

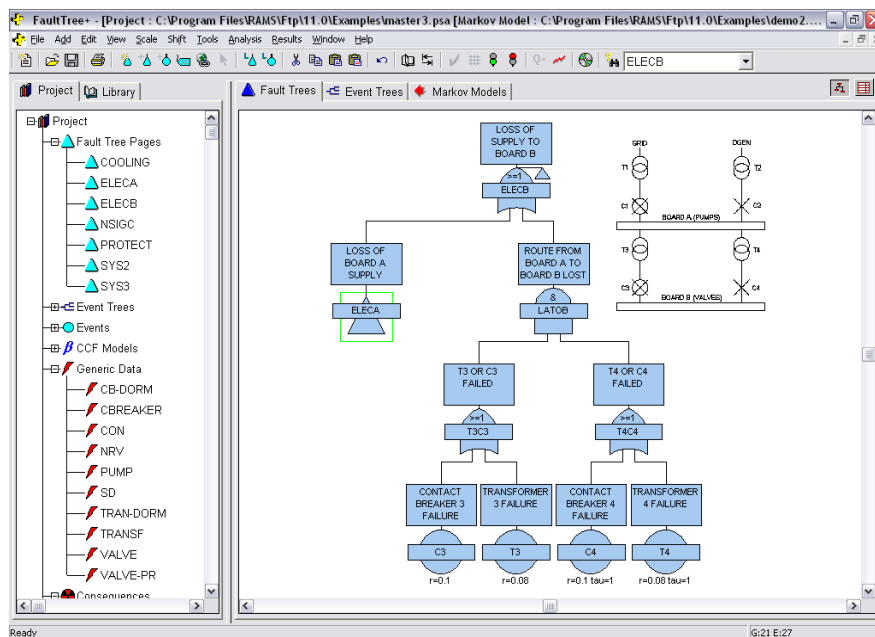
## FaultTree+ V11 Technical Specification

### Platform

Runs under Windows 95, 98, NT, 2000, Me and Xp. Recommended host memory requirements 128Mb+

### Summary of capabilities :

- Automatic drawing facilities produce high quality diagrams without any effort from the user
- Time saving features such as double mouse clicks to bring up gate, event and branch attributes
- Drag and drop add mode for fast tree construction
- Tree control for easy project navigation
- Direct link with IsoLib Parts Libraries
- Extensive diagram scale and shift options including manual shifting of sub-trees and automatic alignment to the screen edit area
- Global and local font selection allowing highlighting of labels and descriptions
- Automatic paging facilities - simply identify gates or branches with a new page tag and the program takes care of pagination
- Append (single or multiple project) facilities for fault trees produced by different users
- Hyperlinks available for gates, events and models
- OR, AND, VOTE, NOT, Exclusive Or, Inhibit and Priority AND gates supported
- Basic, Conditional, Undeveloped, Dormant and House basic event symbols supported
- Multiple branching supported for event trees
- Multiple consequence categories for event trees
- Grid controls for easy data entry
- Integrated fault and event trees allowing full risk analyses to be performed
- Primary and secondary event trees
- Extensive on-line help facility including key word search



Fault Tree Module Screen Shot

- Attributes such as event parameters, generic model codes, branch names and column probabilities may be displayed on diagrams if required
- Generic parameters (such as failure rate and inspection interval) may be attached to an event's local data model
- Disjoint events may be specified
- Cut, copy and paste facilities for fault and event trees
- Library facility for storing common fault and event tree structures
- Flexible labelling formatting allows the user to place descriptive text anywhere within a fault or event tree page
- Project database tables may be easily edited using direct and dependency filtering
- Event and gate names may be globally edited
- Circular logic checks during fault tree construction
- Undo and automatic backup facilities
- Delete hidden data facility for tidying-up large projects
- Comprehensive range of event failure and repair models including fixed rates, dormant, time at risk, binomial, Weibull, Poisson, sequential, standby and initiator failure models
- Fixed-Phased and Rate-Phased models allow different failure data to be specified for different phases of operation
- User-created Markov models for handling dependencies between events
- Event and generic failure model grouping
- Fault tree house event analysis
- Full minimal cut set analysis (including success states if required)
- CCF analysis using the beta factor, MGL, alpha factor or beta BFR methods
- IEC61508 method for CCF beta determination
- CCF's may be attached to a specified group of events with different failure data
- Post-processing facilities for accurate upper bound calculations
- Importance analysis with Fussell-Vesely (both failure and success) , Birnbaum, Barlow-Proshan and Sequential importance measures. Risk importance measures provided for event tree consequences
- Initiator-enabler analysis for sequence dependent analyses
- Uncertainty analyses allowing confidence levels to be determined from event failure and repair data uncertainties
- Sensitivity analysis allowing the automatic variation of event failure and repair data between specified limits
- Time dependent analysis providing intermediate values for time dependent system parameters
- Verification checks providing diagnostic information before commencing an analysis. Checks are made for circular logic, undefined gates, invalid initiators etc.
- Cut set tracing in fault tree diagrams
- Display the cut-sets that contribute to Risk
- Status facility to indicate whether analysis results are out of date with respect to project data
- Partial analysis for individual areas of a project
- Analyse a number of projects and compare results between them
- Incorporate custom bitmap pictures for diagram enhancement
- Customisable reports interfacing with Microsoft Office products
- Graphs, plots, pie charts and time profile histograms
- Import and export facilities
- Interfaces with other reliability products such as AvSim+

## Overview

The FaultTree+ program is a powerful systems reliability analysis tool that allows fault and event tree analyses to be performed in an integrated environment. Customised Markov models may also be linked to events in the fault or event tree diagram. The program may also be used to analyse fault trees, event trees and Markov models independently.

The program runs under Microsoft Windows and is capable of analysing large and complex fault and event trees producing the full minimal cut representation for fault tree TOP events and event tree consequences.

