Reliability graph with general gates (RGGG): A novel method for reliability analysis

SEONG Poong-Hyun[1, 2] and SHIN Seung-Ki[1]

1. Department of Nuclear and Quantum Engineering, Korea Advanced Institute of Science and Technology, 373-1 Guseong-dong, Yuseong-gu, Daejeon, Republic of Korea, 305-701 (phseong@kaist.ac.kr, kankani@kaist.ac.kr)

2. Department of Nuclear Engineering, Khalifa University of Science, Technology & Research, P.O. Box 127788, Abu Dhabi, UAE

Abstract: There are several methods for system reliability analysis such as reliability graphs, fault tree analyses, Markov chains, and Monte Carlo simulations. Among the existing methods, the reliability graphs are the most intuitive modeling method, but they are not widely used due to their limited expression power. In this paper, an intuitive and practical method for system reliability analysis named the reliability graph with general gates (RGGG) is reviewed. The proposed method introduces general gates to the conventional reliability graph method, which creates a one-to-one match from the actual structure of the system to the reliability graph of the system. A quantitative evaluation method is proposed by transforming the RGGG to an equivalent Bayesian network without losing the intuitiveness of the model. In addition, a method of analyzing the dynamic systems and repairable systems which uses the RGGG is introduced, and appropriate algorithms for the quantitative analyses are explained. It is concluded that the RGGG method is intuitive and easy-to-use in the analyses of static, dynamic, and repairable systems compared with other methods while its analysis results are the same as those of other methods.

Keyword: reliability graph; fault tree; Bayesian network; dynamic systems; repairable systems

1 Introduction

System reliability refers to the probability that an item will perform a required function when used for its intended purpose, under the stated conditions, for a given period of time[1]. Several methods are viable for system reliability analyses such as reliability graphs, fault tree analyses, Markov chains, and Monte Carlo simulations. Each method has its own peculiar features and those features should be considered when determining the most suitable method. Among the existing methods, the fault tree analysis is the most widely used due to its expression power, applicability to complex systems, and various tool supports. However, because analysts must draw a fault tree based on the logical relationships among the components in a system, the use of fault tree analyses become more and more cumbersome as systems become more complex. In order to reduce the amount of analysis errors, an intuitive method for system reliability analysis should be developed. Among the existing reliability analysis methods, the reliability graph is the most intuitive method for modeling target systems, but it has one serious drawback. Since it has a limited expression power, it cannot be used widely for system reliability analyses. Kim and Seong proposed a reliability graph with general gates (RGGG), which is an intuitive and practical reliability analysis method, by extending the conventional reliability graph[2]. The proposed method introduces general gates to a conventional reliability graph. Therefore, it possesses the intuitiveness that is characteristic of a conventional reliability graphs and additional powers of expression. At first, the RGGG method was only developed for the reliability analysis of non-repairable static systems. Therefore, it cannot be applied if the failure of a system is related to a sequence of component failures. This system is defined as a dynamic system and the reliability of the system can be estimated using dynamic reliability analysis methods such as the dynamic fault tree and Markov chain. Recently, the capability of the RGGG was upgraded so that it can analyze dynamic systems[3, 4]. In addition, the RGGG was applied to repairable systems to analyze availability based on the Markov process[5].

Received date: August 29, 2010
(Revised date: September 13, 2010)

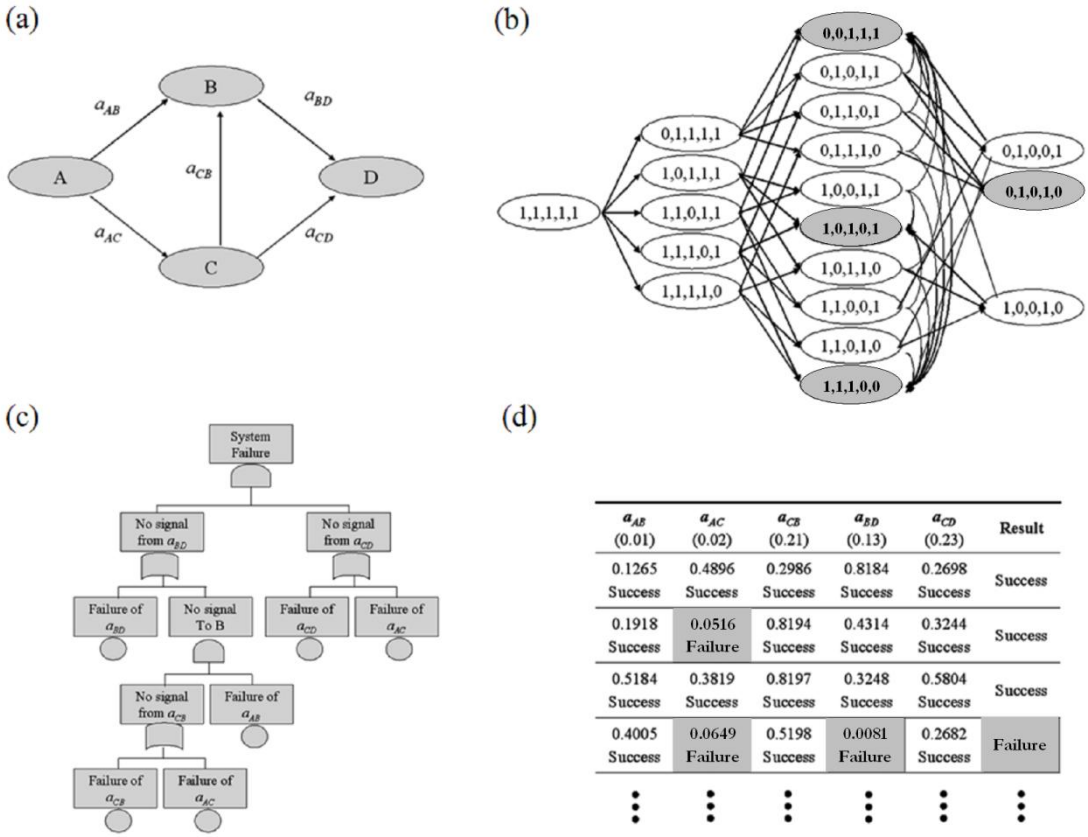

Fig.1 Four existing reliability analysis methods.
(a) Reliability graph; (b) Markov chain; (c) Fault tree analysis; and (d) Monte Carlo simulation.

This paper presents a general review of the RGGG method and the advantages of the RGGG method in reliability analyses. The remainder of the paper is structured as follows: Section 2 gives full details of the RGGG method and Section 3 introduces the extension of the RGGG to dynamic systems and repairable systems. In Section 4, the RGGG method is summarized and discussed.

2 Reliability graph with general gates

2.1 Reliability graph

For a system reliability analysis method to be intuitive, a one-to-one match between the actual structure of the system and the system model should be guaranteed. A reliability graph is composed of nodes and arcs. A node represents a component in the system, while an arc is used to model the link between two components. Therefore, the reliability graph can make a one-to-one match between the actual structure of the system and the system model. An application of the four existing reliability analysis methods is shown in Fig. 1. The example system is a data delivery system from node *A* to node *D*, under random failures of five transmission lines, a_{AB}, a_{AC}, a_{CB}, a_{BC}, and a_{CD}. This example system is from the node-pair (2-terminal) reliability evaluation. The node-pair reliability is the probability that at least one path exists between a source node and a target node in a directed network[6]. In this example, the system is successful if at least one path from node *A* to node *D* exists. In Fig. 1(b), the numbers in each state indicate the success and failure of five transmission lines in order of a_{AB}, a_{AC}, a_{CB}, a_{BC}, and a_{CD}; *0* indicates a failure of the transmission line. The four states colored black are the sink states that correspond to the four minimal cut-sets of the system. In Fig. 1(d), the number in each parenthesis is the failure probability of the transmission line, and the numbers in each string are the generated random numbers for one realization. Among the four methods, it can be seen that the actual structure of the system is most easily understood through the reliability graph. In this sense, it is generally believed that reliability graph is the most intuitive method for understanding and analyzing the reliability of a system.

However, one serious shortcoming of the reliability graph is its low expression power; it can only express the characteristics of an OR gate. Suppose that node *D* in Fig. 1(a) requires inputs from nodes *B* and *C*. Then, it is not possible to describe the system intuitively using the reliability graph because the reliability graph originates from the subject of the node-pair network reliability. Therefore, in order to intuitively analyze general systems, the expression power of the reliability graph should be improved.

2.2 Reliability graph with general gates (RGGG)

In order to overcome the limitation of the reliability graph mentioned above, Kim and Seong proposed that additional general gates be added to the reliability graph[2]. Based on the fault tree analysis, OR, AND, and K-out-of-N gates are the most frequently used gates for system reliability analysis. Therefore, special graphical notations for the three gates are assigned as shown in Figs. 2(a) to 2(c). In addition, the general purpose node shown in Fig. 2(d) is proposed to maintain the intuitiveness of the reliability graph. The general purpose node is defined by the proper probability table that describes the characteristics of the node.

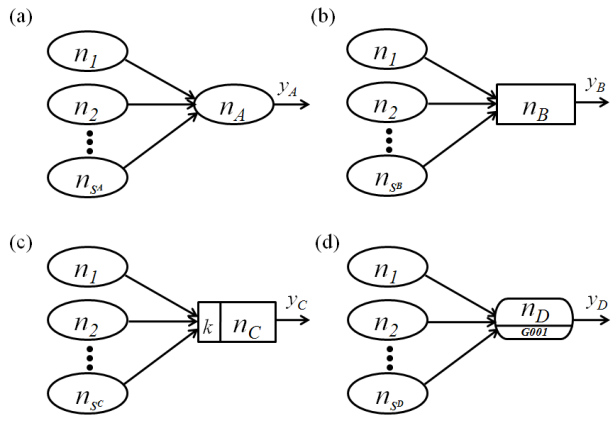

Fig.2 Definition of gates for reliability graph with general gates. (a) OR gate; (b) AND gate; (c) K-out-of-N gate; and (d) general purpose gate.

