ELSEVIER

0951-8320(95)00094-1

Reliability Engineering and System Safety **50** (1995) 271–284

0951-8320/95/$9.50

Design for safety of engineering systems with multiple failure state variables

J. Wang[a], T. Ruxton[a] & C. R. Labrie[b]

[a]*School of Engineering and Technology Management, Liverpool John Moores University, Liverpool L3 3AF, U.K.*
[b]*Engineering Design Centre, University of Newcastle upon Tyne, Newcastle upon Tyne NE1 7RU, U.K.*

(Received 18 May 1993; revised 15 August 1995; accepted 29 August 1995)

Since possible failure events of large engineering systems with a higher level of innovation may not be identified by experience or from previous accidents and incident reports of similar systems, and since 'design for safety' of such systems requires no omission of failure causes associated with possible system failure events, a top-down approach is not always satisfactorily applied in the risk identification and risk estimation phases and a more objective and flexible bottom-up approach may be more effective.

This paper proposes an inductive bottom-up risk identification and estimation methodology combining Failure Mode, Effects and Criticality Analysis (FMECA) and the Boolean Representation Method (BRM). This methodology can be used to identify all possible system failure events and associated causes, and to assess the probabilities of occurrence of them particularly in those cases where multiple state variables and feedback loops are involved. The Boolean representation method is presented together with its use in modelling cause and effect relationships. The overall model and the algorithms are described and tested in association with the associated computer software. The applications of this methodology in association with other formal safety modelling methods are discussed. An illustrative example is presented to demonstrate the methodology.

1 INTRODUCTION

'Design for safety' of an engineering system is a process of identifying the possible failure events (top events) and the associated consequences, estimating them, and finally evaluating them. It provides the designer with a systematic approach to identify high risk areas and attain explicit levels of safety by identifying and implementing ways to reduce the hazard frequency of occurrence and the extent of respective consequences. In such a process, risk identification and risk assessment may be the most difficult and important steps that always attract a great deal of attention by safety researchers.

Given the system description and functional requirements, risk identification consists of identifying the system top events, for which all the possible associated causes and corresponding consequences must be identified.[12,16] The risk identification phase in the 'design for safety' process is, without question, the most critical. Risk identification requires the combined expertise and insight of engineers and scientists to cover all aspects of the system process and operation to systematically decompose the system and analyze the interactions of primary and intermediate events on system safety and performance.[15,16]

On the basis of the information produced from the risk identification phase, risk estimation can be carried out. Risk estimation is a process of estimating the likelihood of occurrence of the identified hazards and the severity of respective potential consequences. Information produced from the risk estimation phase may help designers to minimise the possibilities or possible consequences of critical system failures, to be aware of the characteristics and priorities of components for design actions, and to provide a safe and reliable product design.[19] Risk estimation involves expressing the occurrence of each top event in terms of the simultaneous occurrence of the associated basic events, (i.e. minimal cut sets), and expressing the severity of possible resulting consequences in terms of property loss, injury and death of personnel and contamination of the environment.

Assumptions are always necessary for the convenient application of risk identification and risk estimation. The following typical assumptions may

often be made in the risk identification and risk estimation phases.

1. The components or subsystems at the same analysis level are considered to be independent.
2. A continuous variable may be expressed by two or more discrete states such as high, normal and low, each of which corresponds to a certain range of values.
3. Failures follow exponential distributions.
4. There is no preventive maintenance carried out during the mission.

Various safety analysis methods can be applied to identify and estimate risks. Fault Tree Analysis (FTA) and Failure Mode, Effects and Criticality Analysis (FMECA) are usually used to carry out such an analysis. For a system with a comparatively low level of innovation, the top events may be obtained by experience or from previous accidents and incident reports of similar systems, and the associated cut sets may be identified deductively using FTA which may make use of the information produced from FMECA. Being a top-down deductive method, FTA has the following problems:

- The top events of a system with a comparatively high level of innovation may not be identified.
- It is possible to make omissions of failure causes associated with the top events.
- The representation of variables with multiple states can prove to be comparatively complex. For example, the representation of a temperature variable T with five possible states (i.e. 1. high, 2. too high, 3. normal, 4. low, 5. too low) may require five gates in FTA, but such a variable may be represented simply by T_i ($i = 1, 2, \ldots, 5$) using the Boolean representation method described later, where T_i represents state i of variable T.
- FTA may not completely benefit from the information produced using FMECA to obtain the minimal cut sets associated with the system top events and neither may it directly make use of the information when a complex engineering system is analyzed.
- FTA may not address all the complex interactions present in a complex MTO product in an analytically rigorous manner.

Furthermore, when there is a lack of experience of similar system design solutions and when the complexity of the system and constituent elements increases, a top-down approach like FTA may prove unsuitable and a bottom-up approach may be preferred.

Generally, the decision as to which kind of approach is more appropriate for the analysis of a particular engineering system is dependent on the following considerations:

1. The level of the system breakdown at which the risk identification is carried out.
2. The degree of complexity of the interrelationships of the items at the investigated indenture level of the system breakdown.
3. The degree of innovation associated with the system design (i.e. the availability of product failure data for safety analysis).

A bottom-up approach may be effectively used to deal with the problems discussed above and may yield a higher degree of confidence that all system top events and associated cut sets are identified and no omissions have been made.[18,21,22] Using a bottom-up approach, information generated at a lower level (i.e. the component level) may be inductively related to the analysis at higher levels thus leading to the identification of all possible system top events and associated cut sets.

FMECA is such a bottom-up approach and is usually carried out on the basis of the evaluation of hardware elements. However, FMECA does not close the loop between risk identification and risk estimation.[8] In FMECA, how combinations of occurrence of failure modes affect system performance and safety is not studied. Some combinations of occurrence of failure modes result in definite occurrence of system failures. Such combinations of failure modes are required to be studied. Therefore, an inductive approach is required to efficiently process the information produced from FMECA to close the loop. The Boolean representation modelling is an approach which can be used to automate the construction of the system Boolean representation table to contain all the system top events and associated minimal cut sets. Due to its inductive nature, the Boolean representation method can fully benefit from the information produced from FMECA. An additional benefit of the Boolean representation method over FTA is that systems with feedback loops and multiple state variables can be easily modelled.

This paper proposes an inductive bottom-up Boolean Representation Method (BRM). The BRM is combined with FMECA to form an effective risk identification and risk estimation framework. This paper will describe the framework with particular emphasis on the modelling of systems with multiple state variables and feedback loops, and also possible combinations of BRM with other safety modelling techniques.

2 A PROPOSED RISK IDENTIFICATION AND RISK ESTIMATION FRAMEWORK

A methodology for risk identification and risk estimation of engineering systems is proposed as

shown in Fig. 1. This methodology combines FMECA and the BRM to systematically identify and assess all system top events and associated cut sets.