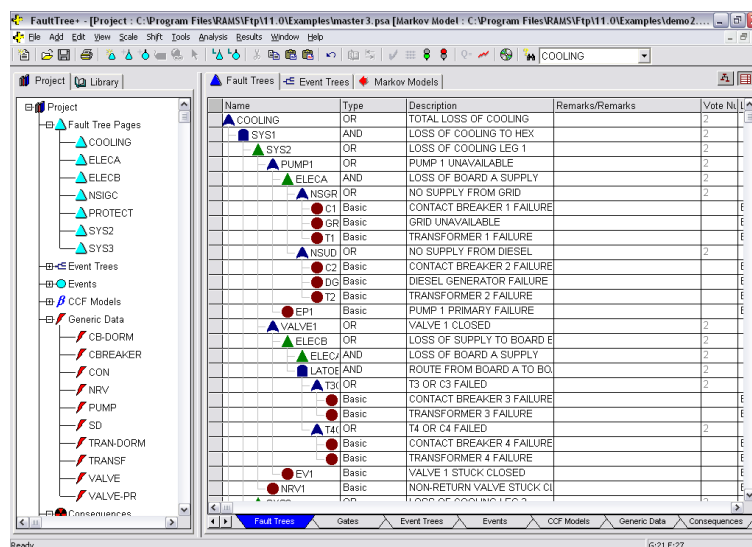
FaultTree+ provides CCF analysis, importance analysis, uncertainty and sensitivity analysis facilities. The program allows users to construct a single project database containing generic data and event tables, fault trees with multiple TOP events, event trees originating from different initiating events, CCF tables and consequence tables. Failure data can be input using many different failure models, including multi-phase models, and can also be directly imported from the IsoLib Parts Libraries. Fault and event tree pagination is automatically controlled by the program. Fault tree TOP events may be used to represent specific columns in the event tree. Multiple branches are also handled to allow for partial failures. Users may feed the end branches of event trees into secondary event trees eliminating the need for the user to reproduce identical event tree structures leading to identical consequences. Hyperlinks can be added to all gates and events.

FaultTree+ uses efficient minimal cut set generation algorithms to analyse large and complex fault and event trees. Partial analysis is possible for individual areas of the fault tree structure. NOT logic may be included in the fault and event trees at any level and the event success states retained in the analysis results as an option.

The FaultTree+ Report Generator allows you to select from a range of standard reports or quickly design your own customised reports. You can design your own headers and footers, choose your own fonts, insert your own pictures, sort and filter data and much more !

Paginated network or fault tree diagram reports are automatically produced and can be transferred to other packages such as Microsoft Word. You may specify the pagination scheme you require for diagram reports and obtain page index reports to allow you to easily find specific gates and events.

You may also choose from a wide range of sophisticated scientific graphs and charts or create your own graphs and charts. You can display multiple graphs on the same page and easily modify scales, legends, titles etc.



*Editing Fault Tree Data using the Grid Control*

FaultTree+ provides a flexible import/export facility that allows the user to transfer data to and from Microsoft Access databases, Microsoft Excel spreadsheets and text delimited and fixed length files.

FaultTree+ has been used to perform systems reliability analysis by a wide range of different industries for over a decade. We hope you enjoy using FaultTree+. Remember that full support and training facilities are available with the program.

### **Fault and Event Tree Construction**

FaultTree+ provides a fully interactive environment for constructing and editing fault and event trees.

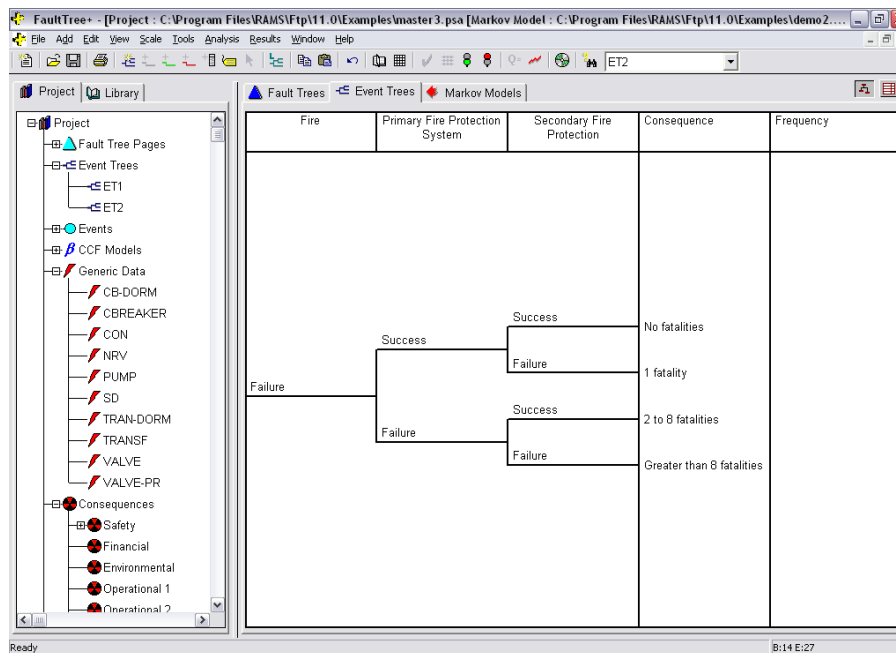
Features include:-

- Automatic drawing facilities produce high quality diagrams without any effort from the user
- Time saving features such as double mouse clicks to bring up gate, event and branch attributes
- Drag and drop add mode for fast tree construction
- Extensive diagram scale and shift options including manual shifting of sub-trees and automatic alignment to the screen edit area
- Global and local font selection allowing highlighting of labels and descriptions
- Automatic paging facilities - simply identify gates or branches with a new page tag and the program takes care of pagination
- Append (single or multiple projects) facilities for fault trees produced by different users
- OR, AND, VOTE, NOT, Exclusive Or, Inhibit and Priority AND gates supported
- Basic, Conditional, Undeveloped, Dormant and House basic event symbols supported
- Disjoint events can be specified
- Multiple branching supported for event trees
- Extensive on-line help facility including key word search
- Attributes such as event parameters, generic model codes, branch names and column probabilities may be displayed on diagrams if required
- Library facility for storing fault and event tree structures
- Cut, copy and paste facilities available for fault tree symbols
- Flexible labelling formatting allows the user to place descriptive text anywhere within a fault or event tree page
- Project database tables may be easily edited using direct and dependency filtering
- Event and gate names may be globally edited
- Generic Parameters (such as failure rate and inspection interval) may be attached to an event's local data model
- Hyperlinks available for gates, events and models
- Direct link with IsoLib Parts Libraries
- Circular logic checks during fault tree construction
- Import and export facilities for data models
- Customize the colour of gates and events
- Undo and automatic backup facilities
- Delete hidden data facility for tidying-up large projects