2.3 Quantification of the RGGG

For the modeling to be as realistic as possible, it should be assumed that both nodes and arcs can fail in a reliability graph. A reliability graph with both node and arc failures can be transformed into an equivalent reliability graph with only arc failures, as shown in Fig. 3[7]. Therefore, in the evaluation methods for the RGGG, only arc failures are developed.

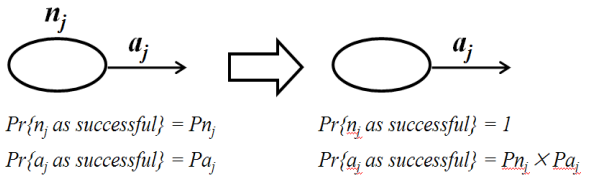

Fig. 3 Transformation to a reliability graph with a perfect node: (a) the original network (with node and arc failures) and (b) the transformed network (with arc failures only).

2.3.1 Transforming to Bayesian networks

In case of directed acyclic graphs, the reliability graph can be transformed to an equivalent Bayesian network without losing the one-to-one match with the actual system structure. A Bayesian network (sometimes called belief network, causal probabilistic network, causal net, probabilistic cause-effect model, or probabilistic influence diagram) is a graphical network that represents the probabilistic relationships among variables[8, 9]. Bayesian networks have attracted much attention as a possible solution for the problems of decision support under uncertainty and are considered to be the most promising method for the estimation of software reliability[9-11]. In order to transform the RGGG to an equivalent Bayesian network, the probability table for each node in the RGGG must be determined. The following sections describe how to determine the probability table for each node in the equivalent Bayesian network.

2.3.2 Modeling of RGGG

G, a reliability graph with general gates as shown in Fig. 4, is a tuple $G = (N, A, F)$ where

A. n_i: *i*th node of G ($i = 0,1,\ldots,t$), where n_0 is the source node and n_t is the target node
B. $N = \{n_i \mid i = 0,1,\ldots,t\}$
C. a_{ij}: the directed arc from n_i to nj ($i = 0,1,\ldots,t-1$ and $j = 1,2,\ldots,t$ and $i \neq j$)
D. $A = \{a_{ij} \mid$ there is a directed arc from n_i to n_j, $i = 0,1,\ldots,t-1$ and $j = 1,2,\ldots,t$ and $i \neq j\}$
E. f_i: the node function for n_i ($i = 1,2,\ldots,t$). There is no node function for n_0.
F. $F = \{f_i \mid i = 1,2,\ldots,t\}$

G. $F(x_1,\ldots,x_n)$ is *symmetric* if $f(x_1,\ldots,x_i,\ldots,x_j,\ldots,x_n) = f(x_1,\ldots,x_i,\ldots,x_j,\ldots,x_n)$, where $i = 1,\ldots,n$ and $j = 1,\ldots,n$ and $i \neq j$.

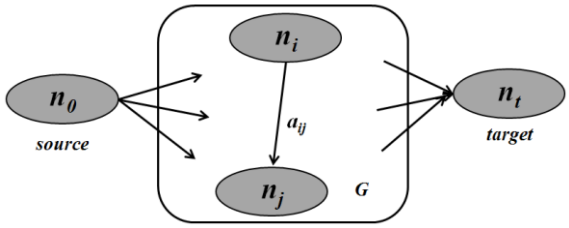

Fig. 4 Modeling reliability graph with general gates.

For a node n_j in G, the following are defined.

H. A node n_i is a *parent node* of n_j if there is a directed arc a_{ij}.
I. Uj: a set of parent nodes for node n_j
J. s_j: the number of parent nodes for n_j
K. y_j: the Boolean variable for the output of n_j ($y_j = 1$ if n_j is in the success state, $y_j = 0$ if n_j is in the failure state)
L. w_{ij}: the Boolean variable for a_{ij} ($w_{ij} = 1$ if a_{ij} is in the success state, $w_{ij} = 0$ if a_{ij} is in the failure state)
M. x_{ij}: the Boolean variable for the input into n_j originated from $n_i \in U_j$. $x_{ij} = y_i w_{ij}$
N. r_{ij}: the reliability of a_{ij}, i.e. $r_{ij} = \Pr\{a_{ij}$ is in the success state$\} = \Pr\{w_{ij} = 1\}$

To determine the probability table for n_j, the success and failure probabilities for given states of the parent nodes should be evaluated. For symmetric node functions such as the node functions for the OR, AND, and K-out-of-N gates, the success and failure probabilities are functions of only the number of successful parent nodes, i.e. the parent nodes in the success state. Thus, for the given states of a parent node, the followings are defined.

O. v: the serial number for given states of parent nodes $(v = 1,2,\ldots,2^{S_j})$
P. $s_j^{(v)}$: the number of successful parent nodes for the vth set of parent node states
Q. $U_j^{(v)}$: the set of successful parent nodes for node n_j for the vth set of parent node states
R. $n_{lj}^{(v)}$: the lth successful parent node for node n_j for the vth set of parent node states ($l = 1,\ldots,s_j^{(v)}$); i.e. $U_j^{(v)} = \{n_{1j}^{(v)},\ldots,n_{(v)}^{(v)}\}$
S. $a_{lj}^{(v)}$: the arc from $n_{lj}^{(v)}$ to n_j
T. $y_{lj}^{(v)}$: the Boolean variable for the output of $j_{lj}^{(v)}$
U. $w_{lj}^{(v)}$: the Boolean variable for $a_{lj}^{(v)}$
V. $r_{lj}^{(v)}$: the reliability of $a_{lj}^{(v)}$
W. $P_j^{(v)}$: $\Pr\{y_j = 1$ for vth set of parent node states for node $n_j\}$
X. $Q_j^{(v)}$: $\Pr\{y_j = 0$ for vth set of parent node states for node $n_j\}$. $Q_j^{(v)} = 1 - P_j^{(v)}$

From now on, the $P_j^{(v)}$'s for the node functions of the OR, AND, and K-out-of-N gates are evaluated.

2.3.3 OR node

The node function for a node with an OR gate (Fig. 2(a)) is given as:[12]

$$y_A = f_A(x_{1A},\ldots,x_{s_A A}) = x_{1A} \vee \ldots \vee x_{s_A A}. \quad (1)$$

Because $x_{iA} = y_i w_{iA}$ ($i = 1,2,\ldots,s_A$), Eq. (1) can be rewritten as:

$$y_A = f_A(y_1 w_{1A},\ldots,y_{s_A} w_{s_A A}) = y_1 w_{1A} \vee \ldots \vee y_{s_A} w_{s_A A}. \quad (2)$$

For the parent nodes that are in the failure state, y_i's are 0 and the corresponding terms in Eq. (2) are removed. Thus, when $s_A^{(v)} \geq 1$, Eq. (2) can be rewritten as:

$$y_A = f_A(y_1 w_{1A},\ldots,y_{s_A} w_{s_A A}) = y_{1A}^{(v)} w_{1A}^{(v)} \vee \ldots \vee y_{s_A^{(v)} A}^{(v)} w_{s_A^{(v)} A}^{(v)}. \quad (3)$$

By definition, $y_{lA}^{(v)} = 1$ ($l = 1,2,\ldots,s_A^{(v)}$). Thus,

$$y_A = f_A(y_1 w_{1A},\ldots,y_{s_A} w_{s_A A}) = w_{1A}^{(v)} \vee \ldots \vee w_{s_A^{(v)} A}^{(v)}. \quad (4)$$

The success probability is given as:

$$P_A^{(v)} = \Pr\{y_A = 1\} = \Pr\{w_{1A}^{(v)} \vee \ldots \vee w_{s_A^{(v)} A}^{(v)} = 1\} = 1 - \Pr\{\overline{w_{1A}^{(v)}} \vee \ldots \vee \overline{w_{s_A^{(v)} A}^{(v)}} = 1\}. \quad (5)$$

If $w_{lA}^{(v)}$'s ($l = 1,2,\ldots,s_A^{(v)}$) are s-independent of each other:

$$P_A^{(v)} = 1 - (1 - r_{1A}^{(v)}) \cdots (1 - r_{s_A^{(v)} A}^{(v)}) = 1 - \prod_{l=1}^{s_A^{(v)}} \left(1 - r_{lA}^{(v)}\right). \quad (6)$$

When $s_A^{(v)} = 0$, $P_A^{(v)} = 0$. Therefore, $P_A^{(v)}$ is given as:

$$P_A^{(v)} = \begin{cases} 1 - \prod_{l=1}^{s_A^{(v)}} (1 - r_{lA}^{(v)}) & \text{when } 1 \le s_A^{(v)} \le s_A, \\ 0 & \text{when } s_A^{(v)} = 0. \end{cases} \quad (7)$$

As mentioned before, $Q_A^{(v)}$ is given as:

$$Q_A^{(v)} = 1 - P_A^{(v)}. \quad (8)$$

Equations (7) and (8) can be used to determine the success and failure probabilities for given states of the parent nodes in the probability table for an OR node. For example, the probability table for an OR node when $s_A = 2$ is given in Table 1.

