INL/CON-08-14682
PREPRINT

Representing Common-Cause Failures in the SAPHIRE Software

IMECE 2008

Curtis Smith

November 2008

Proceedings of Proceedings of IMECE2008
2008 ASME International Mechanical Engineering Congress and Exposition
November 2-6, 2008, Boston, Massachusetts, USA

IMECE2008-67130

REPRESENTING COMMON-CAUSE FAILURES IN THE SAPHIRE SOFTWARE

Curtis Smith
Idaho National Laboratory
Idaho Falls, Idaho, USA

ABSTRACT

Currently, the risk analysis software SAPHIRE has implemented a common-cause failure (CCF) module to represent standard CCF methods such as alpha-factor and multiple Greek letter approaches. However, changes to SAPHIRE are required to support the Nuclear Regulatory Commission's 2007 "Risk Assessment Standardization Project" CCF analysis guidance for events assessment. This guidance provides an outline of how both the nominal CCF probabilities and conditional (e.g., after a redundant component has failed) CCF probabilities should be calculated. Based upon user-provided input and extending the limitations in the current version of SAPHIRE, the CCF module calculations will be made consistent with the new guidance. The CCF modifications will involve changes to (1) the SAPHIRE graphical user interface directing how end-users and modelers interface with PRA models and (2) algorithmic changes as required. Included in the modifications will be the possibility to treat CCF probability adjustments based upon failure types (e.g., independent versus dependent) and failure modes (e.g., failure-to-run versus failure-to-start).

In general, SAPHIRE is being modified to allow the risk analyst to define a CCF object. This object is defined in terms of a basic event. For the CCF object, the analyst would need to specify a minimal set of information, including:

- The number of redundant components
- The failure criteria (how many component have to fail)
- The CCF model type (alpha-factor, MGL, or beta-factor)
- The parameters (e.g., the alpha-factors) associated with the model
- Staggered or non-staggered testing assumption
- Default level of detail (expanded, showing all of the specific failure combinations, or not)

This paper will outline both the theory behind the probabilistic calculations and the resulting implementation in the SAPHIRE software.

INTRODUCTION

In an earlier paper (Smith, 1998), we described the general process of using a probabilistic risk assessment (PRA) model to evaluate operational events at nuclear power plants. While the application of using a PRA to evaluate has increased, there are still analysis areas that are problematic both on the research and application fronts. For example, the treatment of operator actions specific to event-specific context remains difficult. Also, of similar difficulty is the assessment of dependency when dealing with failed components. It is this later issue, that of event-driven failures and their relationship to common-cause failures (CCF), which we address in this paper.

Like operator actions, CCF contributions to the overall risk tend to dominate. Consequently, it is vital that the estimate for these dependent failure events be made to a degree commiserate with the application of the analysis in general. Thus, to support decision making, one needs to have high-fidelity PRA calculations. When a component fails, that failure provides information or evidence about the likelihood of system failure.

The U.S. Nuclear Regulatory Commission (NRC) has recently modified it approach to account for the dependent failure calculation. Consequently, changes to the SAPHIRE software (developed by the NRC) are required to support changes outlined in the NRC's 2007 "Risk Assessment Standardization Project" CCF analysis guidance for events assessment. This guidance provides information of how both the nominal CCF probabilities and conditional (e.g., after a redundant component has failed) CCF probabilities should be calculated.

NOMENCLATURE

P( ) = probability
P(A|B) = probability of A given B has occured.

BACKGROUND

The definition of "independence" implies that two redundant components, A and B, behave according to:

$$P(\text{A and B}) = P(A)\,P(B)\ , \qquad \textbf{(1)}$$

100% of the time.

To illustrate this point, we will cover a simple case of two components (A and B), where the probability of the components failing are:

P(A) = 0.055
P(B) = 0.055 .

If A and B are independent, then from Equation 1, P(A and B) = (0.055)(0.055) = 0.0030. These probabilities are shown in the Venn diagram in Figure 1.