Having completed the risk identification phase using FMECA at the component level, the Boolean representation descriptions of the components of the subsystems of a system can be constructed. The failure modes as identified in the FMECA of a component can be used as the input attributes of the Boolean representation table. To reduce the degree of complexity of the Boolean representation modelling, only the failure modes with severity classes 1, 2 and 3 are used to construct the component Boolean representation table. Experience and a good understanding of the system is very important for the efficient construction of the component Boolean representation table. The component Boolean representation table describes, in the form of a table, the conditions which must exist for the occurrence of the identified component output states. The last column of the Boolean representation table describes the states of the output of the component being modelled while

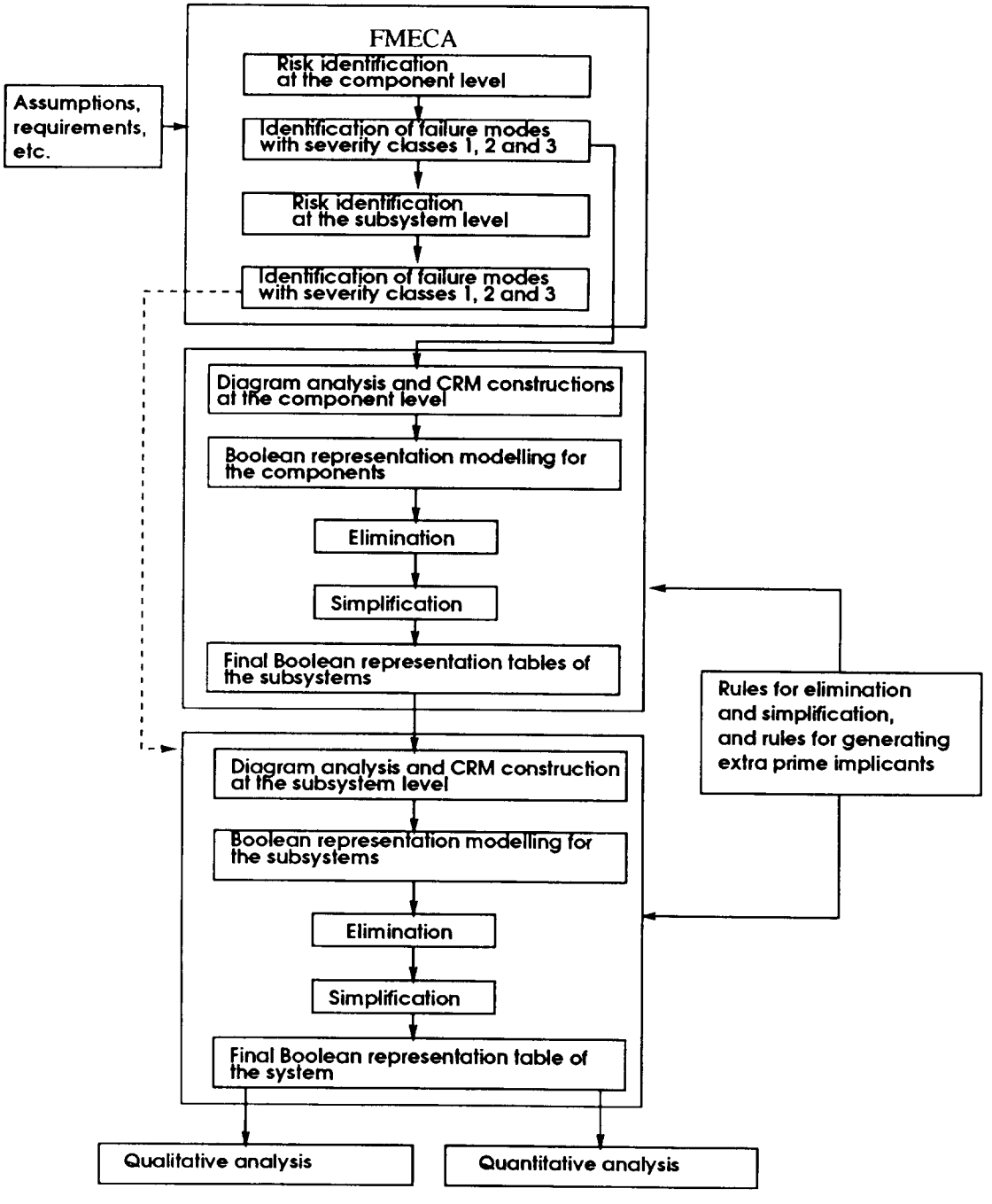

Fig. 1. An inductive bottom-up risk identification and risk estimation framework incorporating FMECA and BRM.

other columns prescribe the states of the input attributes. Each row represents a possible condition for an occurrence of the component's output state.

Constructed from the results of the FMECA, a component Boolean representation table normally has some degree of redundancy. The rules of simplification can be applied to absorb and merge redundant rows and redundant attributes to produce the irreducible Boolean representation table of the component. After all the Boolean representation tables of the components of a subsystem have been constructed, the construction of the subsystem Boolean representation table can be started using a process of aggregation. Intermediate variables need to be eliminated by substituting them with primary variables regarding the interactions of the components. A *Component Relationship Matrix* (*CRM*) can be constructed from the system process diagram to describe the component relationships for the purpose of eliminating intermediate variables. After the elimination, the rules of simplification should be applied again to produce the irreducible Boolean representation table of the subsystem.

After all the Boolean representation tables of the subsystems have been constructed, the Boolean representation modelling can be progressed up to the system level, and the same procedures repeated to ultimately obtain the irreducible Boolean representation table for the system. The rules of deduction of extra prime implicants can then be applied to the irreducible system Boolean representation table to obtain the final system Boolean representation table. The final system Boolean representation table contains all the prime implicants associated with the system output states. A prime implicant can be considered to be the equivalent of a cut set in fault tree analysis but for systems with *multiple state variables*.

If the risk identification phase is completed using FMECA at the subsystem level, the Boolean representation analysis can be carried out directly at that level. Both qualitative and quantitative analysis can be carried out on the basis of the obtained final system Boolean representation table.

In the following sections, FMECA, the components relational model, the rules and procedures for obtaining the final Boolean representation table for a system, and the algorithms for qualitative and quantitative analysis are described. For the simplification of the description, Boolean representation modelling at the component level is progressed directly up to the system level.

3 BOOLEAN REPRESENTATION METHOD

An engineering system can be described in terms of components and their interactions. A component can be modelled by a Boolean representation table which is an extended version of a truth table and which describes how each combination of input events specifies the output event or the state of the output. As described in the last section, Boolean representation modelling can make direct use of the information produced from FMECA to define the input attributes.[1,3–7,10,11,13,14] The Boolean representation table of a component can be constructed by studying all possible combinations of the input variable states. After all the Boolean representation tables of the components have been constructed, Boolean representation modelling can be progressed up to a higher level (i.e. the subsystem or system level) by studying the component relationships.

