

Issues in System Reliability and Risk Model

Hyun Gook Kang

Integrated Safety Assessment Division
Korea Atomic Energy Research Institute
1045 Daedeok-daero, Yuseong-gu, Daejeon, 305-353, Korea
hgkang@kaeri.re.kr

The application of large-scale digital or computer systems involves many components, elements, and modules. System reliability and safety need to be calculated no matter how complicated is the structure. Estimation of system reliability/safety provides useful information for system design and verification. Risk allocation to the designed system in a balanced manner is an application example.

The most conservative method for estimating system failure probability is summing up failure probabilities of components. The result of this conservative calculation equals system failure probability for a series of independent components. Reliability of a series of components is the lower boundary of system reliability. Redundant components and standby components are considered in order to estimate realistic reliability. A module consists of many components and a system consists of several kinds of modules. An analytic calculation of the reliability or risk based on the information of module structure and component reliabilities is relatively simple. An analytic model for the reliability or risk estimates at the system level is complex and sometimes difficult to develop.

Conventional methodologies are used 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. Even the use of the lower boundary value of system reliability is possible for simplicity.

More complicated relationships among system functions should be modeled for the accurate and realistic estimation of risk from safety-critical systems. The faults in an advanced digital system are monitored by a self-monitoring algorithm and recovered before the fault causes system failure. Protecting a system from catastrophic damage is possible even though it is not possible to recover the fault. Multiple-channel processing systems might have cross-monitoring functions and independent heartbeat-monitoring equipment can also be installed in the systems. Intelligence and flexibility from microprocessors and software successfully accommodates these sophisticated reliability-enhancing mechanisms.

Clarification and definition of reliability or risk-modeling objectives are very important because the hazard state varies along this definition. Failure of a status indication lamp or trouble in a cabinet lock button is considered as part of system failure for maintenance purposes. Faults which do not disturb shutdown signal generation will not be considered for risk estimation of safety-critical systems, such as an automatic shutdown system in a nuclear power plant.

Modern digital technologies are expected to significantly improve both economical efficiency and safety of plants owing to the general progress of instrumentation and control (I&C) technologies for process engineering, such as computer technology, control engineering, data processing and transfer technology, and software technology. The assessment of digital system safety becomes a more sensitive issue rather than maintenance reliability, when a digital technology is applied to a safety-critical function.

Economical efficiency improvement due to digital applications seems clear, while safety improvement is not well accepted. There are still many arguable safety issues, even though the use of digital equipment for safety-critical functions provides many advantageous features.

Digital signal-processing system unavailability severely affects total plant safety because many safety signals are generated by the same digital system [1]. A sensitivity study showed that the protection and monitoring system is the most important system for the safety of a Westinghouse AP1000 nuclear power plant [2]. The potential impact of digital system malfunctions on core damage frequency in the advanced boiling water reactor is high [3].

Safety assessment, based on unavailability estimation, can quantitatively show improvement because it demonstrates that a balanced design has been achieved, by showing that no particular class of accident of the system makes a disproportionate contribution to overall risk. The importance of probabilistic risk assessment (PRA) for safe digital applications is pointed out in the HSE guide [4]. The PRA of digital safety-critical systems plays the role of a decision-making tool and has sufficient accuracy.

Characteristics of digital systems from a safety assessment viewpoint are:

- The utilization of hardware is determined by software and inputs.
- The system is multi-purpose.
- The failure modes are not well defined.
- Software might hide the transient faults of hardware.
- Software fails whenever it executes the faulty part of the code.
- Greater effort in the management of software quality can cause a lower expectation for software failure in the operation phase, while quantification is still challenging.
- Various monitoring and recovery mechanisms are adopted, but their coverage is not well defined.
- Apparently different components might cause common-cause failure (CCF) because electronic components consist of a lot of small modules, which are manufactured in a globally standardized environment.
- Digital systems are more sensitive to environmental conditions, such as ambient temperature, than conventional analog systems.
- There might be no warning to operators when a system fails.

- The system failure might cause the blockage of safety-critical information from field to operators.
- New initiating events may be induced by digital system failure.

Many assumptions are used for quantitative analysis of a digital I&C system. Some are intentionally used for analysis simplicity, while others are caused by failing to show enough caution. Unreasonable assumptions result in unreasonable safety evaluation. Fault-free software and perfect coverage of fault tolerance mechanisms are typical examples of unreasonable assumptions.

The characteristics of digital applications are different from those of conventional analog I&C systems, because their basic elements are microprocessors and software, which make the system more complex to analyze. There are important issues in digital system safety analysis which are complicated and correlated [5]. Several system reliability and risk estimation methodologies and important issues related to the reliability and safety modeling of large digital systems are described in this chapter. A brief introduction to these methodologies for reliability calculation or hazard identification is given in Section 2.1. The related issues are categorized into six groups from the viewpoint of PRA: the modeling of the multi-tasking of digital systems, the estimation of software failure probability, the evaluation of fault tolerance mechanisms, the assessment of network safety, the assessment of human failure probability, and the assessment of CCF (Sections 2.2 to 2.7).

2.1 System Reliability Models

Failure of either component will result in system failure for a system composed of two independent modules, in the same way as for a component reliability model. The system is represented by a series of blocks, as shown in a reliability block diagram (RBD) (Figure 2.1(a)). If λ_1 and λ_2 are the hazard rates of the two modules, the system hazard rate will be $\lambda_1 + \lambda_2$. The reliability of the system is the combined probability of no failure of either modules: $R_1 R_2 = \exp[-(\lambda_1 + \lambda_2)t]$. In the case of s -independent modules, for a series of n modules:

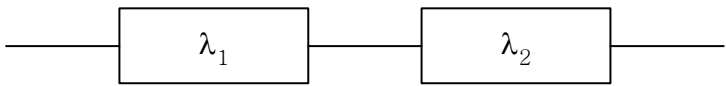
$$R = \prod_{i=1}^n R_i \quad (2.1)$$

This is the simplest basic model. Parts count reliability prediction is based on this model. The failure logic model of the system in real applications is more complex, if there are redundant subsystems or components.

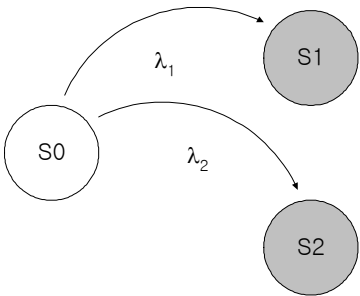
The Markov model is a popular method for analyzing system status. The Markov model provides a systematic method for analysis of a system which consists of many modules and adopts a complex monitoring mechanism. The Markov model is especially useful for a more complicated transition among the system states or the repair of the system. A set of states and probabilities that a system will move from one state to another must be specified to build a Markov

model. Markov states represent all possible conditions the system can exist in. The system can only be in one state at a time. A Markov model of the series system is shown in Figure 2.1(b). State $S0$ is an initial state. States $S1$ and $S2$ represent the state of module 1 failure and module 2 failure, respectively. Both are defined as hazard states.

