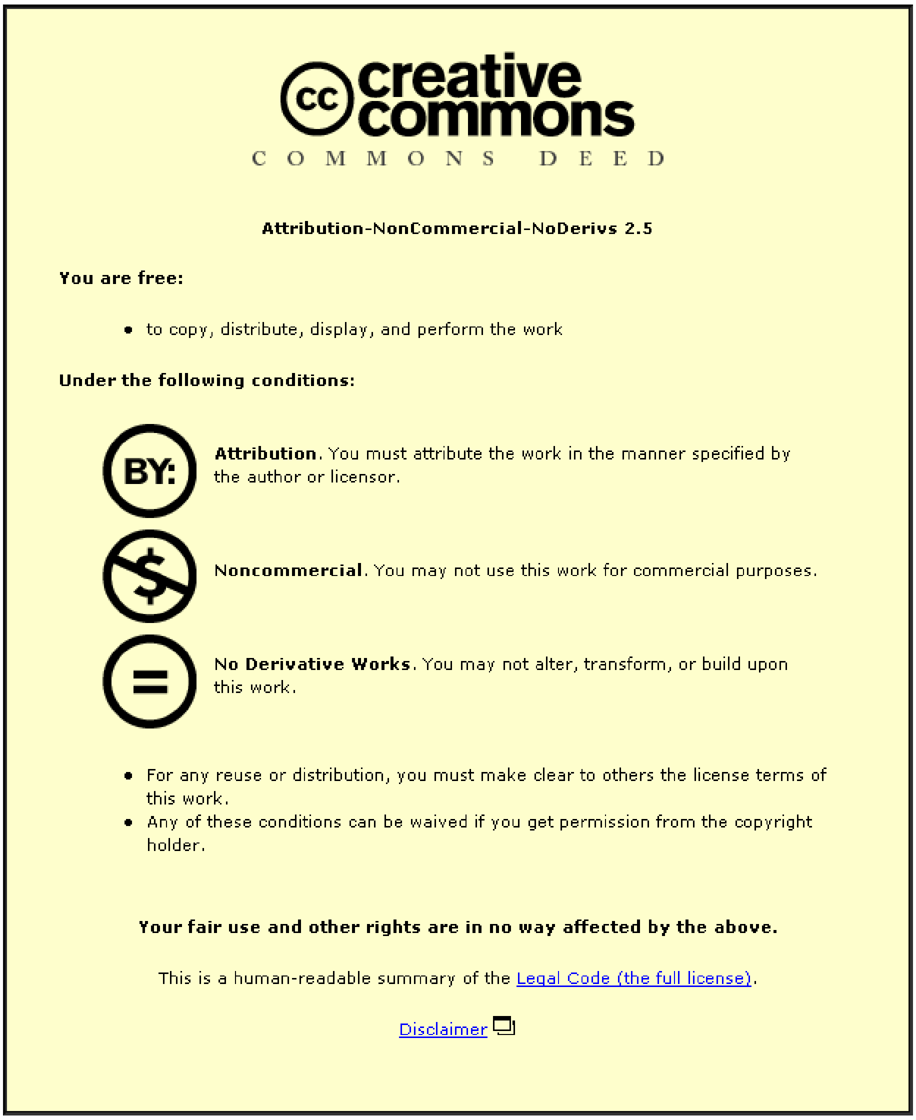


Event-Tree Analysis Using Binary Decision Diagrams

John D. Andrews and Sarah J. Dunnett

***Abstract*—This paper is concerned with ETA (event-tree analysis) where the branch point event causes are defined using fault trees. Attention is on the nontrivial situation where there are dependencies amongst the branch point events. The dependencies are due to component-failures in more than one of the fault trees. In these situations the analysis methods based on traditional FTA (fault-tree analysis) are inaccurate & inefficient.**

The inaccuracies are not consistent across the outcome events. If frequency predictions calculated in this way are then used in a risk assessment then the relative risks would be distorted and could lead to resources being used inappropriately to reduce the overall risk. A new approach using BDD (binary decision diagram) is described which addresses these deficiencies.

***Index Terms*—Binary decision diagrams, event-tree analysis, fault-tree analysis, noncoherent systems.**

I. Introduction

Acronyms[1]

F–N	plot of cumulative frequency (F) *vs* number of fatalities (N)
BDD	binary decision diagram
DBDD	dual BDD
BDD$_j$	BDD for S^j
ETA	event-tree analysis
FTA	fault-tree analysis
MCS	minimal cut set
PI	prime implicant

Definitions

Initiator: the potential hazardous trigger event.

Minimal Cut Set: a combination of component failure events which are necessary & sufficient to cause the top event.

Prime Implicant: a combination of basic events (success or failure) which is both necessary & sufficient to cause the top event.

Path Set: a list of working components which, if they occur at the same time, result in the system working. ◀

ETA is commonly used to identify the consequences that can result following the occurrence of a potentially hazardous event. It was first applied in risk assessments [1] for the nuclear industry but is now used by other industries such as chemical processing, offshore oil & gas production, and transportation. Quantification of the event-tree diagram allows the frequency of each of the outcomes to be predicted. In a risk study, the outcome event consequences, usually expressed in terms of fatalities, can be combined with the frequency of occurrence to produce an F–N curve to help assess the acceptability of the response to hazards. Event trees are an inductive (forward logic) technique which examine all possible responses to the initiating event, progressing left to right across the page. The branch points on the tree structure usually represent the success, failure, or partial failure of systems & subsystems which can respond to the initiating event.

FTA is a deductive analysis that can be used in conjunction with the event tree to identify the causes of the subsystem failures or branch events. Quantification of the fault tree provides the probability of passing along each of the event-tree branches.

The methodology used to quantify event trees has changed very little since the conception of the technique back in the 1960s when it was successfully used in the WASH 1400 study [2]. When the branch point events are s-independent of each other, quantification of the diagram is trivial and is achieved simply by finding the product of the frequency of the initiating event with the probabilities of passing along each branch leading to each outcome scenario. When there are s-dependencies between the branch events then quantifying the probability of passing along different branch points is more complex. It is performed by quantifying a fault tree whose top event is defined as combinations of occurrence and nonoccurrence of the branch-point events that have in turn been developed with fault-tree structures. Therefore there is a very heavy dependence on the efficiency & accuracy of the FTA. FTA is frequently used for safety-system assessments, and the majority of computer codes used for these analyzes are based on the kinetic tree theory [3] formulated in 1970. This theory uses approximations. Recent work at Loughborough University [4]–[6], Bordeaux University [7]–[9] ,and University of Virginia [10], [11] has produced a new assessment technique based on a BDD formulation of the system failure logic. This approach has advantages in terms of both efficiency & accuracy over the conventional kinetic tree theory. Since top-event probabilities can be derived exactly and without the need to evaluate the MCS or PI as intermediate results, this has major implications for improving the accuracy & efficiency of ETA.

