# Appendix D Programs for Reliability Modeling and Analysis

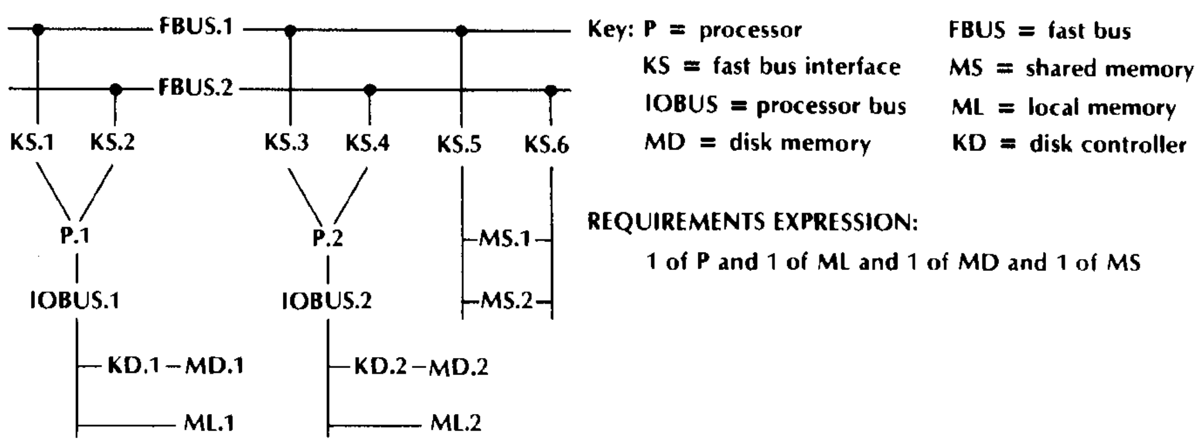
Software Tools →
- Combinatorial
- Markov
- Mathematical S/W Systems

Combinatorial Models accept:

1. Graphs of system component interconnections. Stochastic failures usually homogeneous
2. An intermediate representation that is constructed from the interconnections structure and functionality requirements (FEMA)

Example: CARE II (Computer-Aided Reliability Estimation II) developed for NASA by Raytheon, versatile but doesn't provide for repair. The inputs are the reliability parameters for modules within each stage and a description of the coverage detection and recovery mechanisms. The output includes system reliability/unreliability, MTTF, mission time.

Adviser – see Siewiorek textbook for actual examples (chapter on Evaluation Criteria)
Inputs: an interconnection graph, the reliability or availability of system components and a simple statement of system functionality requirements.




Combinatorial Models assume:

1. You can not get better with a failure.
2. You are either working or failed.
3. Individual module A(t) must be statistically independent → only one allowable repair strategy - system continues to work while the failed module is repaired.

Part-Count Models – models using a MIL-HDBK-217 database (component failure rates based on environment/operational factors).

Reliability Block Diagram Models (RBD) – similar to part-count models but based on tie-set or cut-set algorithms.

Reliability Fault Tree Models (FT) – similar to RBD but inputs based on modes of failure and a system model of these modes. Again computations based on tie-sets or cut-sets and approximations are frequently incorporated.

D2.4 Markov Models

If components are not decoupled (independent) → Markov Models
Techniques similar to Markov Processes (differential equations) as described in class including such things as numerical solutions to differential equations and/or Laplace transforms, etc.

Some Markov Models are built from a simulation program of the system under analysis which provides a lot of flexibility. Complex systems results in complex Markov Models that can only be handled by a computer, especially after the first failure scenario (exponentially increasing possibilities after 1st failure, 2nd failure, etc.)

