Appendix A. Description of Library and Example Files

This appendix lists and describes the library and example files included in the MEADEP package. The modeling files will prove useful in model design and the example files will help you get acquainted with the MEADEP modules.

A.1 Markov Model Library Files

This section describes the Markov Model Library files for 2-component and 3-component redundant systems.

A.1.1 MarkovHW2.mdl - 2 Redundant Component Hardware System

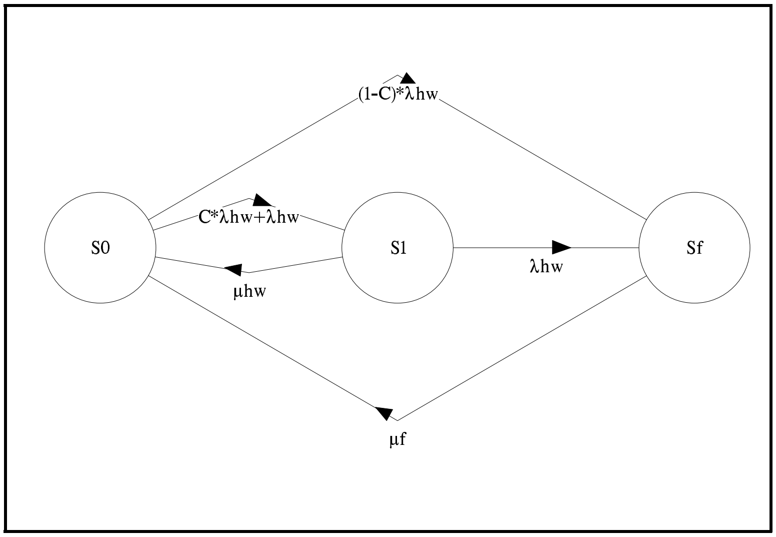

Figure 30   2 Redundant Component Hardware Markov Model

S0	Normal state
S1	State in which one component has failed and the standby component has taken over successfully if the failed component was the primary component
Sf	System failure state in which both components have failed
λhw	Failure rate of a hardware component
μhw	Recovery rate of a hardware component
μf	Recovery rate of the system
C	Coverage of the system

The transition from S0 to S1 models two possible events: (1) The primary component fails and system switches to the standby component successfully (represented by C*λhw). (2) The standby component fails which leads the system to S1 (represented by λhw). The transition from S0 to Sf models the event that the primary component fails and the switchover from the primary to the standby is not successful. In state S1, the failed component is recovering at the rate of μhw, which is modeled by the transition from S1 to S0. During this recovery process, the current primary component can also fail, which is modeled by the transition from S1 to Sf. The transition from Sf to S0 models the event that system is restored to the normal state.

A.1.2 MarkovHW3.mdl - 3 Redundant Component Hardware System

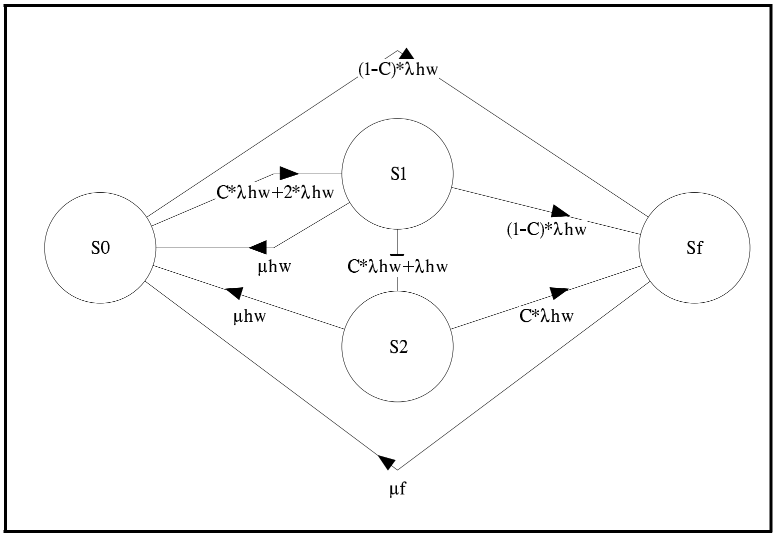

Figure 31 3 Redundant Component Hardware Markov Model

S0	Normal state
S1	State in which one component has failed and a standby component has taken over successfully if the failed component was the primary component
S2	State in which two component have failed and a standby component has taken over successfully if a failed component was the primary component
Sf	System failure state in which all components have failed
λhw	Failure rate of a hardware component
μhw	Recovery rate of a hardware component
μf	Recovery rate of the system
C	Coverage of the system

The transition from S0 to S1 models two possible events: (1) The primary component fails and system switches to a standby component successfully (represented by C*λhw). (2) A standby component fails which leads the system to S1 (represented by 2*λhw). The transition from S0 to Sf models the event that the primary component fails and the switchover from the primary to the standby is not successful. In state S1, the failed component is recovering at the rate of μhw, which is modeled by the transition from S1 to S0. During this recovery process, the events modeled by transitions from S0 to S1 and from S0 to Sf can also occur in S1. These events are now modeled by the transition from S1 to S2 (the current primary component fails and the standby takes over successfully, or the standby fails), the transition from S1 to Sf (the current primary component fails and the switchover to the standby is not successful). In state S2, the failed components are recovering at the rate of μhw, which is modeled by the transition from S2 to S0. During this recovery process, the current primary component can also fail, which is modeled by the transition from S2 to Sf. The transition from Sf to S0 models the event that system is restored to the normal state.

