Proceedings of the Korean Nuclear Society Spring Meeting
Cheju, Korea, May 2001

PSA as a Measure of Digital Systems' safety

Hyun Gook Kang · Taeyong Sung
Korea Atomic Energy Research Institute
P.O Box 105, Yusung, Taejon, 305-600, Korea

Abstract

Microprocessors and software technologies make the digital system very complex to analyze. Even though the conventional probabilistic safety assessment methods are immature for applying to microprocessor-based digital systems, practical needs force to apply it. The aim of this paper is to introduce the role of probabilistic safety assessment in the safety evaluation of digital equipment, to summarize the important factors, and to propose a standpoint of evaluation for digital systems. We expect that the analysis result will provide valuable design feedback if the analysis is performed with careful consideration for avoiding the unrealistic assumptions.

1. Digital systems in nuclear power plants

Since the 1980's many utilities have coped with the aging of analog instrumentation and control (I&C) equipment. The obsolescence and malfunctions of analog I&C components and systems in conventional nuclear power plants has been one of the most severe problems. Furthermore, next-generation advanced nuclear reactors require more complex and smart functions for control systems, protection systems and operator-supporting systems [1].

The modern technologies which are based on both of digital hardware and advanced software algorithms are being rapidly developed and widely used. By the general progress of I&C technologies for process engineering such as computer technology, control engineering, data processing and transfer technology, and software technology, the modern digital technologies are expected to significantly improve the performance and the safety of nuclear power plants. Digital technology was introduced relatively recently in the nuclear power industry and some utilities adopted modern digital technologies to their I&C systems in recent

years.

In France, many of the 900 MWe series and the 1300 MWe series adopted computers and associated data processing systems. Works on the development and implementation of digital I&C systems for advanced reactors are actively underway in Japan. Several US plants have retrofitted digital systems to replace parts of analog systems [2]. Digital technologies are adopted in the late advanced gas cooled reactors (AGRs) in UK for safety features actuation. Primary Protection System (PPS) of Sizewell B in the UK also employed microprocessors [3]. Especially, in Korea, UCN 5&6 units are being constructed and Korean Next Generation Reactor (KNGR) is being designed using the digital I&C equipment for the safety related functions such as a reactor protection system and an engineered safety feature actuation system. Even though the use of digital equipment for safety-related functions provides many advantageous features, there are also many licensing issues which should be solved.

Various researches for applying digital equipments and advanced software algorithms to nuclear power plant I&C systems have been performed by numerous investigators worldwide. Even the protection system which is one of the most safety-critical function systems in nuclear power plants is developed based on microprocessors.

2. Probabilistic safety assessment method

The development of a methodology for the probabilistic safety assessment (PSA) of digital I&C system is one of the most important issues. The PSA has been widely used in nuclear industry for licensing and identifying vulnerabilities to plant safety since 1975. PSA techniques are used to assess the relative effects of contributing events on system-level safety or reliability and provide a unifying means of assessing physical faults, recovery processes, contributing effects, human actions, and other events that have a high degree of uncertainty [4]. However, the PSA using conventional techniques cannot adequately evaluate some features of digital systems. Main difficulties arise from the software and from the many design features such as phased-mission, fault tolerance, and fault detection and removal. It will take a long time to establish a well-accepted standard on the quantitative safety assessment of digital I&C equipment in the nuclear industry.

Unlike conventional standards, new international standards require quantitative analysis [5]. Because of the prematureness of methodologies, many assumptions are used for quantitative analysis. Unreasonable assumptions cause unreasonable results of the analysis. Fault-free software and 100% coverage of fault tolerance mechanism are the representative ones. The

wrong assumptions could distort the result of analysis. In order to obtain more reasonable results, these critical assumptions should be removed.