Table 1 Probability table for an OR node with two inputs

	$y_1 = 1$ (success)		$y_1 = 0$ (failure)	
	$y_2 = 1$ (success)	$y_2 = 0$ (failure)	$y_2 = 1$ (success)	$y_2 = 0$ (failure)
$y_A = 1$ (success)	$r_{1A} + r_{2A} - r_{1A}r_{2A}$	r_{1A}	r_{2A}	0
$y_A = 0$ (success)	$1 - (r_{1A} + r_{2A} - r_{1A}r_{2A})$	$1 - r_{1A}$	$1 - r_{2A}$	1

2.3.4 AND node

The node function for a node with an AND gate (Fig. 2(b)) is given as:

$$y_B = f_B(x_{1B}, \ldots, x_{s_B B}) = x_{1B} \wedge \ldots \wedge x_{s_B B}$$
$$= y_1 w_{1B} \wedge \ldots \wedge y_{s_B} w_{s_B B}. \quad (9)$$

A node with an AND gate can be in the success state only when all parent nodes are in the success state. Therefore, for $P_B^{(v)}$ to have a non-zero value, all y_l's ($l = 1,2,\ldots,s_B$) should be 1, which means that $s_B^{(v)} = s_B$. Thus, when $s_B^{(v)} = s_B$:

$$y_B^{(v)} = Pr\{y_B = 1\} = Pr\{w_{1B} \wedge \ldots \wedge w_{s_B B} = 1\}. \quad (10)$$

If all w_{lb}'s ($l = 1,2,\ldots,s_B$) are *s*-independent of each other,

$$P_B^{(v)} = r_{1B} \cdots r_{s_B B} = \prod_{l=1}^{s_B} r_{lB}. \quad (11)$$

If there is at least one $y_l = 0$ ($l = 1,2,\ldots,s_B$), $P_B^{(v)} = 0$. Thus,

$$P_B^{(v)} = \begin{cases} \prod_{l=1}^{s_B} r_{lB} & when\ s_B^{(v)} = s_B, \\ 0 & otherwise. \end{cases} \quad (12)$$

$$Q_B^{(v)} = 1 - P_B^{(v)}. \quad (13)$$

The probability table for an AND node can be determined using Eqs.(12) and (13). An example probability table for an AND node in case of $n = 2$ is given in Table 2.

Table 2 Probability table for an AND node when $n = 2$

	$y_1 = 1$ (success)		$y_1 = 0$ (failure)	
	$y_2 = 1$ (success)	$y_2 = 0$ (failure)	$y_2 = 1$ (success)	$y_2 = 0$ (failure)
$y_B = 1$ (success)	$r_{1B}r_{2B}$	0	0	0
$y_B = 0$ (success)	$1 - r_{1B}r_{2B}$	1	1	1

2.3.5 K-out-of-N node

The K-out-of-N gate means that the output of a node becomes successful when there are at least *k* successful inputs to the node. The node function for a node with a K-out-of-N gate (Fig. 2(c)) is given as:

$$y_C = f_C(x_{1C}, \ldots, x_{nC})$$
$$= (x_{1C} \wedge \ldots \wedge x_{kC}) \vee \ldots \vee (x_{n-k+1,C} \wedge \ldots \wedge x_{kC}). \quad (14)$$

For a node with a K-out-of-N gate to be successful, there should be at least *k* successful parent nodes. As mentioned before, $P_C^{(v)}$ is a function of only the number of successful parent nodes, because the node function for a node with a K-out-of-N gate is symmetric. When $s_C^{(v)} \ge k$, the success probability is given as:

Table 3 Probability table for a K-out-of-N node (*k = 2, n = 3*)

y_1	Success (S)				Failure (F)			
y_2	S		F		S		F	
y_3	S	F	S	F	S	F	S	F
$y_C = 1$ (success)	$r_{1C}r_{2C}r_{3C} + (1 - r_{1C})r_{2C}r_{3C} + (1 - r_{2C})r_{1C}r_{3C} + (1 - r_{3C})r_{1C}r_{2C}$	$r_{1C}r_{2C}$	$r_{1C}r_{3C}$	0	$r_{2C}r_{3C}$	0	0	0
$y_C = 0$ (success)	$(1 - r_{1C})(1 - r_{2C})(1 - r_{3C}) + r_{1C}(1 - r_{2C})(1 - r_{3C}) + r_{2C}(1 - r_{1C})(1 - r_{3C}) + r_{3C}(1 - r_{1C})(1 - r_{2C})$	$1 - r_{1C}r_{2C}$	$1 - r_{1C}r_{3C}$	1	$1 - r_{2C}r_{3C}$	1	1	1

$P_C^{(v)}$ = Pr{there are more than k successful inputs among $s_C^{(v)}$}

= Pr{k successful inputs} + ⋯

+ Pr{ $s_C^{(v)}$ successful inputs}. (15)

Because the Boolean variables for $s_C^{(v)}$ successful parent nodes are all *1*,

$P_C^{(v)}$ = Pr{k successful arcs} + ⋯

+ Pr{ $s_C^{(v)}$ successful arcs}

$= \sum_{m=k}^{s_C^{(v)}} P_{C,m}^{(v)}$. (16)

Where $P_{C,m}^{(v)}$ is defined as:

$P_{C,m}^{(v)}$ = Pr{there are exactly m successful arcs among $s_C^{(v)}$} ($m = k,k+1,\ldots, s_C^{(v)}$).

If $w_{lC}^{(v)}$'s ($l = 1,2,\ldots, s_C^{(v)}$) are *s*-independent of each other:

$$P_{C,m}^{(v)} = r_{1C}^{(v)} \ldots r_{mC}^{(v)} \left(1 - r_{m+1,C}^{(v)}\right) \ldots \left(1 - r_{s_C^{(v)},C}^{(v)}\right) + \cdots + \left(1 - r_{1C}^{(v)}\right) \ldots \left(1 - r_{s_C^{(v)}-m,C}^{(s_C)}\right) r_{s_C^{(v)}-m+1,C}^{(v)} \ldots r_{s_C^{(v)}C}^{(v)}. \quad (17)$$

Equation (17) consists of $\binom{s_C^{(v)}}{m}$ terms. When the reliabilities of the arcs are identical and the same value is defined as r ($r_{1c}^{(v)} = \ldots = r_{s_c^{(v)}C}^{(v)} \equiv r$), Eq. (17) can be simplified as:

$$P_{C,m}^{(v)} = \binom{s_C^{(v)}}{m} r^m (1 - r)^{s_C^{(v)} - m}. \quad (18)$$

When $s_C^{(v)} < k$, $P_C^{(v)} = 0$. Thus,

$$P_B^{(v)} = \begin{cases} \sum_{m=k}^{s_C^{(v)}} P_{C,m}^{(v)} & \text{when } s_C^{(v)} \geq k, \\ 0 & \text{when } s_C^{(v)} < k. \end{cases} \quad (19)$$

$$Q_C^{(v)} = 1 - P_C^{(v)}. \quad (20)$$

Using Eqs. (17) (or sometimes (18)), (19), and (20), the probability table for a K-out-of-N node can be determined. For example, the probability table for a K-out-of-N node when $n = 3$ and $k = 2$ is given in Table 3.

The probability tables for nodes with other gates such as an XOR gate can be determined using similar methods.

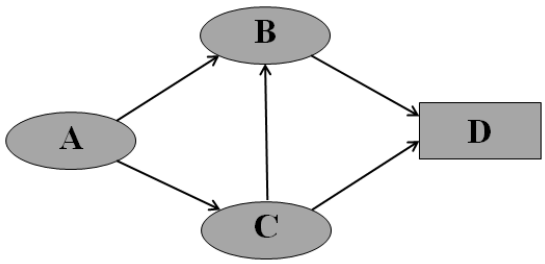

Fig. 5 RGGG for the example system.

2.4 Examples

The RGGG method is applied to a simple example system, shown in Fig. 5, to demonstrate the usefulness of the proposed method. An OR gate is applied to node *B* and an AND gate is applied to node *D*. The reliabilities of a_{AB}, a_{AC}, a_{CB}, a_{BD}, and a_{CD} are assumed to be 0.99, 0.98, 0.79, 0.87, and 0.77, respectively. The probability table for each node is determined based on the equations provided in Section 2.3 and the system reliability can be easily

obtained with various commercial or free software tools for Bayesian networks such as Hugin™, Netica™, and Microsoft Belief Networks (MSBNx). Figure 6 shows the evaluation result using Hugin™ and the reliability of the example system is evaluated to be 0.6551. In order to verify the evaluation result, the example system is analyzed using the fault tree shown in Fig. 7. The minimal cut-sets are found to be $\{a_{AC}\}$, $\{a_{BD}\}$, $\{a_{CD}\}$, and $\{a_{AB}, a_{CB}\}$, and the reliability of the example system is given as:

$$\begin{aligned} R_{sys} &= 1 - \Pr\{\overline{w_{AC}} \vee \overline{w_{BD}} \vee \overline{w_{CD}} \vee \overline{w_{AB}}\,\overline{w_{CB}}\} \\ &= 1 - \Pr\{\overline{w_{AC}} \vee w_{AC}\overline{w_{BD}} \vee w_{AC}w_{BD}\overline{w_{CD}} \\ &\vee w_{AC}w_{BD}w_{CD}\,\overline{w_{AB}}\,\overline{w_{CB}}\}. \end{aligned} \quad (21)$$

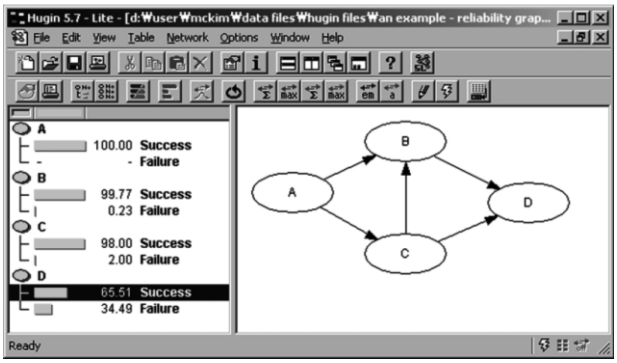

Fig. 6 Reliability analysis of the example system using Hugin™.