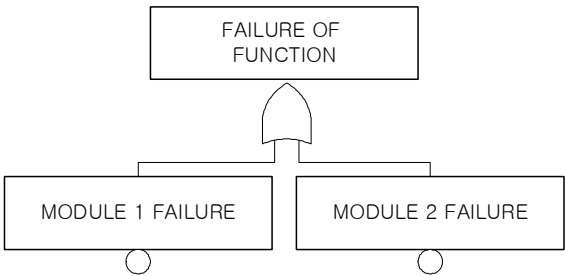
Fault tree modeling is the most familiar tool for analysis staff, whose logical structure makes it easy for system design engineers to understand models. A fault tree is a top-down symbolic logic model generated in the failure domain. That is, a fault tree represents the pathways of system failure. A fault tree analysis is also a powerful diagnostic tool for analysis of complex systems and is used as an aid for design improvement.



(a) Reliability block diagram



(b) Markov model



(c) Fault tree model

Figure 2.1. Series system

The analyst repeatedly asks, “What will cause a given failure to occur?” in using backwards logic to build a faulttree model. The analyst views the system from a top-down perspective. This means he starts by looking at a high-level system failure and proceeds down into the system to trace failure paths. Fault trees are generated in the failure domain, while reliability diagrams are generated in the success domain. Probabilities are propagated through the logic models to determine the probability that a system will fail or the probability the system will operate successfully (*i.e.*, the reliability). Probability data may be derived from available empirical data or found in handbooks.

Fault tree analysis (FTA) is applicable both to hardware and non-hardware systems and allows probabilistic assessment of system risk as well as prioritization of the effort based upon root cause evaluation. An FTA provides the following advantages [6]:

1. Enables assessment of probabilities of combined faults/failures within a complex system.
2. Single-point and common-cause failures can be identified and assessed.
3. System vulnerability and low-payoff countermeasures are identified, thereby guiding deployment of resources for improved control of risk.
4. This tool can be used to reconfigure a system to reduce vulnerability.
5. Path sets can be used in trade studies to compare reduced failure probabilities with increases in cost to implement countermeasures.

2.1.1 Simple System Structure

The probability of failure (P) for a given event is defined as the number of failures per number of attempts, which is the probability of a basic event in a fault tree. The sum of reliability and failure probability equals unity. This relationship for a series system can be expressed as:

$$\begin{aligned}
 P &= P_1 + P_2 - P_1P_2 \\
 &= (1 - R_1) + (1 - R_2) - (1 - R_1)(1 - R_2) \\
 &= 1 - R_1R_2 \\
 &= 1 - R
 \end{aligned} \tag{2.2}$$

The reliability model for a dual redundant system is expressed in Figure 2.2. Two s -independent redundant modules with reliability of R_1 and R_2 will successfully perform a system function if one out of two modules is working successfully. The reliability of the dual redundant system, which equals the probability that one of modules 1 or 2 survives, is expressed as:

$$\begin{aligned}
 R &= R_1 + R_2 - R_1R_2 \\
 &= e^{-\lambda_1 t} + e^{-\lambda_2 t} - e^{-(\lambda_1 + \lambda_2)t}
 \end{aligned} \tag{2.3}$$

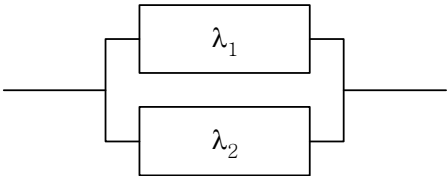
This is often written as:

$$R = 1 - (1 - R_1)(1 - R_2) \tag{2.4}$$

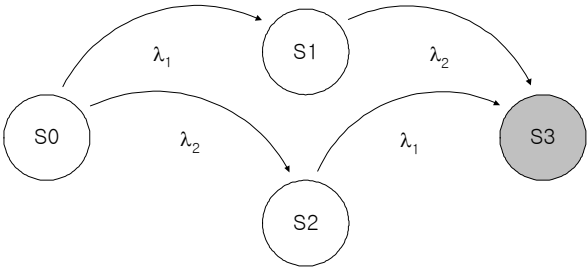
$$1 - R = (1 - R_1)(1 - R_2) \tag{2.5}$$

In the case of s -independent modules, for n redundant modules, the reliability of a system is generally expressed as:

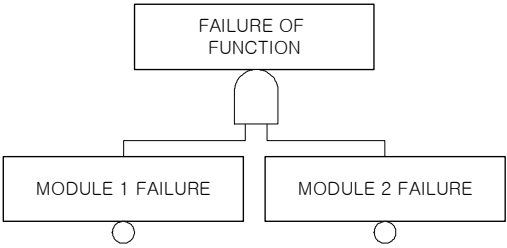
$$R = 1 - \prod_{i=1}^n (1 - R_i) \tag{2.6}$$



(a) Reliability block diagram



(b) Markov model



(c) Fault tree model

Figure 2.2. Dual redundant system

2.1.2 Complicated System Structure

Not all systems can be modeled with simple RBDs. Some complex systems cannot be modeled with true series and parallel branches. Module 2 monitors status information from module 1 and module 2 automatically takes over the system function when an erroneous status of module 1 is detected in a more complicated system. The system is conceptually illustrated in Figure 2.3.

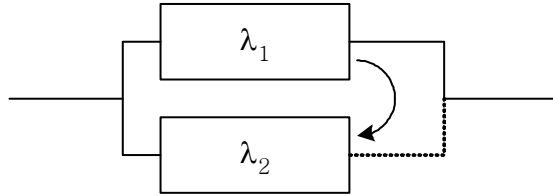


Figure 2.3. Standby and automatic takeover system

In this case, using a successful takeover probability of the module 2, μ , the reliability of the system is generally expressed as:

$$\begin{aligned} 1 - R &= (1 - R_1) \{ (1 - R_2) + (1 - \mu) - (1 - R_2)(1 - \mu) \} \\ &= (1 - R_1) \{ (1 - R_2)\mu + (1 - \mu) \} \end{aligned} \quad (2.7)$$

The Markov model is shown in Figure 2.4. A fault tree is shown in Figure 2.5.