3.1 System modelling

Variables used in Boolean representation modelling can be classified in the following two categories:

1. Intermediate variable.
2. Primary variable.

The output from a component within the system is called an intermediate variable. Any variable which is an input from the system environment or an internal mode of a component is called a primary variable. An internal mode of a component represents its functioning. The examples of internal modes are '*Working*' and '*Failed*'. Each primary variable or intermediate variable may have several states. The investigated system states are top events.

As explained earlier, an engineering system can be described in terms of components and their interactions. Furthermore, a component can be described in the form of Boolean representation table involving primary and intermediate variables. The component relationships within the system can be described in the form of a Component Relationship Matrix (*CRM*) as follows:

$$CRM = \begin{bmatrix} M_{11} & M_{12} & M_{13} & . & M_{1n} \\ M_{21} & M_{22} & M_{23} & . & M_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ M_{n1} & M_{n2} & M_{n3} & . & M_{nn} \end{bmatrix}$$

In a *CRM*, if the element M_{ij} is equal to 0, it means that the output of component i is not an input to component j; if M_{ij} is equal to 1, it means that the output of component i is the output to component j; and if M_{ii} is equal to 1, it means that there is a self-feedback for component i.

Given the process diagram of a system, the components can first be labelled by integer numbers, and then the *CRM* can be constructed. Given the

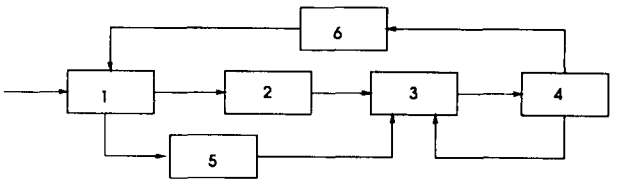

Fig. 2. A process system diagram.

diagram of a system shown in Fig. 2, the *CRM* is constructed as follows:

$$CRM = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The procedures for processing and manipulating the Boolean representation tables of the components to obtain the irreducible table of the system are based on the CRM. The construction of the Boolean representation table starts with the component Boolean representation model for which the output states are top events.

When a component has more than one output variable, Boolean representation modelling should be conducted for each of the output variables, and a 'dummy' component should be provided in the system process diagram for the *CRM* construction.[6,19] More than one Boolean representation description may be required to model a component.

3.2 Rules for Boolean representation manipulation

Based on the binary logic relationships, the rules for manipulation of Boolean representation tables involving variables with multiple states are defined as follows:

1. Definition

$$A_i \cap 1 = A_i \quad (1)$$

$$A_i \cap 0 = 0 \quad (2)$$

$$A_i \cup 1 = 1 \quad (3)$$

$$A_i \cup 0 = -A_i \quad (4)$$

$$\sum_{i=1}^{n} A_i = 1 \quad (5)$$

$$A_i \cap A_j (i \neq j) = 0 \quad (6)$$

2. Identities

$$A_i \cap A_i = A_i \quad (7)$$

3. Commutative law

$$A_i \cap B_i = B_i \cap A_i \quad (8)$$

4. Associative laws

$$A_i \cap (B_i \cap C_k) = (A_i \cap B_j) \cap C_k \quad (9)$$

$$A_i \cup (B_j \cap C_k) = (A_i \cup B_j) \cap (A_i \cup C_k) \quad (10)$$

5. Absorption laws

$$A_i \cup (A_i \cap B_j) = A_i \quad (11)$$

$$A_i \cap (A_i \cap B_j) = A_i \cap B_j \quad (12)$$

where A_i represents state i of variable A, A_j represents state j of variable A and B_j represents state j of variable B.

The rules for Boolean representation simplification are absorption and merging. Two examples of their applications are shown in Tables 1 and 2, where the number of the states of variable B is equal to 3, and F, W and N stand for '*Failed*', '*Working*' and '*Normal*', respectively.

3.3 Elimination of intermediate variables

The input entries of a final system Boolean representation table should be primary variables. Therefore, intermediate variables should be eliminated by substitution with primary variables. During the elimination process, some intermediate variables may be used to replace other intermediate variables. Gradually, all intermediate variables are eliminated and a Boolean representation table in which all the entries are primary variables is obtained. At this stage, a simplification of the Boolean representation table can be carried out. If the number of the entries of a Boolean representation table is large the simplification process may prove time-consuming. Therefore, it is suggested that the simplification rules be applied after each intermediate variable is eliminated. An example

Table 1. Absorption

A	B	C_{output}		A	B	C_{output}
N	*	High	->	N	*	High
N	N	High				

Table 2. Merging

A	B	C_{output}		A	B	C_{output}
F	F	High	->	F	*	High
F	W	High				
F	N	High				

of elimination of intermediate variables is presented as shown in Tables 3 and 4. If

$$Y = A_iB_iE_i + A_iB_j$$

and $$E_i = C_iD_i + C_jD_j$$

Then $$Y = A_iB_i(C_iD_i + C_jD_j + A_iB_j)$$

$$= A_iB_iC_iD_i + A_iB_iC_jD_j + A_iB_j$$

where A, B, C and D are primary variables, and E is an intermediate variable.

Eliminating intermediate variable E, Table 4 is obtained.

An input variable should only occupy one column in a Boolean representation table. However, it may happen that an input variable may occupy more than one column during the elimination of intermediate variables. This is called duplication of variables. Duplication of variables has been found to arise only in the construction of Boolean representation tables of systems in which one or more of the components has multiple outputs. Duplication of variables can be eliminated by applying the following rule in association with rules (6) and (7):

$$V_i \cap (*) = V_i \tag{13}$$

where V_i represents state i of variable V and * stands for 'Don't care'.

During the elimination of intermediate variables, if the combination of a variable in a row is 0, that row is deleted. An example is shown in Table 5 where row 2 is eliminated.

Table 3. The tables concerned with variables *Y* and *E*

A	*B*	*E*	*Y*		*C*	*D*	*E*
F	W	N	High	and	N	N	N
F	N	*	High		F	W	N

Table 4. The Boolean representation table after elimination

A	*B*	*C*	*D*	*Y*
F	W	N	N	High
F	W	F	W	High
F	N	*	*	High

Table 6. Elimination of an output variable appearing in input attributes

Row	*A*	*B*	C_{out}	C_{out}		*A*	*B*	C_{out}
1	N	F	F	F		N	F	F
2	F	N	N	F	->			
3	F	F	*	F		F	F	F

Table 7. An example of extra prime implicants

Row	*A*	*B*	C_{out}
1	N	F	High
2	F	*	High

The difference between the Boolean representation descriptions of systems with and without feedback loops is that the former has the output variable in the input attributes of the Boolean representation table, and the latter does not. For a system with feedback loops, the output variable in the input attributes of the Boolean representation table can be eliminated by applying the rules (6), (7) and (13). An example is shown in Table 6 where row 2 is eliminated.