Examples:
ARIES – **A**utomated **R**eliability **I**nteractive **E**stimation **S**ystem
CARE III – time varying Markov Modeling
SAVE – System Availability Estimator
HARP – Hybrid Automated Reliability Predictor
SHARPE – uses analytical techniques both with and without repair
SURE – **S**emi Markov **U**nreliability **R**ange **E**stimator

Monte Carlo Simulation – when the reliability of systems are too complex for analytical models → simulate their performance by randomly distributing the module states (error and failure distributions), executing the simulation many times while varying the module 'error' states for each execution cycle and then examining the results (actually the distribution of the results)

## Monte Carlo Simulation

Method is based on component failure distributions using actual failure data. Each component of the system is provided with a failure distribution fitted using real failure data. This is the time-to-failure of the component with known variations (confidence level) producing an estimate of the component's reliability.

If the exact values of the reliability characteristic of the system components were known, one could easily calculate the system reliability using such system reliability functions as the series and parallel reliability formulations.

Since there is no failure data for the system as a whole, it is difficult to find the distribution of all the component transformed random variables that lead to the system reliability function. The Monte Carlo approach is a practical tool for solving problems associated with estimating complex system reliability.

Steps for constructing the lower confidence limit for a complex system reliability:

1. For each component of the system, obtain an estimate of the component reliability $R_i$ (i = 1, 2, 3, ….. n where n is the number of components in the system) generating it from the respective estimate distribution

2. Calculate the corresponding estimate of system reliability
   $R_{sys} = f(R_1, R_2, \ldots R_n)$ where f ( ) is the system reliability function

3. Repeat Steps 1 and 2 a sufficiently large number of times (e.g. 1000) to obtain a large sample of $R_{sys}$

4. Using the sample obtained and a chosen confidence level (95% or 99% common), construct the lower confidence limit for the system reliability of interest as a sample percentile of level

**D2.5** – Mathematical S/W Systems

1. Matlab
2. Mathematica
3. Mathcad
4. Macsyma
5. Maple

Table D1 on page 509 lists the information sources for these commercially available mathematical programs. Essentially the slide rule for today's student/ practitioner which should be in everybody's tool box/tricks of the trade.

D4 (page 513) has a list of reliability and availability software programs

**Prisim** - a follow on to the MIL-HDBK-217 efforts with greatly enhanced data input and analysis output capabilities (see following lecture notes).

**CARE** Modules – popular and used extensively

**D5** – Author's Example of Computer Analysis

Built a six-state Markov Model for a spacecraft system with one on-line module and two different standby modules which had a very low probability of failing while in standby (dormancy failure rates).

The Markov Model produced six differential equations of which the first four state probabilities were easily checked but the $5^{th}$ and $6^{th}$ state equations required considerable effort; in fact, the $6^{th}$ state could only be verified numerically given the system's failure rates as inputs.

Author recommends that it is wise to check all results either of two ways:

1. Using two different modeling programs OR
2. Using an analytical solution (sometimes an approximate numerical solution) as well as a modeling program

