Reliability Block Diagram Modeling – Comparisons of Three Software Packages

Aron Brall, SRS Technologies
William Hagen, Ford Motor Company
Hung Tran, SRS Technologies

Key Words: Reliability Modeling, Reliability Software, Simulation

SUMMARY & CONCLUSIONS

The use of commercially available software for analyzing Reliability Block Diagrams (RBD) has become the rule for the vast majority of Reliability Analysts and Engineers. After a model has been developed and checked, a software package is generally used to evaluate the model. For the evaluation of system maintenance, especially in complex redundancy schemes, simulation is required to resolve the availability performance of the model. The results produced by the software packages are usually presented by the analyst without significant questions about the algorithms, simulation methodology, etc used by the particular software package the analyst uses.

This paper provides a comparison of the results of three competitive packages. It was hypothesized that there would be differences in results due to differences in algorithms and simulation methodologies, particularly for complex models. It was not the intent of this paper to judge the relative accuracy of the results produced. The purpose of this paper is to provide awareness to analysts that all results of Reliability Modeling, including those produced by computer simulation packages, need to be understood in the context of the modeling methodology and solution algorithms and methodologies. It is also necessary that the results are presented with the assumptions used by the particular software package.

The three software packages that were compared are Reliasoft BlockSim – Version 6.5.2, ARINC Raptor – Version 7.0.07, and Relex Software Reliability Block Diagram. This evaluation was performed with the cooperation of the software suppliers to the maximum extent possible. The results, particularly any differences, will be reviewed with the suppliers prior to this paper's presentation at RAMS 2007.

The methodology used a one block diagram, a simple diagram, a complex diagram, and a project diagram. The project diagram was based on actual hardware. The other models were hypothetical. The only restriction placed on the models was that each model must be capable of being run on all of the software packages (after conversion to each software package's protocols). Special features that might be available in one or two software packages would not be evaluated. The primary effort was to assess the differences in results created by the differences in algorithms and simulation methodologies. The hypothesis was verified, even in the single block model!

1 INTRODUCTION

Reliability Engineers and Analysts have become increasingly dependent on reliability modeling software. This dependence includes, in many cases, unquestioning confidence in the accuracy of the output of the software, assuming the Reliability Block Diagram (RBD) is correctly input. The problem at hand is that in complex models, especially those with various levels of redundancy, maintenance, spares and crew limitations, and mixed failure rate and maintenance distributions, the different analytical methodologies, simulation methodology, deterministic analysis, simulation parameters, etc. can produce different results for the same model using different software packages. The concern is that the more complex the RBD, the more likely that the results produced will differ from package to package. This paper intends to provide a warning to the analyst that the simulation parameters, as well as software methodology need to be understood so that the results of the analysis can be representative of the design and independent of the software package used.

2 THE SOFTWARE PACKAGES

We chose the three packages because of their popularity, and the fact that they are sufficiently expensive so that a Reliability Engineer or Analyst is very likely to have the use of only one package. The descriptions below are based on the suppliers' literature. We have listed the products alphabetically by Producer. We did not use any capability of the software packages beyond analyzing reliability and our understanding of how they produce results. We used block diagrams with failure and repair distributions. Cost, throughput, capacity etc. are topics for another paper.

2.1 *ARINC Raptor 7.0.07*

From the ARINC Raptor web site: "Raptor is a software tool that simulates the operations of any system. Sophisticated Monte Carlo simulation algorithms are used to achieve these results".

0-7803-9766-5/07/$25.00 ©2007 IEEE

Raptor appears to be a pure Monte Carlo simulation tool to solve reliability block diagrams.

2.2 *Reliasoft BlockSim 6.5.2*

From the Reliasoft BlockSim web site: "Flexible Reliability Block Diagram (RBD) creation. Exact reliability results/plots and optimum reliability allocation. Repairable system analysis via simulation (reliability, maintainability, availability) plus throughput, life cycle cost and related analyses."

BlockSim appears to use Monte Carlo simulation with algorithms used to speed the processing time to solve reliability block diagrams. BlockSim will also provide an analytical calculation of reliability.