3.4 Deduction of prime implicants

A Boolean representation table can be simplified to an irreducible form using the described rules. However, the irreducible table is not guaranteed to contain all of the prime implicants since variables with multiple states may be involved. An example is given in Table 7, where the number of states of variable A is equal to 2.

Obviously, Table 7 is an irreducible table. However, there is one more prime implicant $[A = *]$ $[B = F]$, which is not contained in Table 7. As will be described later, such an extra implicant can be produced from the existing irreducible table.

Quine's algorithm theory can be used to produce the extra prime implicants from the obtained irreducible table.[6,19] Such a method is called consensus operation since it creates new terms out of the terms already in the table by mixing and matching their input events. The theory for obtaining the extra prime

Table 5. An example of elimination of duplicative input variables

Row	A	B	C	C	D	C_{out}		A	B	C	D	C_{out}
1	N	F	*	*	N	F		N	F	*	N	F
2	N	N	F	N	F	F	->					
3	N	F	*	F	F	F		N	F	F	F	F
4	N	N	F	F	W	F		N	N	F	W	F

implicants from the irreducible table is described as follows:

If there is an event variable A and a set of n prime implicants $\sigma_1, \sigma_2, \ldots, \sigma_n$ associated with all the possible states ($A_1, A_2, \ldots$, and A_n) of variable A in the irreducible Boolean representation table, $\prod_{j=1}^{n}\sigma_j$ is also a prime implicant provided that it exists. This can be proved as follows:

Suppose Y represents the total prime implicants associated with all the possible states of variable A. Then

$$Y = \sum_{i=1}^{n} A_i\sigma_i \tag{14}$$

where n is the number of the states of variable A.

From rules (1), (4) and (7), the following equation can be obtained.

$$A_i\sigma_i = A_i\sigma_i \cap \left(1 \cup A_i\prod_{j=1}^{n}\sigma_j\right) = A_i\sigma_i \cup A_i\prod_{j=1}^{n}\sigma_i \tag{15}$$

Therefore

$$Y = \sum_{i=1}^{n}\left(A_i\sigma_i \cup A_i\prod_{j=1}^{n}\sigma_j\right) = \sum_{i=1}^{n} A_i\sigma_i \cup \sum_{i=1}^{n}A_i\prod_{j=1}^{n}\sigma_j \tag{16}$$

Since

$$\sum_{i=1}^{n} A_i = 1 \tag{17}$$

Then

$$Y = \sum_{i=1}^{n} A_i\sigma_i \cup \prod_{j=1}^{n}\sigma_j \tag{18}$$

Therefore $\prod_{j=1}^{n}\sigma_j$ is also a prime implicant.

The extra prime implicants created out of the obtained irreducible Boolean representation table should be added to the obtained irreducible Boolean representation table, and the rules for simplification should be applied again to obtain the final Boolean representation table. An example is shown as follows:

Suppose an irreducible system Boolean representation table is shown in Table 8, where the number of the states of variable B is equal to 2.

Deducing the extra prime implicant, Table 9 is obtained.

Row 3 is the new prime implicant.

The final system Boolean representation table can be obtained by applying the rules for simplification. (Table 10)

It should be pointed out that it is meaningless to study extra prime implicants in fault tree analysis because only one state (i.e. failure) for a variable appears in the minimal cut sets. For a system in which multiple state variables contribute to system failures, the failure cause expressions are prime implicants rather than minimal cut sets in the fault tree analysis. If the state of each variable in a system is 1, the final Boolean representation table would be exactly the same as obtained in the fault tree analysis.

Table 8. An irreducible system Boolean representation table

Row	A	B	E	F	C_{output}
1	N	F	F	F	High
2	*	N	F	F	High

Table 9. Deduction of the extra prime implicant

Row	A	B	E	F	C_{output}
1	N	F	F	F	High
2	*	N	F	F	High
3	N	*	F	F	High

Table 10. The final system Boolean representation table

Row	A	B	E	F	Y_{output}
1	N	*	F	F	High
2	*	N	F	F	High

3.5 System safety analysis

Both qualitative and quantitative safety analysis can be carried out on the basis of the final system Boolean representation table. Such an analysis is described as follows.

3.5.1 Qualitative analysis

In the obtained Boolean representation table, a prime implicant consisting of n primary events is called an n-event prime implicant. One-event prime implicants are significant contributors to the associated top event unless their probabilities of occurrence are very low. If there are no one-event prime implicants, two or three-event prime implicants leading to the top event should be given more attention rather than other higher-order prime implicants. Common cause failures should also be studied if there are some common causes in higher-order prime implicants.

3.5.2 Quantitative analysis

Boolean representation analysis deals with variables with multiple states. The traditional quantitative safety analysis theory which usually deals with variables with single failure state cannot be directly applied to the final system Boolean representation table. Therefore, a modified quantitative safety analysis method is required to assess the probability of occurrence of each system top event. Such a method is developed as follows:

The simultaneous occurrence of the basic events associated with any of the prime implicants C_1, C_2,

$C_3, \ldots,$ and C_N will result in the occurrence of the top event T_c. Thus, the probability of occurrence of the top event T_c can be calculated as follows:

$$\begin{aligned} P(T_c) &= P(C_1 \cup C_2 \cup \ldots \cup C_N) \\ &= (P(C_1) + P(C_2) + \ldots P(C_N)) - (P(C_1 \cap C_2) \\ &\quad + P(C_1 \cap C_3) + \ldots P(C_i \cap C_j)_{[i \neq j]} \ldots) \ldots \\ &\quad + (-1)^{N-1} P((C_1 \cap C_2) \ldots \cap C_N) \\ &= \sum_{i=1}^{N} P(C_i) - \sum_{i=1 i \neq j}^{N} P(C_i \cap C_j) + \ldots \\ &\quad + (-1)^{N-1} P(C_1 \cap C_2 \ldots \cap C_N) \end{aligned} \tag{19}$$

where N is the number of the prime implicants associated with the top event T_c.

Rules (5) and (7) can be applied to simplify the intersections of the prime implicants in the above formula. If any of the terms (say $C_1 \cap C_2 = I^k$) in the expression (19) is expressed in terms of the associated basic events $E_k1, E_k2, \ldots,$ and E_{km}, then

$$P(I_k) = P(E_{k1} \cap E_{k2} \cap \ldots \cap E_{km}) \tag{20}$$

where m is the number of the basic events associated with I_k.