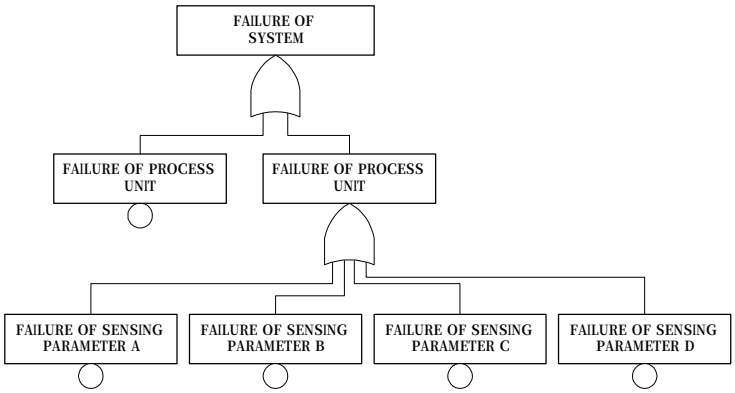


Figure 1. Schematic diagram of typical signal processing using a digital processor unit

3. Important factors in digital system safety assessment

The digital techniques are far from the conventional techniques of analog I&C systems because of some unique features of the digital I&C system. Microprocessors and software technologies are the basic elements of the digital system. They make the system more flexible and powerful but more complex.

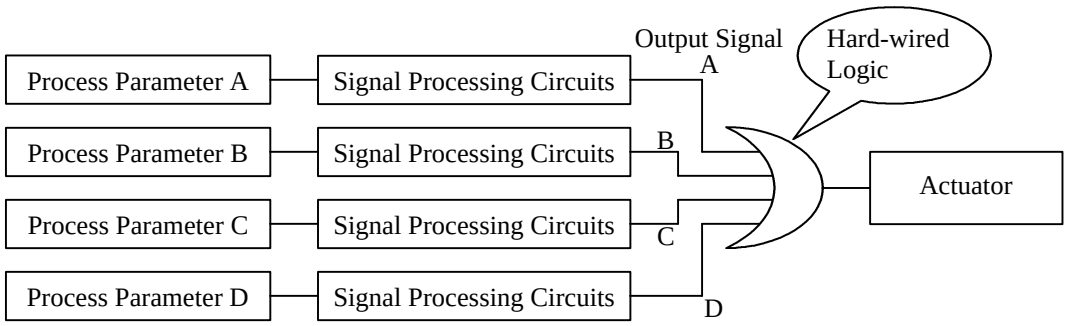
A. Modeling the multi-tasking of digital systems

Microprocessors and software technologies make the digital system multi-functional. That is, a system performs several functions sequentially or conditionally. This multi-tasking feature should be represented in PSA modeling because it will induce risk concentration and deteriorate the reliability of system.

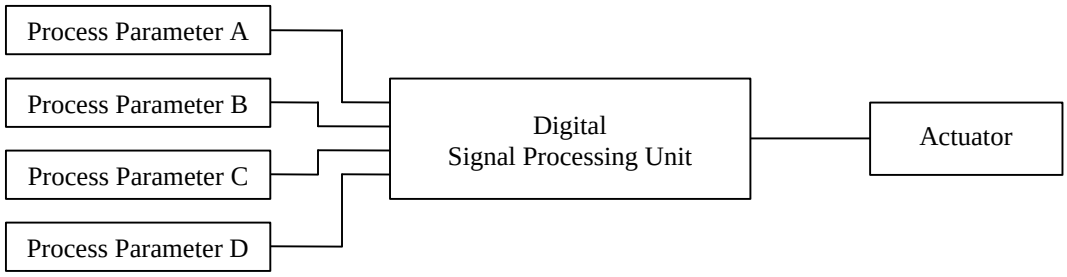
The designers of safety-critical systems such as nuclear power plants have adopted a conservative design strategy and given various functional redundancies through separated systems. When the digital I&C system is applied to the safety-critical system, the software programs of these functions are executed by a system and the redundancy is no more valid. Especially, in order to compare the developed digital system with the conventional analog system, the effects of multi-tasking on the safety should be carefully modeled and evaluated.