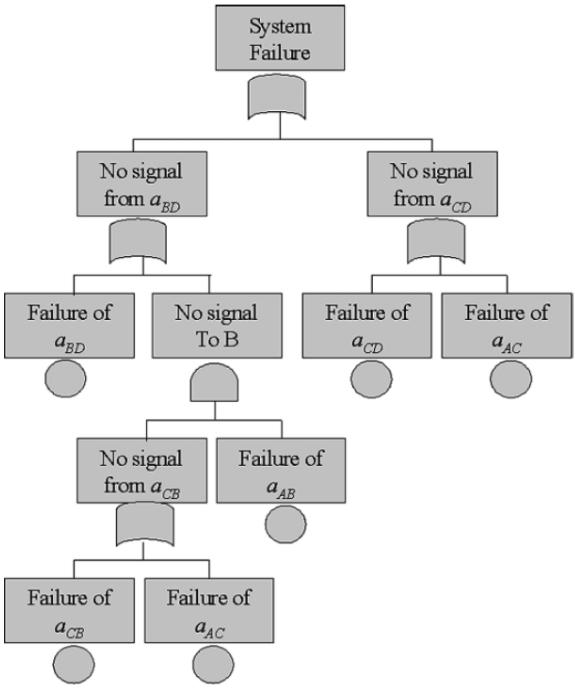

Fig. 7 Fault tree for the example system.

Because the terms in the second line of Eq. (21) are mutually exclusive, Eq. (21) can be calculated as

$$\begin{aligned} R_{sys} &= 1 - [(1 - r_{AB}) + r_{AB}(1 - r_{BD}) + r_{AB}\, r_{BD}(1 - r_{CD}) \\ &\quad + r_{AB}\, r_{BD}\, r_{CD}(1 - r_{AB})(1 - r_{CB})] \\ &= 0.655123. \end{aligned} \quad (22)$$

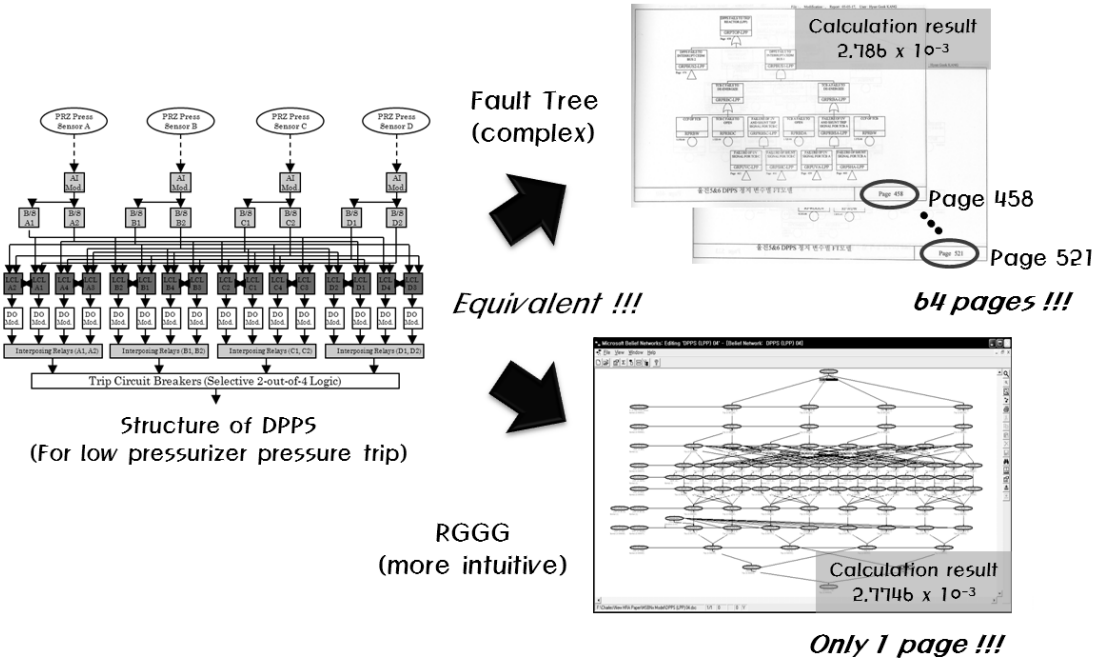

Fig. 8 Modeling of the digital plant protection system with the RGGG and with a fault tree.

From Fig. 5 and Fig. 7, it can be seen that the RGGG method provides a much easier method of modeling and understanding the structure of the system compared with the fault tree method, while the results from both methods are equivalent. For reliability analysis of complex systems, RGGG provides more accurate results compared with the fault tree due to the results of RGGG not possessing truncation errors, which are generally included in large fault tree analyses.

Figure 8 confirms the intuitive power of the RGGG method when applied to a real complex system. Kim and Seong[2] modeled a digital plant protection system for a nuclear power plant using an RGGG and a fault tree. A trip case caused by low pressurizer pressure was analyzed. The failure scenario was modeled through only one page with the RGGG, but through 64 pages with the fault tree: the same quantitative result was calculated in each case.

3 Extension of the RGGG

As with the conventional fault tree method, the RGGG method was developed for the reliability analysis of non-repairable static systems. In order to assess the reliability of dynamic systems and the availability of repairable systems, the dynamic fault tree[13] and repairable fault tree[14, 15] methods were proposed and various research on the evaluation techniques has been conducted. Similar approaches have been performed in order to enhance the advantages of the RGGG method. In this section, two extensions proposed to the original formalism of the RGGG are introduced: the dynamic RGGG[3, 4] and the repairable RGGG[5].

3.1 Dynamic RGGG

A dynamic fault tree technique was developed to handle the difficulties that arise in the reliability analysis of fault-tolerant computer systems when critical applications are complicated by several factors[13]. Four dynamic gates were adopted in a conventional fault tree method: a functional dependency (FDEP) gate, a spare gate (a cold spare (CSP) gate, a hot spare (HSP) gate, and a warm spare (WSP) gate), a priority AND gate (PAND), and a sequence-enforcing (SEQ) gate. Each dynamic gate can express the dynamic failure process that is related to the failure sequence of the component parts. The dynamic fault tree is an effective technique for modeling dynamic systems. However, an intuitive modeling method needs to be developed for easy modeling of real dynamic systems and to ensure that real systems can be understood easily from a diagram of the model. To enable the RGGG to model dynamic systems, additional dynamic nodes are introduced based on the four dynamic gates of the dynamic fault tree as shown in Fig. 9.

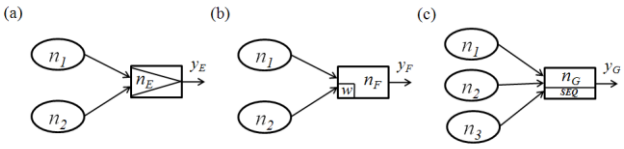

Fig. 9 The dynamic nodes of the dynamic RGGG: (a) a PAND node; (b) a WSP node; and (c) an SEQ node.

3.1.1 Addition of dynamic nodes

(a) PAND node

Figure 9(a) shows a PAND node. Node E (n_E) fails only if both signals from node 1 (n_1) and node 2 (n_2) are disconnected and the signal from n_1 is disconnected before that from n_2.

(b) Spare node

Figure 9(b) shows a spare node. The signal from n_1 is the primary input signal and the signal from n_2 is the spare input signal. The letter *w* in indicates that n_F is a WSP node. The failure rate of spare input is reduced by a *dormancy factor* $\alpha \in [0, 1]$. The spare node fails only if the primary signal and all spare signals are disconnected.

(c) SEQ node

Figure 9(c) shows an SEQ node. The input signals are constrained to be disconnected in a particular order; the SEQ node fails if, and only if, all input signals are disconnected. The constrained disconnection order is from top to bottom.

A novel FDEP node is not required in the RGGG as the existing RGGG can demonstrate the property of an FDEP gate using the OR nodes only.

3.1.2 Quantification of dynamic nodes

As the novel dynamic nodes are proposed, methods for creating the probability tables of those nodes are

introduced for the quantitative assessment. A discrete-time method [16,17] is employed to determine the probability tables.

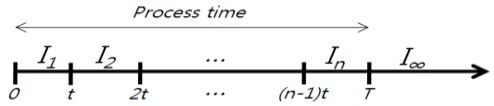

Fig. 10 Discretization of the process time.

As shown in Fig. 10, the line of the total process time (T) is divided into n equal intervals. The time of one interval is defined as t. The output of each node is one of $\{I_1, I_2, ..., I_n, I_\infty\}$. If the output of a certain node is I_k, the node fails in the kth interval; and I_∞ denotes that the node never fails during the total process time. If P_{ij}^{k} denotes the probability that an arc (a_{ij}) from node i (n_i) to node j (n_j) fails in the kth interval, and if $F_{ij}(t)$ denotes the cumulative failure distribution function of a_{ij}, P_{ij}^{k} can be derived as follows:

$$P_{ij}^{k} = \int_{(k-1)t}^{kt} \frac{dF_{ij}(t)}{dt} dt \qquad (23)$$

If the total process time and the discretization number (n) are decided, the probabilities (P_{ij}^{k}) of all the arcs for all values of k ($k = 1, 2, ..., n, \infty$) can be derived using Eq. (23) before creating the probability tables.

In order to estimate accurate reliability, the discretization number should be increased. However, as n increases, the probability table of each node becomes more complex. The number of blanks that should be filled for a probability table of a node with two inputs is $(n+1)^3$. Therefore, a set of rules for creating the probability table of each dynamic node is developed. First, let the outputs of n_1, n_2, n_3, n_E, n_F, and n_G be I_x, I_y, I_z, I_e, I_f, and I_g, respectively ($x, y, z, e, f, g \in 1, 2, ..., n, \infty$).