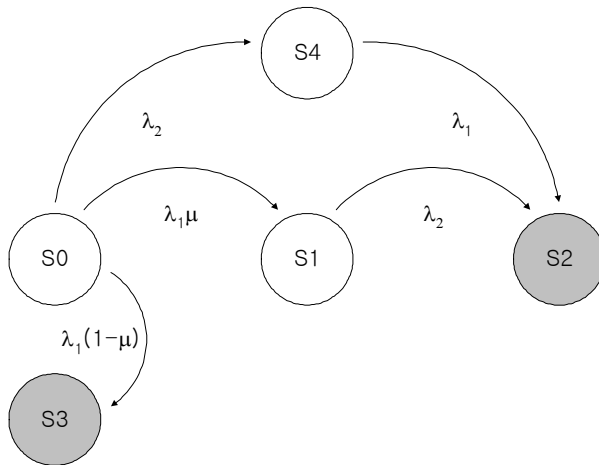


Figure 2.4. Markov model for standby and automatic takeover system

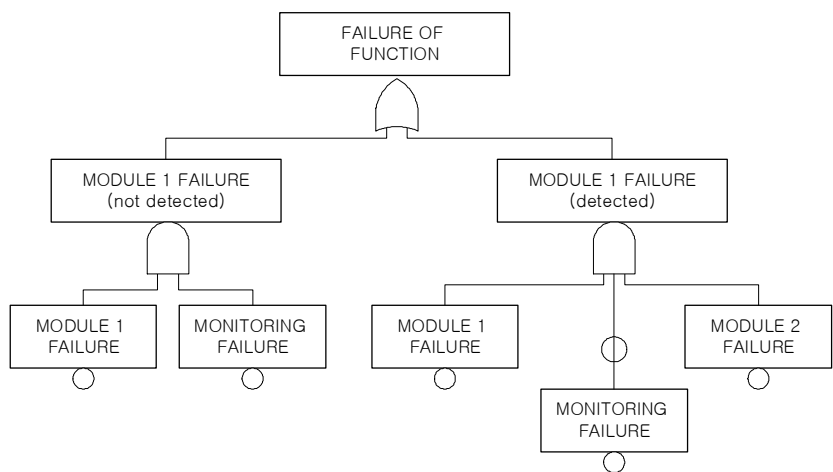


Figure 2.5. Fault tree for standby and automatic takeover system

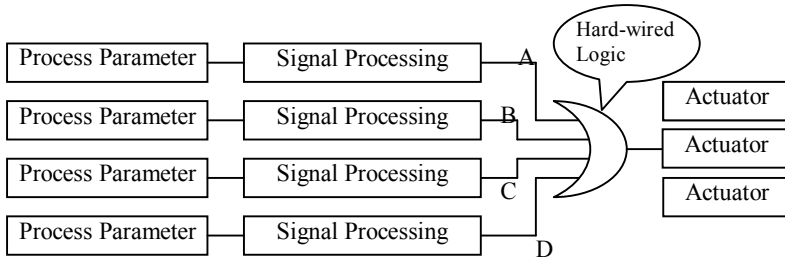
2.2 Modeling of the Multi-tasking of Digital Systems

2.2.1 Risk Concentration

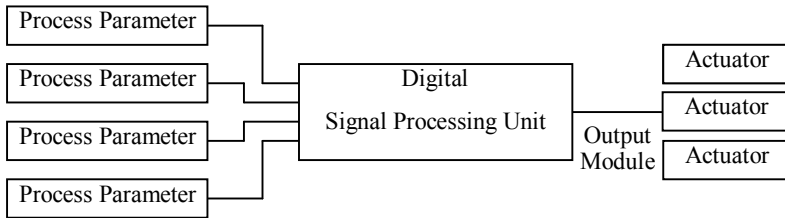
Microprocessors and software technologies make the digital system multi-functional because a system performs several sequential or conditional functions. This multi-tasking feature is represented in safety assessment because it will cause risk concentration and deteriorate the reliability of the system.

The use of a single microprocessor module for multiple safety-critical functions will cause severe concentration of risk in the single microprocessor. Safety-critical applications have adopted a conservative design strategy, based on functional redundancies. However, software programs of these functions are executed by one microprocessor in the case of digital systems. The effects of multi-tasking on safety should be carefully modeled and evaluated in order to compare the developed digital system with the conventional analog system.

A typical example for finding two ways of handling diverse process parameters and functional redundancy is shown in Figure 2.6, when considering the main steam line break accident in a nuclear power plant. Several parameters affected by this accident will move to an abnormal region. First, the “Low steam generator pressure” parameter triggers the output signal A. As time goes on, the parameters of “Low pressurizer pressure,” “Low steam generator level,” “Reactor overpower” will trigger the output signals B, C, and D, respectively. In a conventional analog circuit system (Figure 2.6(a)), the first triggered signal A makes trip circuit breakers open and initiates reactor shutdown. Signals B, C, and D are sequentially



(a) Typical process of signal processing using conventional analog



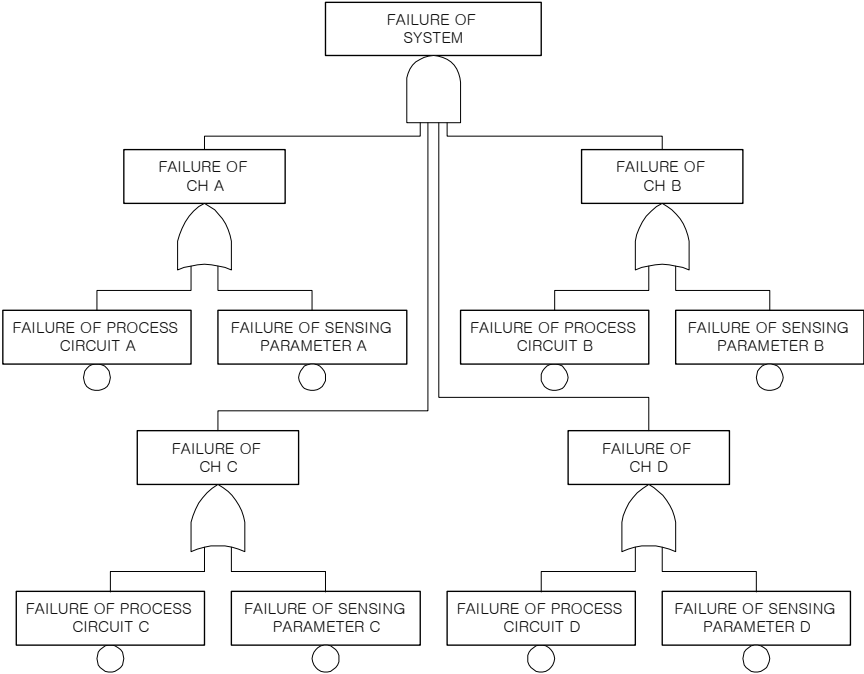
(b) Typical process of signal processing using digital units

