〈Technical Note〉

AN OVERVIEW OF RISK QUANTIFICATION ISSUES FOR DIGITALIZED NUCLEAR POWER PLANTS USING A STATIC FAULT TREE

HYUN GOOK KANG*, MAN CHEOL KIM, SEUNG JUN LEE, HO JUNG LEE, HEUNG SEOP EOM, JONG GYUN CHOI and SEUNG-CHEOL JANG
Korea Atomic Energy Research Institute
1045 Daedeokdaero, Yuseong, Daejeon, 305-353, Korea
*Corresponding author. E-mail : hgkang@kaeri.re.kr

Received January 9, 2009
Accepted for Publication March 10, 2009

Risk caused by safety-critical instrumentation and control (I&C) systems considerably affects overall plant risk. As digitalization of safety-critical systems in nuclear power plants progresses, a risk model of a digitalized safety system is required and must be included in a plant safety model in order to assess this risk effect on the plant. Unique features of a digital system cause some challenges in risk modeling. This article aims at providing an overview of the issues related to the development of a static fault-tree-based risk model. We categorize the complicated issues of digital system probabilistic risk assessment (PRA) into four groups based on their characteristics: hardware module issues, software issues, system issues, and safety function issues. Quantification of the effect of these issues dominates the quality of a developed risk model. Recent research activities for addressing various issues, such as the modeling framework of a software-based system, the software failure probability and the fault coverage of a self monitoring mechanism, are discussed. Although these issues are interrelated and affect each other, the categorized and systematic approach suggested here will provide a proper insight for analyzing risk from a digital system.

KEYWORDS : PRA, Risk, Fault Tree, Digital System, Safety-Critical, Digital

1. INTRODUCTION

One of the most important safety functions in nuclear plants is the generation of automated control signals for manipulating complicated accident-mitigation equipment including the control rod driving mechanism for reactor trips. For this safety-critical signal, many nuclear plants have adopted modern digital instrumentation and control (I&C) technologies. In Korea, Ulchin nuclear power plant (UCN) units 5 & 6 (OPR-1000) are under operation using a digital safety-critical I&C system and the Korean Next Generation Reactor (APR-1400) is being designed to use digital I&C equipment for safety functions such as the reactor protection system, an engineered safety feature actuation system, and a safety equipment control system [1,2]. The characteristics of digital systems are different from those of conventional analog systems, because their basic elements are microprocessors and software, which make a system more flexible to use but complex to analyze.

Conventional methodologies for probabilistic risk/safety assessment (PRA/PSA) could be applied to digitalized systems with rough assumptions if the reliability modeling is for a decision which does not require high accuracy, such as the determination of the number of spare modules for a non-safety-critical system. The use of a lower boundary value for system reliability is possible for simplicity [3]. More complicated relationships among system functions, however, should be modeled for an accurate and realistic estimation of risk from safety-critical digital systems. Digitalized system consists of many complex components such as microprocessors and analog-to-digital converters whose reliabilities are much worse than those of conventional analog components such as resistors and transistors. If we simply sum up the failure probabilities of its components to calculate the overall failure probability of a digitalized system, the result will be disappointing when we compare it to that of conventional analog system. However, the faults in an advanced digital system are monitored by a self-monitoring algorithm and recovered before a fault causes a system failure. Protecting a system from catastrophic damage is possible even for a fault which cannot be perfectly recovered. Multiple-channel processing systems might have cross-channel monitoring functions. Independent heartbeat monitoring

equipment can also be installed in these systems. Software-based intelligence and the flexibility of microprocessors accommodate these sophisticated reliability enhancing mechanisms successfully.

The Health and Safety Executive (HSE) guide [4] highlights the importance of PRA for digital applications as a demonstration of safety. A PRA must quantitatively demonstrate the improvement and deterioration of a developed system. A PRA also provides useful design information since it could demonstrate that a balanced design has been achieved by showing that no particular class of accident in a system causes a disproportionate contribution to the overall risk. The recent trend of risk-informed regulation and application also emphasizes the importance of the realistic modeling of digitalized safety-critical systems. Lu and Jian [5] identified several potential issues when applying PRA to I&C systems, including:

- The nature of software failures,
- The time dependency of unavailability and accident sequences,
- The lack of adequate statistical data on system and equipment failure, and
- The incomplete independence of various systems and operator errors.

The characteristics of digital systems were summarized for PRA modeling in our previous study [6] as follows:

- Modeling the multi-tasking of digital systems
- Estimating software failure probability
- Estimating the effect of software diversity and verification & validation (V&V) efforts
- Estimating the coverage of fault-tolerant features
- Modeling the common cause failure (CCF) in hardware
- Modeling the interactions between hardware and software
- Identification of the failure mode of the digital system
- Environmental effects
- Digital-system-induced initiating events including human errors

In addition to the factors above, progress in related research has revealed the following factors.