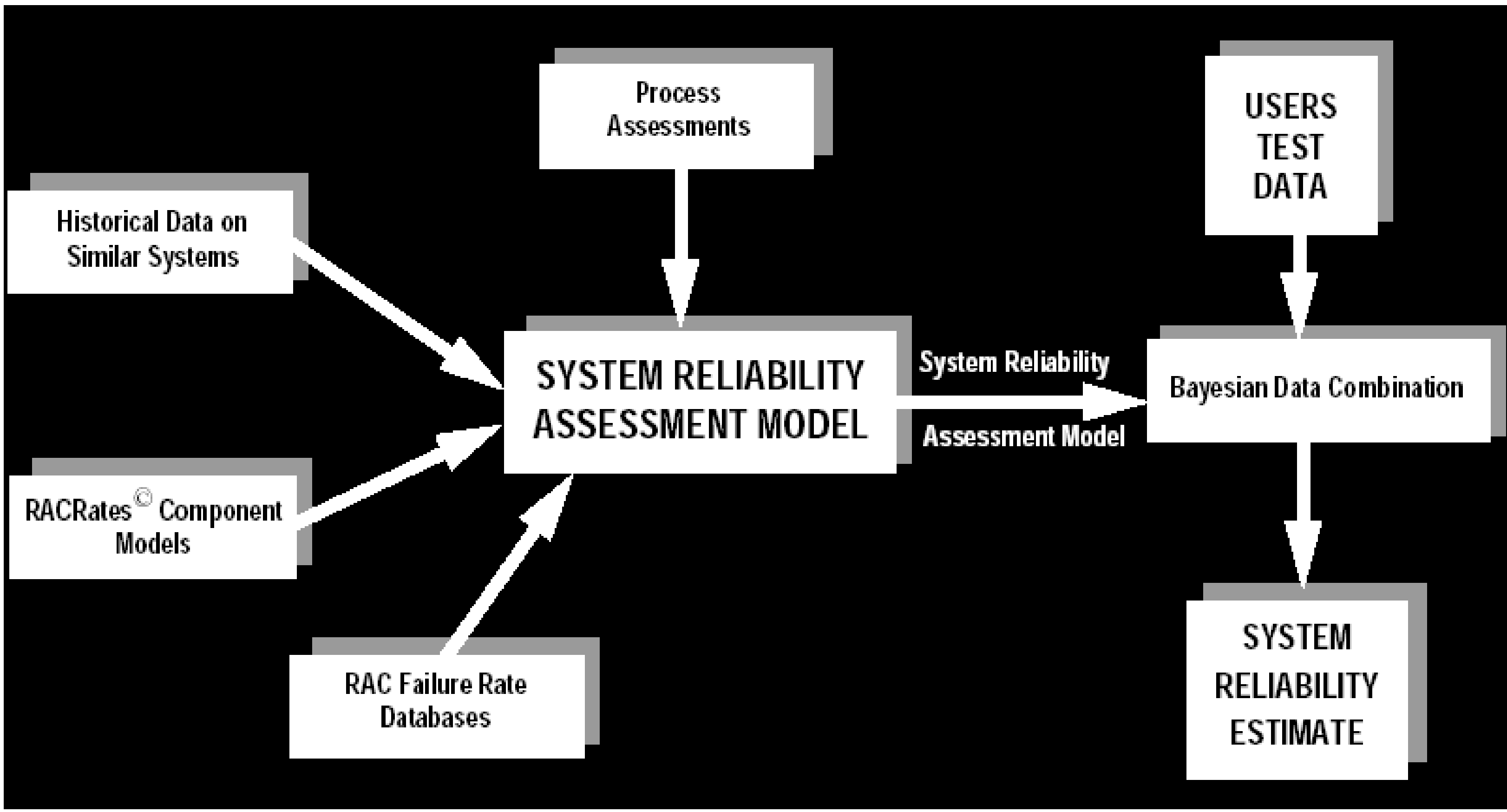
# Comparison of PRISM vs. MIL-HDBK-217 Hardware MTBF Calculations for the Shuttle Cockpit Avionics Upgrade (CAU) Program

Some lecture material courtesy of
Barry Ives
LMSI-Owego Reliability Engineer

# PRISM vs. MIL-HDBK-217

- PRISM® is a Reliability Analysis Center software tool that ties together several tools into a comprehensive system reliability prediction methodology. The PRISM concept accounts for the myriad of factors that can influence system reliability, combining all those factors into an integrated system reliability assessment resource.
- RAC is in Rome, NY now the Reliability Analysis Information Center
- PRISM was developed to overcome inherent limitations in MIL-HDBK-217 that is no longer being actively maintained or updated by the Department of Defense (DoD).
- PRISM calculated the Shuttle Program's CAU Hardware MTBF's which were different (better) from MIL-HDBK-217 calculated MTBF's due to three levels of calculation differences:
  - Component level
  - Subassembly level
  - Mission Profile level