Usually, the basic events $E_{k1}, E_{k2}, E_{k3}, \ldots,$ and E_{km} are assumed to be independent, that is, the occurrence of a given basic event is in no way affected by the occurrence of any other basic events. Thus,

$$P(I_k) = P(E_{k1})P(E_{k2}) \ldots P(E_{km}) \tag{21}$$

If each basic event E_{ki} $(i = 1, 2, \ldots, m)$ is assumed to follow an exponential distribution, then the probability of its occurrence at time t can be calculated by:

$$P(E_{ki}) = 1 - e^{-\lambda_{Eki}} \tag{22}$$

where λ_{Eki} is the failure rate of the basic event E_{ki}.

After $P(E_k1)$, $P(E_k2), \ldots,$ and $P(E_{km})$ have been obtained, $P(I_k)$ can be calculated. The probability of occurrence of the top event $P(T_c)$ can then be obtained using formula (19).

3.6 Software

A computer model has been developed with respect to the described method. The programme is written in *MODSIM II*^{TM} which is an object-oriented simulation language and which can also be used as a general purpose programming language.[2] The selection of this language is justified by the possible future implementation of event-based simulation to predict and assess system performance.

4 AN EXAMPLE

The hydraulic hoisting transmission system of a marine crane is shown functionally in Fig. 3. This

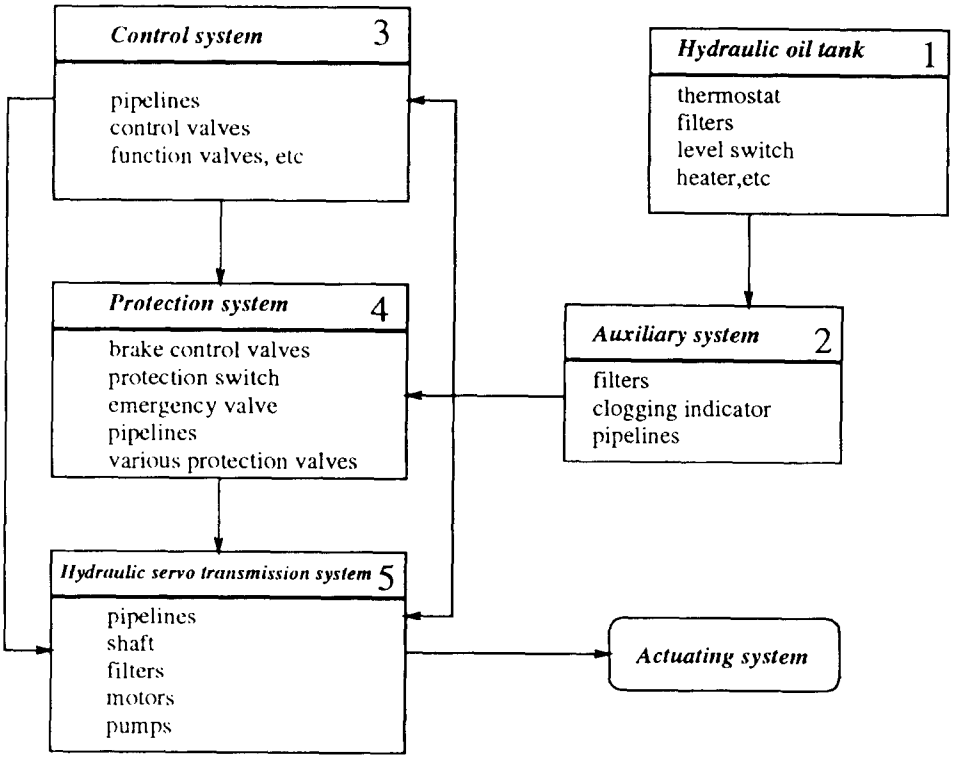

Fig. 3. The diagram of the hydraulic hoisting transmission system of a marine crane.

system is used to control the crane motions such as hoisting up or hoisting down loads as required by the operator.[9,17] It consists of five subsystems, namely a hydraulic oil tank, an auxiliary system, a control system, a protection system and a hydraulic servo transmission system. Each subsystem is associated with several failure modes. The occurrence of each failure mode associated with each subsystem may result in certain possible consequences.

4.1 Risk identification using FMECA

The results of the FMECA for the subsystems of this marine crane hoisting transmission system are shown in Tables 11–15.

For the convenience of constructing the Boolean representation tables of the subsystems, the following notation is given to the failure modes with severity classes 1, 2 and 3, and the output states of the subsystems.

4.1.1 Hydraulic oil tank

HM_1: major leak in the hydraulic oil tank
HM_2: level gauge failure
H_0: the output variable of oil supply tank
H_1: no oil supply from the oil tank
H_2: supplying oil from the oil tank

4.1.2 Auxiliary system

AM_1: failure allowing contaminant into system
AM_2: filter blocked
AM_3: blocking indicator fails to operate
AM_4: major leak
AM_5: no output from control pump
A_0: the output variable of the auxiliary system
A_1: no output
A_2: supplying contaminated hydraulic oil

4.1.3 Control system

CM_1: major leakage
CM_2: no output when required
CM_3: control output can not be closed for 'lowering motion'

Table 11. FMECA of the hydraulic tank

Name	Hydraulic oil tank				
Function	Supplying the oil for hydraulic control system, servo transmission system and protection system				
Failure rate	51 (failures per million hours)				
Failure mode number	Failure mode rate	Failure mode	Effects on system	Detecting method	Sev.
1	0.443	oil temperature too high or too low	reduce efficiency.	self-annunciating	4
2	0.103	level gauge failure	could result in insufficient oil supply.	self-annunciating & by maintenance	3
3	0.059	major leak	no flow for the system supply.	self-annunciating	3
4	0.395	minor leak	none.	self-annunciating	4

Table 12. FMECA of the auxiliary system

Name	Auxiliary system				
Function	Filtering, cooling and supplying the hydraulic oil				
Failure rate	106 (failures per million hours)				
Failure mode number	Failure mode rate	Failure mode	Effects on system	Detecting method	Sev.
1	0.284	failure allowing contaminant into system	pump servo may stick.	by maintenance	3
2	0.011	filter blocked	loss of servo pressure.	by maintenance	3
3	0.085	blocking indicator fails to operate	loss of servo pressure.	self-annunciating	3
4	0.566	minor leak	none.	self-annunciating & by maintenance	4
5	0.011	major leak	loss of servo pressure and motion.	self-annunciating	3
6	0.043	no output from control pump	no flow for system.	self-annunciating & by maintenance	2