For example, consider two systems shown in Figure 2. As explained above, typical safety

critical applications such as reactor protection systems in nuclear power plants handle diverse process parameters and it provides redundancy. Consider the Main Steam Line Break (MSLB) accident. First, 'Low steam generator pressure' parameter (A) triggers the output signal A. As time goes on, the parameters of ' Low pressurizer pressure' (B), ' Low steam generator level' (C), 'Reactor over power' (D) will trigger the output signal B, C and D, respectively.



(a) Typical process of signal processing using conventional analog circuits



(b) Typical process of signal processing using digital units

Figure 2. Schematic diagram of signal processing using analog circuit and digital processor unit

In conventional analog circuit system, as shown in Figure 2 (a), the first triggered output signal, signal A, makes trip circuit breakers open and initiates reactor shutdown. If the signal processing circuits for parameter A fail to generate the proper output signal, the second triggered output signal (B) will trip the reactor. And if the circuits for parameter B also fail, the output signal C, will trip the reactor. However, in the case of digital system, as shown in Figure 2 (b), parameter A, B, C and D use the same equipment for signal processing. If the digital signal-processing unit fails, there is no backup. Of course, in the safety critical application, there are one or more duplicated trip channels, but conventional analog systems also have fully duplicated channels.

Multi-tasking is generally adopted in microprocessor-based systems, so the signal processing systems in a nuclear power plant tends to have multi-input single-output structure. The multi-

tasking of digital systems could result in risk concentration on processing module and output module. It implies that the reliabilities of these components should be analyzed more carefully. At the viewpoint of the designer, it also implies that self-monitoring and fault-tolerant mechanism for these components should be strengthened.

B. Estimating software failure probability

Generally, we recognize that software faults are by definition design faults. That is, software is deterministic and its failure cannot be represented by 'failure rate'. When we focus on the software of a specific application, however, the software is no more deterministic because of the randomness of the input sequences. This is the concept of 'error crystals in software,' which is the most common justification for the apparent random nature of software failure. Error crystals are the regions of the input space that cause software to produce errors and a software failure occurs when the input trajectory enters an error crystal.

Unlike the reliability of hardware components, it has been proved that it is much harder to predict software reliability using a conventional model. Software reliability growth model is the most mature technique for software dependability assessment. It estimates the increment of reliability as a result of fault removal. It is assumed that when a failure occurs there is an attempt to remove the design fault that caused the failure. The repeated occurrence of failure-free working is the input to probabilistic reliability growth models, which use these data to estimate the current reliability of the program under study, and to predict how the reliability will change in the future. However, there is no way to choose a priori the most suitable model for a particular situation [6].

In order to apply software failure probability to the fault tree model, we require the basic event probability of software failure. Conservatively estimated lower limit of software-failure probability by testing can be an alternative. Of course, some researchers insist that the quantification of safety-critical software reliability is infeasible using statistical methods because it leads to exorbitant amounts of testing when applied to safety-critical software [7]. However, in order to show the integrity of developed software, the software must undergo test phase even it is not for calculating reliability. We believe that carefully designed random tests and advanced test methodologies can provide estimates of the lower bound of the reliability that will be experienced in actual use.

For the convenience of explanation, we will show the example of a highly reliable system. The number of observed failures during test is expected to be zero because when we find an error we will debug the responsible code and restart the testing. So the concept of software

failure probability implies the degree of expectation of fault due to the software which shows no error in testing phase.

The reliability is assessed to be no worse than the result of this test with a certain confidence. That is, testing provides the lower bound of the reliability of software. Conventional method to calculate the required number of test can be easily derived as follows. Using the random variable T as the number of tests before the first failure and U as the required number of tests, the confidence level C can be expressed as follows:

$$C = prob(T \le U)$$
$$= \sum_{t=1}^{U} p(1-p)^{t-1} = p[\frac{1-(1-p)^{U}}{1-(1-p)}] \qquad (1)$$

The failure probability is denoted p. We can solve this equation for U as follows:

$$U = \frac{\ln(1-C)}{\ln(1-p)} \quad (2)$$

According to Equation (2), the higher target reliability and higher confidence level implies a greater number of test cases. The testing might demand an impractical number of test cases in some ultra-high reliable systems. Table 1 shows the required number of tests for some failure rates and confidence levels. For example, if we want to show that the expected failure rate is lower than 10^{-6} with 90% confidence level, we have to test the software for 2.3×10^{6} cases without failure. However, we expect that this problem of large number of test cases can be resolved through fully automated testing and parallel testing. Especially, in the case of sequential processing software, which has no feedback interaction with user or other system, the test automation method will be a strong candidate for reducing the test burden. The validity of test-based evaluation is dependent on the coverage of test cases. The test cases should represent the inputs which will be encountered in actual use.

