

NASA/SP-2011-3421
Second Edition
December 2011

Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners



NASA Scientific and Technical (STI) Program ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing help desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

Access the NASA STI program home page at <http://www.sti.nasa.gov>

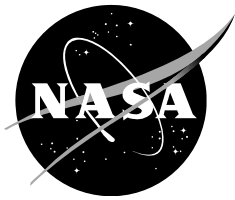
E-mail your question via the Internet to help@sti.nasa.gov

Fax your question to the NASA STI Help Desk at 443-757-5803

Phone the NASA STI Help Desk at 443-757-5802

Write to:
NASA STI Help Desk
NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320

NASA/SP-2011-3421



Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners

NASA Project Managers:

**Michael Stamatelatos, Ph.D., and
Homayoon Dezfuli, Ph.D.**

*NASA Headquarters
Washington, DC*

Second Edition

December 2011

Acknowledgments

The individuals responsible for this document, who managed this project, and were also authors are:

Michael Stamatelatos, NASA Headquarters (HQ), Washington, DC, and
Homayoon Dezfuli, NASA HQ, Washington, DC

The following individuals, listed in alphabetic order, are principal contributors to the present and/or previous edition of this document:

George Apostolakis, previously at Massachusetts Institute of Technology (MIT), now at United States Nuclear Regulatory Commission (NRC)

Chester Everline, NASA Jet Propulsion Laboratory (JPL)

Sergio Guarro, Aerospace Corporation

Donovan Mathias, NASA Ames Research Center (ARC)

Ali Mosleh, University of Maryland (UMD)

Todd Paulos, Alejo Engineering

David Riha, Southwest Research Institute

Curtis Smith, Idaho National Laboratory (INL)

William Vesely, NASA HQ

Robert Youngblood, INL

Additional contributors to this or the previous version of this document are: Harold Blackman, Ron Boring, and David Gertman, INL; Scott Dixon and Michael Yau, ASCA Inc.; Parviz Moieni, Southern California Edison; Hamed Nejad, Science and Technology Corp.; Pete Rutledge, Quality Assurance & Risk Management Services; Frank Groen and Faith Chandler, NASA HQ; Ken Gee, ARC; Susie Go, ARC; Scott Lawrence, ARC; Ted Manning, ARC; Patrick McCabe and Kurt Vedros, INL; and Shantaram Pai, Glenn Research Center.

Reviewers who provided comments on the drafts leading up to this revision are: Allan Benjamin and Christopher Everett, Information Systems Laboratories; Tim Barth, NASA Engineering and Safety Center (NESC); Mark Bigler, Johnson Space Center (JSC); Michael Blythe, NESC; Roger Boyer, JSC; Alfredo Colón, NASA HQ; Charles Ensign, Kennedy Space Center (KSC); Amanda Gillespie, KSC; Teri Hamlin, JSC; Curtis Larsen, JSC; Mike Lutomski, JSC; Mark Monaghan, KSC; Bruce Reistle, JSC; Henk Roelant, JSC.

Document available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320
443-757-5802

National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312
703-605-6000

Contents

Acknowledgments	i
Acronyms and Abbreviations	xviii
1. Introduction	1-1
1.1 Purpose and Scope of This Procedures Guide	1-2
1.2 Knowledge Background.....	1-3
1.3 Application Recommendation	1-3
1.4 References	1-3
2. Risk Management.....	2-1
2.1 Definition of Risk.....	2-1
2.2 Risk Management at NASA.....	2-2
2.2.1 Risk-Informed Decision Making Process (RIDM)	2-4
2.2.2 Continuous Risk Management (CRM).....	2-7
2.3 References	2-11
3. Probabilistic Risk Assessment Overview	3-1
3.1 Historical Background.....	3-1
3.1.1 Design Basis Evaluation vs. Risk Evaluation	3-1
3.1.2 From Regulation Based on Design Basis Review to Risk-Informed Regulation	3-2
3.1.3 Summary of PRA Motivation	3-3
3.1.4 Use of PRA in the Formulation of a Risk-Informed Safety Case (RISC).....	3-4
3.1.5 Management Considerations	3-4
3.2 Example.....	3-5
3.2.1 Propellant Distribution Module Example	3-5
3.2.2 Selected Results	3-6
3.2.3 High-Level Application of Results.....	3-8
3.2.4 Summary	3-9
3.3 Elements of PRA	3-10
3.3.1 Identification of Initiating Events.....	3-11
3.3.2 Application of Event Sequence Diagrams and Event Trees.....	3-13
3.3.3 Modeling of Pivotal Events	3-17
3.3.4 Quantification of (Assignment of Probabilities or Frequencies to) Basic Events ..	3-19
3.3.5 Uncertainties: A Probabilistic Perspective	3-21
3.3.6 Formulation and Quantification of the Integrated Scenario Model	3-23
3.3.7 Overview of PRA Task Flow.....	3-25
3.4 Summary	3-26
3.4.1 Current State of Practice	3-26
3.4.2 Prospects for Future Development.....	3-27
3.5 References	3-27
4. Scenario Development	4-1
4.1 System Familiarization	4-1
4.2 Success Criteria	4-3

4.2.1	Mission Success Criteria	4-3
4.2.2	System Success Criteria	4-4
4.3	Developing a Risk Model	4-5
4.3.1	IE Development	4-7
4.3.2	Accident Progression	4-10
4.3.3	Fault Tree Modeling	4-17
4.4	References	4-20
5.	Data Collection and Parameter Estimation	5-1
5.1	PRA Parameters	5-1
5.2	Sources of Information	5-3
5.2.1	Generic Data Sources	5-3
5.2.2	System-Specific Data Collection and Classification	5-5
5.3	Parameter Estimation Method	5-9
5.4	Prior Distributions	5-10
5.5	Selection of the Likelihood Function	5-11
5.6	Development of the Posterior Distribution	5-12
5.7	Sequential Updating	5-15
5.8	Developing Prior Distributions from Multiple Sources of Generic Information	5-15
5.9	Guidance for Bayesian Inference Calculations	5-16
5.10	References	5-16
6.	Uncertainties in PRA	6-1
6.1	The Model of the World	6-1
6.2	The Epistemic Model	6-2
6.3	A Note on the Interpretation of Probability	6-3
6.4	Presentation and Communication of the Uncertainties	6-7
6.5	The Lognormal Distribution	6-8
6.6	Assessment of Epistemic Distributions	6-10
6.6.1	Bayes' Theorem	6-10
6.6.2	A Simple Example: The Discrete Case	6-11
6.6.3	A Simple Example: The Continuous Case	6-12
6.6.4	Conjugate Families of Distributions	6-15
6.7	The Prior Distribution	6-17
6.8	The Method of Maximum Likelihood	6-18
6.9	References	6-19
7.	Modeling and Quantification of Common Cause Failures	7-1
7.1	Importance of Dependence in PRA	7-1
7.2	Definition and Classification of Dependent Events	7-1
7.3	Accounting for Dependencies in PRAs	7-2

7.4	Modeling Common Cause Failures	7-4
7.5	Procedures and Methods for Treating CCF Events.....	7-6
7.6	Preliminary Identification of Common Cause Failure Vulnerabilities (Screening Analysis).....	7-6
7.6.1	Qualitative Screening	7-6
7.6.2	Quantitative Screening	7-8
7.7	Incorporation of CCFs into System Models (Detailed Analysis)	7-10
7.7.1	Identification of CCBEs	7-10
7.7.2	Incorporation of CCBEs into the Component-Level Fault Tree	7-11
7.7.3	Development of Probabilistic Models of CCBEs	7-13
7.7.4	Estimation of CCBE Probabilities	7-15
7.8	Generic Parameter Estimates	7-16
7.9	Treatment of Uncertainties	7-17
7.10	References	7-18
8.	Human Reliability Analysis (HRA)	8-1
8.1	Basic Steps in the HRA Process	8-1
8.2	Classifications of Human Interactions and Associated Human Errors.....	8-3
8.2.1	Pre-Initiator, Initiator, and Post-Initiator HSIs	8-3
8.2.2	Skill, Rule, and Knowledge-Based Response	8-3
8.2.3	Error of Omission and Error of Commission.....	8-4
8.3	General Modeling of Pre-Initiator, Initiator, and Post-Initiator HSIs in a PRA.....	8-4
8.4	Quantification of Human Interactions (or Errors).....	8-4
8.4.1	Qualitative Screening	8-5
8.4.2	Quantitative Screening	8-6
8.5	HRA Models	8-6
8.5.1	Technique for Human Error Rate Prediction (THERP).....	8-6
8.5.2	Cognitive Reliability and Error Analysis Method (CREAM)	8-11
8.5.3	Nuclear Action Reliability Assessment (NARA).....	8-15
8.5.4	Standard Plant Analysis Risk HRA Method (SPAR-H).....	8-18
8.6	Guidelines on Uses of HRA Models	8-21
8.7	HRA Examples	8-22
8.7.1	Example for a Post-Initiator HSI	8-22
8.7.2	Example for a Pre-Initiator HSI.....	8-25
8.8	References	8-28
9.	Software Risk Assessment.....	9-1
9.1	Concept of Software Risk and Related Definitions.....	9-2
9.1.1	Basic Definitions.....	9-3
9.1.2	Software Defects and Software Failures	9-3
9.2	Lessons Learned from Software Failures in Space Systems	9-5
9.3	Classification of Software Failures for Risk Modeling.....	9-8

9.3.1	Conditional vs. Unconditional Failures	9-8
9.3.2	Recoverable vs. Mission-critical Failures	9-9
9.4	Context-based Software Risk Model (CSRM)	9-10
9.4.1	Conceptual Formulation	9-10
9.4.2	Key Objectives and Characteristics of CSRM Application	9-12
9.4.3	Application Process	9-15
9.4.4	Examples of Application	9-17
9.4.5	CSRM Modeling Detail and Representation of Software Failure Modes	9-31
9.4.6	Software Risk Quantification	9-33
9.5	Use of Software Risk Information	9-39
9.5.1	Conditional Scenarios and Risk-informed Software Testing Strategies	9-39
9.5.2	Integration of Results into Pre-existing PRA Models	9-40
9.6	Definitions	9-41
9.7	References	9-42
10.	Physical and Phenomenological Models	10-1
10.1	Role of Phenomenological Methods in Risk Assessment	10-2
10.2	Phenomenological Modeling During the Design Process	10-2
10.3	Stress-Strength Formulation of Physical Models	10-4
10.4	Range Safety Phenomenological Models	10-6
10.4.1	Inert Debris Impact Models	10-7
10.4.2	Blast Impact Models	10-8
10.4.3	Re-Entry Risk Models	10-12
10.5	MMOD Risk Modeling	10-14
10.5.1	Risk from Orbital Debris	10-14
10.5.2	MMOD Risk Modeling Framework	10-14
10.5.3	Probability of MMOD Impact P_I	10-15
10.5.4	Probability of MMOD Impact Affecting Critical SV Components, $P_{C/I}$	10-15
10.5.5	Probability of Critical Component Damage, $P_{D/C}$	10-16
10.6	Ground-Based Fire PRA	10-16
10.7	A Launch Vehicle Ascent Abort Model	10-23
10.8	Summary	10-24
10.9	References	10-24
11.	Probabilistic Structural Analysis	11-1
11.1	Basic Concepts of Probabilistic Structural Analysis	11-1
11.2	Probabilistic Structural Response Modeling	11-2
11.2.1	Limit State Formulation	11-2
11.2.2	Assigning Uncertainty Models to Random Variables	11-4
11.3	Stress Versus Strength Modeling	11-4
11.3.1	Normal Distributions	11-5
11.3.2	Lognormal Distributions	11-6
11.4	Monte Carlo Simulation and Most Probable Locus Approaches	11-8

11.5	Probabilistic Finite Element Approaches	11-14
11.5.1	When Probabilistic Finite Element Analysis is Needed	11-14
11.5.2	Mapping Random Variables to Finite Element Input	11-14
11.6	Probabilistic Fracture Mechanics.....	11-15
11.6.1	Differences of Probabilistic Fracture Mechanics	11-16
11.6.2	When Probabilistic Fracture Mechanics is Needed	11-17
11.6.3	Probabilistic Characterization of Input Variables	11-17
11.7	Probabilistic Structural Analysis Examples.....	11-19
11.7.1	Example of a Probabilistic Stress versus Strength Analysis	11-19
11.7.2	Example of a Probabilistic Finite Element Analysis.....	11-21
11.8	References	11-23
12.	Uncertainty Propagation.....	12-1
12.1	Problem Statement for Uncertainty Propagation	12-2
12.1.1	How Does Sampling Work?	12-3
12.1.2	Crude Monte Carlo Sampling	12-4
12.1.3	Latin Hypercube Sampling	12-4
12.2	Achieving Convergence.....	12-5
12.3	Example: Uncertainty Propagation for an Accident Scenario Using LHS.....	12-6
12.4	Treatment of Epistemic Dependency	12-12
12.5	Epistemic Uncertainty in Phenomenological Models.....	12-13
12.6	References	12-15
13.	Presentation of Results	13-1
13.1	Graphical and Tabular Expression of Results	13-2
13.2	Communication of Risk Results.....	13-3
13.2.1	Displaying Epistemic Uncertainties	13-3
13.2.2	Displaying Conditional Epistemic Uncertainties	13-4
13.2.3	Displaying Aleatory and Epistemic Uncertainties	13-6
13.3	Importance Ranking	13-10
13.3.1	Importance Measures for Basic Events Only	13-11
13.3.2	Differential Importance Measure for Basic Events and Parameters.....	13-13
13.3.3	Example of Calculation of Importance Rankings.....	13-15
13.4	Sensitivity Studies and Testing Impact of Assumptions	13-19
13.4.1	Impact of Modeling Assumptions	13-19
13.4.2	Analysis of Impact of Hardware Failure Dependence	13-19
13.5	References	13-20
14.	Launch Abort Models	14-1
14.1	Abort Assessment overview	14-1
14.2	Evolution of the Abort Risk Assessment with Program Phases.....	14-2
14.3	Abort Assessment Process Overview	14-3
14.4	Abort Failure Initiators	14-5

14.5	Failure Initiator Propagation & Detection	14-6
14.5.1	Failure Propagation	14-6
14.5.2	Failure Detection, Warning Time, and Abort Triggers	14-9
14.5.3	Failure Propagation and Detection Analysis Example	14-10
14.6	Failure environments	14-11
14.6.1	Explosion Environments	14-12
14.6.2	Blast Overpressure	14-12
14.6.3	Fragments	14-13
14.6.4	Fireball	14-14
14.7	Loss-of-Control Environments	14-14
14.7.1	Example of Failure Environment Quantification: Blast Overpressure	14-14
14.8	Crew module Capability & Vulnerability	14-17
14.8.1	Example of Crew Module Vulnerability Assessment	14-18
14.9	Integrated Abort Modeling	14-20
14.9.1	Integrated Modeling Evolution	14-21
14.9.2	Uncertainty and Sensitivity Analyses	14-22
14.9.3	Abort Model Review	14-22
14.9.4	Example of Integrated Modeling: Ares I Abort Assessment GoldSim Model	14-23
14.10	References	14-27
Appendix A – Probability and its Application to Reliability and Risk Assessment		A-1
A.1	The Logic of Certainty	A-1
A.1.1	Events and Boolean Operations	A-1
A.1.2	Simple Systems	A-4
A.1.3	Structure Functions	A-5
A.2	Probability Basics	A-8
A.2.1	Definition	A-8
A.2.2	Basic Rules	A-9
A.2.3	Theorem of Total Probability	A-10
A.2.4	Bayes' Theorem	A-10
A.3	Failure Distributions	A-12
A.3.1	Random Variables	A-12
A.3.2	Distribution Functions	A-13
A.3.3	Moments	A-15
A.4	References	A-17
Appendix B - Event Frequencies and Hardware Failure Models		B-1
B.1	Probability of Failure on Demand: The Binomial Distribution	B-1
B.2	Failure While Operating	B-2
B.3	The Exponential Distribution	B-3
B.4	The Weibull Distribution	B-5
B.5	Event Frequency: The Poisson Distribution	B-6
B.6	Unavailability	B-7

B.7	References	B-7
Appendix C - Bayesian Inference Calculations		C-1
C.1	Inference for Common Aleatory Models	C-1
C.2	Reference	C-14
Appendix D - Logic-Based PRA Modeling Examples		D-1
D.1	PRA Example 1 Problem Description	D-1
D.1.1	PRA Objectives and Scope	D-1
D.1.2	Mission Success Criteria	D-2
D.1.3	End States	D-2
D.1.4	System Familiarization	D-2
D.1.5	Initiating Events Development	D-4
D.1.6	Master Logic Diagram for IE Development; Pinch Points	D-5
D.1.7	Other IE Development Methods	D-8
D.1.8	IE Screening and Grouping	D-9
D.1.9	Risk Scenario Development	D-9
D.1.10	ESD Analysis	D-9
D.1.11	System Success Criteria	D-12
D.1.12	ET Analysis	D-13
D.1.13	FT Analysis	D-15
D.1.14	Data Analysis	D-20
D.1.15	Model Integration and Quantification	D-20
D.2	PRA Example 2 Problem Description	D-27
D.2.1	PRA Objectives and Scope	D-28
D.2.2	Mission Success Criteria	D-28
D.2.3	End States	D-28
D.2.4	System Familiarization	D-29
D.2.5	Initiating Events Development	D-31
D.2.6	Risk Scenario Development (Including ESD and ET Analysis)	D-31
D.2.7	Remaining Tasks	D-39
D.3	Reference	D-39
Appendix E - PRA Simulation Example		E-1

Figures

Figure 2-1. Implementation of the Triplet Definition of Risk in PRA.	2-2
Figure 2-2. Risk Management as the Interaction of Risk-Informed Decision Making and Continuous Risk Management. [NASA/SP-2010-576].	2-3
Figure 2-3. Flowdown of Performance Requirements (Illustrative).	2-4
Figure 2-4. The RIDM Process.	2-4
Figure 2-5. Uncertainty of Forecasted Outcomes for a Given Alternative Due to Uncertainty of Analyzed Conditions.	2-6
Figure 2-6. Performance Commitments and Risk Tolerances for Three Alternatives.	2-7
Figure 2-7. Decreasing Uncertainty and Risk over Time.	2-8
Figure 2-8. The CRM Process.	2-8
Figure 3-1. Simplified Schematic of a Propellant Distribution Module.	3-6
Figure 3-2. The Concept of a Scenario.	3-11
Figure 3-3. Typical Structure of a Master Logic Diagram (MLD).	3-12
Figure 3-4. Typical Structure of an Event Sequence Diagram (ESD).	3-13
Figure 3-5. Event Tree Representation of the ESD Shown in Figure 3-4.	3-14
Figure 3-6. ESD for the Hydrazine Leak.	3-16
Figure 3-7. ET for the Hydrazine Leak.	3-16
Figure 3-8. Revised ET for the Hydrazine Leak.	3-17
Figure 3-9. Fault Trees for Failure of Leak Detection and Failure of Isolation.	3-18
Figure 3-10. Exponential Distribution Model [$Pr_i(t) = 1 - \exp(-\lambda t)$ for $\lambda = 0.001$ per hour].	3-20
Figure 3-11. Application of Bayes' Theorem.	3-22
Figure 3-12. Propagation of Epistemic Uncertainties for the Example Problem.	3-24
Figure 3-13. A Typical PRA Task Flow.	3-25
Figure 4-1. Event Tree/Fault Tree Linking.	4-5
Figure 4-2. Notional Master Logic Diagram Related to Candidate Initiating Events Caused by Kinetic Energy.	4-8
Figure 4-3. The Elements of an Accident Scenario.	4-9
Figure 4-4. Typical Event Sequence Diagram.	4-12
Figure 4-5. Event Sequence Diagram Development (step 1).	4-13
Figure 4-6. Typical Event Sequence Diagram Development (step 2).	4-14
Figure 4-7. Event Tree Structure.	4-15
Figure 4-8. Event Tree Linking.	4-16
Figure 4-9. Typical Fault Tree Structure and Symbols.	4-18
Figure 5-1. Component Functional State Classification.	5-7
Figure 5-2. Failure Event Classification Process Flow.	5-8

Figure 5-3. Failure Cause Classification Subcategories.....	5-9
Figure 5-4. The Prior and Posterior Distributions of Example 4.....	5-14
Figure 5-5. The Prior and Posterior Distributions of Example 5.....	5-14
Figure 6-1. Representing the World via Bayesian Inference.....	6-5
Figure 6-2. The Probability Mass Function (pmf) of the Failure Rate λ	6-6
Figure 6-3. Aleatory Reliability Curves with Epistemic Uncertainty.....	6-7
Figure 6-4. Aleatory Reliability Curves with a Continuous Epistemic Distribution.....	6-8
Figure 6-5. The Lognormal probability density function (pdf).....	6-10
Figure 6-6. Discretization Scheme.....	6-13
Figure 6-7. Prior (Solid Line) and Posterior (Dashed Line) Probabilities for the Case of No Failures.....	6-15
Figure 7-1. Accounting for CCF Events Using the Beta Factor Model in Fault Trees and Reliability Block Diagrams.....	7-5
Figure 8-1. Basic Steps in the HRA Process.....	8-1
Figure 8-2. Initial Screening Model of Estimated Human Error Probability and Uncertainty Bounds for Diagnosis Within Time T of One Abnormal Event by Control Room Personnel.....	8-8
Figure 8-3. Example of Cassini PRA Fault Tree and Event Sequence Diagram Models.....	8-23
Figure 8-4. FCO's CDS Activation Time Cumulative Distribution Function.....	8-25
Figure 9-1. Software Defects by Development Phase.....	9-4
Figure 9-2. PA-1 Mission Sequence Illustration.....	9-18
Figure 9-3. PA-1 Mission Event-Tree for Identification of Key SW Functions.....	9-19
Figure 9-4. CSRM Entry-Point Events Identified in PA-1 PRA Event-Tree Model.....	9-20
Figure 9-5. CSRM Entry-Point Events Identified in PA-1 PRA Fault-Tree Model.....	9-20
Figure 9-6. DFM Model of Pa-1 GN&C System.....	9-21
Figure 9-7. DFM-Produced Cut-Set for Failure of Pa-1 GN&C Function.....	9-22
Figure 9-8. Mini AERCam Spacecraft and Thruster Arrangement.....	9-25
Figure 9-9. Mini AERCam Mission Event Tree.....	9-25
Figure 9-10. Top-Level DFM Model of the Mini AERCam System.....	9-26
Figure 9-11. Lower-Level DFM Model of the GN&C Sub-System.....	9-26
Figure 9-12. Lower-Level DFM Model of the Propulsion Sub-System.....	9-27
Figure 9-13. DFM Model for Illustration of SW Failure-Mode Representations.....	9-32
Figure 9-14. Notional Example of Expansion of Entry-point Event into CSRM Cut-Sets.....	9-41
Figure 10-1. Event Sequence Diagram for Attitude Control Malfunction at Lift-off.....	10-4
Figure 10-2. Probability Distributions for Time to LV Ground Impact and Time to FTS Activation by FCO.....	10-6
Figure 10-3. Synopsis of the LARA Approach.....	10-8
Figure 10-4. Dataflow for Blast Impact Model.....	10-9

Figure 10-5. Monte Carlo Simulation for Explosive Yield Probability Computation.....	10-10
Figure 10-6. Titan IV-SRMU Blast Scenarios.	10-11
Figure 10-7. Glass Breakage Risk Analysis Modeling Process.	10-11
Figure 10-8. Models for Overpressure Propagation.....	10-12
Figure 10-9. Blast Risk Analysis Output.	10-12
Figure 10-10. Vacuum IIP Trace for a Titan IV/IUS Mission.	10-13
Figure 10-11. Casualty Expectation Distribution in Re-entry Accidents.....	10-13
Figure 10-12. Conceptual MMOD Event Tree Model.....	10-15
Figure 10-13. Approximate Calculation of Probability of MMOD Impact Affecting a Critical Component.	10-16
Figure 10-14. Facility Power Schematic.	10-18
Figure 10-15. Fault Tree for Loss of the Control Computer.	10-19
Figure 10-16. Facility Fire Event Tree.....	10-20
Figure 11-1. A Schematic Representation of Probabilistic Structural Analysis.	11-2
Figure 11-2. Joint Probability Density Function for Two Random Variables showing the Failure Region.	11-4
Figure 11-3. Probabilistic Structural Analysis using Monte Carlo Simulation.	11-9
Figure 11-4. Joint Probability Density Function (JPDF), Exact and Approximate Limit-State, and Most Probable Point (MPP) for Two Random Variables in Transformed (u) Space.	11-10
Figure 11-5. Concepts of 1st Order Reliability Method (FORM) for Probability Approximations.	11-11
Figure 11-6. Concepts of 2nd Order Reliability Method (SORM) for Probability Approximations.	11-11
Figure 11-7. A Random Dimension h and its Effects on the FE Mesh.....	11-15
Figure 11-8. Cantilever Beam.	11-20
Figure 11-9. Beam Finite Element Example.	11-21
Figure 11-10. CDF of Maximum Stress for the Three-Point Bend Specimen Plot on Normal Probability Scale.....	11-23
Figure 12-1. Propagation of Epistemic Uncertainties.....	12-3
Figure 12-2. Crude Monte Carlo Sampling.	12-4
Figure 12-3. Latin Hypercube Sampling (LHS) Technique.	12-5
Figure 12-4. Fault Trees for Systems A and B.....	12-6
Figure 12-5. Event Tree for Uncertainty Propagation.	12-7
Figure 12-6. The pdf for the Risk Metric R.....	12-11
Figure 12-7. A Context for Epistemic Uncertainty in Risk Assessments.....	12-14
Figure 13-1. Three Displays of an Epistemic Distribution.	13-5
Figure 13-2. Alternative Displays for Conditional Epistemic Distribution.	13-6

Figure 13-3. A Representative Aleatory Exceedance Curve (Without Consideration of Epistemic Uncertainties).	13-7
Figure 13-4. Exceedance Frequency versus Consequences for the Example Problem.....	13-9
Figure 13-5. Aleatory Exceedance Curves with Epistemic Uncertainties for a Typical Space Nuclear Risk Analysis.	13-10
Figure 13-6. Ranking Results for the Basic Events of the Example Problem.	13-17
Figure 13-7. Ranking Results for the Parameters of the Example Problem.	13-19
Figure 14-1. Impact of Abort Effectiveness On Crew Risk for Various Booster Reliabilities....	14-1
Figure 14-2. Schematic of the Abort Assessment Problem.	14-4
Figure 14-3. Failure Propagation Elements for an Early Concept Crew Risk Model.	14-7
Figure 14-4. Failure Propagation Elements for a More Mature Crew Risk Model.....	14-8
Figure 14-5. Notional Diagram of Engine Failure Progression Through Three Stages of Failure.	14-10
Figure 14-6. Expanded Failure Progression Showing Basis of Failure Path Branching Between Failure Stages. (Dashed arrows indicate mappings added after additional analysis.)	14-11
Figure 14-7. Sample Simulation of Blast Overpressure Wave Passing Over Crew Module. .	14-13
Figure 14-8. Schematic of the Components and Inputs in the Blast Overpressure Analysis.	14-15
Figure 14-9. Example of Debris Strike Probability as a Function of Abort Time During Ascent.	14-18
Figure 14-10. Debris Mass and Impact Velocity Required to Penetrate the Crew Module Skin.	14-19
Figure 14-11. Example of Reduction in Debris Strike Probability Due to Imposing Penetration Criteria.	14-19
Figure 14-12. Example of Detailed Structural Response Computed for a Crew Module.....	14-20
Figure 14-13. Schematic of Integrated Risk Modeling Elements.	14-21
Figure 14-14. Simplified Representation of Risk Simulation Model Algorithm.	14-23
Figure 14-15. Influence Diagram of Main Sections of GoldSim Model.	14-24
Figure 14-16. Sample GoldSim Inputs That Link to the Excel Spreadsheets.	14-25
Figure 14-17. Sample GoldSim Inputs That Define the Start and End of Each Phase.	14-25
Figure 14-18. Failure Initiation Logic.....	14-26
Figure 14-19. Ares-Initiated Failure Environments.	14-26
Figure A-1. Definition of an Indicator Variable.	A-1
Figure A-2. A Venn Diagram.....	A-2
Figure A-3. The NOT Operation.....	A-2
Figure A-4. The Union of Events.....	A-3
Figure A-5. The Intersection of Events.	A-3
Figure A-6. A Series System of N Components.....	A-4
Figure A-7. Pictorial Representation of Equation (A-6).	A-4

Figure A-8. A Parallel System of N components.....	A-5
Figure A-9. Pictorial Representation of Equation (A-8).....	A-5
Figure A-10. Block Diagram of the Two-out-of-Three System.....	A-6
Figure A-11. Pictorial Representation of Equation (A-14).....	A-7
Figure A-12. Various Cases for the Inspection Example.....	A-12
Figure A-13. The Random Variable for the Die Experiment.....	A-12
Figure A-14. The Cumulative Distribution Function for the Die Experiment.....	A-13
Figure A-15. CDF and pdf for the Example.....	A-15
Figure B-1. Binary States of an Experiment.....	B-1
Figure B-2. The Bathtub Curve.....	B-3
Figure B-3. Weibull Hazard Functions for Different Values of b.....	B-6
Figure C-1. Representation of a Probability Distribution (epistemic uncertainty), Where the 90% Credible Interval (0.04 to 0.36) is Shown.....	C-3
Figure C-2. Comparison of Prior and Posterior Distributions for Example 1.....	C-4
Figure C-3. DAG representing Script 1.....	C-6
Figure C-4. Comparison of Prior and Posterior Distributions for Example 3.....	C-10
Figure D-1. Conceptual Characteristics of an MLD.....	D-6
Figure D-2. Lunar Base MLD Extract.....	D-7
Figure D-3. Energetic Event ESD.....	D-10
Figure D-4. Electrolyte Leakage ESD.....	D-11
Figure D-5. Smoldering Event ESD.....	D-12
Figure D-6. Atmosphere Leak ESD.....	D-12
Figure D-7. Energetic Hazard Event Tree.....	D-13
Figure D-8. Electrolyte Leakage Event Tree.....	D-14
Figure D-9. Event Tree for the Smoldering IE.....	D-14
Figure D-10. Atmosphere Leakage Event Tree.....	D-15
Figure D-11. Lunar Base Oxygen Supply System.....	D-16
Figure D-12. Fault Tree for Inability To Replenish the Base Atmosphere.....	D-17
Figure D-13. Fault Tree for Failure To Supply Oxygen.....	D-18
Figure D-14. Fault Tree for Loss of the Partial Pressure of Oxygen Sensors.....	D-19
Figure D-15. Final Fault Tree for Failure To Supply Oxygen.....	D-20
Figure D-16. Quantification of Linked ETs/Fault Trees.....	D-21
Figure D-17. Event Sequence Diagram for Launch Phase.....	D-31
Figure D-18. Event Tree for Launch Phase.....	D-32
Figure D-19. Simplified Event Tree for Launch Phase.....	D-32
Figure D-20. Preliminary Event Tree for Cruise Phase.....	D-33
Figure D-21. Simplified Event Tree for Cruise Phase.....	D-34

Figure D-22. Probability of Battery Status (as a Function of λt).....	D-35
Figure D-23. Event Tree Model of System Redundancy.	D-36
Figure D-24. Alternative Event Tree Model of System Redundancy.	D-37
Figure D-25. Event Tree for Lander Science Mission.	D-38
Figure E-1. Atmosphere Leak ESD.....	E-1
Figure E-2. Lunar Base Atmospheric Leak Simulation Objects.	E-3
Figure E-3. Compartment Block Diagram.	E-4
Figure E-4. Leak Detection Block Diagram.	E-5
Figure E-5. Maintenance Block Diagram.	E-6
Figure E-6. Escape Craft Block Diagram.	E-7
Figure E-7. Lunar Base Atmospheric Leak Simulation Results.	E-8
Figure E-8. Lunar Base Atmospheric Leak Objects with External Missions	E-9
Figure E-9. MMD Event Generator	E-10
Figure E-10. Containment Objects Block Diagram with External Work	E-11
Figure E-11. Leak Detector Block Diagram for External Work Model	E-12
Figure E-12. Maintenance Block Diagram used with External Work	E-13
Figure E-13. Mission Block Diagram for External Work.....	E-14
Figure E-14. Escape Craft Block Diagram used with External Work	E-15
Figure E-15. Results of Lunar Base Atmospheric leak with External Missions added.....	E-16

Tables

Table 3-1. Scenarios Leading to "Loss of Vehicle" and Their Associated Frequencies.....	3-7
Table 3-2. Examination of Risk Reduction Strategies for the Example Problem.	3-9
Table 4-1. Sample Dependency Matrix.....	4-3
Table 4-2. Boolean Expressions for Figures 4-4 and 4-7	4-15
Table 4-3. Boolean Expressions for Figure 4-8.	4-16
Table 5-1. Typical Probability Models in PRAs and Their Parameters.	5-2
Table 5-2. Typical Prior and Likelihood Functions Used in PRAs.....	5-13
Table 5-3. Common Conjugate Priors Used in Reliability Data Analysis.....	5-13
Table 6-1. Bayesian Calculations for the Simple Example (No Failures).	6-11
Table 6-2. Bayesian Calculations for the Simple Example with New Evidence (One Failure).	6-12
Table 7-1. Screening Values of Global CCF (g) for Different System Configurations.....	7-9
Table 7-2. Simple Point Estimators for Various CCF Parametric Models.....	7-17
Table 8-1. Initial Screening Model of Estimated Human Error Probabilities and Error Factors for Diagnosis Within Time T by Control Room Personnel of Abnormal Events Annunciated Closely in Time.	8-9
Table 8-2. Initial Screening Model of Estimated Human Error Probabilities and Error Factors for Rule-Based Actions by Control Room Personnel After Diagnosis of an Abnormal Event.	8-9
Table 8-3. The Fifteen Cognitive Activities According to CREAM.....	8-12
Table 8-4. PSFs for Adjusting Basic HEPs.	8-14
Table 8-5. Basic HEPs and Uncertainty Bounds According to CREAM.....	8-15
Table 8-6. NARA EPCs and Their Effects (partial list).....	8-16
Table 8-7. The Generic Tasks of NARA (partial list).....	8-17
Table 8-8. The Generic Tasks of NARA for Checking Correct Plant Status and Availability of Plant Resources.....	8-17
Table 8-9. The Generic Tasks of NARA for Alarm/Indication Response.	8-17
Table 8-10. The Generic Tasks of NARA for Communication.	8-17
Table 8-11. Action Error Type Base Rate Comparison.....	8-19
Table 8-12. Diagnosis Error Type Base Rate Comparison.....	8-20
Table 8-13. Mixed-Task Base Rate Comparison.	8-20
Table 8-14. Generic BHEP and RF Estimates [8-1, 8-4].	8-28
Table 9-1. Causes of Major NASA Mission Failures*, 1998-2007.	9-6
Table 9-2. System Actuations and Maneuvers in PA-1 Mission.	9-18
Table 10-1. Fire Progression.	10-21
Table 10-2. Illustrative Values for λ_j and $\Pr(D_j F_j)$	10-23
Table 11-1. Advantages and Disadvantages of Several Common Probabilistic Methods.	11-13
Table 11-2. Parameters for the Stress Limit State.....	11-20

Table 11-3. Uncertain Inputs for the Simply Supported Beam Example.....	11-22
Table 11-4. Example Finite Element Input.....	11-22
Table 12-1. List of Basic Events and Associated Uncertain Parameters.....	12-8
Table 12-2. Uncertainty Distributions for Uncertain Parameters.....	12-9
Table 12-3. Statistics for Scenario 4 pdf.....	12-12
Table 13-1. An Example of Presenting Dominant Risk Scenarios in a Tabular Form.....	13-3
Table 13-2. List of Scenarios and Exceedance Probabilities.....	13-7
Table 13-3. Construction of Exceedance Frequency for the Example Problem.....	13-8
Table 13-4. Relation among DIM and Traditional Importance Measures.....	13-15
Table 13-5. Calculation of Importance Measures for the Example Problem.....	13-16
Table 13-6. DIM Ranking for the Parameters of the Numerical Example.....	13-18
Table D-1. Lunar Base Dependency Matrix.....	D-4
Table D-2. Perfunctory List of Candidate IEs.....	D-5
Table D-3. Battery FMECA Excerpt.....	D-8
Table D-4. Naming Convention Example for the Lunar Base.....	D-16
Table D-5. Input Data Extract.....	D-23
Table D-6. SAPHIRE Quantification Report for Failure of Partial Pressure of Oxygen Sensors.....	D-23
Table D-7. Cut Set Report for Event Sequence 4.....	D-24
Table D-8. Cut Set Report for Loss of Crew.....	D-24
Table D-9. Uncertainty Results for Loss of Crew.....	D-25
Table D-10. Lunar Base Importance Measures.....	D-26
Table D-10 (cont.). Lunar Base Importance Measures.....	D-27
Table D-11. Launch Phase Timeline.....	D-30
Table D-12. Probability of Battery Status (per Mission Phase).....	D-35
Table E-1. Lunar Surface Micrometeoroid Flux.....	E-2
Table E-13. Lunar Base with External Mission Variates List.....	E-15

Acronyms and Abbreviations

ACS	Attitude Control System
ADS	Automatic Destruct System
ARM	Alarm Response Model
ASME	American Society of Mechanical Engineers
BHEP	Basic Human Error Probability
BM	Birnbaum Measure
BP	Basic Parameter
CC	Command and Control
CCBE	Common Cause Basic Event
CCCG	Common Cause Component Group
CCE	Common Cause Event
CCF	Common Cause Failure
CCU	Control Computer Unit
CD	Complete Dependence
CDF	Cumulative Distribution Function
CDS	Command Destruct System
CM	Communication
CR	Collision Rate
CRM	Continuous Risk Management
CRV	Continuous Random Variable
CSRM	Context-Based Software Risk Model
DF	Dependent Failure
DFM	Dynamic Flowgraph Methodology
DIM	Differential Importance Measure
DRM	Design Reference Mission
DRV	Discrete Random Variable
DSMCS	Dependence-Suspect Minimal Cut Sets
ECLS	Environmental Control and Life Support
ECOM	Error of Commission
EE	Emergency Escape
EF	Error Factor
EOM	Error of Omission
EPIX/RADS	Equipment Performance Information Exchange/Reliability and Availability Database System
EPRD	Electronic Parts Reliability Data
ESA	European Space Agency
ESD	Event Sequence Diagram
ET	Event Tree
ETA	Event Tree Analysis
FCO	Flight Control Officer
FMD	Failure Mode Distribution
FMEA	Failure Modes and Effects Analysis
FMECA	Failure Modes and Effects Criticality Analysis
FOIA	Freedom of Information Act

FS	Fire Suppression
FT	Fault Tree
FTA	Fault Tree Analysis
FTLCS	Fluid Tank Level Control System
FTS	Flight Termination System
F-V	Fussell-Vesely
GIDEP	Government-Industry Data Exchange Program
GSFC	Goddard Space Flight Center
HAZOP	Hazard and Operability
HCR	Human Cognitive Reliability
HD	High Dependence
HEP	Human Error Probability
HI	Human Interaction
HMI	Human Machine Interface
HRA	Human Reliability Analysis
HSI	Human-System Integration
IE	Initiating Event
IEEE	Institute of Electrical and Electronics Engineers
IIP	Instantaneous Impact Point
INL	Idaho National Laboratory
ISS	International Space Station
ITAR	International Traffic in Arms Regulations
IUS	Inertial Upper Stage
LARA	Launch Risk Analysis
LD	Low Dependence
LHS	Latin Hypercube sampling
LOC	Loss of Crew
LOM	Loss of Mission
LV	Launch Vehicle
MADS	Modeling Analysis Data Sets
MCS	Minimal Cut Set
MD	Moderate Dependence
MET	Mission Elapsed Time
MIT	Massachusetts Institute of Technology
MLD	Master Logic Diagram
MLE	Maximum Likelihood Estimation
MMI	Man-Machine Interface
MMOD	Micrometeoroid and Orbital Debris
MTBF	Mean Time Between Failure
MTTF	Mean Time to Failure
MTTR	Mean Time to Repair
NASA	National Aeronautics and Space Administration
NASDA	National Space Development Agency of Japan
NPP	Nuclear Power Plant
NPR	NASA Procedural Requirements

NPRD	Non-Electronic Parts Reliability Data
NRC	United States Nuclear Regulatory Commission
NUCLARR	Nuclear Computerized Library for Assessing Reactor Reliability
OK	Mission Success (as used in a PRA model)
OREDA	Offshore Reliability Data
OSMA	Office of Safety and Mission Assurance
pdf	Probability Density Function
PLC	Programmable Logic Computer
PM	Performance Measures
pmf	Probability Mass Function
POF	Probability of Failure
POS	Probability of Success
PRA	Probabilistic Risk Assessment
PRACA	Problem Reporting and Corrective Action
PSA	Probabilistic Safety Assessment
PSF	Performance Shaping Factor
PVC	Population Variability Curve
PW	Power Generation, Storage, and Distribution
QA	Quality Assurance
RAC	Reliability Analysis Center
RAW	Risk Achievement Worth
RF	Recovery Factor
RIAC	Reliability Information Analysis Center
ROCOF	Rate of Occurrence of Failures
RRW	Risk Reduction Worth
RTG	Radioisotope Thermoelectric Generator
RV	Random Variable
S&C	Sensing and Command
SC	Science
STRATCOM	Strategic Command
T&M	Test and Maintenance
THERP	Technique for Human Error Rate Prediction
TRC	Time Reliability Curve
V&V	Verification and Validation
ZD	Zero Dependence

1. Introduction

Probabilistic Risk Assessment (PRA) is a comprehensive, structured, and logical analysis method aimed at identifying and assessing risks in complex technological systems for the purpose of cost-effectively improving their safety and performance. NASA's objective is to better understand and effectively manage risk, and thus more effectively ensure mission and programmatic success, and to achieve and maintain high safety standards at NASA. NASA intends to use risk assessment in its programs and projects to support optimal management decision making for the improvement of safety and program performance.