### **Fault and Event Tree Analysis**

- Range of event failure and repair models including fixed rates, dormant, sequential, initiator, standby, binomial, Weibull and Poisson failure models
- Fixed-Phased and Rate-Phased models allow different failure data to be specified for different phases of operation
- Fault tree house event analysis
- Full minimal cut set analysis (including success states if required)
- CCF analysis using the beta factor, MGL, alpha factor or beta BFR methods
- IEC61508 method for CCF beta determination
- Post-processing facilities for accurate upper bound calculations

- Importance analysis with Fussell-Vesely (failure and success), Birnbaum, Barlow-Proschan and Sequential importance measures. Weighted values provided for event tree consequences
- Initiator-enabler analysis for sequence dependent analyses
- Uncertainty analyses allowing confidence levels to be determined from event failure and repair data uncertainties
- Sensitivity analysis allowing the automatic variation of event failure and repair data between specified limits
- Time dependent analysis providing intermediate values for time dependent system parameters
- Verification checks providing diagnostic information before commencing an analysis. Checks are made for circular logic, undefined gates, invalid initiators etc.
- Selected parts of a project can now be analyzed individually
- Batch analysis possible for a number of projects and their results compared
- Status facility to indicate whether analysis results are out of date with respect to project data



*Event Tree Module Screen Shot*

### **Markov Analysis**

Then Markov module of Reliability Workbench analyses state transition diagrams using numerical integration techniques. The module provides facilities for defining multiple phases representing continuous or discrete transitions. The program also analyses non-homogeneous processes by allowing time-dependent transition rates to be defined. Systems with time-dependent transition rates are strictly non-Markovian, however the addition of this facility in the program allows certain types of ageing processes to be modelled.

The Markov module uses 4th order Runge-Kutta numerical integration techniques to analyse the Markov diagram. The system logic is represented by a state transition diagram that may be easily constructed using the program's interactive graphics facilities. The system lifetime may be split into phases with different transition rates.

Facilities provided by the module are summarised below :

- Graphically constructed transition diagram
- Division of analysis into separate phases
- State attribute editing via easy-to-use dialogs

- Data verification for consistency checks
- Time-dependent transition rates modelled
- Global parameter facility for repetitive data
- Calculation of a wide range of probabilities and frequencies
- Comprehensive reports interfacing with Microsoft Word, Excel etc.
- Graphs and plots showing time-dependent results

### Markov Analysis Methods

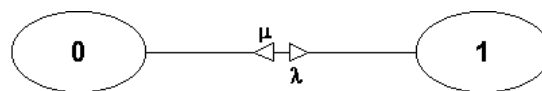
Markov analysis provides a means of analysing the reliability and availability of systems whose components exhibit strong dependencies. Other systems analysis methods (such as the Kinetic Tree Theory method employed in fault tree analyses) generally assume component independence which may lead to optimistic predictions for the system availability and reliability parameters. Some typical dependencies that can be handled using Markov models are

- Components in cold or warm standby
- Common maintenance personnel
- Common spares with a limited on-site stock

The major drawback of Markov methods is that Markov diagrams for large systems are generally exceedingly large and complicated and difficult to construct. However, Markov models may be used to analyse smaller systems with strong dependencies requiring accurate evaluation. Other analysis techniques, such as fault tree analysis, may be used to evaluate large systems using simpler probabilistic calculation techniques. Large systems that exhibit strong component dependencies in isolated and critical parts of the system may be analysed using a combination of Markov analysis and simpler quantitative models.

The state transition diagram identifies all the discrete states of the system and the possible transitions between those states. In a Markov process the transition frequencies between states depends only on the current state probabilities and the constant transition rates between states. In this way the Markov model does not need to know about the history of how the state probabilities have evolved in time in order to calculate future state probabilities. Although a true Markovian process would only consider constant transition rates the Markov module does allow time-varying transition rates to be defined. These time-varying rates must be defined with respect to absolute time or phase time (the time elapsed since the beginning of the current phase).

In order to illustrate the use of Markov methods let us consider a very simple Markov model. The Markov diagram below represents the failure and repair behaviour of a single component.



The component has two states only, the working state (State 0) and the failed state (State 1). It is a repairable component (with failures immediately revealed) and therefore the component may move from the failed state to the working state as well as moving from the working to failed state. These possible transitions are represented by the transition lines and arrows in the Markov diagram.

The Markov diagram represents the logical behaviour of a component or system and should contain all possible states and transitions for the component or system under given conditions.

The Markov diagram above may be translated into a set of linear differential equations that represent the time-dependent behaviour of the state probabilities. These equations are given below.

$$\frac{dP_0(t)}{dt} = -\lambda P_0(t) + \mu P_1(t)$$

$$\frac{dP_1(t)}{dt} = \lambda P_0(t) - \mu P_1(t)$$

where  $P_i(t)$  = probability of being in state  $i$  at time  $t$   
 $\lambda$  = component failure rate  
 $\mu$  = component repair rate

Integration of these equations after applying the initial conditions

$$P_0(0) = 1$$

$$P_1(0) = 0$$

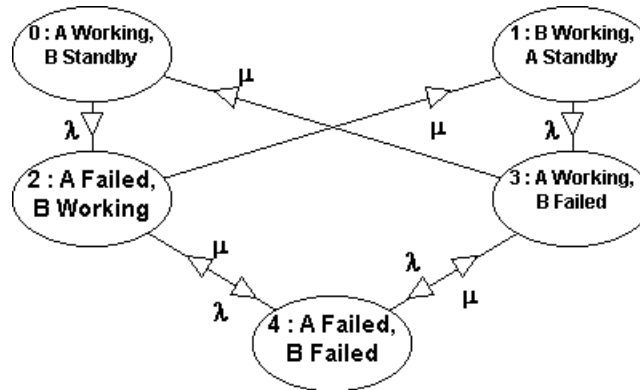
produces the well-known expression for the unavailability of a two-state repairable component with immediately revealed failures :

$$P_1(t) = \frac{\lambda}{\lambda + \mu} (1 - e^{-(\lambda + \mu)t})$$

As  $t$  becomes very large the component unavailability approaches the steady state solution of

$$P_1(\infty) = \frac{\lambda}{\lambda + \mu}$$

The Markov diagram below represents the failure and repair behaviour of a 2-pump standby system. The diagram assumes that the pumps are identical and that there is no possibility of a pump failing if it is in standby (cold standby).