This paper demonstrates the traditional ETA and compares this to a BDD based approach. The inadequacies of the ETA are demonstrated for a very simple system. These inadequacies can be overcome with the BDD approach. However, the nature of the worst-case complexity does not change.

A. Notation

m	number of subsystems
C_i	top-event cause $i, i = 1, \cdots, n$

Manuscript TR1999-068 received June 5, 1999 revised December 3, 1999.

The authors are with Department of Mathematical Sciences, Loughborough University, Loughborough LE11 3TU Great Britian (e-mail: {J.D.Andrews;S.J.Dunnett}@lboro.ac.uk).

Responsible editor: J. B. Dugan.

Publisher Item Identifier S 0018-9529(00)10122-8.

[1] The singular & plural of an acronym are always spelled the same.

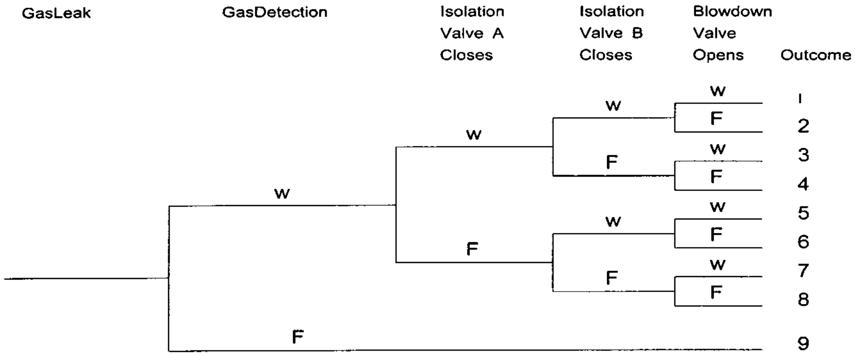


Fig. 1. Gas-leak event-tree.

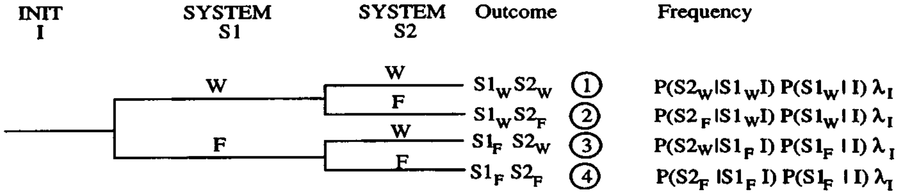


Fig. 2. Simple event-tree structure.

P_f	Pr{system failure}
λ_I	frequency of the initiating event
λ_i	frequency of each event-tree outcome i
T_j	event-tree outcome j; Boolean expression
T_{S_i}	top-event Boolean function for S_i
$\overline{T}_{S_i}$	not T_{S_i}
$+$	OR (for events)
$\cdot$	AND (for events)
$\phi(\mathbf{x})$	structure function
O_i	outcome events, $i = 1, \cdots, n$
c_i	consequence associated with O_i
S_j	subsystem j, $j = 1, \cdots, m$
Q_j	Pr{failure of subsystem j}
f_i	frequency of outcome i
R_i	risk associated with O_i

II. Independent Event Trees

Fig. 1 shows a very simple event-tree for an example safety-system. The initiating event is the release of gas on an offshore platform. The branch points then consider the success (W) and failure (F) of the gas detection system, isolation valve subsystems A and B, and the blowdown valve subsystem in turn. The outcomes determined by the end point of each event-tree branch identifies a different consequence following the initiating event. Fault trees can be constructed to develop the causes of each of these subsystem failures. If the systems fail s-independently then the event tree quantification is the relatively simple task of multiplying the "probabilities of passing along each branch point on any path through the diagram" by the "initiating-event frequency." The P_f can be evaluated by quantifying the relevant fault tree; $1 - P_f$ then gives the probability of passing along the system-success branch.

Strong s-dependencies when

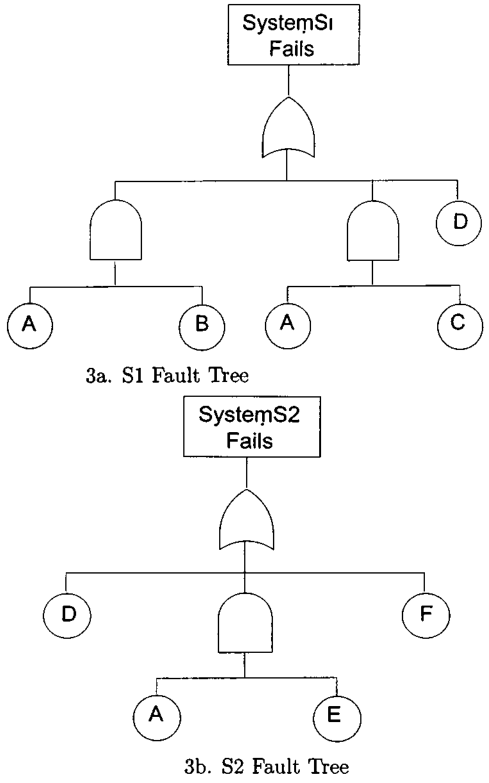
$$\Pr\{A|B\} = 1 \quad \text{or } \Pr\{A|B\} = 0$$

for system A event following the system B event can be incorporated in this approach. As shown in Fig. 1, if the gas-detection system fails then none of the other systems are activated; thus their availability is irrelevant because the consequence in this sequence is already determined. This is represented by the line from the branch representing gas-detection system-failure going completely across the diagram, indicating that this event alone determines the outcome.