A.1.3 MarkovSW2.mdl - 2 Redundant Task Software System

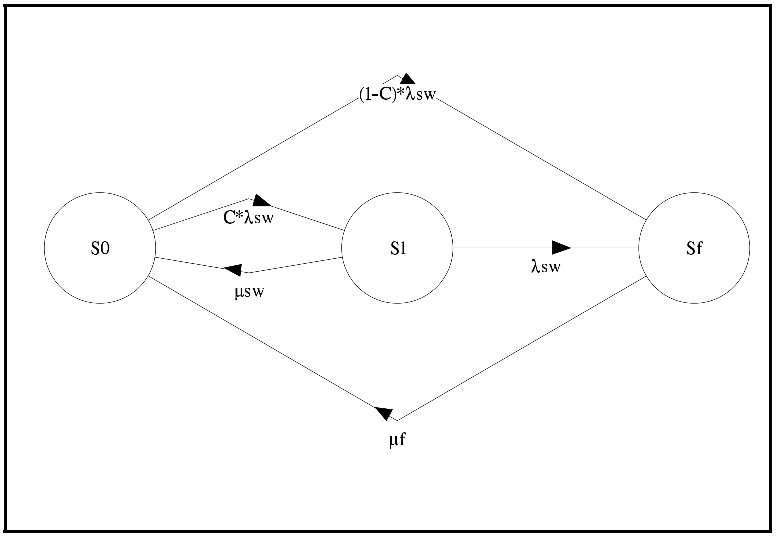

Figure 32 2 Redundant Task Software Markov Model

S0	Normal state
S1	State in which the primary task has failed and the standby task has taken over successfully
Sf	System failure state in which both tasks have failed
λsw	Failure rate of a software task
μsw	Recovery rate of a software task
μf	Recovery rate of the system
C	Coverage of the system

The transition from S0 to S1 models the event that the primary task fails and system switches to the standby task successfully. The transition from S0 to Sf models the event that the primary task fails and the switchover from the primary to the standby is not successful. It is assumed that the standby task cannot fail because it is not actively running. In state S1, the failed task is recovering at the rate of μsw, which is modeled by the transition from S1 to S0. During this recovery process, the current primary task can also fail, which is modeled by the transition from S1 to Sf. The transition from Sf to S0 models the event that system is restored to the normal state.

A.1.4 MarkovSW3.mdl - 3 Redundant Task Software System

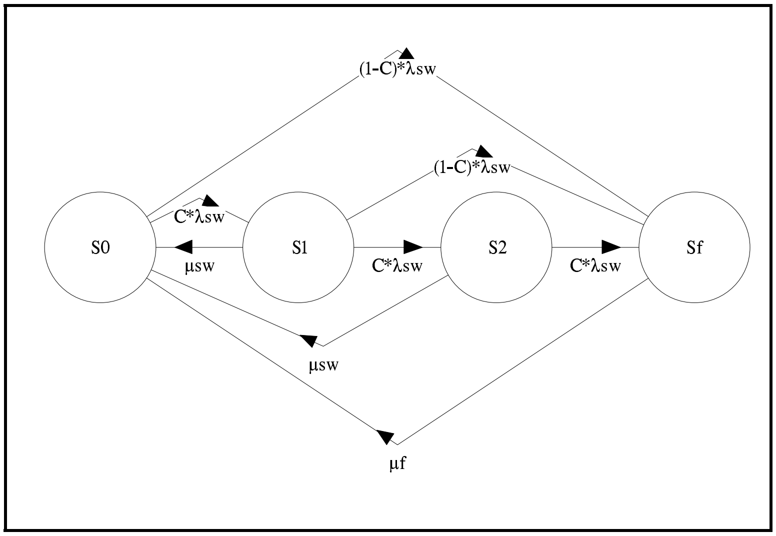

Figure 33 3 Redundant Task Software Markov Model

S0	Normal state
S1	State in which the primary task has failed and a standby task has taken over successfully
S2	State in which two task have failed and a standby task has taken over successfully
Sf	System failure state in which all tasks have failed
λsw	Failure rate of a software task
μsw	Recovery rate of a software task
μf	Recovery rate of the system
C	Coverage of the system

The transition from S0 to S1 models the event that the primary task fails and system switches to the standby task successfully. The transition from S0 to Sf models the event that the primary task fails and the switchover from the primary to the standby is not successful. It is assumed that the standby task cannot fail because it is not actively running. In state S1, the failed task is recovering at the rate of μsw, which is modeled by the transition from S1 to S0. During this recovery process, the events modeled by transitions from S0 to S1 and from S0 to Sf can also occur in S1. These events are modeled by the transition from S1 to S2 (the current primary task fails and the standby takes over successfully), the transition from S1 to Sf (the current primary task fails and the switchover to the standby is not successful). In state S2, the failed tasks are recovering at the rate of μsw, which is modeled by the transition from S2 to S0. During this recovery process, the current primary task can also fail, which is modeled by the transition from S2 to Sf. The transition from Sf to S0 models the event that system is restored to the normal state.

A.2 Block Diagram Library Files

This section describes the Block Diagram Library files which contain diagrams for several parallel and serial block systems.