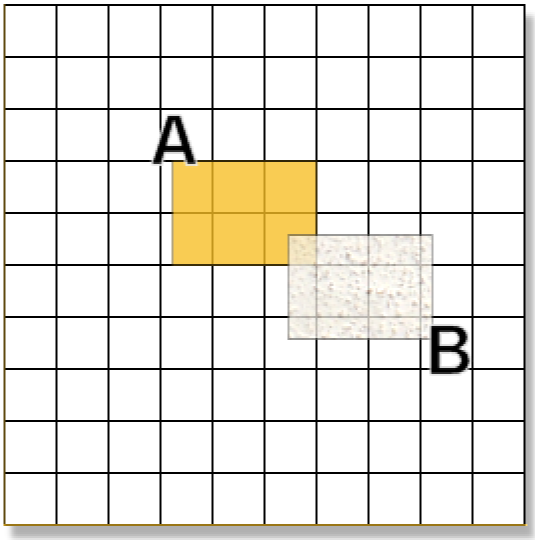


Figure 1. Venn diagram showing two failure events that are independent.

In the case where the failure of A and B are not independent (i.e., a dependence exists between the two components), the overlap of the two components in the Venn diagram changes. If the two components are more likely to fail together compared to the independence case, then the overlap area on the Venn diagram increases. For example, if A and B are not independent and have a joint failure probability P(A and B) = 0.1 then the Venn diagram shown in Figure 2 is applicable.

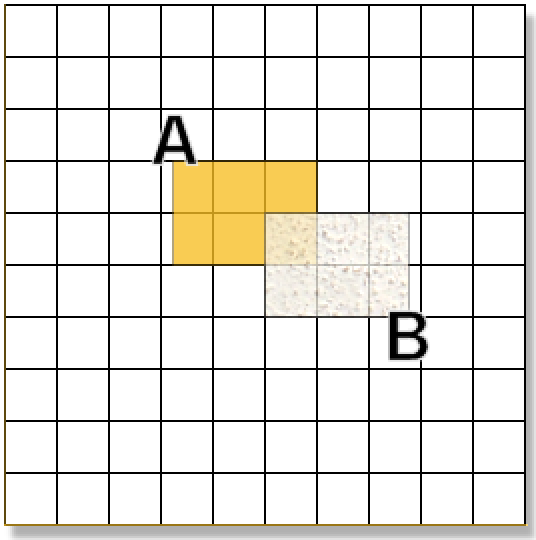


Figure 2. Venn diagram showing two failure events that are dependent.

When components A and B are not independent, the equation to determine the failure of both components is:

$$P(\text{A and B}) = P(A|B)\,P(B) = P(B|A)\,P(A) \qquad \textbf{(2)}$$

If component B is removed from service for testing or maintenance (i.e., not failed), then the system failure probability P(A and B) becomes just P(A), as illustrated in Figure 3.

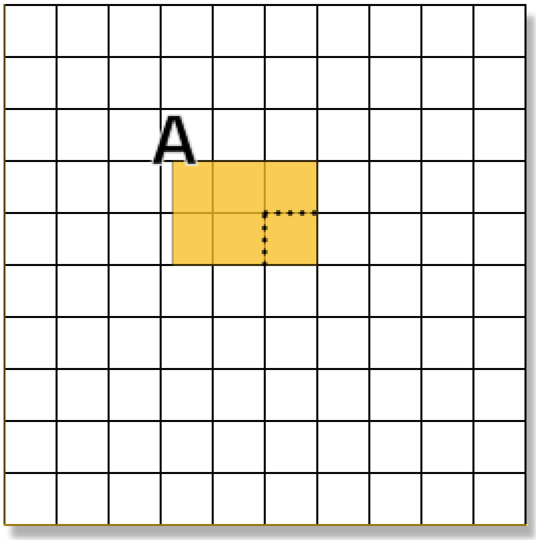


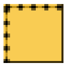
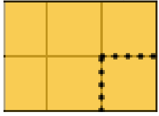
Figure 3. Venn diagram where component B is removed from service.

However, if component B fails, then we need to condition on this failure. We can use Equation 2 to define the notion of a conditional probability:

P(A|B) = P(A and B) / P(B) , assuming that P(B) > 0.

When component B is failed, this conditional probability above is an expression:

P(A|B) = P(A and B) / P(B) = 0.01 / 0.055 = 0.1818

This value (0.1818) is the relative fraction of P(A and B)  to P(B) , or slightly above 18%.