III. s-Dependencies in Event Trees

The procedure to analyze event trees when there are weak s-dependencies uses the FTA much more heavily. Weak s-dependencies occur when basic events representing component failures appear in more than one of the fault trees which develop the branch-point causes. In these circumstances the fault trees representing the relevant system working & failed states need to be combined as inputs to an AND gate whose output now determines the causes of a higher level complex event. Boolean reduction of the combined fault-tree structure produces the combinations of basic events which cause the complex-system event considering the s-dependency. Since some of the subsystem events in the complex event represent system-success, the resulting fault tree is noncoherent [12]. The analysis of noncoherent fault trees using the usual analysis of kinetic-tree theory relies heavily on approximation as detailed in Section VII, and is at times both inaccurate & inefficient. The BDD approach can offer advantages in both efficiency & accuracy which becomes especially important when analyzing very large event-trees such as those in the nuclear industry.

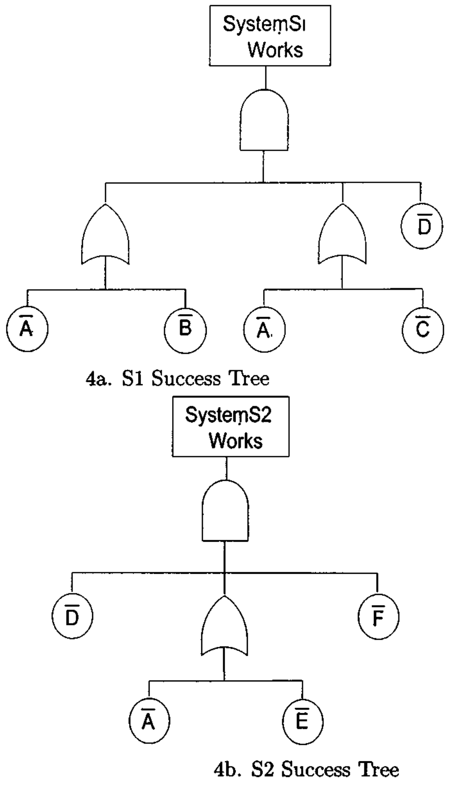
To demonstrate the advantages of BDD over the conventional FTA when used to assess an event tree, both techniques are discussed in relation to the simple event tree in Fig. 2. There are only 2 subsystems (S1 and S2) which respond to the initiating event (I).



3a. S1 Fault Tree

3b. S2 Fault Tree

Fig. 3. System fault trees.



4a. S1 Success Tree

4b. S2 Success Tree

Fig. 4. System success trees.

The fault trees representing the failure of S1 and S2 are in Fig. 3(a) and (b), respectively. Because the basic events A & D occur in both fault trees, there is a "weak" s-dependence between the subsystem failure events.

Due to the weak s-dependencies, the 4 outcomes which can occur in response to the initiating event are developed as fault trees with the top-event expressions in (1). The introduction of the NOT gates in 3 of these structures produces noncoherent trees. The efficiency & accuracy of the event-tree quantification then depends upon the efficiency & accuracy of the FTA for noncoherent fault trees.

$$\begin{aligned} T_1 &= \overline{T}_{S_1}\cdot\overline{T}_{S_2}, \quad T_2 = \overline{T}_{S_1}T_{S_2}, \\ T_3 &= T_{S_1}\overline{T}_{S_2}, \quad T_4 = T_{S_1}T_{S_2} \qquad (1) \end{aligned}$$

Fig. 4(a) and 4(b) show the dual formulations of the fault trees representing failure of S1 & S2 [Fig. 3(a) and (b)], respectively.

IV. Qualitative FTA

When NOT logic is introduced to a fault-tree structure, that structure no longer is nondecreasing; thus its structure function is noncoherent. Boolean reduction, of the logic function representing the top event of a coherent fault tree, to a sum-of-products or disjunctive normal form identifies the MCS. When the fault tree is noncoherent, then the equivalent logic expression for the top event produces the PI.

When the tree structure is coherent the Boolean reduction process results in MCS.

From the fault trees representing the causes of the two safety-system failures [Fig. 3(a) and (b)] in the simple event tree, the Boolean expressions (2) & (3) which provide the failure combinations derived for the top event.

$$\begin{aligned} T_{S_1} &= A\cdot B + A\cdot C + D \quad (\text{MCS: } AB,\ AC,\ D) \qquad (2) \\ T_{S_2} &= D + F + A\cdot E \quad (\text{MCS: } D,\ F,\ AE) \qquad (3) \end{aligned}$$

From the fault trees representing system success [Fig. 4(a) and 4(b)], the top event causes are derived from:

$$\begin{aligned} \overline{T}_{S_1} &= (\overline{A}+\overline{B})\cdot(\overline{A}+\overline{C})\cdot\overline{D} = \overline{A}\cdot\overline{D}+\overline{B}\cdot\overline{C}\cdot\overline{D} \\ &\quad (\text{PI: } \overline{AD},\ \overline{BCD}) \qquad (4) \\ \overline{T}_{S_2} &= \overline{D}\cdot\overline{F}\cdot(\overline{A}+\overline{E}) = \overline{A}\cdot\overline{D}\cdot\overline{F}+\overline{D}\cdot\overline{E}\cdot\overline{F} \\ &\quad (\text{PI: } \overline{ADF},\ \overline{DEF}) \qquad (5) \end{aligned}$$

V. Quantitative FTA

Once the C_i are determined, viz, MCS or PI, then $\Pr\{T\}$ can be found by evaluating the inclusion-exclusion expansion.

$$\begin{aligned} \Pr\{T\} &= \sum_{i=1}^{n}\Pr\{C_i\} - \sum_{all\,i}\sum_{all\,j}\Pr\{C_i\cap C_j\} \\ &\quad + \sum_{all\,i}\sum_{all\,j}\sum_{all\,k}\Pr\{C_i\cap C_j\cap C_k\} \\ &\quad + \cdots + (-1)^{n+1}\cdot\Pr\{C_1\cap C_2\cdots\cap C_n\} \qquad (6) \end{aligned}$$

When the C_i are MCS, this series expression (5) is frequently approximated by truncating the expansion after the first 1 or 2 terms or by using an alternate approximation such as the MCS upper bound:

$$\Pr\{T\} \leq 1 - \prod_{i=1}^{n}[1 - \Pr\{C_i\}] \tag{7}$$

For coherent fault trees the truncation of the expansion in (6) is justified because the terms which account for the simultaneous occurrence of higher-order failure combinations have a rapidly diminishing numerical contribution to the top-event probability. If the tree is noncoherent and if the C_i are PI, then the truncation of (6) or the use of approximation (7), might not be valid, and many terms in the series expansion might need to be calculated to gain the required accuracy. For large fault trees, it is beyond the capability of modern computers to evaluate the full expansion in any reasonable time. To quantify the top-event probability, the PI are frequently reduced to their coherent approximations by assuming any working states for the components in the expression are assumed to be TRUE. On the basis that

$$\Pr\{\text{component works}\} \approx 1,$$

the resulting approximations to the MCS are then minimized, and approximations such as (7) are used.