Table 1. Required number of test cases

p \ C	50%	90%	99%
10^{-2}	6.90×10	2.29×10^{2}	4.58×10^{2}
10^{-3}	6.93×10^{2}	2.30×10^{3}	4.60×10^{3}
10^{-4}	6.93×10^{3}	2.30×10^{4}	4.60×10^{4}
10^{-5}	6.93×10^{4}	2.30×10^{5}	4.61×10^{5}
10^{-6}	6.93×10^{5}	2.30×10^{6}	4.61×10^{6}
10^{-7}	6.93×10^{6}	2.30×10^{7}	4.61×10^{7}
10^{-8}	6.93×10^{7}	2.30×10^{8}	4.61×10^{8}

C. Estimating the effect of software diversity and V&V efforts

In order to assess the expected failure rate of software, we also should consider the efforts on the lifecycle of software [8]. Previous experimental researches showed that the application of formal methods to the software development process and the usage of mathematical verification of the software specifications could reduce the possibility of fault due to design failure [9]. As explained in above section, the failure rate of safety-critical software represents the degree of expectation of failure or the possibility of failure. As we expect that the application of software verification and validation (V&V) methodologies could reduce the number of potential faults remained in the software, this effect should be reflected on the probability estimation of basic events. That is, the quantification of the rigidity of software V&V should be performed through PSA process.

Formal methods including formal specification technique are particular examples of software V&V processes. Formal methods express the functional requirements of a computer system in logic symbols based on set theory. The notion of mathematical proof is the most important effect of these methods. Even though the extent of this kind of proofs is limited, they are still one of the strongest aids for developing extremely high reliable software. Welbourne [10] stated that these methods had been widely shown to be feasible in other industries. Besides these formal methods, there are many kinds of approach for improving the quality of software production.

Diversity of software plays an important role in fault tolerance of digital systems. Diversity can be implemented without modification of hardware components by installing two or more versions of software which are developed by different teams because we expect that faults will tend to be different so failures can be masked by a suitable voting mechanism. However, design diversity does bring an increase in reliability compared with single versions, but this increase is much less than what completely independent failure behavior would imply. Littlewood and Strigini [6] also insist that this independence assumption is often unreasonable in practice. Therefore, the degree of dependence must be estimated for each particular case.

D. Estimating the coverage of fault-tolerant features

In the nuclear industry, we should especially concentrate on watchdog timer and duplication technique used in fault-tolerant system. They are the simplest way to establish the fault-tolerant system and already applied to some nuclear applications. When we analyze the duplication, we

should carefully consider the common cause failures among duplicated components.

Microprocessors and software technologies make it possible to implement various fault-tolerant mechanisms which check the integrity of the system itself and to monitor the integrity of each other. The experience shows that these fault-tolerant mechanisms effectively detect the fault on the system but they are not perfect. Digital systems have various kinds of fault and the coverage of the fault-tolerant mechanism is limited.

We expect that this aspect can be expressed using the concept of the coverage factor. In the fault tree, this coverage must be considered. Because the safety systems in nuclear plants adopt 'fail-safe' concept, the coverage factor plays a critical role on assessing the safety of the digital system. That is, the watchdog device is widely adopted for the fault-tolerance feature of safety systems in nuclear power plants to generate trip signal at the failure of microprocessor-based devices.

For the convenience of explanation, consider the simplest example of a watchdog timer application. It is illustrated in Figure 3. When watchdog timer detects the failure of processor, it will isolate the power.