Table 13. FMECA of the hydraulic servo transmission system

Name **Function** **Failure rate** Failure mode number	Failure mode rate	Hydraulic servo transmission system Producing hydraulic power for hoisting 265 (failures per million hours) Failure mode	Effects on system	Detecting method	Sev.
1	0.094	major leak	loss of hoisting pressure; in lowering motion, load could fall.	self-annunciating	1
2	0.522	minor leak	none	self-annunciating & by maintenance	4
3	0.013	shaft failure	loss of hoisting motion; no output.	self-annunciating & by maintenance	1
4	0.311	no output from the package motor	loss of hoisting pressure; no output.	self-annunciating & by maintenance	1
5	0.026	hydraulic short circuit	loss of hoisting pressure; in lowering motion, load could fall.	self-annunciating & by maintenance	1
6	0.026	motor seizure	load holds.	self-annunciating & by maintenance	3
7	0.008	pipe burst	major leak will happen; hoisting pressure will lose; in lowering motion, load could fall.	self-annunciating	1

Table 14. FMECA of the control system

Name **Function** **Failure rate** Failure mode number	Failure mode rate	Control system Controlling the servo hydraulic transmission system 36 (failures per million hours) Failure mode	Effects on system	Detecting method	Sev.
1	0.015	major leak	loss of hoisting pressure; in lowering motion, load could fall.	self-annunciating	2
2	0.310	minor leak	none.	self-annunciating	4
3	0.365	no output when required	loss of hoisting pressure; in lowering motion, load could fall.	by maintenance	3
4	0.155	control output for "lower" motion can not be closed when required	when de-energised by slack rope/lowering limit hoist, possibility of fall or damage of snagged load.	by maintenance	1
5	0.155	control output for "hoist up" motion can not be closed when required	jib and boom could be damaged.	by maintenance	1

CM_4: control output for 'hoisting up' motion can not be closed when required

C_0: the output variable of the control system

C_1: no output from the control system when required

C_2: control signal for 'hoisting up' can not be closed when required

C_3: control signal for 'lowering motion' can not be closed when required

4.1.4 Hydraulic servo transmission system

SM_1: major leak

SM_2: shaft failure

SM_3: no output from the package motor

Table 15. FMECA of the protection system

Name **Function** **Failure rate** Failure mode number	Failure mode rate	Protection system Protecting the various consequences caused by hazards 92 (failures per million hours) Failure mode	Effects on system	Detecting method	Sev.
1	0.132	failure of switch when energised	lost hoist motion.	self-annunciating & by maintenance	3
2	0.066	failure of return for hoisting up when de-energised	possibility of damage of jib.	by maintenance	1
3	0.530	minor leak	possibility of fall of snagged load.	self-annunciating	4
4	0.046	major leak	when brakes are applied, pump goes to zero stroke; "emergency release" and "wave following" disable.	self-annunciating	1
5	0.066	failure of emergency stop	load could be hoisted up or lowered down not as required even in emergency situation.	by maintenance	1
6	0.066	failure of hoisting up limit	when de-energised, pump remains at stroke and motor runs. otherwise no effect.	by maintenance	1
7	0.066	failure of hoisting down limit/slack rope prevention.	when de-energised by limit hoist, pump is not returned to zero stroke.	by maintenance	1
8	0.028	low boost pressure switch fails to open	hoisting pump is allowed to continue running at low pressures with a risk of cavitation damage.	by maintenance	1

* Sev.: Severity Class

SM_4: hydraulic short circuit
SM_5: motor seizure
SM_6: pipe burst
S: the output variable of the hydraulic servo transmission system
S_1: hoisting up continuously not as required
S_2: lowering continuously not as required
S_3: no output from the package output motor

4.1.5 Protection system
PM_1: failure of switch when energised
PM_2: failure to return for hoisting up when de-energised
PM_3: major leak
PM_4: failure of emergency stop
PM_5: failure of hoist up limit
PM_6: failure of hoist lower limit/slack rope prevention
PM_7: low boost pressure switch fails to open
P_0: the output variable of the protection system
P_1: no protection for emergency stop
P_2: no protection for 'hoist up' limit
P_3: no protection for 'hoist lower' limit/slack rope
P_4: no low boost pressure protection

4.2 Construction of the Boolean representation tables and assessment of the probability of occurrence of each system top event

The information produced from the FMECA of a subsystem can be utilised to construct the subsystem Boolean representation table by studying each possible combination of input attributes (i.e. the possible failure modes with severity class 1, 2 and 3). The Boolean representation tables of the five subsystems are constructed as shown in Tables 16–20, respectively. In the constructed Boolean representation tables, N stands for '*Not happening*' of a variable state and F stands for '*Failure happening*'.

The failure events of the hydraulic hoisting transmission system are the same as those of the hydraulic servo transmission system. Therefore, the construction of the system Boolean representation table starts from the hydraulic servo transmission

Table 16. Hydraulic oil tank

HM_1	HM_2	H_0
F	F	H_1
N	*	H_2
*	N	H_2

Table 17. Auxiliary system

AM_1	AM_2	AM_3	AM_4	AM_5	H_0	A_0
*	F	F	*	*	*	A_1
*	*	*	F	*	H_1	A_1
*	*	*	*	F	H_1	A_2
F	N	*	N	N	H_2	A_2
F	*	N	N	N	H_2	A_2

Table 18. Control system

CM_1	CM_2	CM_3	CM_4	A_0	C_0
*	F	*	*	*	C_1
*	*	*	*	A_1	C_1
F	*	*	*	A_2	C_1
*	*	F	*	*	C_2
*	*	*	F	*	C_3

system. The CRM is constructed as follows to describe the subsystem relationships.

$$CRM = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

The final Boolean representation table of the hydraulic hoisting transmission system is obtained as shown in Table 21 using the developed software.

The failure probabilities for S_1, S_2 and S_3 at time $t = 10000$ hours can be calculated using formula (19).

$$P(S_1) = 0.101 \quad P(S_2) = 0.015 \quad P(S_3) = 0.039$$

The consequences resulting from the occurrence of S_1, S_2 and S_3 can be described as follows:

S_1: Possibility of damage to the boom, ranging from minor distortion to total collapse (buckling). Possible rupture of the hoisting rope resulting in a dropped load. A dropped load may result in a total destruction of the lifted load, damage to the surrounding structure and other goods within the operating radius and possible death or severe injury to personnel.

