Qualitative Analysis in Reliability & Safety Studies

R.B. Worrell
G.R. Burdick, Member IEEE

Abstract—The qualitative evaluation of system logic models is described as it pertains to assessing the reliability and safety characteristics of nuclear systems. Qualitative analysis of system logic models, i.e., models couched in an event (Boolean) algebra, is defined, and the advantages inherent in qualitative analysis are explained. Certain qualitative procedures that were developed as a part of fault-tree analysis are presented for illustration. Five fault-tree analysis computer-programs that contain a qualitative procedure for determining minimal cut sets are surveyed. For each program (SETS, MOCUS, PREP, MICSUP, ELRAFT), the minimal cut-set algorithm and limitations on its use are described. The recently developed common-cause analysis for studying the effect of common-causes of failure on system behavior is explained. This qualitative procedure does not require altering the fault tree, but does use minimal cut sets from the fault tree as part of its input. The method is applied using two different computer programs, COMCAN and SETS.

Key Words—Nuclear power, Nuclear reactor, Logic model, Safety, Fault tree, Qualitative analysis, Cut set, Common-cause failure.

Reader Aids:
Purpose: Tutorial
Special math needed for explanations: Boolean algebra, Probability
Special math needed for results: Same
Results useful to: Reliability, safety, and system analysts.

1. INTRODUCTION

The study of the behavior of complex nuclear systems has become increasingly important in recent years. The usefulness of nuclear systems when they behave as intended, coupled with the potential consequences if they should not, has stimulated considerable study of the reliability and safety characteristics of nuclear systems. In all of this activity, effort has been concentrated in developing and applying quantitative procedures to assess the reliability and safety of these systems. Even so, many useful qualitative procedures have been developed. Indeed, some qualitative procedures often provide the framework for subsequent quantitative procedures. We shall describe some of the qualitative procedures that have been developed and indicate how they are used to study the reliability and safety of nuclear systems.

The problems that interest us are couched in a Boolean algebra which, for our particular application, shall be referred to as event algebra. This means that

1) the basic events that are important in the behavior of the system must be identified;

2) the logical relationships that exist among the basic events must be specified;

3) the combinations of basic events that represent system behavior can be determined.

A generally accepted concept exists as to what is meant by qualitative analysis for this kind of problem, but we define qualitative analysis with respect to event algebra to eliminate possible ambiguity. Qualitative analysis is comprised of those analytic procedures that either do not require the assignment of values to any of the events defined on the event space, or which require only the assignment of the values 1 or 0 to some or all of these events. Ordinarily, the values are the probabilities of occurrence of the basic events, but some analytic techniques do involve assigning values other than event probabilities, e.g., the amount of time required to accomplish an event. Thus, qualitative analysis does not include any procedure that requires the application of a measure to the event space. The assignment of the value 1 or 0 to an event is simply equivalent to the assertion that the given event either occurs or does not occur, respectively.

The importance of qualitative analysis stems from its very nature, i.e., the results obtained are independent of any values that might be assigned to the events. Frequently, the values are only estimates at best, and qualitative analysis may well be a better approach to studying the behavior of a system under these conditions. If inaccurate values are used, invalid conclusions might be drawn. Moreover, even if the accuracy of the data is not a factor, the results that are obtained by quantitative procedures are only known to be valid for the particular values used to obtain them. Qualitative analysis, however, tends to deal with the problem on a more basic level, and to identify fundamental relationships that can be established from the logic model without quantification.

Fault-tree analysis is perhaps the most well-known analytic method in use today for studying the behavior of complex nuclear systems. Of course, other methods are also used: logic diagrams, cause-consequence charts, event trees, event-tree & fault-tree combinations, and many others including unpublished techniques. The differences among many of these methods are often ones chiefly of display. For example, a cause-consequence chart (CCC) can be mapped into event-tree & fault-tree combinations and both can be mapped into a set of fault trees for which the TOP events are the consequences of the CCC or event tree [1, 2]. Obviously, this mapping from one form of display to another can be accomplished if the model for each of the methods is equivalent to the basic underlying system logic. Nevertheless, we describe only qualitative procedures that were developed as a part of fault tree analysis, although with suitable modification the procedures could be applied to many of these other analytic methods.