2.3 *Relex Reliability Block Diagram*

From the Relex web site: "At the core of Relex RBD is a highly intelligent computational engine. First, each diagram is analyzed to determine the best approach for problem solving using pure analytical solutions, simulation, or a combination of both. Once a methodology is determined, the powerful Relex RBD calculations are engaged to produce fast, accurate results."

Relex RBD appears to be a hybrid tool that uses algorithms and simulation in varying combinations to solve reliability block diagrams.

3 THE MODELS

We used several models to put the software packages through their paces and identify differences in results. A total of four models were used in varying combinations across the packages. The intent was not to pick a winner, but to increase awareness of the care that must be taken in simulating.

3.1 *One Block Model*

This model consists of one simple block with a Weibull failure distribution and Lognormal repair distribution. The simulation was set at 1,000 hours run time and 10,000 runs. See Figure 3-1 and Table 3-1 for the details of this model.

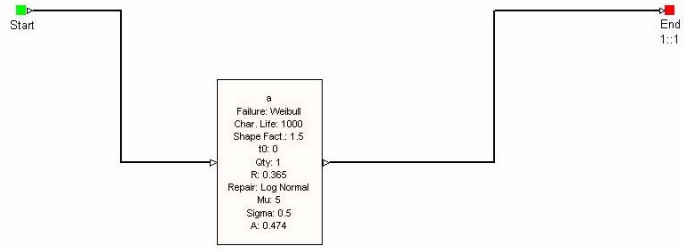

Figure 3-1 – One Block Model

3.2 *Simple Model*

This model consists of 17 blocks with some redundancy and k of n, but no maintenance. The simulation was set at 100 hours and 1,000 runs. See Figure 3-2 and Table 3-2 for the details of this model

Block Parameter		Probability Distribution	Parameter 1	Parameter 2
Failure Distribution	a	Weibull	Shape 1.5	Scale 1,000
Repair Distribution	a	Lognormal	Mu 5	Sigma 0.5

Table 3-1- One Block Model Input Data

3.3 *Complex Model*

This model consists of 194 blocks, redundancy, k of n, and corrective maintenance. The simulation was set at 100 hours and 10,000 runs. Figure 3-3 gives an impression of the complexity of this model. Due to its size, it can't be shown effectively within the page limitations of this paper.

3.4 *Large Exponential Model*

This model consists of 83 blocks, all modeled with the Exponential distribution for failure, and no repair distribution. The simulation was set at 61,312 hours and 1,000 runs. Figure 3.4 shows the block diagram.

Block Name	Failure Distribution	Parameter 1	Parameter 2
a	Weibull	Shape 1.5	Scale 1,000
b	Normal	Mean 250	Std Dev 50
c	Exponential	10,000	0
d	Lognormal	Mu 6	Sigma 2
e	Weibull	Shape 1.5	Scale 2,300
f	Normal	Mean 250	Std Dev 50
g	Exponential	10,000	0
h	Lognormal	Mu 8	Sigma 1
i	Weibull	Shape 1.5	Scale 1,000
j	Normal	Mean 250	Std Dev 50
k	Exponential	10,000	0
l	Lognormal	Mu 8	Sigma 3
m	Weibull	Shape 2.0	Scale 1,000
n	Weibull	Shape 3.0	Scale 1,000
o	Weibull	Shape 4.0	Scale 1,000
p	Weibull	Shape 0.5	Scale 1,000
q	Weibull	Shape 0.4	Scale 1,000

Table 3-2 - Simple Model Input Data

4 RESULTS OF SIMULATIONS/ANALYSIS

The discussion below will not call out the shortcomings of any package, just the differences in values produced, or, if a package had severe modeling limitations, they will be noted. It is not the intent of this effort to pick a winner, but to caution practitioners regarding the pitfalls of using any reliability modeling software.