S_2: A dropped load resulting in the probable consequences described in S_1

Table 19. Protection system

A_0	C_0	PM_1	PM_2	PM_3	PM_4	PM_5	PM_6	PM_7	P_0
*	*	*	*	*	F	*	*	*	P_1
A_1	*	*	*	F	*	*	*	*	P_1
*	C_2	*	*	*	*	*	*	*	P_2
A_1	*	*	F	*	*	F	*	*	P_2
A_1	C_3	*	*	*	*	*	*	*	P_3
A_2	*	F	*	*	*	*	F	*	P_3
A_1	*	*	*	*	*	*	*	F	P_4

Table 20. Hydraulic servo transmission system

A_0	C_0	P_0	SM_1	SM_2	SM_3	SM_4	SM_5	SM_6	S_0
*	C_2	P_2	*	F	*	*	*	*	S_1
*	C_1	P_2	*	*	*	*	*	*	S_1
A_2	*	P_2	*	*	*	*	F	*	S_1
*	C_3	P_3	*	*	*	*	*	*	S_2
*	C_1	P_3	*	*	*	*	*	*	S_2
A_1	*	P_3	*	*	*	*	*	*	S_2
*	*	P_3	F	*	*	*	*	*	S_2
*	*	P_3	*	*	*	*	F	*	S_2
*	*	P_3	*	F	*	*	*	*	S_2
*	*	P_3	*	*	F	*	*	*	S_2
*	*	P_3	*	*	*	F	*	*	S_2
*	*	P_3	*	*	*	*	*	F	S_2
*	*	P_1	*	F	*	*	*	*	S_2
*	*	P_1	*	*	*	F	*	*	S_2
*	*	P_1	*	*	*	*	*	F	S_2
A_1	*	*	F	*	*	*	*	*	S_3
A_1	*	*	*	*	F	*	*	*	S_3
A_1	*	*	*	*	*	F	*	*	S_3
A_1	*	*	*	*	*	*	*	F	S_3
A_2	*	*	*	*	*	*	F	*	S_3
A_2	*	P_4	*	*	*	*	*	*	S_3

S_3: A dropped load resulting in the probable consequences described in S_1.

The safety information produced above can be used by the designer to determine whether design actions are required to eliminate or control serious system failure events, and to prepare maintenance policies.

5 DISCUSSIONS AND APPLICATIONS

Compared to the fault tree method, BRM has the following advantages:

1. It can be used to easily analyze engineering systems with multiple state variables and feedback loops.
2. The system top events of a large engineering system with a relatively higher level of innovation can be completely identified.
3. Omissions of failure causes associated with the system top events are less likely than in fault tree analysis.

Table 21. The final system Boolean representation table

HM	*AM*	*CM*	*PM*	*SM*	
1 2	1 2 3 4 5	1 2 3 4	1 2 3 4 5 6 7	1 2 3 4 5 6	*S*
* *	* * * * *	* * *F* *	* * * * * * *	* *F* * * *	S_1
* *	* F F * *	* * * *	* F * * F * *	* * * * * *	S_1
F F	* * * F *	* * * *	* F * * F * *	* * * * * *	S_1
* *	* F F * *	* * F *	* * * * * * *	F * * * * *	S_1
F F	* * * F *	* * F *	* * * * * * *	F * * * * *	S_1
F F	* * * * F	* * F *	* * * * * * *	* * * * F *	S_1
N *	F N * N N	* * F *	* * * * * * *	* * * * F *	S_1
* N	F N * N N	* * F *	* * * * * * *	* * * * F *	S_1
N *	F * N N N	* * F *	* * * * * * *	* * * * F *	S_1
* N	F * N N N	* * F *	* * * * * * *	* * * * F *	S_1
* *	* F F * *	* * * F	* * * * * * *	* * * * * *	S_2
F F	* * * F *	* * * F	* * * * * * *	* * * * * *	S_2
F F	* * * * F	* * * F	F * * * * F *	* * * * * *	S_2
N *	F N * N N	* * * F	F * * * * F *	* * * * * *	S_2
* N	F N * N N	* * * F	F * * * * F *	* * * * * *	S_2
N *	F * N N N	* * * F	F * * * * F *	* * * * * *	S_2
* N	F * N N N	* * * F	F * * * * F *	* * * * * *	S_2
* *	* * * * *	* F *	F F * * * * *	* F * * * *	S_2
F F	* * * * F	F * *	F F * * * * *	* F * * * *	S_2
* N	F N * N N	F * *	F F * * * * *	* F * * * *	S_2
N *	F * N N N	F * *	F F * * * * *	* F * * * *	S_2
* N	F * N N N	F * *	F F * * * * *	* F * * * *	S_2
* *	* * * * *	* * * *	F * * * * * *	F * * * * *	S_2
F F	* * * * F	* * * *	F * * * * * *	F * * * * *	S_2
* N	F N * N N	* * * *	F * * * * * *	F * * * * *	S_2
N *	F * N N N	* * * *	F * * * * * *	F * * * * *	S_2
* N	F * N N N	* * * *	F * * * * * *	F * * * * *	S_2
* *	* * * *	* * * *	F * * * * * *	* * F * * *	S_2
F F	* * * * F	* * * *	F * * * * * *	* * F * * *	S_2
* N	F N * N N	* * * *	F * * * * * *	* * F * * *	S_2
N *	F * N N N	* * * *	F * * * * * *	* * F * * *	S_2
* N	F * N N N	* * * *	F * * * * * *	* * F * * *	S_2
* *	* * * * *	* * * *	F * * * * * *	* * * F * *	S_2
F F	* * * * F	* * * *	F * * * * * *	* * * F * *	S_2
* N	F N * N N	* * * *	F * * * * * *	* * * F * *	S_2
N *	F * N N N	* * * *	F * * * * * *	* * * F * *	S_2
* N	F * N N N	* * * *	F * * * * * *	* * * F * *	S_2
* *	* * * * *	* * * *	F * * * * * *	* * * * F *	S_2
F F	* * * * F	* * * *	F * * * * * *	* * * * F *	S_2
* N	F N * N N	* * * *	F * * * * * *	* * * * F *	S_2
N *	F * N N N	* * * *	F * * * * * *	* * * * F *	S_2
* N	F * N N N	* * * *	F * * * * * *	* * * * F *	S_2
* *	* * * * *	* * * *	F * * * * * *	* * * * * F	S_2
F F	* * * * F	* * * *	F * * * * * *	* * * * * F	S_2
* N	F N * N N	* * * *	F * * * * * *	* * * * * F	S_2
N *	F * N N N	* * * *	F * * * * * *	* * * * * F	S_2
* N	F * N N N	* * * *	F * * * * * *	* * * * * F	S_2
* *	* * * * *	* * * F	* * * * * * *	F * * * * *	S_2
* *	* F F * *	* * F *	* * * * * * *	F * * * * *	S_2
F F	* * * F *	* * F *	* * * * * * *	F * * * * *	S_2
* *	* F F * *	* * F *	* * * * * * *	* * F * * *	S_2
F F	* * * F *	* * F *	* * * * * * *	* * F * * *	S_2
* *	* F F * *	* * F *	* * * * * * *	* * * * * F	S_2
F F	* * * F *	* * F *	* * * * * * *	* * * * * F	S_2
* *	* F F * *	* * * *	* * * * * * *	F * * * * *	S_3
F F	* * * F *	* * * *	* * * * * * *	F * * * * *	S_3
* *	* F F * *	* * * *	* * * * * * *	* * F * * *	S_3
F F	* * * F *	* * * *	* * * * * * *	* * F * * *	S_3
* *	* F F * *	* * * *	* * * * * * *	* * * * * F	S_3
F F	* * * F *	* * * *	* * * * * * *	* * * * * F	S_3
* *	* F F * *	* * * *	* * * * * * *	* * * F * *	S_3
F F	* * * F *	* * * *	* * * * * * *	* * * F * *	S_3