The terminology and notation of fault tree analysis has, in general, been developed in one of two ways. In one development the concepts of Boolean algebra have been adopted [3-7], and in the other the concepts of cut and path sets have evolved from coherent-structure theory [8-10]. Thus, a prime-implicant of the Boolean function defined by an equation of system behavior, corresponds to a minimal cut-set obtained

from a system structure function. Differences as well as similarities occur in these approaches, but both approaches are appropriate as long as an event and its complement cannot both occur. We shall use the terminology and notation associated with minimal cut-sets except when the possible occurrence of both an event and its complement requires the concept of prime implicants.

The construction of fault trees is an important task in the overall activity of fault tree analysis. However, we address fault tree evaluation here and not fault tree construction. The system logic model (a fault tree) is assumed to be given and to correctly represent a state of affairs existing within the system. Direct analysis of the system and its components has ended and analysis of the logic model has begun.

An important aspect of fault tree evaluation is determining the minimal cut sets for the TOP event or some intermediate event. The importance stems from the fact that determining minimal cut sets is not only an analytic goal itself, i.e., finding all of the fundamental ways that an event can occur, but it is a required initial step in many other fault tree evaluation techniques. Several computer programs have been designed to achieve this important goal. Section 2 describes some of the computer programs that are used to determine minimal cut sets; in essence it surveys those programs that are designed to determine minimal cut sets using qualitative procedures.

The usefulness of qualitative procedures is not confined to determining minimal cut sets. Indeed, it is potentially useful throughout the evaluation process. For example, an area of recent concern is the effect upon a system by some "common" event, viz., an event that does not appear in the fault tree but directly affects some of the basic events that do appear. The interest centers around the development of techniques for ferreting out such dependence and evaluating its impact on the operation of the system. Section 3 presents common-cause analysis, a recently developed qualitative procedure for dealing with one aspect of this problem. Common-cause analysis provides a systematic method for locating secondary events which could cause the TOP event to occur.

2. COMPUTER PROGRAMS FOR FINDING MINIMAL CUT SETS

All of the simulation schemes for finding minimal cut sets [11 (FATE option), 12-15] are based on Monte Carlo techniques which require that probabilities of occurrence be assigned to the basic events. None of these, then, are qualitative procedures. However, the known deterministic schemes for finding minimal cut sets are all qualitative procedures. The distinction between Monte Carlo simulation and deterministic schemes may seem slight, because any procedure for finding the minimal cut sets for a given fault tree event would, in a general sense, be as satisfactory as any other procedure to accomplish the same task. However, Monte Carlo simulation procedures may not find all of the minimal cut sets. As stated by Vesely [9], "The Monte Carlo simulation method of determing minimal cut sets or minimal path sets never ensures that all such sets for a system have been found." Moreover, the minimal cut sets that are determined by these procedures will be those that are probabilistically important, i.e., whether or not a minimal cut set is found depends on the probabilities of the basic events that comprise the minimal cut set. The advantage of deterministic procedures for finding minimal cut sets is that they are inherently capable of delivering all of them. In practice, however, since the number of minimal cut sets for a given fault tree event can be enormous, the selection of some more manageable subset of the minimal cut sets is often necessary. Still, deterministic procedures provide the advantage of completeness in determining every element of the subset with certainty.

The surveyed computer programs are all fault-tree analysis programs and each contains a qualitative, deterministic algorithm for finding minimal cut sets. Each program has some capability for selecting a subset of the minimal cut sets for consideration or further processing, and all of the programs have received considerable use. Not much has been done to compare these programs, or more formally, the algorithms contained in them, even though a meaningful comparison of qualitative minimal cut-set algorithms would be welcome and useful. What little has appeared in the way of comparison is neither comprehensive nor persuasive; so no attempt is made to include comparison information in the descriptions.

2.1 SETS

The Set Equation Transformation System (SETS) [5, 6] symbolically and directly manipulates Boolean equations. By this kind of algebraic processing, an equation can be transformed into a more useful or desirable form, particularly by applying Boolean identities. Thus, SETS implements basic capabilities that can be invoked to determine the Boolean equation that represents the TOP event (or any intermediate event) in a fault tree.