- Effect of automated periodic testing
- Modeling the network communication failure and protocol errors
- Effect of commercial-off-the-shelf (COTS) software and software developed using COTS
- Estimating the human error probabilities affected by a digitalized information system

In general, there are two kinds of modeling method for system safety. Dynamic methods are defined as those that explicitly attempt to model (1) the interactions between a plant system and the plant's physical processes, i.e., the values of process variables, and (2) the timing of these interactions, i.e., the timing of the progress of accident sequences. Static methods are well-established but they differ from dynamic methods in that they do not explicitly model the interactions between the plant system being modeled and the plant physical processes, nor do they explicitly model the timing of these interactions [7]. A fault tree is a typical example of a static modeling method. Such static methods offer the advantage of relatively easy integration with plant risk models, since conventional plant risk models are constructed using static fault trees and event trees. There is as yet no widely-accepted method for developing a realistic risk model for a digitalized safety-critical system with a static modeling method. This article aims to provide an overview of the issues related to developing such a risk model and to introduce the current status of research into the development of these models, as well as into techniques related to the issues identified in the overview.

In order to develop a static risk model of a digitalized system, the many factors listed above must be considered effectively. We categorize the complicated issues of digital system PRA into four groups in consideration of the practically available data, the depth of modeling, the dependencies among components and systems, and the effect on the plant risk. The four groups are: hardware module issues, software issues, system issues, and safety function issues. These categories are described in sections 2 to 5.

2. RISK MODEL ISSUES RELATED TO HARDWARE MODULES

The issues related to the hardware modules of a digital I&C system are the failure mode, experienced failure data, in-module fault tolerance, and imbedded components. The failure modes and data of newly developed hardware modules are not easily acquirable since digital technology evolves very rapidly and the nature of a digital fault is non-linear. Databases reflecting extensive experience and containing data from many experiments can be employed for the prediction of each component failure rate, but they may not be accurate for a newly developed component or for safety-critical equipment. MIL-HDBK-217FN2 [8], Bellcore TR-TSY-332 issue 6 [9], and RAC8 ERPD-97 [10] are popular reliability prediction databases. The Korea Atomic Energy Research Institute (KAERI) performed a case study to compare the calculated failure rate based on the database/standard method with the manufacturer's reliability data and the real experienced data of digital system failure in a nuclear plant [11]. Old manufacturer's data certainly overestimated the failure rate, but the calculations based on the database/standards revealed results in a reasonable range, as shown in Figure 1. The field data in Figure 1 shows the failure rate of the system calculated by using the collected faults from field experience.

The lack of failure data is one of the major deficiencies in assessments of the risk of computer-based systems in nuclear facilities. To remedy this situation, the Organization for Economic Co-operation and Development/Nuclear

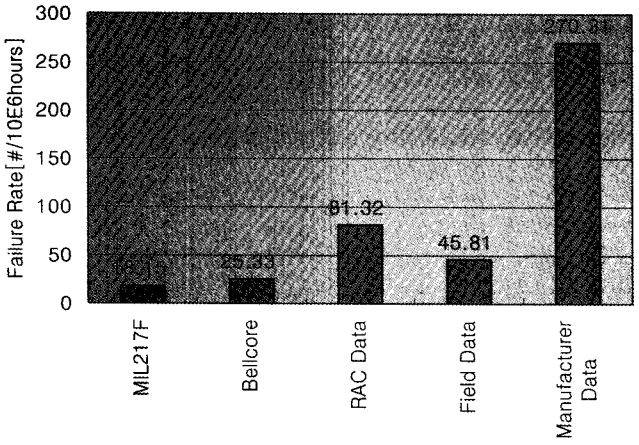

Fig. 1. Failure Rate of the PDC System in CANDU Nuclear Plant [11]

Energy Agency (OECD/NEA) established an international data collection and analysis project (Computer-Based Systems Important to Safety (COMPSIS) project) [12]. This project is on-going, and it is expected to enable the identification of the root cause of a computer-based system failure as well as the effect of the failure and the determination of how the failure could have been prevented, if the project successfully gathers the digital system failure data from many countries.

Analysis of failure modes and their effects on a system is very important. Chu et al. [7] performed a failure modes and effects analysis (FMEA) for a digital control system and pointed out that an FMEA for generic components of a digital module successfully supports the model to realistically represent the system, but that incomplete knowledge of the digital components' failure modes must be supplemented.

In safety-critical applications, the applied hardware modules might be equipped with an in-module fault tolerance mechanism to enhance their safety. Some hardware modules are linked with system-level or function-level fault monitoring mechanisms. Careful investigation of these fault-tolerance mechanisms and quantification of their effectiveness are necessary for a realistic risk model. The effects of these mechanisms should be carefully modeled to avoid double counting and should be consistently treated in the probability estimation of component failure, hardware module failure, system failure, and function failure.

Lee et al. proposed a method for calculating the function failure probability of a typical digital module [13]. Figure 2 shows the functional block diagram of a typical digital hardware module. The components of the hardware module can be categorized into 4 sub-function groups according to their functions. The G1 sub-function group receives input signals, transforms them, and compares

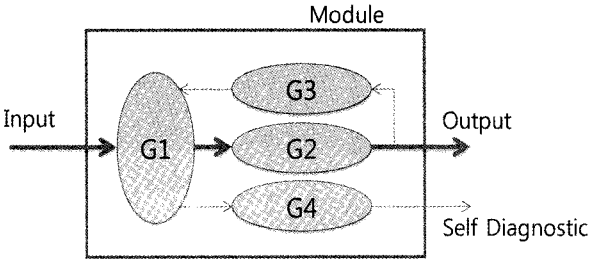

Fig. 2. The Structure of a Typical Digital Module [13]

the transformed signal with the feedback loop, composed of sub-function groups G2-G3. G1 generates an error signal through G4 whenever a deviation is detected by the loop-back test.