In order to make these kinds of conditional adjustments, we need to be able to modify the CCF parts of a PRA to reflect what is known about a component failure or inoperability. It is these types of calculations that motivated the NRC to modify its conditional CCF calculations and embody these calculations into its SAPHIRE software [Rasmuson and Kelly, 2007].

CCF MODELING

The CCF calculation featured in SAPHIRE is based on the Basic Parameter Model. A group of four components with a failure of all four components necessary was selected to demonstrate the calculations, but other group sizes (up to eight) may be specified in SAPHIRE.

The event failures are denoted as follows:

I_1, I_2, $I_{3,}$ I_4 Indicates the independent failure of the subscripted component.

C_{12}, C_{13}, $C_{14,}$ C_{23}, C_{24}, C_{34} Indicates the common-cause failure of the two subscripted components.

$C_{123,}$ $C_{124,}$ $C_{134,}$ C_{234} Indicates the common-cause failure of the three subscripted components.

$C_{1234\,-}$ Indicates the common-cause failure of the four subscripted components.

The basic parameter model for CCF analysis defines the following:

$$1_T = I_1 \cup C_{12} \cup C_{13} \cup C_{14} \cup C_{123} \cup C_{124} \cup C_{134} \cup C_{1234}$$

$$2_T = I_2 \cup C_{12} \cup C_{23} \cup C_{24} \cup C_{123} \cup C_{124} \cup C_{234} \cup C_{1234}$$

$$3_T = I_3 \cup C_{13} \cup C_{23} \cup C_{34} \cup C_{123} \cup C_{134} \cup C_{234} \cup C_{1234}$$

$$4_T = I_4 \cup C_{14} \cup C_{24} \cup C_{34} \cup C_{124} \cup C_{134} \cup C_{234} \cup C_{1234}$$

where the subscript $_T$ denotes total failure from all causes for the i'th component.

The failure probability of $1_{T,}$ $2_{T,}$ $3_{T,}$ or 4_T is given by the following equation:

$$Q_T = Q_1 + 3Q_2 + 3Q_3 + Q_4$$

where, because of symmetry assumptions in the Basic Parameter Model,

$Q_1 = P[I_1] = P[I_2] = P[I_3] = P[I_4]$

$Q_2 = P[C_{12}] = P[C_{13}] = P[C_{14}] = P[C_{23}] = P[C_{24}] = P[C_{34}]$

$Q_3 = P[C_{123}] = P[C_{124}] = P[C_{134}] = P[C_{234}]$

$Q_4 = P[C_{1234}]$.

The definitions of the Q values, Q_1, Q_2, Q_3, Q_4,.. Q_n , are where the CCF models (e.g., alpha-factor, MGL, or beta-factor) calculations vary. We will demonstrate just the alpha-factor approach since this is the model used by the NRC.

Alpha-Factor Model (non–staggered testing) methodology defines the Q values in the following manner.

$$Q_1^{(4)} = \frac{\alpha_1^{(4)}}{\alpha_t} Q_T \qquad Q_2^{(4)} = \frac{2\alpha_2^{(4)}}{3\alpha_t} Q_T$$

$$Q_3^{(4)} = \frac{\alpha_3^{(4)}}{\alpha_t} Q_T \qquad Q_4^{(4)} = 4\frac{\alpha_4^{(4)}}{\alpha_t} Q_T$$

The generalization of this definition is given by

$$Q_k^{(m)} = \frac{k}{\begin{pmatrix} m-1 \\ k-1 \end{pmatrix}} \left(\frac{\alpha_k^{(m)}}{\alpha_t} Q_T \right)$$

where

$$\alpha_t = \sum_{k=1}^{m} k\alpha_k^{(m)} \qquad \text{and} \qquad \begin{pmatrix} m-1 \\ k-1 \end{pmatrix} = \frac{(m-1)!}{(k-1)!(m-k)!}$$

Alpha-Factor Model (staggered testing) methodology defines the Q values in the following manner:

$$Q_1^{(4)} = \alpha_1^{(4)} Q_T \qquad Q_2^{(4)} = \frac{1}{3}\alpha_2^{(4)} Q_T$$

$$Q_3^{(4)} = \frac{1}{3}\alpha_3^{(4)} Q_T \qquad Q_4^{(4)} = \alpha_4^{(4)} Q_T$$