In addition to using quantitative/probabilistic risk assessment to improve safety and enhance the safety decision process, NASA has incorporated quantitative risk assessment into its system safety assessment process, which until now has relied primarily on a qualitative representation of risk. Also, NASA has recently adopted the Risk-Informed Decision Making (RIDM) process [1-1] as a valuable addition to supplement existing deterministic and experience-based engineering methods and tools.

Over the years, NASA has been a leader in most of the technologies it has employed in its programs. One would think that PRA should be no exception. In fact, it would be natural for NASA to be a leader in PRA because, as a technology pioneer, NASA uses risk assessment and management implicitly or explicitly on a daily basis. NASA has probabilistic safety requirements (thresholds and goals) for crew transportation system missions to the International Space Station (ISS) [1-2]. NASA intends to have probabilistic requirements for any new human spaceflight transportation system acquisition.

Methods to perform risk and reliability assessment in the early 1960s originated in U.S. aerospace and missile programs. Fault tree analysis (FTA) is an example. It would have been a reasonable extrapolation to expect that NASA would also become the world leader in the application of PRA. That was, however, not to happen.

Early in the Apollo program, estimates of the probability for a successful roundtrip human mission to the moon yielded disappointingly low (and suspect) values and NASA became discouraged from further performing quantitative risk analyses until some two decades later when the methods were more refined, rigorous, and repeatable. Instead, NASA decided to rely primarily on the Hazard Analysis (HA) and Failure Modes and Effects Analysis (FMEA) methods for system safety assessment.

In the meantime, the nuclear industry adopted PRA to assess safety. This analytical method was gradually improved and expanded by experts in the field and has gained momentum and credibility over the following decades, not only in the nuclear industry, but also in other industries like petrochemical, offshore platforms, and defense. By the time the Challenger accident occurred in 1986, PRA had become a useful and respected tool for safety assessment. Because of its logical, systematic, and comprehensive approach, PRA has repeatedly proven capable of uncovering design and operational weaknesses that had escaped even some of the best deterministic safety and engineering experts. This methodology showed that it was very important to examine not only single low-probability and high-consequence mishap events, but also high-consequence scenarios that can emerge as a result of the occurrence of multiple high-probability and low consequence or nearly benign events. Contrary to common perception, the latter in its aggregate is oftentimes more detrimental to safety than the former.

Then, the October 29, 1986, “Investigation of the Challenger Accident” [1-3], by the Committee on Science and Technology, House of Representatives, stated that, without some credible means of estimating the probability of failure (POF) of the Shuttle elements, it was not clear how NASA could focus its attention and resources as effectively as possible on the most critical Shuttle systems.

In January 1988, the Slay Committee recommended, in its report called the “Post-Challenger Evaluation of Space Shuttle Risk Assessment and Management” [1-4], that PRA approaches be applied to the Shuttle risk management program at the earliest possible date. It also stated that databases derived from Space Transportation System failures, anomalies, flight and test results, and the associated analysis techniques should be systematically expanded to support PRA, trend analysis, and other quantitative analyses relating to reliability and safety.

As a result of the Slay Committee criticism, NASA began to use PRA, at least in a “proof-of-concept” mode, with the help of contractors. A number of NASA PRA studies were conducted in this fashion over the next 10 years.

During the first decade of this century, PRA has gained significant momentum at NASA. It was applied to assess the safety of major human flight systems, including the Space Shuttle, the International Space Station and the Constellation Program. It was also applied for flight approval of all nuclear missions, i.e., missions carrying radioactive material. PRA, as a safety assessment method, was incorporated into System Safety to use when quantitative risk assessment is deemed necessary. Moreover, a RIDM approach was developed to help bring risk assessment to the engineering and management decision table. Meanwhile top level NASA policy documents (e.g., NPD 1000.5A [1-5]) have begun to call for increasingly quantitative approaches to managing risk at the Agency level.

1.1 Purpose and Scope of This Procedures Guide

During the past several decades, much has been written on PRA methods and applications. Several university and practitioner textbooks and sourcebooks currently exist, but they focus on applications of PRA to industries other than aerospace. Although some of the techniques used in PRA originated in work for aerospace and military applications, no comprehensive reference currently exists for PRA applications to aerospace systems.

This PRA Procedures Guide, in the present second edition, is neither a textbook nor an exhaustive sourcebook of PRA methods and techniques. It provides a set of recommended procedures, based on the experience of the authors, that are applicable to different levels and types of PRA that are performed for aerospace applications. It therefore serves two purposes, to:

1. Complement the training material taught in the NASA PRA course for practitioners, and together with the Fault Tree Handbook [1-6], the Risk-Informed Decision Making Handbook [1-1], the Bayesian Inference handbook [1-7], the Risk Management Handbook [1-8], and the System Safety Handbook [1-9] to provide quantitative risk methodology documentation, and to
2. Provide aerospace PRA practitioners in selecting an analysis approach that is best suited for their applications.

The material in this Procedures Guide is organized into five parts:

1. A management introduction to PRA and the Risk Management framework in which it is used is presented in Chapters 1-3.

2. Chapters 4-12 cover the details of PRA: methods for scenario development, data collection and parameter estimation, uncertainty analysis, dependent failure analysis, human reliability analysis, software reliability analysis, modeling of physical processes for PRA, probabilistic structural analysis, and uncertainty propagation. The Human Reliability Analysis (Chapter 8) was updated in the present edition. The Software Risk Assessment (Chapter 9) was also re-written but this area is still not mature enough to include several recommended methodology approaches.
3. Chapter 13 discusses the presentation of results. The discussion addresses what results should be presented and in what format. Presentation and communication of PRA results is extremely important for use in risk-informed decision making.
4. Given the importance of crew safety, Chapter 14 presents details on launch abort modeling including the factors that must be considered, the analysis methodologies that should be employed, and how the assessment should be included in the vehicle development process.
5. Finally, Appendix A through C contain basic information to supplement one's existing knowledge or self-study of probability, statistics, and Bayesian inference. Then two PRA examples are provided in Appendix D and, finally, the use of simulation in the probabilistic assessment of risk is covered in Appendix E.

1.2 Knowledge Background

Users of this Guide should be well grounded in the basic concepts and application of probability and statistics. For those lacking such a background, some tutorial material has been provided in the Appendices, which should be supplemented by formal and/or self-study. However, this prior knowledge is not essential to an understanding of the main concepts presented here.

1.3 Application Recommendation

The authors recommend that the users of this guide adhere to the philosophy of a “graded approach” to PRA application. That is, the resources and depth of assessment should be commensurate with the stakes and the complexity of the decision situations being addressed. Depending on project scale, life cycle phase, etc., different modeling detail and complexity are appropriate in PRA. As a general rule of thumb, the detail and complexity of modeling should increase with successive program/project life cycle phases. For a given phase, parametric, engineering, and logic modeling can be initiated at a low level of detail and complexity; the level of detail and complexity can then be increased in an iterative fashion as the project progresses. Further discussion of the graded approach philosophy is provided in NASA System Safety Handbook [1-9].

1.4 References

- 1-1 *NASA Risk-Informed Decision Making Handbook*, NASA/SP-2010-576, April 2010.
- 1-2 Decision Memorandum for Administrator, “Agency’s Safety Goals and Thresholds for Crew Transportation Missions to the International Space Station (ISS),” Washington, DC, 2011.

- 1-3 *Investigation of the Challenger Accident: Report of the Committee on Science and Technology, House Report 99-1016*, Washington, DC: U.S. House of Representatives Committee on Science and Technology, 1986.
- 1-4 *Post-Challenger Evaluation of Space Shuttle Risk Assessment and Management*, Committee on Shuttle Criticality Review and Hazard Analysis Audit of the Aeronautic and Space Engineering Board, National Research Council; National Academy Press, January 1988.
- 1-5 NPD 1000.5A, Policy for NASA Acquisition, January 15, 2009.
- 1-6 *Fault Tree Handbook with Aerospace Applications*, Version 1.1, NASA, August 2002.
- 1-7 *Bayesian Inference for NASA Probabilistic Risk and Reliability Analysis*, NASA-SP-2009-569, <http://www.hq.nasa.gov/office/codeq/doctree/SP2009569.htm>, 2009.
- 1-8 *NASA Risk Management Handbook*, NASA/SP-2011-3422, November 2011.
- 1-9 *NASA System Safety Handbook*, Volume 1, NASA/SP-2010-580, December 2011.

2. Risk Management

This chapter addresses the subject of risk management in a broad sense. Section 2.1 defines the concept of risk. There are several definitions, but all have as a common theme the fact that risk is a combination of the undesirable consequences of accident scenarios and the probability of these scenarios.

In Section 2.2 we will discuss the concepts of Risk-Informed Decision Making (RIDM) and Continuous Risk Management (CRM), which together provide a disciplined environment for proactive decision making with regard to risk.

2.1 Definition of Risk

The concept of risk includes both undesirable consequences and likelihoods, e.g., the number of people harmed, and the probability of occurrence of this harm. Sometimes, risk is defined as a set of single values, e.g., the expected values of these consequences. This is a summary measure and not a general definition. Producing probability distributions for the consequences affords a much more detailed description of risk.

A very common definition of risk represents it as a set of triplets [2-1]: scenarios, likelihoods, and consequences. Determining risk generally amounts to answering the following questions:

1. What can go wrong?
2. How likely is it?
3. What are the associated consequences?

The answer to the first question is a set of accident scenarios. The second question requires the evaluation of the probabilities of these scenarios, while the third estimates their consequences. Implicit within each question is that there are uncertainties. The uncertainties pertain to whether all the significant accident scenarios have been identified, and whether the probabilities of the scenarios and associated consequence estimates have properly taken into account the sources of variability and the limitations of the available information.

Scenarios and uncertainties are among the most important components of a risk assessment. Figure 2-1 shows the implementation of these concepts in PRA. In this Figure, uncertainty analysis is shown to be an integral part of each step of the process rather than just a calculation that is performed at the end of the risk quantification.

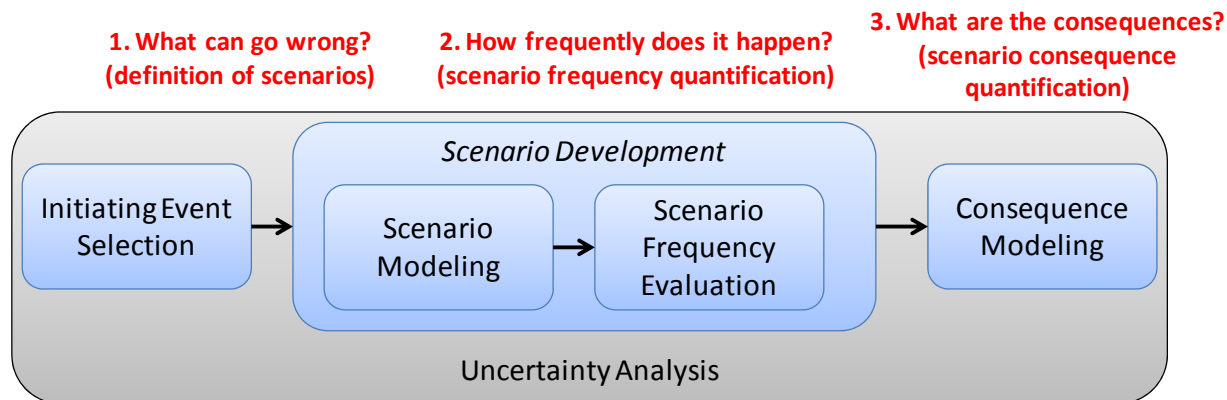


Figure 2-1. Implementation of the Triplet Definition of Risk in PRA.

The accident scenarios begin with a set of “initiating events” (IEs) that perturb the system (i.e., cause it to change its operating state or configuration), representing a deviation in the desired system operation. For each IE, the analysis proceeds by determining the pivotal events that are relevant to the evolution of the scenario which may (or may not) occur and may have either a mitigating or exacerbating effect on the accident progression. The frequencies of scenarios with undesired consequences are determined. Finally, the multitude of such scenarios is put together, with an understanding of the uncertainties, to create the risk profile of the system. This risk profile then supports risk management.

2.2 Risk Management at NASA

Risk management (RM) is an integral aspect of virtually every challenging human endeavor. Although the complex concepts that RM encapsulates and the many forms it can take make it difficult to effectively implement, effective risk management is critical to program and project success.

In the context of risk management, *performance risk* refers to shortfalls with respect to performance requirements in any of the mission execution domains of safety, technical, cost, and schedule. The term *performance risk* is also referred to simply as *risk*. This generalization makes the concept of risk broader than in typical PRA contexts where the term *risk* is used to characterize only safety performance, and not necessarily with respect to defined requirements. *Individual risk* is different from performance risk, in that it refers to a particular issue that is expressed in terms of a departure from the program/project plan assumptions. Individual risks affect performance risks but are not synonymous with them. For example, an unusually high attrition of design engineers could affect the date within which the design is completed and thereby affect the ability to launch within a required time window. The unexpectedly high attrition would be classified as an individual risk that affects the the ability to meet the required schedule for launch, a performance risk. The role of PRA in the context of risk management is to quantify each performance risk, taking into account the individual risks that surface during the program/project.

Until recently, NASA’s RM approach had been based almost exclusively on Continuous Risk Management (CRM), which stresses the management of individual risk issues during implementation. In December of 2008, NASA revised its RM approach, in order to more effectively foster proactive risk management. This approach, which is outlined in NPR 8000.4A, *Agency Risk Management Procedural Requirements* [2-2], and further developed in NASA/SP-

2011-3422, *NASA Risk Management Handbook* [2-3], evolves NASA's risk management to entail two complementary processes: Risk-Informed Decision Making (RIDM) and CRM. RIDM is intended to inform systems engineering (SE) decisions (e.g., design decisions) through better use of risk and uncertainty information, such as that resulting from PRA, in selecting alternatives and establishing baseline performance requirements

CRM is then used to manage risks over the course of the development and implementation phases of the life cycle to assure that requirements related to safety, technical, cost, and schedule are met. In the past, RM was considered equivalent to the CRM process; now, RM is defined as comprising both the RIDM and CRM processes, which work together to assure proactive risk management as NASA programs and projects are conceived, developed, and executed. Figure 2-2 illustrates the concept.

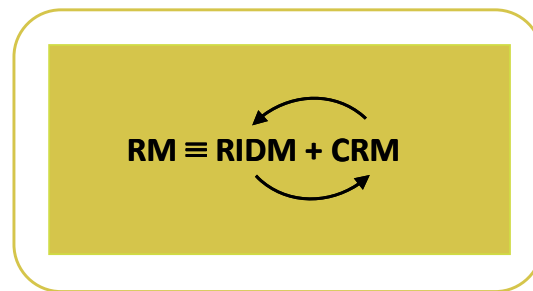


Figure 2-2. Risk Management as the Interaction of Risk-Informed Decision Making and Continuous Risk Management. [NASA/SP-2010-576].

Within the NASA organizational hierarchy (see Figure 2-3), high-level objectives, in the form of NASA Strategic Goals, flow down in the form of progressively more detailed performance requirements (PR), whose satisfaction assures that the objectives are met. Each organizational unit within NASA negotiates with the unit(s) at the next lower level in the organizational hierarchy a set of objectives, deliverables, performance measures (PM), baseline performance requirements, resources, and schedules that defines the tasks to be performed by the unit(s). Once established, the lower level organizational unit manages its own risks against these specifications, and, as appropriate, reports risks and elevates decisions for managing risks to the next higher level based on predetermined risk thresholds that have been negotiated between the two units. Invoking the RIDM process in support of key decisions as requirements flow down through the organizational hierarchy assures that objectives remain tied to NASA Strategic Goals while also capturing why a particular path for satisfying those requirements was chosen.

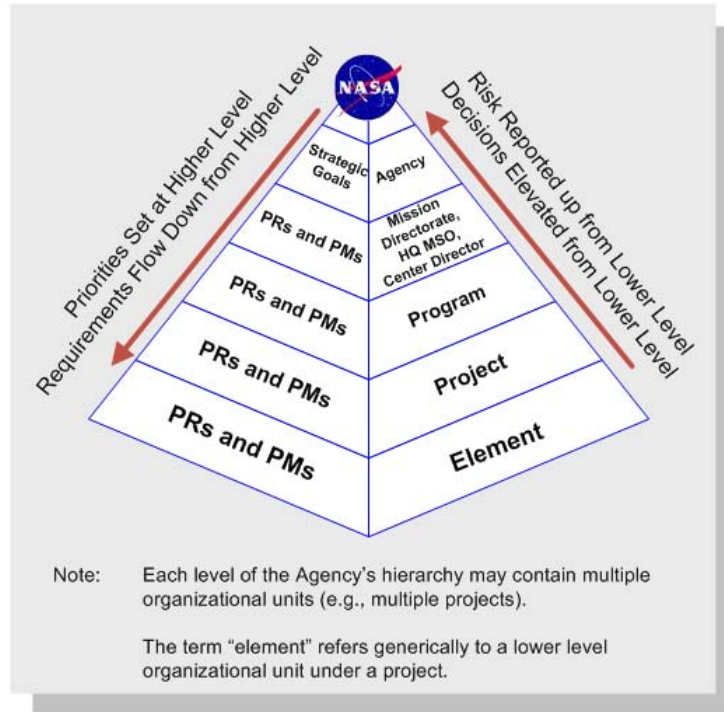


Figure 2-3. Flowdown of Performance Requirements (Illustrative).

2.2.1 Risk-Informed Decision Making Process (RIDM)

As specified in NPR 8000.4A, the RIDM process itself consists of the three parts shown in Figure 2-4.

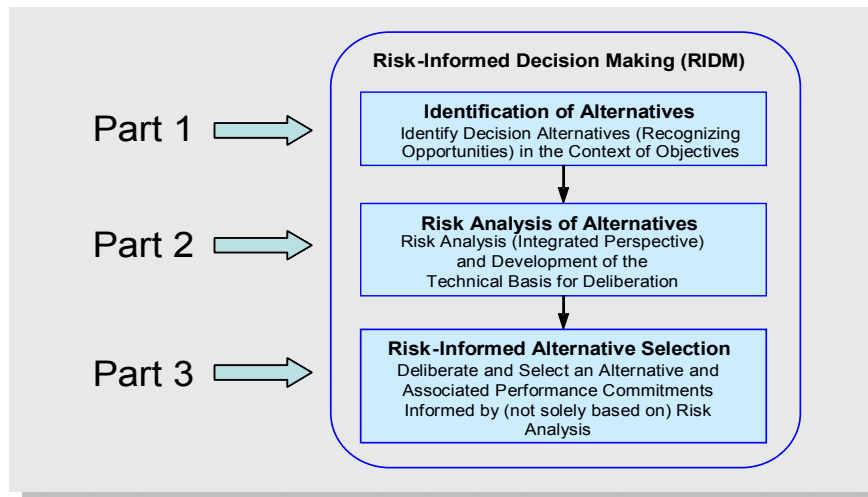


Figure 2-4. The RIDM Process.

2.2.1.1 Part 1, Identification of Alternatives in the Context of Objectives

Decision alternatives are identifiable only in the context of the objectives that they are meant to satisfy. Objectives, which in general may be multifaceted and qualitative, are captured through interactions with the relevant stakeholders. They are then decomposed into their constituent derived objectives, using an objectives hierarchy. Each derived objective reflects an individual issue that is significant to some or all of the stakeholders. At the lowest level of decomposition are quantifiable *performance objectives*, each of which is associated with a *performance measure* that quantifies the degree to which the performance objective is met. Typically, each performance measure has a “direction of goodness” that indicates the direction of increasing benefit.

A comprehensive set of performance measures is considered in decision making, reflecting stakeholder interests and spanning the mission execution domains of interest. Safety-related performance measures are typically probabilistic, expressing the likelihood, per mission or per unit time, that the undesired safety consequences will be experienced. Examples include:

- Probability of Loss of Crew (P(LOC)): The probability (typically per a defined reference mission) of death or permanently debilitating injury to one or more crewmembers. This performance measure is commonly used to assess crew safety. It is a sufficient measure for overall crew safety (i.e., freedom from LOC, injury, and illness) for short-duration missions where LOC is the dominant concern. For longer duration missions it may be more useful to explicitly address injury and illness using separate performance measures for each.
- Probability of Loss of Vehicle (P(LOV)): The probability that the vehicle will be lost during a mission. In the context of expendable vehicles, P(LOV) has typically been used to quantify the probability that a vehicle will be lost or damaged prior to meeting its mission objectives. In the context of reusable vehicles, P(LOV) has typically been used to quantify the probability that, during a mission, a vehicle will be rendered unusable for future missions.
- Probability of Loss of Mission (P(LOM)): The probability that mission objectives will not be met. For expendable vehicles such as during deep-space robotic missions, P(LOM) is closely related to P(LOV) since, in that context, loss of vehicle is only relevant inasmuch as it affects the achievement of mission objectives.

Objectives whose performance measure values must remain within defined limits give rise to *imposed constraints* that reflect those limits. A threshold for P(LOC), P(LOV), or P(LOM) is an example of an imposed constraint.

Following identification of objectives and associated performance measures, techniques such as trade trees [2-4] are used to generate decision alternatives for consideration. Initially, the trade tree contains high-level decision alternatives representing high-level differences in the strategies used to address objectives. The tree is then developed in greater detail by determining general categories of options that are applicable to each strategy.

2.2.1.2 Part 2, Risk Analysis of Alternatives

For each feasible alternative, uncertainty distributions for the performance measures are quantified, taking into account whatever significant uncertainties stand between the decision to implement the alternative and the accomplishment of the objectives that drive the decision

making process to begin with. Given the presence of uncertainty, the actual outcome of a particular decision alternative will be only one of a spectrum of outcomes that could result from its selection, depending on the occurrence, nonoccurrence, or quality of occurrence of intervening events. Therefore, it is incumbent on risk analysts to model each significant possible outcome, accounting for its probability of occurrence, to produce a distribution of forecasted outcomes for each alternative, as characterized by probability density functions (pdf) over the performance measures (see Figure 2-5). PRA provides a means to generate pdfs for safety-related performance measures.

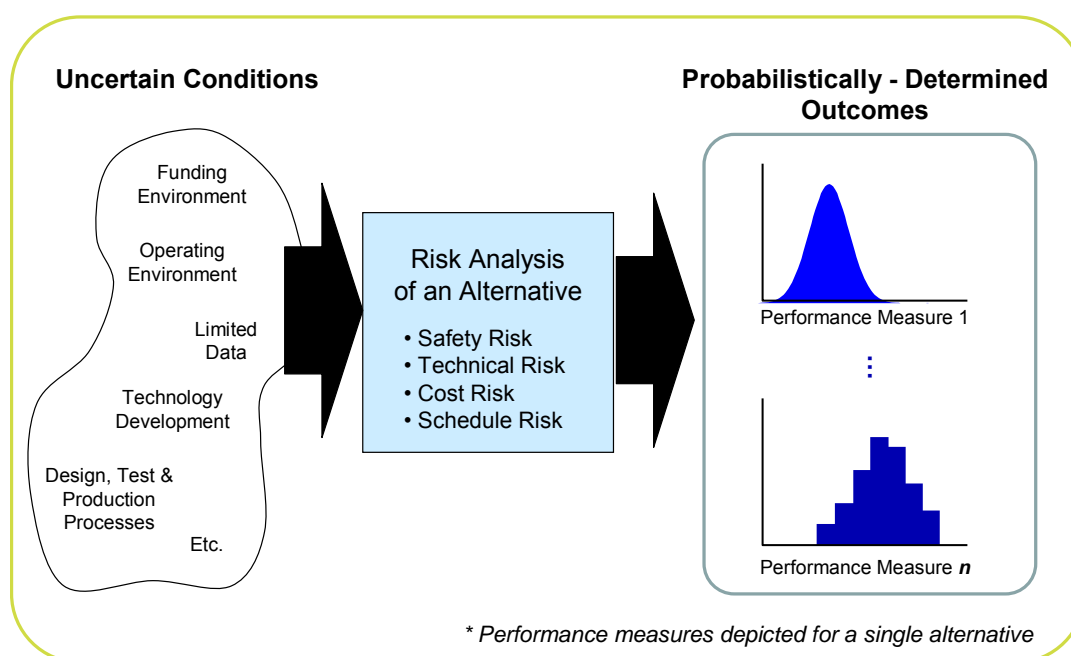


Figure 2-5. Uncertainty of Forecasted Outcomes for a Given Alternative Due to Uncertainty of Analyzed Conditions.

2.2.1.3. Part 3, Risk Informed Alternative Selection

In Part 3, Risk Informed Alternative Selection, deliberation takes place among the stakeholders and the decision maker, and the decision maker either culls the set of alternatives and asks for further scrutiny of the remaining alternatives OR selects an alternative for implementation OR asks for new alternatives.

To facilitate deliberation, the RM handbook introduces the concept of *performance commitments*. A performance commitment is a performance measure value set at a particular percentile of the performance measure's pdf, so as to anchor the decision maker's perspective to that value as if it would be his/her commitment, were he/she to select that alternative. For a given performance measure, the performance commitment is set at the same percentile for every decision alternative, so that the probability of failing to meet it is the same across alternatives, even though the performance commitments themselves differ from one alternative to the next. Performance commitments are not themselves performance requirements. Rather, performance commitments are used to risk-inform the development of credible performance requirements as part of the overall SE process.

The use of performance commitments in RIDM supports a *risk-normalized* comparison of decision alternatives, in that a uniform level of risk tolerance is established prior to deliberating the merits and drawbacks of the various alternatives. Put another way, risk normalized performance commitments show what each alternative is capable of, at an equal likelihood of achieving that capability, given the state of knowledge at the time.

Figure 2-6 presents notional performance commitments for three alternatives (A, B, and C) and three performance measures (cost and schedule have been combined into one performance measure for illustration purposes).

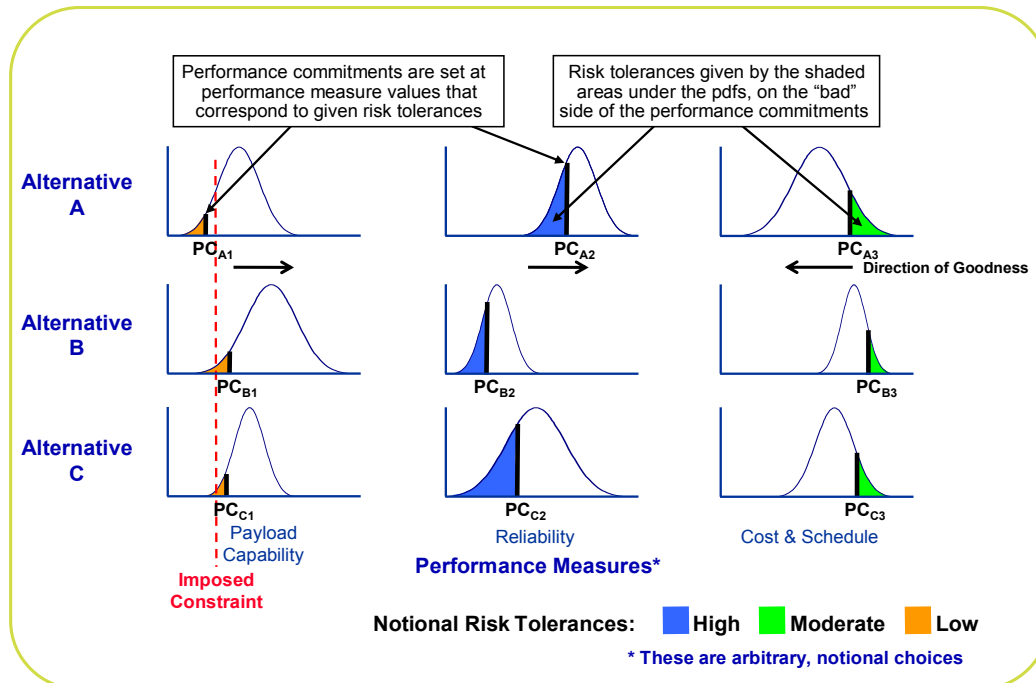


Figure 2-6. Performance Commitments and Risk Tolerances for Three Alternatives.

2.2.2 Continuous Risk Management (CRM)

Once an alternative has been selected using RIDM, performance objectives, imposed constraints, and performance commitments are used as an aid for determining performance requirements through the Systems Engineering process. The term *performance risk* will be used henceforth to denote the probability of not meeting the performance requirements, and PRA will be the principal tool for determining that risk.

After performance requirements have been developed, the risk associated with implementation of the design decision is managed using the CRM process. Because CRM takes place in the context of explicitly-stated performance requirements, the risk that the CRM process manages is the potential for performance shortfalls that may be realized in the future, with respect to these requirements.

The risk tolerance levels for each performance measure obtained from RIDM establish initial levels of risk considered to be tolerable by a decision maker for the achievement of performance

requirements. During the initialization of CRM, the decision maker may choose to levy improvements on the risk tolerance levels in the form of a tightening of these levels according to a risk burn-down schedule at key program/project milestones. In other words, as the program/project evolves over time, design and procedural changes are implemented in an attempt to mitigate risk. In turn, as risk concerns are lowered or retired and the state of knowledge about the performance measures improves, uncertainty should decrease, with an attendant lowering of residual risk (see Figure 2-7).

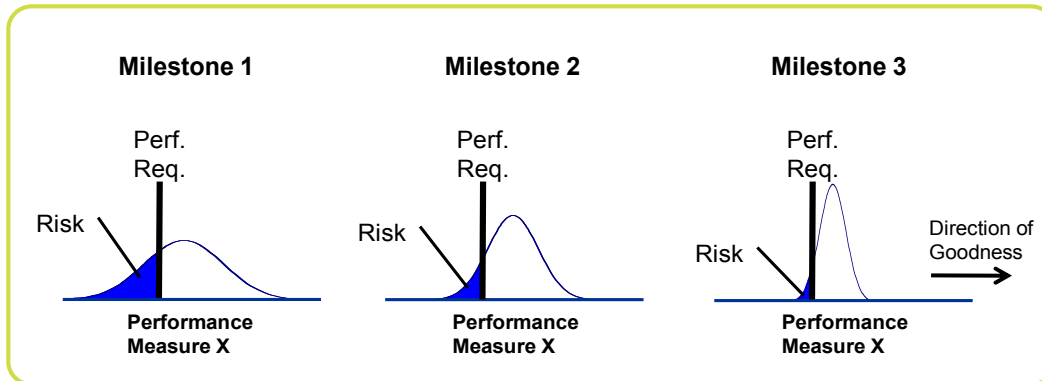


Figure 2-7. Decreasing Uncertainty and Risk over Time

The CRM process starts from the five cyclical functions of *Identify*, *Analyze*, *Plan*, *Track*, and *Control*, supported by the comprehensive *Communicate* and *Document* function [2-5], as shown in Figure 2-8.



Figure 2-8. The CRM Process.

Step 1, Identify

The purpose of the *Identify* step is to capture stakeholders' concerns regarding the achievement of performance requirements. These concerns are referred to as individual risks, and collectively represent the set of undesirable scenarios that put the achievement of the activity's performance requirements at risk. The RM handbook defines "performance risk" as the probability of not meeting a performance requirement. Each performance requirement has an associated performance risk that is produced by those individual risks which, in the aggregate, threaten the achievement of the requirement. Quantification of a performance

requirement's performance risk is accomplished by means of a scenario-based risk model that incorporates the individual risks so that their aggregate effect on the forecasted probabilities of achieving or not achieving the performance requirements can be analyzed.

Step 2, Analyze

The objectives of the *analyze* step are:

- To estimate the likelihoods of the departures and the magnitudes of the consequence components of individual risks, including timeframe, uncertainty characterization, and quantification;
- To assign, in a timely fashion, a criticality rank to each individual risk based on:
 - The probability that the departure will occur;
 - The magnitude of the consequence given occurrence of the departure;
 - The point in the activity's timeline when the individual risk first surfaced (e.g., PDR, CDR);
 - The magnitude of the uncertainties; and
 - The amount of time available after the condition is identified before a departure can possibly occur.
- To update the performance risk to incorporate new individual risks or changes in existing individual risks;
- To determine which departure events and parameters within the models are the most important contributors to each performance risk, i.e., the *risk drivers*.

Step 3, Plan

The objective of the *Plan* step is to decide what action, if any, should be taken to reduce the performance risks that are caused by the aggregation of identified individual risks. The possible actions are:

- *Accept* – A certain level of performance risk can be accepted if it is within the risk tolerance of the program/project manager;
- *Mitigate* – Mitigation actions can be developed which address the drivers of the performance risk;
- *Watch* – Risk drivers can be selected for detailed observation, and contingency plans developed;
- *Research* – Research can be conducted to better understand risk drivers and reduce their uncertainties;

- *Elevate* – Risk management decisions should be elevated to the sponsoring organization at the next higher level of the NASA hierarchy when performance risk can no longer be effectively managed within the present organizational unit;
- *Close* – An individual risk can be closed when all associated risk drivers are no longer considered potentially significant.