Only one pump is required to be working at any time to provide full functionality. If the operating pump should fail, the standby pump will be started and the failed pump will be repaired. There is therefore a dependence between the two pumps.

Even for this small system of two components it can be seen that the number of states in the Markov model is rapidly increasing. The steady-state solution for the unavailability of the two-component system is equal to the steady-state probability for state 4 :

$$P_4 = \frac{\lambda^2}{\lambda^2 + 2\lambda\mu + 2\mu^2}$$

As the size of the Markov diagram increases the task of evaluating the expressions for time-dependent unavailability by hand becomes impractical. Computerised numerical methods may be employed, however, to provide a fast solution to large and complicated Markov systems. In addition these

numerical methods may be extended to allow the modelling of phased behaviour and time-dependent transition rates. MKV employs a Runge-Kutta 4th order numerical integration technique to determine the time-dependent behaviour of state probabilities. The time step employed during the integration may be specified by the user. MKV also provides three different error indicators to allow the user to assess the accuracy of the result.

### Continuous Time and Discrete Transition Phases

MKV allows the user to split the system lifetime into discrete fixed-interval phases. Each phase may be represented by a set of transitions unique to that particular phase. States may not vary between phases. Phases may be specified as *continuous time* phases or *discrete transition* phases. Continuous time phases have transitions that are quantified with transition rates. Transition rates are generally failure and repair rates. Continuous time phases have finite phase durations. Discrete phases do not have a phase duration associated with them as they represent fixed probability transitions between states. They may be used to represent fixed interval inspections and preventive maintenance actions. The transitions in a discrete phase must be identified with fixed probabilities.

For continuous time phases the user may specify transition rates that vary with absolute system time or absolute phase time. The time-varying transition rates are specified in the form of a Weibull distribution which is superimposed on the base failure rate :

$$\lambda(t) = \lambda_0 + \frac{\beta(t - \gamma)^{\beta-1}}{\eta^\beta}$$

where  $\lambda_0$  = base failure rate  
 $\eta$  = Weibull characteristic lifetime  
 $\beta$  = Weibull shape parameter  
 $\gamma$  = Weibull location parameter

### Calculated Parameters

MKV calculates a wide range of system parameters during the integration process. These parameters are

- Unavailability
- Availability
- Unreliability
- Reliability
- Failure frequency (unconditional failure intensity)
- Repair frequency (unconditional repair intensity)
- Conditional failure intensity
- Conditional repair intensity
- Number of expected failures
- Number of expected repairs
- Mean unavailability over lifetime
- Mean availability over lifetime
- Expected total downtime over lifetime
- Expected total uptime over lifetime

MKV also calculates mean and lifetime probabilities for states in the transition diagram.

## **New Features for Version 11**

### **New Fault and Event Tree Library Facility**

The new fault and event tree library facility allows users to connect a library and copy fault and event tree structures from the library to the current project. Fault tree structures may be dragged from a library tree control in the left-hand window to the appropriate gate in the project fault tree in the right-hand window. Other library elements such as event and model groups may also be copied from a library into the current project. Project fault and event tree structures, as well as individual elements, may be saved to the currently connected library. The structure of library files is similar to the structure of project files (a project may be connected to another project as a library) and the new facility offers an alternative method for appending part of one project to another.

### **Multiple Project Append**

A new append function has been added to allow users to append data from a group of projects all in one go. If there is a conflict in data definition (e.g. a gate has the same name in two projects but different inputs), priority is given to the first project in the list. Users may define a list of projects to be appended and save this list in a special template file.

### **Generic Parameters**

Users may now create 'generic parameters' as well as 'generic models'. Generic parameters (such as failure rate and inspection interval) may be attached to an event's local data model. This allows a parameter's value to be modified for a group of events in one go without affecting other parameters, such as failure rate, that may vary between events.

### **CCF Model Extended to Handle Events with Different Failure Data**

A single CCF model may now be assigned to a group of events with different failure models or parameters assigned to them. Previously a CCF model would apply the same failure model to all events in the CCF group. Now, the program applies the CCF using the failure data assigned to each individual event. If the event failure models are different for the same CCF group the program will use the maximum, mean or minimum total event probability to calculate the CCF probability values. Users may select which method to adopt in the 'Project Options' Dialog (Sets Generation Tab).

### **Calculation of Independent Unavailability for CCF Models**

Users may now optionally choose to take the total unavailability of an event as the independent unavailability value.

Calculation of the independent unavailability and CCF unavailability values with the 'Adjust Independent Q' method set on in the 'Project Options' Dialog (Sets Generation Tab) :

$$Q_I = (1 - \beta) \cdot Q_T$$

$$Q_{CCF} = \beta \cdot Q_T$$

where  $\beta$  = beta factor  
 $Q_I$  = independent unavailability  
 $Q_T$  = total unavailability  
 $Q_{CCF}$  = unavailability due to CCF

Calculation of the independent unavailability and CCF unavailability values with the 'Adjust Independent Q' method set off in the 'Project Options' Dialog (Sets Generation Tab) :

$$Q_I = Q_T$$

$$Q_{CCF} = \beta \cdot Q_T$$

**IEC 61508 Method for CCF Beta Determination**

The IEC 61508 method for determining CCF beta values has been added (IEC 61508-6 Annex D). This facility may be accessed from the ‘Add CCF’ and ‘Edit CCF’ Dialogs. Users make a number of selections in response to questions and the program automatically calculates the beta factor for the single parameter beta method.

**Hyperlinks**

Users may now assign hyperlinks to gates, events and models. Hyperlinks may be made active in the fault and event tree diagrams by setting the appropriate view option (accessed via the ‘View’ pull-down menu).