4.1 *Comparison of Results*

Table 4-1 shows the results of running all of the models through the various software packages. The differences in the

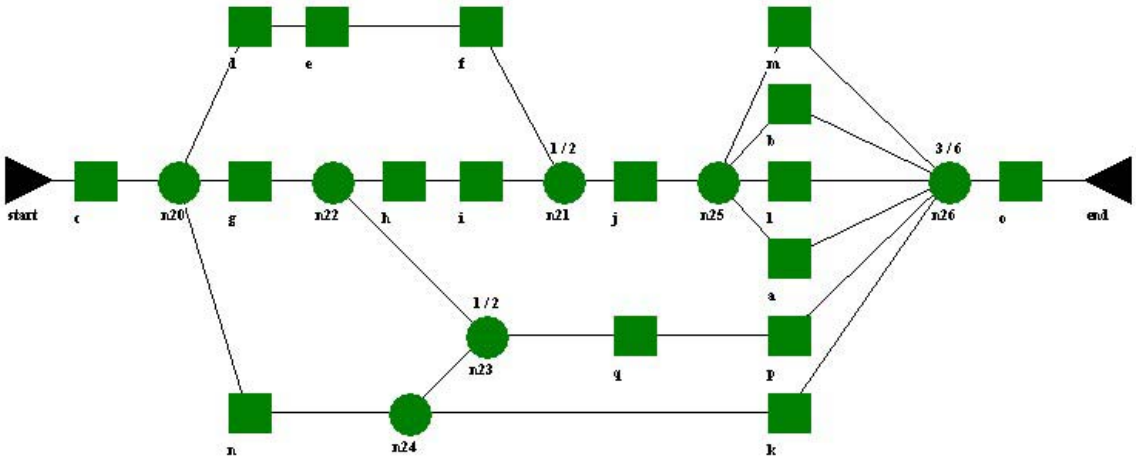

Figure 3-2 - Simple Model

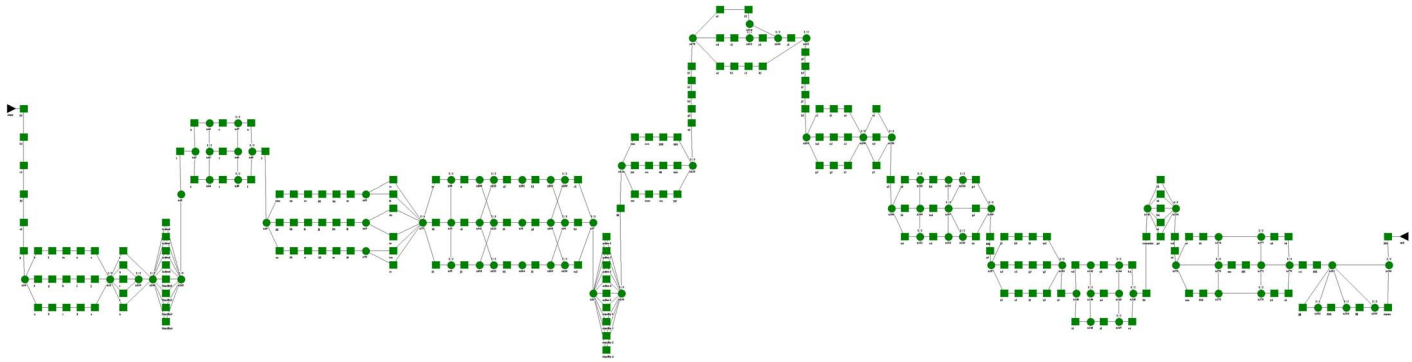

Figure 3-3 - Complex Model

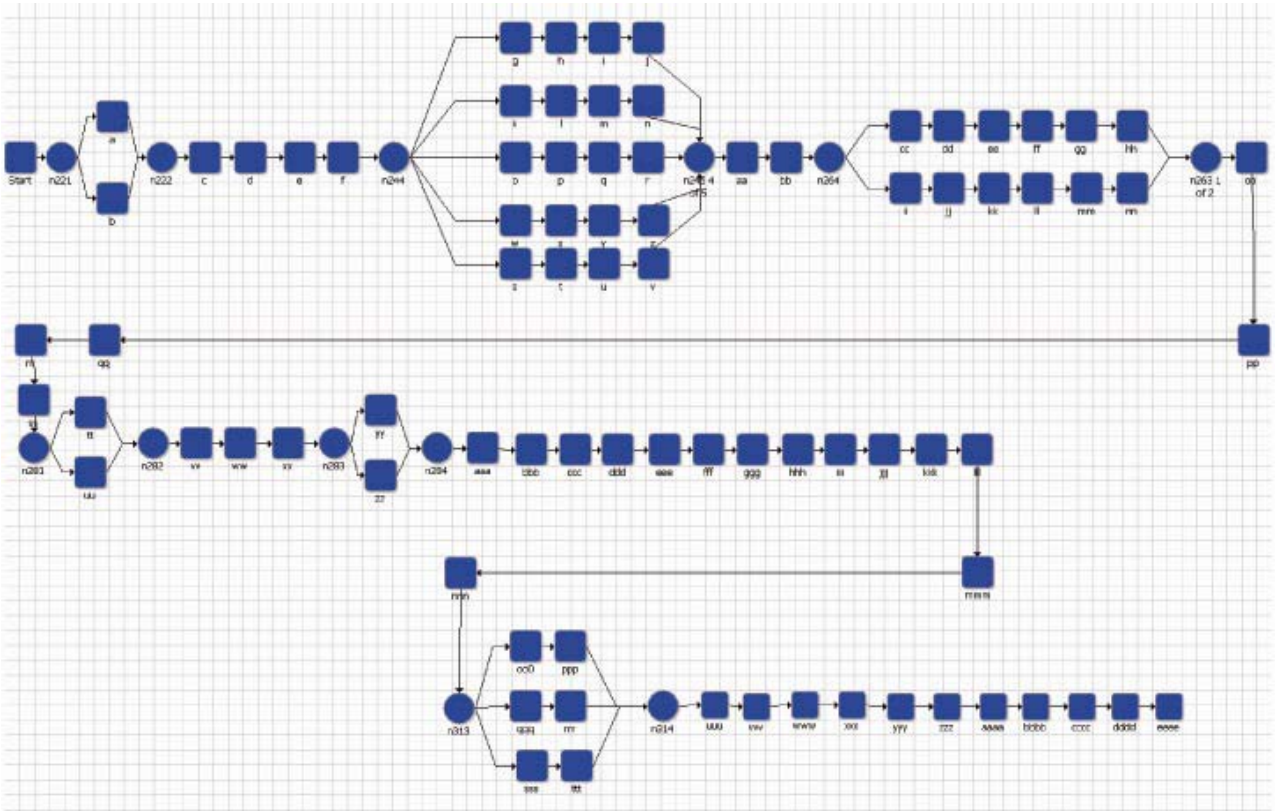

Figure 3-4 - Large Exponential Model

key parameters from the simulations, Reliability at the end of simulation time (for non-maintainable systems), Availability (for maintainable systems) and Mean Time to First Failure or similar measure for non-repairable systems vary by small to surprisingly large numbers, especially for MTTFF and MTBF. The Large model was a no maintenance model. Reporting MTTFF and Mean Down Time are reasonable parameters for this model. However, one of the packages reported a MTBF and MTTR, which are inappropriate. Note that the Availability for this model is actually the percentage of the simulation time until first failure. In this case, the availability for one of the packages is 85.8% of 61362 hours.

To extract some of the parameters from the packages requires an intimate knowledge of how they work, and what tool within the package will provide the parameter desired. One of the packages requires simulating a second time to one failure to produce an MTTFF.

The values produced for the Large model show some curious behavior. For example, as the number of trials is increased, one software package had decreasing results and the other had increasing results. The implication is that the software packages are iterating from a different direction. The difference in the MTTFF value raises serious concerns. The difference is greater than 40%! The difference in reliability is just 5% and the difference in availability is 2%. Care must be taken in interpreting or using any or all of these values. The differences between Raptor and BlockSim vs. Relex need to be investigated for differences in how models are built and interpreted.

The packages provide statistical measures of all or some of the calculated values. When running a large number of runs, the Standard Error of the Mean can be used to show the range of the mean reliability. Using +/- 3 SEM will give you a good estimate of the range.