Selection of an appropriate risk management action is supported by risk analysis of alternatives and subsequent deliberation, using the same general principles of risk-informed decision making that form the basis for the RIDM process.

Step 4, Track

The objective of the *Track* step is to acquire, compile, and report observable data to track the progress of the implementation of risk management decisions, and their effectiveness once implemented. The tracking task of CRM serves as a clearing house for new information that could lead to any of the following:

- A new risk item;
- A change in risk analysis;
- A change in a previously agreed-to plan;
- The need to implement a previously agreed-to contingency.

Step 5, Control

When tracking data indicate that a risk management decision is not impacting risk as expected, it may be necessary to implement a *control* action. Control actions are intended to assure that the planned action is effective. If the planned action becomes unviable, due either to an inability to implement it or a lack of effectiveness, then the Plan step is revisited and a different action is chosen.

Communicate and Document

Communication and documentation are key elements of a sound and effective CRM process. Well-defined, documented communication tools, formats, and protocols assure that:

- Individual risks are identified in a manner that supports the evaluation of their impacts on performance risk;
- Individual risks that impact multiple organizational units (i.e., cross-cutting risks) are identified, enabling the coordination of risk management efforts;
- Performance risks, and associated risk drivers, are reported by each organizational unit to the sponsoring organization at the next higher level of the NASA hierarchy in a manner that allows the higher level organization to integrate that information into its own assessment of performance risk relative to its own performance requirements;

- Risk management decisions and their rationales are captured as part of the institutional knowledge of the organization.

2.3 References

- 2-1 S. Kaplan and B.J. Garrick, "On the Quantitative Definition of Risk," *Risk Analysis*, 1, 11-37, 1981.
- 2-2 NASA NPR 8000.4A: Agency Risk management Procedural Requirements.
- 2-3 NASA/SP-2011-3422, "NASA Risk Management Handbook," November 2011.
- 2-4 NASA-SP-2007 6105 Rev 1: NASA Systems Engineering Handbook.
- 2-5 Carnegie Mellon University Software Engineering Institute. *Continuous Risk Management Guidebook*, 1996.

3. Probabilistic Risk Assessment Overview

3.1 Historical Background

To motivate the technical approaches discussed in the following sections—that is, to understand the “what” and the “why” of the PRA methods discussed in this Guide—it is appropriate to begin with a brief history of PRA, to show how it differs from classical reliability analysis, and to show how decision-making is informed by PRA.

In many respects, techniques for classical reliability analysis had already been highly developed for decades before PRA was seriously undertaken. Reliability texts from the 1970s emphasized highly quantitative modeling of component-level and system-level reliability—the probability that an item (component or system) would not fail during a specified time (or mission). This kind of modeling was at least theoretically useful in design evaluation. Design alternatives could be compared with respect to their reliability performance. Some sources discussed “probabilistic” reliability modeling, by which they meant propagation of parameter uncertainty through their models to obtain estimates of uncertainty in model output.

The changes in PRA that have taken place since those days represent not only technical advances in the tools available, but also changes in the way we think about safety. In order to understand the “why” of many PRA tools, it is useful to understand this evolution from a historical point of view. Much of this evolution took place in the context of the nuclear power industry. This is not meant to imply that NASA tools are, or should be, completely derived from standard commercial nuclear PRA tools. Some remarks about what is needed specifically in NASA PRA tools are provided in the summary to this chapter (Section 3.4). However, the broader conclusions regarding how PRA can be applied properly in decision-making have evolved largely in the context of commercial nuclear power, and key historical points will be summarized in that context.

3.1.1 Design Basis Evaluation vs. Risk Evaluation

Traditionally, many system designs were evaluated with respect to a design basis, or a design reference mission. In this kind of approach, a particular functional challenge is postulated, and the design evaluation is based on the likelihood that the system will do its job, given that challenge. If a system is simple enough, quantitative reliability calculations can be performed. Alternatively, FMEA can be used essentially to test for redundancy within a system or function, and in some contexts, functional redundancy is presumed to achieve adequate reliability.

Because this approach is not based on a quantitative risk perspective, it does not typically lead to an allocation of resources that is optimal from a risk point of view, even in cases where the designs can be considered “adequate” from a traditional system safety point of view. Moreover, the adequacy of the selection of IEs against which to evaluate the system is extremely difficult to ensure, without the equivalent of a systematic, PRA-style assessment of some kind. Unless highly off-normal events are postulated, systems will not be evaluated for their ability to cope with such events; but appropriately selecting extremely severe events against which to evaluate mitigating capability is nearly impossible without risk perspective.

Moreover, it is found that certain thought processes need to be carried out in failure space to ensure that risk-significant failure modes are identified. Completeness is clearly necessary if prevention resources are to be allocated appropriately.

In general, optimal resource allocation demands some kind of integrated risk evaluation: not just a finding regarding system adequacy, and not a series of unrelated system-level assessments.

3.1.2 From Regulation Based on Design Basis Review to Risk-Informed Regulation

The first comprehensive PRA, the Reactor Safety Study (WASH-1400), was completed in the mid-1970s [1]. Its stated purpose was to quantify the risks to the general public from commercial nuclear power plant (NPP) operation. This logically required identification, quantification, and phenomenological analysis of a very considerable range of low-frequency, relatively high-consequence scenarios that had not previously been considered in much detail. The introduction here of the notion of “scenario” is significant; as noted above, many design assessments simply look at system reliability (success probability), given a design basis challenge. The review of nuclear plant license applications did essentially this, culminating in findings that specific complements of safety systems were single-failure-proof for selected design basis events. Going well beyond this, WASH-1400 modeled scenarios leading to large radiological releases from each of two types of commercial NPPs. It considered highly complex scenarios involving success and failure of many and diverse systems within a given scenario, as well as operator actions and phenomenological events. These kinds of considerations were not typical of classical reliability evaluations. In fact, in order to address public risk, WASH-1400 needed to evaluate and classify many scenarios whose phenomenology placed them well outside the envelope of scenarios normally analyzed in any detail.

WASH-1400 was arguably the first large-scale analysis of a large, complex facility to claim to have comprehensively identified the risk-significant scenarios at the plants analyzed. Today, most practitioners and some others have grown accustomed to that claim, but at the time, it was received skeptically. Some skepticism still remains today. In fact, it is extremely challenging to identify comprehensively all significant scenarios, and much of the methodology presented in this Guide is devoted to responding to that challenge. The usefulness of doing this goes well beyond quantification of public risk and will be discussed further below. Both for the sake of technical soundness and for the sake of communication of the results, a systematic method in scenario development is essential and is a major theme of this Guide.

Significant controversy arose as a result of WASH-1400. These early controversies are discussed in many sources and will not be recapitulated in detail here. Methods have improved in some areas since the time of WASH-1400, but many of the areas considered controversial then remain areas of concern today. Completeness, which was mentioned above, was one issue. Quantification, and especially quantification of uncertainties, was also controversial then and remains so today. This topic, too, receives a great deal of attention in this Guide. Scrutability was an issue then; the formulation and presentation of many of the methods covered in this Guide are driven implicitly by a need to produce reports that can be reviewed and used by a range of audiences, from peer reviewers to outside stakeholders who are non-practitioners (i.e., communication is an essential element of the process).

Despite the early controversies surrounding WASH-1400, subsequent developments have confirmed many of the essential insights of the study, established the essential value of the approach taken, and pointed the way to methodological improvements. Some of the ideas presented in this Guide have obvious roots in WASH-1400; others have been developed since then, some with a view to NASA applications.

In addition to providing some quantitative perspective on severe accident risks, WASH-1400 provided other results whose significance has helped to drive the increasing application of PRA in the commercial nuclear arena. It showed, for example, that some of the more frequent, less

severe IEs (e.g., “transients”) lead to severe accidents at higher expected frequencies than do some of the less frequent, more severe IEs (e.g., very large pipe breaks). It led to the beginning of the understanding of the level of design detail that must be considered in PRA if the scenario set is to support useful findings (e.g., consideration of support systems and environmental conditions). Following the severe core damage event at Three Mile Island in 1979, application of these insights gained momentum within the nuclear safety community, leading eventually to a PRA-informed re-examination of the allocation of licensee and regulatory (U.S. Nuclear Regulatory Commission) safety resources. In the 1980s, this process led to some significant adjustments to safety priorities at NPPs; in the 2010s and beyond, regulation itself is being changed to refocus attention on areas of plant safety where that attention is more worthwhile.

3.1.3 Summary of PRA Motivation

In order to go deeper into the “why” of PRA, it is useful to introduce a formal definition of “risk.” (Subsequent sections will go into more detail on this.) Partly because of the broad variety of contexts in which the concepts are applied, different definitions of risk continue to appear in the literature. In the context of making decisions about complex, high-hazard systems, “risk” is usefully conceived as a set of triplets: scenarios, likelihoods, and consequences [3-2]. There are good reasons to focus on these elements rather than focusing on simpler, higher-level quantities such as “expected consequences.” Risk management involves prevention of (reduction of the frequency of) adverse scenarios (ones with undesirable consequences), and promotion of favorable scenarios. This requires understanding the elements of adverse scenarios so that they can be prevented, and the elements of successful scenarios so that they can be promoted.

PRA quantifies “risk metrics.” The term “risk metric” refers to probabilistic performance measures that might appear in a decision model: such things as the frequency or probability of consequences of a specific magnitude, or perhaps expected consequences. Risk metrics of interest for NASA include the probabilities of loss of crew or vehicle for some specific mission type, probability of mission failure, probability of large capital loss, etc. Figures of merit such as “system failure probability” can be used as risk metrics, but the phrase “risk metric” ordinarily suggests a higher-level, more consequence-oriented figure of merit.

In order to support resource allocation from a risk point of view, it is necessary to evaluate a comprehensive set of scenarios. This is logically required because “risk” depends on a comprehensive scenario set, not only on performance in a reference mission (e.g., a design basis). The set of scenarios may need to include events that are more severe than those specified in the design basis, and more success paths than were explicitly factored into the design basis. Additionally, system performance must be evaluated realistically. In order to support resource allocation decisions, the point is not usually to establish a boundary on system capability or reliability, but rather to quantify capability and reliability. In other words, risk-informed resource allocation requires identification and quantification of all risk-significant scenarios, where “risk-significant” depends on the context of the evaluation.

Finally, in all but the simplest cases, decision support requires that uncertainty be addressed. Because risk analysis frequently needs to address severe outcomes of complex scenarios, uncertainties may be highly significant. These need to be reflected in the decision model, not only because they may influence the decision, but also because it is important to understand which uncertainties strongly affect the decision outcome and are potentially reducible through testing or research.

In summary, PRA is needed when decisions need to be made that involve high stakes in a complex situation, as in a high-hazard mission with critical functions being performed by complex systems. Intelligent resource allocation depends on a good risk model; even programmatic research decisions need to be informed by a state-of-knowledge risk model. (Allocating resources to research programs needs to be informed by insight into which uncertainties' resolution offers the greatest payback.) Developing a comprehensive scenario set is a special challenge, and systematic methods are essential.

3.1.4 Use of PRA in the Formulation of a Risk-Informed Safety Case (RISC)

The above discussion has been carried out with emphasis on the role of PRA in assessing system adequacy, especially with regard to selection of design features. This sort of application began before "safety goals" were widely discussed. Increasingly, risk managers need to argue that system designs satisfy explicit risk thresholds; nowadays, even if there is no absolute regulatory or policy requirement, the promulgation of safety goals and thresholds creates an expectation that goals and thresholds will be addressed in the course of safety-related decision-making. This creates an issue for PRA, because in general, it is impractical or even fundamentally impossible to "prove" that the level of risk associated with a complex, real-world system is below a given decision threshold.

Partly because PRA results cannot be "proven," a "Risk-Informed Safety Case" (RISC) is developed [3]. The RISC marshals evidence (tests, analysis, operating experience) and commitments to adhere to specific manufacturing and operating practices in order to assure that PRA assumptions, including the performance and reliability parameters credited in the PRA, are fulfilled. Among the commitments needed to justify confidence in the safety of the system is a commitment to analyze operating experience on an ongoing basis, including "near misses," in order to improve operations, improve the risk models, and build additional confidence in the models' completeness. This is not the same as "proving" that the PRA results are correct, but it is the best proxy for safety that can be obtained.

In many NASA contexts, decisions regarding design features (especially safety features) are faced with competing objectives: for example, if a candidate safety system performs well but has a large mass, the decision to include it must be made carefully. Once design decisions are made, they need to be reflected in the RISC. Not only do the features need to be modeled: in addition, the trade process itself needs to be presented in the RISC. There are good reasons for this: it shows not only the decision-makers but also the risk-takers (e.g., the astronauts) that the best possible job has been done in trading safety, and documentation of the process creates a better starting point for future design exercises.

3.1.5 Management Considerations

PRA requires a methodical effort from a technically diverse team. Although individual scenarios are understandable by project engineers, explicit manual enumeration of all of them in detail is completely impractical. The essential characteristic of the methods widely applied in scenario development is that they map complex reality into a set of logical relationships so that they can be efficiently analyzed through computer-based algorithms based on input that has been carefully formulated by engineers. Development of a comprehensive scenario set for a complex facility or mission is almost necessarily a team effort, not only because of the volume of work but because of the diversity of technical disciplines involved. The above discussion has emphasized the need for a methodical approach. This point extends beyond the thought process itself.

Despite the use of computers, the effort required can be substantial. Scenario modeling is not typically accomplished in a single pass; formulation of the scenario model needs to be iterated with quantification of scenario frequencies. Needed design information and performance data are frequently scattered through many sources, rather than being compiled in a form that directly supports PRA applications. Practitioners should be cognizant of the issues when estimating level of effort needed for a given analysis.

3.2 Example

This subsection discusses a simplified example to illustrate the ideas presented above. The subject system is briefly described first. Then an overview of the analysis results is presented: the significant findings that emerge from the PRA of this example, and how they might be used by a decision maker. Then the analysis leading to these results is discussed with a view to showing how the techniques discussed above need to be applied in order to reach these findings.

3.2.1 Propellant Distribution Module Example

The subject of the analysis is a spacecraft propellant distribution module. The purpose of the analysis is to inform decisions regarding this module, and the analysis and its results will eventually be input to formulation of a Risk-Informed Safety Case. There are two independent and redundant sets of thrusters in the spacecraft. Both sets of thrusters are completely redundant for all functions. Figure 3-1 shows the propellant distribution module associated with one set of thrusters. As shown, the relevant portions are a hydrazine tank, two propellant distribution lines leading to thrusters, a normally-open isolation valve in each line, a pressure sensor in each line, and control circuitry capable of actuating the isolation valves based on pressure sensed in the distribution lines. When the attitude-control system signals for thruster operation, the controller opens the solenoid valves (not shown) to allow hydrazine to flow. Part of the design intent of this system is that in the event of a leak in the distribution lines, the leak should be detected by the pressure sensors (the leak should cause a pressure reduction) and thereafter should be isolated by closure of both isolation valves. The controller is designed to differentiate between the normal thruster operation and a leak. The scenarios analyzed in this example are those leading to (1) loss of vehicle or (2) loss of scientific data as a result of a hydrazine leak. The overall system design can tolerate a single isolated leak that does not cause damage to critical avionics, but a more broadly scoped model would, of course, address the possibility of additional failures. A complete model might also need to address the potential for a spurious isolation signal, taking a propellant distribution module off-line. The present example is narrowly scoped to the prevention and mitigation of a single leak and is formulated to illustrate the form and characteristic application of PRA results in a simplified way.

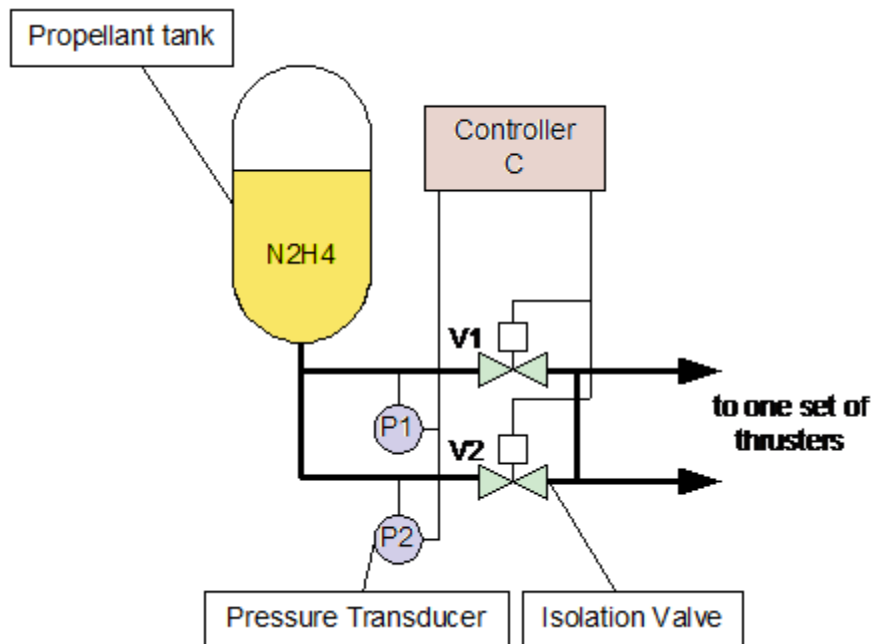


Figure 3-1. Simplified Schematic of a Propellant Distribution Module.

3.2.2 Selected Results

The scenarios leading to “loss of vehicle” are shown in Table 3-1, together with estimates of their frequencies (actually per-mission probabilities). In the second column, the scenarios are specified in terms of aggregated or functional-level events: success or failure of systems, occurrence or non-occurrence of particular phenomena. Typically, a given scenario can arise in different ways. For each system failure occurring in a particular scenario, there may be many distinct combinations of component-level failures that yield that system failure. Correspondingly, scenarios that involve several distinct system failures may contain a very large number of such combinations. These combinations are called “minimal cut sets (MCSs).”^a Each MCS of each scenario is also displayed in Table 3-1, along with the probabilities of the elements and the resulting probability of the MCS. The MCSs are one of the major outputs of a PRA. They are a basis for quantification of top event likelihood and also provide qualitative insight.

These results indicate that the frequency of “loss of vehicle” from this cause (hydrazine leak) is 1.02E-4 per mission, and that the dominant contributor to this frequency is the following scenario, having a mean frequency of 1.0E-4:

- Leak of hydrazine (symbol “IE”, frequency of 0.01) AND
- Leak location is upstream of isolation valves (implying that isolation cannot succeed) (symbol “L,” probability of 0.1) AND
- Physical damage actually occurring to wiring as a result of attack by hydrazine (symbol “/A2,” probability of 0.1) [leading to loss of vehicle].

a. A “cut set” is a set of conditions (such as failures of specific components) whose collective satisfaction causes the undesired outcome, which is loss of vehicle in this case. A *minimal* cut set is one that no longer causes the top event if any of its constituent conditions is not satisfied.

This contribution is said to be “dominant” because its magnitude is on the order of the overall result. In this case, other contributing scenarios are lower in probability by orders of magnitude. (Some analysts use a much looser definition of “dominant”; some will refer to the largest contributor as “dominant” even if it is a small fraction of the total result.)

Table 3-1. Scenarios Leading to "Loss of Vehicle" and Their Associated Frequencies.

Scenario	Description of Scenario (See Figure 3-7)	Cut Set	Symbol	Meaning	Probability	Total
3	Hydrazine Leak, Isolated Promptly but Avionics Fail Anyway	1	IE	Leak	1.0E-2	1.0E-7
			/A1	Avionics fail even after successful isolation	1.0E-5	
9	Hydrazine Leak, Detection Failure Leading to Isolation Failure, Avionics Failure	2	IE	Leak	1.0E-2	1.0E-7
			PP	Common cause failure of pressure transducers	1.0E-4	
			/A2	Avionics fail after unsuccessful isolation	1.0E-1	
		3	IE	Leak	1.0E-2	1.0E-7
			CN	Controller fails	1.0E-4	
			/A2	Avionics fail after unsuccessful isolation	1.0E-1	
		4	IE	Leak	1.0E-2	1.0E-9
			P1	Pressure transducer 1 fails	1.0E-3	
			P2	Pressure transducer 2 fails	1.0E-3	
			/A2	Avionics fail after unsuccessful isolation	1.0E-1	
6	Hydrazine Leak, Detection Succeeds but Isolation Fails, Avionics Failure	5	IE	Leak	1.0E-2	1.0E-4
			L	Leak occurs upstream of isolation valves	1.0E-1	
			/A2	Avionics fail after unsuccessful isolation	1.0E-1	
		6	IE	Leak	1.0E-2	9.0E-7
			/L	Leak occurs downstream of isolation valves	9.0E-1	
			V2	Isolation valve V2 fails to close	1.0E-3	
			/A2	Avionics fail after unsuccessful isolation	1.0E-1	
		7	IE	Leak	1.0E-2	9.0E-7
			/L	Leak occurs downstream of isolation valves	9.0E-1	
			V1	Isolation valve V1 fails to close	1.0E-3	
			/A2	Avionics fail after unsuccessful isolation	1.0E-1	

3.2.3 High-Level Application of Results

The absolute magnitude of the overall risk has some usefulness without regard to the characteristics of the dominant contributor. A detailed exposition of the decision-making potential is beyond the scope of the present subsection, but even at this stage, consideration can be given to the level of unacceptability of this frequency of loss of spacecraft. The uncertainty in this quantity is also of interest and is discussed further in Section 3.3.6. Here, we suppose that the frequency is considered high enough that prevention measures are worth evaluating.

Quite generally, a scenario is prevented through prevention of all of its MCSs, and each MCS is prevented through prevention of any of its elements. In this example, we can prevent the dominant scenario by preventing any one of its elements. This suggests that we consider preventing one or more of the following:

- Occurrence of hydrazine leak
- Occurrence of leak upstream of isolation valves
- Conditional damage due to hydrazine attack.

In this example, an overall leak frequency is quantified and then split into a fraction upstream of the isolation valves (“L”) and a complementary fraction downstream (“/L”). Some ways of reducing the upstream fraction would leave the downstream fraction unaltered, while other methods would reduce the upstream fraction while increasing the downstream fraction. For example, keeping the piping layout as is, but relocating the isolation valves as close to the source as possible, would tend to reduce the upstream fraction (by reducing the length of piping involved) and increase the downstream fraction (by increasing the length of piping involved). On the other hand, reducing the number of fittings in the upstream portion alone (if it were practical to do this) might reduce the upstream frequency while leaving the downstream frequency unchanged. Table 3-2 shows the effect on scenario frequency of reducing the upstream frequency by a factor of 2, while leaving the downstream fraction unchanged. Essentially, the frequency of this scenario is reduced by whatever reduction factor is achieved in the frequency of upstream leaks.

The remaining element is failure of avionics wiring, given that it is subjected to hydrazine attack. In the example, this has been modeled as having a probability of 0.1. This is a function of the physical characteristics of the wiring, in particular its chemical susceptibility to hydrazine. If it is practical to use different insulation, sheathing, conduit, etc. that is impervious to hydrazine, so that the conditional probability of failure given hydrazine attack is reduced, then the scenario frequency will be reduced proportionally. If it is practical to re-route the wiring to reduce the exposure, this helps as well. Table 3-2 shows the effect of an overall order of magnitude reduction in the probability of damage to critical avionics.

Because these two prevention measures are independent of each other, their probabilities combine multiplicatively in the dominant scenario probability. The overall potential probability reduction from applying them jointly is a factor of 20, as shown in Table 3-2. If the measures actually adopted to achieve these reductions also influenced other scenarios, or even changed the logic modeling, then it would be important to examine their impact in the context of the overall model. Re-routing the wiring, for example, might create other hazards. Examining risk reduction measures in too narrow a context can lead to distorted conclusions.

Table 3-2. Examination of Risk Reduction Strategies for the Example Problem.

	Structure of Dominant Scenario			
	IE: Leak occurs	Leak occurs upstream of isolation valves	Leak damages critical avionics	Frequency
OPTIONS	IE	L	/A2	
Do nothing	0.01	0.1	0.1	1.0E-4
Option 1: Reduce the likelihood of leak between the propellant tank and isolation valves (e.g., change in piping design)	0.01	0.05 (see note below)	0.1	5.0E-5
Option 2: Reduce susceptibility of avionics to leak (e.g., rerouting of wires and fortify wire harnesses)	0.01	0.1	0.01 (see note below)	1.0E-5
Option 1 and 2	0.01	0.05	0.01	5.0E-6
Note: The numerical values shown in this table are hypothetical.				

The above discussion has been carried out applying the results to address a design issue. The use of the analysis does not stop there, however; the analysis also plays a role in the risk-informed safety case (RISC), which marshals evidence, including this analysis, to support a decision regarding the overall suitability of the system, and provides a roadmap to implementation aspects needed to make the safety claim “come true.” The following aspects of this example will be captured in the risk-informed safety case.

1. The documentation of the analysis itself will capture the system configuration and the concept of operations on which the analysis is predicated.
2. The results (together with the rest of the risk analysis) will show how safe the system is (providing evidence that safety threshold requirements are met).
3. The results, together with documentation of the process that was followed to address the dominant risk contributor, will provide evidence that the configuration is not only adequate (thresholds are satisfied) but also optimal (goals are addressed).
4. Since the risk reduction measures are configurational in nature, a functional test of the wiring will not confirm that the routing minimizes the risk of hydrazine damage, so confirmation of this aspect may require inspection at the time of system acceptance.

3.2.4 Summary

From the risk analysis,

- A quantitative estimate of risk was obtained,
- Potential risk reduction measures were identified, and

- The potential benefits of these prevention measures were quantified.

If trustworthy, these results are clearly of significant use to a decision maker. What is required for these results to be trustworthy?

First, the scenario set must be substantially complete. If dominant scenarios are not identified, then the overall frequency result is in error. Moreover, if these unidentified scenarios have ingredients that are not present in the scenarios that are identified, then potentially useful prevention measures are not identifiable from the results.

The requirement for completeness, and the potential complexity of the scenario model, argue for development of the model in a hierarchical fashion. In Table 3-1, contributors are identified at the “scenario” level and at the “cut set” level. Several of the elements of PRA discussed in the next section have evolved to support development of the scenario model in this hierarchical fashion. Completeness is easier to assess for a model developed in this way. Arguably, at the functional level of detail in the scenario specification, completeness should be achievable in principle: if we know what functional performance corresponds to “success,” then we know what functional performance corresponds to “failure.” At the basic event level, the argument is more difficult, because it is difficult to be sure that all causes have been identified. However, the tools discussed in the following section have a lot to offer in this regard.

Even if the scenario set is substantially complete, poor decisions may result if the numbers used in quantification are significantly off. The relative dominance of scenarios may be misstated, in which case attention will be diverted from prevention of more likely scenarios to prevention of less likely ones. The overall risk may be overstated or understated, distorting priorities for different prevention measures. The absolute benefit of any given prevention measure will be in error. All of these issues are capable of significantly misinforming the decision maker.

3.3 Elements of PRA

This subsection discusses the elements of PRA. Major elements of PRA are introduced and briefly described; each is then illustrated with respect to the very simplified example introduced above. For simplicity, the example emphasizes the logic-based (ET/FT) modeling approach, however the concepts described in this section are equally applicable to other modeling approaches such as simulation.

The PRA ultimately presents a set of scenarios, frequencies, and associated consequences, developed in such a way as to inform decisions regarding the allocation of resources to accident prevention. This allocation could be changes in design or operational practice, or could be a finding that the design is optimal as is. Decision support in general requires quantification of uncertainty, and this is understood to be part of modeling and quantification.

A scenario contains an IE and (usually) one or more pivotal events leading to an end state (see Figure 3-2). As modeled in most PRAs, an IE is a perturbation that requires some kind of response from operators, pilots, or one or more systems. Note that for an IE to occur, there may need to be associated enabling event(s) that exist (e.g., for a fire IE to occur, there would need to be combustible material present). The pivotal events in a scenario include successes or failures of responses to the IE, or possibly the occurrence or non-occurrence of external conditions or key phenomena. Then, the scenario end state(s) are formulated according to the decisions being supported by the analysis. Scenarios are classified into end states according to the kind and severity of consequences, ranging from completely successful outcomes to losses of various kinds, such as:

- Loss of life or injury/illness to personnel (including public, astronauts [i.e., loss of crew (LOC)], ground crew, and other workforce);
- Damage to, or loss of, equipment or property (including space flight systems [i.e., loss of vehicle (LOV)], program facilities, and public properties);
- Loss of mission (LOM);
- Unexpected or collateral damage;
- Loss of system availability; and
- Damage to the environment (Earth and planetary contamination).

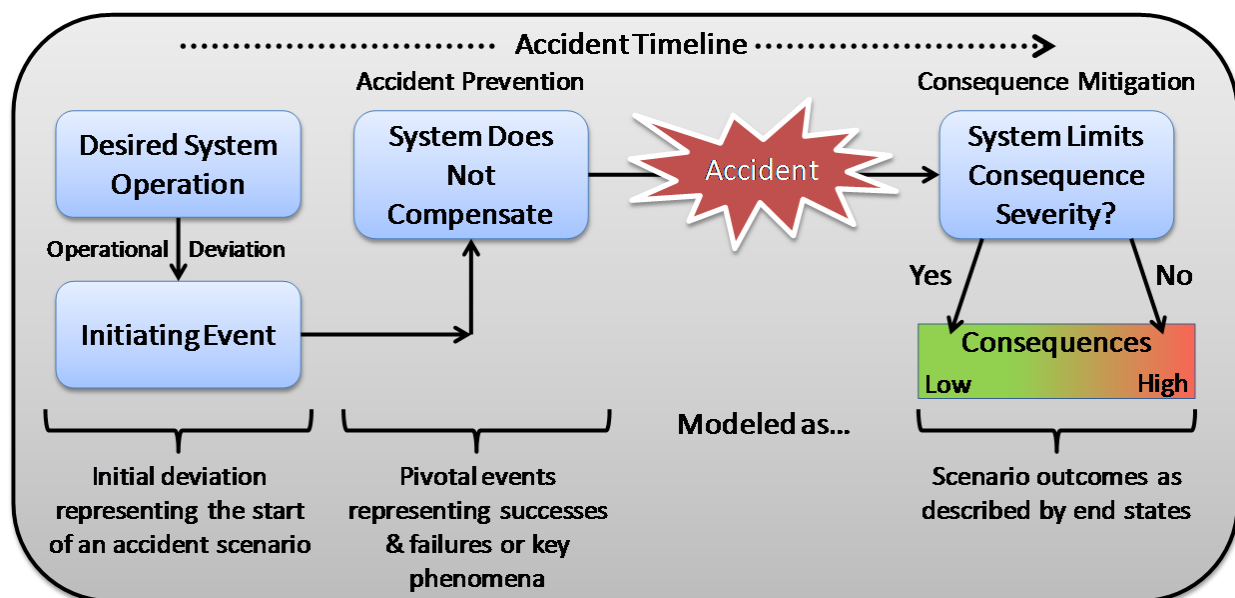


Figure 3-2. The Concept of a Scenario.

These consequence types are identified by NPR 8715.3C [3-4] as consequence types to be identified, analyzed, reduced, and/or eliminated by the program / project safety and mission success activity. These and other consequences of concern need to be identified early in the project so that the model can reflect the necessary distinctions and analysis can be planned to address them.

3.3.1 Identification of Initiating Events

Chapter 4 of this guide, discusses approaches for identification of IEs, including the use of master logic diagrams (MLDs). An MLD (Figure 3-3) is a hierarchical, top-down display of IEs, showing general types of undesired events at the top, proceeding to increasingly detailed event descriptions at lower tiers, and displaying initiating events at the bottom. The goal is not only to support identification of a comprehensive set of IEs, but also to group them according to the challenges that they pose (the responses that are required as a result of their occurrences). IEs that are completely equivalent in the challenges that they pose, including their effects on subsequent pivotal events, are equivalent in the risk model.

A useful starting point for identification of IEs is a specification of “normal” operation in terms of (a) the nominal values of a suitably chosen set of physical variables and (b) the envelope in

this variable space outside of which an IE would be deemed to have occurred. A comprehensive set of process deviations can thereby be identified, and causes for each of these can then be addressed in a systematic way.

The present example corresponds to a small piece of a potentially large MLD. An early step in the process is a focus on the consequence types of interest. In this case, two consequence types of interest have been identified: loss of spacecraft and loss of scientific data. Both imply a loss of at least the scientific mission, but the additional loss of spacecraft is a more severe event than just loss of scientific data. For these consequence types, certain functional failures are obvious candidates for initiating scenarios leading to these consequences, and physical damage to certain system elements is an obvious mechanism potentially leading to functional failure.

It should be kept in mind in this example that failure of the thrusters is not the IE being analyzed: rather, loss of the function(s) supported by the wiring (avionics, scientific instruments) is the concern. Both of these consequence types can be caused by physical damage to wiring.^a Among many possible causes of physical damage to wiring is attack by hydrazine. Accordingly,

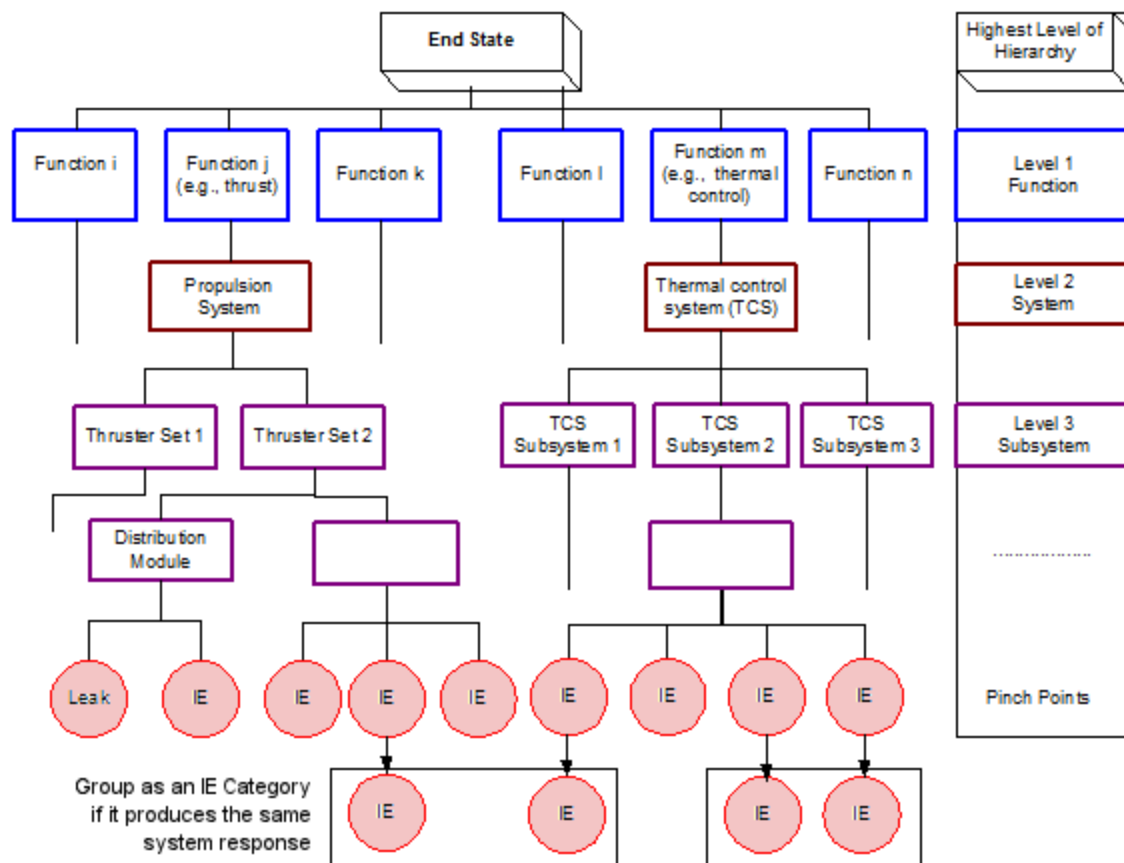


Figure 3-3. Typical Structure of a Master Logic Diagram (MLD).

a. A propellant leak could cause an attitude disturbance exceeding the ability of the spacecraft to recover. For simplicity, this loss of attitude-control function as a result of a leak is not considered in this example.

an MLD development should identify this potential. Indeed, the design intent of the system clearly implies recognition by the designer of the undesirability of an unisolated hydrazine leak (though there are reasons for this besides the potential for damage to wiring).

3.3.2 Application of Event Sequence Diagrams and Event Trees

The scenarios that may ensue from a given IE may be developed initially in a timeline, block diagram, event tree (ET), or Event Sequence Diagram (ESD). The ESD is essentially a flowchart, with paths leading to different end states; each path through this flowchart is a scenario. Along each path, pivotal events are identified as either occurring or not occurring (refer to Figure 3-4 and Figure 3-5). It will be seen below that an ESD can be mapped into an ET, which relates more directly to practical quantification of accident scenarios, but the ESD representation has the advantage over the ET of enhancing communication between risk engineers, designers, and crews. In situations that are well covered by operating procedures, the ESD flow can reflect these procedures, especially if the procedures branch according to the occurrence of pivotal events (due to the flowchart nature of the ESD). Instrument readings that inform crew decisions can be indicated at the appropriate pivotal event. This representation should make more sense to crews than ETs do. At each pivotal event along any given path, the events preceding that event are easily identified, so that their influence on the current pivotal event can be modeled adequately. A good deal of information (e.g., system-level mission success criteria at each pivotal event) can also be displayed on the ESD, making it a very compact representation of a great deal of modeling information.