# PRISM Methodology Model



## Why the differences?

- PRISM was developed by the Reliability Analysis Center based on the following identified industrial needs:
  - To develop a method to calculate the effects of non-component variables, considered to be major contributors to reliability:
    - Design deficiencies
    - Manufacturing defects
    - Poor system management
    - Wear-out mechanisms
    - Software failures
    - Induced failures
    - No defect-found failures

## Why the differences?
## Component Level

- At the component level, separate failure rates were developed for each class of failure mechanisms:
  - Operational stresses
  - Environmental stresses
  - Stresses due to cycling temperature and/or power
  - Induced stresses
  - Solder joint stresses
- Empirical component failure rates modified by acceleration factors
  - Based on analysis of RAC data base stress data
  - Analysis of categories of failure modes/failure causes
  - Failure analysis of parts failed in the field

# Why the differences?
# Assembly/System Level

- PRISM System Model based on the premise that failure rates which are attributable to predominant system-level failure causes can be quantified.
- RAC conducted survey to determine the percentages of failures for each cause.
- Survey data statistically analyzed to develop quantified assembly-level factors
- User can duplicate the survey in their own facility
  - PRISM calculates new Process Grade Factors based on results of surveys
  - Can be better or worse than average Process Grade Factors
  - Quantifies the capability of an organization to mitigate failures for each cause

# MIL-HDBK-217 component failure rate calculation example

- Linear Oscillator, G4 (failure rate = .006733 failures/million hours)
- Formula:

$$\lambda_p = (C1\pi_t + C2\pi_e)*\pi_q*\pi_l$$

- Where:

$\lambda_p$ = component failure rate

C1 = 0.01 based on number of transistors

$\pi_t$= 2.596173 based on Junction Temp of 69 degrees C and Linear IC model

C2 = .001939 based on formula C2 = 2.8E-4 * N^1.08 (N = number of pins)

$\pi_e$ = 0.5 for Space Flight environment

$\pi_q$ = 0.25 for Class S components (full M38510 requirements)

$\pi_l$ = 1 which is learning factor for components > or = 2 years in manufacture

# PRISM and RACRates component failure rate calculations

- PRISM component failure rates based on RACRates model if available.
- Where RACRates are not available, historical failure rate data is available for both electronic and non-electronic components.
- Empirical data can also be entered (PRISM uses Bayesian math to incorporate )
- RACRates models are based on the generic formula:

$$\lambda_p = \lambda o \pi o + \lambda e \pi e + \lambda c \pi c + \lambda i + \lambda sj \pi sj$$

Where:

$\lambda_p$ = calculated failure rate

$\lambda o$ = base **operational** failure rate

$\pi o$ = operational failure rate acceleration factor

$\lambda e$ = base **environmental** failure rate

$\pi e$ = environmental accelerational factor

$\lambda c$ = base **cycling** failure rate

$\pi c$ = cycling acceleration factor

$\lambda i$ = **induced** failure rate

$\lambda sj$ = **solder joint** failure rate

$\pi sj$ = solder joint acceleration factor

# PRISM and RACRates component failure rate calculation Example

- RACRates model for Linear Oscillator G4:
  - Resulting failure rate = .00186 failures/million hrs
    - Equal to approx. ¼ of the 217 failure rate calculated (Page 6)
  - Based on the specific formula:

$\lambda_p = \pi g * (\lambda ob*\pi dco*\pi to + \lambda eb*\pi dcn*\pi rht + \lambda tcb*\pi cr*\pi dt + \lambda eos) + \lambda sj$

Where:

$\pi g$ = reliability growth factor = .071576

$\lambda ob$ = base operational failure rate = .000013

$\pi dco$ = operational duty cycling factor = 3.571429

$\pi to$ = operational temperature cycling acceleration factor = 119.026820

$\lambda eb$ = non-op base failure rate = .001997

$\pi dcn$ = non-op duty cycle = 0.00000

$\pi rht$ = non-op temperature and humidity acceleration factor = .047781

$\lambda tcb$ = temperature cycling base failure rate = .000089

$\pi cr$ = temperature cycling acceleration factor = .002073

$\pi dt$ = delta temperature acceleration factor = .792094

$\lambda eos$ = electrical overstress failure rate = .0016

$\pi sj$ = failure rate due to solder joint stress = .001352

- $\pi$&$\lambda$ values are calculated, based on environmental and component data entered by user.