(a) PAND node

The rules for creating the probability table of the PAND node shown in Fig. 9(a) are explained. In the table, each blank that is defined by x, y, and e can be filled on the basis of the following rules:

A. If $e > y$,
0.

B. If $e = y \le x$,
$Pr\{a_{1E}$ fails before the eth interval$\} \cdot (1 - Pr\{a_{2E}$ fails before the eth interval$\})$.

C. If $e \le x$, $e < y$,
$Pr\{a_{1E}$ fails before the eth interval$\} \cdot Pr\{a_{2E}$ fails at the eth interval$\}$.

D. If $e = y > x$,
$1 - Pr\{a_{2E}$ fails before the eth interval$\}$.

E. If $x < e < y$,
$Pr\{a_{2E}$ fails at the eth interval$\}$.

F. If $e = \infty$,
1 – (sum of the other probabilities in the same row).

Table 4 Probability table for a PAND node for *n=3*

		n_E			
n_1	n_2	I_1	I_2	I_3	I_∞
I_1	I_1	0	0	0	1
	I_2	0	$1 - P_{2E}^{1}$	0	1-Σ
	I_3	0	P_{2E}^{2}	$1 - P_{2E}^{1} - P_{2E}^{2}$	1-Σ
	I_∞	0	P_{2E}^{2}	P_{2E}^{3}	1-Σ
I_2	I_1	0	0	0	1
	I_2	0	$P_{1E}^{1}(1 - P_{2E}^{1})$	0	1-Σ
	I_3	0	$P_{1E}^{1} P_{2E}^{2}$	$1 - P_{2E}^{1} - P_{2E}^{2}$	1-Σ
	I_∞	0	$P_{1E}^{1} P_{2E}^{2}$	P_{2E}^{3}	1-Σ
I_3	I_1	0	0	0	1
	I_2	0	$P_{1E}^{1}(1 - P_{2E}^{1})$	0	1-Σ
	I_3	0	$P_{1E}^{1} P_{2E}^{2}$	$(P_{1E}^{1} + P_{1E}^{2})(1 - P_{2E}^{1} - P_{2E}^{2})$	1-Σ
	I_∞	0	$P_{1E}^{1} P_{2E}^{2}$	$(P_{1E}^{1} + P_{1E}^{2})P_{2E}^{3}$	1-Σ
I_∞	I_1	0	0	0	1
	I_2	0	$P_{1E}^{1}(1 - P_{2E}^{1})$	0	1-Σ
	I_3	0	$P_{1E}^{1} P_{2E}^{2}$	$(P_{1E}^{1} + P_{1E}^{2})(1 - P_{2E}^{1} - P_{2E}^{2})$	1-Σ
	I_∞	0	$P_{1E}^{1} P_{2E}^{2}$	$(P_{1E}^{1} + P_{1E}^{2})P_{2E}^{3}$	1-Σ

When the probability table is filled by using rules i to v, the case where the value of e is ∞ is excluded.

Furthermore, because the sum of the probabilities in each row should be *1*, rule vi is applied. Table 4 shows the probability table for a PAND node; the table is based on the rules of the PAND node for $n = 3$. For an arbitrary n, a probability table can be obtained based on the six abovementioned rules.

(b) Spare node

The rules for creating the probability table of the WSP node shown in Fig. 9(b) are explained. Because the CSP and HSP nodes are types of WSP nodes (where the dormancy factor (α) is *0* for CSP and *1* for HSP), only the WSP node is described. In the table, each blank that is defined by x, y, and f can be filled on the basis of the following rules:

A. If $f > x, y$,
0.

B. If $f < x, y$,
$Pr\{a_{1F}$ fails at the fth interval$\} \cdot Pr\{a_{2F}$ fails at or before the fth interval$\} + Pr\{a_{1F}$ fails before the fth interval$\} \cdot P\{a_{2F}$ fails at the fth interval$\}$.

C. If $x < f < y$,
$Pr\{a_{2F}$ fails at the fth interval$\}$.

D. If $y \leq f < x$,
$Pr\{a_{1F}$ fails at the fth interval$\}$.

E. If $f = x < y$,
$Pr\{a_{1F}$ doesn't fail before the fth interval$\} \cdot Pr\{a_{2F}$ fails at or before the fth interval$\} + Pr\{a_{1F}$ fails before the fth interval$\} \cdot Pr\{a_{2F}$ fails at the fth interval$\}$.

F. *1 – (sum of the other probabilities in the same row).*

When the value of $Pr\{a_{2F}$ fails at the fth interval$\}$ is calculated in rules ii, iii, and v, the period of the spare status of a_{2F} should be distinguished from the period of the active status of a_{2F}, because the failure rates of each status differ in terms of the dormancy factor. In the dynamic fault tree, the inputs of the spare gate are only basic events[13]. If the RGGG also allows a spare node to have only basic events, which means that n_1 and n_2 have no input, the probability table can be filled on the basis of rules ii and vi because the x and y values are both ∞. Therefore, the task of filling the table becomes simple.

(c) SEQ node

The rules for creating the probability table of the SEQ node shown in Fig. 9(c) are explained. The SEQ node only allows basic events as inputs excepting n_1; because if n_2 and n_3 have inputs, the SEQ node cannot constrain the failure order of the inputs. Therefore, only the case in which y and z are ∞ is described. Each blank under that case can be filled in on the basis of the following rules:

A. If $g < 3$,
0.

B. If $3 \leq g < x+2$,
$\Sigma_{\forall a,b,c} \Pr\{a_{1G}$ fails at the ath interval$\} \cdot Pr\{a_{2G}$ fails at the bth interval$\} \cdot Pr\{a_{3G}$ fails at the cth interval$\}$,
when $a + b + c = g$.

C. If $g \geq x+2$,
$Pr_1 + Pr_2$.
$Pr_1 = \Sigma_{\forall a,b,c} \Pr\{$ *a_{1G} fails at the ath interval*$\} \cdot Pr\{a_{2G}$ *fails at the bth interval*$\} \cdot Pr\{a_{3G}$ *fails at the cth interval*$\}$,
when $1 \leq a < x–1$, and $a + b + c = g$.
$Pr_2 = Pr\{a_{1G}$ does not fail before the xth interval$\}$ $\Sigma_{\forall a,b,c} \Pr\{a_{2G}$ fails at the bth interval$\} \cdot Pr\{a_{3G}$ fails at the cth interval$\}$,
when $b + c = g – x$.

D. If $g = \infty$,
1 – (sum of the other probabilities in the same row).

In rule ii, the bth interval does not mean the real bth interval in the total process time, rather it indicates that a_{2G} fails after b intervals from the interval in which a_{1G} fails. For example, if $g = 7$ and $a = 2$, $b = 3$, and $c = 2$, then a_{1G} fails at the 2nd interval, a_{2G} fails at the 5th interval, and a_{3G} fails at the 7th interval. This outcome is due to the properties of the SEQ node, where a_{2G} cannot fail before a_{1G} fails and a_{3G} cannot fail before a_{2G} fails.

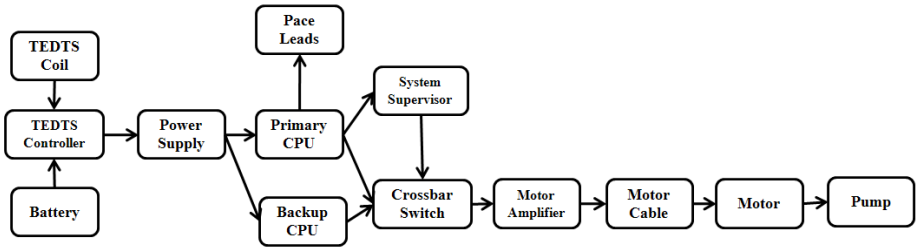

Fig. 11 Block diagram of the cardiac assist system.

As the dynamic RGGG uses a discrete time method, it can avoid the state space explosion problem of the Markov chain method, which is widely employed for dynamic reliability analyses. For example, assume that a dynamic system is composed of k components and the state of each component can be either a success or a failure. If this system is analyzed using the Markov method, 2^k states are needed. Therefore, as k increases, the complexity of the calculation increases exponentially. Whereas in the dynamic RGGG method when n is the number of time discretizations, $(n+1)^3$ blanks should be filled in the probability table of each node (component). Therefore, the total of blanks in the RGGG is $(n+1)^3 \cdot k$ and as k increases, the complexity of the calculation increases linearly with k, not exponentially. Consequently, the RGGG method has a great advantage, particularly when the target system is very complex.

3.1.3A software tool for the dynamic RGGG

The accuracy of the dynamic RGGG method is limited due to the assumption of discrete time, but it can be ensured as the number of time discretizations increases. A software tool to evaluate the dynamic RGGG was developed using the algorithms explained in Section 3.1.2. Therefore, the almost accurate results can be computed and the conventional static RGGG can also be estimated using the tool. The software tool is utilized to calculate the reliability of an example system in the following section.

3.1.4 Example

In this section, the ability of the proposed dynamic RGGG method is verified through application in a cardiac assist system[18, 19]. The block diagram of this system is shown in Fig.11 and it has two dynamic properties: the backup CPU is a warm spare for the primary CPU and failure of either the crossbar switch or the system supervisor results in failures in both the primary and backup CPU. A detailed explanation of the cardiac assist system can be found in References [18] and [19].

Figures 12 and 13 show the dynamic fault tree and dynamic RGGG for the example system, respectively. From these two figures, it can be seen that the RGGG models the system more intuitively than the dynamic fault tree. The RGGG has an almost identical structure as the block diagram of the actual system in Fig. 11.