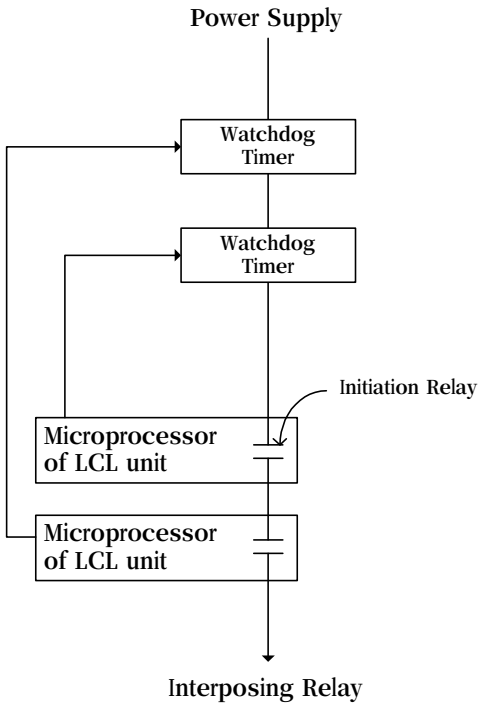


Figure 3. Schematic diagram of a typical watchdog timer application

We categorize watchdog timer failures into two groups: The first is the failure of watchdog timer itself (recovery failure). The second is that watchdog timer cannot detect the failure of microprocessor (functional failure). The symbol *p*, *c* and *w* represent, the coverage factor and,

respectively.

For the illustration of the effect of the coverage factor, assume that the probability of processor failure and the probability of watchdog failure are equal to 10^{-3} and 10^{-7}. The value of 10^{-3} failure/demand is the typical level of failure rate for programmable logic processors. The value of 10^{-7} failure/demand represents the typical failure rate of simple circuit and contact. If the watchdog mechanism is perfect, the reliability of microprocessor-based device will be negligible and the system reliability totally depends on the reliability of the watchdog device. In this case, the system unavailability is 10^{-20}. If the coverage equals zero, the system unavailability is 10^{-6}. Generally, it is well known that the coverage of the watchdog timer is not so high because it is the simplest method among the fault-tolerant mechanisms.

The remaining problem is the estimation of the value of coverage factor. Unfortunately, there is no widely-accepted method except experiment. However, we expect that the simulation using a fault injection method will be promising for estimating the coverage factor. The knowledge of domain experts will also be helpful. Before the credible methodology is developed, even though the exact coverage of the watchdog timer is hard to evaluate, we can establish the lower bound of the coverage using similar method explained in software failure probability.

E. Estimating the common cause failure probability in hardware

The importance of precise estimation of the common cause failure (CCF) of digital equipment should be emphasized. As explained above, the application of digital equipment to the safety-critical system will induce more risk concentration. In the case of adopting the same equipment as the redundancy, this concentration will be more critical. Even the products from different vendors do not guarantee the independence of faults. Global standardization and large manufacturer in electric part market lead to produce similar digital hardware products by different vendors.

In the case that operating experience is enough and generic CCF data is available, we can use the conventional methodologies (β factor approach or Multiple Greek Letter approach). However, in the case of newly designed dedicated systems such as safety-critical calculators in nuclear power plants, generic data is unavailable. Thus, the development of new and precise estimation methodology for the CCF factor of digital hardware is required.

F. Modeling the interactions between hardware and software

Conventionally, the research on the hardware reliability and that of software reliability has

been independently performed. Therefore, there are some attempts which estimate the reliability of digital system by calculating that of hardware and software separately [11]. In this case, however, we cannot evaluate the effect of interactions between hardware and software.

Most microprocessor-based systems have fault-tolerant mechanisms which are based on hardware and software. They make the system complex but are expected to reduce the number of system failures appear on the outside. There exists obvious effect of hardware fault masking by software. That is, a substantial number of faults do not affect the program results for several reasons: faults whose errors are neutralized by the next instructions, faults affecting the execution of instructions that do not contribute to the benchmark results, and faults whose errors are tolerated by the semantic of the benchmark under execution. He insists that these interactions might be very important factors to estimate the dependability of systems. Therefore, the system dependability measurement technique should not consider software and hardware separately and the effect of interaction should be considered properly because even a small change of system fault coverage value can affect the system dependability.