The generalization of this definition is given by

$$Q_k^{(m)} = \frac{1}{\begin{pmatrix} m-1 \\ k-1 \end{pmatrix}} \left(\alpha_k^{(m)} Q_T \right)$$

where

$$\begin{pmatrix} m-1 \\ k-1 \end{pmatrix} = \frac{(m-1)!}{(k-1)!(m-k)!}$$

The expression to determine the mode of the alpha factors themselves is:

$$\alpha_k^{(m)} = \frac{n_k}{\sum_{j=1}^{m} n_j}$$

where n_k is the number of observed events where k components fail due to CCF in a group of redundant components of size m. This expression for the α's is independent of the staggered or non-staggered testing scheme.

A key to understanding the SAPHIRE CCF plug-in functions is to think in terms of minimal cut sets. The following list contains the 15 cut sets, for the four-redundant component system if the success criterion is 1-of-4, that describe the failure of these components.

[I_1, I_2, $I_{3,}$ I_4]

[I_1, $I_{2,}$ C_{34}]

[I_1, $I_{3,}$ C_{24}]

[I_1, I_4, C_{23}]

[I_2, $I_{3,}$ C_{14}]

[I_2, I_4, C_{13}]

[I_3, $I_{4,}$ C_{12}]

[C_{12}, C_{34}]

[C_{13}, C_{24}]

[C_{14}, C_{23}]

[I_1, C_{234}]

[I_2, C_{134}]

[I_3, C_{124}]

[I_4, C_{123}]

[C_{1234}] .

The failure probability for the system, which we denote by Q_s, is given in terms of the Basic Parameter Model (for example, see Mosleh [1998]) by:

$Q_s = Q_1^4 + 6Q_1^2Q_2 + 3Q_2^2 + 4Q_1Q_3 + Q_4$

Q-values Assuming non-Staggered Testing

Calculating the Q values from the alpha factors, assuming a non-staggered testing scheme yields the following (the alpha factor values are in **bold**):

$$\alpha_t = \sum_{k=1}^{m} k\alpha_k^{(m)} = 1*\mathbf{0.9958} + 2*\mathbf{0.0030} + 3*\mathbf{0.0008} + 4*\mathbf{0.0004}$$

$$= 1.0058$$

$Q_1^{(4)} = \frac{\alpha_1^{(4)}}{\alpha_t} Q_T$ = 0.9958/1.0058 * 0.01 = 9.9E-3

$Q_2^{(4)} = \frac{2\alpha_2^{(4)}}{3\alpha_t} Q_T$ = 2*0.0030/3*1.0058*0.01 = 1.99E-5

$Q_3^{(4)} = \frac{\alpha_3^{(4)}}{\alpha_t} Q_T$ = 0.0008/1.0058 * 0.01 = 7.95E-6

$Q_4^{(4)} = 4\frac{\alpha_4^{(4)}}{\alpha_t} Q_T$ = 4*0.0004/1.0058*0.01 = 1.59E-5

Calculating a value of Q_s from the above equation yields the following:

$Q_s = Q_1^4 + 6Q_1^2Q_2 + 3Q_2^2 + 4Q_1Q_3 + Q_4$

$= 9.9E\text{-}3^4 + 6*(9.9E\text{-}3)^2 * (1.99E\text{-}5) + 3*(1.99E\text{-}5)^2 + 4*(9.9E\text{-}3)*(7.95E\text{-}6) + 1.59E\text{-}5$

= 1.593E-5

Q Values Assuming Staggered Testing

Calculating the Q values under a staggered testing scheme yields the following:

$Q_1^{(4)} = \alpha_1^{(4)} Q_T$ = 0.99*0.01 = 9.9E-3

$Q_2^{(4)} = \frac{1}{3}\alpha_2^{(4)} Q_T$ = 0.006*0.01/3 = 2.0E-5

$Q_3^{(4)} = \frac{1}{3}\alpha_3^{(4)} Q_T$ = 0.0024*0.01/3 = 8.0E-6

$Q_4^{(4)} = \alpha_4^{(4)} Q_T$ = 0.0016*0.01 = 1.6E-5

Where $\alpha_1 = 0.99$, $\alpha_2 = 0.006$, $\alpha_3 = 0.0024$, $\alpha_4 = 0.0016$, and $Q_T = 0.01$