**Sequential Calculation Model Extended**

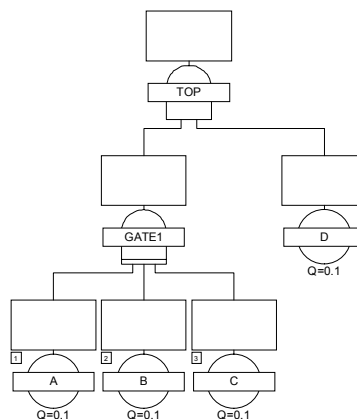
The event sequential calculation model has been extended to apply to unavailability calculations as well as frequency calculations. Individual events may be assigned a position of first, second, third, fourth, fifth or last in a sequence. The position indicates the allowable position for the event in a time sequence. The program will adjust the calculated unavailability and frequency values for cut sets containing events with a sequence assignment. The sequence restrictions will be calculated based on the number of events in a cut set sequence. Modular gates in a fault tree will affect the results of sequence calculations in some circumstances. Users may wish to set the ‘Always Modularise’ flag on for a gate to affect sequence calculations. For example, consider the fault tree illustrated below. Events A, B and C must occur in sequence (A first, B second and C third) for the event represented by GATE1 to occur. If GATE1 is modularised in the analysis (users may force a gate to be modularised using the ‘Always Modularise’ flag in the ‘Edit Gate’ Dialog) then the TOP gate will be represented by a single cut set GATE1\*.D (GATE1\* is the super event for GATE1). As the super event GATE1\* and D1 are not sequence dependent this implies that the following sequences are allowable :

- A->B->C->D
- D->A->B->C

If GATE1 was not modularise during the analysis we would obtain the cut set A.B.C.D for the TOP gate. As the events A, B, and C must occur in positions 1, 2 and 3 respectively in a cut set, only one sequence is permitted :

- A->B->C->D

The two cases will lead to different results for the predicted unavailability and frequency of the TOP gate.



### Weibull Model Added

A Weibull distribution model has been added to the list of standard failure models that may be assigned to an event. The Weibull model requires users to specify three parameters that define the basic Weibull distribution :

Characteristic Lifetime  
 Shape Parameter  
 Location Parameter

In addition, users may specify a standard deviation or error factor if confidence analysis is to be performed. The Weibull model may be used to represent a component whose failure rate varies over the lifetime of the analysis. The Weibull failure rate is given by

$$r(t) = \frac{\beta(t - \gamma)^{\beta-1}}{\eta^\beta}$$

where  $r(t)$  is the failure rate  
 $\eta$  = characteristic life parameter  
 $\beta$  = shape parameter  
 $\gamma$  = location parameter

The unreliability,  $F(t)$ , is given by

$$F(t) = 1 - \exp\left[-\left(\frac{t - \gamma}{\eta}\right)^\beta\right]$$

This is equivalent to a failure density function,  $f(t)$ , given by

$$f(t) = \frac{\beta(t - \gamma)^{\beta-1}}{\eta^\beta} \exp\left[-\left(\frac{t - \gamma}{\eta}\right)^\beta\right]$$

and a mean time to failure (MTTF) given by

$$MTTF = \eta \Gamma\left(\frac{1 + \beta}{\beta}\right)$$

where  $\Gamma$  = gamma function

### Multiple Event Groups

Events may now be associated with up to 16 different event groups. For example, an event 'ELECTRIC PUMP FAILURE' may be associated with a system 'PRIMARY COOLING' as well as being assigned to a component class 'PUMPS'. Event groups 'PRIMARY COOLING' and 'PUMPS' may be defined and the event 'ELECTRIC PUMP FAILURE' assigned to both groups. One of the advantages of associating multiple event groups with a single event is that finding an event in the tree control becomes easier. Events may be located by more than one event group. Multiple event groups also allow the user to take advantage of the new 'disjoint' event analysis function and the new group importance ranking facility described below.

### **Disjoint Events**

Disjoint (exclusive) events are events that cannot occur together at the same time. Examples of disjoint events are 'valve failed open' and 'valve failed closed' (exclusive failure modes for the same component) or 'night' and 'day'. Disjoint events should be used as an alternative to adding NOT gates into a fault tree as the cut set calculations will be more efficient. To define a group of disjoint events (there may be two or more in a group), simply define an event group and 'drag and drop' the disjoint events into the group using the tree control. Alternatively, assign each event to the disjoint group using the 'Edit Event' Dialog. You will need to set the 'Disjoint Event Group' flag in the 'Event Group Definition' Dialog. The program will then remove any cut sets that contain two or more events in the same disjoint event group during the analysis.

### **Event Group Importance Rankings**

Event group importance rankings are now produced for any event groups defined in the project. To view event group importance rankings in the 'Results Summary' Dialog simply select the 'Group Importance' flag. Group importance values may be viewed in reports. You may wish to filter certain categories of group importance rankings in reports. For example, you may wish to view importance rankings only associated with location groups. Users may now assign a category to an event group to allow such filter operations in reports. You may assign one of 10 event group categories to a single event group. This may be done in the 'Event Group Definition' Dialog. You may also customise the event group category titles. To do this, select the 'Customise Event Group Categories' option on the 'Tools' pull-down menu. A dialog will appear allowing you to specify the new titles for each of the 10 categories.

### **Link to IsoLib Generic Libraries**

Generic failure data may now be extracted from Isograph's new parts library (IsoLib). In order to access this data you will need to purchase a licence for the IsoLib library facility. IsoLib provides generic data such as the NPRD parts library. Failure data may be inserted into a project by selecting the appropriate parts using IsoLib and then selecting the 'Tools, Insert Data from the Isograph Parts Library' pull-down menu option. Data may be inserted into the 'Generic Parameter', 'Generic Model' or 'Event' tables.

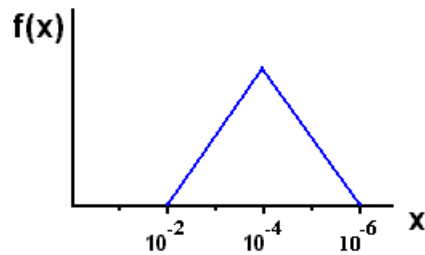
### **Dependent Sampling for Confidence Analysis**

Users may now request that dependent or independent sampling be used for events during a confidence analysis. In version 10.1, event failure and repair parameters were sampled independently even if they were associated with the same generic model. Now users may choose in the 'Project Options' Dialog whether or not to sample independently (Confidence Analysis Tab).

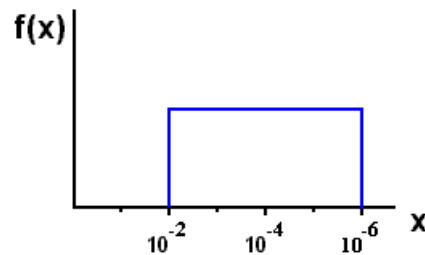
### **New Distributions for Confidence Analysis**

Two new distributions have been added for uncertainty data. These distributions are the log-triangular and log-uniform distributions.