If only a component in G2 fails, the module does not create the final output, but the self-diagnostic feedback loop through G3-G1-G4 immediately generates an error alarm signal. This is a so-called safe failure. The loss of safety function is caused by dangerous failures, which are those failures that go undetected by self diagnostics. The dangerous failure (DF) probability is calculated based on a Boolean expression of module function failure events.

$$P(\mathrm{DF}) = P\left\{\overline{G_1} + G_1\overline{G_2}\,\overline{G_3} + G_1\overline{G_2}\,\overline{G_4} + G_1\overline{G_2}\,\overline{G_3}\,\overline{G_4}\right\} \quad (1)$$

where G_i and $\overline{G_i}$ denote the success and failure of Gi, respectively. By the absorption rule of Boolean algebra,

$$P(\mathrm{DF}) = P\left\{\overline{G_1} + G_1\overline{G_2}\,\overline{G_3} + G_1\overline{G_2}\,\overline{G_4}\right\} \quad (2)$$

We assume that the failures of G1, G2, G3 and G4 are independent and that no CCF exists among them. If $P(\overline{G_1}) >> P(\overline{G_2})P(\overline{G_3})$, $P(\overline{G_1}) >> P(\overline{G_2})P(\overline{G_4})$ and $P(\overline{G_1}) << 1$, then with rare event approximation, $P(\mathrm{DF}) \approx P(\overline{G_1})$.

It should be noted that Equation (2) assumes that the fault detection scheme is complete, and is established with a loop for failure detection. If there is any possibility of a failure of the loop-back test, it should be considered as:

$$P(\mathrm{DF}) = P\left\{\overline{G_1} + G_1\overline{G_2}\,\overline{G_3} + G_1\overline{G_2}\,\overline{G_4} + \overline{C}\,\overline{G_2}\right\} \quad (3)$$

where $\overline{C}$ denotes the failure in detecting a fault. If $P(\overline{C})$ is considerably large, the last term in Equation (3) is not negligible. In such a case, the coverage, $P(C)$, must be estimated quantitatively. The methods for estimating the coverage will be discussed in section 4.

Microprocessors and software technologies make a

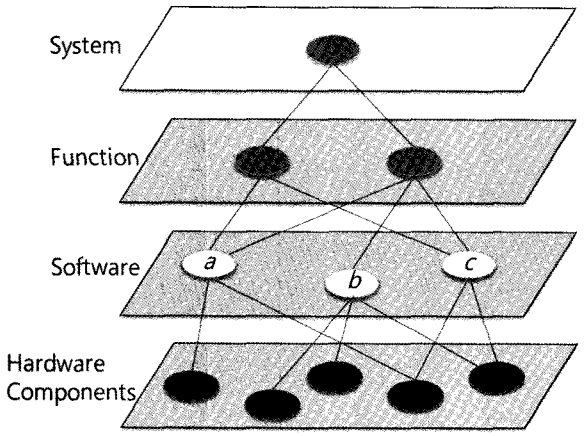

Fig. 3. An Example of a Hierarchical Structure in a Digital System

digital system multi-functional and such a system performs several sequential or conditional functions. In accordance with a change in function, the use of hardware components may be changed. Therefore, for the failure probability estimation of hardware modules, the functions of each module must be investigated carefully. Figure 3 shows an example of a hierarchical structure in a digital system which explains the dependency among the function, the software, and the hardware components of that system.

In the example shown in Figure 3, the hardware components which are utilized by software programs *a* and *c* must be considered to obtain the failure probability of function *A*. Hardware components 2 and 3 are not utilized and their failure probabilities must therefore be excluded from the calculation. That is, the calculation of the hardware module failure probability must include an analysis of the target function or software.

Logics in some components are programmable (in so-called imbedded components). This implies that some special components such as a field programmable gate array (FPGA) may have a hierarchical structure inside the hardware. The logics in these special components can be treated as a kind of software. In this case, the component itself might be the subject of investigation with an identical scheme for a system and function analysis.

3. RISK MODEL ISSUES RELATED TO SOFTWARE

The issues related to software are its failure mode and failure probability quantification (in consideration of software testing and verification and validation (V&V)). Software is at the center of many of the important safety issues in digital system safety assessment. Software failure modes are hard to define, so all software failure modes might be assumed to be hazardous if a software hazard analysis cannot appropriately limit hazardous failure mode identifications. If we consider an automated signal processing software, from the viewpoint of system function, there are two software failure modes: abnormal response and wrong output generation. Software halt caused by an infinite loop is a good example of abnormal response. In instances of wrong output generation, the software continues running but generates incorrect output. In terms of recovery, the failure modes can be subdivided into detectable and undetectable.

It is notable that there is much discussion among software engineering researchers about whether a software failure can be treated in a probabilistic manner [14]. Software faults are, by definition, design faults. This means that software is deterministic and its failure cannot be represented by a 'failure probability'. However, a fault in a software program causes system failure only when the input sequence activates that fault. If we assume that input sequences are random during the real use of a software program, its failure could be treated based on a probabilistic method. Furthermore, if the input profile of the software can be determined statistically, we can estimate the software reliability in a probabilistic manner based on the input profile.