VI. Evaluating the Causes of the Event-Tree Outcomes

Identifying the causes of each event-tree outcome, where the fault trees have weak s-dependencies, is equivalent to producing the Boolean expressions in (1).

$$\begin{aligned} T_1 &= \overline{T}_{S_1} \cdot \overline{T}_{S_2} \\ &= \left(\overline{A} \cdot \overline{D} + \overline{B} \cdot \overline{C} \cdot \overline{D}\right) \cdot \left(\overline{D} \cdot \overline{F} \cdot \overline{A} + \overline{D} \cdot \overline{F} \cdot \overline{E}\right) \\ &= \overline{A} \cdot \overline{D} \cdot \overline{F} + \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot \overline{E} \cdot \overline{F}; \end{aligned} \tag{8}$$

$$\begin{aligned} T_2 &= \overline{T_{S_1}} \cdot T_{S_2} \\ &= \left(\overline{A} \cdot \overline{D} + \overline{B} \cdot \overline{C} \cdot \overline{D}\right) \cdot (D + F + A \cdot E) \\ &= \overline{A} \cdot \overline{D} \cdot F + \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot F + A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot E; \end{aligned} \tag{9}$$

$$\begin{aligned} T_3 &= T_{S_1} \cdot \overline{T}_{S_2} \\ &= (A \cdot B + A \cdot C + D) \cdot \left(\overline{D} \cdot \overline{F} \cdot \overline{A} + \overline{D} \cdot \overline{F} \cdot \overline{E}\right) \\ &= A \cdot B \cdot \overline{D} \cdot \overline{E} \cdot \overline{F} + A \cdot C \cdot \overline{D} \cdot \overline{E} \cdot \overline{F}; \end{aligned} \tag{10}$$

$$\begin{aligned} T_4 &= T_{S_1} \cdot T_{S_2} \\ &= (A \cdot B + A \cdot C + D) \cdot (D + F + A \cdot E) \\ &= A \cdot B \cdot F + A \cdot B \cdot E + A \cdot C \cdot F + A \cdot C \cdot E + D \end{aligned} \tag{11}$$

VII. Quantifying the Frequency of the Event-Tree Outcomes

The MCS or PI evaluated during the qualitative FTA are used along with the probability of each basic event to evaluate all λ_i. For exact results, (7) is used to determine the probability of the particular responses for S1 & S2. This probability is then multiplied by the "initiating event frequency" to determine the frequency of each event-tree outcome.

TABLE I
Comparison of Event-Tree Results

Event Tree Outcome	Exact Probability	Coherent Approx.	Error (%)
1	0.788049	1.0	26.9
2	0.094851	0.109	14.9
3	0.013851	0.019	37.2
4	0.103249	0.103249	0.0

A. Exact Calculation for Non-Coherent Outcomes

$$\begin{aligned} \lambda_1 &= \lambda_I \cdot \Pr\{T_1\} \\ &= \lambda_I \cdot \left[\Pr\left\{\overline{A} \cdot \overline{D} \cdot \overline{F}\right\} + \Pr\left\{\overline{B} \cdot \overline{C} \cdot \overline{D} \cdot \overline{E} \cdot \overline{F}\right\}\right. \\ &\qquad \left. - \Pr\left\{\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot \overline{E} \cdot \overline{F}\right\}\right] \end{aligned} \tag{12}$$

$$\begin{aligned} \lambda_2 &= \lambda_I \cdot \Pr\{T_2\} \\ &= \lambda_I \cdot \left[\left(\Pr\left\{\overline{A} \cdot \overline{D} \cdot F\right\} + \Pr\left\{\overline{B} \cdot \overline{C} \cdot \overline{D} \cdot F\right\}\right.\right. \\ &\qquad \left. + \Pr\left\{A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot E\right\}\right) \\ &\qquad - \left(\Pr\left\{\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot F\right\}\right. \\ &\qquad \left.\left. + \Pr\left\{A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} \cdot E \cdot F\right\}\right)\right] \end{aligned} \tag{13}$$

$$\begin{aligned} \lambda_3 &= \lambda_I \cdot \Pr\{T_3\} \\ &= \lambda_I \cdot \left[\Pr\left\{A \cdot B \cdot \overline{D} \cdot \overline{E} \cdot \overline{F}\right\} + \Pr\left\{A \cdot C \cdot \overline{D} \cdot \overline{E} \cdot \overline{F}\right\}\right. \\ &\qquad \left. - \Pr\left\{A \cdot B \cdot C \cdot \overline{D} \cdot \overline{E} \cdot \overline{F}\right\}\right] \end{aligned} \tag{14}$$

Because this is a very small, simple example, the exact calculations can be performed. When there are several thousand, perhaps hundreds of thousands, of the MCS or PI, then these calculations cannot be performed. For a coherent fault tree the inclusion-exclusion expression (6) converges, and truncation after the first or second term usually yields a result of acceptable accuracy. Alternatively the MCS upper-bound (7) is a better approximation (exact when the MCS are s-independent).