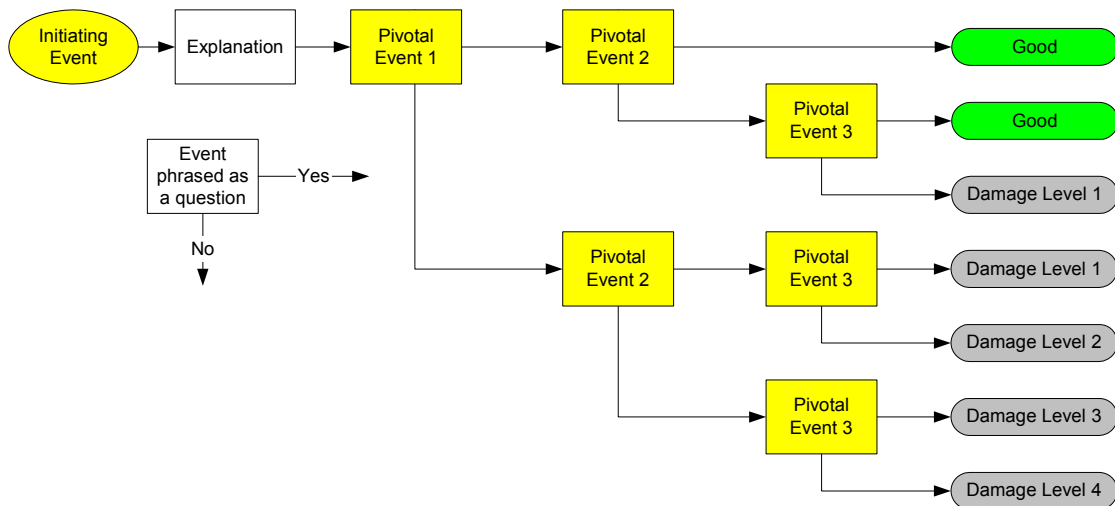


Figure 3-4. Typical Structure of an Event Sequence Diagram (ESD).

From the ESD, it is possible to derive an ET (see Figure 3-5). An ET distills the pivotal event scenario definitions from the ESD and presents this information in a tree structure that is used to help classify scenarios according to their consequences. The headings of the ET are the IE, the pivotal events^a, and the end state. The “tree” structure below these headings shows the possible

^a Pivotal events specify only two options, success or failure. This is a simplification for the analysis. Gradually degraded states are not considered in this approximation called a “Bernoulli trial.”

scenarios ensuing from the IE, in terms of the occurrence or non-occurrence of the pivotal events. Each distinct path through the tree is a distinct scenario. According to a widespread but informal convention, where pivotal events are used to specify system success or failure, the “down” branch is considered to be “failure.” For example, begin at the upper left of the tree in Figure 3-4. At this point on the tree, the IE has occurred. Moving to the right along this path, we come to a branch under “Pivotal Event 1.” The path downward from this point corresponds to scenarios in which the system queried under “pivotal event 1” fails; the path upward corresponds to success of that system. Continuing the example, suppose that all of the pivotal events in Figure 3-4 query the successful operation of specific systems. In the top-most path in Figure 3-4 leading to the end state “good,” the following occur:

- The IE
- Success of system 1
- Success of system 2.

In the next path down, the following occur:

- The IE
- Success of system 1
- Failure of system 2
- Success of system 3.

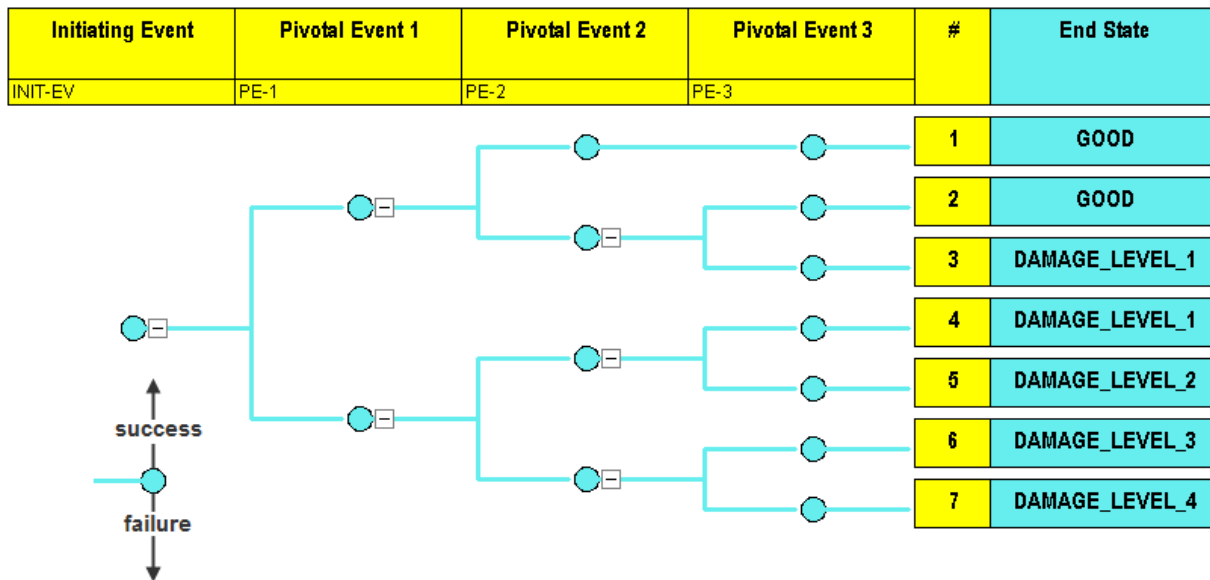


Figure 3-5. Event Tree Representation of the ESD Shown in Figure 3-4.

Though an ET and an ESD can be logically equivalent, it is important to recognize that the actual structure of an ET derived from a given ESD is not completely specified by the ESD structure alone but may depend on the relationships between pivotal events and the consequence types of interest. For example, in Figure 3-4, the failure or success of system 3 does not change the outcome as long as both systems 1 and 2 succeed. For this reason, “pivotal event 3” is not queried in the top most path (i.e., sequence 1) of the ET.

In most ETs, the pivotal event splits are binary: a phenomenon either does or does not occur, a system either does or does not fail. This binary character is not strictly necessary; some ETs show splits into more than two branches. What is necessary is that distinct paths be mutually exclusive and quantified as such (at least to the desired level of accuracy).

ETs made their first appearance in risk assessment in the WASH-1400 reactor safety study, where they were used to generate, define, and classify scenarios specified at the pivotal event level. Because an ET is a useful picture of a very complex calculation, many PRA software packages base their approaches on ET representations of scenarios.

In general, an ESD will reflect the design intent of the system(s) being analyzed. In the propellant distribution module example, the design of the system addresses mitigation of hydrazine leakage by the safety function “closure of the isolation valves in the event of a hydrazine leak as sensed by decreasing pressure in the distribution lines.” This design intent implies at least one, and potentially several, pivotal events.

Examination of the simplified system schematic shows that successful performance of the isolation function is conditional on the location of the leak. Leaks upstream of the isolation valve cannot be isolated by closure of the valves. Therefore, leak location has to be reflected either as a pivotal event in the ESD or in the definition of the IE itself (i.e., develop an ESD just for the IE “leak upstream of isolation valves”).

Recognition of the potential for a leak in this system that cannot be isolated provides an important example of the value of “completeness.” Failure to recognize the potential for this type of leak would lead to missing the dominant scenario in this example. This would understate the risk and lose an opportunity to consider potentially beneficial design changes. Given that the designers provided an isolation function specifically to address leaks, it is easy enough to imagine supposing that leaks were no longer an issue, and missing this potential. Experience has shown the value of systematic approaches for identification of this kind of situation.

Attack of the wiring, given an unisolated hydrazine leak, is not necessarily a given. In many situations, it is not practical to model all physically possible permutations of a messy problem in fine detail. In this case, the actual flow from a leak might depend in detail on the size, shape, and precise location of the leak, as well as the orientation of the spacecraft and numerous other factors. In many modeling situations, analogous complicating factors will govern the actual likelihood of a consequence that is clearly possible but far from assured. In this situation, the originally assigned probability of 0.1 is associated with damage to wiring for critical avionics.

Figure 3-6 shows an ESD for this IE and these pivotal events (for simplicity we assume the functionality of the redundant set of thrusters is not affected by hydrazine attack and omit consideration of other common cause interactions with the second thruster subsystem).

Given the ESD, an initial ET is developed (see Figure 3-7). Later, the ET in Figure 3-8 shows a revision for this example. Per the earlier discussion, the “down” branches under each pivotal event correspond to an adverse outcome for that pivotal event: either a system failure or an adverse phenomenon. In Figure 3-7, two pivotal events are defined (as in the ESD): leak detection and leak isolation. The subsequent sequence evolution is conditional on whether the leak was isolated, not on whether it was detected. Therefore, in Figure 3-8, it is shown that these two can be combined into one, leading to a more compact ET (and fewer scenarios to compute) without loss of information. Only redundant scenarios are eliminated.

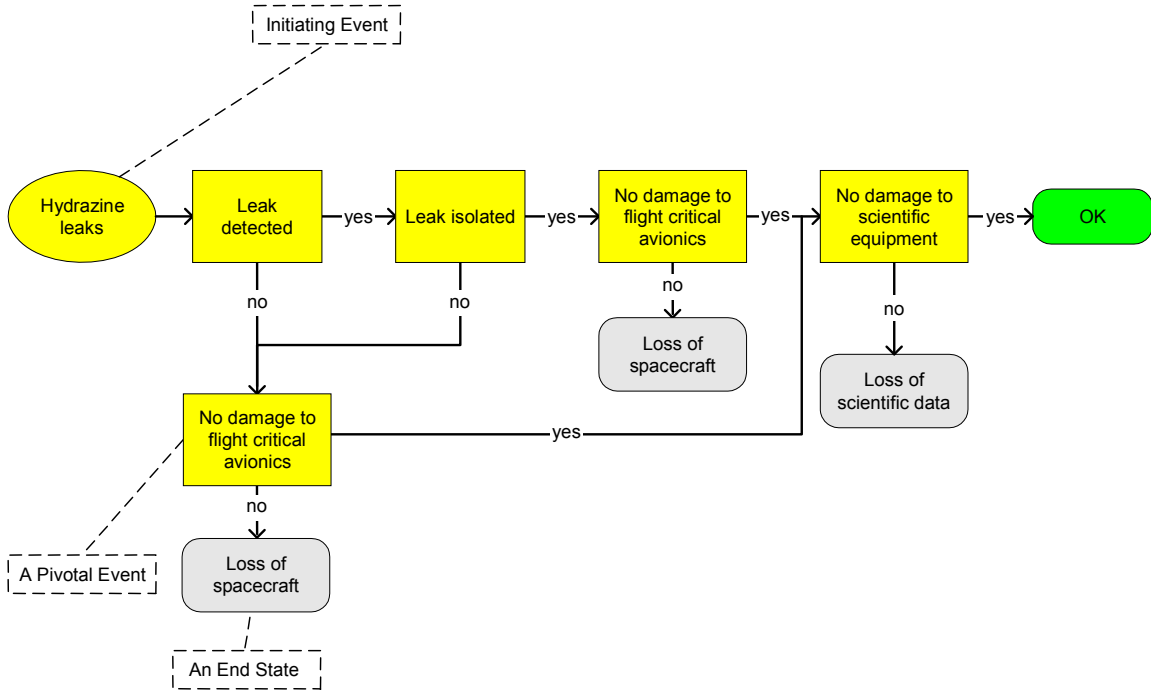


Figure 3-6. ESD for the Hydrazine Leak.

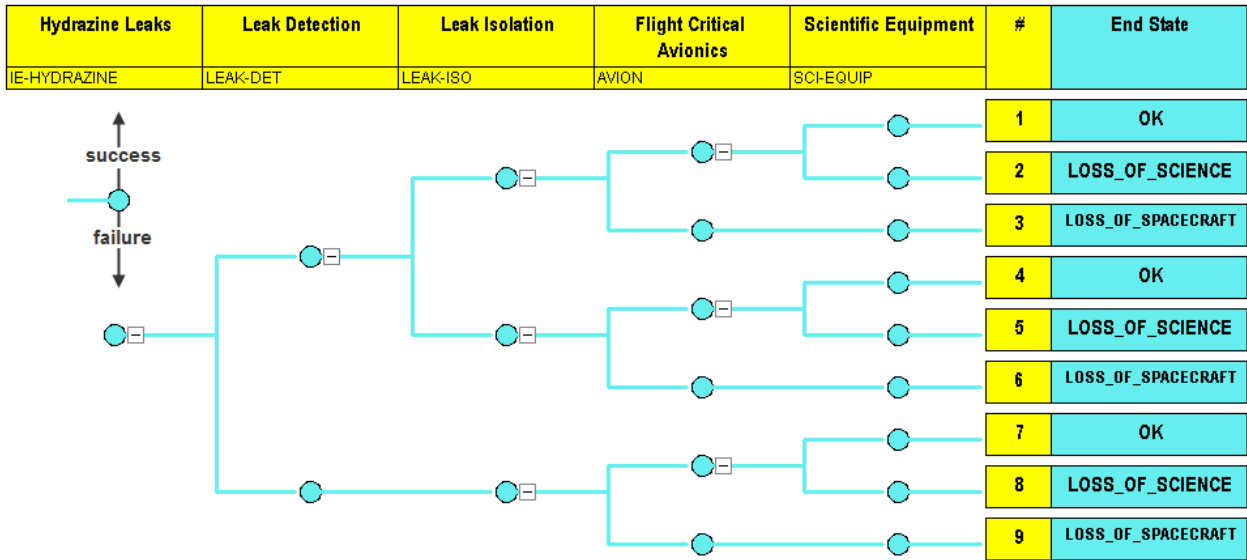


Figure 3-7. ET for the Hydrazine Leak.

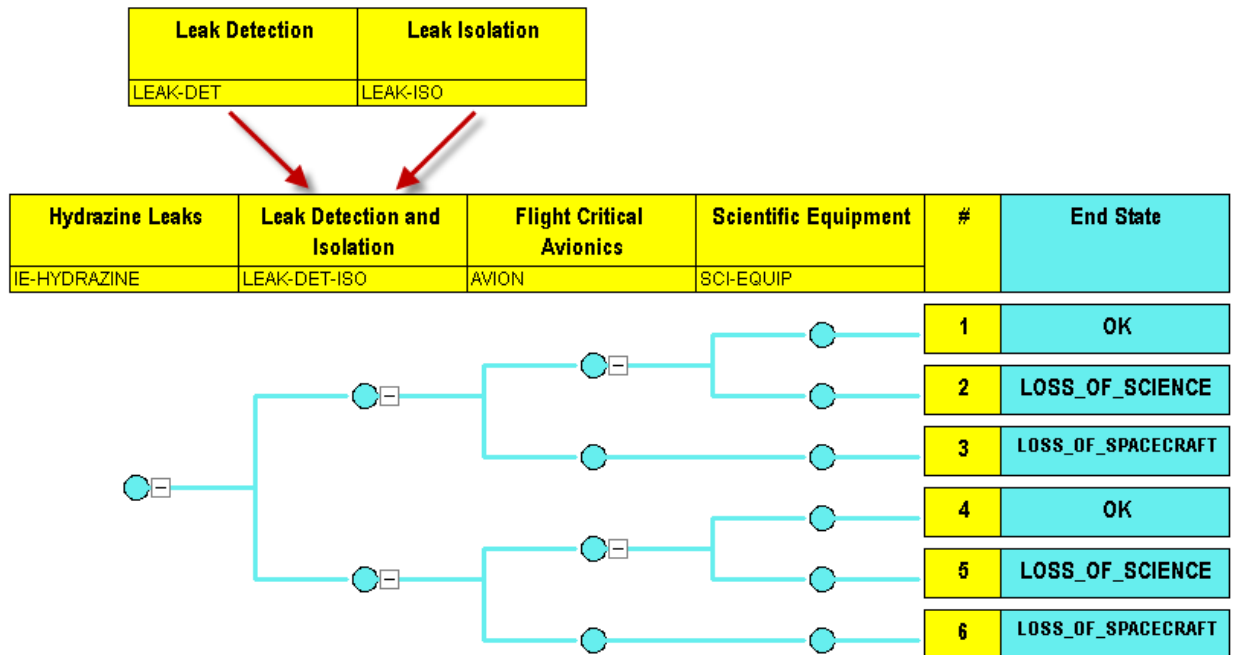


Figure 3-8. Revised ET for the Hydrazine Leak.

3.3.3 Modeling of Pivotal Events

Pivotal events must be modeled in sufficient detail to support valid quantification of scenarios. As a practical matter, the model must reach a level of detail at which data are available to support quantification of the model's parameters. Additionally, much of the time, pivotal events are not independent of each other, or of the IEs; the modeling of pivotal events must be carried out in such a way that these conditionalities are captured properly. For example, pivotal events corresponding to system failure may have some important underlying causes in common. If the purposes of the PRA are to be served—if such underlying causes are to be identified and addressed—it is imperative to capture such conditionalities in the scenario model. If pivotal events were known to be independent of each other, so that their probabilities could be combined multiplicatively, there would be less reason to analyze them in detail; it is because they can be mutually conditioned by shared influences that their modeling in some detail is important.

Complex pivotal events can frequently be modeled using fault trees (FTs). An FT is a picture of a set of logical relationships between more complex (more aggregated) events such as system-level failures, and more basic (less aggregated) events such as component-level failures. FT modeling is applicable not only to modeling of hardware failures, but also other complex event types as well, including descriptions of the circumstances surrounding software response and crew actions.

The mapping of scenarios into logic representations leans heavily on engineering analysis: physical simulation of system behavior in specified conditions, determination of time available for crew actions, determination of the severity of the consequences associated with scenarios. Behind every logic model is another body of modeling whose results are distilled into the logical relationships pictured in the scenario model. Assignment of system states into “success” or “failure” depends on such modeling, as does classification of scenarios into consequence categories. The specification of the physical system states that are deemed “successful” system

responses to a given challenge is the “mission success criterion” for that challenge. The FT logic for system response to a given challenge yields a logic expression for system failure in terms of combinations of basic events that violate the mission success criterion.

The FT leads to a representation of the top event “Pivotal Event Fails To Occur” in terms -of combinations (potentially many, many combinations) of basic events such as “component x fails.” This enables the transformation of scenarios specified in terms of pivotal events to scenarios specified in terms of basic events. As mentioned above, basic events that appear in multiple Pivotal Events correspond to potentially significant interdependencies. The development of FTs must be carried out in such a way that these interdependencies are properly recognized. This has implications for the level of detail to which basic events are developed by the analysts, and the way in which they are designated and processed in scenario generation and quantification.

3.3.3.1 Pivotal Events in the Simple Example

The FTs corresponding to failure of detection and failure of isolation are shown in Figure 3-9. Please note the FTs are developed for failure of pivotal events of Figure 3-7.

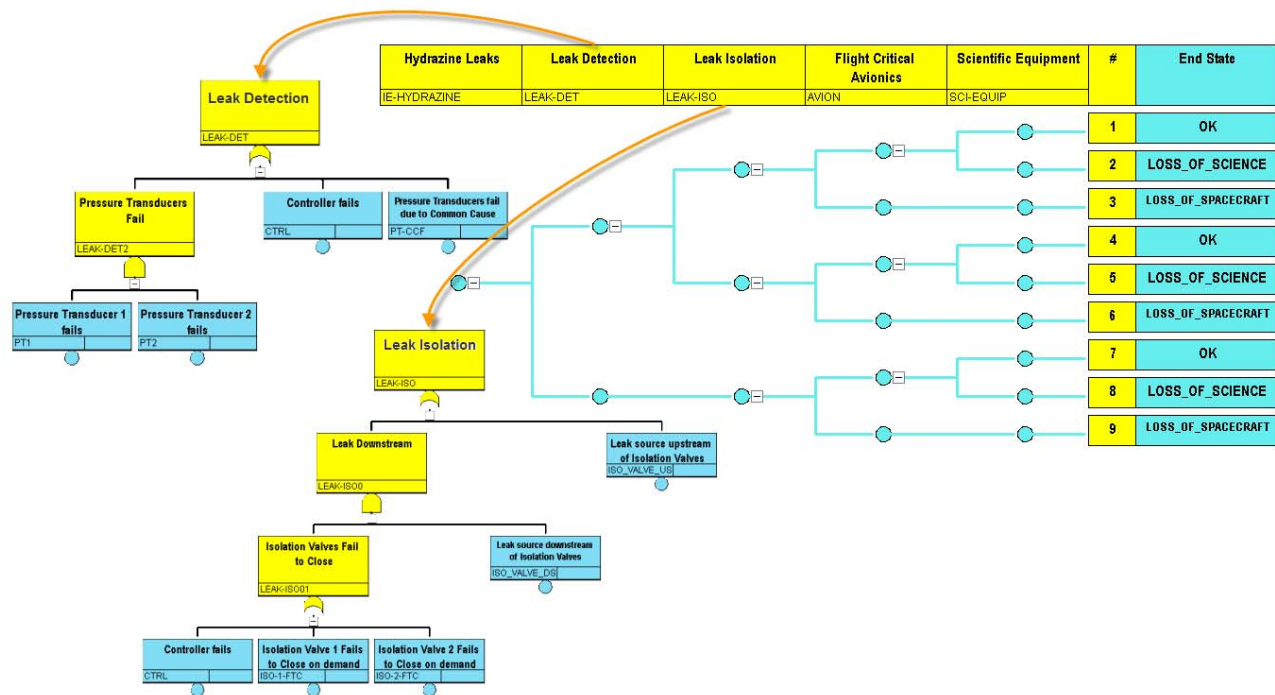


Figure 3-9. Fault Trees for Failure of Leak Detection and Failure of Isolation.

It is possible that the probability of wiring failure conditional on an unisolated leak would be different for upstream and downstream leaks, as a result of differing amounts of wiring being co-located with the upstream segments and the downstream segments, but this is not a feature of the present example.

3.3.3.2 Failure of Leak Detection and Failure of Isolation Given Detection Successful

Failure of the function is due either to failure to detect the leak or failure to isolate it, given detection. Because of the relative complexity of these pivotal events, failure of leak detection and failure of isolation given detection are appropriately addressed using FTs, which are shown in Figure 3-9. Each FT is a picture of the relationships that link its top event (e.g., “Leak not detected”) to its basic events (“Controller fails,” “common cause failure of pressure transducers,” “pressure transducer 1 fails,” “pressure transducer 2 fails”). The symbol under the top event “Leak not detected” is an OR gate, meaning that the top event occurs if any of the inputs occur. The symbol linking “Pressure Transducer 1 fails” and “Pressure Transducer 2 fails” to the top event is an AND gate, meaning that both inputs must be satisfied in order for its output condition to occur. This means that failure of an individual transducer (with the other transducer still working) will not trigger the AND gate, and therefore will not trigger the OR gate. This fault tree confirms that “Leak not detected” will result from “Controller fails” OR “Pressure Transducers fail due to Common Cause” OR “Pressure Transducer 1 fails” AND “Pressure Transducer 2 fails”. These are, in fact, the “minimal cut sets” for the pivotal event “Leak not detected.”

In real examples, functional FTs are far more complex and must be processed by computer. In a properly structured FT, the individual logical relationships are tautological viewed in isolation, but surprising complexity can be manifest in the top-level results if certain basic events appear in more than one place on the tree. Moreover, when the MCSs for pivotal events are logically ANDed together to form scenario-level expressions in terms of basic events, the conditionality between pivotal events is to be captured through the appearance in both pivotal event FTs of the basic events that correspond to this conditionality. The logic expression for the whole scenario then properly reflects the conditionality of the pivotal events.

One way of failing isolation is that the leak cannot be isolated by virtue of being upstream of the isolation valves (as shown on the isolation FT as event L). If the leak can be isolated, failure to isolate given detection is caused by failure of either isolation valve, or failure of the controller to issue the actuation signal. This FT also shows the event “/L” (NOT L) (leak is NOT upstream of the isolation valves, i.e., IS downstream of the isolation valves) ANDed with the logic associated with failure of the isolation function, given detection. This is done in order to make the quantification more accurate. If the probability of event “leak occurs upstream of isolation valves^a is small, $\Pr(/L)$ is nearly equal to 1, so little would be lost by suppressing event /L in that spot on the fault tree; but if $\Pr(/L)$ were a smaller number, neglect of it in the cut set quantification would overstate the probability contribution from scenarios in which the valves or the controller failed.^b

3.3.4 Quantification of (Assignment of Probabilities or Frequencies to) Basic Events

One of the defining characteristics of a basic event is that it should be directly quantifiable from data, including, if necessary, conditioning of its probability on the occurrence of other basic

a. This probability would be based on an engineering assessment of the physical characteristics of the upstream and downstream distribution lines (number and type of fittings, ...) and the operating environments of each (cycling of mechanical stresses ...).

b. Strictly speaking, when an event such as L and its complement (/L) both appear in an FT, as is the case in this example, the model is said to be non-coherent. For such a model, we should speak of “prime implicants” rather than MCSs. Subtleties of interpretation and of quantification arise for non-coherent models. These are beyond the scope of an overview discussion.

events. Usually, basic events are formulated to be statistically independent, so that the probability of the joint occurrence of two basic events can be quantified simply as the product of the two basic event probabilities. Basic events corresponding to component failure may be quantified using reliability models. A simple and widely used model is the exponential distribution that is based on the assumption of constant failure rate (see Figure 3-10). Other kinds of models may be appropriate for basic events corresponding to crew errors, and still others to basic events corresponding to simple unavailability.

In the example, several kinds of basic events are quantified:

- The IE, corresponding to failure of a passive component, quantified on a per-mission basis;
- Failures of active components, such as valves and the controller;
- A common cause event (CCE) of both pressure sensors;
- Events corresponding to phenomenological occurrences (probability of failure of wiring, given hydrazine attack).

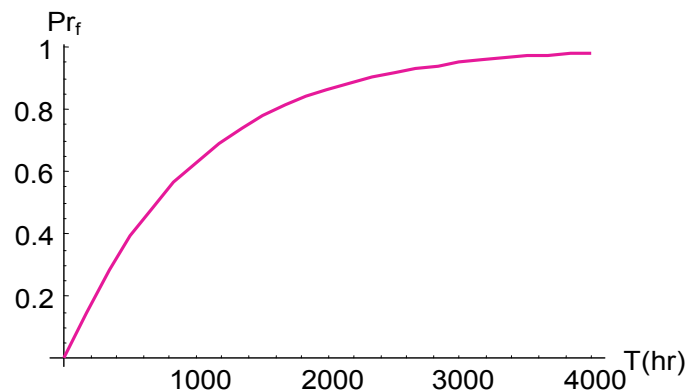


Figure 3-10. Exponential Distribution Model [$\text{Pr}_f(t) = 1 - \exp(-\lambda t)$ for $\lambda = 0.001$ per hour].

The probabilities of these events are quantified probabilistically, i.e., using probability density distributions that reflect our uncertainty—our limited state of knowledge—regarding the actual probabilities of these events. For basic events that are well understood and for which a substantial experience base exists, the uncertainty in probability may be small. The probability of basic events for which the experience base is limited may be highly uncertain. In many cases, we are sure that a given probability is small, but we are not sure just how small.

In this example, all event probabilities [other than $\text{Pr}(L)$, which is determined by the value of $\text{Pr}(L)$] were assumed to be lognormally distributed. Means and error factors (a measure of dispersion) for these event probabilities are shown in Table 3-3. The mean is the expected value of the probability distribution, and the error factor is the ratio of the 95th percentile of the distribution to the median.

Table 3-3. Lognormal Distribution Parameters for Basic Event Probabilities.

Event	Mean	Error Factor
CN	1.00E-04	10
P1	1.00E-03	3
P2	1.00E-03	3
PP	1.00E-04	5
L	1.00E-01	3
V1	1.00E-03	3
V2	1.00E-03	3
/L	Dictated by L since /L = 1 - L	
/A1	1.00E-05	5
/A2	1.00E-01	3
IE	1.00E-02	4

3.3.5 Uncertainties: A Probabilistic Perspective

Randomness (variability) in the physical processes modeled in the PRA imposes the use of probabilistic models (referred to as “aleatory” models), which is central to risk analysis. The development of scenarios introduces model assumptions and model parameters that are based on what is currently known about the physics of the relevant processes and the behavior of systems under given conditions. It is important that both natural variability of physical processes (i.e., aleatory or stochastic uncertainty) and the uncertainties in knowledge of these processes (i.e., “epistemic” or state-of-knowledge uncertainty) are properly accounted for.

In many cases, there is substantial epistemic uncertainty regarding basic event probability. Failure rates are uncertain, sometimes because failure information is sparse or unavailable, and sometimes because the very applicability of available data to the case at hand may be in doubt. Uncertainty in basic event probabilities engenders uncertainty in the value of the risk metric. The most widely used method for determining the uncertainty in the output risk metric is to use a sampling process (e.g., Monte Carlo sampling), because of the complexity of the risk expression and the magnitudes of the uncertainties of the basic events. In the sampling process, values for each basic event probability are derived by sampling randomly from each event’s probability distribution; these are combined through the risk expression to determine the value of the risk metric for that sample. This sampling process is repeated many times to obtain a distribution on the risk metric (the number of samples is determined based on the precision needed in properties of the distribution of the risk metric).

Uncertainty can have a strong effect on the output mean of the risk metric. Even when the output mean is not strongly affected, it may be of interest to understand the percentiles associated with the output distribution (e.g., the probability of accident frequency being above a certain value of concern), such as when assessing if a safety threshold has been met. Depending on the decision context, it can also be very useful to quantify the value to the decision maker of investing resources to reduce uncertainty in the risk metric by obtaining additional information that would reduce the uncertainty in selected parameters. In other words, the value of reducing the uncertainty in the input to the decision can be quantified, and an informed decision can be made regarding whether to invest analytical resources in narrowing the uncertainty of a specific parameter.

How is uncertainty characterized in the first place? If directly applicable data for a specific parameter are sufficiently plentiful, it may be straightforward to derive an uncertainty distribution, or even (if there is relatively little uncertainty in the parameter) to neglect uncertainty in that parameter. However, in many cases, a useful assessment of uncertainty cannot be obtained solely from existing performance data (e.g., Bernoulli trials of a particular probability). This is certainly true when there are no directly applicable data, as for certain phenomenological basic events. Even for component-related basic events, the applicability of certain performance data may be in doubt if obtained under different operating conditions or for a different manufacturer. In these cases, it is necessary to do the best that one can, integrating such information as is available into a state-of-knowledge probability distribution for the parameter in question.

An important tool for developing these probability distributions is Bayes' Theorem, which shows how to update a "prior" distribution over basic event probability to reflect new evidence or information, and thereby obtain a "posterior" distribution. (Refer to Figure 3-11.) Application of Bayes' Theorem is discussed at length in Chapter 5. The general idea is that as more evidence is applied in the updating process, the prior distribution is mapped into a posterior distribution that comports with the new evidence. If there is substantial uncertainty in the prior, corresponding to relatively few data supporting the prior, then new evidence will tend to dominate the characteristics of the posterior distribution.

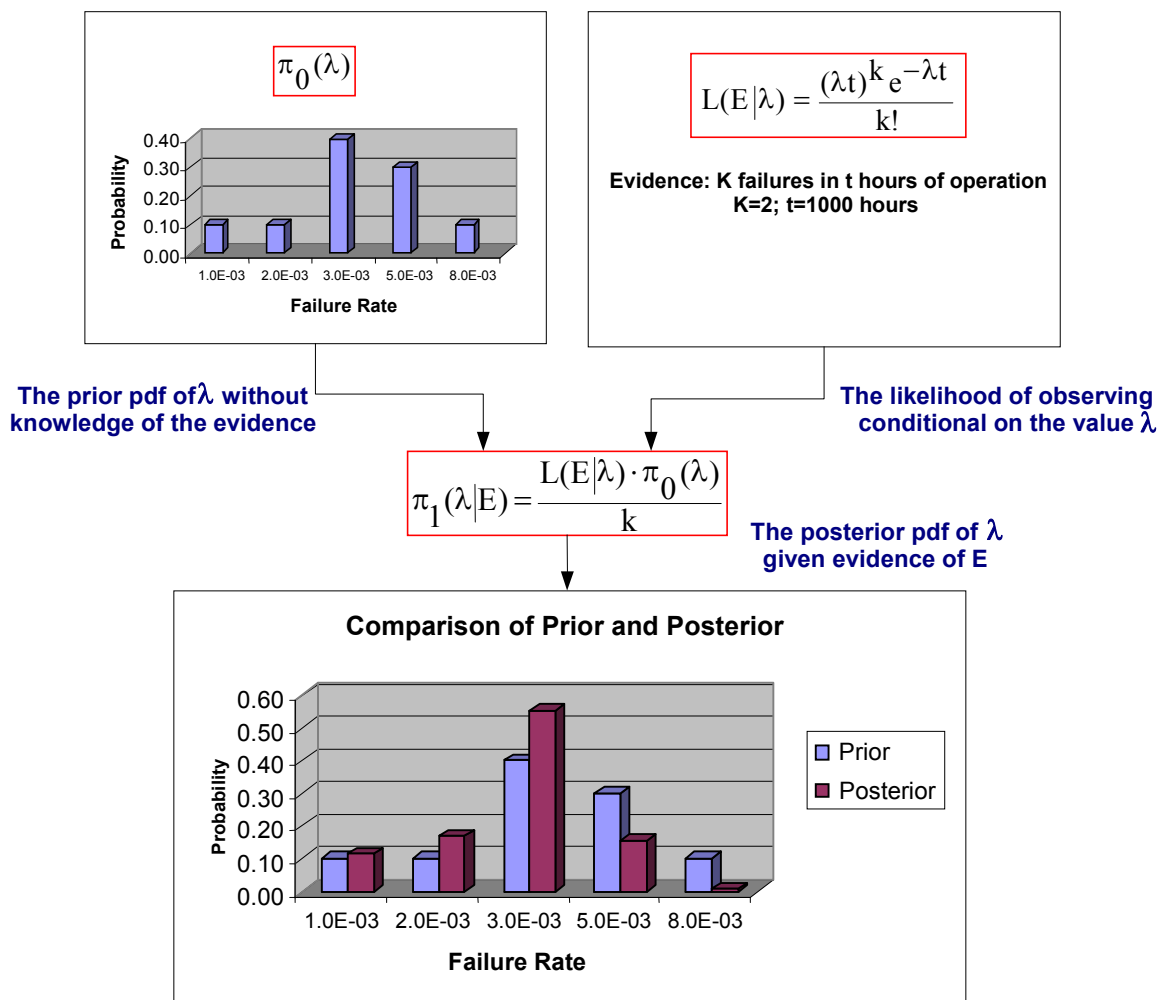


Figure 3-11. Application of Bayes' Theorem.

If there is relatively little uncertainty in the prior, corresponding to a significant body of supporting evidence, then more new evidence will be needed to shift the characteristics of the posterior away from the prior. The figure shows an example in which the prior distribution of a particular failure rate is highest at $3E-3$ per hour, almost as high at $5E-3$ per hour, and significantly lower for other values of the failure rate. The new evidence in that example is two failures in 1,000 hours; this corresponds to a maximum likelihood estimate of $2E-3$, which is lower than the apparent peak in the prior distribution. Correspondingly, we see that in the posterior, the probability of the lower-frequency bins is enhanced, and the probability of bins higher than $3E-3$ is reduced. The bin at $8E-3$ is reduced very significantly, because the new evidence is inconsistent with that bin; at $8E-3$, in 1000 hours of operation, the expected number of failures would be 8 rather than 2, and it is unlikely that a discrepancy of this magnitude is a statistical fluke. In essence, the weight of the bins in the prior distribution shifts toward the evidence.

Many decision processes require that uncertainty be treated explicitly. In the simple example discussed here, significant insights are realizable without it, but this is not universal. First, some decisions depend on more than just the mean value of the risk metric. Second, even when mean values are the desired output, it is formally necessary to derive them from valid underlying distributions of basic event probabilities. Moreover, as noted previously, complex risk expressions may contain terms that are non-linear in certain parameters, and the mean values of such terms are greater than the products of the corresponding powers of the parameter mean values. For all these reasons, it is necessary at least to consider the treatment of uncertainty in evaluating PRA outputs and in planning the work.

3.3.6 Formulation and Quantification of the Integrated Scenario Model

Once scenarios have been represented in terms of sets of pivotal events that are appropriately conditioned on what is occurring in each scenario, and pivotal events are represented in terms of basic events, it is possible to develop a representation of scenarios in terms of basic events. It is also possible to quantify this representation to determine its probability (or frequency, depending on the application). Indeed, all scenarios leading to a given outcome can be combined, leading to a quantifiable representation in terms of basic events of the occurrence of the outcomes of specific interest.