Prediction of software reliability using a conventional model is generally much harder than for hardware reliability. The software reliability growth model (SRGM) is the most mature technique for software dependability assessment, and it functions by estimating the increment of the reliability as a result of a fault removal. However, this approach is known to be inappropriate for safety-critical systems since the applied fixes cannot be assumed to be effective and the last fix may have introduced new faults [15]. A recent study identified both some possibilities for and some limitations on the application of conventional SRGMs to safety-critical software [16]. Even though conventional SRGMs can sometimes prove the high reliability of a safety-critical piece of software, their major limitations for application to safety-critical systems are the high sensitivity of the estimated total number of inherent software faults to the time-to-failure data and the uncertainty of the availability of sufficient software failure sets.

Applying the lower limit of a software failure probability estimated conservatively through testing can be an alternative to using conventional SRGMs. The number of observed failures of a highly reliable software program during a test is expected to be zero because the elucidated errors will be debugged in the corresponding code and the test will be repeated. The method of calculating the required number of tests is easily derived using conventional statistics. Test stopping rules are also available for those cases where testing has been resumed after error correction [17].

One of the important aspects of test-based software reliability assessment is that the test cases should represent the inputs which are encountered during actual use. The

test inputs for safety-critical applications such as the reactor protection system (RPS) of a nuclear power plant are inputs which cause the activation of a protective action such as a reactor trip. Thus, an appropriate input profile must be determined for effective software failure probability quantification. A digital system treats inputs from instrumentation sensors as discrete digits by using an analog-to-digital converter. The input profile for the RPS must therefore be estimated based on these characteristics of the digital system and based on the plant dynamics [18].

High-integrity software validation testing (such as model-based statistical testing) has proven successful in the computing industry and should be investigated for use in nuclear power plants through experimental demonstration. In addition, fault-seeding approaches and testing programs that use massively parallel computers have shown promise and could be developed further [19].

The software development process can also be considered in order to assess the expected software failure rate. The number of remaining potential faults in a piece of software is reduced by using software V&V methodologies. This effect is reflected in the probability estimation of the software failure events. Thus, the quantification of the rigidity of software V&V could be performed through the PRA process. A Bayesian belief network (BBN) can be used for estimating the effectiveness of these quality-improving efforts in a more systematic manner [20, 21]. Applying the BBN methodology to the PRA of digital equipment is helpful to integrate many aspects of software engineering and quality assurance. The BBN is a network-based formalism which can model uncertain and complex domains. In evaluating software reliability, a BBN can explicitly consider the important factors that are relevant to the reliability, such as the quality of the developer, the development process, the complexity of the problem, the testing effort, and the operation environment, etc [22]. It is widely recognized that the quantitative reliability of safety-critical software is difficult to assess by any specific method [23]. For this reason, a human expert must synthesize a large volume and a wide variety of evidence to reach a conclusion about the reliability of safety-critical software. This kind of expert judgment is expressed representatively in the licensing process of digital systems intended for use in a nuclear power plant [24]. However, synthesizing the requisite evidence is not easy even for a human expert [25]. BBNs promise to solve this difficulty in a consistent and systematic way, so the BBN-based reliability evaluation method seems promising.

There still exist some difficulties in the practical application of a BBN. For example, it is difficult to obtain expert knowledge in a form that can be converted into a probability distribution [26], and there is also not enough software failure data, especially in the nuclear field, that can be used as the reference for the validation of a constructed BBN model.

The diversity in software codes and hardware-software interactions are additional considerations. There is as yet no widely-accepted method for the quantification of these effects.

4. RISK MODELING ISSUES RELATED TO THE SYSTEM

Modeling issues related to the system to be modeled are the risk concentration and diversity (including CCF), the fault coverage of self/peer monitoring, the effectiveness of automated periodic system testing, and network communication failures. The use of a single microprocessor module for multiple safety-critical functions will cause severe concentration of risk in that single microprocessor. Safety-critical applications have adopted a conservative design strategy, based on functional redundancies. However, the software programs for these functions are executed sequentially by one microprocessor. Therefore, the level of redundant design in digital systems is usually higher than in conventional mechanical systems. This higher redundancy will clearly reduce the risk from a single component failure, but raise the severity of CCF consequences.

This higher level of redundancy exponentially increases the number of CCF events modeled in a fault tree, if conventional CCF modeling methods are applied. In some nuclear power plants, there are four signal processing channels for the safety parameters, and each channel consists of two or four microprocessor modules for the same function. For example, in the RPS of the OPR-1000 plant, there are 16 processors that share the identical function of local coincidence logic. In this case, the system model will have 65519 events to represent the CCFs of the local coincidence logic processors. The simplified alpha-factor (SAF) method is a promising alternative for modeling the CCF in digital safety-critical systems. By using the SAF method and the results of prior investigation on the system failure logics, the probability of a single CCF event can be used to represent all the CCF events that may result in a system failure [27]. The probability of a single CCF event for m redundant components can be defined as follows.

$$Q_{CCF} = \sum_{k=2}^{m} ({}_{m}C_{k} \times p_{k} Q_{k}^{m}) \tag{4}$$

where Q_k^m is the probability of a CCF of k out of m components and p_k denotes the ratio of system failure CCFs over all possible k component CCFs. If a non-staggered test is applied and the CCFs are analyzed using the alpha factor method, Q_k^m can be calculated as follows.