For noncoherent fault trees, the convergence of the inclusion-exclusion expansion can be very slow and many terms need to be evaluated. For large fault trees this is not possible. An alternative method (frequently used in commercial packages) is to use the coherent approximation. For the simple example in this section, the result is:

$$\lambda_1 = \lambda_I \cdot \Pr\{T_1\} = \lambda_I \tag{15}$$

$$\begin{aligned} \lambda_2 &= \lambda_I \cdot \Pr\{T_2\} \\ &= \lambda_I \cdot [\Pr\{F\} + \Pr\{A \cdot E\} - \Pr\{A \cdot E \cdot F\}] \end{aligned} \tag{16}$$

$$\begin{aligned} \lambda_3 &= \lambda_I \cdot \Pr\{T_3\} \\ &= \lambda_I \cdot [\Pr\{A \cdot B\} + \Pr\{A \cdot C\} - \Pr\{A \cdot B \cdot C\}] \end{aligned} \tag{17}$$

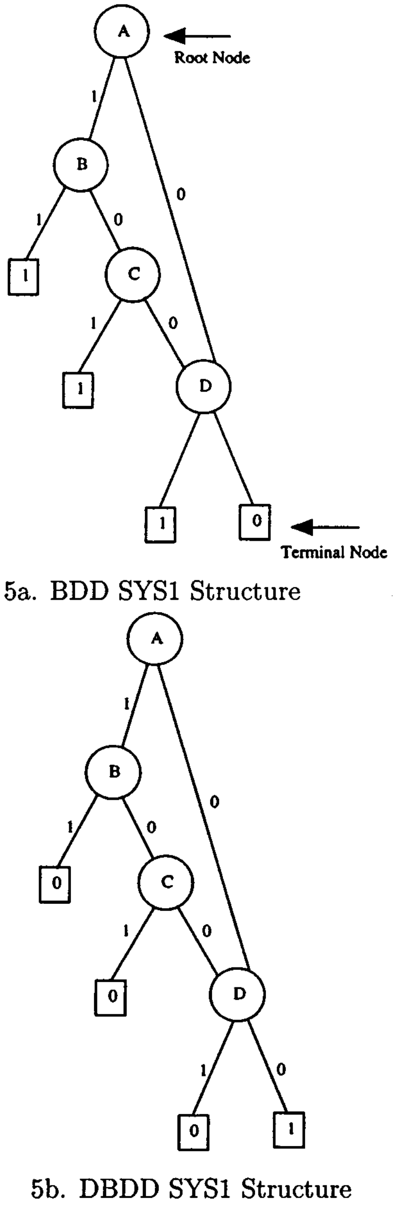
To provide a numerical comparison of results, and the effects of the approximations, let:

- each component-failure probability $= 0.1$
- $\lambda_I = 1.0$/year.

The results are summarized in Table I. Large percentage errors exist even in this relatively small problem.

VIII. Binary Decision Diagrams

BDD provide an alternative logic form to the fault-tree structure to express the system failure causes. BDD encodes



5a. BDD SYS1 Structure

5b. DBDD SYS1 Structure

Fig. 5. Structures for system-A fault tree.

a Shannon form of the structure function; thus the exact $\Pr\{\text{system failure}\}$ can be deduced without the need to resort to any approximations. BDD has the additional advantages that its quantification does not require the MCS & PI to be determined as an intermediate stage, and that it improves the accuracy & efficiency. However, nothing comes without cost and, for BDD, the cost is the effort expended converting from a fault-tree structure to the BDD. Previous work [8] investigating the relative efficiency of the conventional FTA and BDD approaches has shown that orders of magnitude reduction in computer processing time for large fault trees can be achieved. This improvement is anticipated to be even more important for noncoherent fault trees which tend to produce a vast number of system failure modes which include component success states (PI).

Fig. 5(a) shows the BDD SYS1 structure for the fault tree in Fig. 3(a). BDD construction requires the basic events to be ordered. For this SYS1 example, the order is $A < B < C < D$. Rules for the fault tree to BDD conversion process are covered extensively in [5], [8].

The diagram features a root vertex placed at the top of the tree structure. Each vertex (node) represents a basic event from the fault tree and has two paths which leave the node, a 1 branch and a 0 branch, which indicate the occurrence (failure) and nonoccurrence of the basic event respectively. Paths through the BDD terminate at one of two types of terminal node, labeled 1 and 0. Paths which lead to a terminal-1 node specify the conditions for the fault-tree top event to occur: $\phi(\mathbf{x}) = 1$. Listing just the failure events on such a path is equivalent to producing the cut sets for the fault tree. Unless the selected basic-event ordering has produced a minimal-form BDD, the cut sets will have to be processed to remove redundancies and produce the MCS. There is a procedure to transform the BDD to encode only MCS [8]. While this is an important source of information to the analyst it is not needed to evaluate the event-tree outcome frequencies. Conversely, paths terminating in a 0 terminal-node represent the top event nonoccurrence.

Tracing the paths through the BDD in Fig. 5 produces cut sets: $A \cdot B$, $A \cdot C$, $A \cdot D$, D.

Removing the redundant cut-set results in the extraction of the third combination from the list, leaving the 3 MCS obtained in (2).

Due to the binary branching, each path in the BDD is mutually exclusive; thus the $\Pr\{\text{system failure}\}$ is obtained by summing the $\Pr\{\text{each disjoint path leading to a terminal-1 node}\}$. $\Pr\{\text{each disjoint path}\}$ is the likelihood of the combination of the basic events (success & failure) represented by the path.

IX. Dual Formulation Using the BDD

A feature of BDD is the ease with which the dual can be formulated. The primary BDD represents $\phi(\mathbf{x})$. The dual function is:

$$\overline{\phi}_D(\mathbf{x}) = 1 - \phi(\mathbf{x}) \tag{18}$$

The dual BDD represents $\overline{T}_A$; it is created by changing the terminal 1's to terminal 0's and vice-versa. (In this formulation of the dual, the nodes on the BDD still represent component-failure states.) Applying these rules to the BDD illustrated in Fig. 5(a) gives the dual in Fig. 5(b).

The path through the dual BDD to a terminal 1 which includes each node passed through on the 0 branch (working components) represents the path sets of the fault tree.

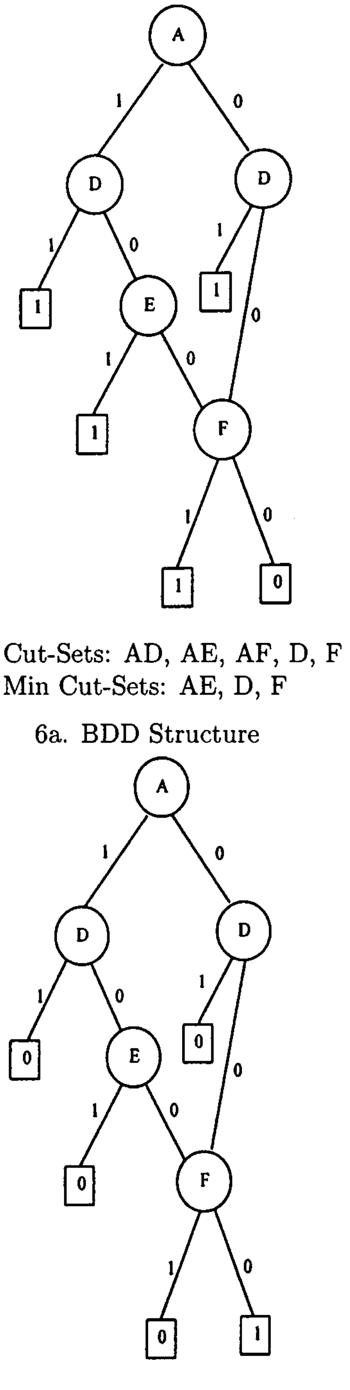
The path sets for the BDD shown in Fig. 5(b) are:

$$B \cdot C \cdot D, \quad A \cdot D.$$

These path sets are also minimal, and agree with those produced by the conventional analysis method (4). The transformation between primal and dual is very efficient.