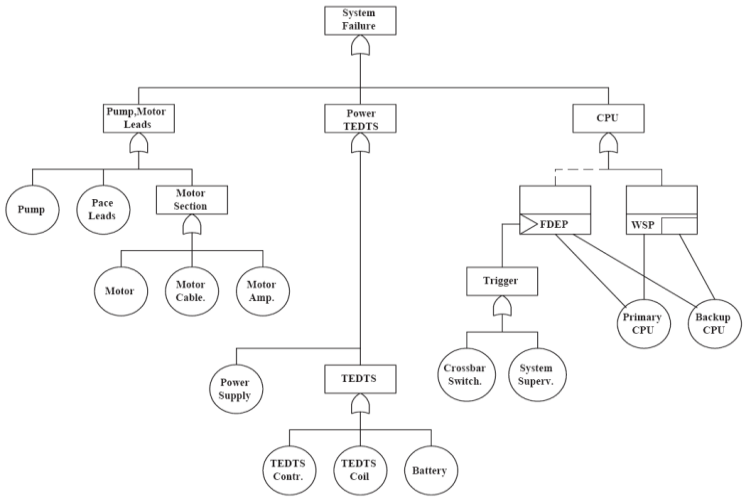

Fig. 12 Dynamic fault tree of the cardiac assist system.

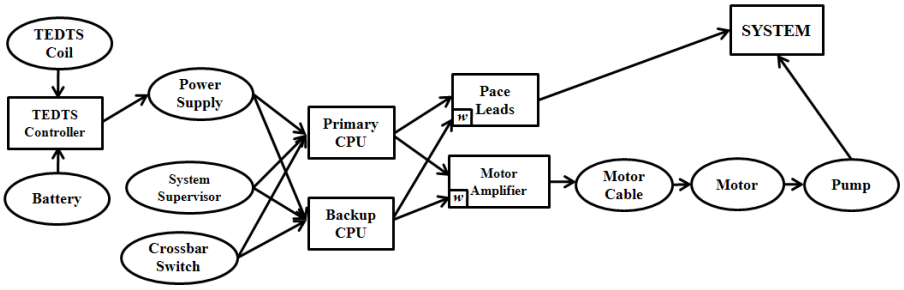

Fig. 13 Dynamic RGGG of the cardiac assist system.

In order to compare the evaluation result of each method, the dynamic fault tree is evaluated using a demonstration version of the commercial software tool Relex Studio 2008, and the reliability of the example system is determined to be 0.609544. The RGGG is evaluated using the software tool developed by the authors; Table 5 shows the evaluation results of the RGGG with an increasing discretization number (n). Errors are detected when the results of the RGGG are compared with those of Relex Studio, but it can be verified that the error becomes smaller as n increases; when n is *200*, the numerical difference is smaller than 10^{-5}.

Table 5 Evaluation results of the RGGG

n	Reliability
10	*0.6096437*
20	*0.6095939*
50	*0.6095641*
100	*0.6095541*
150	*0.6095508*
200	*0.6095492*

3.2 Repairable RGGG

Conventional risk analysis methods assume that the target system is a non-repairable system, which means that the systems are not repaired once it fails. Reliability is the probability that a system performs a specified function or mission under given conditions for a prescribed time. In non-repairable systems, reliability is a proper concept for representing system safety. However, many real world systems, such as automobiles, airplanes, computers, and nuclear power plants, are repairable systems. Repairable systems are those that are repaired when they fail. This is done by repairing or replacing the failed components in the system. Availability is defined as the probability of a system performing a specified function or mission under given conditions at a prescribed time[20]. Thus, the availability of repairable systems is focused on instead of the reliability.

3.2.1 Availability of simple repairable process

In repairable systems, two types of distribution are considered: failure distribution and repair distribution. A failure distribution describes the time required for a component to fail and a repair distribution describes the time required to repair a component. The availability of a simple repairable process requires a Markov analysis. Figure 14 presents a Markov transition diagram of a repairable process with constant failure and repair rates[21].

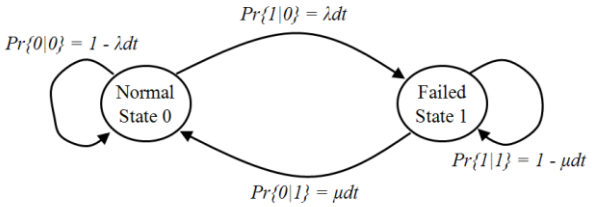

Fig. 14 Markov transition diagram for a simple repair process.

In a repairable system, components have two states: a normal state and a failed state. Let $x(t)$ be an indicator variable defined by $x(t) = 1$, if the component is in a failed sate at time t, and $x(t) = 0$, if the component is in a normal state at time t. The definition of the conditional failure rate λ and repair rate μ can be used to give:

$$
\begin{aligned}
&Pr\{1|0\} \equiv Pr\{x(t+dt)=1|x(t)=0\} = \lambda dt \\
&Pr\{0|0\} \equiv Pr\{x(t+dt)=0|x(t)=0\} = 1 - \lambda dt \\
&Pr\{1|1\} \equiv Pr\{x(t+dt)=1|x(t)=1\} = 1 - \mu dt \\
&Pr\{0|1\} \equiv Pr\{x(t+dt)=0|x(t)=1\} = \mu dt
\end{aligned} \qquad (24)
$$

Table 6 Probability table for an OR node in a repairable system

	$y_1 = 1$ (success)		$y_1 = 0$ (failure)		
	$y_2 = 1$ (success)		$y_2 = 0$ (failure)	$y_2 = 1$ (success)	$y_2 = 0$ (failure)
$y_A = 1$ (success)	$\frac{\mu_1}{\lambda_1+\mu_1}+\frac{\mu_2}{\lambda_2+\mu_2}-\frac{\mu_1}{\lambda_1+\mu_1}\cdot\frac{\mu_2}{\lambda_2+\mu_2}$		$\frac{\mu_1}{\lambda_1+\mu_1}$	$\frac{\mu_2}{\lambda_2+\mu_2}$	0
$y_A = 0$ (failure)	$1-\left(\frac{\mu_1}{\lambda_1+\mu_1}+\frac{\mu_2}{\lambda_2+\mu_2}-\frac{\mu_1}{\lambda_1+\mu_1}\cdot\frac{\mu_2}{\lambda_2+\mu_2}\right)$		$\frac{\lambda_1}{\lambda_1+\mu_1}$	$\frac{\lambda_2}{\lambda_2+\mu_2}$	1

The term $Pr\{x(t+dt)=1|x(t)=0\}$ is the probability of failure at $(t+dt)$, given that the component is working at time t, and so on.

Unavailability is the reverse concept of availability. That is, unavailability $Q(t)$ equals $1 - A(t)$. In this repairable process, the unavailability $Q(t+dt)$ is the probability of $x(t+dt)=1$, which is expressed in terms of the two possible states of $x(t)$ and the corresponding transitions to $x(t+dt)=1$:

$$\begin{aligned} Q(t+dt) &= Pr\{x(t+dt) = 1\} \\ &= Pr\{1|0\}Pr\{x(t)=0\} + Pr\{1|1\}Pr\{x(t)=1\} \\ &= \lambda dt[1-Q(t)] + (1-\mu dt)Q(t), \end{aligned} \quad (25)$$

This identity can be rewritten as:

$$\begin{aligned} &Q(t+dt) - Q(t) = dt(-\lambda-\mu)Q(t) + \lambda dt \\ &dQ(t)/dt = -(\lambda+\mu)Q(t) + \lambda, \end{aligned} \quad (26)$$

with the initial condition at $t=0$ of $Q(0) = 0$,
and the solution of this linear differential equation is:

$$Q(t) = \frac{\lambda}{\lambda+\mu}(1-e^{-(\lambda+\mu)t}), \quad (27)$$

Since $Q(t) = 1 - A(t)$,

$$A(t) = \frac{\mu}{\lambda+\mu}(1-e^{-(\lambda+\mu)t}), \quad (28)$$

Finally, the steady state availability can be obtained as follows:

$$A(\infty) = \frac{\mu}{\lambda+\mu}. \quad (29)$$

This result is used for analyzing the availability in the RGGG method.

3.2.2 Independent repairable system

If there are sufficient repairmen for a repairable system, then every component can be repaired immediately upon failure. Assume that other components cannot be affected when one component fails or is being repaired. Then, each component is independent of the other components' behavior. This is called an independent repairable system. Table 6 shows the probability table for an OR node with two inputs in a repairable system when λ_i, and μ_i represent the failure rate and repair rate of an arc from node i to target node A, respectively. The probability tables for an AND node and K-out-of-N node in a repairable system can be similarly derived.

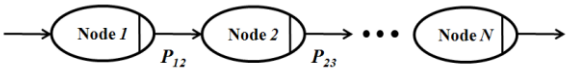

Fig. 15 Model of the RGGG for a dependent series repairable system.

3.2.3 Dependent series repairable system

When one component fails in a series repairable system, then the system also fails. Therefore, the system is immediately in a repair state if one component fails. At this time, if other components of this series system shut down and fail no more, each component is no longer independent. In this system, it is not possible for two or more components to be simultaneously in the repair state. This dependence is defined as a shut down dependence. The model of the RGGG for the dependence series repairable system is shown in Fig. 15. The thick outline of the nodes denotes that the components are repairable and a vertical line at the right of the node indicates that each component in the series system has the

characteristic of shutdown dependence. In this dependent series repairable system, the notation definitions are as follows:

$P_i = Pr\{y_i$ in the success or normal state$\}$

$$P_{ij} = A_{ij} = \frac{\mu}{\lambda + \mu}. \quad (30)$$

The components cannot be in the repair state simultaneously; therefore, the calculation algorithm should be as follows:

$$P2 = P_1P_{12}/(1-(1-P_1)(1-P_{12})). \quad (31)$$

Using this formula, the probability table can be determined and is shown in Table 7.