Table 3-1, presenting the scenarios and MCSs for the simple example, is an especially simple case of this. All MCSs are shown. It is easy to verify the “total” quoted by summing the MCS probabilities estimated as the product of the mean basic event probabilities. In many practical problems, the scenarios contributing to a given consequence category are so numerous and complex that the result is essentially unsurveyable in this form. It is normal practice to view PRA results making use of certain sensitivity coefficients called “importance measures.” These measures represent a level of detail somewhere between the hopelessly complex detail of the MCS representation and the complete absence of detail in the presentation of the top-level risk metric.

For reasons discussed above, it is necessary to address uncertainty in the value of the risk metric. This is done as indicated in Figure 3-12. This figure shows the risk expression R as a function of all of the basic event probabilities. The “rare-event approximation” to the functional form of R is obtained by interpreting the MCS expression as an algebraic quantity; but in general, the probability of the top event is overestimated by this approximation, and in many cases, use of a more complex form is warranted. Whichever approach is used, the probability distribution of the risk metric is determined by sampling as discussed above.

The mean and the percentiles of the distribution of the risk metric in the simple example are indicated on Figure 3-12. (The mean value is the average, or “expected,” value. The mth percentile value is the value below which m% of the cumulative probability lies. Since the 95th percentile is 3.74E-4, for example, we are 95% sure that the value lies at or below 3.74E-4.) This is predicated on the assumption that the model is valid. In many decision contexts, the mean value of the distribution will be used directly. In other decision contexts, other properties of the distribution may receive scrutiny. For example, we might be willing to accept a 1E-4 probability of loss of vehicle, but reluctant to accept a significantly higher value; we might therefore wish to reduce the uncertainty. It is possible to identify the scenarios and the constituent basic event probabilities that most strongly influence the right-hand portion of the distribution (corresponding to high top event probability), and this set of events may be different from the set of events that most strongly influence the mean value (although usually those that drive the high-probability end also strongly affect the mean).

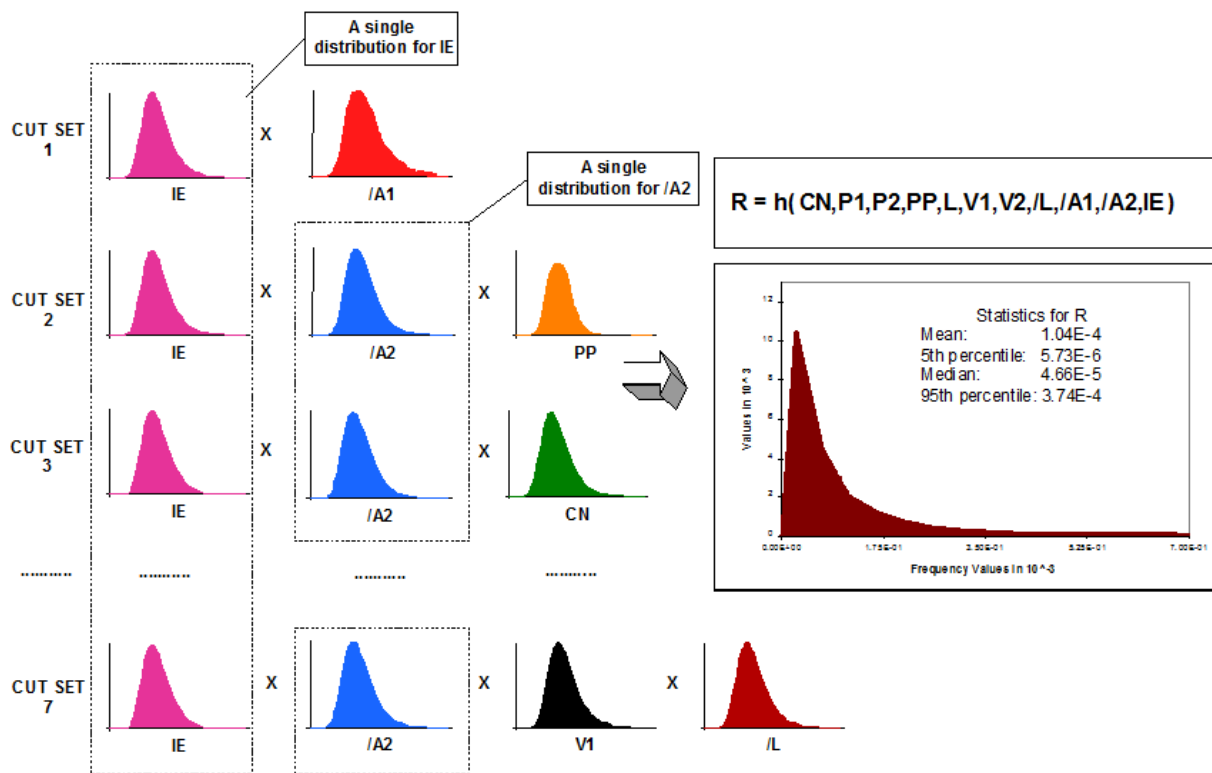


Figure 3-12. Propagation of Epistemic Uncertainties for the Example Problem.

If we simply insert these mean values into an approximate expression for top event probability, we obtain 1.02E-4. This is not the mean value of the top event probability, because while the basic events are independent, their probability values are correlated (identical sensors, identical valves). In the present case, the uncertainties were propagated accounting for these correlations.

In this example, according to the distribution shown in Figure 3-12, there is some chance that the top event probability is nearly four times higher than the mean value (based upon the 95th percentile). In some cases, the uncertainty will be even greater. The magnitude of the uncertainty needs to be considered in decisions regarding whether to accept a given situation.

3.3.7 Overview of PRA Task Flow

The preceding discussion essentially defines certain elements to be found in a PRA. The actual task flow is approximated in Figure 3-13. A task plan for the actual conduct of a PRA could be loosely based on this figure, although of course additional detail would need to be furnished for individual tasks.

As discussed in Chapter 2, PRA is performed to support risk management. The process therefore begins, as does RIDM generally, with a formulation of the objectives. This is logically necessary to inform the specification of the consequence categories to be addressed in scenario development, possibly to inform the scope of the assessment, and also to inform the specification of the frequency cutoffs that serve to bound the analysis.

After system familiarization, identification of IEs can begin, and other scenario modeling tasks can be undertaken as implied in Figure 3-13. Feedback loops are implicit in this figure. Also implicit in this figure is a possible need to evaluate the phenomenology of certain scenarios. Partly because analysis of mission success criteria can be expensive, it is easy in a logic-model-driven effort to shortchange the evaluation of mission success criteria. This is logically part of “structuring scenarios,” which includes ESD and ET development.

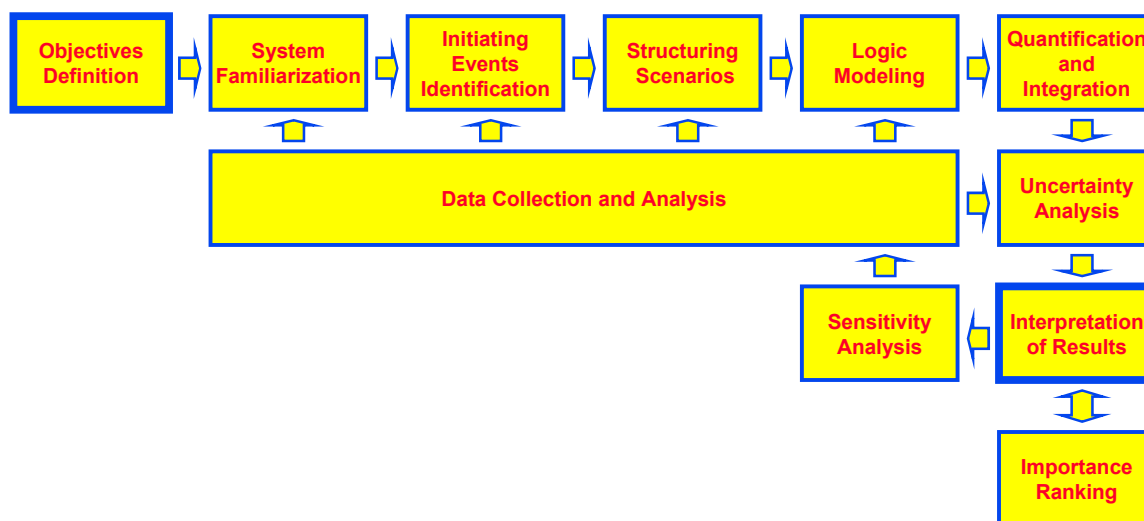


Figure 3-13. A Typical PRA Task Flow.

It is significant in Figure 3-13 that the “data analysis” block spans much of the figure and appears in iterative loops. This block influences, and is influenced by, many of the other blocks. The blocks that identify scenarios specify events whose probabilities need to be determined. Once initial estimates are obtained for probabilities, preliminary quantification may determine that some of the parameters need additional refinement.

Previous comments regarding the need for methodical development of the scenario model, and the need for a comprehensive scenario set, are reflected in this diagram. The entire top row of blocks is associated with formulation of the scenario model. Even “quantification” feeds back to “logic modeling” through “uncertainty analysis,” “interpretation of results,” “sensitivity analysis,” and “data collection and analysis.” In other words, scenario modeling is not generally accomplished in a single pass.

Risk analysis is necessarily a self-focusing activity. Scenarios can be postulated endlessly (extremely unlikely events can be postulated, basic events can be subdivided, etc.), but

resources are finite. An important aspect of risk analysis is to sort out patently insignificant contributors and avoid expenditure of effort in modeling them. The guideline for discarding scenarios is to be based on “risk significance” as defined by the decision objectives. This is part of what is going on in the feedback loops appearing in Figure 3-13. In the interest of efficient use of resources, some risk analyses are conducted as phased analyses, with a first-pass analysis culminating in a rough quantification presented in order to decide which scenarios deserve more careful modeling. It is strongly emphasized that prioritization of analysis based on risk significance has been found to lead to very different priorities than design-basis-oriented thought processes.

It is rare for the application of a PRA to be limited to citation of the expected accident frequency. Usually, more practical and robust outputs are desired. “Importance measures,” to be discussed at length later on (see Section 13.3), are a key part of “sensitivity analysis.” Importance measures not only serve as key aids in debugging the model, but also provide useful insight into the model results after the model is considered to be final. Some applications of risk models are based more closely on the relative risk significance of certain groups of scenarios than on the overall risk metric itself. For example, appearance of certain basic events in scenarios contributing a significant fraction of accident frequency may signal a vulnerability that needs to be addressed, possibly through a change in design or procedures. This might be a worthwhile (cost-effective) improvement even if the overall accident frequency appeared satisfactory without the fix.

3.4 Summary

3.4.1 Current State of Practice

For purposes of resource allocation, safety performance relative to goals and thresholds, and many other kinds of decisions, a comprehensive risk model is necessary. In situations characterized by physical complexity and high stakes, adequate decision support is not obtainable from assessment of individual system reliability metrics outside the context of a risk model. Without a good risk model, relatively unimportant issues may receive too much attention, and relatively important issues may go unidentified.

The need for completeness in a risk model implies a significant effort in development of the scenario set. This effort is justified by the stakes associated with the decisions driving the risk assessment. A corollary requirement is a need for significant project quality assurance (QA). Much of the methodology presented in this Guide has evolved over many years to promote completeness, to support peer review of the model, and to foster communication of the modeling results to end users and outsiders.

Although too simple to illustrate the real value of the highly systematic methods discussed in this Guide, even this simple example shows the need for completeness in the scenario set. A traditional system-level failure evaluation might have concluded that the engineered isolation function reduced the risk from leaks to an acceptable level; the risk analysis indicated that the potential for leaks that cannot be isolated dominates the risk, and the decision maker needs to consider the value of this latter probability—including uncertainty in that value—in deciding whether further prevention measures are necessary.

If it is decided that prevention measures are necessary, the PRA results direct the decision-maker to areas where expenditure of resources in design improvements might be fruitful. Again, in order for this kind of resource allocation to be supported appropriately, the scenario set has to be complete, and the quantification needs to be good enough to support the decisions being made.

Because of the stakes involved in the decisions, the complexity of typical models, and the potentially substantial investment in the analysis itself, it is frequently appropriate to conduct peer reviews of the analysis, even as it proceeds. One feature of the methods mentioned in this section and discussed at greater length later is that they generate intermediate products that support this kind of review.

3.4.2 Prospects for Future Development

In the introduction to this section, it was remarked that the strengths and weaknesses of the tools that have evolved within the commercial nuclear power application are not necessarily optimal for NASA. One area occasioning some differences is that of identification of IEs. In commercial reactors, IEs at full power are by definition those events that should generate a shutdown signal; therefore, they are extensively studied as part of the design process, and have the property of leading to upsets in very well-defined process variables. Systematic methods for identification are correspondingly well-developed. In facilities of other types, and arguably for certain NASA applications, the identification of IEs needs to go farther afield than for commercial nuclear plants.

Another area worthy of comment is the quantification of reliability and availability metrics. In commercial nuclear applications, relatively little effort is invested in time-dependent quantification of expressions; “point” values are used for basic event probabilities independently of possible correlation (due to dynamic effects) between basic event probabilities. In commercial nuclear power applications, this is arguably acceptable in many contexts, because the point of the analysis is to distinguish 1E-3 scenarios from 1E-6 scenarios, and low precision will suffice. In other applications, arguably including certain NASA applications, the actual reliability of certain systems is of some interest, and better numerical evaluations of failure probability are warranted.

Recent years have seen increasing application of simulation in risk analysis. In the presentation of the example earlier in this section, simulation was described as a way of quantifying pivotal event probabilities. This is a worthwhile start, but it is desirable to push simulation technology farther. The event tree structure itself may impose important simplifications on the scenario model that a simulation-based treatment would not require. For example, specification of a given pivotal event in an event tree may entail restrictive (typically bounding) assumptions about event timing, but simulation of time histories can address event timing without such restrictive assumptions.

3.5 References

- 3-1 *Reactor Safety Study*, Report WASH-1400, Nuclear Regulatory Commission, 1975.
- 3-2 S. Kaplan and B.J. Garrick, “On the Quantitative Definition of Risk,” *Risk Analysis*, 1, 11-37, 1981.
- 3-3 *NASA System Safety Handbook: Volume 1*, NASA/SP-2010-580, December 2011.
- 3-4 NASA NPR 8715.3C: *NASA General Safety Program Requirements (w/Change 4, dated 7/20/09)*.

4. Scenario Development

According to Section 2.1, risk is usefully conceived as a set of triplets involving:

- Scenarios;
- Associated frequencies; and
- Associated consequences.

Clearly, developing scenarios is fundamental to the concept of risk and its evaluation. Moreover, application of Boolean algebra to a scenario generates the mathematical expression needed to quantify its frequency. The mathematical expression for the frequency of a specific scenario, $A_{j,k}$, is:

$$A_{j,k} \equiv \Lambda(ES_{j,k}) = \lambda_j \Pr(ES_{j,k} | IE_j) \quad (4-1)$$

Where:

- λ_j denotes the frequency of the j th initiating event (IE) modeled in the PRA; and
- $\Pr(ES_{j,k} | IE_j)$ symbolizes the conditional probability for the end state of event sequence, k , in the event tree initiated by IE_j , given that IE_j has occurred.

Fundamentally then, scenario development begins with the entity whose risk is being assessed (e.g., a ground-based facility, launch vehicle, orbiting asset, or scientific instrument) and concludes with a mathematical model resembling Equation (4-1). Quantification of this model (the subject of Section 3.3.6) provides the frequency needed by the risk triplet.

4.1 System Familiarization

System familiarization is a prerequisite for model development. The task of system familiarization is not trivial. Understanding every nuance of a system can be a time-consuming chore. Since all models are approximations, not every nuance of the system will be incorporated into the risk model. Nevertheless, the PRA team^a must have sufficient familiarity with the system to derive a rationale for any aspect of system behavior that their model ignores.

Resources available to facilitate system familiarization may include:

- Design manuals;
- Design blueprints and technical requirement documentations;
- Operations and maintenance manuals;
- Operations and maintenance personnel;
- Operations and maintenance logs;

^a "PRA team" is used in this document to refer to the people and organizations who are involved in the development of the PRA model, including domain experts, risk analysts, systems engineers, etc.

- The technical staff (including system and design engineers);
- The crew (if applicable); and
- Visual inspection, whenever possible.

Of course, the amount of detail available is directly related to the system maturity. During conceptual design the amount of detail may be quite sparse. Here, it is necessary for the PRA team to elicit system familiarization information from the technical staff. During final design, detailed system descriptions and some manuals may be available. For an operating system (e.g., an established ground-based facility), operations and maintenance personnel and logs afford excellent insights into how the system actually behaves.

Section 1.1.1 warns that much of the time, pivotal events are not independent of each other. Although Chapter 10 explains the mathematical aspects of dependency modeling, a thorough understanding of dependencies must initially be obtained through system familiarization. A useful technique, but not the only technique for documenting system dependencies involves a matrix. Other techniques, or combinations thereof, include function-to-system dependency matrix, mission event timelines, functional block diagrams, interface diagrams, engineering drawings, etc.

The dependency matrix is usually developed during the system analysis portion of a PRA. It is an explicit list that describes how each system functionally supports other systems. This is useful in developing scenarios because it allows the analyst to see how failures in one system can cause failures in other systems. Dependency matrices facilitate event sequence development by ensuring that failures in one pivotal event are correctly modeled in subsequent events.

The dependency matrix concept can be illustrated by considering a simple, habitable space vehicle capable of re-entry into the Earth's atmosphere, such as a crew return vehicle. If the vehicle systems are:

- Propulsion (PROP);
- Thermal Protection System (TPS);
- Reaction Control System (RCS);
- Flight Control and Actuation System (FCAS);
- Electrical power generation and distribution (ELEC);
- Environmental Control and Life Support System (ECLSS);
- Vehicle Management System (VMS);
- Landing gear and braking (GR/BR);
- Communication (COMM); and
- Structure (STRUCT);

then Table 4-1 is a sample dependency matrix for a crew return vehicle.

The matrix is read column by column, where the system listed at the top of the column is supported by the systems marked in the rows beneath with a "X." For example, the FCAS receives support from:

- ELEC;

- VMS; and
- STRUCT.

Table 4-1 is only an illustration, but a fully developed dependency matrix could contain more information than merely a "X." For example, endnotes appended to the matrix could describe the types of support functions provided. Further, dependencies could be a function of the mission phase and may be noted in the matrix. Developing a dependency matrix allows all the analysts to be consistent in their modeling and to fully understand the system dependencies.

Table 4-1. Sample Dependency Matrix.

This → Supported by ↓	PROP	TPS	RCS	FCAS	ELEC	ECLSS	VMS	GR/BR	COMM	STRUCT
PROP			X			X				
TPS	X		X		X	X		X		X
RCS	X									
FCAS										
ELEC	X		X	X		X	X	X	X	
ECLSS	X		X		X		X		X	
VMS	X		X	X	X	X		X	X	
GR/BR										
COMM							X			
STRUCT	X	X		X				X	X	

4.2 Success Criteria

Success criteria are needed to define satisfactory performance. Logically, of course, if performance is unsatisfactory, then the result is failure.

There are two types of success criteria, for:

1. Missions; and
2. Systems.

Relative to their content, the criteria are analogous. The essential difference is that the first set applies to the overall mission (e.g., under what conditions does a crew return vehicle function satisfactorily), while the second set addresses individual system performance (e.g., performance of the RCS or FCAS in Table 4-1). They are the subjects of Sections 4.2.1 and 4.2.2, respectively.

4.2.1 Mission Success Criteria

Mission success criteria are necessary to define risk assessment end states (i.e., ES_j in Equation (4-1)). Mission success criteria as a minimum must:

- Define what the entity being evaluated is expected to accomplish in order to achieve success; and
- Provide temporal or phase-dependent requirements.

Defining what the entity being evaluated is expected to accomplish is essential for ascertaining whether a scenario results in success or failure. This facet of the criteria permits the analyst to develop logic expressions or rules for determining what combinations of IEs and

pivotal events prevent the entity being evaluated from performing satisfactorily. Temporal or phase-dependent requirements:

1. Allow the PRA to differentiate between mission phases (e.g., the GR/BR is not needed until a crew return vehicle is ready to land); and
2. Define operating durations.

This second aspect is important because probabilities are time dependent (recall Figure 3-10).

Sometimes, multiple mission success criteria are imposed. For example, a science mission may contain multiple instruments to collect different types of data. If one instrument fails, the data furnished by the remaining instruments will still have some scientific value. Therefore, while successful operation of all instruments may correspond to mission success, even the acquisition of limited data may satisfy minimum mission requirements. Thus, possible end states in this situation are:

- Complete mission success;
- Limited mission success; and
- Mission failure.

A crucial requisite for mission success criteria is that they must be mutually exclusive in a logical context. Generally, the genesis of mission success criteria coincides with conceptualization of the mission.

The reader is encouraged to consult the NASA System Safety Handbook [4-2], section 3.1.1, for additional information on probabilistic requirements for mission success.

4.2.2 System Success Criteria

The principal difference between system success criteria and mission success criteria is that system success criteria apply only to individual systems. However, mission and system success criteria are not completely independent. For example, mission success criteria impose operating requirements on the systems needed to successfully perform a particular mission phase, and the duration of that phase determines the system operating time.

System success criteria should include a temporal component and a statement of system redundancy (e.g., at least one of three strings should start on demand and operate for 20 minutes). Top event FT logic is established from the Boolean complement of the success criteria (e.g., all three strings must fail to start on demand or fail to operate for 20 minutes). Basically, then, mission success criteria are used to determine event sequence end states, while system success criteria pertain to FT top events and logic.

Defining system success criteria should occur during the system analysis portion of the study. Some examples of system success criteria are:

- At least one of the two electric power generation strings needs to provide between 22 and 26 VDC for the duration of the mission;
- The Vehicle Management System needs to have at least one of its four mission computers operational at all times; and
- The Inertial Navigation System needs to maintain at least one out of three boxes operational during the ascent and descent phases.

Success criteria should be clearly defined. All assumptions and supporting information used to define the success criteria should be listed in the documentation (i.e., what is considered to constitute system success needs to be explicitly stated).

4.3 Developing a Risk Model

The risk model is basically the PRA model developed to represent the entity being assessed. Traditionally, scenarios are developed through a combination of ETs and FTs. Although it is theoretically possible to develop a risk model using only FTs or ETs, such a theoretical exercise would be inordinately difficult except for simple cases. Since the level of effort that can be devoted to risk assessments, like all other applied technical disciplines, is constrained by programmatic resources, in practice ETs are typically used to portray progressions of events over time (e.g., the various phases of a mission), while FTs best represent the logic corresponding to failure of complex systems. This is illustrated in Figure 4-1.

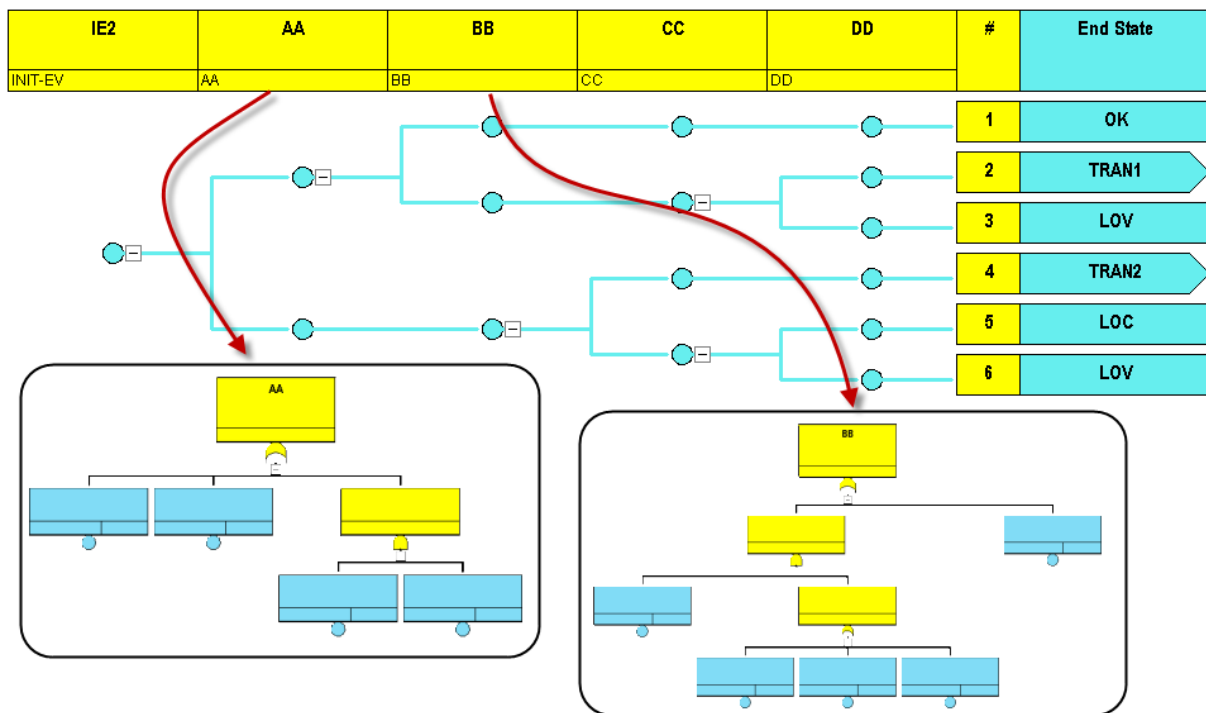


Figure 4-1. Event Tree/Fault Tree Linking.

The process of combining ETs with FTs is known as linking. The ET in Figure 4-1 contains an IE, IE2, and four pivotal events:

1. AA;
2. BB;
3. CC; and
4. DD.

Three end states are identified:

1. OK (i.e., mission success);

2. LOC (signifying a loss of the crew); and
3. LOV (denoting loss of vehicle).

Of course, the assignment of these end states to particular event sequences is predicated upon the mission success criteria addressed in Section 4.2.1.

Figure 4-1 also has two transition states:

1. TRAN1; and
2. TRAN2.

End states terminate an event sequence because the outcome of the scenario relative to mission success criteria is known. However, if the event sequence has not progressed far enough to ascertain which end state results, a transition state transfers the scenario to another ET where additional modeling is performed. Ultimately, every event sequence is developed sufficiently to determine its end state, and at that point the scenario model stops.

The FTs illustrated in Figure 4-1 are linked to pivotal events AA and BB. This is a standard PRA technique where the top event in the FT corresponds to failure of a specific pivotal event. However, it is not necessary to develop an FT for every pivotal event. If applicable probabilistic data are available from similar missions or testing, these data can be assigned directly to the pivotal events without further modeling. In this situation the pivotal events behave as basic events in the PRA model.

Once the ETs and FTs are developed and linked, the evaluation of the scenario frequency can commence. The process begins by assigning exclusive names to all unique basic events in the model. The only real constraint on the basic event naming convention adopted in a PRA is that it must be compatible with all software that is used in the assessment. Typically, this constraint will limit only the number of characters that comprise a basic event name. Besides software compatibility, the basic event naming should be informative (i.e., it should convey information about the nature of the event being modeled). Types of information that could be encoded in a basic event name are the:

- Hardware item being modeled (e.g., a valve or thruster);
- Failure mode (e.g., failure to operate);
- Mission phase; and
- System to which the hardware item belongs.

Generally, the basic event names have the form, A...A-B...B-C...C-...-Z...Z, where, for example:

- A...A might represent the hardware item being modeled;
- B...B could signify the failure mode;
- C...C may possibly symbolize the mission phase; while
- The last set of characters may denote the system.

Each character set (e.g., the failure mode) is separated from the others by a delimiter (e.g., a dash).

By applying Boolean algebra to the risk model, a mathematical (Boolean) expression for each scenario is derived.

Relative to Figure 4-1, the first event sequence terminates with end state, OK. The Boolean equation for this event sequence is:

$$OK_{2,1} = IE2 \cap AA \cap BB \quad (4-2)$$

where it is inferred that the IE in Figure 4-1 is the second IE modeled in the PRA.

The Boolean equations for the remaining five scenarios in Figure 4-1 are:

$$TRAN1_{2,2} = IE2 \cap AA \cap \overline{BB} \cap DD \quad (4-3)$$

$$LOV_{2,3} = IE2 \cap AA \cap \overline{BB} \cap \overline{DD} \quad (4-4)$$

$$TRAN2_{2,4} = IE2 \cap \overline{AA} \cap CC \quad (4-5)$$

$$LOC_{2,5} = IE2 \cap \overline{AA} \cap \overline{CC} \cap DD \quad (4-6)$$

and

$$LOV_{2,6} = IE2 \cap \overline{AA} \cap \overline{CC} \cap \overline{DD} \quad (4-7)$$

With respect to Equation (4-2), the frequency of the first event sequence is

$$\lambda(OK_{2,1}) = \lambda(IE2 \cap AA \cap BB) = \lambda_2 \Pr(AA \cap BB | IE2) \quad (4-8)$$

Similar equations can readily be derived for the other Figure 4-1 scenarios.

Equation (4-8) does not include the basic events from the linked FTs. However, these portions of the logic model can be incorporated into the frequency equation by directly substituting the Boolean expressions for the FT top events and performing any appropriate simplification. For any sizeable PRA, of course, this exercise in Boolean algebra becomes tedious if performed manually, which is one incentive for using PRA software to evaluate risk.

Three aspects of Figure 4-1 merit further explanation:

1. IE development;
2. accident progression (i.e., ET construction); and
3. FT modeling.

These are the topics of Sections 4.3.1 through 4.3.3, respectively.

4.3.1 IE Development

One of the first modeling issues that must be resolved in performing a PRA is the identification of accident scenarios. This modeling of "what can go wrong?" follows the systematic identification of accident initial causes, called initiating events, grouping of individual causes into like categories, and subsequent quantification of its likelihood. In general, accident scenarios are the result of an upset condition (the initiating event) and the consequential outcome following the upset condition. Note that initiating events may lead directly to undesirable outcomes or may require additional system/component failures prior to reaching a negative outcome.

Since the number of different initiating events is, in theory, very large (e.g., a rocket may fail at $t=1.1$ sec, at $t=1.2$ sec, at $t=1.3$ sec, etc.), individual types of initiating events will be grouped into similar categories. For example, in the case of a rocket failing, rather than have many different initiating events at multiple times, we may combine these and only consider the frequency of a rocket failing from $t=0$ sec to $t=10$ sec.

The depiction of initiators comes from a variety of techniques. Precursor events may directly or indirectly indicate the types and frequencies of applicable upsets. Conversely, analysts may deduce initiating events through techniques such as failure modes and effects analysis and master logic diagrams (MLD). For example, Figure 4-2 shows an example of a MLD that might be used to identify initiating events (not exhaustive) related to upsets caused by kinetic energy. A deductive method such as fault tree can be useful for determining initiating events in order to find situations where localized component faults can cause an upset condition.

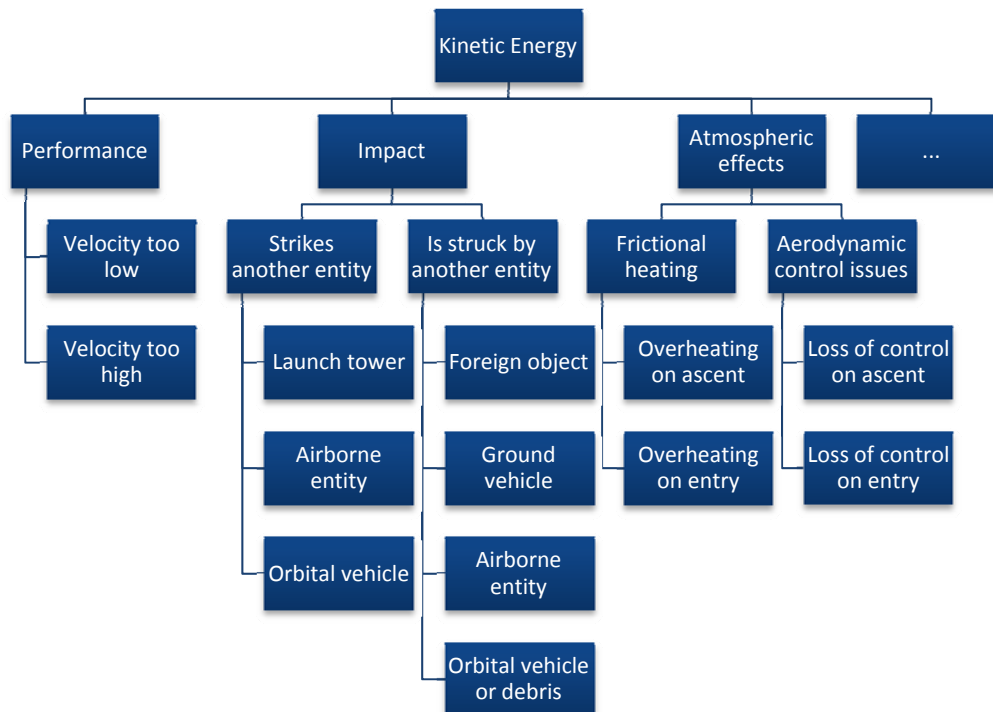


Figure 4-2. Notional Master Logic Diagram Related to Candidate Initiating Events Caused by Kinetic Energy.

For the system or mission being analyzed, the IEs that are evaluated are those that potentially may result in the undesired outcome of interest (i.e., failure to meet one of the applicable risk-informed decision performance measures). These IEs are situations that arise from an operational deviation or departure from the desired system operation.

Initiating Event (IE)

A departure from a desired operational envelope to a system state where a control response is required either by human or machine intervention.

Previously, we described the concept of a “scenario” showing how adverse consequences leading to a spectrum of consequences might occur when IEs occur, system control responses

fail, and the consequence severity is not limited as well (Figure 4-3). In this scenario representation, hazards^a may impinge on the system or mission in several ways:

- They may provide enabling events (i.e., conditions that provide the opportunity to challenge system safety, potentially leading to an accident);
- They may affect the occurrence of IEs;
- They may challenge system controls (safety functions);
- They may defeat mitigating systems;
- They may fail to ameliorate the consequences of mitigating system failures.

Systems are safe because IEs do not occur very often – they seldom leave the desired operational state. Further, even if the system *does* leave the desired state, control and mitigation systems fail infrequently. Thus, accidents and experiencing the associated consequences is unlikely. However, it is the task of the PRA to represent these scenarios by decomposing the sequence of events, starting with the IE, through system failures (or success) to the end states of interest.

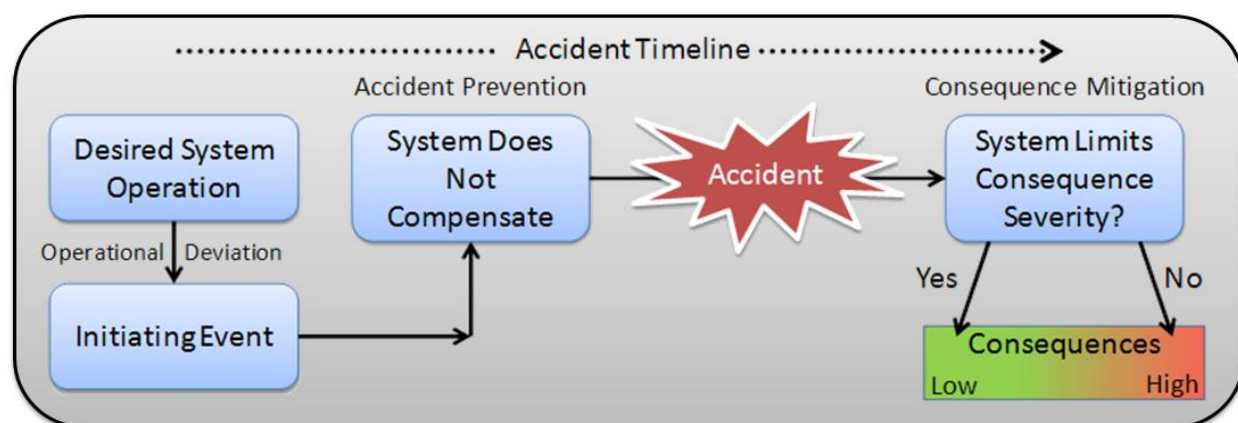


Figure 4-3. The Elements of an Accident Scenario.

Quantification of an IE generally takes place via a Bayesian approach wherein operational data are evaluated to determine the initiator frequency, including the uncertainty on the frequency (this approach is described in Chapters 5 and 6). In some cases, little data may exist – in these situations the analysis typically relies on models (perhaps physics-based models) or expert judgment to provide the frequency.

Two basic approaches to IE development have been typically used in aerospace PRAs. The first approach develops a set of IEs using techniques such as the MLD described earlier. The second approach is to replace the IE with a single “mission entry point,” and then model the

^a Here “hazard” can be defined as a condition that is, or potentiates, a deviation in system operation. Existence of a hazard implies that controls should be considered (if affordable).

entire mission using FTs linked to the ET structure. This single IE in the PRA is the point in the mission when risk becomes appreciable – typically denoted as “launch.” When this second approach is used, it is important that the analyst ensures completeness of the accident scenarios since the actual upset conditions may be buried among the FT and ET models.