X. Event-Tree Outcomes

Using BDD to analyze the outcomes of the event-tree in Fig. 2 requires the BDD and dual formulation for the fault trees in



Cut-Sets: AD, AE, AF, D, F
Min Cut-Sets: AE, D, F

6a. BDD Structure

Path-Sets: DEF, ADF
Min Path-Sets: DEF, ADF

6b. DBDD Structure

Fig. 6. Structures for system-B fault tree.

Fig. 3(a) and (b). Fig. 5(a) and (b) contain the BDD and DBDD for the fault tree in Fig. 3(a). For the fault tree in Fig. 3(b), the equivalent BDD is illustrated in Fig. 6(a) and its DBDD in Fig. 6(b). The variable ordering, $A < D < E < F < G$, has been assumed.

Using the rules to manipulate these structures, the BDD which represent the causes of each outcome for

$$\overline{T}_A \cdot \overline{T}_B, \quad \overline{T}_A \cdot T_B, \quad T_A \cdot \overline{T}_B, \quad T_A \cdot T_B$$

can be produced. Fig. 7 gives these along with the implicant paths derived from the BDD and their likelihood of occurrence; the implicant paths are shown with the probability in parentheses. The likelihoods derived from the BDD agree with those calculated using the exact method of a full expansion of the inclusion-exclusion series. Thus confirming the advantage of using the BDD.

XI. ETA Algorithm

The calculations in section 10 show that using a BDD structure to evaluate the likelihood of event-tree outcomes when weak s-dependencies exist is both accurate and efficient. It is only desirable to construct the combined BDD when such s-dependencies are encountered. For s-independent sections, only the product of the probabilities is required. The algorithm in this section quantifies a general event-tree structure with both weak & strong s-dependencies. The event-tree structure must be drawn to account for the strong s-dependencies. The event-tree has λ_I, resulting in O_i & c_i. Each path through the event-tree diagram leading to an O_i considers the functionality or failure of all S_j. Fault trees have been constructed to represent the cause of each subsystem failure. Component failures are repeated in k of the m fault trees, providing k weakly s-dependent and $m - k$ s-independent fault-tree structures.

```
Algorithm
  1. Scan each of the $S_j$ fault trees.
If the fault tree is $s$-independent of all
  other fault trees,
  Then enter its label $j$ in set $I$;
  Else place its label $j$ in set $W$.
  2. Convert all of the $S_j$ fault trees to
  their $\mathrm{BDD}_j$.
  3. For each of the $m - k$ entries in set
  $I$, use the relevant BDD to evaluate the
  $Q_j$, $j \in I$.
  4. For each of the $k$ entries in set $W$,
  formulate and store the $\mathrm{DBDD}_j$, $j \in W$.
  5. Set $Q_I = 1.0$
    a. Over each path leading to $O_i$, con-
  sider each branch point on the path from
  the initiating event.
If (the branch point label $j \in I$)
  Then
    If branch-point represents subsystem
  functionality
      Then $Q_I = Q_I \cdot (1 - Q_j)$,
      Else $Q_I = Q_I \cdot Q_j$,
      End_If
  Else
    If (branch point represents subsystem
  functionality)
      Then place $j$ in the dual set $D$
      Else place $j$ in the primal set $P$
      End_If
  End_If
```