A.2.1 RBD2P.mdl - 2 Parallel Reliability Blocks

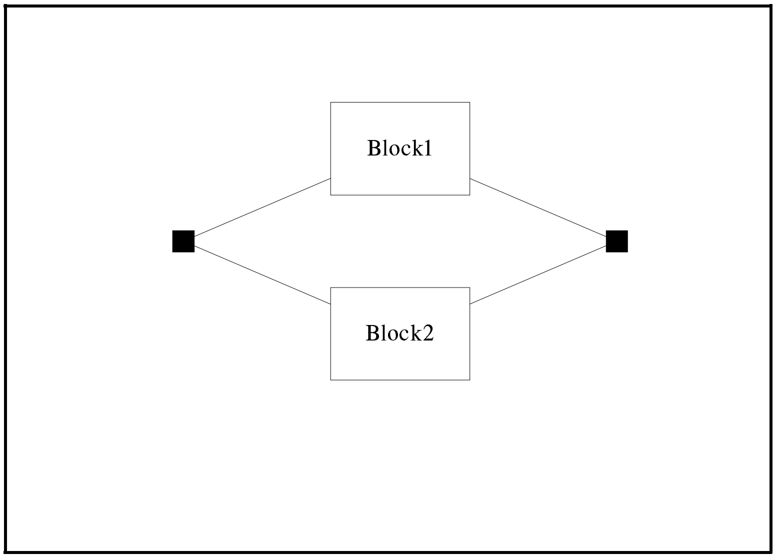

Figure 34 2 Parallel Reliability Blocks

A.2.2 RBD2S.mdl - 2 Serial Reliability Blocks

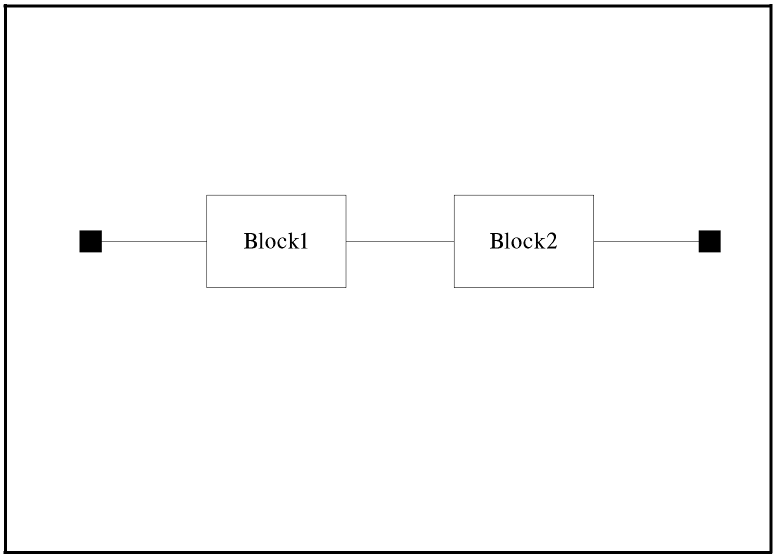

Figure 35 2 Serial Reliability Blocks

A.2.3 RBD3P.mdl - 3 Parallel Reliability Blocks

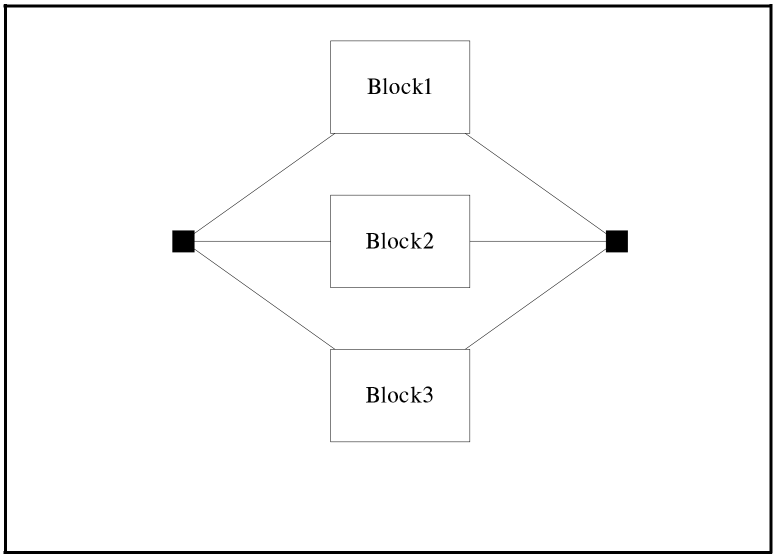

Figure 36 3 Parallel Reliability Blocks

A.2.4 RBD3S.mdl - 3 Serial Reliability Blocks

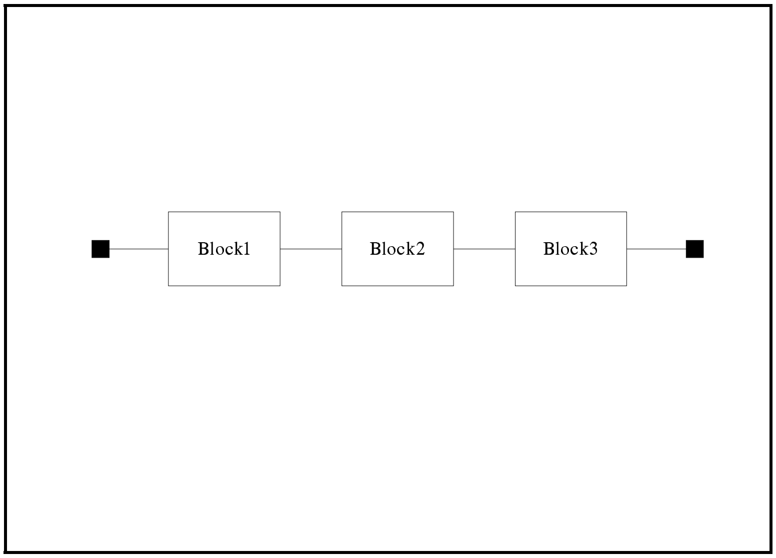

Figure 37 3 Serial Reliability Blocks

A.2.5 RBD4P.mdl - 4 Parallel Reliability Blocks

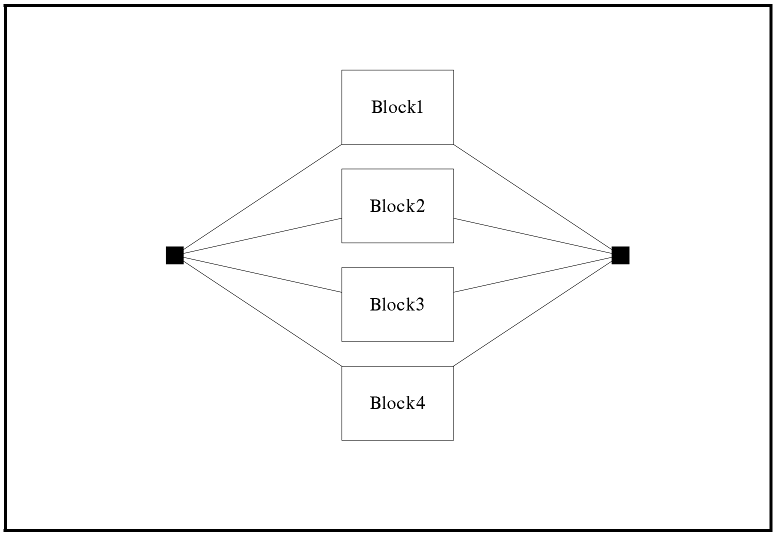

Figure 38 4 Parallel Reliability Blocks

A.2.6 RBD4S.mdl - 4 Serial Reliability Blocks

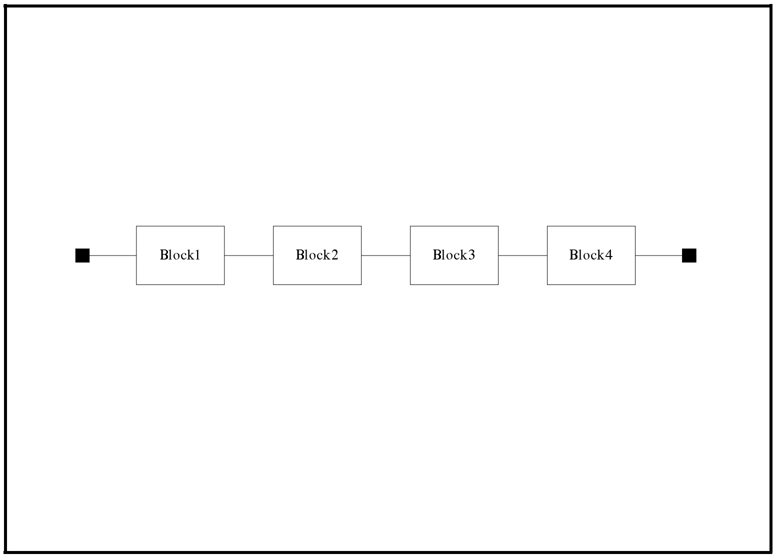

Figure 39 4 Serial Reliability Blocks

A.2.7 RBD5P.mdl - 5 Parallel Reliability Blocks

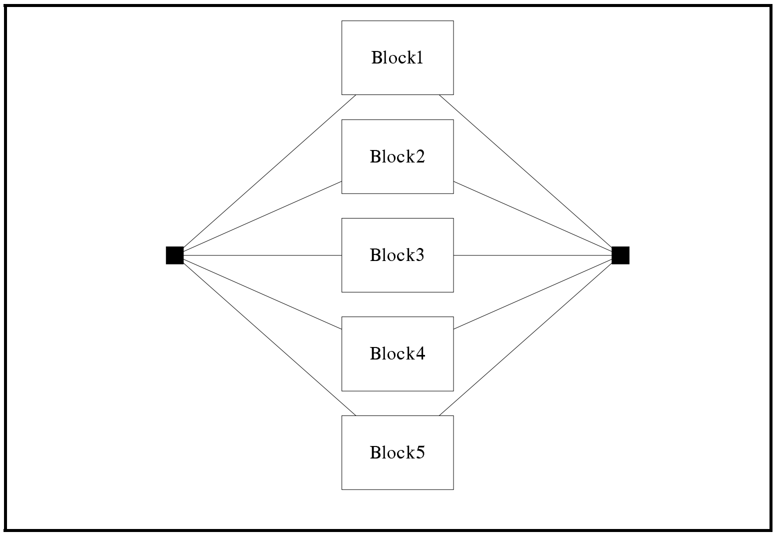

Figure 40 5 Parallel Reliability Blocks

A.2.8 RBD5S.mdl - 5 Serial Reliability Blocks

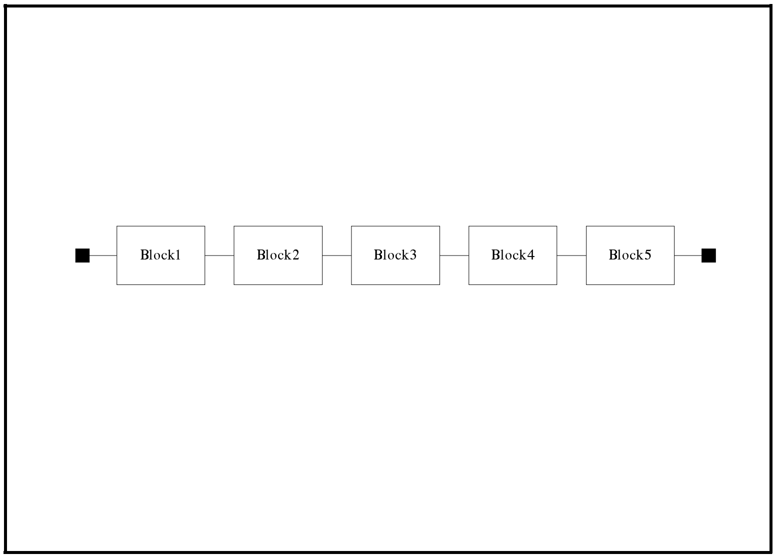

Figure 41 5 Serial Reliability Blocks

A.3 The “Plant” Model

This section describes the files in the Example directory and the plant model for which the files were constructed.

Plant.mdb	MEADEP database file for artificial failures of a plant safety system
Plant.bch	Query file for obtaining multiple statistics from Plant.mdb
Plant.mdg	Graphical modeling file for the plant model (See Figures 9-11)
Plant.mdt	Text modeling file for the plant model
Parameter.txt	Parameter file for the plant model

The modeled configuration has two major components: a plant and a digital safety system which protects the plant by responding to and processing challenges from the plant instrumentation. A 3-level hierarchical model was developed for this configuration. Figure 9 shows the top-level plant model which reflects the intermittent operating profile of safety systems. Figure 10 shows the middle-level model, a safety system which consists of four channels working on a basis of 2-out-of-4 votes for a reactor trip (shut down reactor). Figure 11 shows the bottom-level model of the hierarchy, a single channel which consists of four components. In this analysis, channel failures are assumed to be of the Byzantine type[4] because this type is the worst case failure mode and is hazardous to the protection function.