Figure 2.6. Schematic diagram of signal processing using analog circuit and digital processor unit

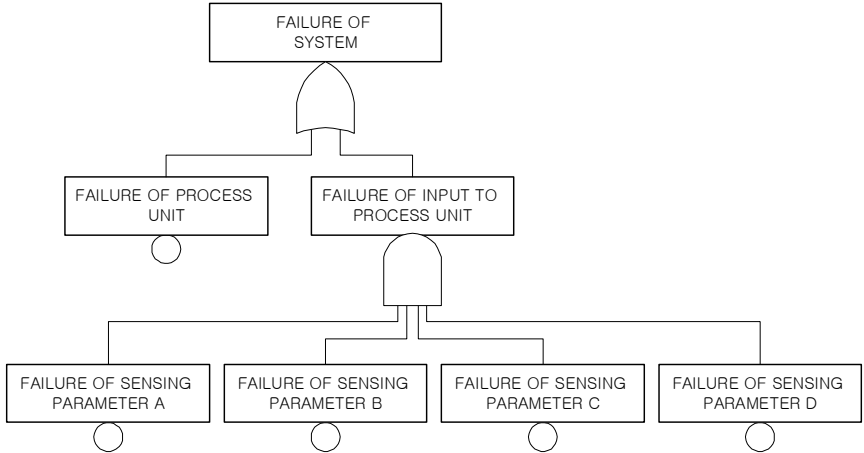
generated if the signal processing circuits for parameter A fail. However, parameters A, B, C, and D use the same equipment for signal-processing in the case of digital system (Figure 2.6 (b)). There is no functional backup if the digital signal-processing unit fails.

The risk concentration on a processing unit is demonstrated in Figure 2.7 by fault trees for the systems in Figure 2.6. Component reliabilities should be carefully analyzed. Self-monitoring and fault-tolerant mechanisms for these components should be strengthened in the design phase to improve system reliability.

There are two or more duplicated trip channels in safety-critical applications that are not functional backups and are vulnerable to the CCF. The dominant contributor to system unavailability is the CCF of digital modules in 2-out-of-3 trains voting logic (Figure 2.8). The importance of precise estimation of digital equipment CCF should be emphasized. Products from different vendors do not guarantee the independence of faults, since global standardization and the large manufacturer in the electronic parts market lead to similar digital hardware products.



(a) The fault tree model of the example in Figure 2.6(a)



(b) The fault tree model of the example in Figure 2.6(b)

Figure 2.7. The fault trees for the systems shown in Figure 2.6

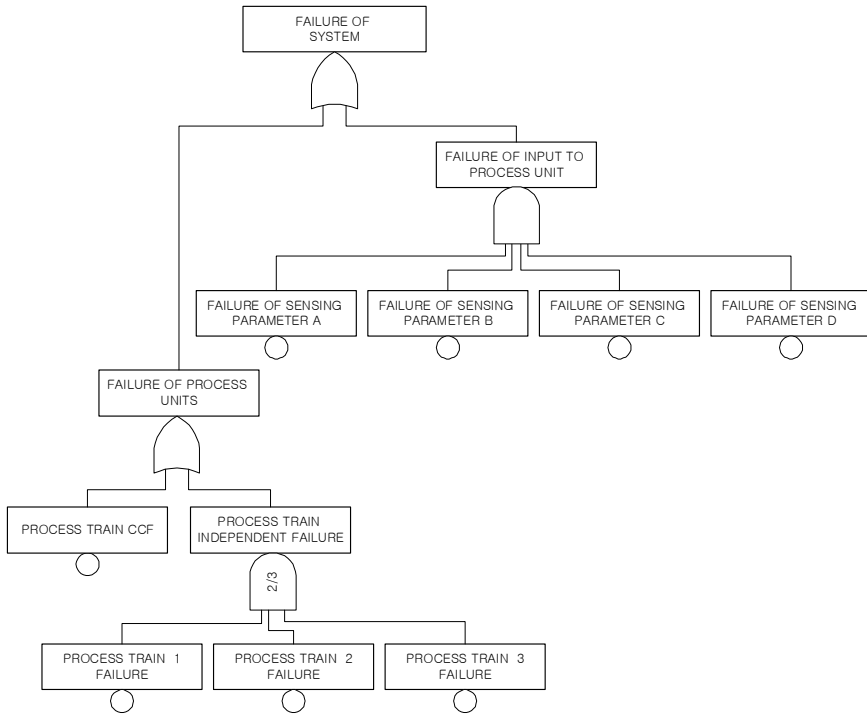


Figure 2.8. The fault tree model of a three-train signal-processing system which performs 2-out-of-3 auctioneering

2.2.2 Dynamic Nature

Static modeling techniques, such as a classical event tree and a fault tree, do not simulate the real world without considerable assumptions, since the real world is dynamic. Dynamic modeling techniques, such as a dynamic fault tree model, accommodate multi-tasking of digital systems [7], but are not very familiar to designers.

Estimating “how many parameters will trigger the output signals within the specific time limit for specific kind of accident” is very important, in order to build a sophisticated model with the classical static modeling techniques. Several assumptions, such as the time limit and the severity of standard accidents are required. Parameters for several important standard cases should be defined. For example, a reactor protection system should complete its actuation within 2 hours and the accident be detected through changes in several parameters, such as “Low steam generator pressure,” “Low pressurizer pressure,” and “Low steam generator level” in the case of a steam line break accident in nuclear power units. The digital system also provides signals for human operators. The processor module in some cases generates signals for both the automated system and human operator. The effect of digital system failure on human operator action is addressed in Section 2.6.

2.3 Estimation of Software Failure Probability

Software is a basis for many of the important safety issues in digital system safety assessment. This section discusses the effect of safety software on safety modeling of a digital system. Software-related issues are dealt with in Chapters 4 and 5 in a more detailed manner.

2.3.1 Quantification of Software Reliability

There is much discussion among software engineering researchers about whether software failure can be treated in a probabilistic manner [8]. Software faults are design faults by definition. That is, software is deterministic and its failure cannot be represented by “failure probability.” However, software could be treated based on a probabilistic method because of the randomness of the input sequences, if the software of a specific application is concerned. 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. A software failure occurs when the input trajectory enters an error crystal.