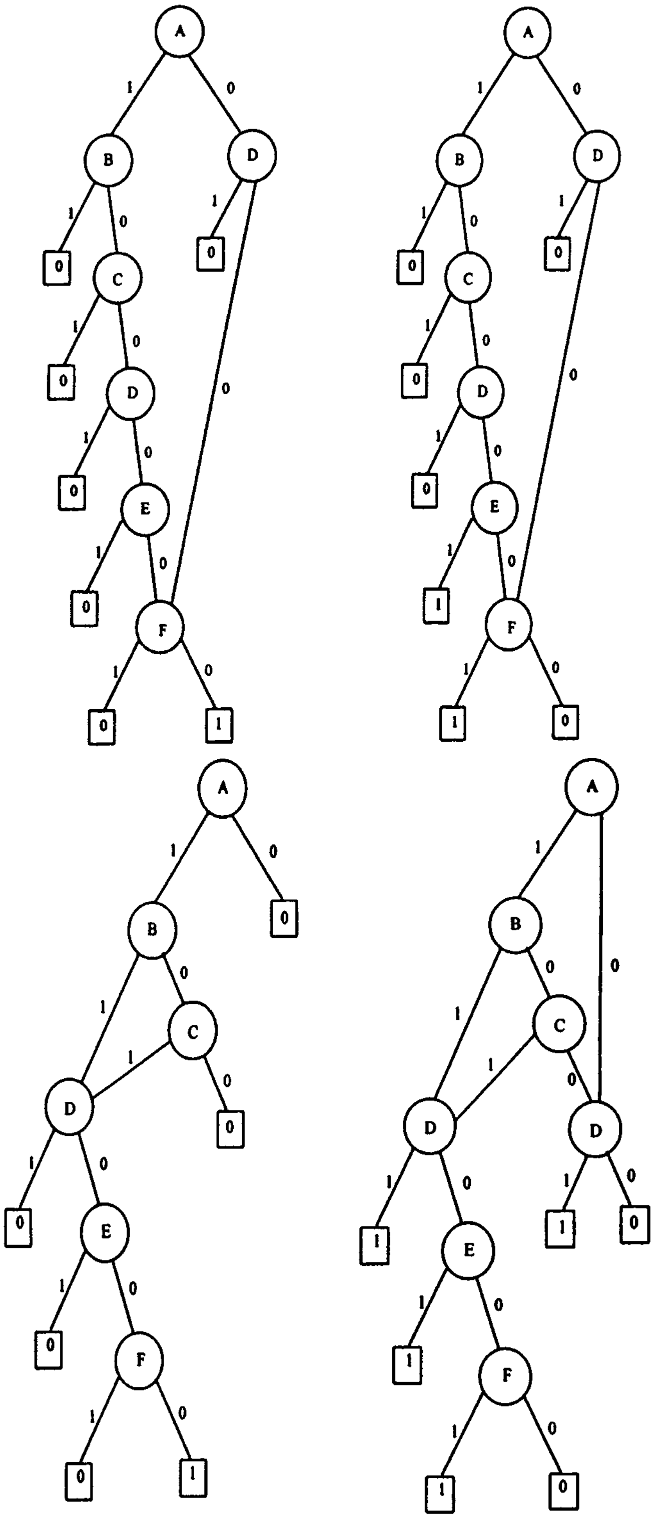


Fig. 7. BDD for various outcomes.

```
    b. Construct the overall $s$-dependency
  BDD for:
$T_i = \cup_{j\in P}\ \mathrm{BDD}_i \cup_{j\in D}\ \mathrm{DBDD}_j$
Calculate $Q_{T_i}$ using the newly formed BDD.
    c. Calculate $f_I = \lambda_I \cdot Q_I \cdot Q_{T_i}$
    d. Calculate $R_i = f_i \cdot c_i$
  End_Algorithm
```

XII. Example

Leak-Detection System On An Offshore Structure

As an example application of the analysis procedure in Section XI, consider the gas-leak system event-tree in Fig. 1. It is a simplified system taken from an offshore oil & gas production

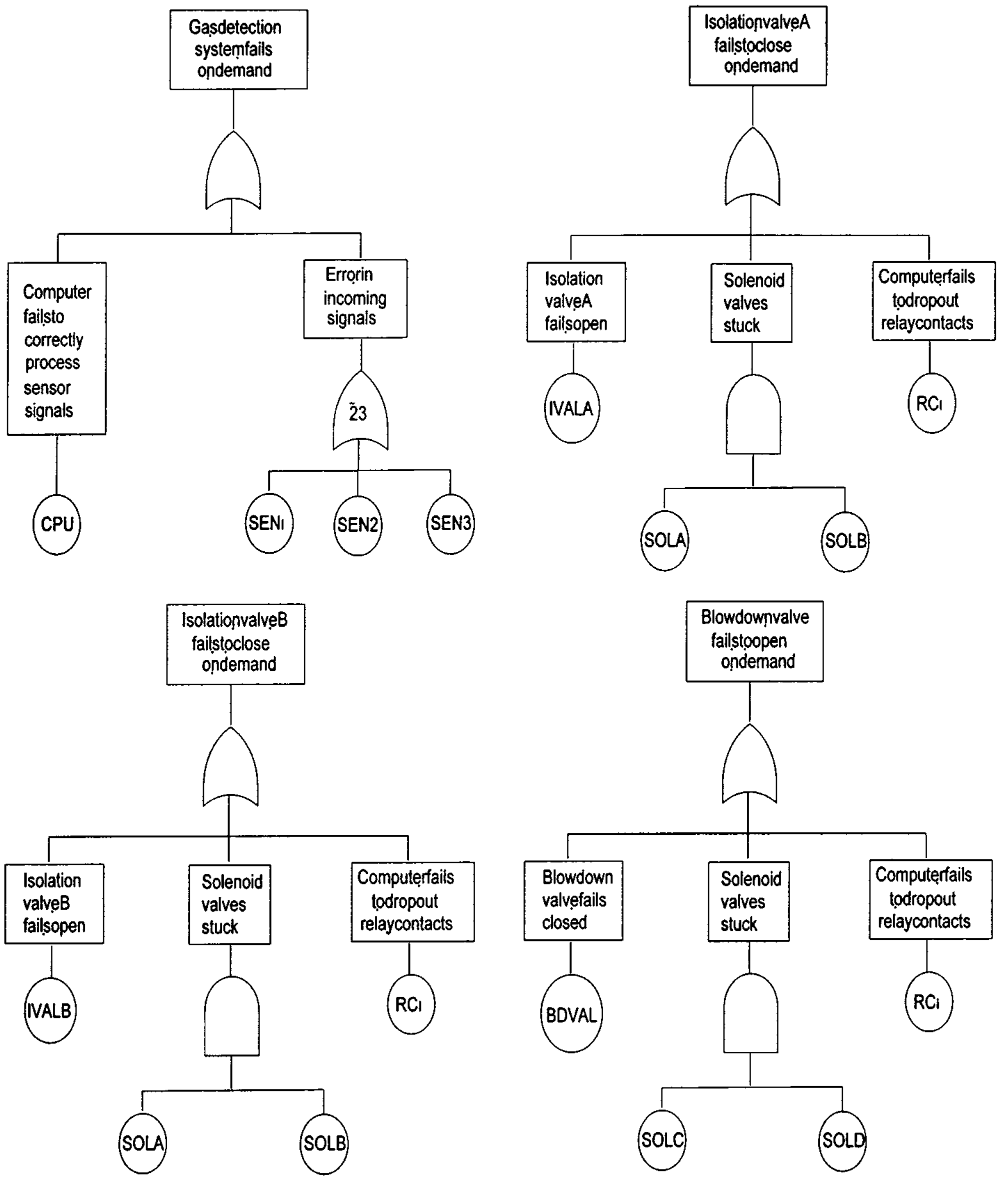

Gas detection system fails on demand
Computer fails to correctly process sensor signals
Error in incoming signals
23
CPU
SEN1
SEN2
SEN3
Isolation valve A fails to close on demand
Isolation valve A fails open
Solenoid valves stuck
Computer fails to drop out relay contacts
IVALA
SOLA
SOLB
RC1
Isolation valve B fails to close on demand
Isolation valve B fails open
Solenoid valves stuck
Computer fails to drop out relay contacts
IVALB
SOLA
SOLB
RC1
Blowdown valve fails to open on demand
Blowdown valve fails closed
Solenoid valves stuck
Computer fails to drop out relay contacts
BDVAL
SOLC
SOLD
RC1

Fig. 8. Gas-leak system fault-trees.