In theory, having an IE of "launch" is fine, as long as the PRA analyst then models holistically the hazards and enabling conditions that lead to IEs in the other parts of the model (i.e., the phased top events after launch). However, by decomposing the model into discrete phases, the creation of fault trees representing the system components and interactions across phases becomes a modeling challenge. With this modeling approach, the idea of a scenario gets discarded since failures become compartmentalized due to the nature of the phased model. But, this is not strictly due to the use of "launch" as an initiator. Instead, this is the result of not coupling directly models to hazards and failure causes that represent IEs as part of an accident scenario.

In practice, ETs are typically used to portray progressions of sequential events over time (e.g., the various phases of a mission), while FTs best represent the logic corresponding to failure of complex systems. Therefore, when repair (or recovery) is precluded, some analysts prefer a large ET model for the risk assessment, while large FT models are preferred by some analysts for situations where maintenance (or recovery) is performed.

Large FT models are most often applied to repairable complex systems such as ground-based facilities. Because maintenance is routinely performed, each time a system or critical component fails, it is repaired and the facility resumes normal operation. Attempting to assess risk in these circumstances with a single ET results in a complex model due to the potential numerous changes in system states. However, since the facility will enter a time independent availability state, a simpler approach is to use a static logic model to:

- Postulate that the facility is operating normally;
- Identify IEs capable of perturbing this normal operating state;
- Develop relatively simple ETs for each IE;
- Construct FTs that link to the pivotal events; and
- Quantify the risk.

Although this modeling process (use of large FT for repairable, complex systems) is more involved than the use of launch as an initiating event (as used in many robotic missions), the difference in effort results from the greater complexity inherent in ground-based facilities (relative to robotic spacecraft, where a single failure may not be repairable, and simply leads to termination of the mission).

Since the large FT methodology is most conducive to modeling systems with a time independent availability, it lacks the capacity to directly assess when failure end states occur. No comparable limitations apply to the large ET modeling technique. Nevertheless, it must be reiterated that large ETs with a single entry point has practical limitations when applied to complex systems or facilities that have repair capabilities.

4.3.2 Accident Progression

Accident progression can be modeled using an event sequence diagram (ESD) or its derivative, an ET. Both are inductive logic models used in PRAs to provide organized displays of sequences of system failures or successes, and human errors or successes, that can lead to specific end states. An ESD is inductive because it starts with the premise that some IE has

occurred and then maps out what could occur in the future if systems (or humans) fail or succeed. The ESD identifies accident sequences (or pathways) leading to different end states. The accident sequences form part of the Boolean logic, which allows the systematic quantification of risk (e.g., Equation (4-1)).

A traditional accident progression analysis begins with an ESD, refines it, and then transforms it into an ET format. The advantage of this process is that the morphology of an ESD is less rigidly structured than an ET. Hence, ESDs permit the complex relationships among IEs and subsequent responses to be displayed more readily.

Typically, one ESD is developed for each IE. The objective is to illustrate all possible paths from the IE to the end states. An ESD is a success-oriented graphic in that it is developed by considering how human actions and system responses (including software) can prevent an accident or mitigate its severity.

An important attribute of an ESD is its ability to describe and document assumptions used in ETs. An ESD can be very detailed, depicting all sequences considered by the PRA analyst. When simplifying assumptions are used to facilitate ET construction or quantification, the ESD may furnish a basis for demonstrating why such assumptions are conservative, or probabilistically justified.

ESDs are the subject of Section 4.3.2.1.

Event Trees (Section 4.3.2.2) are quantitative graphics that display relationships among IEs and subsequent responses. Similar to ESDs, one ET is developed for each IE. The objective is to develop a tractable model for the important paths leading from the IE to the end states. This can be accomplished either by a single ET, or with linked ETs. ET logic may be simpler than the corresponding ESD. However, the ET sequences still form part of the Boolean logic, which allows the systematic quantification of risk. Generally, risk quantification is achieved by developing FT models for the pivotal events in an ET. This linking between an ET and FTs permits a Boolean equation to be derived for each event sequence. Event sequence quantification occurs when reliability data are used to numerically evaluate the corresponding Boolean equation [recall Equation (4-1)].

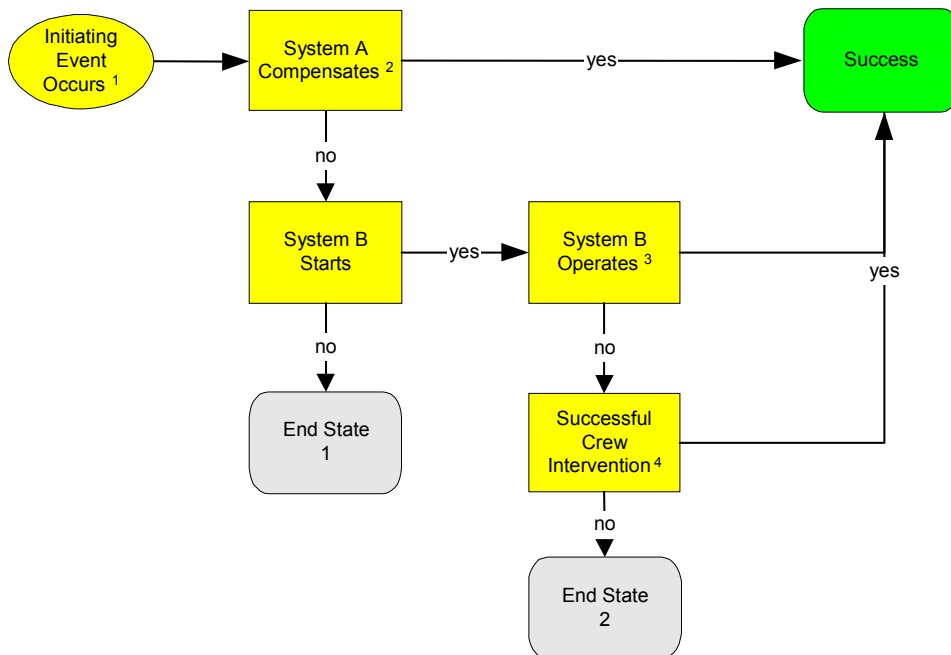
4.3.2.1 Event Sequence Diagrams

Figure 4-4 depicts a typical ESD and its symbols. The Figure 4-4 ESD begins with an IE that perturbs the entity being modeled from a stable state. Compensation for this perturbation is provided by System A. Typically, such a system is a normally operating control or protection system, which does not have to start in response to the IE. If System A compensates for the IE, a successful end state results.

System B can compensate for the IE if System A fails. System B is a standby system because it must start before it can compensate for the IE. According to Figure 4-4, a successful end state ensues if System B starts and operates satisfactorily.

Failure of System B to start on demand results in End State 1. If System B starts but does not operate properly, successful crew intervention can still prevent an accident. If the crew efforts are unsuccessful, End State 2 results. Examples of crew actions that could lead to a successful end state include:

- Restoring System A during the period that System B operates; or
- Manually compensating for the IE.



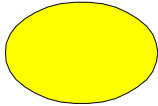
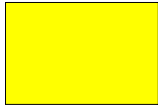
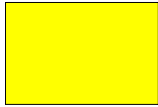
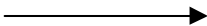
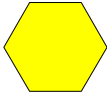

Legend	
1. Initiating event	
2. Anticipated response of System A.	
3. System B success criteria.	
4. Mitigation options to the crew, including procedures.	
Initiation Symbol—Marks the beginning	
Mitigation Block—Denotes system or actions capable of preventing an accident mitigating its severity.	
Aggravating Block—Denotes system human actions capable of increasing severity of accident.	
Arrow—Indicates the event from IE to end state.	
Connector—Used to connect ESD when the diagram size exceeds one page. unique designation (e.g., a letter or should be inserted into the connector so that the two ESD segments being can be identified.	
Termination Symbol—Marks the end	

Figure 4-4. Typical Event Sequence Diagram.

The use of two different end state designations in Figure 4-4 indicates that the severity of the accident depends upon the response of System B. If System B starts but does not operate properly, it may nevertheless partially compensate for the IE, resulting in less severe consequences to crew safety or mission success. The consequences of interest should be understood before ESD development commences.

Figure 4-4 includes a legend affording:

- A description of the IE;
- The anticipated response of System A to the IE;
- Criteria for the successful operation of System B; and
- Mitigation options available to the crew.

Including legends with an ESD is beneficial because it furnishes explanations directly with the diagram. However, in some situations the accompanying information can become quite voluminous (e.g., explaining mitigation procedures the crew will use in response to certain event sequences or IEs). In such circumstances, the detailed explanations should be included in a report appended to the ESD.

Figure 4-5 and Figure 4-6 illustrate the process of ESD development. Since an ESD is success oriented, the process begins by identifying the anticipated response to the IE. For this example, the anticipated response is for System A (which is normally operating) to compensate. If System A functions satisfactorily, the IE is mitigated and an accident is averted. This anticipated success path is developed first in the ESD, as Figure 4-5 indicates.

Failure of System A does not necessarily result in an accident. A standby system, System B, is available if System A fails. Hence, a second success path can be developed for the ESD by modeling the successful actuation and operation of System B. However, if System B fails to start on demand, End State 1 results. These additions to the initial ESD success path are depicted in Figure 4-6.

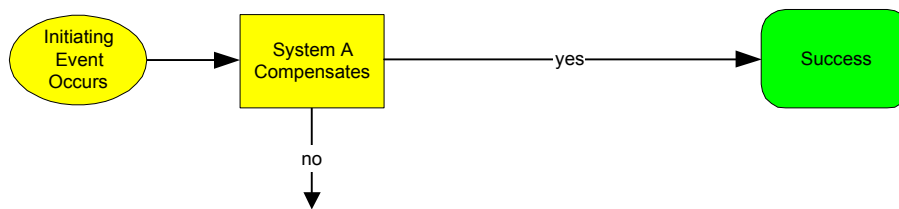


Figure 4-5. Event Sequence Diagram Development (step 1).

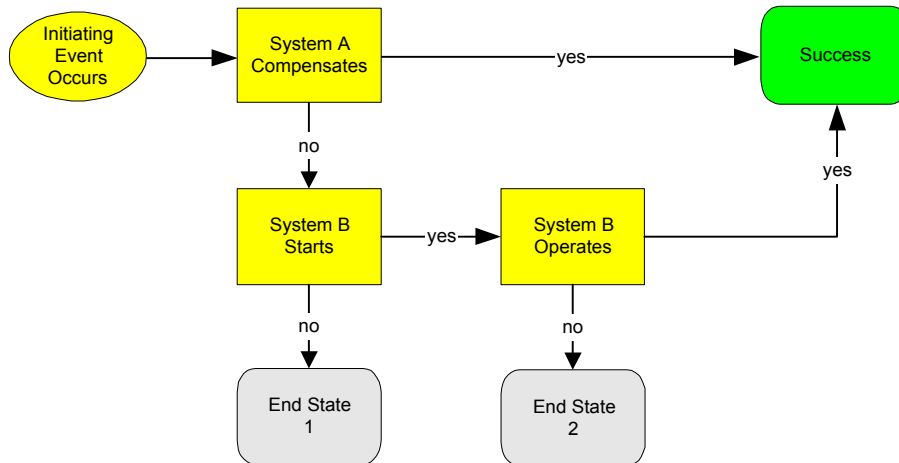


Figure 4-6. Typical Event Sequence Diagram Development (step 2).

Inability of System B to operate does not result in an undesirable end state if the crew intervenes successfully. If this human recovery action fails, the event sequence terminates with End State 2. Appending this final mitigation block to the ESD in Figure 4-6 and adding the legend results in Figure 4-4. This same basic process, i.e.,

- Beginning with the IE, model the anticipated response;
- Adding mitigation by backup systems or human actions for each failure that can occur during the anticipated response; and then
- Identifying the resulting end states for those event sequences where the backup systems and human actions fail

can be used to develop an ESD for any system or facility.

4.3.2.2 Event Trees

Figure 4-7 is the ET corresponding to the Figure 4-4 ESD. Comparing Figure 4-4 and Figure 4-7 discloses that the event sequences displayed in each are identical. This is because the accident progression is relatively simple. For more complicated accident scenarios, the detailed information incorporated into the ESD may be abridged during ET development.

Both ESDs and ETs are graphical representations of Boolean equations. This is an attribute they share with FTs. Let:

- I_E symbolize the set of elements capable of causing the IE in Figure 4-7 to occur;
- \bar{A} denote the set of events that prevent System A from compensating for I_E ;
- \bar{B}_S represent the set corresponding to failure of System B to start on demand;
- \bar{B}_O designate the set of elements capable of preventing System B from operating successfully; and
- \bar{R} signify the set of human errors that preclude successful crew intervention.

Then the Boolean expressions for Figure 4-4 and Figure 4-7 are listed in Table 4-2.

The Boolean expressions in Table 4-2 would be expanded by linking them to FT models for the pivotal events in Figure 4-7. Specifically, FT models would be developed for:

- \bar{A} ;
- \bar{B}_S ; and
- \bar{B}_O ;

(see Section 4.3.3). Recovery by successful crew intervention would be modeled using the human reliability analysis (HRA) techniques described in Chapter 7. Ultimately, by linking the ET to logic models from FTs and HRA, the expressions in Table 4-2 are expanded until they relate the event sequences directly to the basic events comprising the PRA model.

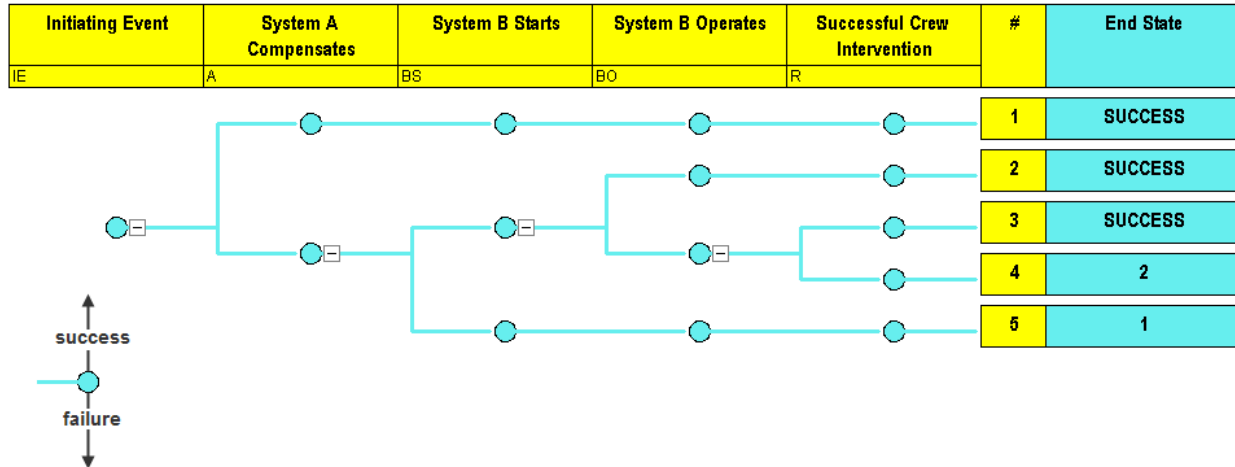


Figure 4-7. Event Tree Structure.

Table 4-2. Boolean Expressions for Figures 4-4 and 4-7

Sequence #	Boolean Expression
1	$I_E \cap A$
2	$I_E \cap \bar{A} \cap B_S \cap B_O$
3	$I_E \cap \bar{A} \cap B_S \cap \bar{B}_O \cap R$
4	$I_E \cap \bar{A} \cap B_S \cap \bar{B}_O \cap \bar{R}$
5	$I_E \cap \bar{A} \cap \bar{B}_S$

Figure 4-4 and Figure 4-7 are more representative of large FT models, where much of the detailed logic is embodied in the FTs.

ET linking is usually necessary when constructing large ET models. Conceptually, an event sequence that links to another ET can be considered as an IE for the second tree. This is illustrated in Figure 4-8. Table 4-3 lists the Boolean expressions for Figure 4-8. The pivotal events in Figure 4-8 will ultimately be linked to FTs or other models.

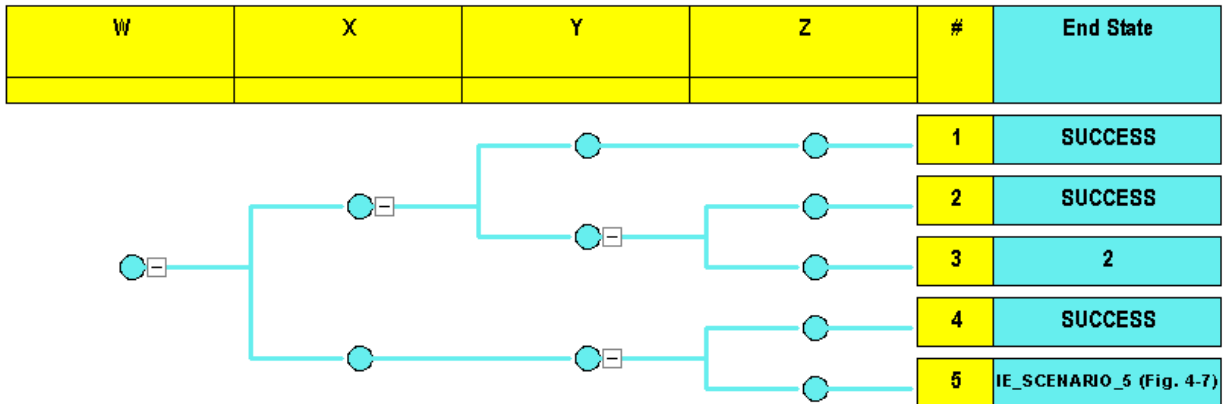


Figure 4-8. Event Tree Linking.

Table 4-3. Boolean Expressions for Figure 4-8.

Sequence #	Boolean Expression
1	$W \cap X \cap Y$
2	$W \cap X \cap \bar{Y} \cap Z$
3	$W \cap X \cap \bar{Y} \cap \bar{Z}$
4	$W \cap \bar{X} \cap Z$
5	$W \cap \bar{X} \cap \bar{Z}$

Notice that event sequence 5 in Figure 4-8 is linked to the Figure 4-7 ET. Let the notation, W5-IE_n, signify the event sequence involving the concatenation of sequence 5 in Figure 4-8 with the nth sequence in Figure 4-7. To determine the Boolean equation for sequence W5-IE1, let

$$I_E = W \cap \bar{X} \cap \bar{Z} \tag{4-9}$$

Then combining Equation (4-9) with the first entry in Table 4-2:

$$W5 - IE1 = W \cap \bar{X} \cap \bar{Z} \cap A \tag{4-10}$$

Accordingly, the linked event sequence involves IE, W, conjoined with:

- Failure of X;
- Failure of Z; and
- Compensation by System A.

Similarly:

$$W5 - IE3 = W \cap \bar{X} \cap \bar{Z} \cap \bar{A} \cap B_S \cap \bar{B}_O \cap R \tag{4-11}$$

$$W5 - IE4 = W \cap \bar{X} \cap \bar{Z} \cap \bar{A} \cap B_s \cap \bar{B}_o \cap \bar{R} \quad (4-12)$$

and

$$W5 - IE5 = W \cap \bar{X} \cap \bar{Z} \cap \bar{A} \cap \bar{B}_s \quad (4-13)$$

Moreover:

- Event sequences W5-IE1 through W5-IE3 result in success;
- Event sequence W5-IE4 leads to End State 2; while
- End State 1 results from event sequence W5-IE5.

Once Boolean equations for the linked event sequences are derived, their likelihood can be quantified and ultimately combined into end state probabilities.

4.3.3 Fault Tree Modeling

An FT is a deductive logic model whereby a system failure is postulated (called the top event) and reverse paths are developed to gradually link this consequence with all subsystems, components, software errors, or human actions (in order of decreasing generality) that can contribute to the top event, down to those whose basic probability of failure (or success) is known and can be directly used for quantification. Graphically, a FT at its simplest consists of blocks (e.g., rectangles or circles) containing descriptions of failure modes and binary logic gates (e.g., union or intersection) that logically link basic failures through intermediate level failures to the top event. The basic principles and procedures for fault tree construction and analysis are discussed in Reference [4-1].

Figure 4-9 depicts a typical FT structure and the symbols used.

FTs are constructed to define all significant failure combinations that lead to the top event—typically the failure of a particular system to function satisfactorily. Satisfactory performance is defined by success criteria, which are the subject of Section 4.2.2.

Ultimately, FTs are graphical representations of Boolean expressions. For the FT in Figure 4-9, the corresponding Boolean equation is:

$$T = E \cap C \cap D = (A \cup B) \cap C \cap D \quad (4-14)$$

where:

- T is the top event; and
- A through E are the basic and intermediate events in Figure 4-9.

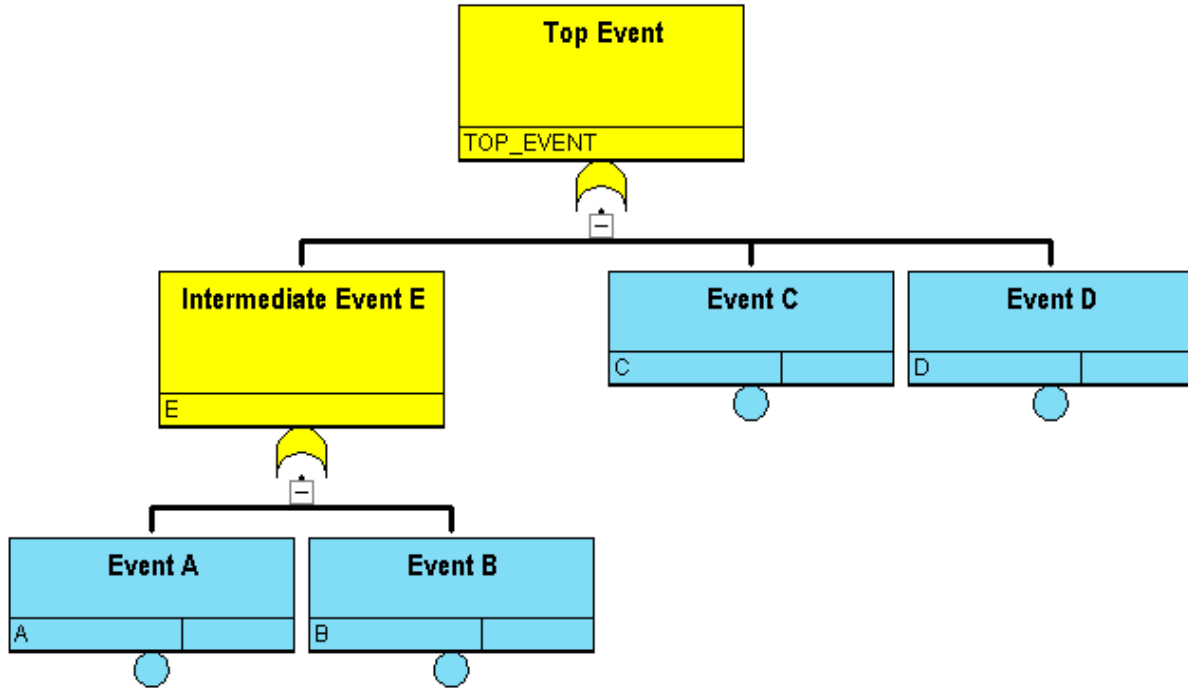


Figure 4-9. Typical Fault Tree Structure and Symbols.

If $\Pr(X)$ signifies the probability of event X , then the top event probability associated with Figure 4-9 is

$$\Pr(T) = \{Pr(A)[1 - Pr(B|A)] + Pr(B)\}Pr(C|A \cup B)Pr[D|(A \cup B) \cap C] \quad (4-15)$$

Some PRA software does not consider conditional probabilities unless they are expressly modeled, and employs the rare event approximation to quantify unions of events. With these restrictions, the corresponding software approximation to Equation (4-15) is

$$\Pr(T) \approx [Pr(A) + Pr(B)]Pr(C)Pr(D) \quad (4-16)$$

Because of these limitations, caution must be exercised to ensure that logic models are compatible with all approximations programmed into the PRA software algorithms.

The evaluation of a FT can be accomplished in two major steps:

1. reduction; and
2. quantification.

A collection of basic events whose simultaneous occurrence engenders the top event is called a cut set. Minimal cut sets (MCSs) are cut sets containing the minimum subset of basic events whose simultaneous occurrence causes the top event to occur. Boolean reduction of a FT has the objective of reducing the FT to an equivalent form that contains only MCSs. This is accomplished by sequential application of the basic laws of Boolean algebra to the original logic embodied in the FT until the simplest logical expression emerges. Quantification of the FT is the evaluation of the probability of the top event in terms of the probabilities of the basic events using the reduced Boolean expression of MCSs. By combining the Boolean expression for

individual FTs into event sequences (by linking them through the ETs), an expression analogous to Equation (4-1) results.

FT construction is guided by the definition of the top event. This is predicated upon the system success criteria. The top event is derived by converting the success criteria for the system into a statement of system failure.

Starting with the top event, the FT is developed by deductively determining the cause of the previous fault, continually approaching finer resolution until the limit of resolution is reached. In this fashion the FT is developed from the system end point backward to the failure source. The limit of resolution is reached when FT development below a gate consists only of basic events (i.e., faults that consist of component failures, faults that are not to be further developed, phenomenological events, support system faults that are developed in separate FTs, software errors, or human actions).

Basic events appear at the bottom of a FT and determine the level of detail the FT contains. FTs should be developed down to a level where appropriate failure data exist or to a level providing the results required by the analysis.

House events are often used in FT analysis as switches to turn logic on and off. Since their probability is quantified as unity or zero, they require no reliability data input. House events are frequently used to simulate conditional dependencies.

Failure rates for passive or dormant components tend to be substantially less than for active components. Hence, they are not always included in FTs. Exceptions are single component failures (such as a pipe break, bus failure, or structural fault) that can fail an entire system (i.e., single failure points), and failures that have a likelihood of occurrence comparable to other components included in the FT. Spurious signals that cause a component to enter an improper state can be excluded from the model if, after the initial operation, the component control system is not expected to transmit additional signals requiring the component to alter its operating state. Likewise, basic events relating to a component being in an improper state prior to an IE are not included if the component receives an automatic signal to enter its appropriate operating state under accident conditions.

Testing and maintenance of components can sometimes render a component or system unavailable. Unavailability due to testing or maintenance depends on whether the component or train is rendered inoperable by the test or maintenance, and, if so, on the frequency and the duration of the test or maintenance act. Component failure due to a fault, and component unavailability due to test or maintenance, are mutually exclusive events. Consequently, caution must be exercised during FT reduction to ensure that cut sets containing such impossible combinations are not included in the reduced model.

Two types of human errors are generally included in FTs. Pre-accident human errors occur prior to the IE. Post-accident human errors modeled in FTs involve failure to activate or align systems that do not receive an automatic signal following the initiation of an accident. Other human recovery actions are generally not modeled in system FTs. Chapter 7 describes the modeling and quantification of human errors.

Dependent failures defeat the redundancy or diversity that is employed to improve the availability of systems. They are the subject of Chapter 8.

Software errors that can cause or contribute to the top event must be incorporated into the FT model. A key issue in modeling the contribution of software errors is to fully comprehend the impact these errors can have on the system. For example, if successful system operation is dependent on software control, a catastrophic software error would fail the entire system,

regardless of the mechanical redundancy or diversity the system contains. Hence, such errors can directly cause the top event to occur. However, other software errors may only degrade system performance. In these situations a combination of software errors and component failures may be needed to cause the top event. To ensure that the FT analyst satisfactorily incorporates software errors into the system model, the FT and software risk assessments (subject of Chapter 9) should proceed in concert.

4.4 References

- 4-1 *Fault Tree Handbook with Aerospace Applications, Version 1.1*, NASA, August 2002.
- 4-2 *NASA System Safety Handbook, Volume 1*, NASA/SP-2010-580, December 2011.

5. Data Collection and Parameter Estimation

The focus of a data collection process is to inform future risk/reliability assessments, which themselves inform decision-making processes. The key idea here is that data “collection” and “analysis” are not performed in isolation – an understanding of the intended use and application of the process results should be present during the design and implementation of the analysis methods. In general though, PRA data analysis refers to the process of collecting and analyzing information in order to estimate various parameters of the PRA models, particularly those of the epistemic models. These include the parameters used to obtain probabilities of various events such as component failure rates, initiator frequencies, and human failure probabilities. Therefore, the two main phases of developing a PRA database are:

1. Information Collection and Classification
2. Parameter Estimation

Typical quantities of interest are:

- Internal Initiating Events (IEs) Frequencies
- Component Failure Frequencies
- Component Test and Maintenance Unavailability
- Common Cause Failure (CCF) Probabilities
- Human Error Rates
- Software Failure Probabilities

Developing a PRA database of parameter estimates involves the following steps:

- Model-Data Correlation (identification of the data needed to correspond to the level of detail in the PRA models, determination of component boundaries, failure modes, and parameters to be estimated, e.g., failure rates, MTTR)
- Data Collection (determination of what is needed, such as failure and success data to estimate a failure rate, and where to get it, i.e., identification of data sources, and collection and classification of the data)
- Parameter Estimation (use of statistical methods to develop uncertainty distributions for the model parameters)
- Documentation (how parameter uncertainty distributions were estimated, data sources used, and assumptions made)

5.1 PRA Parameters

Typical PRA parameters, and the underlying probability models, are summarized in Table 5-1. Note for each of these probability models, one or more parameters are to be evaluated since they represent epistemic uncertainty – these parameters are shown in **bold** in the table.

Table 5-1. Typical Probability Models in PRAs and Their Parameters.

Basic Event	Probability Models	Data Required
Initiating event	Poisson model $\Pr(k) = e^{-\lambda t} \frac{(\lambda t)^k}{k!}$ t: Mission time λ: frequency	Number of events k in time t
Component fails on demand	Constant probability of failure on demand, or q	Number of failure events k in total number of demands N
Standby component fails in time, or component changes state between tests (faults revealed on functional test only)	Constant standby failure rate $Q = 1 - \frac{1 - e^{-\lambda_s T_s}}{\lambda_s T_s}$ T_s : Time between tests λ_s: Standby failure rate	Number of events k in total time in standby T
Component in operation fails to run, or component changes state during mission (state of component continuously monitored)	Constant failure rate $U = 1 - e^{-\lambda_o T_m} \approx \lambda_o T_m$ T_m : Mission time λ_o: Operating failure rate	Number of events k in total exposure time T (total time standby component is operating, or time the component is on line)
Component unavailable due to test	$Q = \frac{T_{TD}}{T_s}$ T_{TD} : Test duration (only in the case of no override signal) T_s : Time between tests	Average test duration (T_{TD}) and time between tests (T_s)
Component unavailable due to corrective maintenance (fault revealed only at periodic test, or preventative maintenance performed at regular intervals)	$Q = \frac{T_U}{T_T}$ T_U : Total time unavailable while in maintenance (out of service) T_T : Total operating time	Total time out of service due to maintenance acts while system is operational, T_U , and total operating time T_T .
Component unavailable due to unscheduled maintenance (continuously monitored components)	$Q = \frac{\mu T_R}{1 + \mu T_R}$ T_R : Average time of a maintenance outage. μ : Maintenance rate	Number of maintenance acts r in time T (to estimate μ)
Standby component that is never tested. Assumed constant failure rate.	$Q = 1 - e^{-\lambda_s T_p}$ T_p : Exposure time to failure λ_m: Standby failure rate.	Number of failures r, in T units of (standby) time
CCF probability	α_1 through α_m where m is the redundancy level	n_1 through n_m where n_k is the number of CCF events involving k components

Table 5-1 also shows the data needed to estimate the various parameters. The type of data needed varies depending on the type of event and their specific parametric representation. For example, probabilities typically require Event Counts (e.g., Number of Failures), and exposure or “Success Data” (e.g., Total Operating Time). Other parameters may require only one type of data, such as Maintenance/Repair Duration for mean repair time distribution, and counts of multiple failures in the case of CCF parameter estimates.

5.2 Sources of Information

Ideally, parameters of PRA models of a specific system should be estimated based on operational data of that system. Often, however, the analysis has to rely on a number of sources and types of information if the quantity or availability of system-specific data are insufficient. In such cases surrogate data, generic information, or expert judgment are used directly or in combination with (limited) system-specific data. According to the nature and degree of relevance, data sources may be classified by the following types:

- Historical performance of successes and failures of an identical piece of equipment under identical environmental conditions and stresses that are being analyzed (e.g., direct operational experience).
- Historical performance of successes and failures of an identical piece of equipment under conditions other than those being analyzed (e.g., test data).
- Historical performance of successes and failures of a similar piece of equipment or similar category of equipment under conditions that may or may not be those under analysis (e.g., another program’s test data, or data from handbooks or compilations). General engineering or scientific knowledge about the design, manufacture and operation of the equipment, or an expert’s experience with the equipment.

5.2.1 Generic Data Sources

Generic data is surrogate or non-specific information related to a class of parts, components, subsystems, or systems. Most generic data sources cover hardware failure rates. All other data categories, particularly human and software failure probabilities, tend to be much more mission-specific, system-specific, or context dependent. As such, generic data either do not exist or need to be significantly modified for use in a PRA.

NASA has performed risk and reliability assessments for a variety of vehicles and missions for over 40 years. Each of these quantitative evaluations tends to increase the general collection of risk and reliability information when this information is stored or published for later use. In addition to the individual quantitative evaluations, NASA also manages incident reporting systems, for example the Problem Reporting and Corrective Action (PRACA) system. PRACA systems have served as key information repositories and have been used in analyses such as the Shuttle PRA and the Galileo RTG risk assessment. A selection of other NASA data collection systems includes:

- Center-specific Problem Reporting systems (to record pre- and operational anomalies)
- The Spacecraft On-Orbit Anomaly Reporting System (SOARS)
- The Problem Report/Problem Failure Report (PR/PFR) system
- Incident, surprise, and anomaly reports
- PRA and reliability analysis archives (e.g., Shuttle, ISS)
- Apollo Mission Reports

- The Mars Exploration Rover Problem Tracking Database
- Results of expert elicitation

Outside of NASA and associated industries, a large set of risk and reliability data/information is collected. While many of these knowledge sources fall into the category of “generic” data, their applicability to NASA applications may be high in certain instances. Examples of these sources include:

- Nonelectronic Parts Reliability Data, NPRD-2011, Reliability Information Analysis Center (RIAC)
- Electronic Parts Reliability Data, EPRD-1997, RIAC
- IEEE Std 500-1984
- NUCLARR (updated version is called NARIS)
- Nuclear industry EPIX/RADS system
- The Military Handbook for Reliability Prediction of Electronic Equipment, MIL-HDBK-217F
- Government-Industry Data Exchange Program (GIDEP)
- International Common Cause Failure Data Exchange (ICDE)

The format and content of the data vary depending on the source. For example, a failure mode/mechanism database provides fraction of failures associated with each Mode or Mechanism. Others provide direct or formula-based estimates of failure rates.

The first two databases are maintained by The Reliability Information Analysis Center (RIAC) in Rome, New York. These RIAC databases provide empirical field failure rate data on a wide range of electronic components and electrical, mechanical, and electro mechanical parts and assemblies. The failure rate data contained in these documents represent cumulative compilation from the early 1970s up to the publication year for each document. The RIAC handbooks provide point estimate parameter estimations for failure rates (or demand probabilities). No treatment of uncertainty is provided.

The Part Stress Analysis Prediction method of MIL-HDBK-217 provides base failure rates and method of specializing them for specific type or applications. The specialization considers factors such as part quality, environmental conditions, and part-type specific factors such as resistance and voltage (for resistors).

For example, for semiconductors:

$$\lambda_p = \lambda_b (\pi_E \times \pi_A \times \pi_{S2} \times \pi_C \times \pi_Q) \quad (5-1)$$

where, λ_p is part failure rate, λ_b is base failure rate, dependent on electrical and thermal stresses, and the π factors modify the base failure rate based on environmental conditions and other parameters affecting part reliability.

GIDEP is a cooperative activity between government and industry for the goal of sharing technical information essential during the life cycle of systems or components. GIDEP includes a database of “Reliability and Maintainability Data.”

Other sources of data include non-U.S. experience such as launch vehicle performance (e.g., ESA’s Ariane and Russia’s Soyuz and Proton). However, the availability of quality non-U.S. data is generally limited, with a few exceptions (e.g., the OREDA Offshore RELiability DAta).

In any given PRA a mix of generic and system-specific data sources may be used. The International Space Station PRA, for example, has relied on the following sources for hardware data:

- Modeling Analysis Data Sets (MADS)
- Contractor Reliability & Maintainability Reports
- Russian Reliability & Maintainability Reports
- Non-electronic Parts Reliability Database 1995 (NPRD)
- Electronic Parts Reliability Database 1997 (EPRD)
- Failure Mode Distribution 1997 (FMD)
- Bellcore TR-332: Reliability Prediction Procedure for Electronic Equipment
- Problem Reporting and Corrective Action (PRACA) System.

Irrespective of the source of data used, generic data must be evaluated for applicability, and often modified before being used as surrogate data.

5.2.2 System-Specific Data Collection and Classification

System-specific data can be collected from sources such as:

- Maintenance Logs
- Test Logs
- Operation Records.

As shown in Table 5-1, the data needed vary depending on the type of event and their specific parametric representation. Most cases require counts of events (e.g., failures) and corresponding exposure data (e.g., operating hours).

In the majority of cases, system-specific data are gathered from operation and test records in their “raw” form (i.e., in the form that cannot be directly used in a statistical analysis). Even when data have already been processed (e.g., reduced to counts of failure), care must be exercised to ensure that the data reduction and processing are consistent with PRA modeling requirements, such as having a consistent failure mode classification, and correct count of the total number of tests or actual demands on the system).