Prediction of software reliability using a conventional model is much harder than for hardware reliability. Microprocessor applications fail frequently when first installed and then become reliable after a long sequence of revisions. The software reliability growth model is the most mature technique for software dependability assessment, which estimates the increment of reliability as a result of fault removal. The repeated occurrence of failure-free working is inputted into probabilistic reliability growth models, which use these data to estimate the current reliability of the program, and to predict how the reliability will change in the future. However, this approach is known to be inappropriate in safety-critical systems since the fixes cannot be assumed effective and the last fix may have introduced new faults [9].

The lower limit of software-failure probability estimated conservatively by testing can be an alternative. The feasibility of reliability quantification of safety-critical software using statistical methods is not accepted by some researchers because exorbitant amounts of testing when applied to safety-critical software are required [10]. However, the developed software must undergo a test phase to show integrity, even if it is not for calculating reliability. Carefully designed random tests and advanced test methodologies provide an estimate of the lower bound of the reliability that is experienced in actual use.

The number of observed failures of highly reliable software during the test is expected to be zero because found errors will be debugged in the corresponding code and the test will be performed again. The concept of software failure probability implies the degree of fault expectation due to software that showed no error in the testing phase. The conventional method to calculate the required number of tests is easily derived. The confidence level C is expressed using the random variable T as the number of tests before the first failure and U as the required number of tests as:

$$C = \Pr(T \leq U)$$

$$= \sum_{t=1}^U p(1-p)^{t-1} = p \left[\frac{1-(1-p)^U}{1-(1-p)} \right] \quad (2.8)$$

The failure probability is denoted p . This equation can be solved for U as:

$$U = \frac{\ln(1-C)}{\ln(1-p)} \quad (2.9)$$

An impractical number of test cases may be required for some ultra-high reliable systems. A failure probability that is lower than 10^{-6} with 90% confidence level implies the need to test the software for more than 2.3×10^6 cases without failure. Test automation and parallel testing in some cases is able to reduce the test burden, such as sequential processing software which has no feedback interaction with users or other systems. The validity of test-based evaluation depends on the coverage of test cases. Test cases represent the inputs which are encountered in actual use. This issue is addressed by the concept of reliability allocation [11]. The required software reliability is calculated with target reliability of the total system. The cases of no failure observed during test are covered by Equations 2.8 and 2.9. Test stopping rules are also available for the cases of testing restart after error fixing [11]. The number of needed test cases for each next testing is discussed in a more detailed manner in Chapter 4.

2.3.2 Assessment of Software Development Process

The development process of software is considered in order to assess the expected software failure rate. The application of formal methods to the software development process and usage of mathematical verification of software specifications reduces the possibility of failures due to design faults [12]. The number of remaining potential faults in software is reduced by using software verification and validation (V&V) methodologies. This effect is reflected on the probability estimation of basic events. Thus, the quantification of rigidity of software V&V is performed through the PRA process.

Formal methods, including the formal specification technique, are examples of software V&V processes. Formal methods use ideas and techniques from mathematical or formal logic to specify and reason about computational systems [13, 14]. Formal methods are one of the strongest aids for developing highly reliable software, even though the extent of this kind of proofs is limited. These methods had been widely shown to be feasible in other industries [15]. There are many kinds of approaches for improving the quality of software production besides these formal methods.

The Bayesian belief network (BBN) can be used for estimating the effectiveness of these quality-improving efforts in a more systematic manner [16, 17]. Applying the BBN methodology to the PRA of digital equipment is helpful to integrate many aspects of software engineering and quality assurance. This

estimation is performed in consideration of various kinds of activities from each stage of software lifecycle. Difficulties in establishing the BBN include topology and data gathering.

2.3.3 Other Issues

Issues in software reliability are diversity in software codes and hardware–software interaction, in addition to quantification and lifecycle management. Diversity of software plays an important role in fault tolerance of digital systems. Diversity is implemented without modification of hardware components by installing two or more versions of software which are developed by different teams. Faults are expected to be different. As a result, failures can be masked by a suitable voting mechanism. Proving the high reliability of software contains many difficulties. Diversity is helpful in reducing the degree of proof.

Design diversity brings an increase in reliability compared with single versions. This increase is much less than what completely independent failure behavior would imply. The assumption of independence is often unreasonable in practice [18]. Therefore, the degree of dependence must be estimated for each particular case.

Estimation of digital system reliability by calculating the reliability of hardware and software separately [19] does not reflect the effect of hardware–software interactions. An obvious effect of hardware fault masking by software has been reported [20]. A substantial number of faults do not affect program results. Hardware–software interaction may be a very important factor in estimating the dependability of systems. Therefore, the effect of such interactions should be considered.

The interaction problem becomes more complex when aging of hardware is considered. The aging effect induces slight changes in hardware. Different software may cause faulty output. The correlated effect of hardware design and software faults and the correlation between diverse hardware and software should also be considered. These considerations result in very complex and impractical models. The realistic modeling of interactions between hardware and software requires extensive investigation.

Software safety is an important issue in safety assessment of a large digital system. Further discussions regarding these issues are found in Chapters 4 and 5.

2.4 Evaluation of Fault Tolerance Features

Microprocessor and software technologies are used to implement fault-tolerant mechanisms and network communication, to improve efficiency and safety. Fault-tolerant mechanisms are implemented to check the integrity of system components.

Greater attention to watchdog timers and duplication techniques are needed. These are popular and simple ways to establish a fault-tolerant system in industry. Fault-tolerant mechanisms effectively enhance system availability, although their coverage is limited. Digital systems have various faults. Fault-tolerant mechanisms are unable to cover all the faults. The limitation of a fault-tolerant mechanism is

expressed using the concept of coverage factor, which must be considered in developing a fault tree. The coverage factor plays a critical role in assessing the safety of digital systems, if a safety-critical system adopts the “fail-safe” concept.

Watchdog devices have been widely adopted as a fault-tolerance feature for safety systems to generate a protective signal at failure of microprocessor-based devices. A simple example of a watchdog timer is illustrated in Figure 2.9. The power for signal generation will be isolated when the watchdog timer detects the failure of a processor. A fault tree of a watchdog timer application (Figure 2.9) is shown in Figure 2.10. Watchdog timer failures are categorized into two groups: failure of the watchdog timer switch (recovery failure); and failure of the watchdog timer to detect microprocessor failure (functional failure). Assume the values of p and w as 10^{-3} failure/demand and 10^{-7} failure/demand, respectively. They are reasonable failure probabilities for typical programmable logic processors and contact relays. System unavailability (Figure 2.10) equals 10^{-20} with a perfect watchdog mechanism ($c = 1$). System unavailability equals 10^{-6} if the coverage equals zero ($c = 0$). The effect of watchdog timer coverage estimation on the system unavailability is shown in Figure 2.11.