The notation used in these figures is as follows:

S_{ns} Normal/safe state in which either both plant and safety system are functioning within technical specifications or the plant is in a safe trip (reactor is shut down safely)

S_{sp} Safety processing state in which the safety system is processing a challenge

S_{sf} Safety failure state in which the safety system is not able to respond to a challenge properly while the plant is functioning within technical specifications

S_{ph} Plant hazard state which is the result of a failure of the safety system to process a challenge successfully in terms of initiating a necessary reactor trip

P_S Probability of success upon demand, i.e., the safety system will be successful in responding a challenge (initially set to 0.9999)

r Arrival rate of challenges from the plant requiring a response of the safety system (assumed to be once a year, a typical value)

τ Challenge processing time (assumed to be a half hour, a conservative assumption)

λ_{ss} Failure rate of the safety system (evaluated from the safety system model in Figure 10)

μ_{ss} Rate for detection and handling of a safety system failure (evaluated from the safety system model in Figure 10)

μ_f Recovery rate of the plant after a plant failure (which has no impact on the plant MTBH)

S_0 Normal state in which all four channels are functioning properly

S_n State in which one channel has failed and the output of the failed channel votes for “no trip”

S_y State in which one channel has failed and the output of the failed channel votes for “trip”

S_{nn} State in which two channels have failed and both failed channels vote for “no trip”

[4] The faulty channel continues execution and lies when asked for information.

S_{yn} State in which two channels have failed and one failed channel votes for “trip” and another failed channel votes for “no trip”

S_{yy} State in which two channels have failed and both failed channels vote for “trip”

S_{nnn} State in which at least three channels have failed and at least three failed channels vote for “no trip”; This state is equivalent to state S_{sf} in Figure 9 because the safety system would generate a “no trip” signal should a challenge arrive.

S_{yxx} State in which three channels have failed and at least one of the failed channels vote for “trip”

S_{trip} Plant trip state (reactor is shut down)

P_n Probability that the channel output votes for “no trip”, given a channel failure (assumed to be 0.5)

λ_c Failure rate of a channel (evaluated from the channel model in Figure 11)

μ_c Recovery rate of a channel (evaluated from the channel model in Figure 11)

λ_{com} Common mode failure rate for the safety system (assumed to be once every 10 years)

T_{dh} Failure detection and handling time, given that at least three channels have failed (assumed to be one hour)

T_{trip} Plant trip duration (assumed to be 50 hours)

CPU The component “CPU” of a channel

IO The component “IO” of a channel

Power The component “Power” of a channel

OS The component “OS” (operating system) of a channel

λ, μ Failure rate and recovery rate for the above components. These parameters are evaluated from the data in Plant.mdb.

In Figure 9, if a challenge arrives in the normal/safe state, the safety system will respond to it successfully with probability P_S and go to the safety processing state S_{sp} (modeled by transition P_S*r, from S_{ns} to S_{sp}). During the safety processing, if the safety system fails due to random failures, the plant will be in the hazard state (transition λ_{SS}, from S_{sp} to S_{ph}). Otherwise, the safety system will go back to the normal/safe state after the mean processing time τ (transition $1/\tau$, from S_{sp} to S_{ns}). When a challenge arrives in the normal/safe state, the safety system may respond to it unsuccessfully due to hardware/software design or implementation problems and go to the plant hazard state (transition $(1-P_S)*r$, from S_{ns} to S_{ph}).[5] Thus, maximizing P_S is the major goal for this model. Sometimes the safety system random failures occur in the normal/safe state and enters the safety failure state S_{sf} (transition λ_{SS}, from S_{ns} to S_{sf}).[6] The safety system will go back to the normal/safe state when the safety system failure is detected and handled (transition μ_{ss}, from S_{sf} to S_{ns}). But during the failure detection and handling period in the state S_{sf}, should a challenge arrive, the plant would fail to initiate a trip and would go to the plant hazard state (transition r, from S_{sf} to S_{ph}) because the safety system is not able to vote for “trip” in this state.

[5] An example of such failures is that the software makes a wrong judgement on an unusual combination of sensed physical parameters such that it fails to initiate a necessary trip.

[6] An example of such failures is that a problem (e.g., memory leaking) of the underlying operating system blocks the running of the application software for all channels, i.e., a common mode failure.

In Figure 10, each channel can fail with its output left at either a state voting for “no trip” or a state voting for “trip”, before the failure is detected and handled. When at least three channels have failed (due to either independent or common mode faults) and have left at least three votes for “no trip” (state S_{nnn}), the safety system would not respond a challenge correctly because the required 2-out-of-4 votes for “trip” never satisfy in this state. This state is regarded as the failure state of the safety system and is equivalent to state S_{sf} in Figure 9. Minimizing the occupancy probability of this state is the major goal for this model. The common mode failure rate (λ_{com}) and the failure detection and handling time (T_{dh}) are key parameters for minimizing this occupancy probability. All of the other states in this model do not affect the ability of the safety system to vote for “trip” in case a challenge arrives, and therefore none of them is designated as a failure state.

The diagram shown in Figure 11 is a rough modeling of the four components in a safety channel. Although the four components can be further decomposed at lower levels, this further detailed modeling will not have much impact on the results because the single channel failure rate has little effect on the plant model.

Appendix B. Numerical Methods Used in MEADEP

This appendix describes statistical and numerical methods used in MEADEP for parameter estimation, model evaluation, and probability distribution function estimation in fitting empirical distributions (histograms).

B.1 Parameter Estimation Methods

Three essentially different types of parameters can be estimated from data by MEADEP (through the use of the IMSL Statistical Library): failure rate or MTBF, recovery rate or MTTR, and coverage. For each type, the sample mean, a lower bound and an upper bound at a certain level of confidence are provided, whenever applicable.