The log-triangular distribution defines a possible range of the reliability parameter on a log scale as illustrated below. The probability density function,  $f(x)$ , indicates the probability that the value of the parameter falls between  $x$  and  $x + dx$ . The value of the parameter specified by the user is the mid-point of the parameter range on a log scale. Dividing or multiplying the mid-point value by the error factor gives the minimum and maximum parameter values. The example below corresponds to a parameter value (for example, failure rate) of  $10^{-4}$  and error factor of 100.



The log-uniform distribution defines a possible range of the reliability parameter on a log scale as illustrated below. The probability density function,  $f(x)$ , indicates the probability that the value of the parameter falls between  $x$  and  $x + dx$ . The value of the parameter specified by the user is the mid-point of the parameter range on a log scale. Dividing or multiplying the mid-point value by the error factor gives the minimum and maximum parameter values. The example below corresponds to a parameter value (for example, failure rate) of  $10^{-4}$  and error factor of 100.

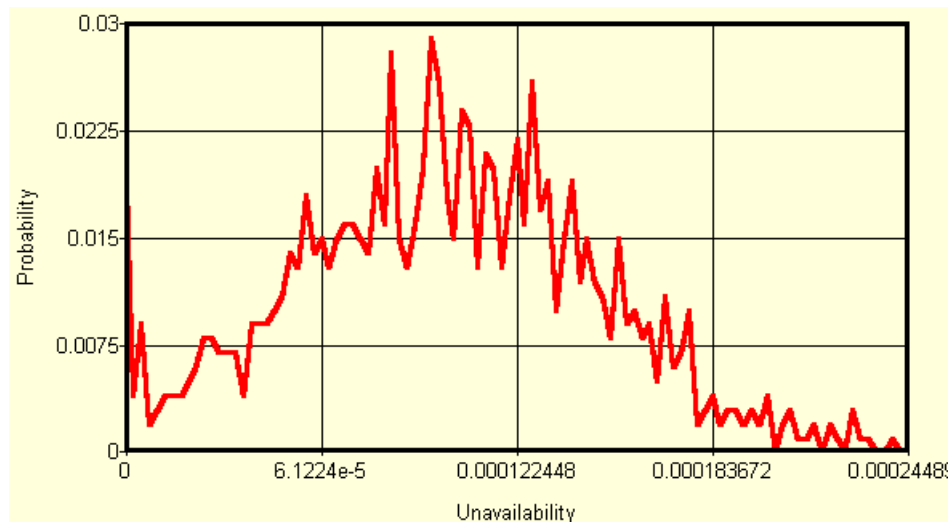


#### Upper and Lower Bounds for Confidence Analysis

Users may now request the program to present confidence analysis results as double-sided upper and lower bounds of the predicted parameter rather than just a single-sided upper bound value. The required option may be set in the 'Project Options' Dialog (Confidence Analysis Tab).

#### Generalised Distribution for Confidence Analysis

Version 10.1 assumed that the predicted system parameter variations conformed to a normal distribution. In version 11, the user can instruct the program to calculate upper and lower bound values using a generalised distribution. This is done by selecting the 'Generalised Distribution for Results' flag in the 'Project Options' Dialog (Confidence Analysis Tab). If this option is selected the program will store the predicted parameter (e.g. system unavailability) for each individual simulation. A histogram representing the probability density function for the predicted parameter value will be constructed (and may be viewed by the user, when an analysis is completed, as a confidence distribution graph). The program will numerically integrate the area under the distribution curve to determine the upper and lower bounds of the parameter. Using a generalised distribution requires more computer time than assuming a normal distribution but provides more accurate results for confidence analysis where the confidence distribution for the predicted parameter is skewed.



*Generalise Distribution Plot for Predicted Unavailability*

### Confidence Correlation Coefficients

Correlation coefficients are now calculated for the first parameter (usually failure rate or unavailability) of generic models if the ‘Generalised Distribution for Results’ flag has been selected and parameters are sampled with ‘Independent Sampling for Generic Data’ set off.

Confidence correlation coefficients indicate how much a generic model influences the uncertainty in the predicted probability or risk for the system. The correlation coefficient for parameter  $\lambda_k$  is given by

$$\theta(\lambda_k) = \frac{\sum_{j=1}^n (\lambda_{k,j} - \bar{\lambda}_k)(Q_j - \bar{Q})}{\sqrt{\sum_{j=1}^n (\lambda_{k,j} - \bar{\lambda}_k)^2 \cdot \sum_{j=1}^n (Q_j - \bar{Q})^2}}$$

where  $\theta(\lambda_k)$  = correlation coefficient for  $\lambda_k$

$\lambda_{k,j}$  = sample for parameter k during simulation k

$\bar{\lambda}_k$  = mean value for  $\lambda_k$  over n simulations

$Q_j$  = predicted probability or risk during simulation k

$\bar{Q}$  = mean value for the predicted probability or risk over n simulations

### Partial Analysis

Users may now request the program to analyse selected parts of a project rather than the whole project at once. This facility reduces the computing time for large and complex projects when the user is only interested in the results for part of a fault tree, a single event tree, a group of event trees or a specific group of consequences. Before starting a partial analysis, the user must set the ‘Include in Partial Analysis’ flag on for gates (in the ‘Edit Gate’ Dialog), event trees (in the ‘Edit Branch’ Dialog) or consequences (in the ‘Edit Consequence’ Dialog).

A partial analysis is initiated by selecting the 'Analysis, Perform Partial Analysis' pull-down menu option. Gates and event tree sequences that are not included in a partial analysis run will be labelled '<Not calculated>' where appropriate.

All the partial analysis flags in a project may be removed by selecting the 'Clear Partial Analysis Flags' option on the 'Analysis' pull-down menu.

### Batch Analysis

The new batch analysis facility enables users to define a group of fault tree project files that are to be analysed one after another without any interaction by the user. Once the analyses are completed it is possible to compare summary results from the different projects. This is a useful facility if you are comparing predicted parameters for slight design variations of the same system. To perform a batch analysis select the 'Analysis, Perform Batch Analysis' pull-down menu option. A dialog will appear allowing you to define the projects to be analysed. The dialog also contains buttons enabling you to start a full or partial analysis and compare results once the analyses have been completed.