Table 7 Probability table for a node in the dependent series repairable system

	$y_1 = 1$ (success)	$y_1 = 0$ (failure)
$y_2 = 1$ (success)	$\frac{\mu}{\mu + P_1\lambda}$	0
$y_2 = 0$ (failure)	$1 - \frac{\mu}{\mu + P_1\lambda}$	1

3.2.4 K/M redundant parallel repairable system

A K-out-of-M redundant parallel repairable system is similar to an independent K-out-of-M system. When *k* or more components are at a normal state among *m* parallel input nodes, the system is in a normal state. In contrast, when less than the *k* components are in a normal state, the system fails. That is, if the number of components in the repair state is more than *(m-k)*, the system is in a repair state. If the system shuts down and the other *(k-1)* components shut down and fail no more when the *(m-k+1)* components are in the repair state, the components are no longer independent. This dependence is defined as a shutdown dependence. It requires another assumption in the K-out-of-M redundant parallel repairable system. If there are not enough repairmen, then *m* components cannot be repaired at one time. This dependence is defined as the repair dependence. If there are only *L* repairmen (*L*<*m*), only *L* components can be repaired at the same time.

Figure 16 presents the model of the RGGG for the K-out-of-M redundant parallel repairable system. The thick outline of the nodes represents the components that are repairable and the D-shaped notation is added at the right of the node, which means that all input components in the redundant parallel repairable system have characteristics of shutdown dependence. Additionally, in the D shape, the repair dependence can be represented by writing the number of repairmen denoted by *L*.

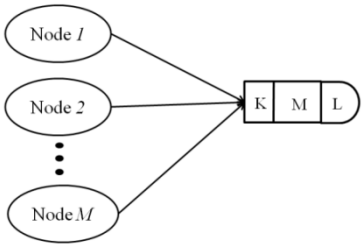

Fig. 16 Model of the RGGG for the K/M redundant parallel repairable system.

The system with *m* identical components has *(K+1)* possible states which, respectively, express the state with *0, …, K* components at the failure state and are denoted by the numbers *0, …, K*. Figure 17 shows a state transition diagram of a system with *m* identical components. P_i denotes the probability of the system at state *i*. The square represents the state and the arrow between the squares presents the state transition. The rate of transition from state *i-1* to *i* is a_i, whereas b_i expresses the rate of transition from state *i* to *i-1*.

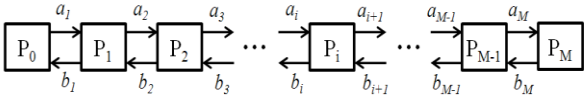

Fig. 17 State transition diagram of the system with *M* identical components.

P_i : probability of the system which has *i* number of components at repair state.

a_i : rate of transition from state *i-1* to *i*.

b_i : rate of transition from state i to *i-1*.

When the system is at a steady state, both a_i and b_i are constant and the state probability P_i is not changed. Therefore, the probability shifted in any state equals the one shifted out of the same state. Then, the following relations can be obtained:

Table 8 Probability table for a node in the K/M redundant parallel repairable system (*K=2, M=3, L=1*)

y_1	1				0			
y_2	1		0		1		0	
y_3	1	0	1	0	1	0	1	0
$y_M = 1$	$(\mu^2+3\lambda\mu) / (\mu^2+3\lambda\mu+6\lambda^2)$	$\mu / (\mu+3\lambda)$	$\mu / (\mu+3\lambda)$	0	$\mu / (\mu+3\lambda)$	0	0	0
$y_M = 0$	$6\lambda^2 / (\mu^2+3\lambda\mu+6\lambda^2)$	$3\lambda / (\mu+3\lambda)$	$3\lambda / (\mu+3\lambda)$	1	$3\lambda / (\mu+3\lambda)$	1	1	1

State *0* : $P_1 b_1 = P_0 a_1, \quad => \quad P_1 = P_0 \cdot (a_1/b_1)$

State *i*: $P_{i-1} a_i + P_{i+1} b_{i+1} = P_i a_{i+1} + P_i b_i$

$$P_{i+1} = P_o \prod_{j=1}^{i+1} \frac{a_j}{b_j} \quad , \quad i = 1 \ldots k\text{-}1 \qquad (32)$$

Assume λ is the failure rate of each component; then, the rate of transition is expressed as follows:

$$a_i = (m\text{-}i+1)\lambda. \qquad (33)$$

Assume μ is the repair rate of each component. At state *i*, *i* components fail, but the number of components that are being repaired is dependent on *L*. Therefore, the rate of transition is as follows:

$$\begin{aligned} b_i &= i\mu, && \text{when } i \le L, \\ &= L\mu && \text{when } i > L, \end{aligned} \qquad (34)$$

Then, the calculation formulas of the system normal state probability are obtained as follows:

$$\begin{aligned} G &= \Sigma_{i=0}^{k} P_i \\ P_R(\textit{Availability}) &= \Sigma_{i=0}^{m-k} \frac{P_i}{G}. \end{aligned} \qquad (35)$$

Using this formula, the probability table can be determined. Table 8 shows a probability table for a node in a 2-out-of-3 redundant parallel repairable system with 1 repairman.

3.2.5 Example

In this section, an example of the modeling and quantitative analysis for the charging pumps subsystem of a chemical and volume control system (CVCS) is introduced. The CVCS is a major support system for reactor coolant systems. The main function of the CVCS system is to inject water into the primary circuit in the case of a loss of coolant accident (LOCA) to prevent core meltdown. Figure 18 shows a diagram of the charging pumps subsystem where the interfaces with other subsystems are represented by polygonal boxes. In the schematics in Fig. 18, RCV is the reactor control volume and SS is the acronym for any subsystem in the CVCS partitioning.

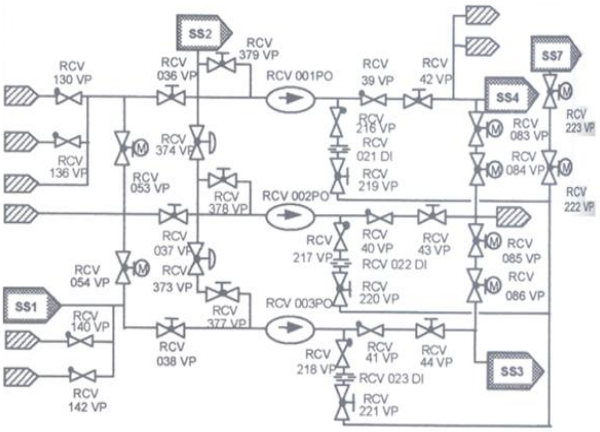

Fig. 18 P&ID of the charging pump subsystem.

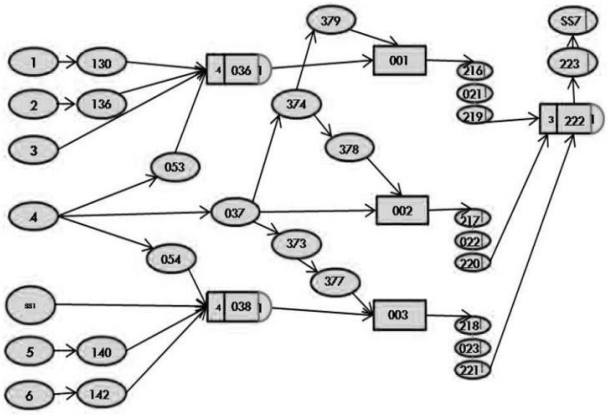

Fig. 19 Model of the RGGG for the repairable charging pump subsystem.

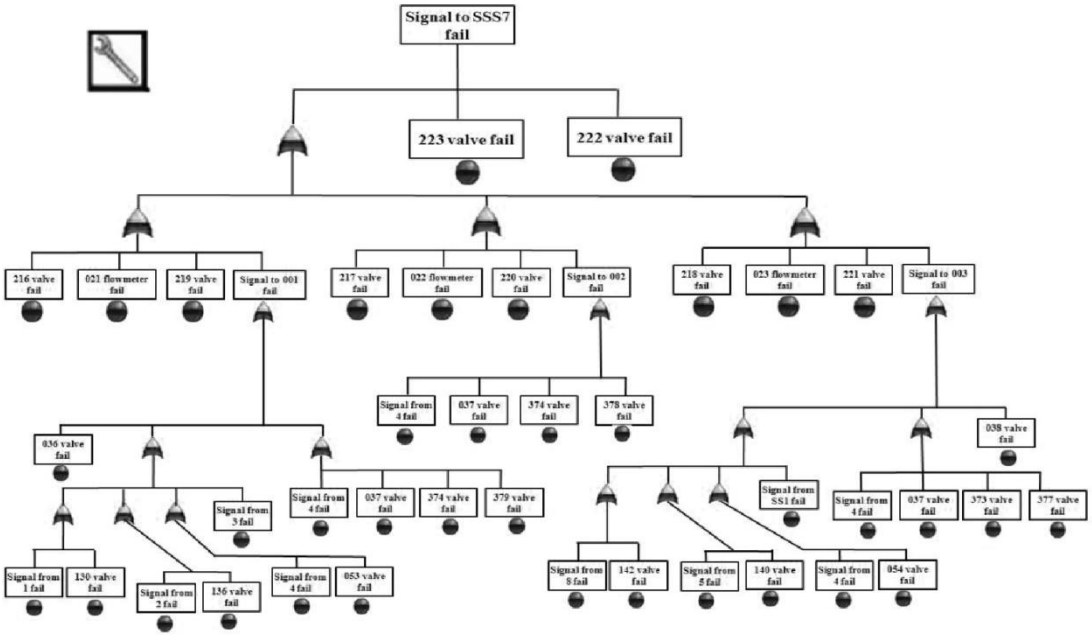

Fig. 20 Model of the RFT for the repairable charging pump subsystem.