By writing a SETS user program, the basic capabilities are invoked in the order and manner called for by the analyst using a special versatile higher level language. A program named SETS is used to read, interpret, and execute the statements of a SETS user program. SETS is coded in FORTRAN-Extended for the CDC 6600 computer. In addition to the AND and OR gates that are customary in fault-tree analysis, the fault trees used with SETS can contain EXCLUSIVE OR gates and SPECIAL gates. These additional gates are included because the program accommodates Boolean negation (NOT) as well as conjunction (AND) and disjunction (OR). The EXCLUSIVE OR gate has its normal interpretation; the SPECIAL gate simply allows the analyst to specify any logical combination of basic or intermediate events that cannot be readily expressed with other types of gates. The existing definitions of minimal cut and path sets are not adequate for equations that contain an event and its complement. Rather, the prime implicants of such an equation represent the fundamental ways that an event can occur [6].

The way in which a particular program determines the minimal cut sets of a fault tree, i.e., from top to bottom or vice versa is usually a pertinent part of the description of that

program. However, this characteristic is not pertinent with regard to SETS because equations can be developed in stages from the top down, or from the bottom up, or by a combination of both. Normally, the result of each stage in developing an equation will be simplified before proceeding to the next stage. However, as useful as this simplification may be at each stage, the processing required to develop an equation in one way can differ significantly from the processing required to develop it another way. Thus, when using SETS, the analyst can develop fault tree equations in whatever way best suits his purpose, but the choice should be made with care.

With SETS, basic events and intermediate events (gate output events) are assigned arbitrary alphanumeric names containing up to 16 characters. The fault tree representation used for computer input is thoroughly and extensively checked when it is read by the program, and the possibility of keypunch and other mechanical errors in the fault tree is all but eliminated. It is possible to select from all of the minimal cut sets only those which contain no more than n basic events. In addition, any basic events that are not to be counted in this process can be specified. Output options available to the analyst include the listing of an equation in disjunctive normal form. When the equation represents the logical sum of the minimal cut sets, a disjunctive normal form of display is tantamount to a list of the minimal cut sets.

2.2 MOCUS

MOCUS [10, 16] is a program used to determine the minimal sets, i.e., cut sets or path sets, from fault trees. It is coded in FORTRAN IV for the IBM 360/75 computer. The algorithm begins with the TOP event of the fault tree and proceeds, by successive substitutions of gate equations, to move down the tree until only basic events remain. As the expansion proceeds, a matrix is generated which is a disjunctive normal form representation of the equation at each stage of its development. Each row of the matrix is a term of the equation, and each term of the equation is a product of the elements in the row. Thus, substitutions for OR gates add rows to the matrix and substitutions for AND gates add elements to a row. Repeated elements in a row are removed as they occur and when only basic events remain in the matrix, the rows are the cut sets. The cut sets are compared, the supersets are eliminated, and the rows that remain are the minimal cut sets for the TOP event. The algorithm, then consists of substitution and expansion (application of the distributive law) together with application of the Boolean identities $P \cup P = P$ and $P \cup (P \cap Q) = P$.

The program is designed to accept only AND and OR gates. The fault tree description that is used as input to the program is extensively checked for errors. The program output includes a listing of all the minimal cut sets, or of those that contain up to n basic events, for the TOP event and for any desired intermediate event. MOCUS can be used in place of PREP.

2.3 PREP (COMBO Option)

The COMBO Option of the PREP [11], program is designed to determine minimal cut sets by a process of deterministic testing. In an exhaustive fashion, the basic events are imagined to have occurred first one at a time, then two at a time, ..., up to n at a time. For each combination, the logic of the fault tree is tested to determine whether or not that combination causes the TOP event. Combinations that cause the TOP event are recorded, and cut sets that are not minimal are eliminated using an element by element comparison to other cut sets. The result of this process leaves only the minimal cut sets.

The PREP program is coded in FORTRAN IV for the IBM 360/75 computer. The program is limited to analysis of fault trees that contain only AND and OR gates. Moreover, the number of combinations of basic events can become so large that, in practice, the combinations to be tested are often restricted to $n \leqslant 3$. However, the program does allow combinations as high as $n = 10$ to be specified.

2.4 MICSUP

The Minimal Cut Set Algorithm, Upward (MICSUP) program [17, 18] is used to obtain minimal cut sets starting with the basic events and working upward to the TOP event. A preprocessing program named TREEL is executed prior to MICSUP. TREEL reads the input representation of the fault tree, performs a few tasks that include testing the fault tree for errors, and then generates the representation of the fault tree that is required as input to MICSUP. The minimal cut sets for each intermediate event (gate output event) are then determined gate by gate up the tree by MICSUP. At a particular gate, the cut sets are determined and arranged according to cardinality. Within each cut set the basic events (elements) are ordered according to the numbers that were assigned to the basic events by TREEL, and an element by element comparison is made of the sets to determine containment. The supersets are then discarded leaving only the minimal cut sets.