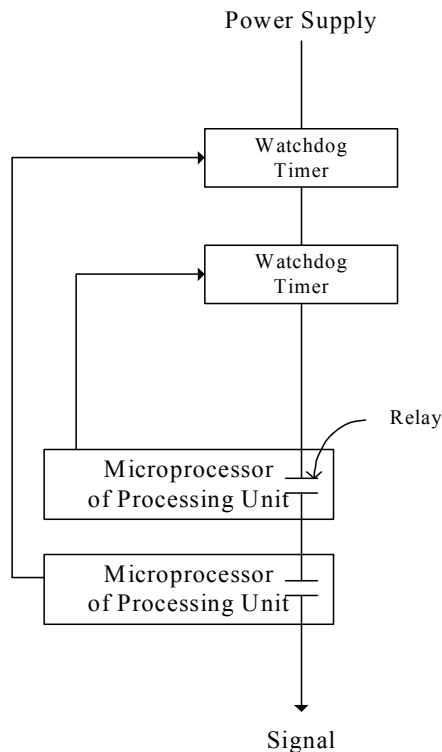


Figure 2.9. Schematic diagram of a typical watchdog timer application

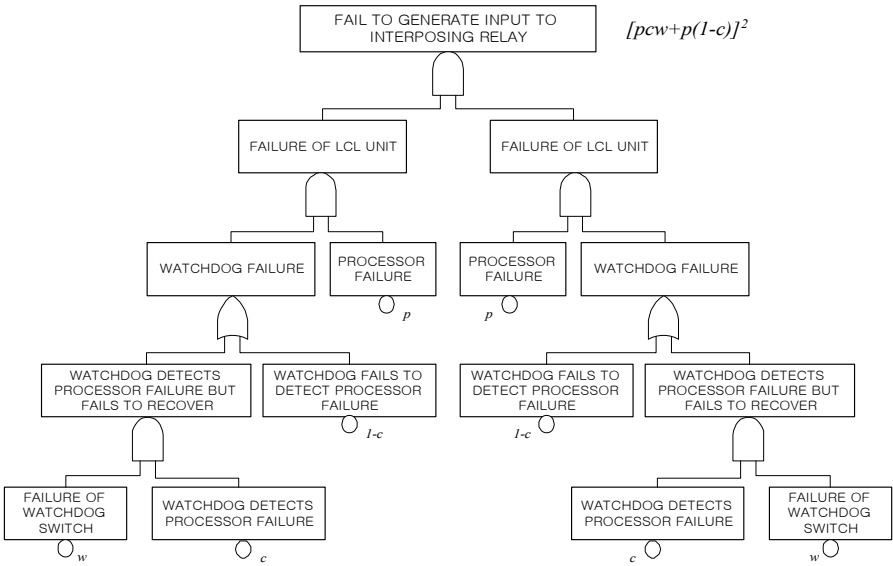


Figure 2.10. Fault tree model of the watchdog timer application in Figure 2.9 (p : the probability of processor failure, c : the coverage factor, w : the probability of watchdog timer switch failure)

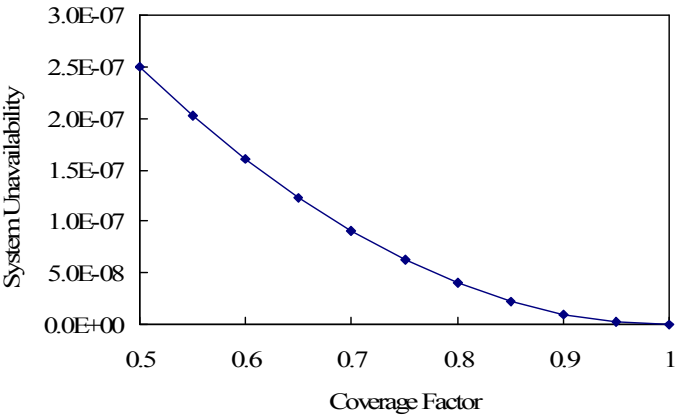


Figure 2.11. System unavailability along the coverage factor of watchdog timer in Figure 2.9

Coverage of the watchdog timer depends on the completeness of the integrity-checking algorithm. Fault coverage of the processor-based monitoring systems or fully duplicated backups are higher than those of watchdog timers because the former has a higher computing power, wider monitoring range, and more sophisticated algorithm.

Quantification of the coverage factor is very important. There is no widely accepted method except experiment for each specific system. Simulation using fault injection is one of the promising methods for estimating coverage factor. The expert knowledge might be used to estimate the rough bounds of the coverage.

2.5 Evaluation of Network Communication Safety

Application of the network communication technique is useful in reducing the cabling number when a system consists of many components and processor modules. The use of signal transmission components, such as fiber-optic modems and opto-couplers, is reduced by using network communication.

Distributed real-time systems have found widespread use in most major industries. Protocols and networking are at the heart of systems. Reliable data acquisition and distribution are essential. Safety-critical networks include information networks in nuclear plants, distributed battle management, intelligent transportation systems, distributed health care, and aviation traffic monitoring systems [21].

Network equipment and functions are closely monitored and controlled to ensure safe operation and prevent costly consequences. The probability of system failure increases as networks become more complex. Failure of any network element can cause an entire network break-down, and in safety-critical settings, the consequences can be severe. A well-known example of such failure is the 1990 nationwide AT&T network failure.

Metrics, such as the processing delay at each node, the capacity of each link, round trip propagation delay, the average queue length of messages awaiting service, utilization, throughput, node delay, and end-to-end delay metrics, have been used as performance criteria. Redundancy is also a metric that is often considered in network evaluations. Redundancy is considered a key feature of a safety-critical network, which drastically improves safety. However, redundancy may increase network complexity and increase network usage, especially in applications where network survivability is crucial [21].

Ethernet is the most widely deployed computer networking technology in the world. Applicability of common networks to safety-critical systems is impractical due to non-determinism. Ethernet cannot establish bounds on time required for a packet to reach its destination. This behavior is not acceptable in safety-critical systems, where a timely response is considered vital.

Safety network communication in the safety-critical system must be evaluated and proved, even though the technique provides many advantages for system development. Proving safety based on the "fail safe" concept is possible in some applications. The system is designed to perform safety actions when the network fails to transfer the information. This is intrinsic safety. Increased spurious transients and expense are also noted.

The probability that the system becomes unsafe due to network failure is evaluated to quantify the risk. Hazard analysis and the identification of paths which might lead the system to an unsafe state are performed, and the probabilistic quantification of each path is also required. Network failure is caused by defects in

hardware of network modules or a fault in network protocol, which is the basis of network software.