$$Q_{k}^{m} = \frac{k}{{}_{m-1}C_{k-1}} \cdot \frac{\alpha_{k}^{m}}{\sum_{i=1}^{m} (i \cdot \alpha_{i}^{m})} \cdot Q_{t} \tag{5}$$

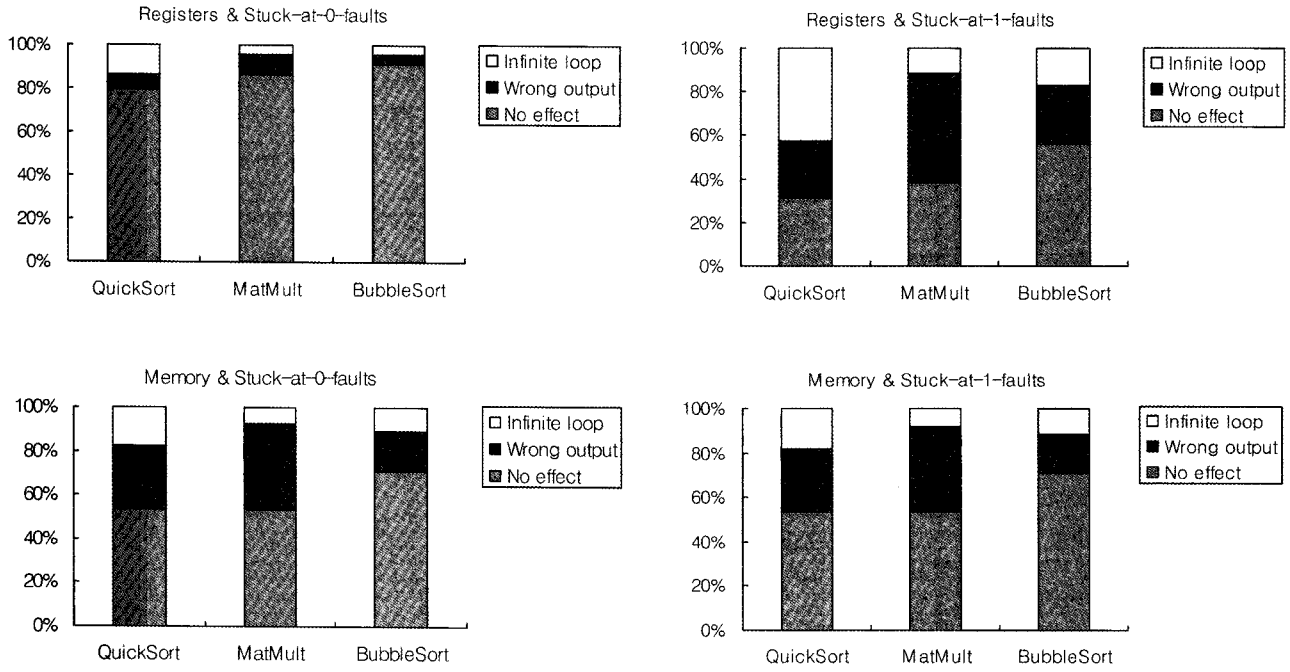

Fig. 4. Results of Simulated Fault Injection Experiments

where Q_t is the failure probability of a component, and α_k^m denotes the portion of the k components' CCF over the m-component CCF group.

The effectiveness of fault-tolerant mechanisms dominates system reliability. The quantification of fault coverage must be treated carefully. Fault-tolerant mechanisms are implemented to check the integrity of system components. Self monitoring is one of the fault-tolerant mechanisms. If more than one processor module is installed in a system, they might monitor each other. Some monitoring algorithms also consider the status of the in-module fault tolerance mechanisms, as mentioned in section 2. The concept and formulas of fault coverage are described well in the literature [28,29]. Simulation using fault injection is a promising method for estimating the coverage factor. Expert knowledge might also be used to estimate the rough bounds of the coverage.

Kim et al. [30] and Lee et al. [31] showed that the fault coverage of a safety-critical digital system can be quantified effectively using the simulated fault injection method. Kim et al. [30] proposed a method for evaluating the fault coverage of a system by combining the fault coverage of various fault-tolerant mechanisms. Lee et al. [31] evaluated the error detection coverage for component faults and showed how error detection coverage for a system can be calculated based on the error detection coverage for the component faults.

In an effort to correctly estimate the fault detection coverage of digital I&C systems, KAERI recently performed simulated fault injection experiments on a 32-bit processor-based digital system without any fault-tolerant mechanisms. Quick sort, bubble sort and matrix multiplication, which are the three most frequently used software applications in the literature, were selected as the software applications. Permanent stuck-at-0 and stuck-at-1 faults were injected into the target system. The experimental result is shown in Figure 4. It was found that a large portion of the permanent stuck-at faults has no effect on output generation, even when no fault detection mechanism is applied to the target system. It was also found that fault detection coverage is dependent on the fault type, fault location, and the software application. It is, however, notable that if an improper fault profile is applied, the injected faults may not represent all the conditions that may be experienced during actual operation.