				Software Package		
Model	**Parameter**	**Trials or Runs**	**Time (hours)**	**Raptor**	**BlockSim**	**Relex**
One Block	Reliability	1,000	1,000	0.3797	0.3663	0.365
One Block	Availability	1,000	1,000	0.8927	0.8894	0.8430
Simple	Reliability	1,000	100	0.983	0.977	0.978
Simple	Availability	1,000	100	0.9955	0.9892	0.978
Simple	System Failures	1,000	100	0.017	0.023	Not Reported
Large	Reliability	10,000	61,362	0.7024	0.737	0.6914
Large	Reliability	1,000	61,362	0.718	0.729	0.707
Large	Availability	1,000	61,362	0.858	0.861	0.691
Large	Availability	10,000	61,362	0.847	0.865	0.6866
Large	MTTFF: (Hours)	10,000	61,362	144,775.992	201,679.125	146,321.53
Complex	Reliability	10,000	100	0.1313	0.1215	0.0988
Complex	Availability	10,000	100	0.3877	0.3741	0.3333
Complex	MTBF (MTBDE) (Hours)	10,000	100	36.2732	37.2032	33.92
Complex	MTTR (MDT) (Hours)	10,000	100	68.3853	62.2399	74.51

Table 4-1- Results of Simulations

5 OBSERVATIONS AND CAUTIONS

Building models and entering data is a human activity subject to human reliability problems. We all fell into the abyss and made errors in connecting boxes, entering data, and setting up simulations. If we weren't comparing the results of several software packages, these errors may have gone undetected. This error rate points to one important caution. If the results of the RBD analysis are critical to a decision making process, and not just for information, it is important that a redundant analysis path be developed to assure the results are correct within the limitations of the software, and not a product of erroneous modeling. We offer three approaches. (1) Have two analysts independently model the design using the same software package. (2) Have a second analyst review the first analyst's work in detail, including all modeling decisions and data entries. (3) Have one analyst use two different packages for modeling.

Many times the results of these simulations are used to demonstrate compliance with a specified reliability or availability requirement. A result that would show a Reliability of 0.85 when the requirement was 0.90 might cause redesign, request for waiver, or other action to address the shortfall. However, the shortfall may be due to the parameters used for the simulation, the algorithms used by the software, a lack of understanding of how long to simulate, how many independent random number streams to use, and/or how many runs to use. Analytical solutions for highly complex models are based on approximations and simulations produce statistics which represent the results of multiple simulation runs.

The various programs do not necessarily describe variables in the same manner. When using the Lognormal distribution for example, we encountered a difference in terminology between Raptor and BlockSim. Raptor allows the Lognormal to be entered as Mean and Std. Dev. or Mu and Sigma. BlockSim only uses Mean and Std. Dev., but this is the same as Raptor's Mu and Sigma. A novice could waste a

great deal of time clarifying what needs to be entered as data.

Modeling special cases can be difficult because of the way the programs handle standby (which was in our models) and phasing (which was not in our models).

The output parameters were not consistently labeled, and the user should understand the difference between MTTF, MTTFF, MTBDE, and MTBF for reliability and MDT and MTTR for maintainability. The products also provide reliability and availability results with various adjectives such as “mean”, “point”, “conditional”, etc. A review of the literature provided with the packages is necessary to understand these terms and relate them to those found in specifications, handbooks, references, and texts. It is a serious issue that there doesn’t appear to be standard and/or consistent terminology and notation from one program to another as well as to standard literature in the field.

Each of the packages have tabs, checkboxes, preferences, defaults, multiple random number streams, selectable seeds for random numbers, etc to facilitate the modeling, analysis, and simulation process. However, this flexibility can provide huge pitfalls to the analyst. Care in modeling, and use of support services provided by the software supplier is a good practice. Each of the authors worked with the software package each was most familiar with. Despite this familiarity, numerous runs and reruns were necessary due to idiosyncrasies of the software, as well as errors in modeling, confusion of parameter definition, etc. Simple RBDs (parallel-series combinations of Exponential failure rate blocks without maintenance) are not the issue here. The problems compound as a variety of failure distributions are intermixed with a similar grouping of repair distributions. As these become more complex, a simulation becomes mandatory.

Some additional observations and cautions are given in the paragraphs that follow.