Calculating value of Q_s from the above equation yields the following:

$Q_s = Q_1^4 + 6Q_1^2Q_2 + 3Q_2^2 + 4Q_1Q_3 + Q_4$

$= 9.9E\text{-}3^4 + 6*(9.9E\text{-}3^2 * 2.0E\text{-}5) + 3*(2.0E\text{-}5)^2 + 4*(9.9E\text{-}3)*8.0E\text{-}6) + 1.6E\text{-}5$

= 1.633E-5

SAPHIRE 8 IMPLEMENTATION OF CCF MODELING

To demonstrate the CCF calculation, we will create a new CCF object. We will model a system of three redundant motor operated valves (MOVs), represented by basic events MOV-CC-A, MOV-CC-B, and MOV-CC-C. The success criterion for this system is that two components have to work for the system to work. Consequently, if two (of three) components fail, the system is failed.

The basic event representing CCF will be called MY_CCF. The fault tree representing the system is shown in Figure 4.

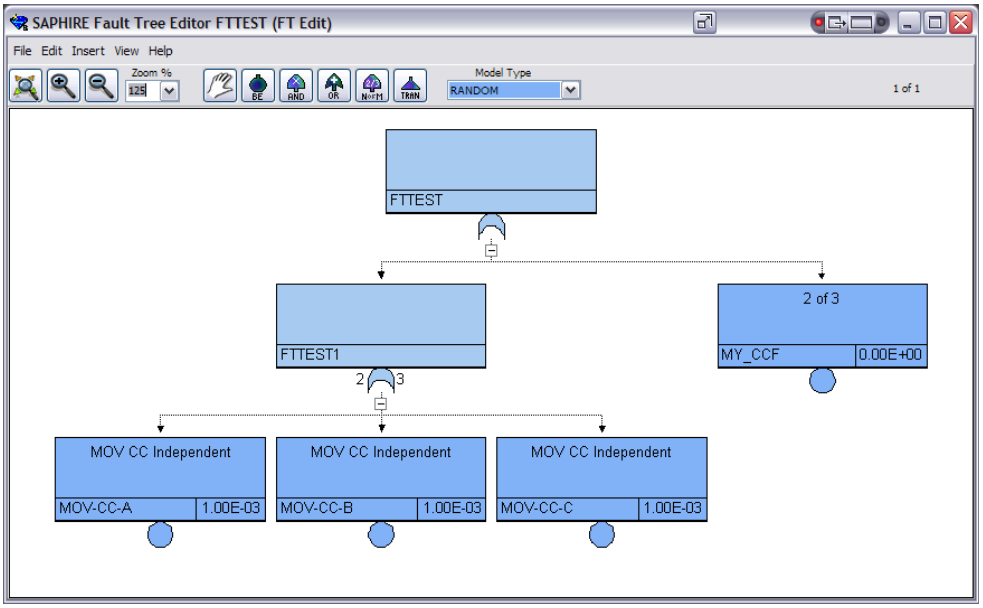
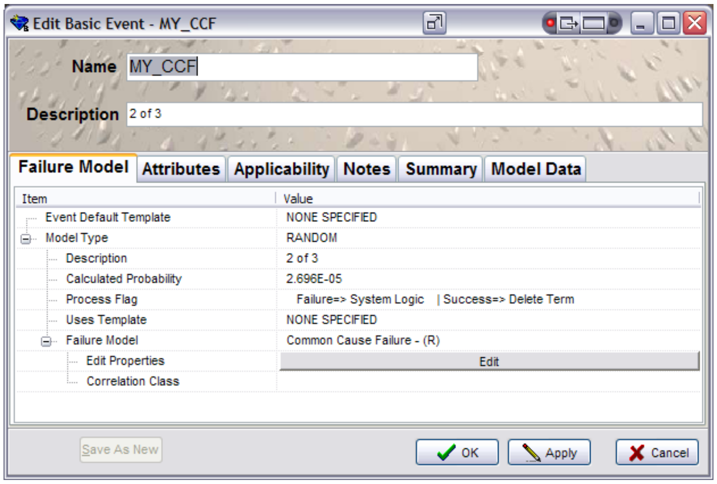


Figure 4. Fault tree with CCF representing the example system.