A human operator or equipment outside a system initiates the testing, and the successful completion of the testing implies non-faulty status of the corresponding components. If periodic testing is automated, the inspection period can be reduced, and thus system reliability can be improved. Automatic periodic tests affect the system reliability in a manner similar to self-diagnostics. The issue that must be addressed is the quantification of the effectiveness of automated tests. Assume an example system checks its availability through self-diagnostics, online monitoring, automatic periodic tests, and manual periodic tests. If we assume that all the monitoring processes and tests are perfect, then when the self-diagnostic system or online monitoring system detects a fault without a time delay, the unavailability can be calculated using the

following equations. If $\lambda_S T_S < 0.1$, $\lambda_A T_A < 0.1$ and $\lambda_M T_M < 0.1$, with the first term of Tailor's series,

$$Q = \lambda_S \frac{T_S}{2} + \lambda_S T_R + \lambda_A \frac{T_A}{2} + \lambda_A T_R + \lambda_M \frac{T_M}{2} + \lambda_M T_R$$
$$= \lambda_S (\frac{T_S}{2} + T_R) + \lambda_A (\frac{T_A}{2} + T_R) + \lambda_M (\frac{T_M}{2} + T_R) \quad (6)$$

and if $T_S << T_R$ and $T_R << T_M$,

$$Q \approx \lambda_S T_R + \lambda_A (\frac{T_A}{2} + T_R) + \lambda_M \frac{T_M}{2} \quad (7)$$

where

Q = System unavailability

λ_S = Failure rate for the portion detected by a self-diagnostic or online monitoring

λ_A = Failure rate for the portion detected by an automatic periodic test

λ_M = Failure rate for the portion detected by a manual periodic test

T_S = Time interval of self-diagnostics or online monitoring (e.g. 10 secs)

T_A = Time interval of automatic periodic tests (e.g. 8 hrs)

T_M = Time interval of manual periodic tests (e.g. 720 hrs)

T_R = Time required for maintenance (e.g. 24 hrs)

A module failure may be detected by one or several inspection methods. For example, some failures in bus modules are detectable with self-diagnostics, online monitoring, and periodic tests. Some failures in input modules may be detected only by periodic tests. The characteristics of failures in a module must be considered carefully. The FMEA can be utilized to identify the corresponding inspection methods. An example module was analyzed using this method and the sensitivity analysis result is shown in Figure 5, which illustrates the unavailability of the system. As shown in the graph, the shorter test period results in lower unavailability in all cases. While the effects of period reductions from 48 hours to 16 hours are evident, the reduction of inspection interval periods from 8 hours to 2 hours does not show remarkable effects. Based on these results, it is expected that an optimized period could be found that would guarantee sufficiently high reliability with a minimum number of required tests.

The use of signal transmission components, such as fiber-optic modems and opto-couplers, is reduced by using network communication. If safety-critical signals are transmitted through network communication, the probability that a system becomes unsafe due to a network failure must be evaluated to quantify the system risk. Hazard analysis and the identification of paths which might lead a system to an unsafe state are required, as is the probabilistic quantification of each path. Network failure is caused by defects in the hardware of the network modules or a fault in the network protocol, which is the basis of network software.

5. RISK MODEL ISSUES RELATED TO SAFETY FUNCTION

The issues related to a safety function are human-system and system-system interactions, plant condition dependency and coverage of the test inputs both before and after installation. The PRA provides a unifying means

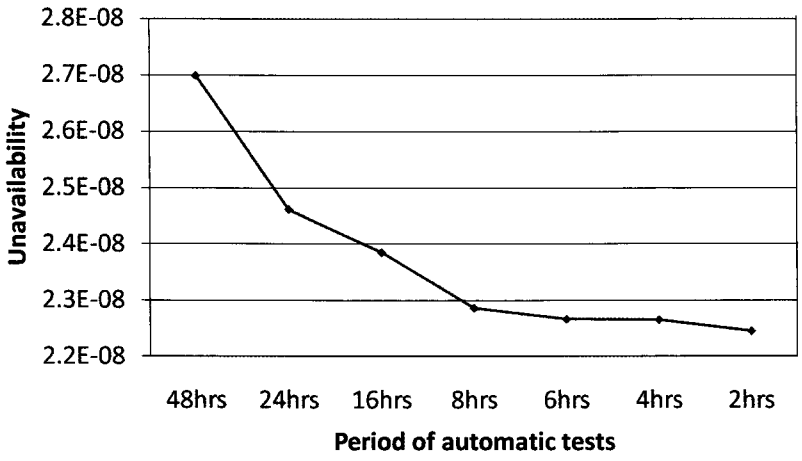

Fig. 5. Unavailability of the Example System along with an Automatic Periodic Test Interval Change (Manual Periodic Test Interval is 720 hrs and Self-diagnostics Continuously Monitor the System Integrity)

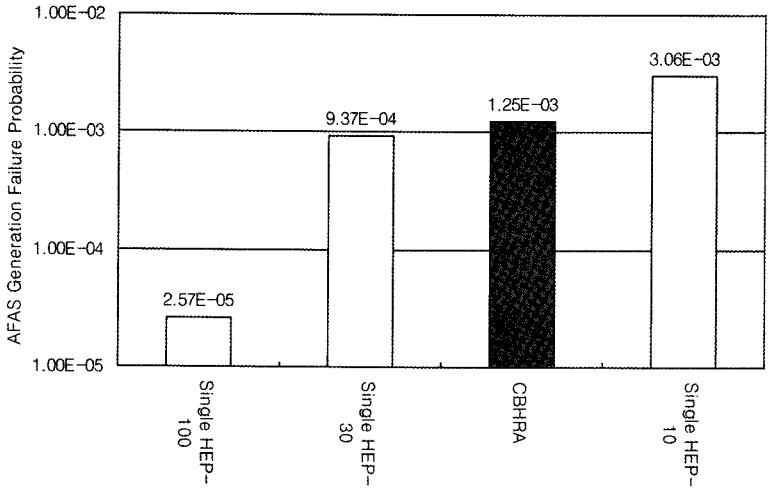

Fig. 6. A Comparison among the Single HEP Methods and the CBHRA Method for the AFAS Generation Failure Probabilities