# Subassembly level calculation differences
# Failure rate models

- The MIL-HDBK-217 **assembly** failure rate is the **simple sum** of the failure rates of the components:
  - Current MIL-HDBK-217 predicted single computer failure rate to be 7.993698 per million hours.
    - MTBF of 125,099 hours at 54 degrees ambient, Space Flight environment
- The PRISM **assembly** failure rate is calculated, based on **Process Grade Factors** determined through surveys of facility management, manufacturing and engineering processes along with details of testing and modeling based on the actual environment, plus software failure rates, if applicable.
  - The current preliminary PRISM calculated single computer failure rate is 1.112561/million hours
    - Average Process Grade factors (PRISM defaults)
    - MTBF of 898,827 with similar assumptions
    - About seven times the current MIL-HDBK-217 calculated single computer MTBF

# Subassembly level calculation differences, continued
# The PRISM assembly failure rate model

- The PRISM **assembly** failure rate is based on the **PRISM System Reliability model:**
  - $\lambda p$ = ( Initial Failure Rate Assessment * Process Grade Factor )
    - ( plus Software Failure Rate, if applied, which can be calculated by PRISM )
    - Initial Failure Rate Assessment is the sum of the component failure rates (Pages 7 & 8 )
  - Process Grade Factor = $\Pi p \Pi im \Pi e + \Pi o \Pi g + \Pi m \Pi im \Pi e \Pi g + \Pi s \Pi g + \Pi i + \Pi n + \Pi w$

Where:

$\Pi p$ = parts process multiplier ( default = .243130 )

$\Pi im$ = infant mortality rate (ESS testing factor) ( default = .972205 )

$\Pi e$ = environmental factor = .152355

- Based on 54 degrees C Space Airborne environment with no temperature change for dormant state, 40% relative humidity and no vibration, and 100% duty cycle.

$\Pi d$ = design process multiplier ( default = .094085 )

$\Pi g$ = reliability growth fact (default = 1)

$\Pi m$ = manufacturing process multiplier ( default = .142422 )

$\Pi s$ = system management process multiplier ( default = .036012 )

$\Pi i$ = induced process multiplier ( default = .141194 )

$\Pi n$ = no-defect process multiplier (NTFs) ( default = .237019 )

$\Pi w$ = wear-out process multiplier ( default = .107730 )

Note: Default factors are industrial averages based on surveys done by RAC.

# Mission Profile level calculation differences

- In the CAU program, an Average **Mission** MTBF is calculated, based on the details of a "typical mission":
  - Average MTBF = Weighted summation of failure rates for the various phases
  - Considers the on–time of varying quantities of equipment during different phases
  - Considers the various environmental factors present during different phases -
    - Ground operation
    - Missile launch
    - Space flight
    - Missile re-entry
    - Ground mobile phase after landing
- PRISM failure rates at different environments display much less variation due to environment change than MIL-217 values (if temperature is constant). Example:
    - MIL 217 Single Board Computer (SBC) failure rate at Missile Launch = 47.19223 per million hours
    - MIL 217 SBC failure rate at Space Flight = 7.993698
    - PRISM SBC failure rate at Missile Launch = 2.28591
    - PRISM SBC failure rate at Space Flight = 1.112561
    - This is a Ratio of to 2:1 for PRISM rates compared to 6:1 for MIL-217.
  - Average Mission MTBF for the SRU's and the LRU's will show less change due to mission phase environment using PRISM than using MIL 217.