For the failure rate estimation, the exponential MTBF distribution is assumed (a reasonable assumption by theory and measurement). If n failure events are included in the data for the measurement period T, the failure rate upper bound, λ_U, and lower bound, λ_L, at the 100(1 − α)% confidence level (α is the significance level) are given by [Kececioglu93]

$$\lambda_L = \frac{\chi^2_{\alpha;2n}}{2T} \,, \qquad \lambda_U = \frac{\chi^2_{1-\alpha;2n+2}}{2T} \tag{2}$$

where χ^2 represents the Chi-square distribution. The above λ_U formula is also applicable to the case in which n is zero, i.e., no failure occurred in T. When n is zero, this formula is equivalent to the following estimator given in [Tang95]:

$$\lambda_U = \frac{-\ln(\alpha)}{T} \tag{3}$$

For MTTR, either the exponential or normal distribution can be assumed. In the first case, estimators similar to Equation (3) are used. In the second case, the student’s t distribution is used. Compared to the normal distribution, the student’s t distribution provides better estimates when the variance is unknown and the sample size is less than 25 [Kececioglu93]. The estimators for the lower and upper bounds are the following:

$$MTTR_L = M - t_{\alpha;n-1}\sqrt{\frac{S}{n}} \,, \qquad MTTR_U = M + t_{\alpha;n-1}\sqrt{\frac{S}{n}} \tag{4}$$

where t represents the student’s t distribution, α is the significance level, and n, M and S are the sample size, sample mean and sample variance, respectively.

For the coverage C, the binomial distribution is used. If the number of successes, s, in n trials is greater than zero and less than n, the lower bound (C_L) and upper bound (C_U) of C can be approximated by [Kececioglu93]

$$C_L = \frac{1}{1 + \frac{n-s+1}{s} F_{1-\alpha;2(n-s)+2;2s}} , \qquad C_U = \frac{1}{1 + \frac{n-s}{s+1} F_{\alpha;2(n-s);2s+2}} \tag{5}$$

where *F* represents the *F* distribution and α is the significance level. If *s* equals *n* (a 100% coverage), a conservative lower bound is given by [Tang95]

$$C_L = \alpha^{\frac{1}{n}} \tag{6}$$

B.2 Model Evaluation Methods

Models are developed hierarchically from the top level to the bottom level, but the evaluation has to e performed from the bottom level to the top level. In the hierarchy tree, each node is a model representing a system or subsystem. For each node, four measures — failure rate (λ), recovery rate (μ), availability (*A*), and reliability (*R*) — are evaluated by MEADEP. At the bottom level, failure rates and recovery rates for all modeled components (elemental blocks) are given by the user or obtained from data. For a Markov model, all transition rates, the initial state, and the failure state are also specified by the user. Based on these parameters, the four measures are evaluated from bottom to top using methods discussed below.

Assume a reliability block diagram consists of *n* blocks connected either in series or in parallel. The four measures for block *i* are denoted by λ_i, μ_i, A_i and R_i. If block *i* is an elemental block, A_i and R_i are calculated by

$$A_i = \frac{\mu_i}{\lambda_i + \mu_i} , \qquad R_i = e^{-\lambda_i T} \tag{7}$$

where *T* is a time interval (an integer representing hours) specified by the user during evaluation. If block *i* is decomposed into a lower level diagram, λ_i, μ_i, A_i and R_i are calculated using the formulas for block diagrams described below.

Let λ, μ, *A* and *R* denote the four measures for the block diagram. If the *n* blocks in the diagram are connected in series, *A* and *R* are calculated by

$$A = \prod_{i=1}^{n} A_i , \qquad R = \prod_{i=1}^{n} R_i \tag{8}$$

If the *n* blocks are connected in parallel, *A* and *R* are calculated by

$$A = 1 - \prod_{i=1}^{n} (1 - A_i) \ , \qquad R = 1 - \prod_{i=1}^{n} (1 - R_i) \tag{9}$$

No matter whether the *n* blocks are connected in series or in parallel, λ and μ are calculated by

$$\mu = \sum_{i=1}^{n} \frac{\lambda_i \mu_i}{\sum_{j=1}^{n} \lambda_j} \ , \qquad \lambda = \frac{\mu(1-A)}{A} \tag{10}$$

For a Markov chain with *n* states, the availability *A* is calculated from state reward rates (r_i) and occupancy probabilities (p_i):

$$A = \sum_{i=1}^{n} r_i \ p_i \tag{11}$$

where r_i is defined by the user and p_i is obtained by solving the *Q* matrix (infinitesimal generator) of the Markov chain (Eq. 8.13 and 8.14, [Trivedi82]). Since reward rate is used, partially available states are allowed in the model. The reliability *R* at time *T* (an integer representing hours) is calculated based on the uniformization technique [Reibman88] and the Chapman-Kolmogorov equation:

$$R = \sum_{i=1}^{n} r_i \ p_i(T) \ , \qquad P(T) = P(0) U^{T} \tag{12}$$

where $P(T)=(p_0(T), p_1(T), ..., p_n(T))$ is the state probability vector at time *T*, *P(0)* is the initial state probability vector, and *U* is the unit time transition probability matrix converted from the *Q* matrix by using the uniformization technique and by setting the failure state to the absorbing state:

$$U = (Q/2^s + I)^{2^s} \tag{13}$$

where *s* is the smallest integer such that 2^s is greater than the largest element, q_{max}, in *Q*. Since *T* is an integer, it can be expressed as the following sum:

$$T = \sum_{i=0}^{m} C_i 2^i \tag{14}$$

where C_i (coefficient) is either 1 or 0 and *m* is the maximum integer such that $2^m < T$. Thus, the computation of *U* and U^T can be done by a matrix squaring iteration [Reibman88] for $s+m$ times. The computation complexity for this algorithm is $O(n^3(s+m))$, or $O(n^3 \ lg(q_{max}T))$, where *n* is the number of states in the Markov chain and is restricted to a maximum of 100 in the current MEADEP version. Normally we have *s<10*. *T* is restricted to a maximum of 10^9 hours (10^5 years) in MEADEP and *m* is thus bounded by 30. For the two remaining measures of the Markov chain, the recovery rate μ is simply the transition rate out of the failure state, and the failure rate λ is calculated from *A* and μ by Eq. (9).