of assessing the system safety, including the activities of human operators. The effect of a human operator who plays the role of a backup for an automated digital system must be modeled. If there is more than one automated system, the interaction among the systems should also be modeled. Failure of a safety function implies the concurrent failures of automated systems and human operator backups. In some systems, the failure of an automated system is closely correlated with the failure of the manual human operator backup [32]. The dependency must be carefully investigated for realistic evaluation of the system function failure. Figure 6 shows an example analysis result of condition-based human reliability assessment (CBHRA) in consideration of the human-system dependency. In this study, the human error probability (HEP) is the probability of an unsafe action (UA) that is affected by the error forcing context (EFC). Given an accident scenario, the HEP (H) is calculated as in [33] and [34]:

$$H = \sum_{i} P(UA \mid EFC_i) P(EFC_i) \tag{8}$$

CBHRA treats the status of the information provided by a system as a kind of EFC. In comparison with the conventional single HEP method, CBHRA considers possible changes of the HEP along with the signal generation system status. In Figure 6, Single HEP-100, Single HEP-30 and Single HEP-10 indicate that the HEP is calculated based on the assumption that 100%, 30% and 10% of the diagnosis time is available, respectively. Detailed information for CBHRA is available in the literature [32].

The function of an I&C system is usually initiated by the input signals, which depend on the plant condition and occurring accidents. Thus, the availability and the validity of an input signal are very important. Redundant input signals or an operator's manual input are available in some cases. The development of a PRA model requires in-depth analysis of input availability, including a document survey, a simulation and an expert judgment [35].

The test input coverage largely affects the credibility of the reliability, whether system/component reliability is determined based on a pre-install test or the system is tested after installation. Some tests that cover only a portion of system components are partially effective. In a safety-critical system, demand usually arrives when the plant is in a deviation status. A digital system has a limited number of possible input spaces because of its limited resolution, if we exclude the dependency on time. In consideration of these conditions, the test input coverage must be evaluated.

6. CONCLUSIONS

As the digitalization of safety systems progresses, risk model development and quantification become increasingly important tasks in the safety assessment of nuclear power plants. In order to utilize the existing PRA models, a method for the static modeling of digital equipment is required. In this article, we provided an overview of the issues affecting the development of a static fault-tree-based risk model for digitalized safety-critical systems and of the recent research activities aimed at addressing these issues. The issues were categorized into four groups: hardware issues, software issues, system issues and safety function issues.

The issues related to the hardware modules of a digital I&C system are the failure mode, the experienced failure data, the in-module fault tolerance, the dynamism, and imbedded components. Those related to the software are

the failure mode and failure probability quantification. Those related to systems are the risk concentration and risk diversity (including CCF), the fault coverage of self/peer monitoring schemes, the effectiveness of automated periodic system testing, and network communication failures. Issues related to safety functions are the human-system and system-system interactions, the plant condition dependency and the coverage of the test inputs both before and after installation. Quantification of these issues dominates the quality of a developed model. Recent research activities for addressing various issues are briefly discussed.

ACKNOWLEDGEMENTS

This work was supported by Nuclear Research & Development Program of the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korean government (MEST). (grant code: M20702030002-08M0203-00210)

REFERENCES

[1] Kang, H.G, et al., Survey of the Advanced Designs of Safety-Critical Digital Systems from the PSA Viewpoint, Korea Atomic Energy Research Institute, KAERI/AR-00669/2003, 2003.

[2] Shin, H.G, Nam, S.G, Sohn, S.D and Chang, H.S, "Development of an advanced digital reactor protection system using diverse dual processors to prevent common-mode failure," Nuclear Technology, Vol.141, 2003.

[3] Seong, P.H, et al., Reliability and Risk Issues in Large Scale Safety-critical Digital Control Systems, Springer London, 2008.

[4] HSE, The use of computers in safety-critical applications, London, HSE Books, 1998.

[5] Lu, L and Jiang, J, "Probabilistic Safety Assessment for Instrumentation and Control Systems in Nuclear Power Plants: An Overview," Journal of Nuclear Science and Technology, Vol. 41, No.3, 2004.

[6] Kang, H.G and Sung, T, "An analysis of safety-critical digital systems for risk-informed design," Reliability Engineering and Systems Safety, Vol. 78, No. 3, 2002.

[7] Chu, T.L, Martinez-Guridi, Yue, M, Lehner, J, and Samanta, P, "Traditional Probabilistic Risk Assessment Methods for Digital Systems," NUREG/CR-6962, October 2008.

[8] US MIL-HDBK-217, Reliability Prediction of Electronic Equipment, version F, DOD, USA, 1991.

[9] Bellcore Technical Ref. TR-TSY-000332, Reliability prediction procedure for electronic equipment: issue 6, 1997.

[10] ERPD-97, Electronic Parts Reliability Data, RAC, 1996.