TABLE II
EVENT-TREE *s*-DEPENDENT-PATH PROBABILITIES

Order #	Outcome Event	Exact Prob.	MCS	Coherent Approximation Value	Error (%)
1	$\overline{TIA}$ $\overline{TIB}$ $\overline{TBD}$	$8.1044 \cdot 10^{-1}$	—	1.0000	23.4
2	$\overline{TIA}$ $\overline{TIB}$ TBD	$4.4792 \cdot 10^{-2}$	1. BDVAL 2. SOLC SOLD	$5.2375 \cdot 10^{-2}$	16.9
3	$\overline{TIA}$ TIB $\overline{TBD}$	$4.2655 \cdot 10^{-2}$	1. IVALB	$5.0000 \cdot 10^{-2}$	10.7
4	$\overline{TIA}$ TIB TBD	$2.3575 \cdot 10^{-3}$	1. IVALB BDVAL 2. IVALB SOLC SOLD	$2.6247 \cdot 10^{-3}$	11.3
5	TIA $\overline{TIB}$ $\overline{TBD}$	$4.2655 \cdot 10^{-2}$	1. IVALA	$5.0000 \cdot 10^{-2}$	10.7
6	TIA $\overline{TIB}$ TBD	$2.3575 \cdot 10^{-3}$	1. IVALA BDVAL 2. IVALA SOLC SOLD	$2.6247 \cdot 10^{-3}$	11.3
7	TIA TIB $\overline{TBD}$	$4.4956 \cdot 10^{-3}$	1. IVALA IVALB 2. SOLA SOLB	$4.9938 \cdot 10^{-3}$	11.1
8	TIA TIB TBD	$5.0248 \cdot 10^{-2}$	1. RC1 2. IVALA, IVALB, BDVAL 3. SOLA, SOLB, BDVAL 4. IVALA, IVALB, SOLC, SOLD 5. SOLA, SOLB, SOLC, SOLD	$5.0249 \cdot 10^{-2}$	0.02

platform. In the event of a loss of containment on the gas section, then the detection system should function to identify the event occurrence. The isolation & blowdown systems are then triggered, the isolation system closes valves A & B to ensure the potential escape inventory is limited. Blowdown valves then open to de-pressurize the sections safely by allowing the gas to be flared. Fig. 8 has simplified fault trees showing the causes of failure of the gas detection, isolation, and blowdown systems. The fault trees feature the important characteristics of a full study and can be used to demonstrate the validity of the method for a complete analysis. As indicated on the event tree, the isolation & blowdown system strongly s-depend on the gas-detection system. Also there is a weak s-dependency, failure of the relay contacts RC1, common to 3 of the fault trees; 2 other such s-dependencies representing solenoid failures, are common to the isolation-valve fault trees.

To obtain realistic orders of magnitude for the failure probabilities in the fault trees, each basic event is assigned an availability of 95%.

The Algorithm (in Section XI) places the detection system into set I and the remaining systems into set W. Complications arise when considering the outcome probabilities of the "routes through the event-tree which have s-dependencies" (outcomes 1–8 in Fig. 2). Table II, column 3, shows the results from the Algorithm for these event probabilities. Having considered the s-dependencies, the final frequency of occurrence for each outcome, 1–8, are obtained by multiplying the probabilities in Table II by $\Pr\{\text{detection system works}\}$, 5.569×10^{-2}, and the initiating event frequency. To compare with conventional techniques, an analysis using coherent approximations has been performed; the results are in Table II, column 5; the MCS are listed in column 4. The error in the approximate approach ranges from 10.7% to 23.4% for the noncoherent outcomes.

REFERENCES

[1] J. D. Andrews and T. R. Moss, *Risk and Reliability Assessment*: Longman, 1993.

[2] N. C. Rasmussen, "Reactor safety study: An assessment of accident risks in US commercial nuclear power plants," US Nuclear Regulatory Commission, 1975.

[3] W. E. Vesley, "A time dependent methodology for fault tree evaluation," *Nuclear Engineering and Design*, vol. 13, pp. 337–360, 1970.

[4] R. M. Sinnamon and J. D. Andrews, "Quantitative fault tree analysis using binary decision diagrams," *European J. Automation*, vol. 30, no. 8, pp. 1051–1071, 1996.

[5] ——, "Improved efficiency in qualitative fault tree analysis," *Quality and Reliability Engineering Int'l.*, vol. 13, pp. 293–298, 1997.

[6] ——, "Improved accuracy in quantitative fault tree analysis," *Quality and Reliability Engineering Int'l.*, vol. 13, pp. 285–292, 1997.

[7] A. Rauzy, "A brief introduction to binary decision diagrams," *European J. Automation*, vol. 30, no. 8, pp. 1033–1050, 1996.

[8] ——, "New algorithms for fault tree analysis," *Reliability Engineering and System Safety*, vol. 40, pp. 203–211, 1993.

[9] Groupe Aralia, "Computation of prime implicants of a fault tree within Aralia," in *Proc. European Safety and Reliability Assoc. Conf.*: ESREL, 1995, pp. 190–202.

[10] R. Gulati and J. B. Dugan, "A modular approach for analyzing static and dynamic fault trees," in *Proc. Ann. Reliability and Maintainability Symp.*, 1997, pp. 57–63.

[11] J. B. Dugan and S. Doyle, "Incorporating imperfect coverage into a BDD solution of a combinational model," *European J. Automation*, vol. 30, no. 8, pp. 1073–1086, 1996.

[12] R. E. Barlow and F. Proshan, *Mathematical Theory of Reliability*: SIAM, 1996.

John D. Andrews is a Senior Lecturer in the Department of Mathematical Sciences at Loughborough University. He joined this department in 1989 having previously gained nine years industrial experience at British Gas and two years lecturing experience at the University of Central England. His current research interests concern the assessment of the safety & risks of potentially hazardous industrial systems. This research has been heavily supported by funding from industry. Recent grants have been secured from Mobil North Sea Ltd., Daimler-Chrysler, and Rolls Royce Aero Engines.

Sarah J. Dunnett is a Lecturer in the Department of Mathematical Sciences at Loughborough University. Prior to her appointment in 1994, she has held positions at the Health and Safety Executive Laboratories, and Leeds University. Her research interests include: the numerical modeling of safety-related problems caused by contaminant flow around air samplers and, more recently, risk-assessment using event-trees and fault-trees. Dr. Dunnett is currently Vice-President of the Aerosol Society. Her research is supported by funding from the Health and Safety Executive.