B.3 Probability Distribution Estimation Methods

MEADEP allows to super-plot, over a histogram, five different analytical probability distribution functions determined by the sample mean and sample variance: exponential, gamma, Weibull, normal and lognormal. Meanwhile, the estimated parameters for these functions as well as the results of the Chi-Square and Kolmogorov-Smirnov goodness-of-fit tests [Iyer96] are also provided. The following is the definition for the five functions and the estimators used to determine parameters in these functions [Trivedi82, Shapiro90]. The notation used is as follows:

f(t)	probability distribution function (pdf)
F(t)	denotes cumulative distribution function (cdf)
M	sample mean
S^2	sample variance

Exponential Distribution

$$f(t) = \lambda e^{-\lambda t}, \qquad F(t) = 1 - e^{-\lambda t} \tag{15}$$

where

$$\lambda = \frac{1}{M} \tag{16}$$

Gamma Distribution

$$f(t) = \frac{\beta(\beta t)^{\alpha-1} e^{-\beta t}}{\Gamma(\alpha)}, \qquad F(t) = \frac{1}{\Gamma(\alpha)} \int_0^t e^{-x} x^{\alpha-1} dx \tag{17}$$

where

$$\beta = \frac{M}{S^2}, \qquad \alpha = \frac{M^2}{S^2} \tag{18}$$

Weibull Distribution

$$f(t) = \beta\alpha t^{\alpha-1} e^{-\beta t^{\alpha}}, \qquad F(t) = 1 - e^{-\beta t^{\alpha}} \tag{19}$$

where

$$\frac{M^2}{S^2} = \frac{[\Gamma(1+\frac{1}{\alpha})]^2}{\Gamma(1+\frac{2}{\alpha}) - [\Gamma(1+\frac{1}{\alpha})]^2}, \qquad \beta = \frac{1}{[M \;/\; \Gamma(1+\frac{1}{\alpha})]^\alpha} \tag{20}$$

Normal Distribution

$$f(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-\mu)^2}{2\sigma^2}}, \qquad F(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-\frac{x^2}{2}} dx \tag{21}$$

where

$$\mu = M, \qquad \sigma^2 = S^2 \tag{22}$$

Lognormal Distribution

$$f(t) = \frac{1}{\sigma t\sqrt{2\pi}} \; e^{-\frac{(\ln(t)-\mu)^2}{2\sigma^2}}, \qquad F(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\ln(t)} e^{-\frac{x^2}{2}} dx \tag{23}$$

where

$$\mu = \frac{\sum_{i=1}^{N} \ln(x_i)}{N}, \qquad \sigma^2 = \frac{\sum_{i=1}^{N} (\ln(x_i) - \mu)^2}{N} \tag{24}$$

where x_i's are sample instances and N is the sample size.

References

[Iyer96] R. K. Iyer and D. Tang, “Experimental Analysis of Computer System Dependability,” *Fault-Tolerant Computer System Design*, D. K. Pradhan (Ed.), Prentice Hall PTR, Upper Saddle River, NJ, 1996, pp. 282-392.

[Kececioglu93] D. Kececioglu, *Reliability and Life Testing Handbook*, Vol. 1 & 2, PTR Prentice Hall, Englewood Cliffs, NJ, 1993.

[Reibman88] A. Reibman and K. S. Trivedi, “Numerical Transient Analysis of Markov Models,” *Computational Operations Research*, Vol. 15, No. 1, 1988, pp. 19-36.

[Shapiro90] S. S. Shapiro, “Selection, Fitting, and Testing Statistical Models,” Chapter 6 of *Handbook of Statistical Methods for Engineers and Scientists*, H. M. Wadsworth, Jr., Editor, McGraw-Hill, New York, 1990, pp. 6.1-6.34.

[Tang95] D. *Tang and H. Hecht, Measurement-Based Dependability Analysis for Critical Digital Systems*, SBIR NRC-04-94-061 Final Report, SoHaR Incorporated, May 1995.

[Trivedi82] K. S. Trivedi, *Probability & Statistics with Reliability, Queuing, and Computer Science Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

Appendix C. Glossary

Availability

Probability that a system is operating properly and is available to perform its functions. It is calculated by the ratio between the total available system time and total supposed system operating time for a given period.

Bind

Here, it means to connect (assign) a value to a parameter.

Byzantine Failure

The faulty computing unit continues execution and lies when asked for information. This is the worst case failure mode because the faulty unit can generate misleading information.

Chi-Square Goodness-of-Fit Test

A statistical method to test if one probability distribution matches another probability distribution, based on the sum of squared numerical errors between the two probability density functions (pdf). See also Goodness-of-Fit Test.

Cluster Density

The number of events in a cluster. See also Clustering Analysis.

Cluster Span

The elapsed time between the first event and the last event in a cluster. See also Clustering Analysis.

Clustering Analysis

Clustering analysis is a statistical analysis method to identify related events. The method merges multiple events into a single cluster if time between any two neighbor events is less than a specified interval. A cluster can have one or multiple events. The number of events in a cluster is called *cluster density*. The elapsed time between the first event and the last event in a cluster is called *cluster span*.

Combinatorial Model

A method of developing an analytical expression for a system's reliability. Examples of combinatorial models are reliability block diagrams and *k*-out-of-*n* models.

Confidence Interval

An interval determined by a lower bound and an upper bound derived from a sample. We have a certain level of confidence that the a statistic is included in the interval. See also Confidence Level.