The TREEL and MICSUP programs are coded in FORTRAN for the CDC 6400 computer. The program output includes the minimal cut sets for the TOP event and for any desired intermediate events.

2.5 ELRAFT

The Efficient Logic Reduction Analysis of Fault Trees (ELRAFT) program [7] is designed to find minimal cut sets using the unique factorization property of the natural numbers. Stated as a theorem the property reads: Every natural number greater than unity can be expressed as a product of powers of prime factors; the expression is unique except for the order in which the factors appear. For example, $60 = 2^2 \cdot 3 \cdot 5 = 3 \cdot 2^2 \cdot 5 = 5 \cdot 2^2 \cdot 3$ = etc. In the algorithm, every basic event is assigned a unique prime number. The tree is examined from the bottom up, and the cut sets for the gate events on successively higher levels are determined as a product of the numbers associated with each of the input events. Set containment is indicated whenever one number is a factor of another, e.g., the cut set indicated by 30 is logically contained in the cut set indicated by 15, and the cut set represented by the larger number

can be eliminated. Continuation of the procedure delivers all of the minimal cut sets for the TOP event.

The Program is coded in FORTRAN IV for the CDC 6600 computer. With respect to the fault tree representation used for input, the TOP event is designated as 1; the events of the next lower level are designated 11, 12, ..., 19; the events of the next lower level that come from 11 are designated 111, 112, ..., 119, those that come from 12 are designated 121, 122, ..., 129, ..., and those that come from 19 are designated 191, 192, ..., 199, etc., until every event is designated by a number. Thus, within one numbering sequence, both the number of levels and the number of inputs to a gate are limited to nine although both of these limitations can be circumvented. A limitation of greater severity is that the product of prime numbers representing a given minimal cut set can exceed the capacity of the machine to represent the number. Presently, the program will find minimal cut sets containing up to n basic events, $1 \leqslant n \leqslant 6$, for the TOP event and for other specified intermediate events.

3. COMMON-CAUSE ANALYSIS

We use the phrase 'common-cause of failure' in preference to the phrase 'common-mode failure' [19-22], because a minimal cut set can be comprised of basic events that represent the failure of components which are quite different and have no failure mode in common. Nevertheless, there can be secondary events that do not explicitly appear in a minimal cut set, but which, if they occur, cause one or more of the basic events in the cut set. For example, an earthquake, high temperature, and moisture are events which, if they occur, might cause certain of the basic events in the fault tree. A common-cause event, then, is a secondary event that causes one or more basic events. Thus, a multievent minimal cut set of basic events might become a single event cut set under the influence of some common-cause event.

The effect of common-cause events could be studied by altering the fault tree so that these events are explicitly represented in the tree, and then finding the minimal cut sets for the tree. Done in this way, the number of minimal cut sets would tend to increase substantially. Worse still, many of the minimal cut sets would be mixtures of basic events and common-cause events that would be somewhat difficult to interpret. However, a methodology has been developed that can be used to analyze a system under the influence of causal events without requiring that these events appear explicitly in the fault tree. The method is called common-cause analysis [23], and was developed by the Aerojet Nuclear Company for the US Energy Research and Development Administration under Contract AT (10-1)-1375.

As part of the common-cause analysis methodology, a system was developed for classifying common-cause events. Since any attempt to exhaustively list specific common-cause events is subject to both redundancies and omissions, the classification system is based instead on the concept of natural groupings of generic causes. By listing only generic causes (a generic cause represents a class of conditions), redundancies can be largely eliminated, and by grouping generic causes according to the nature of the causes, omissions can be minimized. The classification system has four categories of generic causes: Mechanical/Thermal, Electrical/Radiation, Chemical/Miscellaneous, and Common Links.