The main issues in network safety quantification are grouped into two categories: software and hardware. Network configuration and hazard states should be reflected and carefully modeled in a safety assessment model.

2.6 Assessment of Human Failure Probability

The PRA provides a unifying means of assessing the system safety, including the activities of human operators. Human factors and human failure probability are described in Chapters 7 and 8. Issues caused by the interaction of human operators and the digital information system are described in this section.

Two aspects are considered for human failure: the human operator as a generator of manual signals for mitigation when an accident happens, and the human operator as an initiator of spurious plant transients. Both are related to the digital system because its purpose is not only the generation of an automatic signal but also the provision of essential information, such as pre-trip and trip alarms to the operator. These are treated in a different manner from a PRA viewpoint, because the former is related to the accident mitigation, while the latter is related to accident initiation.

The multi-tasking feature of digital systems enables safety-critical signal generation systems to supply the alarms and key information to the human operator. Several functions, such as alarm generation, trip signal generation, and a safety-function-actuation signal generation are simultaneously performed for all the parameters by the digital processing system. An operator will not receive adequate data regarding plant status in the event of system failure.

Reasons for a specific safety function failure are expressed by these relationships (Figure 2.12). A signal generation failure implies human operator interception of an automatically generated signal or the concurrent occurrence of an automatic signal-generation failure and a manual signal generation failure, since a human operator or an automatic system generates safety-actuation signals. A human operator does not generate the signal if an automatic system successfully generates the safety signal. Human error probability (HEP) of a manual signal generation is a conditional probability, given that the automatic signal generation fails. This is an error of omission (EOO). The reason for automatic generation failure is the failure of processing systems or of instrumentation sensors. A processing system failure deteriorates the performance of a human operator, since it implies that the alarms from the processing system will not be provided to the operator. Concurrent failure of multiple redundant sensors also deteriorates human performance, since it causes the loss of corresponding sensor indications and failure of the automated signal generation system, causing the loss of corresponding alarms.

An operator may also wrongly turn off the automatically generated signal (Figure 2.12). This is an error of commission (EOC). The probability of an EOC is a conditional probability if the automatic system successfully generates a proper signal using the sound sensors.

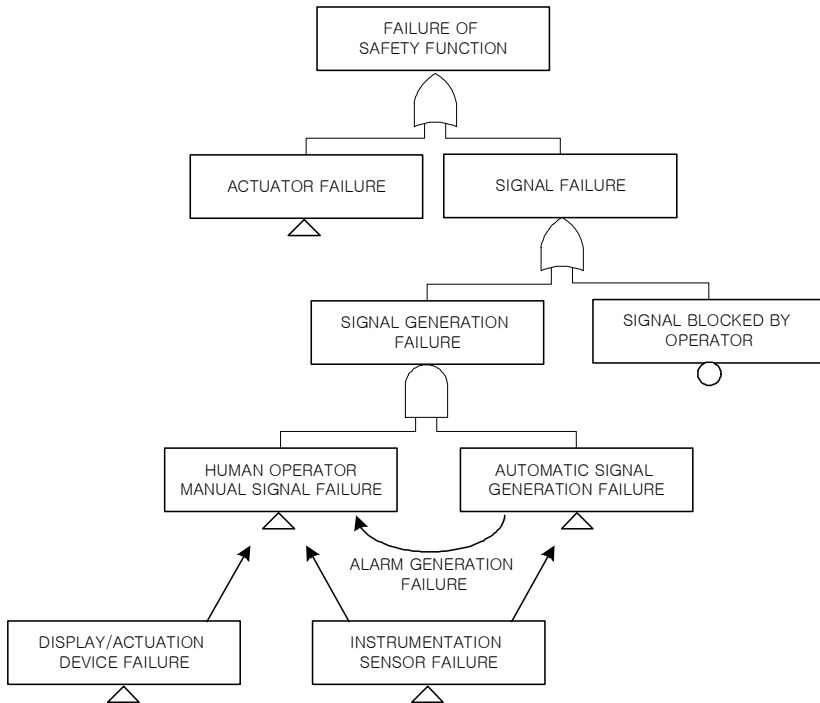


Figure 2.12. The schematic of the concept of the safety function failure mechanism [22]

The failure of a human operator to generate a safety-action signal (EOO) is modeled in a typical PRA. A human operator is usually treated as the backup for an automated digital system. The event of an EOO is followed by the failure of automatic signal generation. The probability of an EOO is evaluated, based on assumptions which reflect the reasons for automatic generation failure. Situations after digital processing system failure are different from that of a conventional analog system failure (trip and pre-trip alarms will be provided to the operator in a more confusing manner in the case of digital system). The probability of an EOO will increase [23–25]. The increased EOO probability results in higher plant risk.

The initiation of a spurious transient by a human operator is treated as an EOC. The loss or the faulty provision of essential information results in an increase in EOCs. EOCs have a greater potential for being significant contributors to plant risk [26].

2.7 Assessment of Common-cause Failure

Safety-critical systems in nuclear power plants adopt multiple-redundancy design in order to reduce the risk from single component failure. The digitalized safety-signal generation system is based on a multiple-redundancy strategy that consists

of redundant components. The level of redundant design of digital systems is usually higher than those of conventional mechanical systems. This higher redundancy will clearly reduce the risk from a single component failure, and raise the importance of CCF analysis. CCF stands for failure of multiple items occurring from a single cause that is common to all.

Environmental causes for digital system failure are smoke, high temperature, manufacturing fault, and design fault. There are several definitions of CCF events. A common-cause event is defined as “A dependent failure in which two or more component fault states exist simultaneously, or within a short time interval, and are a direct result of a shared cause,” according to NUREG/CR-6268.

Two kinds of dependent events have been identified by OEDC/NEA when modeling common-cause failures in systems consisting of redundant components [27, 28]:

- *Unavailability of a specific set of components of the system, due to a common dependency, for example on a support function. If such dependencies are known, they can be explicitly modeled in a PRA.*
- *Unavailability of a specific set of components of the system due to shared causes that are not explicitly represented in the system logic model. Such events are also called “residual” CCFs, and are incorporated in PRA analyses by parametric models.*

Arguments in the analysis of CCF events have been raised concerning the applicability of multiple failure data. Acquiring data for CCF analysis is difficult since the components and modules in a newly designed digital system are different from those in old ones. CCF events tend to involve very plant-specific features. Whether events occurring at a specific system in one plant are directly applicable in the analysis of another system in a different plant is not clear.