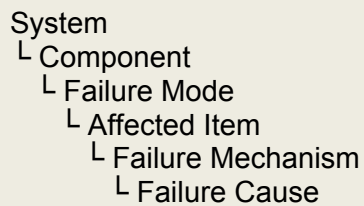
In collecting and classifying hardware failure, a systematic method of classification and failure taxonomy is essential. A key element of such taxonomies is a classification of the functional state of components. One such classification system has been offered in Reference [5-1].

Using a taxonomy implies a knowledge structure used to describe a parent-child relationship (i.e., a hierarchy). Under the guidelines for evaluation of risk and reliability-related data, the taxonomy provides the structure by which data and information elements provide meaning to analysts. Within the risk and reliability community, a variety of taxonomies and associated definitions are used.

If one were concerned about the physical *causes* of failures, a set of physics-based causal factors such as these would be required. However, this low level of information is not necessary if the inference being made for a specific component or system is concerned with – in general – failures or successes, as shown in Table 5-1. If, instead, we wished to infer the probability of

failure conditional upon a specific failure mechanism, we would need to have information related to the nature of failure (e.g., the physical causal mechanisms related to specific failures).

In other words, this classification can take place via a failure modes and effects analysis, similar to the functional failure modes and effects analysis. Henley and Kumamoto carried this idea one step further when they proposed a formal cause-consequences structure to be stored in an electronic database [5-2]. In their approach, specific keywords, called modifiers, would be assigned to equipment failures. For example, modifiers for on-off operation included: close, open, on, off, stop, restart, push, pull, and switch. Alternative hierarchy related to system/component/failure modes may look like:



Outside of NASA, a new standard, ISO 14224, focused on the collection and processing of equipment failure data has been produced. Other guidance on data collection taxonomies may be found from the following sources:

- ISO 6527:1982 Nuclear power plants -- Reliability data exchange -- General guidelines
- ISO 7385:1983 Nuclear power plants -- Guidelines to ensure quality of collected data on reliability
- ISO 14224:2006 Petroleum, petrochemical and natural gas industries -- Collection and exchange of reliability and maintenance data for equipment

An example component state classification is shown in Figure 5-1. With regard to the intended function and in reference to a given performance criterion, a component can be in two states: *available* or *unavailable*. The unavailable state includes two distinct sub-states: *failed* and *functionally unavailable*, depending on whether the cause of the unavailability is damage to the component or lack of necessary support such as motive power. The state classification also recognizes that even when a component may be capable of performing its function (i.e., it is available), an incipient or degraded condition could exist in that component, or in a supporting component. These failure situations are termed *potentially failed* and *potentially functionally unavailable*, respectively. These concepts have proven useful in many PRA data applications.

Another aspect of reliability data classification is the identification of the failure cause. In the context of the present discussion, the cause of a failure event is a condition or combination of conditions to which a change in the state of a component can be attributed. It is recognized that the description of a failure in terms of a single cause is often too simplistic. A method of classifying causes of failure events is to progressively unravel the layers of contributing factors to identify *how* and *why* the failure occurred. The result is a chain of causal factors and symptoms.

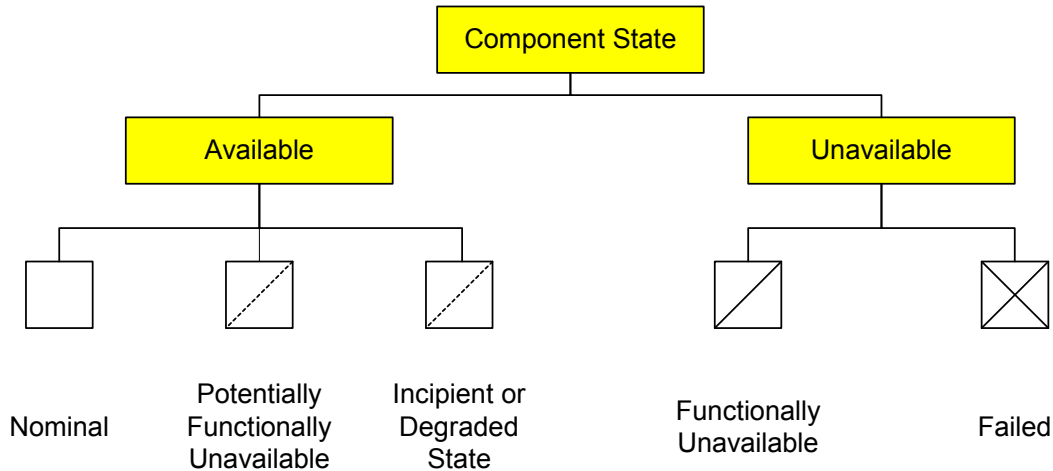


Figure 5-1. Component Functional State Classification.

A hierarchy of *parts or items* that make up a component is first recognized, and the functional failure mode of the component is attributed to the failure or functional unavailability of a subset of such parts or items. Next the physical sign or *mechanism* of failure (or functional unavailability) of the affected part(s) or item(s) are listed. Next the *root cause* of the failure mechanism is identified. Root cause is defined as the most basic reason or reasons for the failure mechanism, which if corrected, would prevent reoccurrence. The root cause could be any causal factor, or a combination of various types of causal factors.

Figure 5-2 shows the system/component/failure event classification process highlighting the part that deals with failure cause classification. We note that the cause classification starts by identifying the part or item within the components that was affected by the failure event. It is assumed that other attributes in failure event classification such as component type and functional failure mode (e.g., failure to start) at the component level are recorded earlier. The second step is to identify the failure mechanism affecting the part or item within the component. Finally the root cause of the failure mechanism is listed.

Figure 5-3 provides an example of a more detailed listing of the various classification categories under each of the three steps of the cause classification process. The level of details and sub-categories provided for each step is not necessarily complete or comprehensive for all applications. However, the structure, classification flow, and categories capture the essence of a large number of failure cause classification approaches in the literature. In real world applications, due to the limitations in the information base, it may be difficult or impossible to identify some of these attributes for a given event.

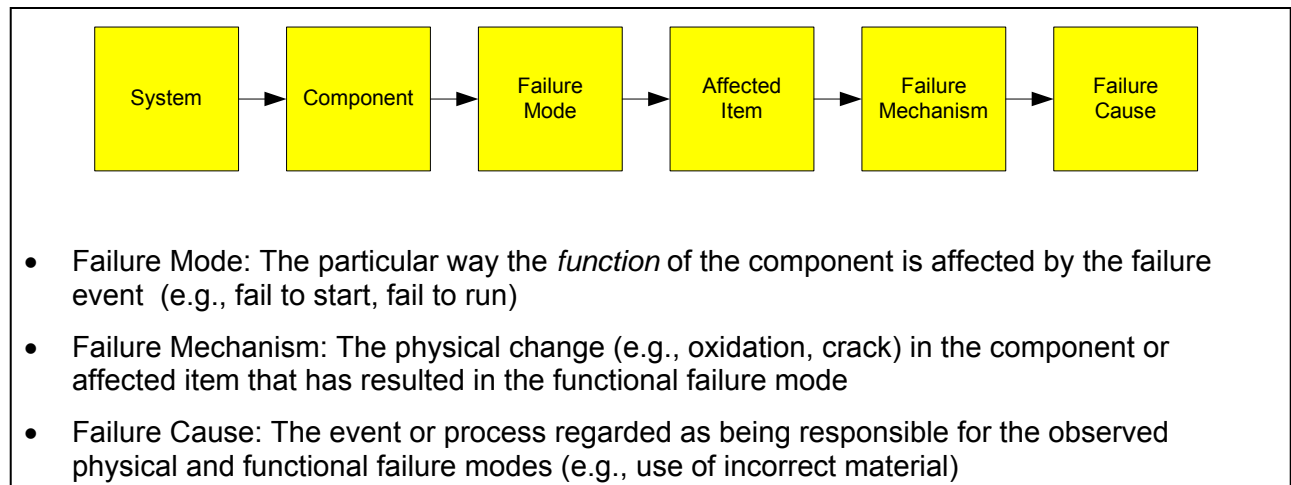


Figure 5-2. Failure Event Classification Process Flow.

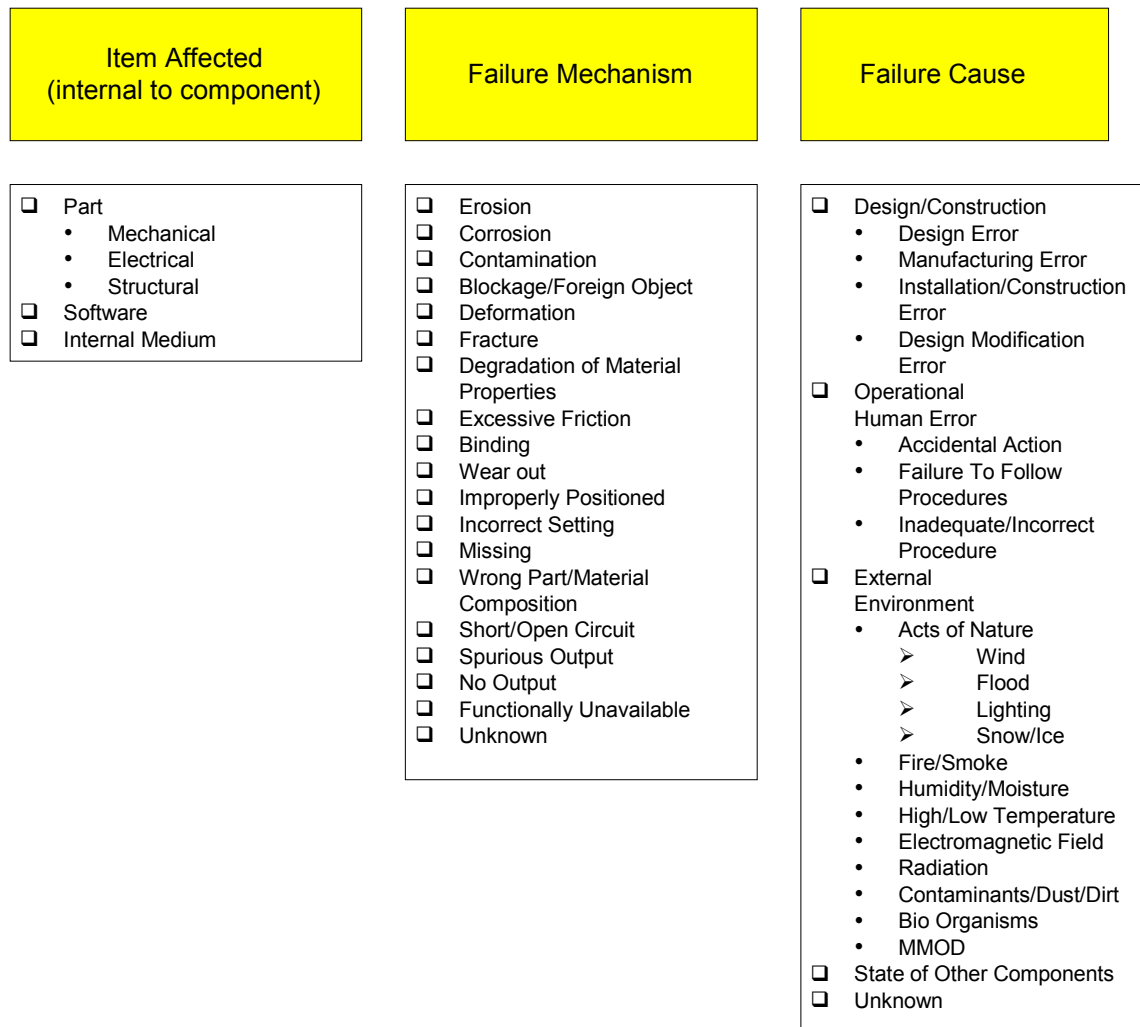


Figure 5-3. Failure Cause Classification Subcategories.

5.3 Parameter Estimation Method

As discussed earlier in this Guide, Bayesian methods are widely used in PRA, while classical estimation has found only limited and restricted use. Therefore, this section describes only the Bayesian approach to parameter estimation.

Bayesian estimation incorporates degree of belief and information beyond that contained in the data sample, forming the practical difference from classical estimation. The subjective interpretation of probability forms the philosophical difference from classical methods. Bayesian estimation is comprised of two main steps. The first step involves using available information to fit a prior distribution to a parameter, such as a failure rate. The second step of Bayesian estimation involves using additional or new data to update the prior distribution. This step is often referred to as “Bayesian Updating.”

Bayes’ Theorem, presented in Section 6.6 transforms the prior distribution via the likelihood function that carries the new data. Conceptually

$$\text{Posterior Distribution} = \frac{\text{Prior Distribution} \times \text{Likelihood}}{\text{Normalizing Constant}} \propto \text{Prior Distribution} \times \text{Likelihood} \quad (5-2)$$

Bayes' Theorem has been proven to be a powerful coherent method for mathematically combining different types of information while also expressing the inherent uncertainties. It has been particularly useful in encapsulating our knowledge about the probability of rare events about which information is sparse.

Bayes' Theorem provides a mathematical framework for processing new data, as they become available over time, so that the current posterior distribution can then be used as the prior distribution when the next set of data becomes available.

Bayesian inference produces a probability distribution. A "credible interval" consists of the values at a set of specified percentiles (one low, one high) from the resultant distribution. For example, a 90% credible interval ranges from the value of the 5th percentile to the value of the 95th percentile. Note that the credible interval will also be referred to as a "probability interval."

For PRA applications, determining the prior distribution is usually based on generic data, and the new or additional data usually involve system-specific test or operating data. The resulting posterior distribution would then be the system-specific distribution of the parameter. In the case where system-specific data did not exist, the applicability of other data or information would need to be evaluated and used – this treatment falls under the topic of uncertain data and is described in Section 4.5 of the NASA Bayesian Inference Guide [5-4].

5.4 Prior Distributions

Prior distributions can be specified in different forms depending on the type and source of information as well as the nature of the random variable of interest. Possible forms include:

- Parametric (gamma, lognormal, beta):
 - Gamma or lognormal for rates of events (time-based reliability models)
 - Beta or truncated lognormal for event probabilities per demand
- Numerical (histogram, CDF values/percentiles)
 - Applicable to both time-based and demand-based reliability parameters.

Among the parametric forms, a number of probability distributions are extensively used in risk studies as prior and posterior distributions. These are:

- Lognormal (μ, σ)

$$\pi(x) = \frac{1}{\sqrt{2\pi} \sigma x} e^{-\frac{1}{2} \left(\frac{\ln x - \mu}{\sigma}\right)^2} \quad 0 < x < \infty. \quad (5-3)$$

where μ and σ are the parameters of the distribution. The lognormal distribution can be truncated (truncated lognormal) so that the random variable is less than a specified upper bound.

- Gamma (α, β)

$$\pi(x) = \frac{x^{\alpha-1} \beta^\alpha}{\Gamma(\alpha)} e^{-\beta x} \quad 0 \leq x < \infty. \quad (5-4)$$

where α and β are the parameters of the distribution.

- Beta (α, β)

$$\pi(x) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1} \quad 0 \leq x \leq 1 \quad (5-5)$$

where α and β are the parameters of distribution.

Information content of prior distributions can be based on:

- Previous system-specific estimates
- Generic, based on actual data from other (similar) systems
- Generic estimates from reliability sources
- Expert judgment (see discussion in Chapter 6)
- “Non-informative.” This type is used to represent the state of knowledge for the situations where little *a priori* information exists or there is indifference about the range of values the parameter could assume. A prior distribution that is uniformly distributed over the interval of interest is a common choice for a non-informative prior. However, other ways of defining non-informative prior distributions also exist.

The NASA Bayesian Inference Guide, NASA-SP-2009-569, between pages 47 and 54, suggests prior distributions and provides examples for use when faced with limited information [5-4]:

Information Available	Suggested Prior Distribution
Mean value for lambda in the Poisson distribution	Gamma distribution with alpha = 0.5 and beta = 1/(2 × mean)
Mean value for p in the binomial distribution	Beta distribution with alpha ≈ 0.5 and beta = (1 – mean)/(2 × mean)
Mean value for lambda in the exponential distribution	Gamma distribution with alpha = 1 and beta = 1/mean
p in binomial distribution lies between a and b	Uniform distribution between a and b

5.5 Selection of the Likelihood Function

The form of the likelihood function depends on the nature of the assumed *Model of the World* representing the way the new data/information is generated:

For **data generated from a Poisson Process** (e.g., counts of failures during operation), the Poisson distribution is the proper likelihood function

$$\Pr(k|T, \lambda) = \frac{(\lambda T)^k}{k!} e^{-\lambda T} \quad (5-6)$$

which gives the probability of observing k events (e.g., number of failures of a component) in T units of time (e.g., cumulative operating time of the component), given that the rate of occurrence of the event (failure rate) is λ . The MLE of λ is (see Chapter 6)

$$\hat{\lambda} = \frac{k}{T} \quad (5-7)$$

It is also possible to combine data from several independent Poisson processes, each having the same rate λ . This applies to the case where data are collected on identical equipment to estimate their common failure rate. The failure counting process for each equipment is assumed to be a Poisson process. In particular, suppose that the i th Poisson process is observed for time t_i , yielding the observed count k_i . The total number of event occurrences is $k = \sum_i k_i$ where the sum is taken over all of the processes, and the exposure time is $T = \sum_i t_i$. This combined evidence can be used in the likelihood function of Equation (5-6).

For **data generated from a Bernoulli Process** (e.g., counts of failures on system demands), the Binomial distribution is the proper likelihood function:

$$\Pr(k | N, q) = \binom{N}{k} q^k (1 - q)^{N-k} \quad (5-8)$$

which gives the probability of observing k events (e.g., number of failures of a component) in N trials (e.g., total number of tests of the component), given that the probability of failure per trial (failure on demand probability) is q . The MLE of q is:

$$\hat{q} = k / N \quad (5-9)$$

Similar to the case of Poisson Likelihood, data generated by independent Bernoulli Processes having the same parameter q may be combined. Denoting the number of failures and demands at data source j by k_j and n_j , respectively, let $k = \sum_j k_j$ and $N = \sum_j n_j$.

These cumulative numbers are then used in the likelihood function of Equation (5-8).

For **data in the form of expert estimates** or values for data sources (e.g., a best estimate based on MIL-STD-217), the lognormal distribution is a common likelihood function.

5.6 Development of the Posterior Distribution

Using Bayes' Theorem in its continuous form, the *prior* probability distribution of a continuous unknown quantity, $\Pr_o(x)$ can be updated to incorporate new evidence E as follows:

$$\Pr(x|E) = \frac{L(E|x) \Pr_o(x)}{\int L(E|x) \Pr_o(x) dx} \quad (5-10)$$

where $\Pr(x|E)$ is the *posterior* or updated probability distribution of the unknown quantity X given evidence E (occurrence of event E), and $L(E|x)$ is the *likelihood* function (i.e., probability of the evidence E assuming the value of the unknown quantity is x). The various combinations of prior and likelihood functions as well as the form of the resulting posterior distributions are listed in Table 5-2.

Table 5-2. Typical Prior and Likelihood Functions Used in PRAs.

Prior	Likelihood	Posterior
Lognormal	Poisson	Numerical
Gamma	Poisson	Gamma
Beta	Binomial	Beta
Truncated Lognormal	Binomial	Numerical

Many practical applications of Bayes' Theorem require numerical solutions to the integral in the denominator of Bayes' Theorem. Simple analytical forms for the posterior distribution are obtained when a set of prior distributions, known as *conjugate prior distributions*, are used. A conjugate prior distribution is a distribution that results in a posterior distribution that is a member of the same family of distributions as the prior.

Two commonly used conjugate distributions are listed in Table 5-3. The formulas used to calculate the mean and the variance of the resultant posterior in terms of the parameters of prior and likelihood functions are provided.

Table 5-3. Common Conjugate Priors Used in Reliability Data Analysis.

Conjugate Prior Distribution	Likelihood Function	Posterior Distribution	Mean of Posterior	Variance of Posterior
Beta (α, β)	Binomial (k, N)	Beta	$\bar{x} = \frac{\alpha + k}{\alpha + \beta + N}$	$\text{var}(x) = \frac{(\alpha + k)(\beta + N - k)}{(\alpha + \beta + N)^2 (\alpha + \beta + N + 1)}$
Gamma (α, β)	Poisson (k, T)	Gamma	$\bar{x} = \frac{\alpha + k}{\beta + T}$	$\text{var}(x) = \frac{\alpha + k}{(\beta + T)^2}$

Example 4: Updating of Prior for a Poisson Example

It is assumed that the total operational data for the component category indicate 2 failures in 10,000 hours. Since the prior distribution is lognormal, and the likelihood function is Poisson, the posterior distribution must be derived numerically. Both the prior and posterior distributions are shown in Figure 5-4. The Prior and Posterior Distributions of Example 4.. Note that the pdfs are plotted as a function of $\log \lambda$.

The shift toward the operational data is a characteristic of the posterior distribution, as compared to the prior distribution (see Chapter 5 for discussion on relation between posterior and data used in Bayesian updating).

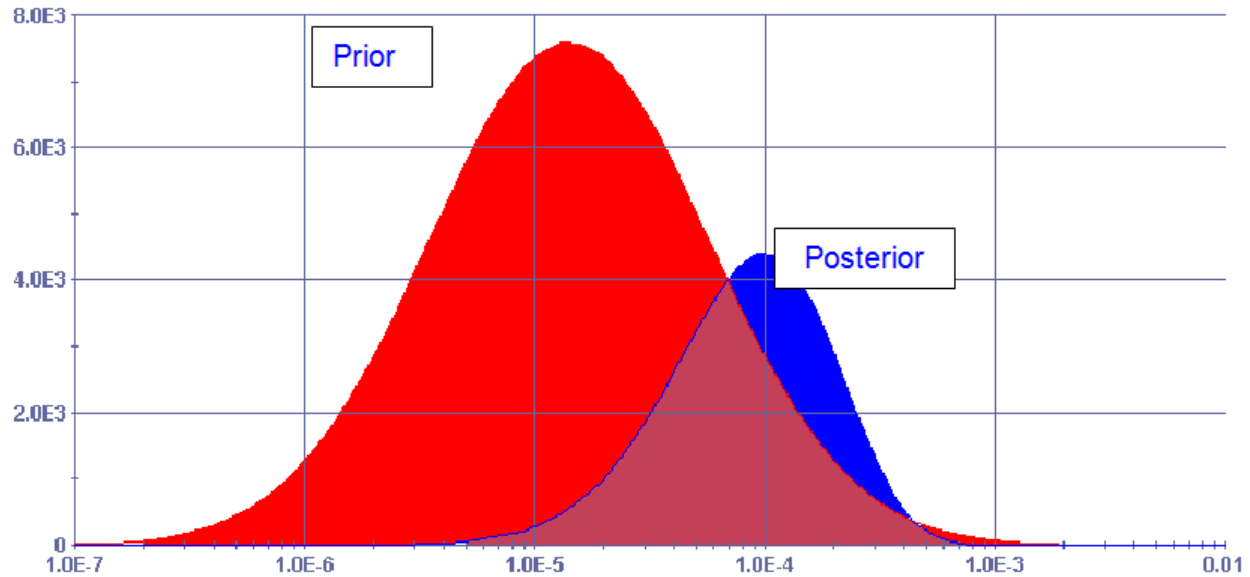


Figure 5-4. The Prior and Posterior Distributions of Example 4.

Example 5: Updating Distribution of Failure on Demand Probability

It is assumed that the prior distribution of a component failure probability on demand is characterized by a beta distribution with Mean = $1E-4$ failures per demand, and Standard Deviation = $7E-5$. It is also assumed that the operational data for the component category indicate 1 failure in 2,000 demands. Since the prior distribution is a Beta, and the likelihood function is Binomial, the posterior distribution is also a Beta distribution. Both the prior and posterior distributions are shown in Figure 5-5.

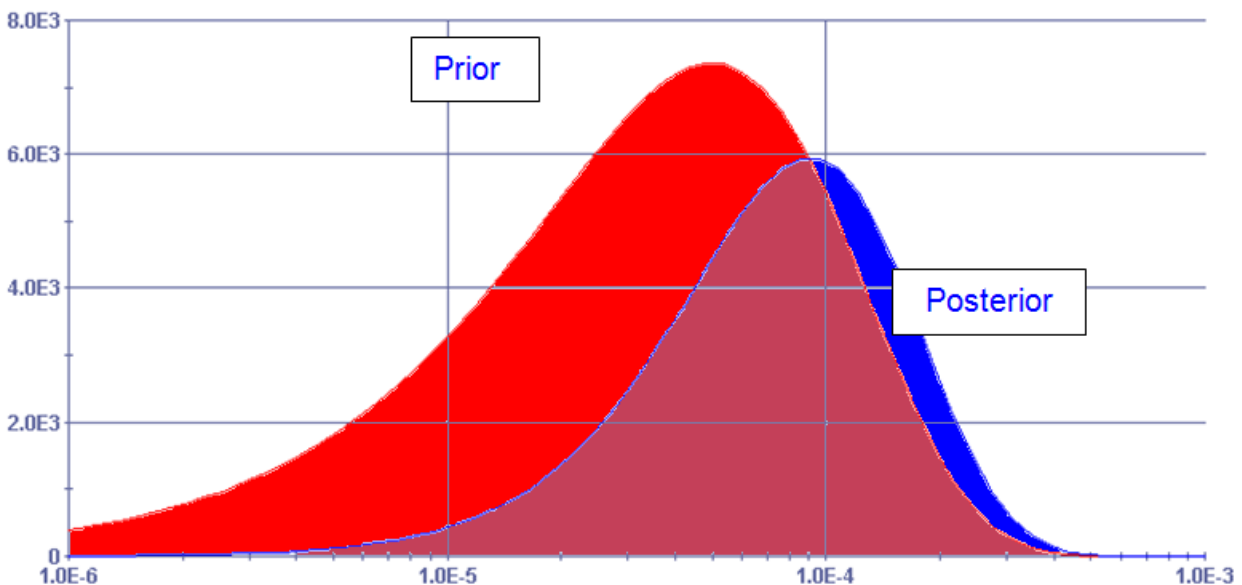


Figure 5-5. The Prior and Posterior Distributions of Example 5.

5.7 Sequential Updating

Bayes' Theorem provides a mechanism for updating the state-of-knowledge when the information is accumulated in pieces. The updating process can be performed sequentially and in stages corresponding to the stages in which various pieces of information become available. If the total amount of information is equivalent to the "sum" of the pieces, then the end result (posterior distribution) is the same regardless of whether it has been obtained in stages (by applying Bayes' Theorem in steps) or in one step (by applying Bayes' Theorem to the cumulative information).

Example 6: Updating Failure Rate for a Poisson Process

A component is tested for 1000 hours in one test and 4000 hours in another. During the first test the component does not fail, while in the second test one failure is observed. We are interested in an updated estimate of the component failure rate assuming a gamma prior distribution with parameters $\alpha = 1$, $\beta = 500$.

Approach 1: We first start with prior (Gamma distribution): $G(x|\alpha = 1, \beta = 500)$. We also use Poisson as the likelihood function: $\Pr(k_1 = 0|T_1 = 1000, \lambda)$ representing the results of the first test ($k_1 = 0$ in $T_1 = 1000$ hours). The parameters of the resulting Gamma posterior distribution are $\alpha' = \alpha + k_1 = 1 + 0 = 1$, and $\beta' = \beta + T_1 = 500 + 1000 = 1500$ (see Table 5-3).

Next, we use this posterior as prior distribution and update it with the information from the second test. Therefore, the prior is $G(\lambda|\alpha' = 1, \beta' = 1500)$ and the likelihood is again Poisson: $\Pr(k_2 = 1|T_2 = 4000, \lambda)$. The parameters of the posterior after the second update are $\alpha'' = \alpha' + k_2 = 1 + 1 = 2$, and $\beta'' = \beta' + T_2 = 1500 + 4000 = 5500$. The posterior mean is given by (see Table 5-3):

$$\bar{\lambda} = \frac{\alpha''}{\beta''} = \frac{2}{5500} = 3.6\text{E-}4 \text{ failures/hour}$$

Approach 2: The total evidence on the failure history of the component in question is $k = k_1 + k_2 = 0 + 1 = 1$, and $T = T_1 + T_2 = 1000 + 4000 = 5000$. Starting with our prior distribution with parameters, $\alpha = 1$, $\beta = 500$, the above cumulative evidence can be used in one application of Bayes' Theorem with Poisson likelihood: $\Pr(k = 1|T_2 = 5000, \lambda)$

The parameters of the resulting Gamma posterior distribution are $\alpha' = \alpha + k = 1 + 1 = 2$, $\beta' = \beta + T = 500 + 5000 = 5500$, and

$$\bar{\lambda} = \frac{\alpha'}{\beta'} = \frac{2}{5500} = 3.6\text{E-}4 \text{ failures/hour}$$

which are identical to values obtained with the first approach.

5.8 Developing Prior Distributions from Multiple Sources of Generic Information

Typically, generic information can be categorized into two types:

- **Type 1** Failure data from operational experience with other similar but not identical components, or from identical components under different operating conditions. This

information is typically in the form of failure and success data collected from the performance of similar equipment in various systems. The data in this case are assumed to come from a “non-homogenous” population.

- **Type 2** Failure rate estimates or distributions contained in various industry compendia, such as several of the databases discussed earlier. Estimates from expert judgment elicitations would be included in this category. Type 2 data are either in the form of point estimates (or “best estimates”), or a range of values centered about a “best estimate.” Ranges of the best estimate can be expressed in terms of low, high, and recommended values, or as continuous probability distributions.

When multiple sources of generic data are available, then it is likely that we are dealing with a non-homogeneous population. In these cases the data cannot be pooled, and the reliability parameter of interest (e.g., failure rate) will have an inherent variability. The probability distribution representing this variability is known as a *population variability distribution* of the reliability parameter of interest.

NASA-SP-2009-569 [5-4] describes both Type 1 and Type 2 approaches for Bayesian inference. For example, Section 4.5 of that document discusses population variability models for Binomial, Poisson, and CCF models. For the case where we need to combine different sets of data or information, Sections 4.8.2 and 4.8.4 describe various Bayesian approaches.

5.9 Guidance for Bayesian Inference Calculations

As mentioned, NASA-SP-2009-569 [5-4] provides a collection of quantitative methods to address the analysis of data and its use in PRA. The coverage of the technical topics in that guide addresses items such as the historical genesis of Bayesian methods; comparisons of “classical statistics” approaches with Bayesian ones; the detailed mathematics of particular methods; and sources of reliability or risk data/information. Bayesian methods and multiple examples are provided for a variety of PRA inference topics, including:

- Binomial modeling (conjugate, noninformative, non-conjugate)
- Poisson modeling (conjugate, noninformative, non-conjugate)
- Exponential modeling (conjugate, noninformative, non-conjugate)
- Multinomial modeling (conjugate, noninformative, non-conjugate)
- Model validation
- Time-trend modeling
- Pooling and population variability modeling
- Time-to-failure modeling
- Repairable system modeling
- Uncertain data
- Regression models
- Expert elicitation

5.10 References

- 5-1 A. Mosleh et al., “Procedures for Treating Common Cause Failures in Safety and Reliability Studies,” U.S. Nuclear Regulatory Commission and Electric Power Research Institute, NUREG/CR-4780, and EPRI NP-5613. Volumes 1 and 2, 1988.

- 5-2 S Henley, E. J. and H. Kumamoto, 1985, *Designing for Reliability and Safety Control*, Prentice-Hall.
- 5-3 S. Kaplan, "On a 'Two-Stage' Bayesian Procedure for Determining Failure Rates from Experiential Data," PLG-0191, *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-102, No. 1, PLG-0191, January 1983.
- 5-4 NASA-SP-2009-569, *Bayesian Inference for NASA Risk and Reliability Analysis*, 2009.

6. Uncertainties in PRA

The purpose of this chapter is to present the basic structure of uncertainty analyses in PRAs. This chapter discusses how PRA models are constructed and why uncertainties are an integral part of these models.

6.1 The Model of the World

The first step in doing a PRA is to structure the problem, which means to build a model for the physical situation at hand. This model is referred to as the *model of the world* [6-1]. It may occasionally be referred to it as the “model” or the “mathematical model.” It is built on a number of model assumptions and typically includes a number of parameters whose numerical values are assumed to be known.

An essential part of problem structuring in most PRAs is the identification of accident scenarios (event sequences) that may lead to the consequence of interest, e.g., system unavailability, loss of crew and vehicle, and so forth. Many methods have been developed to aid the analysts in such efforts. Examples are: Failure Modes and Effects Analysis (FMEA), hazard and operability analysis, FTA, and ETA. These analyses consider combinations of failures of the hardware, software, and human actions in risk scenarios.

The development of scenarios introduces model assumptions and model parameters that are based on what is currently known about the physics of the relevant processes and the behavior of systems under given conditions. For example, the calculation of heat flux in a closed compartment where a fire has started, and the response of the crew, are the results of conceptual models that rely on assumptions about how a real accident would progress. These models include parameters whose numerical values are assumed to be available (for example, in the case of fires, the heat of combustion of the burning fuel).

There are two types of models of the world, *deterministic* and *probabilistic*. A simple example of a deterministic model is the calculation of the horizontal distance that a projectile travels under the influence of the force of gravity. If the projectile is launched at an angle ϕ with the horizontal axis and with an initial speed v , Newton’s law yields:

$$q(v, \phi | M) = \frac{v^2}{g} \sin(2\phi) \quad (6-1)$$

where g is the gravitational acceleration. This expression shows explicitly that the calculated distance is a function of v and ϕ , and that it is conditional on the assumption, M , that the hypothesis that the projectile is under the influence of the force of gravity only, is valid.

Many important phenomena cannot be modeled by deterministic expressions such as that of Equation (6-1). For example, the failure times of equipment exhibit variability that cannot be eliminated; given the present state of knowledge and technology, it is impossible to predict when the next failure will occur. So one must construct models of the world that include this uncertainty. A simple example that will help; it involves the failure of a pump. The “random variable” is T , the failure time. Then, the distribution of this time is usually taken to be exponential, i.e.,

$$F(t|\lambda, M) = 1 - \exp(-\lambda t) \quad (6-2)$$

This is the probability that T is smaller than t , i.e., that the pump will fail before t (see Figure 3-10). Note that a probability is a measure of the degree of plausibility of a hypothesis and is evaluated on a 0 to 1 scale.

The parameter λ , the failure rate in Equation (6-2), specifies $F(t)$. Its value depends on the kinds of pumps that have been included in the class of pumps and on the conditions of their use. Thus, the value of λ depends on what is included in the model. It is important to realize that the pumps and conditions of operation that are included in the model are assumed to be completely equivalent (as far as the behavior of T is concerned). That is, if there is no distinction between two different systems, it is assumed that two pumps, of the type of interest, in these two systems are not distinguishable. Similar to Equation (6-1), Equation (6-2) shows explicitly that this model is conditional on the set of assumptions M . The fundamental assumption behind the exponential failure distribution is the constancy of the failure rate λ .

The uncertainty described by the model of the world is sometimes referred to as “randomness,” or “stochastic uncertainty.” Stochastic models of the world have also been called *aleatory* models. This chapter will also use this terminology because, unlike the terms “randomness” and “stochastic,” it is not used in other contexts, so that confusion is avoided. For detailed discussions on PRA uncertainties, see References [6-2] through [6-4].

It is important to point out that models of the world, regardless of whether they are deterministic or aleatory, deal with observable quantities. Equation (6-1) calculates a distance, while Equation (6-2) deals with time. Both distance and time are observable quantities.

6.2 The Epistemic Model

As stated in the preceding section, each model of the world is conditional on the validity of its assumptions and on the availability of numerical values for its parameters. Since there may be uncertainty associated with these conditions, this section introduces the *epistemic* model, which represents the state of knowledge regarding the numerical values of the parameters and the validity of the model assumptions.

The issue of alternative model assumptions (model uncertainty or epistemic uncertainty) is usually handled by performing sensitivity studies. In the large majority of cases, the focus is on the uncertainties regarding the numerical values of the parameters of a given model (parameter uncertainty), rather on the uncertainty regarding the validity of the model itself.

For the example of Equation (6-2), the epistemic probability density function (pdf) $\pi(\lambda)$ is introduced, which expresses the state of knowledge regarding the numerical values of the parameter λ of a given model. Unlike aleatory models of the world, the epistemic models deal with non-observable quantities. Failure rates and model assumptions are not observable quantities.

A consequence of this formulation is as follows. Consider a system of two nominally identical pumps in series. Let R_S be the system reliability and R_1 and R_2 the reliabilities of the two pumps. Then, under the assumption of independence of failures, the reliability of the system is given by

$$R_S = R_1 R_2 \quad (6-3)$$

Suppose now that the failure times of these pumps follow the exponential distribution, Equation (6-2). Suppose further that the epistemic pdf for the failure rate is $\pi(\lambda)$. Even though the two pumps are physically distinct, the assumption that they are nominally identical requires that the same value of λ be used for both pumps. Then, Equation (6-3) becomes

$$R_s = \exp(-2\lambda t) \quad (6-4)$$

The reason is that saying that the pumps are nominally identical means that they have the same failure rate [6-5]. The epistemic model simply gives the distribution of the values of this failure rate according to our current state of knowledge.