The models can run quickly even on old Pentium II PCs, or they can take hours to run. Length of simulation time, number of runs, and failure rate of the system can all contribute to lengthening of simulation time. One of the models took in excess of 1 hour on a 3 GHz Pentium IV.

Convergence of the results is heavily dependent on how consistent the block failure rates are. For example, one block with an MTBF of 1,000 hours, can double or triple simulation time in a system where the other blocks have MTBFs in the 100,000 hour range. The display during simulation on some of the packages shows the general trend, but there can be a lot of outliers.

The display of Availability and or Reliability during simulation can be useful for seeing how the simulation is behaving. For most models, this rapidly stabilizes to the first decimal place, and then the second decimal place tends to bounce around. Usually you get the first two significant figures in a hundred runs.

One of the models was so complex that it failed to converge on one of the packages – again this may have been due to a subtle preference selection (or non-selection) or a human error.

We have the impression that most of the user interfaces were designed by software designers, working with R&M engineers. The problem is that we seem to have gotten what an R&M engineer would tell someone never having used the product, not the interface he would like after he becomes familiar with the product. For example, double-clicking and working through multiple tabs to put data into blocks in a block diagrams is very modern. Sometimes an alternative method using tables of properties is easier to use even if it doesn't let you create blocks or change probability distributions.

BIOGRAPHIES

Aron Brall
SRS Technologies, Mission Support Division
NASA Goddard Space Flight Center
Code 302.9, Building 6
Greenbelt, Maryland 20771 USA

abrall@pop300.gsfc.nasa.gov

Aron Brall is the Reliability Team Lead at NASA Goddard Space Flight Center for SRS Technologies. Previously he held several positions in Product Assurance, including Vice President of Quality, in 14 years at Landis Grinding Systems, a Division of UNOVA Industrial Automation Systems. Prior to that he worked 12 years for the Amecom Division of Litton Systems as a Systems Effectiveness Project Engineer. Out of thirty-nine years professional experience, thirty-two have been in Reliability and Product Assurance. He received a BS in Electrical Engineering in 1967 from Columbia University, NY, NY, and an MBA in 1987 from Loyola College, Baltimore, MD. He is a senior member of ASQ, IEEE, and SME and a member of the SRE and SAE. He is an ASQ Certified Reliability Engineer. He is a contributing member of the committees that prepared the initial and revised editions of SAE M-110, Reliability and Maintainability Guideline for Manufacturing Machinery and Equipment. He is also a member of the RAMS 2007 Management Committee.

William Hagen
Ford Motor Company, Powertrain Manufacturing Engineering
R&M Section, Global Engineering Alignment
36200 Plymouth Road T3A
Livonia, MI 48150 USA

e-mail: whagen2@ford.com

Mr. Hagen has worked in manufacturing equipment Reliability and Maintainability at Ford Motor Company’s Powertrain Division for the last 11 years. He received a Bachelor of Science degree in Electrical Engineering from Michigan State University in 1978, and has undertaken graduate studies in computer science at Worcester Polytechnic Institute. His background in Reliability engineering includes four years as a reliability engineer working on sonar systems at Raytheon Submarine Signal Division and thirteen years on facsimile, communication, electronic support measures and spacecraft at Litton Industries Amecom Division.

Hung Tran
SRS Technologies, Mission Support Division

NASA Goddard Space Flight Center
Code 302.9, Building 6
Greenbelt, Maryland 20771 USA

htran@pop300.gsfc.nasa.gov

Hung Tran has over 6 years of reliability engineering experience in relation to unmanned and manned spacecraft systems. Currently he works as Reliability Engineer at NASA Goddard Space Flight Center for SRS Technologies. Previously, he held positions as Reliability Engineer and Risk Management at NASA Goddard Space Flight Center for MEI Technologies and at NASA Lyndon B. Johnson Space Center for GHG Corporation. He graduated from University of Houston with Bachelor of Science in Mathematics. He has extensive experience with utilizing computation tools such as Rapid Availability Prototyping for Testing Operational Readiness (RAPTOR), System Analysis Programs for Hands-On Integrated Reliability Evaluation (SAPHIRE), and Relex Reliability Software.