The basic event MY_CCF is a part of the fault tree graphic as shown above. However, this is a basic event and must be edited. Editing this basic event shows a screen similar to that in Figure 5.



Name: MY_CCF

Description: 2 of 3

Item	Value
Event Default Template	NONE SPECIFIED
Model Type	RANDOM
Description	2 of 3
Calculated Probability	2.696E-05
Process Flag	Failure=> System Logic | Success=> Delete Term
Uses Template	NONE SPECIFIED
Failure Model	Common Cause Failure - (R)
Edit Properties	Edit
Correlation Class	

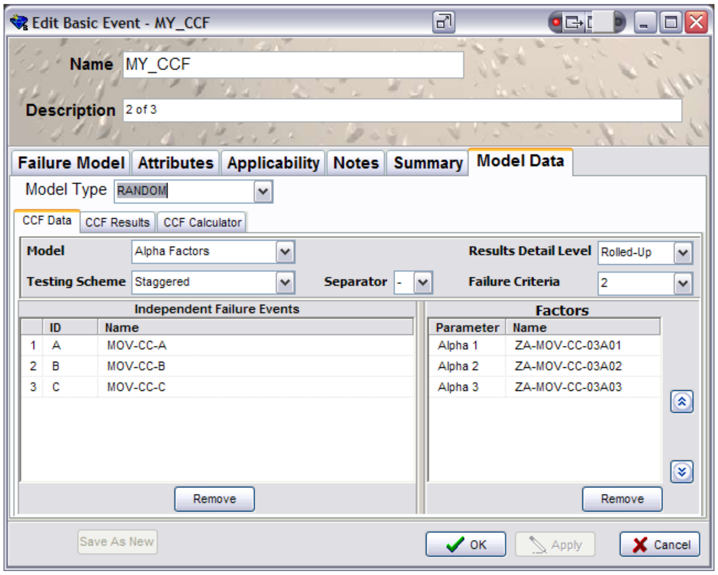
Figure 5. Editing the CCF example event.

To make a CCF object, its calculation type must be set to “Common Cause Failure – (R).” Once this calculation type is selected, the user can edit the parameters of the CCF object by clicking the “Edit” button or the “Model Data” tab shown in Figure 5. The Model Data for CCF objects appears as shown in Figure 6. For the example case, we need to specify the three Independent Failure Events (MOV-CC-A, MOV-CC-B, and MOV-CC-C) and the CCF Factors.

Since we selected the alpha-factor model, the “factors” asked by SAPHIRE are alpha$_1$, alpha$_2$, and alpha$_3$ (labeled Alpha 1, Alpha 2, and Alpha 3, respectively). These factors are other basic events that must already exist in the database.

For this example, we will assume a “staggered” testing scheme. As already indicated, the Failure Criteria is set to “2.”

The Separator character is used by SAPHIRE to determine the “auto generated” event names, as we will see when describing the results of the CCF calculation. Initially, the default results are given as “Rolled-up,” meaning the overall CCF probability is returned by this object. In this example, the overall CCF probability is 2.696E-5. However, we can see the details of this calculation by changing the “Results Detail Level” to “Full Detail.”



Name: MY_CCF

Description: 2 of 3

Model Type: RANDOM

Model: Alpha Factors

Results Detail Level: Rolled-Up

Testing Scheme: Staggered

Separator: -

Failure Criteria: 2

Independent Failure Events

ID		Name
1	A	MOV-CC-A
2	B	MOV-CC-B
3	C	MOV-CC-C

Factors

Parameter	Name
Alpha 1	ZA-MOV-CC-03A01
Alpha 2	ZA-MOV-CC-03A02
Alpha 3	ZA-MOV-CC-03A03

Figure 6. Editing the parameters of the CCF object.