### Copy and Paste Event Tree Structures

Users may now select whether to copy across column data when copying and pasting event tree structures to new columns in the target event tree.

### Fussell-Vesely Failure and Success Importance Measures

FaultTree+ V11 now computes Fussell-Vesely Failure and Success Importance Measures in addition to the Standard Fussell-Vesely measure.

The Standard Fussell-Vesely Importance Measure is given by

$$I_i^{FV} = \frac{Q_{SYS} - Q_{SYS}(q_i = 0)}{Q_{SYS}}$$

where  $I_i^{FV}$  = Fussell-Vesely importance for event i

$Q_{SYS}$  = system probability or risk

$Q_{SYS}(q_i = 0)$  = system probability or risk with the probability of event i set to 0

The Failure Importance Measure is determined by considering the failure and success states of the event to be independent :

$$I_{fi}^{FV} = \frac{Q_{SYS} - Q_{SYS}(q_{fi} = 0)}{Q_{SYS}}$$

where  $I_{fi}^{FV}$  = Fussell-Vesely failure importance for event i

$Q_{SYS}$  = system probability or risk

$Q_{SYS}(q_{fi} = 0)$  = system probability or risk with the probability of event i set to 0 where the event occurs in its failure state only

The Success Importance Measure is calculated from

$$I_{si}^{FV} = \frac{Q_{SYS} - Q_{SYS}(q_{si} = 1)}{Q_{SYS}}$$

where  $I_{si}^{FV}$  = Fussell-Vesely failure importance for event i

$Q_{SYS}$  = system probability or risk

$Q_{SYS}(q_{si} = 1)$  = system probability or risk with the probability of event  $i$  set to 1 where the event occurs in its success state only

### Importance Calculation Method Extended

In version 10.1 of FaultTree+ importance measures were calculated using the rare approximation method irrespective of the 'Quantitative Calculation Method' used to calculate system probability values. Now users may request the program to use the same method as used to calculate system probability values (Rare, Esary-Proschan or Optimum Upper Bound). This may be done in the 'Project Options' Dialog (Calculation Tab). Note that the rare approximation method may be substantially quicker than other methods when large numbers of minimal cut sets are being processed.

### Risk Cut Sets in the Results Summary Dialog

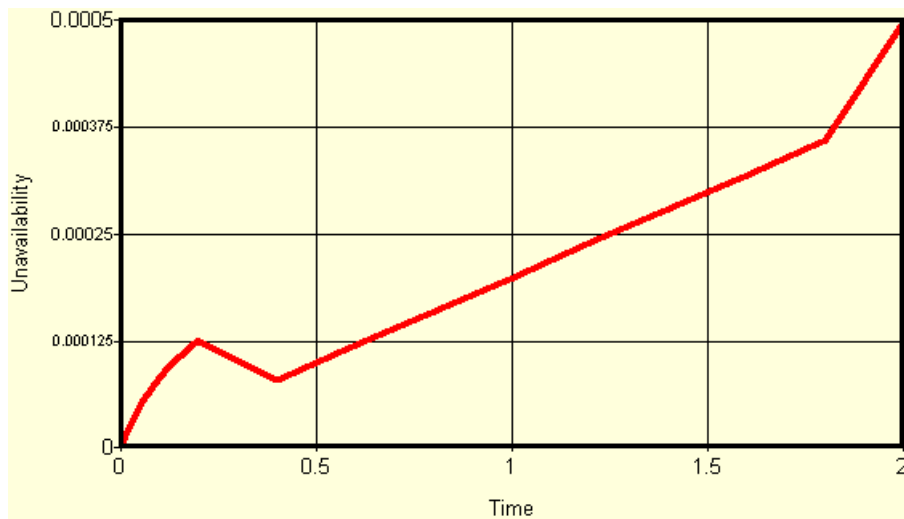
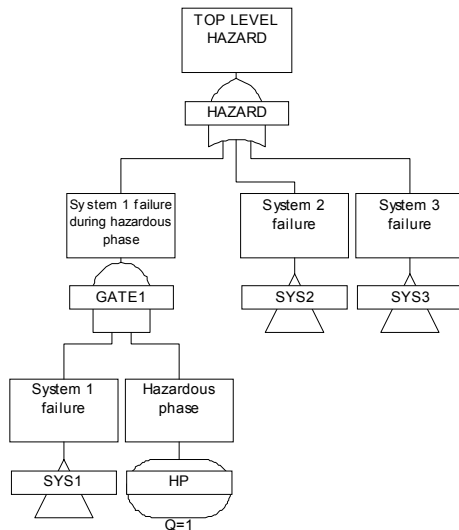
Users may now display cut sets contributing to risk in the 'Results Summary' Dialog.

### Operational Phase Analysis

Two new failure models allow users to change the failure parameters associated with individual events during different phases of the system lifetime. These new models are the 'Fixed-Phased' model and the 'Rate-Phased' models. They are similar to the 'Fixed' and 'Rate' models except they allow users to change the unavailability, failure frequency and failure rate parameters during different phases of operation.

Before using these models, the user must set the number of operational phases required, together with the phase durations, in the 'Project Options' Dialog (Phases Tab). Note that if the total of all the phase durations is less than the specified lifetime (specified on the Calculation Tab of the 'Project Options' Dialog) the program will assume that phases are cyclic until the specified lifetime.

The 'Fixed-Phased' model requires the user to enter unavailability and failure frequency values. In addition, the user must specify an adjustment factor for each phase. The adjustment factor simply multiplies the unavailability and failure frequency parameters during the appropriate phase. The 'Fixed-Phased' model is particularly useful if you wish to effectively modify the structure of the fault tree during a given phase. Consider the example below. A special conditional event has been included in the fault tree. The event has a local 'Fixed-Phased' data model assigned to it. Three phases are defined in the project and the event is assigned a base unavailability of 1 and adjustment factors 1, 0, 1 for the three phases respectively. During the second phase, when the unavailability is adjusted to 0, system 1 cannot contribute to the hazard defined by the top event. For example, an aircraft system may only contribute to a hazard if it fails during take-off or landing, but will not contribute to the hazard if it fails at any other time during the flight.