This research focuses only on the operational probability (i.e. availability) of the output signal to subsystem 7, represented as SS7. Using the RGGG method, the charging pumps subsystem model is structured as shown in Fig. 19. The RGGG has a structure very similar to that of the charging pumps subsystem shown in Fig. 18. The availability of the system can be estimated using the probability tables proposed previously in this paper. The availability of the charging pumps subsystem was determined to be 0.995610 using MSBNx^{TM}, which is a software tool for Bayesian Belief Networks.

Figure 20 shows the modeling of the charging pumps subsystem using RFT. It can be seen that the RFT model is more complex than the RGGG model. To estimate the system's availability, the Markov chain analysis method is used. Figure 21 shows some Markov chain diagrams required for evaluating the availability of the charging pumps subsystem. The availability of the charging pumps subsystem is calculated to be 0.995610 using RELEX Studio, a software tool for fault tree and Markov chain analyses. It is confirmed that the availability estimation results from both methods are identical, while the repairable RGGG method provides a much easier method for modeling and understanding the actual structure of the system compared with the fault tree analysis.

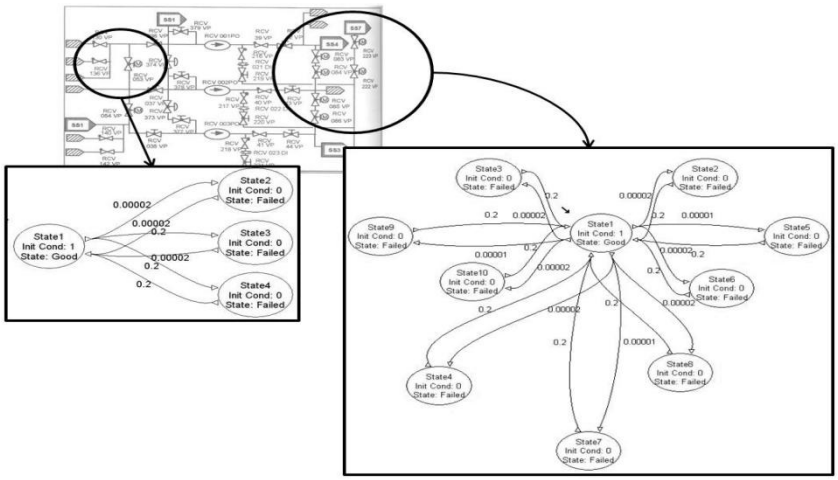

Fig. 21 Markov chain diagrams to evaluate the repairable charging pump subsystem.

4 Summary and conclusions

There are many methods of reliability analysis, such as fault trees, reliability graphs, reliability block diagrams, Markov chains, and Monte Carlo simulations. Among the existing methods, the fault tree analysis is the most widely used method due to its expression power, applicability to complex systems, and various tool supports. However, because analysts must draw a fault tree based on the logical relationships among the components in a system, the use of the fault tree analysis is becoming more and more cumbersome as systems become more complex. To overcome the limitations of fault tree method, the reliability graph with general gates (RGGG) method is proposed by introducing general gates to a conventional reliability graph method. The reliability graph is particularly attractive due to its intuitiveness, but its most serious shortcoming is its limitation in the expression power. However, by introducing general gates to the reliability graph, the expression power is improved so that the RGGG can create a one-to-one match from the actual structure of a system to the reliability graphs of the system. Also, a quantitative evaluation method is proposed by transforming the RGGG to an equivalent Bayesian network without losing the intuitiveness of the model. The practicability of the RGGG method has been confirmed through applications to a simple system and an existing complex system. It has been shown that the RGGG method provides a much easier way to model the systems compared with the fault tree method and the results from both methods are equivalent. Furthermore, for complex systems, the RGGG provides more accurate results compared with the fault tree because the results of the RGGG do not possess truncation errors, which are generally included in large fault tree analysis.

As with the conventional fault tree method, the RGGG method has been developed for the reliability analysis of non-repairable static systems. In order to utilize the RGGG to estimate the reliability of dynamic systems and the availability of repairable systems without losing the advantages of the conventional RGGG, dynamic RGGG and repairable RGGG have been developed.

It was found that the accuracy of the dynamic RGGG method is limited due to the assumption of a discrete time, but it is ensured that the results are accurate as the number of time discretizations increases. For a simple dynamic system, the dynamic fault tree and the Markov chain methods can be used to analyze the system reliability without difficulty, but as the system becomes more complex, the dynamic RGGG method is more useful than those methods. From the viewpoint of a quantitative analysis, the complexity of the Markov chain increases exponentially as the number of the system components increases, whereas the complexity of the RGGG increases linearly with the increasing system components. For that reason, the RGGG does not undergo a state space explosion problem. Moreover, the structure of the RGGG is almost identical to the block diagram of the actual system.

By applying the RGGG method to various repairable systems, the RGGG method that has been extended for repairable systems was found to have the same characteristic intuitiveness as the original RGGG method. In addition, the repairable RGGG method and the repairable fault tree method are compared by applying these methods to identical existing systems. The availability analysis result from the repairable RGGG method is identical to the result from the fault tree analysis.

In conclusion, the RGGG method is believed to handle target systems more easily and intuitively compared with other methods, even when the system is dynamic or repairable, while its analysis result is the same as those of other methods.

References

[1] US NRC: PRA Procedures Guide: a Guide to The Performance of Probabilistic Risk Assessments for Nuclear Power Plants (NUREG/CR-2300), Washington, DC: US Nuclear Regulatory Commission, 1983.

[2] KIM, M. C., SEONG, P. H.: Reliability Graph with General Gates: an Intuitive and Practical Method for System Reliability Analysis, Reliability Engineering and System Safety, 2002, 78(3):239-246.

[3] SHIN, S. K., SEONG, P. H.: Adding Dynamic Nodes to RGGG and Making Probability Tables, Trans. American Nuclear Society, 2007, 97:131-132.

[4] SHIN, S. K., SEONG, P. H.: Review of Various Dynamic Modeling Methods and Development of an Intuitive

Modeling Method for Dynamic Systems, Nuclear Engineering and Technology, 2008, 40(5):375-386

[5] GOH, G. T., SHIN, S. K., SEONG, P. H.: Analysis of Repairable System Using Reliability Graph with General Gates, Trans. American Nuclear Society, 2009, 101:529-530.

[6] TORRIERI, D.: Calculation of Node-pair Reliability in Large Networks with Unreliable Nodes, IEEE Trans. Reliability, 1994, 43(3):375-382.

[7] AGGARWAL, K. K., GUPTA, J. S., MISRA, K. B.: A Simple Method for Reliability Evaluation of a Communication System, IEEE Trans. Communication, 1975, 23:563-565.

[8] JENSEN, F. V.: An Introduction to Bayesian Networks, New York: Springer, 1996.

[9] BOUISSOU, M., MARTIN, F., OURGHANLIAN, A.: Assessment of a Safety-Critical System Including Software: a Bayesian Belief Network for Evidence Sources, Proceedings of Annual Reliability and Maintainability Symposium, Washington DC, USA, IEEE Press, 1999.

[10] FENTON, N. E., LITTLEWOOD, B., NEIL, M., STRIGINI, L. SUTCLIFFE, A, WRIGHT, D.: Assessing Dependability of Safety Critical Systems Using Diverse Evidence. IEE Proceedings Software Engineering, 1998, 145(1):35-39.

[11] FENTON, N. E., NEIL, M.: Software Metrics: Successes, Failures and New Directions. Journal of Systems and Software, 1999, 47(2):149-157.

[12] CHERNYAK, A. A.: Combinatorial-graphic Method of Reliability Analysis of Complex Systems with Monotone Boolean Functions. Automation and Remote Control, 1991, 52(4):572-579.

[13] DUGAN, J. B., BAVUSO, S. J., BOYD, M. A.: Dynamic Fault-tree Models for Fault-tolerant Computer Systems. IEEE Transactions on Reliability, 1992, 41(3):363-377.

[14] BOBBIO, A., RAITERI, D. C.: Parametric Fault Trees with Dynamic Gates and Repair Boxes, Proceedings of Annual Reliability and Maintainability Symposium, Los Angeles, USA, IEEE Press, 2004.

[15] RAITERI, D. C., FRANCESCHINIS, G., IACONO, M., VITTORINI, V.: Repairable Fault Tree for the Automatic Evaluation of Repair Policies, Proceedings of the International Conference on Dependable Systems and Networks, Firenze, Italy, IEEE Press, 2004

[16] BOUDALI, H., DUGAN, J. B.: A Discrete-time Bayesian Network Reliability Modeling and Analysis Framework. Reliability Engineering & System Safety, 2005, 87(3):337-349.

[17] GALAN, S. F.,DIEZ, F. J.: Networks of Probabilistic Events in Discrete Time. International Journal of Approximate Reasoning, 2002, 30(3):181-202.

[18] REN, Y., DUGAN, J. B.: Design of Reliable Systems Using Static & Dynamic Fault Trees. IEEE Transactions on Reliability, 1998, 47(3):234-244.

[19] OU, Y., DUGAN, J. B.: Sensitivity Analysis of Modular Dynamic Fault Trees, Proceedings of IEEE International Computer Performance and Dependability Symposium, Chicago, USA, IEEE Press, 2000.

[20] McCORMICK, N. J.: Reliability & Risk Analysis: Methods and Nuclear Power Application, New York: Academic Press, 1981.

[21] KUMAMOTO, H., HENLEY, E. J.: Probabilistic Risk Assessment and Management for Engineers and Scientists, 2nd ed., New York: IEEE Press, 1996.