When evaluating the “Full Detail” results for this example, we can see the results by clicking on the “CCF Results” tab. The detailed results appear as:

```
2 inputs out of 3 possible must fail - All independent only
groups are not counted.

# 1, 6.4500E-06, MY_CCF-AB

# 2, 6.4500E-06, MY_CCF-AC

# 3, 6.4500E-06, MY_CCF-BC

# 4, 7.6100E-06, MY_CCF-ABC

Q1 = 9.7950E-04

Q2 = 6.4500E-06

Q3 = 7.6100E-06

3 * Q2

1 * Q3

4 permutations.

2.6960E-05 total failure value.
```

If we evaluate these results in more detail, we see the following:

```
2 inputs out of 3 possible must fail - All independent only
groups are not counted.
```

This line above indicates that two of three components have to fail to have failure of this part of the fault tree. Also, that a term with ALL independent failures are not counted in the CCF probability.

Next, we see:

```
# 1, 6.4500E-06, MY_CCF-AB

# 2, 6.4500E-06, MY_CCF-AC

# 3, 6.4500E-06, MY_CCF-BC

# 4, 7.6100E-06, MY_CCF-ABC
```

These lines represent the "cut sets" included in the CCF calculation. The first three cut sets represent a CCF of two components, specifically AB or AC or BC. The last line represent CCF of all three components. Their respective probabilities are also listed, for example the probability of seeing A and B fail due to CCF is 6.45E-6. Also note the name of the CCF "events" listed here. For A and B failing due to CCF, SAPHIRE automatically creates the name "MY_CCF-AB" by appending the CCF basic event name with the separator character ("-") and then the identifier for the A and B components (A and B, respectively).

Next, we see:

```
Q1 = 9.7950E-04

Q2 = 6.4500E-06

Q3 = 7.6100E-06
```

These lines represent the Q_1, Q_2, and Q_3 terms in the CCF basic parameter model used by the calculation.

Next, we see:

```
3 * Q2

1 * Q3
```

These lines indicate that we see three combinations of Q_2 terms, and only one Q_3 term. Consequently, for this example, the full expression to determine the CCF probability is given by:

P(CCF) = 3 * Q_2 + Q_3 = 3 * 6.45E-6 + 7.61E-6 = 2.696E-5 .

Next, we see a line indicating how many CCF terms we are accounting for in the CCF calculation.

```
4 permutations.
```

Lastly, we see the CCF probability.

```
2.6960E-05 total failure value.
```

Note that if we set MY_CCF to have results "Rolled up" then only a basic event with probability of 2.696E-5 will appear in the fault tree cut sets, or:

PROB/FREQ	TOTAL%	CUT SET
2.696E-5	89.99	MY_CCF
1.000E-6	3.34	MOV-CC-A, MOV-CC-B
1.000E-6	3.34	MOV-CC-A, MOV-CC-C
1.000E-6	3.34	MOV-CC-B, MOV-CC-C
2.996E-5		

However, if we allow detailed results to appear, the results will show all of the combinations of the CCF terms, or:

PROB/FREQ	TOTAL%	CUT SET
7.610E-6	25.40	MY_CCF-ABC
6.450E-6	21.53	MY_CCF-AB
6.450E-6	21.53	MY_CCF-AC
6.450E-6	21.53	MY_CCF-BC
1.000E-6	3.34	MOV-CC-A, MOV-CC-B
1.000E-6	3.34	MOV-CC-A, MOV-CC-C
1.000E-6	3.34	MOV-CC-B, MOV-CC-C
2.996E-5		

By having all of the detail from the CCF, the fault tree calculation is equivalent to having the fault tree logic as shown in Figure 7 and Figure 8.