When we consider aging effect on hardware, the problem becomes more complex. The aging effect will induce slight changes on hardware. By some software, the system will make faulty output but by the other software, it will not.

Clearly, the modeling of interactions between hardware and software requires much further and extensive investigation. For more realistic results, however, these complex interactions should be considered properly.

4. Summary and Conclusions

In this paper, we introduce the PSA methodology for the digital equipment and we also summarize the factors which should be considered in modeling digital systems for PSA as follows:

– Modeling the multi-tasking of digital systems,

– Estimating software failure probability,

– Estimating the effect of software diversity and V&V efforts,

– Estimating the coverage of fault-tolerant features,

– Estimating the CCF probability in hardware, and

– Modeling the interactions between hardware and software.

We expect that the proper consideration of these factors will make PSA result more realistic and useful.

Software failure in digital safety-critical system induces very severe problems on assessing system safety. It might remove the redundancy effect if the same software is installed in redundant systems. We also cannot detect the failure of software by hardware-based monitoring mechanism. In order to get the reasonable result of safety assessment, the software failure probability should not be ignored.

Because of its simplicity, the reliability of a watchdog device is extremely higher than that of a microprocessor-based device. If we assume that the watchdog mechanism is perfect, the reliability of microprocessor-based device will be negligible and the system reliability totally depends on the reliability of watchdog device and non-monitored devices. We proposed the methodology in order to avoid such an unrealistic analysis using the concept of coverage factor.

Due to the complexity of microprocessor-based system, there are lots of unsolved problems. Further investigation on these problems is strongly recommended. Major problems which are not mentioned in this study can be listed as follows:

– Failure mode of digital system,

– Environmental effects, and

– Digital system induced initiating events including human errors.

Last but not least, even though we cannot quantify the safety of digital systems in a very accurate manner, the active design feedback of the insight, which comes from quantitative and qualitative approaches of PSA, should be encouraged. For example, the improved design by which the coverage of watchdog mechanism is enlarged to the extent of input/output modules will contribute toward reducing the probability of system failure. Properly designed on-line testing and monitoring mechanism will also improve the system integrity by reducing the inspection interval.

Reference

[1] J.L. Mourlenvat, A. Parry, J.F. Petetrot and J.F. Aschenbrenner, “Instrumentation and Control Revamping,” Nuclear Technology, Vol. 92, pp. 300-308, December 1990.

[2] G. Ives, “Digital Systems: Review of safety critical applications,” Nuclear Engineering International, pp. 37-40, April 1994.

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

[4] National research council, Digital Instrumentation and Control Systems in Nuclear Power Plants, National Academy Press, Washington, D.C., 1997.

[5] J.L. Rouvroye & A.C. Brombacher, "New quantitative safety standards: different techniques, different results?" Reliability Engineering in System Safety, Vol. 66, p. 121-125, 1999.

[6] B. Littlewood and L. Strigini, "Validation of ultrahigh dependability for software based systems," Communications of ACM, Vol. 36, No. 11, 1993.

[7] R.W. Butler and G.B. Finelli, "The infeasibility of quantifying the reliability of life-critical real-time software," IEEE Transactions on software engineering, vol19. no.1, 1993.

[8] H.G. Kang, T. Sung et al., A Technical Survey on Issues of the PSA of digital I&C systems, KAERI/AR-560/2000, 2000.

[9] H. Saiedian, "An Invitation to Formal Methods," Computer, April 1996.

[10] D. Welbourne, "Safety Critical Software in Nuclear Power," The GEC Journal of Technology, Vol. 14, No. 1, 1997.

[11] W. Bastl and H.W. Bock, "German qualification and assessment of Digital I&C systems important to safety," Reliability Engineering and System Safety, Vol. 59, p. 163-170, 1998.