A partial listing of the generic causes in the Mechanical/Thermal category is given in Table I, along with some examples of specific secondary causes for each generic class. Changes to the category, either by deletion or addition of a generic cause, can be made without affecting the analytical portion of the methodology. This property is shared by all categories. The first three categories, however, do differ from the Common Link category in one respect. The Common Link category contains generic causes that transcend physical boundaries, e.g., components installed by the same contractor, components all supplied by the same power source, etc. The generic causes of the other categories can be considered without regard to physical boundaries, but the option exists to consider the effect of boundaries that are said to contain a given secondary cause. The boundaries for a particular generic cause define the *domain* for that generic cause. For example, an oil spill would normally be confined to the room in which it occurred. This option provides a way of tempering any conclusions that may be indicated from the analysis of the generic causes alone, with the information that the domains of certain generic causes may preclude their occurrence together.

TABLE I
Generic Causes–Mechanical/Thermal

Generic Cause Symbol	Generic Cause	Specific Secondary Causes
I	Impact	Pipe whip, water hammer, missiles earthquake, structural failure
V	Vibration	Machinery in motion, earthquake
P	Pressure	Explosion, out-of-tolerance system changes (pump overspeed, flow blockage)
G	Grit	Airborn dust, metal fragments, generated by moving parts with inadequate tolerances
S	Stress	Thermal stress at welds of dissimilar metals
T	Temperature	Fire, lightning, welding equipment, coolant system faults, electrical short circuits

A Common Cause Analysis (COMCAN) program was written [25] to implement the methodology. The program requires as input whatever minimal cut sets have been selected from the fault tree and the generic cause susceptibility for each basic event in each category. If the location option is used, the domain of each generic cause must also be supplied as input. In essence, the algorithm then searches for those minimal cut sets that are comprised of basic events that are all susceptible to

the same generic cause, and this search is repeated for each category. Suppose, for example, that four minimal cut sets have been selected {BE1, BE3, BE4}, {BE1, BE2}, {BE3, BE6}, and {BE2, BE4, BE5, BE6}, where BEi, $1 \leqslant i \leqslant 6$, represent basic events. Now assume that the location option is not being used and that the Mechanical/Thermal generic cause susceptibility for the basic events is as indicated in Table II. The program would find that

{BE1, BE3, BE4} is susceptible to P (pressure),
{BE1, BE2} is susceptible to I (impact) and G (grit),
{BE3, BE6} has no Mechanical/Thermal susceptibility, and
{BE2, BE4, BE5, BE6} is susceptible to G (grit).

TABLE II
Example Mechanical/Thermal Susceptibility

Basic Event	Generic Cause Symbols	Generic Cause Susceptibility
BE1	I, P, G, T	Impact, Pressure, Grit, Temperature
BE2	I, G	Impact, Grit
BE3	P	Pressure
BE4	P, G	Pressure, Grit
BE5	G	Grit
BE6	G, T	Grit, Temperature

Interestingly, a closely related analytical technique was being developed independently (by Sandia Laboratories for the US Energy Research and Development Administration) about the same time that the common cause analysis was developed [24]. However, in this case the emphasis was on the development of a technique based on the manipulation of Boolean equations. It was known that if a Boolean manipulation technique could be identified, the SETS program, which was already available, could be used to implement the technique and no further computer program development would be required. Like the development of the common cause analysis method, interest centered on a way of identifying from the minimal cut sets of a fault tree, those minimal cut sets whose constituent basic events could be caused by some event that was not explicitly represented. However, unlike the common cause analysis method, which identifies the multievent minimal cut sets whose constituent basic events can be caused by a single common cause event, the equation manipulation technique shows the effect of common events on all of the minimal cut sets.

The equation manipulation technique that was developed is based simply on a transformation of variables that can be directly and easily accomplished with the SETS program. Although the technique was developed for slightly different applications [25], its use can be shown with the same example that was used to illustrate common cause analysis. Recall the example, and assume that the minimal cut sets are represented by a Boolean equation that is the logical sum of them

X = BE1 ∩ BE3 ∩ BE4 ∪ BE1 ∩ BE2 ∪ BE3 ∩ BE6 ∪ BE2 ∩ BE4 ∩ BE5 ∩ BE6.