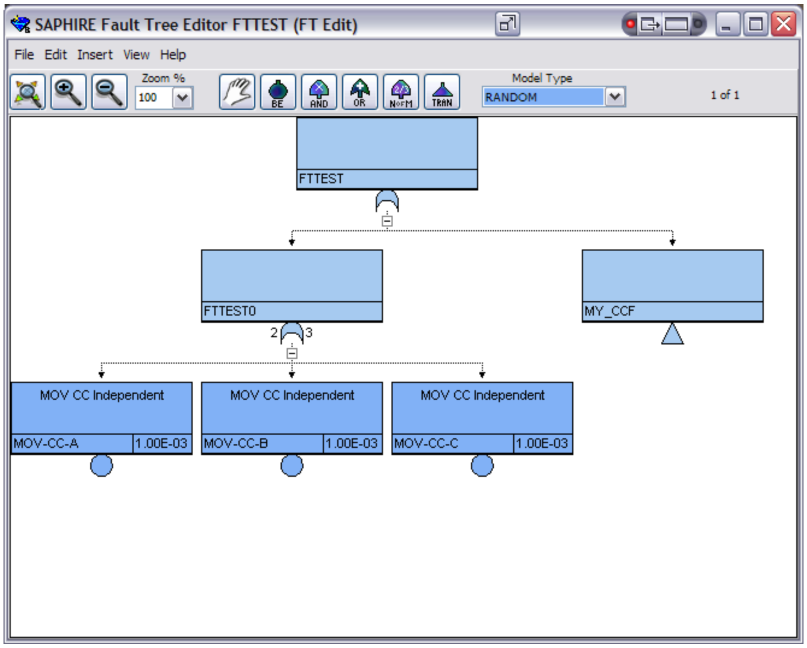


Figure 7. Fault tree representation of the CCF module.

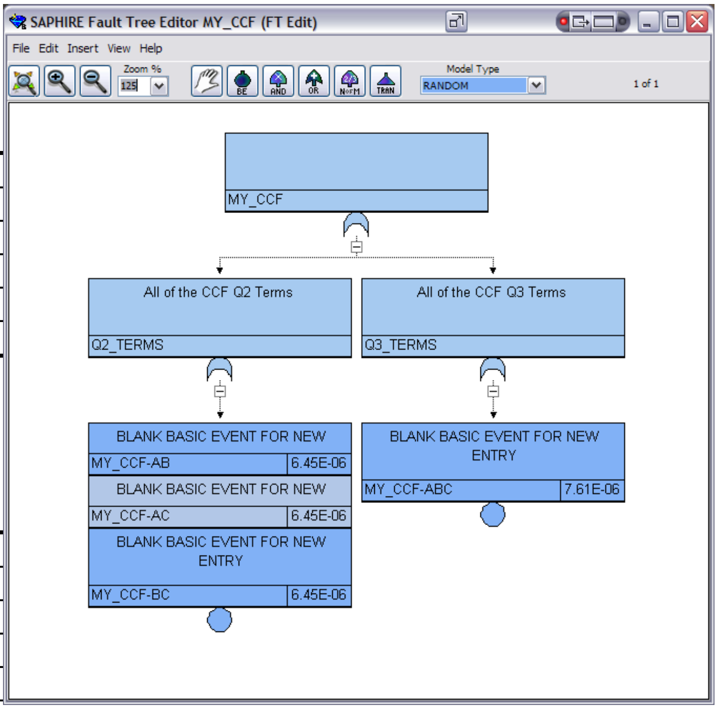


Figure 8. Representation of the CCF terms.

CONCLUSIONS

Changes to SAPHIRE have been required to support the NRC CCF analysis guidance for events assessment. Based upon user-provided input and limitations in the old versions of SAPHIRE, the CCF module calculations have been updated as outlined in this paper.

The modifications to SAPHIRE 8 involved changes to (1) the SAPHIRE graphical user interface directing how end-users and modelers interface with PRA models and (2) algorithmic changes as required. Included in the modifications are the possibility to treat CCF probability adjustments based upon failure types (e.g., independent versus dependent) and failure modes (e.g., failure-to-run versus failure-to-start). Reporting capability has been augmented, including reports outlining the algorithms to be used and basic event changes.

The extensions to the SAPHIRE 8 design related to CCF analysis have accomplished:

- Transition the old SAPHIRE 7 approach to a new CCF construction approach
- Modified the SAPHIRE solution to improve the CCF adjustments
- Facilitate enhancements in the existing CCF methods

ACKNOWLEDGMENTS

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights. The views expressed in this report are not necessarily those of the U.S. Nuclear Regulatory Commission.

REFERENCES

Mosleh, A., et al., 1998, *Guidelines on Modeling Common-Cause Failures in Probabilistic Risk Assessment*, NUREG/CR-5485.

Rasmuson, D. M. and Kelly, D. L., 2007, *Common-Cause Failure Analysis in Event Assessment*, U. S. Nuclear Regulatory Commission.

Smith, Curtis L., 1998, "Calculating conditional core damage probabilities for nuclear power plant operations" *Reliability Engineering and System Safety*, 59, 299-307