A higher level of redundancy increases the difficulty of a CCF analysis, since an impractically large number of CCF events need to be modeled in the fault tree, if conventional CCF modeling methods are applied. For example, 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. If the number of redundancy for safety signal-processing modules is 16, the system model will have 65,519 CCF events (${}_{16}C_2 + {}_{16}C_3 + {}_{16}C_4 + \dots + {}_{16}C_{15} + {}_{16}C_{16} = 2^{16} - 16 - 1 = 65,519$). The number of CCF events in a model will increase to 131,054, 262,125, and 524,268, if a system has redundancies of 17, 18, and 19, respectively. These large numbers of CCF events are not practical for treatment in a PRA.

CCFs are a major cause of system failure for highly redundant systems. The occurrence of a CCF event will also affect operator performance, if the system provides important operational information. CCF events in a digital system model are carefully treated with consideration of:

- Proper CCF data are collected and analyzed to estimate the probability of each CCF event.
- Large number of CCF events is reduced in an acceptable manner for developing a practical PRA model.

- Information for operator performance estimation is available after the reduction in number of events.

2.8 Concluding Remarks

The factors which are carefully considered in modeling the safety of digital systems are listed:

- CCF estimation
- Modeling for dynamic system
- Software testing for failure probability estimation
- Evaluation of software verification and validation
- Dependency between diverse software programs
- Effect of hardware and software interaction
- The fault coverage of fault-tolerant mechanisms
- The safety of network communication
- The probability of human error of omission
- The probability of human error of commission

The proper consideration of these factors makes the safety assessment results more realistic. The active design feedback of insight from the risk assessment will improve the large safety-critical system reliability in an effective manner. Fault monitoring of input/output modules in addition to the processor module is an example of extended design feedback of risk information. Properly designed on-line testing and monitoring mechanisms will improve system integrity by reducing inspection intervals.

References

- [1] Kang HG, Jang SC, Ha JJ (2002) Evaluation of the impact of the digital safety-critical I&C systems, ISOFIC2002, Seoul, Korea, November 2002
- [2] Sancaktar S, Schulz T (2003) Development of the PRA for the AP1000, ICAPP '03, Cordoba, Spain, May 2003
- [3] Hisamochi K, Suzuki H, Oda S (2002) Importance evaluation for digital control systems of ABWR Plant, The 7th Korea-Japan PSA Workshop, Jeju, Korea, May 2002
- [4] HSE (1998) The use of computers in safety-critical applications, London, HSE books
- [5] Kang HG, et al. (2003) Survey of the advanced designs of safety-critical digital systems from the PSA viewpoint, Korea Atomic Energy Research Institute, KAERI/AR-00669/2003
- [6] Goldberg BE, Everhart K, Stevens R, Babbitt N III, Clemens P, Stout L (1994) System engineering "Toolbox" for design-oriented engineers, NASA Reference Publication 1358
- [7] Meshkat L, Dugan JB, Andrews JD (2000) Analysis of safety systems with on-demand and dynamic failure modes, Proceedings of 2000 RM
- [8] White RM, Boettcher DB (1994) Putting Sizewell B digital protection in context, Nuclear Engineering International, pp. 41–43

- [9] Parnas DL, Asmis GJK, Madey J (1991) Assessment of safety-critical software in nuclear power plants, *Nuclear Safety*, Vol. 32, No. 2
- [10] Butler RW, Finelli GB (1993) The infeasibility of quantifying the reliability of life-critical real-time software, *IEEE Transactions on Software Engineering*, Vol. 19, No. 1
- [11] Kang HG, Sung T, et al (2000) Determination of the Number of Software Tests Using Probabilistic Safety Assessment KNS conference, *Proceeding of Korean Nuclear Society*, Taejon, Korea
- [12] Littlewood B, Wright D (1997) Some conservative stopping rules for the operational testing of safety-critical software, *IEEE Trans. Software Engineering*, Vol. 23, No. 11, pp. 673–685
- [13] Saiedian H (1996) An Invitation to formal methods, *Computer*
- [14] Rushby J (1993) Formal methods and the certification of critical systems, SRI-CSL-93-07, Computer Science Laboratory, SRI International, Menlo Park
- [15] Welbourne D (1997) Safety critical software in nuclear power, *The GEC Journal of Technology*, Vol. 14, No. 1
- [16] Dahll G (1998) The use of Bayesian belief nets in safety assessment of software based system, HWP-527, Halden Project
- [17] Eom HS, et al. (2001) 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
- [18] Littlewood B, Popov P, Strigini L (1999) A note on estimation of functionally diverse system, *Reliability Engineering and System Safety*, Vol. 66, No. 1, pp. 93-95
- [19] Bastl W, Bock HW (1998) German qualification and assessment of digital I&C systems important to safety, *Reliability Engineering and System Safety*, Vol. 59, pp. 163-170
- [20] Choi JG, Seong PH (2001) Dependability estimation of a digital system with consideration of software masking effects on hardware faults, *Reliability Engineering and System Safety*, Vol. 71, pp. 45-55
- [21] Bayrak T, Grabowski MR (2002) Safety-critical wide area network performance evaluation, *ECIS 2002*, June 6–8, Gdańsk, Poland
- [22] Kang HG, Jang SC (2006) 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
- [23] Kauffmann JV, Lanik GT, Spence RA, Trager EA (1992) Operating experience feedback report – human performance in operating events, *USNRC, NUREG-1257*, Vol. 8, Washington DC
- [24] Decortis F (1993) Operator strategies in a dynamic environment in relation to an operator model, *Ergonomics*, Vol. 36, No. 11
- [25] Park J, Jung W (2003) The requisite characteristics for diagnosis procedures based on the empirical findings of the operators' behavior under emergency situations, *Reliability Engineering & System Safety*, Volume 81, Issue 2
- [26] Julius JA, Jorgenson EJ, Parry GW, Mosleh AM (1996) Procedure for the analysis of errors of commission during non-power mode of nuclear power plant operation, *Reliability Engineering & System Safety*, Vol. 53
- [27] OECD/NEA Committee on the safety of nuclear installations, 1999, *ICDE project report on collection and analysis of common-cause failures of centrifugal pumps*, NEA/CSNI/R(99)2
- [28] OECD/NEA Committee on the safety of nuclear installations, 2003, *ICDE project report: Collection and analysis of common-cause failures of check valves*, NEA/CSNI/R(2003)15

Reliability and Risk Issues in Large Scale Safety-critical
Digital Control Systems

Seong, P.-H. (Ed.)

2009, XXIV, 304 p., Hardcover

ISBN: 978-1-84800-383-5