Now suppose that the Mechanical/Thermal generic causes for each basic event (Table II) are represented by the equations

$$\text{BE1} = I \cup P \cup G \cup T$$

$$\text{BE2} = I \cup G$$

$$\text{BE3} = P$$

$$\text{BE4} = P \cup G$$

$$\text{BE5} = G$$

$$\text{BE6} = G \cup T.$$

By the proper sequence of variable transformations (two are required in order to preserve the basic event names and at the same time avoid circular substitutions), a SETS user program can be written which, when executed, will generate and display the equation

TERM NUMBER	NUMBER OF LITERALS	$X =$
1	3	BE1 ∧ BE2 ∧ G ∨
2	3	BE1 ∧ BE2 ∧ I ∨
3	4	BE3 ∧ BE6 ∧ P ∧ G ∨
4	4	BE3 ∧ BE6 ∧ P ∧ T ∨
5	4	BE1 ∧ BE3 ∧ BE4 ∧ P ∨
6	5	BE4 ∧ BE2 ∧ BE6 ∧ BE5 ∧ G

For this example, six terms remain in the equation after simplification. The terms represent the effect on all of the minimal cut sets (4) by all of the Mechanical/Thermal generic causes. The terms numbered 1, 2, 5 and 6 correspond to those that are found by the common-cause analysis methodology. Terms number 3 and 4 represent the minimal cut sets which are susceptible to the occurrence of two generic events. For this example all terms that remained after simplification were retained, but the option exists to limit the terms retained to only those that contain no more than n generic events.

4. CONCLUSIONS

The role of qualitative analysis in assessing the reliability and safety characteristics of nuclear systems has been discussed. Qualitative analysis of system logic models was defined and the advantages of this kind of analysis were described. In fault tree analysis, for example, qualitative procedures for finding minimal cut sets are preferable to the Monte Carlo simulation techniques for accomplishing this task. The several computer

programs that were described indicate the considerable effort that has been directed toward developing qualitative minimal cut set algorithms. No attempt was made to compare these programs as to their efficiency, flexibility, etc., but it is believed that the results of such a comparison would be useful.

A new qualitative analytical method called common-cause analysis was also described. The description served not only to introduce the method, but also to show that qualitative procedures can be useful throughout the process of evaluating system logic models. The input required for common-cause analysis is comprised of the minimal cut sets for a selected fault tree event and a generic classification of secondary events. From this input information, potential secondary sources of failure in nuclear systems can then be determined. Common-cause analysis has been directly implemented in a computer program called COMCAN, and it was shown that the basic capabilities available in the SETS program could also be used to implement the methodology.

A preference was expressed for the phrase 'common-cause of failure' as opposed to the phrase common mode failure for referring to the secondary events involved in this new methodology. Common-mode failure has been used to refer to a collection of so many kinds of common failure events, that it is not possible to distinguish between them even when they are quite different. As more effort is devoted to developing techniques to deal with the effect of common failures on the behavior of a system, the terminology and notation should be refined so that the concepts involved are accurately described and represented. Thus the preference for the phrase common-cause of failure and for the resulting name of the methodology: common-cause analysis.

REFERENCES

[1] J.R. Taylor, *Sequential Effects in Failure Mode Analysis,* Risö-M-1740, Danish A.E.C., Roskilde, Denmark.

[2] R.A. Evans, "Fault-trees and cause-consequence charts," *IEEE Trans. Rel.,* vol. R-23, p. 1, April 1974.

[3] R.G. Bennetts, "On the analysis of fault trees," *IEEE Trans. Rel.,* vol. R-24, pp. 175-185, August 1975.

[4] J.B. Fussell, G.J. Powers, R.G. Bennetts, "Fault-trees–a state of the art discussion," *IEEE Trans. Rel.,* vol. R-23, pp. 51-55, April 1974.

[5] R.B. Worrell, *Set Equation Transformation System (SETS),* SLA-73-0028A Sandia Laboratories, Albuquerque, New Mexico, USA, May 1974.

[6] R.B. Worrell, "Using the set equation transformation system in fault tree analysis," *Reliability and Fault Tree Analysis,* SIAM Conference Volume, pp. 165-185, 1975.

[7] S.N. Semanderes, "ELRAFT, a computer program for the efficient logic reduction analysis of fault trees," *IEEE Trans. Nuclear Science,* vol. NS-18, pp. 481-487, 1971.

[8] J.D. Esary, F. Proschen, "Coherent structures of non-identical components," *Technometrics,* vol. 5, pp. 191-209, May 1963.

[9] W.E. Vesely, *Analysis of Fault Trees by Kinetic Tree Theory,* IN-1330, Idaho Nuclear Corporation, Idaho Falls, Idaho USA, October 1969.