Confidence Level

A probability (or percentage of confidence) with which we believe that a statistic is included in a confidence interval derived from a sample. See also Confidence Interval.

Coverage

Here, it represents *failure recovery coverage* which is the probability that a system will recover from a failure at the subsystem or component level, i.e., the failure is masked by fault tolerance provisions in the system. It is calculated by the ratio between the number of recovered failures and the number of total failures in a system for a given period.

Critical Application

An application in which the incorrect performance of computations can create devastating results such as jeopardizing human life or having high economic impact.

Criticality

The extent of the effect of a failure to the system.

Dependability

The quality of service that a digital system provides. Reliability, availability, safety, maintainability are some of the dependability measures.

Error

The manifestation of a fault. Specifically, an error is a deviation of the internal system state from accuracy or correctness. Error can occur some distance from the fault sites.

Failure

A deviation in the expected performance of a system as observed externally. Failures are caused by errors.

Failure Rate

The expected number of failures per unit time. The mean failure rate is calculated by the ratio between the number of failures which occurred in a given period and the time span of the period.

Fault

An incorrect state of hardware or software resulting from failures of components, physical interference from the environment, operator error, or incorrect design and implementation. A fault may manifest itself to errors, and the errors may cause failures.

Fault Tolerance

The ability to continue the correct performance of functions in the presence of faults. A fault tolerance technique is a hardware or software method to provide a service complying with the specification in spite of faults. A fault-tolerant system is a system that can continue the correct performance of its functions in the presence of faults.

Goodness-of-Fit Test

A statistical method to measure if an assumed analytical probability distribution fits an empirical probability distribution constructed from data. The test result is a value in the range [0, 1]. This value is the significance level at which the assumption that the analytical distribution matches the empirical distribution cannot be rejected. A value of 0.05 is the minimum significance level typically used for acceptance of the assumption. The Chi-Square and Kolmogorov-Smirnov goodness-of-fit tests are two commonly used test methods.

Graphical Modeling File

A graphical modeling file contains all information about a graphical model and associated parameters defined by the user using the MEADEP Model Generator module. It is the working file of Model Generator.

Interval-Reliability

The average reliability for a given time interval *T*. See also Reliability.

Kolmogorov-Smirnov Goodness-of-Fit Test

A statistical method to test if one probability distribution matches another probability distribution, based on the maximum numerical error between the two cumulated probability functions (cdf). See also Goodness-of-Fit Test.

K-out-of-n Model

A system in which *k* out of the *n* components in the system must operate correctly for the system to operate properly.

Library File

A library file is a graphical modeling file that defines the structure of a dependability model, but does not contain parameter values. It can be read into a diagram screen in the modeling process. Thus the user can make use of library files in constructing models to reduce development time. See also Graphical Modeling File.

Markov Diagram

A diagram that shows a Markov model (states and transitions). See also Markov Model.

Markov Model (Markov Chain)

A mathematical model to describe a system. It consists of system states and transitions between states. A system state represents a combination of operational and failed components in a system. The system stays in a state for a random time which follows an exponential distribution, and then goes into another state. A transition from one state to another state is characterized by a transition rate. Some of the states are failure states. A Markov model can be solved mathematically to obtain dependability measures. For example, the expected proportion of time that the system spends in the failure states, which is just the system unavailability, can be calculated.

ODBC

Open DataBase Connectivity (ODBC) is an interface for programs to access data in any database for which an ODBC driver exists in the system. By using ODBC, a program can interface with multiple database formats such as Access, dBase and Paradox. MEADEP uses ODBC to access its internal data (in the Access format) and to convert other data formats supported by the ODBC drivers installed on the system to the MEADEP data format.

Parallel System

A system in which only one of the *n* components in the system must operate properly for the system to operate properly. See also Reliability Block Diagram.

Parameter File

A parameter file is a text file containing a list of parameters and values to initialize them. The parameter file is used by the MEADEP Model Evaluator module to initialize parameters in the model evaluation process.

Query File

A query file is a text file that represents one or more query specifications. By using query files, the user can save a large amount of time in interactive screen specifications for selecting records to estimate statistics.

Reconfiguration

The process of eliminating a faulty entity from a system and restoring the system to some operational state.

Recovery

The process of regaining operational status and restoring a system's integrity after the occurrence of an error or failure. In MEADEP, recovery also includes repair.

Recovery Rate

The expected number of recoveries per unit time. It is the reciprocal of the mean recovery time. In MEADEP, recovery rate also includes repair rate.

Reliability

The probability that a system performs properly throughout a time interval, given that the system was performing properly at the beginning of the time interval.

Reliability Block Diagram

A graphical method of depicting the components in a system and their connections in terms of functioning requirements. Each block represents a component. Blocks may be connected in one of the two basic patterns: serial and parallel. If two or more blocks are connected in series, they constitute a serial system in which all the blocks are required to function for the system to function. If two or more blocks are connected in parallel, they constitute a parallel system in which one of the blocks is required to function for the system to function. See also serial system and parallel system.

Safety

The probability that a system will either perform its functions correctly or will discontinue its functions in a well-defined, safe manner to avoid a state in which human life, economics, or environment are endangered.

Serial System

A system in which all components must operate properly for the system to operate properly. See also Reliability Block Diagram.

Stiffness

In a Markov dependability model, failure rates tend to be very small numbers while recovery rates tend to be much larger. Stiffness means the technical difficulty in model solution caused by the difference between the largest and the smallest parameters in the model.

Text Modeling File

A text modeling file contains model specifications for directing the MEADEP Model Evaluator module to evaluate the model to obtain results. The text modeling file is generated by the MEADEP Model Generator module from the graphical model and parameters defined by the user. Although a text modeling file includes a parameter initialization section, parameters used in the model can be initialized by a parameter file. See also Parameter File.

Unavailability

The opposite of availability. It is calculated as 1-availability. See also Availability.

Yearly-Downtime

The average downtime for a year. It is calculated by Unavailability × 8760 hours. See also unavailability.

Appendix D. Index