[11] Jung, H.S, Jang, S.C, Kim, M.C, Jun, S.T, "Analysis of Hardware Reliabilities for NPP Digital I&C Equipment Predicted by Various Methods," International congress on advances in nuclear power plants; ICAPP '03, 2003.

[12] OECD/NEA, Computer-Based Systems Important to Safety (COMPSIS) Project: 3 Years of Operation (2005-2007), Draft Report, NEA/CSNI/R(2008). 2008.

[13] Lee, D.Y, Choi, J.G, and Lyou, Y, "A Safety Assessment Methodology for a Digital Reactor Protection System," International Journal of Control, Automation, and Systems, Vol. 4, No. 1, 2006.

[14] White, R.M and Boettcher, D.B, "Putting Sizewell B digital protection in context," Nuclear Engineering International, pp. 41-43, 1994.

[15] Parnas, D.L, Asmis, G.J.K, and Madey, J, "Assessment of Safety-critical Software in Nuclear Power Plants," Nuclear Safety, Vol. 32, No. 2., 1991.

[16] Kim, M.C, Jang, S.C, and Ha, J, "Possibilities and limitations of applying software reliability growth models to safety-critical software," Nuclear Engineering and Technology, vol.39, no.2, pp.145-148, 2007.

[17] Littlewood B, Wright D, "Some conservative stopping rules for the operational testing of safety-critical software," IEEE Trans. Software Engineering, Vol. 23, No. 11, 1997, pp. 673-685.

[18] Kang, H.G, Lim, H.G, Lee, H.J, Kim, M.C, and Jang, S.C, "A Test-Based Software Failure Probability Quantification Method for Safety-Critical Applications," The 7th International Topical Meeting on Nuclear Reactor Thermal Hydraulics, Operation and Safety, Seoul, Korea, October 5-9, 2008.

[19] INL, Technology Roadmap on Instrumentation, Control, and Human-Machine Interface to Support DOE Advanced Nuclear Energy Programs, INL/EXT-06-11862, Idaho National lab., March 2007.

[20] Dahll, G, The use of Bayesian Belief Nets in Safety Assessment of Software based System, HWP-527, Halden Project, 1998.

[21] Eom, H.S, et al., Survey of Bayesian Belief Nets for Quantitative Reliability Assessment of Safety Critical Software Used in Nuclear Power Plants, Korea Atomic Energy Research Institute, KAERI/AR-594/2001, 2001.

[22] Fenton, N, Neil, M, David Marques, "Using Bayesian Networks to Predict Software Defects and Reliability," 5th International Mathematical Methods in Reliability Conference (MMR 07), July 2007.

[23] Butler, R.W and Finelli, G.B, "Infeasibility of Quantifying the Reliability of Life-Critical Real-Time Software," IEEE Transaction on Software Engineering, Vol.19, Issue 1, IEEE Press, 1993.

[24] IEEE, IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations, IEEE-7.4.3.2, 2003.

[25] Kahneman, D, Slovic, P, and Tversky, A, Judgment under uncertainty: Heuristics and biases, Cambridge University Press, 1982.

[26] Uusitalo, L, "Advantages and challenges of Bayesian networks in environmental modeling," Ecological modeling, Vol. 203, pp.312-318, 2007.

[27] Kang, H.G, et al., The Common Cause Failure Probability Analysis on the Hardware of the Digital Protection System in Korean Standard Nuclear Power Plant, KAERI/TR-2908/2005, 2005.

[28] Kaufman, L.M, Johnson, B.W, and Bechta Dugan, J, "Coverage Estimation Using Statistics of Extremes for When Testing Reveals No Failures", IEEE Transactions on Computers, Vol. 51, No. 1, 2002.

[29] DeLong, T, Smith, D, and Johnson, B, "Dependability Metrics to Assess Safety-Critical Systems," IEEE Transactions on Reliability, Vol. 54, No. 3, 2005.

[30] Kim, S.J, Seong, P.H, Lee, J.S, Kim, M.C, Kang, H.G, and Jang, S.C, "A Method of Fault Coverage Evaluation

for Digitalized Systems in Nuclear Power Plants using Simulated Fault Injection," Reliability Engineering and System Safety, vol.91, pp.614-623, 2005.
[31] Lee, J.S, Kim, M.C, Seong, P.H, Kang, H.G, and Jang, S.C, "Evaluation of error detection coverage and fault-tolerance of digital plant protection system in nuclear power plants," Annals of Nuclear Energy, vol.33, pp.544-554, 2006.
[32] Kang, H.G and Jang, S.C, "Application of condition-based HRA method for a manual actuation of the safety features in a nuclear power plant," Reliability Engineering & System Safety, Vol. 91, 2006.
[33] US Nuclear Regulatory Commission (USNRC), Technical basis and implementation guidelines for a technique for human event analysis (ATHEANA), Washington, D.C., NUREG-1624 Rev. 1, 2000.
[34] Forester, J, Bley, D, Cooper, S, Lois, E, Siu, N, Kolaczkowski, A, and Wreathall, J, "Expert elicitation approach for performing ATHEANA quantification," Reliability Engineering and System Safety, Vol. 83, 2004.
[35] Kang, H.G, Jang, S.C, and Lim, H.G, "ATWS Frequency Quantification Focusing on Digital I&C Failures," Journal of Korea Nuclear Society, Vol. 36, 2004.