[10] J.B. Fussell, E.B. Henry, N.H. Marshall, *MOCUS–A Computer Program to Obtain Minimal Sets from Fault Trees,* ANCR-1156, Aerojet Nuclear Company, Idaho Falls, Idaho USA, March 1974.

[11] W.E. Vesely, R.E. Narum, *PREP and KITT: Computer Codes for the Automatic Evaluation of a Fault Tree,* IN-1349, Idaho Nuclear Corporation, Idaho Falls, Idaho USA, August 1970.

[12] P.A. Crosetti, *Computer Program for Fault Tree Analysis,* DUN-5508, Douglas United Nuclear, Inc., Richland, Washington USA, April 1969.

[13] R.J. Schroder, *Fault Tree Simulation with Importance Sampling,* Documentation For Computer Program AS 2798, The Boeing Co., Seattle, Washington USA, 1967.

[14] P.A. Crosetti, *Fault Tree Simulation Computer Program,* DUN-7697, Douglas United Nuclear, Inc., Richland, Washington USA, June 1971.

[15] B.J. Garrick, et al., *Reliability Analysis of Nuclear Power Plant Protective Systems,* HN-190, Holmes and Narver, Inc., May 1967.

[16] J.B. Fussell, W.E. Vesely, "A new methodology for obtaining cut sets for fault trees," *Trans. Amer. Nucl. Soc.,* vol. 15, pp. 262-263, June 1972.

[17] P.K. Pande, M.E. Spector, P. Chatterjee, *Computerized Fault Tree Analysis:* TREEL and MICSUP, ORC 75-3, Operations Research Center, University of California, Berkeley, California USA, April 1975.

[18] P. Chatterjee, *Fault Tree Analysis: Min Cut Set Algorithms,* ORC 74-2, Operations' Research Center, University of California, Berkeley, California USA, January 1974.

[19] W.C. Gangloff, "Common mode failure analysis," *IEEE Trans. Power Apparatus and Systems,* vol. PAS-94, pp. 27-30, January/February 1975.

[20] E.P. Epler, "Common mode failure considerations in the design of systems for protection and control," *Nuclear Safety,* vol. 10, pp. 38-45, January/February 1969.

[21] E.P. Epler, "The ORR emergency cooling failure," *Nuclear Safety,* vol. 11, pp. 323-327, July/August 1970.

[22] W.C. Gangloff, "Common mode failure analysis is 'in'," *Electrical World,* pp. 30-33, October 1972.

[23] J.B. Fussell, D.M. Rasmuson, J.R. Wilson, G.R. Burdick, J.C. Zipperer, *A Collection of Methods for Reliability and Safety Engineering,* ANCR-1273, Aeroject Nuclear Company, Idaho Falls, Idaho USA, February 1976.

[24] R.B. Worrell, *Common Event Analysis Using Variable Transformations,* SAND76-0024, Sandia Laboratories, Albuquerque, New Mexico USA, to be published 1976.

[25] G.R. Burdick, N.H. Marshall, J.R. Wilson, "COMCON–A computer program for common-cause analysis," ANCR-1314, Aerojet Nuclear Company, Idaho Falls, Idaho USA, May 1976.

Manuscript received December 1, 1975; revised January 11, 1976

R.B. Worrell//System Studies Engineering Div.//Sandia Laboratories//Albuquerque, New Mexico 87115 USA

R.B. Worrell is a staff member at Sandia Laboratories, a prime contractor to the U.S. Energy Research and Development Administration. He received a B.S. in Physics and an M.S. in Engineering from the University of Oklahoma. He was Director of Programming at the University of Oklahoma's Computer Laboratory and has designed and implemented several large computer software systems. While at Sandia he has been involved in system safety and reliability analysis, with particular emphasis on qualitative analysis as it can be acheived through the symbolic manipulation of Boolean equations. He designed and implemented the Set Equation Transformation System (SETS) which is a computer software system for this kind of analysis. He has authored and coauthored papers on Boolean algebra and the use of SETS in graph theory and fault-tree analysis.

Dr. G.R. Burdick//Systems Analysis Methods & Procedures//Aerojet Nuclear Company//550 Second Street//Idaho Falls, Idaho 83401 USA

G.R. Burdick is an Associate Scientist with Aerojet Nuclear Company. He received an A.B. degree in physics from Miami (Ohio) and A.M. and Ph.D. degrees in mathematics from the University of Cincinnati. Dr. Burdick has taught mathematics at the University of Cincinnati and Northern Kentucky State College. Prior to his appointment with Aerojet Nuclear company he was employed as a guidance system analyst by the Autonetics Group of Rockwell International. He has coauthored research papers in univalent function theory and nuclear reactor reliability and safety analysis. His present main area of interest is the development of methods and procedures for reliability and safety analysis of nuclear power plant systems.

