executed by the PLC microprocessor [22]. Fig. 2 shows the safety-grade PLC compile procedure used to generate the machine code from the user application program, written in FBD/LD language.

3. Test bed development

In this section, the test bed development processes are described. The microprocessor architecture and operation mechanisms of the safety-grade PLC are emulated in the simulated environment. The methods of test bed verification are also described.