*Unavailability Plot for Top Gate 'HAZARD'*

The 'Rate-Phased' model requires the user to enter failure rate and repair rate values. In addition, the user must specify an adjustment factor for each phase. The adjustment factor simply multiplies the failure rate parameter during the appropriate phase. The 'Rate-Phased' model is particularly useful if you wish to model standby phases or phases under which a system is placed under high stress (launch of a satellite for example) in addition to normal operational phases.

**Consequence Limit Increased to 10,000**

Users may now define up to 10,000 consequences in a single project.

**Pause Analysis**

Pause analysis now responds more quickly.

**Diagram to Clipboard**

Users may now copy the visible fault or event tree diagram to the clipboard as a colour or 'black and white' enhanced metafile.

**Tree Control Ordering of Gates**

Fault tree gates are now ordered alphanumerically in the tree control.

### **Gate and Event Colours**

Users may now change the default colours of gates and events in the fault tree diagram. The default colours may be set in the 'Project Options' Dialog (Colours Tab). Individual gates and event may also be assigned different colours. This is achieved by selecting the 'Background Colour' Button in the 'Edit Gate' and 'Edit Event' Dialogs.

### **Viewing CFI, MTTF, MTBF and MTTR in Fault Tree Diagrams**

Users may now request the program to display conditional failure intensity (CFI) values for gates rather than frequency values. This may be done by setting the 'Show CFI Preference' flag in the 'Project Options' Dialog (View Tab). Users may also request the program to display MTTF, MTBF and MTTR values rather than unavailability and frequency values. This may also be done using the 'Project Options' Dialog (View Tab).

### **Event Tree Branch Notation**

Users may now request the program to use an alternative notation for 'failure' and 'success' branches in an event tree. The notation 'true' and 'false' may be used as an alternative. To change the notation set the 'Use TRUE/FALSE for ET Branch Labels' flag in the 'Project Options' Dialog (General Tab).

### **Viewing Event Tree Branch Probability**

Users may now request the program to display event tree branch probability values in the diagram. This may be done by setting the 'Show Probability on Branches' flag in the 'Project Options' Dialog (View Tab).

### **Maximising Results Summary and Table Dialogs**

Users may now expand the 'Results Summary' and 'Table' Dialogs to full screen size by selecting the 'Maximise' Button at the top right of the dialog.

### **Cut Set List Filtering**

The 'Results Summary' Dialog cut set lists may now be filtered with partial event names. For importance rankings, events are shown that contain the beginning of any of the event strings in the filter field. For cut sets, the set must contain a matching event or partially matching event for all items in the filter string.

### **Event Tree Probability Verification**

Users may now request the program to check that the probability values of all event tree branches originating from a single branch in the previous column summate to 1. Users must set the 'Check ET Branch Probability Consistence' flag in the 'Project Options' Dialog (General Tab) before performing an analysis. The check is performed as part of the results verification process so that inconsistencies will only be identified after an analysis has been performed.

### **Changes to Modularisation Options**

For large and complex fault trees, producing the minimal cut sets may be a time-consuming process even for modern computers. This process may be speeded-up significantly by modularising parts of the fault tree. FaultTree+ employs a sophisticated algorithm to automatically modularise fault trees and hence compute the minimal cut sets efficiently. FaultTree+ will only modularise independent fault tree structures. These independent fault tree structures do not contain gates or events that are repeated outside of the modular structure. Independent fault tree structures may be analysed separately as a sub-tree and the TOP event of the sub-tree may then be replaced by a super-event in the higher-level fault tree calculations. Super-events appearing in cut set lists and importance rankings are identified by the

original gate name followed by an asterisk (\*). The default 'Sets Generation' options will automatically modularise the fault tree.

In some circumstances users may wish to override the default methods and manually set modularisation flags for an analysis.

In version 11 users may switch modularisation off altogether. This may be done by setting 'Approximation Methods' to 'Custom' in the 'Project Options' Dialog (Sets Generation Tab) and then setting the 'Disable Automatic Modularisation' flag on in the 'Custom Options' Dialog. Setting the 'Disable Automatic Modularisation' flag on will not reduce the efficiency of the cut set generation process (FaultTree+ still modularises independent fault tree structures during the calculation and then expands any super-events it created down to their constituent events). However, it will mean that basic events are visible in cut set lists and importance rankings. You should only disable automatic modularisation if you need to resolve cut sets and importance ranking to the basic event level.

In version 11 users may also override the automatic modularisation status of a gate from within the 'Edit Gate' Dialog. This dialog contains an 'Always Modularise' check box. If set, then FaultTree+ will always convert the gate to a modular event during the analysis process, even if there are gates or events feeding into the modular gate that appear in other parts of the project fault trees. Users should only modularise gates in this way if dependencies between the modularise gate and the rest of the fault tree are very weak. Ignoring dependencies in this way could lead to optimistic unavailability and failure frequency predictions at higher levels in the fault tree.

Users may also modularise gates attached to event tree columns by setting the 'Always Modularise Enabler Gates' and 'Always Modularise Initiator Gates' in the 'Custom Options' Dialog.

### **Grid Control Filter**

Users may now filter the records displayed in the right-hand window grid control by selecting nodes in the left-hand window tree control. To activate this facility, select the 'Filter' option on the right-hand window pop-up menu (activated by pressing the right mouse button over the grid control). The 'Filter' Dialog will appear. Set the 'Filter by tree control selection' flag on. Selection of the appropriate tree control nodes in the left-hand window will then determine the display in the grid control. For example, if the 'Filter by tree control selection' flag is set on for the 'Event' table, then selecting fault tree page nodes in the tree control will show those events that appear on or below the selected page. This is a quick method for identifying events that are dependent on a particular gate. Dependent generic data models could be identified in the same way.

### **Grid Control Find and Replace**

Users may now find and replace strings of text in the grid control. To access this facility select the 'Find & Replace' option on the right-hand window pop-up menu (activated by pressing the right mouse button over the grid control).

### **Offsets Disabled if Paging Disabled**

Manual offsets (created when users manually shift the relative position of symbols in a fault tree diagram) are now disabled when paging is disabled in the fault tree diagram. In previous versions the offsets were maintained, often causing symbols to overlap in the diagram.

### **Identification of Notes in Printed Diagram Reports**

Users may now identify which gates, events and event tree columns are associated with notes within a printed report. If the 'Show Note Symbols' flag is set on in the 'Project Options' Dialog (Reports Tab) then a note tag will be shown alongside gates, events and event tree columns that are associated with one or more note types.