¤ ¤ ¤

Book Reviews

Ralph A. Evans, Product Assurance Consultant

Mechanical Reliability

A.D.S. Carter, 1972, $17.50, 146 pp. Halsted Press, 605 Third Avenue, NY 10016 USA. ISBN: 333 13831 7.

Table of Contents

This book has many good parts in it. It is probably a good addition to a company library, but few individuals will find it worth the price. A good quote from the book is "... although a considerable effort has been expended in recent years in formulating a 'science' of reliability, by far the greater part of this work is just plain common (engineering) sense combined with an effort to anticipate the consequences," I interpret this to mean that the engineer must pay an uncommonly large amount of attention to detail – which is true.

Chapter 2 is over 1/3 of the book and is on the math/stat aspects of reliability. It has some good practical (non-theoretical) advice in it (see page 11) but in large part is irrelevant to many problems that many mechanical machine designers face. The discussion on weak-link vs. series systems is poor. What ought to have been said is that failures are not *s*-independent if any ordering of lives (or strength) is known; e.g., if the weakest link (shortest-lived unit) is known, then of course the reliability is not the product of all the series reliabilities, but is the reliability of the weak link (because it must fail first).

Chapter 3 on design is reasonably good. It applies to fairly large production runs, not to much of machine design. Again, it is the detailed math/stat where the author gets in trouble (e.g., "... on account of the flexibility of the Weibull function to represent any distribution." Obviously, the Weibull does NOT represent ANY distribution.)

Chapter 4 on manufacturing is too short. It is here that much reliability growth (for production units) will take place.

Chapter 5 largely treats maintenance and is generally good. It is not emphasized enough that the failure rate is likely to be highest shortly after maintenance (preventive or otherwise).

Chapter 6 on random failures is (thankfully) short. The word "random" is unwisely (although he has lots of company) used to represent a Poisson process with constant failure rate.

Chapter 7 management is generally good. Specs such as MIL-Q-9858 are mentioned; they are worthwhile for many companies to use.

All in all, most of what is said is not peculiar to mechanical reliability and much that tends to be peculiar to mechanical reliability is not mentioned.

Product Liability: A Management Response

Irwin Gray with Albert L. Bases, Charles H. Martin, Alexander Sternberg, 1975, $12.95, 239 pp. Amacom, a Division of American Management Associations//135 West 50th Street//New York, NY 10020 USA. ISBN: 0-8144-5373-2; LCCCN: 75-1261.

Table of Contents

This is generally a good book. It certainly provides a good introduction to the topic for nonexperts. Some people might argue with some of the emphasis or completeness here or there but, given the constraints of size and reader background, the authors have made reasonable tradeoffs. Naturally, as with any technical book, it needs to be read critically.

Product liability is an up and coming topic (for many companies, it's already here) and one with which reliability and quality engineers ought to be familiar. Another source of information is the several Proceedings of the Product Liability Prevention Conferences.

Chapters 1 and 2 are good. Chapter 3 is generally good, except for the rather poor section on "The company's attitude toward profits". That section is oversimplified to the point of being ridiculous. On p. 77, the concept of safety margin is poorly defined, but is not part of the mainstream of the section.

Chapters 4 and 5 are good. Chapter 6 is generally good except for a few minor points. On p. 157, there is the presumption that researchers (on safe noise levels) always know what they're talking about – would that they did. Also on p. 157, the section on "Reputation" presumes more awareness and memory on the part of consumers than often exists. The section on checklists is good; the difficulty is getting designers really to use them. More often than not, they are honored in the breach. On p. 188, the short section on "Fault tree analysis" is ok only for those who won't remember any of the details.

Chapter 7 is generally good. The discussion on quality costs is grossly oversimplified. It too is okay only for those who won't remember any of the details. On p. 221, the discussion of trade associations ought to have mentioned (at least in passing) the caution against violating anti-trust laws.

All in all, the book is well written and its points are made clearly. It's not a bad price; so you ought seriously to consider buying a copy.