Table 21. (*Continued*)

HM	*AM*	*CM*	*PM*	*SM*	
F F	* * * * F	* * * *	* * * * * * *	* * * * F *	S_3
N *	F N * N N	* * * *	* * * * * * *	* * * * F *	S_3
* N	F N * N N	* * * *	* * * * * * *	* * * * F *	S_3
N *	F * N N N	* * * *	* * * * * * *	* * * * F *	S_3
* N	F * N N N	* * * *	* * * * * * *	* * * * F *	S_3
F F	* * * * *	* * * *	* * * * * * F	* * * * * *	S_3

4. The information produced from FMECA can be used directly for Boolean representation modelling.

In addition, the BRM can also be used together with other formal safety analysis techniques such as fault tree analysis, qualitative reasoning analysis and the Monte Carlo simulation. The use of the BRM is greatly extended by such combinations. These combinations are briefly discussed as follows:

1. The inductive BRM can be combined with the inductive qualitative reasoning approach to form a combined modelling methodology in which qualitative reasoning is applied at the component level and the BRM is used at the system level.[20] Such a combined modelling methodology allows a bottom-up approach to be taken even in those cases where it is difficult to construct the Boolean representation tables for some components of a system. The qualitative descriptions of such components can form the basis for generating the respective input-output relations.
2. The inductive BRM can be used together with the fault tree method to form a mixed bottom-up and top-down safety analysis framework. Such a mixed framework would involve partial top-down fault tree analysis to focus upon areas of interest and partial bottom-up Boolean representation analysis to explore specific areas at a greater level of detail. Such a mixed methodology may be useful as a balance between the two techniques in order to exploit the advantages of each.
3. The BRM can also be used together with the Monte Carlo simulation techniques. The probabilities of occurrence of the system top events and the associated prime implicants can be simulated on the basis of the obtained Boolean representation table.[12,16,17] Different distribution types of basic event failures, cover (dormant) and revealed failures as well as maintenance activities can be dealt with.[17]

6 CONCLUDING REMARKS

A generalised Boolean representation modelling methodology is developed in this paper. In the

methodology, the information produced from FMECA is directly and efficiently used to carry out Boolean representation modelling to obtain the final system Boolean representation table in which all the possible system failure events and associated causes are contained. Both qualitative and quantitative safety analysis can then be carried out to assess the probabilities of occurrence of each system top event and associated prime implicants. Systems with multiple failure state variables and feedback loops can be easily analyzed using this methodology. The Boolean representation modelling approach can also be combined with other formal safety analysis methods to extend its use in safety analysis.

ACKNOWLEDGEMENT

This work forms part of a project on design for safety supported by the UK Science and Engineering Research Council under Grant No. GR/F 95306. In addition, the authors would like to express their gratitude to Dr P. Sen for useful discussions.

REFERENCES

1. Apostolakis, G. E., Salem, S. L. & Wu, J. S., *CAT: A computer code for automated construction of fault trees*, EPRI Report, March 1978.
2. CACI Products Company, *MODSIM II™: The language for Object-Oriented Programming (OOP) and SIMGRAPHICS II™*, Reference Manual, La Jolla, USA, May 1991.
3. Dixon, P., Decision tables and their applications. *Computer and Automation*, **13** (1964) 376–386.
4. Fussel, J. B., A formal methodology for fault tree construction. *Nucl. Sci. Engng*, **2** (1973) 421–432.
5. Fussel, J. B., Synthetic tree model-formal methodology for fault tree construction. *ANCR*-1098, Spring Field, VA 11151, March 1973.
6. Henley, E. J., Kumamoto H., *Probabilistic risk assessment*, IEEE Press, New York, 1992.
7. Kumamoto, H. & Henley, E. J., Safety and reliability synthesis of systems with control loops. *AIChE J.*, **25** (1979) 108–113.
8. MIL-STD-1629A, *Procedures for performing a failure mode, effects and criticality analysis*, Military Standard, Naval Ship Engineering Center, Washington D.C.
9. NEL, FMECA of NEI pedestal crane. *Report No. NECL*/01, May 1987.
10. Pollack, S. L., *Decision tables: theory and practice*, Wiley-Interscience, New York, 1971.
11. Powers, G. J., Tompkins, F. C., Fault tree analysis for chemical processes. *AIChE J.*, **20** (1974) 376–386.
12. Ruxton, T. & Wang J., Advances in marine safety technology applied to marine engineering systems. In *Proc. First Joint Conference on Marine Safety and Environment*, Delft, The Netherlands, 1–5 June 1992, 421–432.
13. Salem, S. L., A new methodology for the computer-aided construction of fault tree. *Ann. Nucl. Energy*, **4** (1977) 417–433.
14. Salem, S. L., Decision table development and application to the construction of fault trees. *Nucl. Tech.*, **42** (1979) 51–64.
15. Sen, P., Labrie, C. R., Wang, J., Ruxton, T., A general design for safety framework for large made-to-order engineering products. In *Proc. First Newcastle International Conference on Quality and Its Applications*, Newcastle, 1–3 September 1993, 499–505.
16. Wang, J. & Ruxton, T., *Design for safety of Made-To-Order (MTO) products*, ASME Publication, 93-DE-1, 1993 National Engineering Design Conference, Chicago, March 1993, 1–12.
17. Wang, J., Labrie, C. R. & Ruxton, T., Computer simulation techniques applied to the prediction and control of safety in maritime engineering. *Institute of Marine Engineers, Transactions (C)*, **105** Marine Management (Holdings) Ltd, 1993, 21–34.
18. Wang, J., System modelling for safety and reliability analysis. *EDCN/SAFE/RESC*/5/1, 1991.
19. Wang, J., Design for safety of marine engineering systems with multiple state variables. *Research Report EDCN/SAFE/RESC*/10/1, *EDC*, 1991.
20. Wang, J., Sen, P., Thompson, R. V., A mixed modelling approach for safety analysis. In *SRA-Europe*; *4th Conference on European Technology and Experience in Safety Analysis and Risk Management*, 18–20 Oct 1993, Rome, Italy, pp.1–7.
21. Wang, J., Ruxton, T. & Thompson, R. V., Failure analysis of Made-To-Order (MTO) products. ASME Publication, 93-*WA/DE*-8, Presented at the Winter Annual Meeting on Failure Analysis/Failure Prevention, New Orleans, Louisiana, 29 November–3 December, 1993, 1–10.
22. Waters, A. & Ponton, J. W., Qualitative simulation and fault propagation in process plants. *Chem. Eng. Res. Des.*, **67** (1989) 407–422.