Further discussion on the need for separating aleatory and epistemic uncertainties can be found in References 6-6 and 6-7.

6.3 A Note on the Interpretation of Probability

To evaluate or manipulate data, we must have a "model of the world" (or simply "model") that allows us to translate real-world observables into information. [6-2] Within this model of the world, there are two fundamental types of model abstractions, aleatory and deterministic. The term "aleatory" when used as a modifier implies an inherent "randomness" in the outcome of a process. For example, flipping a coin^a is modeled as an aleatory process, as is rolling a die. When flipping a coin, the "random" but observable data are the outcomes of the coin flip, that is, heads or tails. Note that since **probabilities** are **not** observable quantities, we do not have a model of the world directly for probabilities. Instead, we rely on aleatory models (e.g., a Bernoulli^b model) to predict probabilities for observable outcomes (e.g., two heads out of three tosses of the coin).

Model (of the world) A mathematical construct that converts information (including data as a subset of information) into knowledge. Two types of models are used for risk analysis purposes, aleatory and deterministic.

Aleatory Pertaining to stochastic (non-deterministic) events, the outcome of which is described by a probability. From the Latin *alea* (game of chance, die).

Deterministic Pertaining to exactly predictable (or precise) events, the outcome of which is known with certainty if the inputs are known with certainty. As the antitheses of aleatory, this is the type of model most familiar to scientists and engineers and include relationships such as $E=mc^2$, $F=ma$, $F=G \frac{m_1 m_2}{r^2}$, etc.

The models that will be described herein are parametric, and most of the model parameters are themselves imprecisely known, and therefore uncertain. Consequently, to describe this second layer of uncertainty, we introduce the notion of epistemic uncertainty. **Epistemic** uncertainty represents how precise our state of knowledge is about the model (including its parameters), regardless of the type of

^a Flipping a coin is deterministic in principle, but the solution of the "coin-flipping dynamics" model, including knowledge of the relevant boundary conditions, is too difficult to determine or use in practice. Hence, we abstract the flipping process via an aleatory model of the world.

^b A Bernoulli trial is an experiment outcome that can be assigned to one of two possible states (e.g., success/failure, heads/tails, yes/no). The outcomes are assigned to two values, 0 and 1. A Bernoulli process is obtained by repeating the same Bernoulli trial, where each trial is independent. If the outcome assigned to the value 1 has probability p , it can be shown that the summation of n Bernoulli trials is binomial distributed $\sim \text{Binomial}(p, n)$.

model. Whether we employ an aleatory model (e.g., Bernoulli model) or a deterministic model (e.g., applied stress equation), **if** any parameter in the model is imprecisely known, then there is epistemic uncertainty associated with the model. Stated another way, if there is epistemic uncertainty associated with the parametric inputs to a model, then there is epistemic uncertainty associated with the output of the model, as well.

Epistemic Pertaining to the degree of knowledge of models and their parameters. From the Greek episteme (knowledge).

It is claimed that models have epistemic uncertainty, but is there epistemic uncertainty associated with other elements of our uncertainty taxonomy? The answer is yes, and in fact almost all parts of our taxonomy have a layer of epistemic uncertainty, including the data, context, model information, knowledge, and inference.

In summary:

- We employ mathematical models of reality, both deterministic and aleatory.
- These models contain parameters – whose values are estimated from information – of which data are a subset.
- Uncertain parameters (in the epistemic sense) are inputs to the models used to infer the values of future observables, leading to an increase in scientific knowledge. Further, these parameters may be known to high precision and thus have little associated epistemic uncertainty (e.g., the speed of light, the gravitational constant), or they may be imprecisely known and therefore subject to large epistemic uncertainties (e.g., frequency of lethal solar flares on the Moon, probability of failure of a component).

Visually, our taxonomy appears as shown in Figure 6-1. Key terms, and their definitions, pertaining to this taxonomy are:

Data Distinct observed (e.g., measured) values of a physical process. Data may be factual or not, for example they may be subject to uncertainties, such as imprecision in measurement, truncation, and interpretation errors.

Information The result of evaluating, processing, or organizing data/information in a way that adds to knowledge.

Knowledge What is known from gathered information.

Inference The process of obtaining a conclusion based on what one knows.

Examples of data include the **number** of failures during system testing, the **times** at which a component has failed and been repaired, and the **time** it takes until a heating element fails. In these examples, the measured or observed item is bolded to emphasize that data are observable. Note, however, that information is not necessarily observable; only the subset of information that is called data is observable. The availability of data/information, like other types of resources, is crucial to analysis and decision-making. Furthermore, the process of collecting, storing, evaluating, and retrieving data/information affects its organizational value.

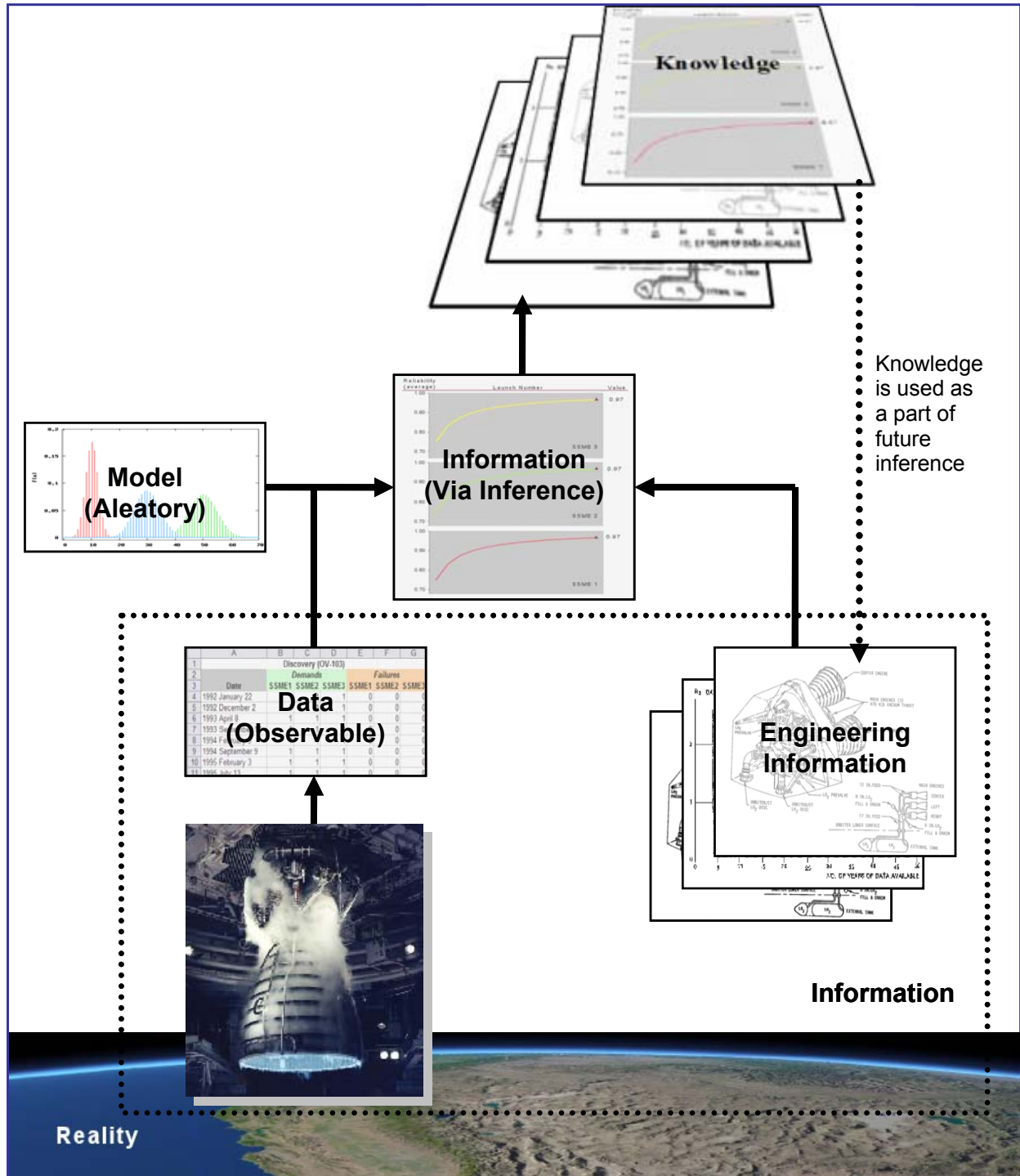


Figure 6-1. Representing the World via Bayesian Inference.

The issue of which interpretation to accept has been debated in the literature and is still unsettled, although, in risk assessments, there has not been a single study that has been based solely on relative frequencies. The practical reason is that the subjective interpretation naturally assigns

(epistemic) probability distributions to the parameters of models. The large uncertainties typically encountered in PRAs make such distributions an indispensable part of the analysis.

The probabilities in both the aleatory and the epistemic models are fundamentally the same and should be interpreted as degrees of belief. This section makes the distinction only for communication purposes. Some authors have proposed to treat probabilities in the aleatory model as limits of relative frequencies, and the probabilities in the epistemic model as subjective. From a conceptual point of view, this distinction is unnecessary and may lead to theoretical problems. In summary:

- Bayesian inference produces information, specifically probabilities related to a hypothesis. Bayesian Inference = Information, where Information = Models + Data + Other Information.
- Probability is a measure of the degree of plausibility of a hypothesis. Probability is evaluated on a 0 to 1 scale.
- Unlike observables such as mass or temperature though, probability – in the objective sense – does not exist (it is not measured, therefore it is never considered data).
- Since probability is subjective, for any hypothesis there is no true value for its associated probability. Furthermore, because model validity is described probabilistically, there is no such thing as a true, perfect, or correct model.

Consider a simple example that will help explain these concepts. Consider again the exponential failure distribution, Equation (6-2). Assume that our epistemic model for the failure rate is the simple discrete model shown in Figure 6-2. There are two possible values of λ , 10^{-2} and 10^{-3} , with corresponding probabilities 0.4 and 0.6. The pmf of the failure rate is:

$$\Pr(\lambda = 10^{-2}) = 0.4 \quad \text{and} \quad \Pr(\lambda = 10^{-3}) = 0.6 \quad (6-5)$$

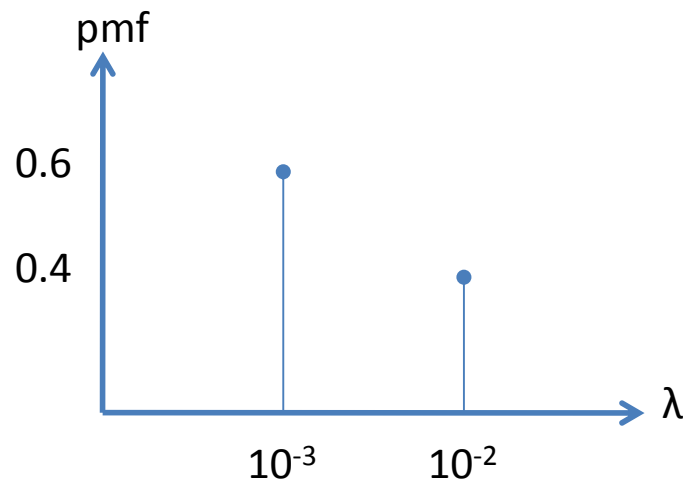


Figure 6-2. The Probability Mass Function (pmf) of the Failure Rate λ .

The reliability of a component for a period of time (0, t) is given by the following pmf:

$$\Pr(e^{-0.001t}) = 0.6 \text{ and } \Pr(e^{-0.01t}) = 0.4 \quad (6-6)$$

One way of interpreting Equation (6-6) is to imagine that a large number of components for a time t are tested. The fraction of components that do not fail will be either $e^{-0.001t}$ with probability 0.6 or $e^{-0.01t}$ with probability 0.4.

Note that, in the frequentist interpretation of probability, there is no place for Equation (6-6), since there is no epistemic model [Equation (6-5)]. One would work with the reliability expression (see Equation (6-2))

$$R(t) = 1 - F(t) = \exp(-\lambda t) \quad (6-7)$$

and the failure rate λ would have an estimated numerical value (see later section on the maximum likelihood method). Note that the explicit notation $F(t|\lambda, M)$ of Equation (6-2) that shows the dependence on λ and M is usually omitted.

6.4 Presentation and Communication of the Uncertainties

A major task of any PRA is to communicate clearly its results to various stakeholders. The simple example of the preceding section can also serve to illustrate the basis for the so-called “risk curves,” which display the uncertainties in the risk results.

Equation (6-6) shows that there are two reliability curves, each with its own probability. These curves are plotted in Figure 6-3.

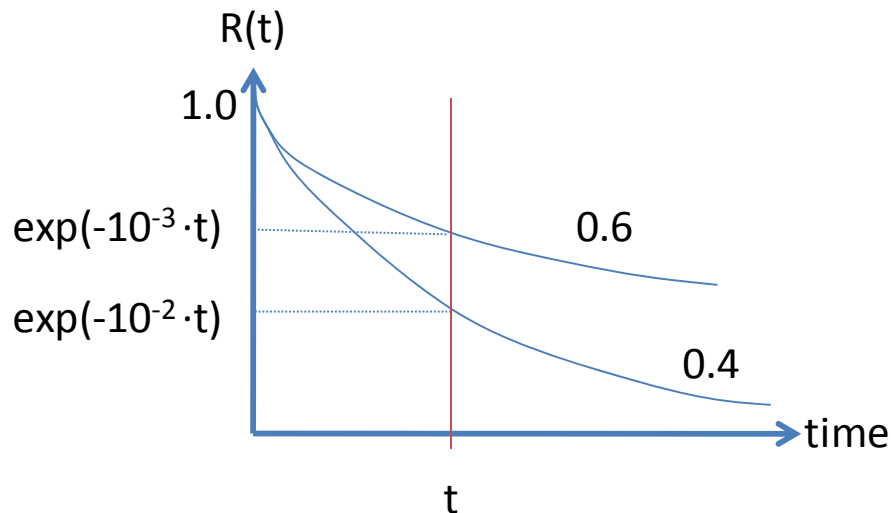


Figure 6-3. Aleatory Reliability Curves with Epistemic Uncertainty.

Figure 6-3 shows the two reliability curves with the two values of the failure rate (one, with probability of 0.6, has a value of 10^{-3} while the second, with probability 0.4, has value 10^{-2}). These curves are, of course, aleatory, since they deal with the *observable* quantity “time.” The epistemic probability is shown for each curve. Thus, for a given time t , the figure shows clearly that there are two possible values of the reliability, each with its own probability.

In this simple example, it is assumed that only two values of the failure rate are possible. In real applications, the epistemic uncertainty about λ is usually expressed using a continuous pdf $\pi(\lambda)$. Then, it is customary to display a family of curves for various percentiles of λ . Figure 6-4 shows three curves with λ being equal to the 10th, 50th, and 90th percentiles of $\pi(\lambda)$. Also shown are three values of the (aleatory) reliability for a given time t' . The interpretation of these values is now different from those in Figure 6-3. For example, we are 0.90 confident that the reliability at t' is greater (not equal to) than $\exp(-\lambda_{90}t')$.

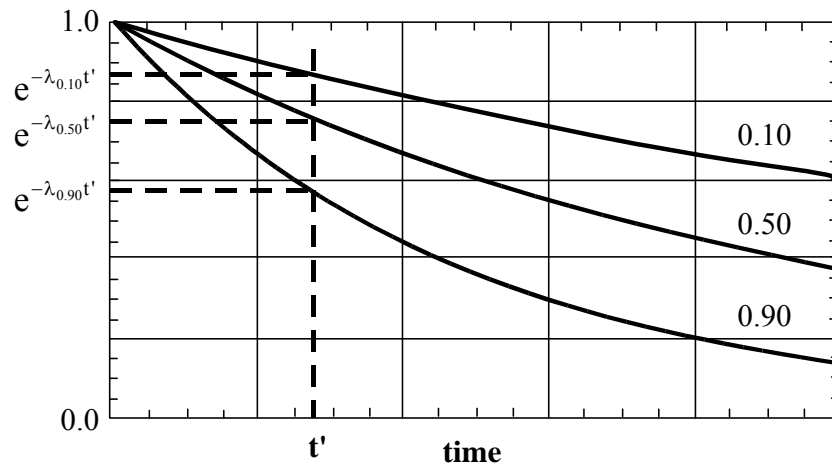


Figure 6-4. Aleatory Reliability Curves with a Continuous Epistemic Distribution.

In addition to the various percentiles, this example calculates the epistemic mean values of aleatory probabilities. These epistemic means are also called *predictive* probabilities. Thus, for the discrete case in Figure 6-3, the predictive reliability is

$$R(t) = 0.6e^{-0.001t} + 0.4e^{-0.01t} \quad (6-8)$$

In the continuous case, the epistemic mean reliability is

$$R(t) = \int e^{-\lambda t} \pi(\lambda) d\lambda \quad (6-9)$$

It is noted that, in the frequentist interpretation, the concept of families of curves does not exist.

6.5 The Lognormal Distribution

The lognormal distribution is used frequently in safety studies as the epistemic distribution of failure rates. The lognormal pdf for λ is given by

$$\pi(\lambda) = \frac{1}{\sqrt{2\pi\sigma\lambda}} \exp\left[-\frac{(\ln \lambda - \mu)^2}{2\sigma^2}\right] \quad (6-10)$$

where $0 < \lambda$; $-\infty < \mu < +\infty$; and $0 < \sigma$. Specifying the numerical values of μ and σ determines the lognormal distribution.

Several characteristic values of the lognormal distribution are

$$\text{mean} = m = \exp\left[\mu + \frac{\sigma^2}{2}\right] \quad (6-11)$$

$$\text{median} = e^{\mu} \quad (6-12)$$

$$\text{95th percentile: } \lambda_{95} = \exp(\mu + 1.645\sigma) \quad (6-13)$$

$$\text{5th percentile: } \lambda_{05} = \exp(\mu - 1.645\sigma) \quad (6-14)$$

$$\text{Error Factor} = \frac{\lambda_{50}}{\lambda_{05}} = \frac{\lambda_{95}}{\lambda_{50}} = e^{1.645\sigma} \quad (6-15)$$

The random variable λ has a lognormal distribution, if its logarithm follows a normal distribution with mean μ and standard deviation σ . This allows the use of tables of the normal distribution.

For example, the 95th percentile of the normal variable $\ln\lambda$ is

$$\ln \lambda_{95} = \mu + 1.645\sigma \quad (6-16)$$

where the factor 1.645 comes from tables of the normal distribution. Equation (6-13) follows from Equation (6-16).

The shape of the lognormal pdf is shown in Figure 6-5.

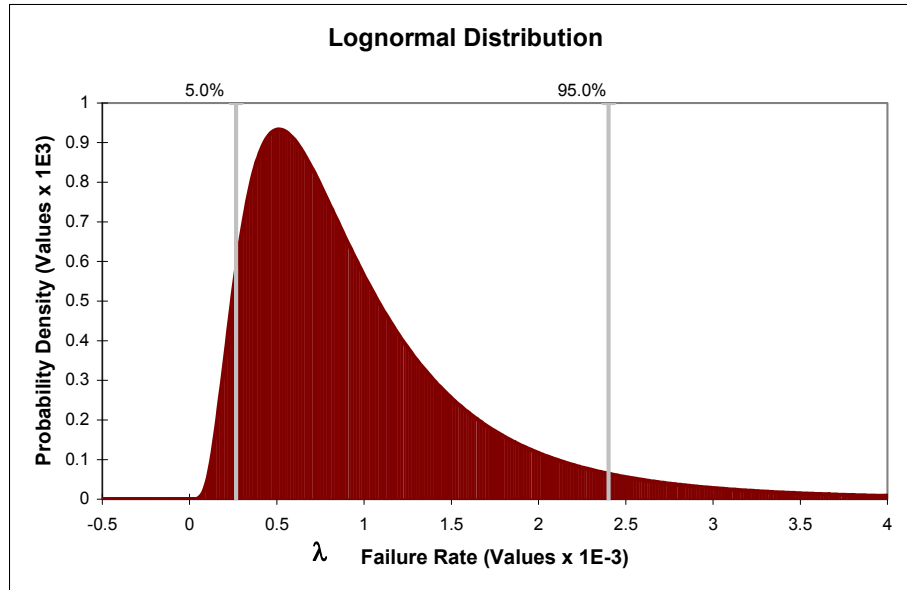


Figure 6-5. The Lognormal probability density function (pdf).

The distribution is skewed to the right. This (in addition to its analytical simplicity) is one of the reasons why it is chosen often as an epistemic distribution for failure rates. It allows high values of λ that may represent extreme environments.

6.6 Assessment of Epistemic Distributions

When evidence E becomes available, it is natural to change the epistemic models shown here to reflect this new knowledge. The typical problem encountered in practice involves an aleatory model of the world. The evidence is in the form of statistical observations. The analytical tool for changing (“updating”) our epistemic distributions of the parameters of the aleatory model is Bayes’ Theorem.

6.6.1 Bayes’ Theorem

The rule of conditional probabilities gives the conditional probability of an event A given that we have received evidence E as

$$\Pr(A|E) = \Pr(A) \frac{\Pr(E|A)}{\Pr(E)} \quad (6-17)$$

Equation (6-17) shows how the “prior” probability $\Pr(A)$, the probability of A prior to receiving E , is modified to give the “posterior” probability $\Pr(A|E)$, subsequent to receiving E . The likelihood function $\Pr(E|A)$ demands that the probability of this evidence be evaluated, assuming that the event A is true. Equation (6-17) is the basis for Bayes’ Theorem, which is so fundamental to the subjectivistic theory that this theory is sometimes referred to as the Bayesian theory of probability.

Consider an aleatory model of the world that contains a parameter θ . An example is the exponential distribution of Equation (6-2) with parameter λ . This example will distinguish between a discrete and a continuous epistemic model.

In the discrete case, θ is assumed to have the pmf $\Pr(\theta_i) = p_i$, $i = 1, \dots, n$, where n is the total number of possible values of θ . In this case, Equation (6-17) leads to the discrete form of Bayes' Theorem:

$$\Pr(\theta_i|E) = \Pr(\theta_i) \frac{L(E|\theta_i)}{\sum_1^n \Pr(\theta_i)L(E|\theta_i)} \quad (6-18)$$

or

$$p'_i = p_i \frac{L(E|\theta_i)}{\sum_1^n L(E|\theta_i)p_i} \quad (6-19)$$

where the primed quantity is the posterior probability.

In the continuous case, this example has

$$\pi'(\theta|E) = \frac{L(E|\theta)\pi(\theta)}{\int L(E|\theta)\pi(\theta)d\theta} \quad (6-20)$$

Note that the evaluation of the likelihood function requires the use of the aleatory model.

6.6.2 A Simple Example: The Discrete Case

Consider again the simple example of Equation (6-5). Suppose that the evidence is: 5 components were tested for 100 hours and no failures were observed. Since the reliability of each component is $\exp(-100\lambda)$, the likelihood function is

$$L(E|\lambda) = \prod_1^5 e^{-100\lambda} = e^{-500\lambda} \quad (6-21)$$

Note that the aleatory model, Equation (6-2), is indeed used to derive this expression. The question now is: how should the prior epistemic probabilities of Equation (6-5) be updated to reflect this evidence? Since the epistemic model is discrete in this case, this example uses Equation (6-19) (here, λ is the parameter θ). The calculations required by Equation (6-19) are shown in Table 6-1.

Table 6-1. Bayesian Calculations for the Simple Example (No Failures).

Failure Rate	Prior probability p_i	Likelihood	(prior) x (likelihood)	Posterior probability (p'_i)
0.001 hr ⁻¹	0.6	0.60653	0.36391	0.99267
0.01 hr ⁻¹	0.4	0.00673	0.00269	0.00733
	Sum = 1.0		Sum = 0.36660	Sum = 1.00000

The likelihood functions are calculated using Equation (6-21) and the failure rates of the first column. The posterior probabilities are simply the normalized products of the fourth column, e.g., $0.36391/0.36660 = 0.99267$.

Note the dramatic impact of the evidence. The posterior epistemic probability of the failure rate value of 0.001 hr^{-1} is 0.99267, while the prior probability of this value was 0.60.

To appreciate the impact of different kinds of evidence, assume that one failure was actually observed at 80 hours during this test. For each of the surviving components, the contribution to the likelihood function is $\exp(-100\lambda)$ for a total of $\exp(-400\lambda)$. For the failed component, the probability of failure at 80 hours is given in terms of the failure density, i.e., $80\lambda \exp(-80\lambda)dt$. Note that the factor dt appears in the denominator of Equation (6-19) also, so it is not carried. Thus, the new likelihood function is the product of these contributions, i.e.,

$$L(E|\lambda) = 80\lambda e^{-480\lambda} \quad (6-22)$$

With this new likelihood function, Table 6-1 is modified as shown in Table 6-2.

Table 6-2. Bayesian Calculations for the Simple Example with New Evidence (One Failure).

Failure Rate	Prior Probability p_i	Likelihood	(prior) x (likelihood)	Posterior Probability (p_i')
0.001 hr^{-1}	0.6	0.04852	0.04950	0.88266
0.01 hr^{-1}	0.4	0.00538	0.00658	0.11734
	Sum = 1.0		Sum = 0.05608	Sum = 1.00000

Note that the fact that one failure occurred has reduced the posterior probability of the failure rate value of 0.001 hr^{-1} from 0.99267 to 0.88266. In both cases, however, the evidence is strongly in favor of this value of the failure rate.

6.6.3 A Simple Example: The Continuous Case

Very often, a continuous distribution is used for the parameter of interest. Thus, for the failure rate of our simple example, assume a lognormal prior distribution with a median value of $3 \times 10^{-3} \text{ hr}^{-1}$ and a 95th percentile of $3 \times 10^{-2} \text{ hr}^{-1}$, i.e., an error factor of 10 is assumed. The lognormal density function is given in Equation (6-10).

Using the given information, two equations for the parameters μ and σ are used:

$$\lambda_{50} = \exp(\mu) = 3 \times 10^{-3} \text{ hr}^{-1} \quad (6-23)$$

$$\lambda_{95} = \exp(\mu + 1.645\sigma) = 3 \times 10^{-2} \text{ hr}^{-1} \quad (6-24)$$

Solving Equations (6-23) and (6-24) yields $\mu = -5.81$ and $\sigma = 1.40$. The mean value is

$$E[\lambda] = \exp\left(\mu + \frac{\sigma^2}{2}\right) = 8 \times 10^{-3} \text{ hr}^{-1} \quad (6-25)$$

and the 5th percentile

$$\lambda_{05} = \exp(\mu - 1.645\sigma) = 3 \times 10^{-4} \text{ hr}^{-1} \quad (6-26)$$

It is evident that the calculations of Equation (6-20) with the prior distribution given by Equation (6-10) and the likelihood function by Equation (6-21) or (6-22) will require numerical methods. This will require the discretization of the prior distribution and the likelihood function.

Consider the following distribution (pdf), $\pi(\lambda)$, of the continuous variable λ (not necessarily the lognormal distribution).

If one wishes to get a discrete approximation to $\pi(\lambda)$, it can be done simply by carving λ up into the intervals as shown in Figure 6-6. The idea is to assign the probability that λ will fall in an interval $(\lambda_{j-1}, \lambda_j)$ to a single point λ_j^* inside that interval. This probability, say P_j , is simply:

$$P_j = \int_{\lambda_{j-1}}^{\lambda_j} \pi(\lambda) d\lambda \quad (6-27)$$

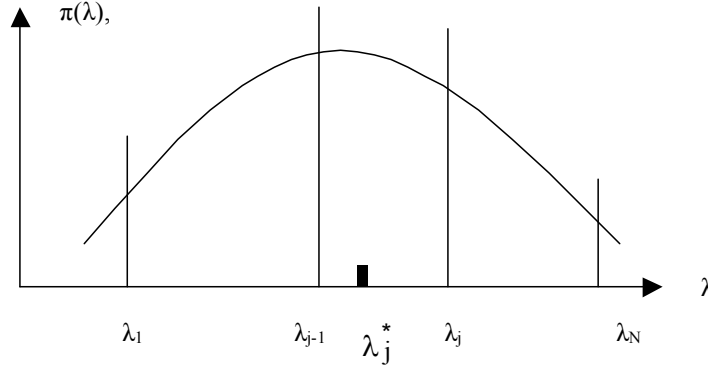


Figure 6-6. Discretization Scheme.

The points λ_j^* can be determined in various ways. For example, λ_j^* can be the mean value of the points in each interval. Thus, with the understanding that $\lambda_0 = 0$ and $\lambda_{N+1} = \infty$, it is determined:

$$\lambda_j^* = \frac{1}{P_j} \int_{\lambda_{j-1}}^{\lambda_j} \lambda \pi(\lambda) d\lambda \quad (6-28)$$

with $j = 1 \dots N$.

A second method is to simply take λ_j^* as the arithmetic midpoint of the interval, i.e.,

$$\lambda_j^* = \frac{\lambda_j + \lambda_{j-1}}{2} \quad (6-29)$$

A third method, which is appropriate for the lognormal distribution, is to take λ_j^* as the geometric midpoint of the interval, i.e.,

$$\lambda_j^* = \sqrt{\lambda_j \lambda_{j-1}} \quad (6-30)$$

The reason why Equation (6-30) is appropriate for the lognormal distribution is that the range of λ is usually very wide. Note that, in using Equations (6-29) and (6-30), this example cannot use the values $\lambda_0 = -\infty$ and $\lambda_{N+1} = \infty$. However, it will be satisfactory to pick λ_0 and λ_{N+1} so that the probability that λ falls outside the interval $(\lambda_0, \lambda_{N+1})$ will be negligibly small.

It is evident that the accuracy of the discretization increases as the number of intervals increases (i.e., for N large). The intervals do not have to be of equal length. Special care should be taken when the pdf has a long “high” tail.

In this example, we used 700 points, i.e., $N = 700$.

Using Equation (6-21), evidence with no failures, as the likelihood function, we find a posterior histogram with the following characteristic values:

$$\lambda'_{05} = 1.5 \times 10^{-4} \text{ hr}^{-1} \quad (6-31)$$

$$\lambda'_{50} = 9 \times 10^{-4} \text{ hr}^{-1} \quad (6-32)$$

$$\lambda'_{95} = 3.7 \times 10^{-3} \text{ hr}^{-1} \quad (6-33)$$

$$E'(\lambda) = 1.3 \times 10^{-3} \text{ hr}^{-1} \quad (6-34)$$

The impact of the evidence has, again, been the shifting of the epistemic distribution toward lower values of the failure rate. Thus, the mean moved from $8 \times 10^{-3} \text{ hr}^{-1}$ (Equation (6-25)) to $1.3 \times 10^{-3} \text{ hr}^{-1}$ (Equation (6-34)), and the median from $3 \times 10^{-3} \text{ hr}^{-1}$ (Equation (6-23)) to $9 \times 10^{-4} \text{ hr}^{-1}$ (Equation (6-32)). The most dramatic impact is on the 95th percentile, from $3 \times 10^{-2} \text{ hr}^{-1}$ (Equation (6-24)) to $3.7 \times 10^{-3} \text{ hr}^{-1}$ (Equation (6-33)). The prior and posterior distributions are shown in Figure 6-7. Note that these are not pdfs but histograms. This example has connected the tips of the vertical histogram bars for convenience in displaying the results. The shift of the epistemic distribution toward lower values of the failure rate is now evident.

Note that for the example described above, a numerical approximation was used to determine a posterior distribution. With modern software, many difficult calculations can now be performed that were previously intractable. For more information on these approaches, see Reference [6-13].

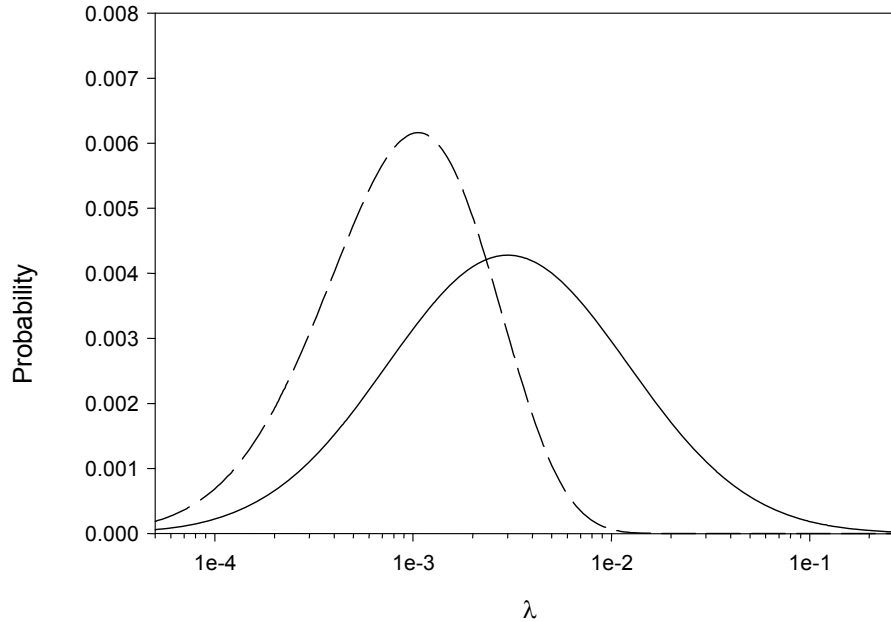


Figure 6-7. Prior (Solid Line) and Posterior (Dashed Line) Probabilities for the Case of No Failures.

6.6.4 Conjugate Families of Distributions

The previous section has already shown that Equation (6-20) requires, in general, numerical computation. It discretized both the lognormal prior distribution and the exponential distribution of the model of the world in order to produce the posterior distribution.

It turns out that, for a given model of the world, there exists a family of distributions with the following property. If the prior distribution is a member of this family, then the posterior distribution will be a member of the same family and its parameters will be given by simple expressions. These families of distributions are called *conjugate* distributions.

As an example, the conjugate family with respect to the exponential model of the world is the gamma distribution whose pdf is

$$\pi(\lambda) = \frac{\beta^\alpha \lambda^{\alpha-1}}{\Gamma(\alpha)} e^{-\beta\lambda} \quad (6-35)$$

where α and β are the two parameters of the distribution and $\Gamma(\alpha)$ is the gamma function. For integer values of α , we have $\Gamma(\alpha) = (\alpha - 1)!$ The mean and standard deviation of this distribution are

$$E[\lambda] = \frac{\alpha}{\beta} \quad \text{and} \quad SD[\lambda] = \frac{\sqrt{\alpha}}{\beta} \quad (6-36)$$

Suppose now that one has the following failure times of n components: t_1, t_2, \dots, t_r , with $r < n$. This means that one has the failure times of r components and that $(n-r)$ components did not fail. Define the total operational time T as:

$$T \equiv \sum_1^r t_i + (n-r)t_r \quad (6-37)$$

Then Bayes' Theorem, Equation (6-20), shows that the posterior distribution is also a gamma distribution with parameters

$$\alpha' = \alpha + r \quad \text{and} \quad \beta' = \beta + T \quad (6-38)$$

These simple relations between the prior and posterior parameters are the advantage of the conjugate distributions. However, with the availability of modern Bayesian analysis software, the need for simplifying expressions for distribution evaluation has diminished.

Returning to the simple example, assume that the prior distribution for λ is gamma with the same mean and standard deviation as the lognormal distribution that were used in the preceding section. Then, the parameters α and β will be determined by solving Equation (6-36), i.e.,

$$E[\lambda] = \frac{\alpha}{\beta} = 8 \times 10^{-3} \quad \text{and} \quad SD[\lambda] = \frac{\sqrt{\alpha}}{\beta} = 1.98 \times 10^{-2} \quad (6-39)$$

Thus, the two parameters are: $\alpha = 0.16$ and $\beta = 20$. For the evidence of one failure at 80 hours and no failures for 400 hours (see Equation (6-22)), $T = 480$ and $r = 1$; therefore, from Equation (6-38), $\alpha' = 1.16$ and $\beta' = 500$. The new mean and standard deviation of the epistemic (posterior) distribution of λ are:

$$E'[\lambda] = \frac{\alpha'}{\beta'} = \frac{\alpha + r}{\beta + T} = \frac{0.16 + 1}{20 + 480} = 2.32 \times 10^{-3} \text{ hr}^{-1} \quad (6-40)$$

and

$$SD'[\lambda] = \frac{\sqrt{\alpha'}}{\beta'} = 2.15 \times 10^{-3} \text{ hr}^{-1} \quad (6-41)$$

As expected, the evidence has reduced the mean value of the failure rate. It has also reduced the standard deviation.

For the evidence of 0 failures in 500 hours, Equation (6-21), $r = 0$ and $T = 500$; thus,

$$E'[\lambda] = \frac{\alpha'}{\beta'} = \frac{\alpha + r}{\beta + T} = \frac{0.16 + 0}{20 + 500} = 3.07 \times 10^{-4} \quad (6-42)$$

and

$$SD'[\lambda] = \frac{\sqrt{\alpha'}}{\beta'} = \frac{\sqrt{0.16 + 0}}{20 + 500} = 7.7 \times 10^{-4} \quad (6-43)$$

Conjugate distributions for other models can be found in the literature [6-9, 6-13].

6.7 The Prior Distribution

This chapter has introduced the epistemic model and has shown how Bayes' Theorem is used to update it when new evidence becomes available. The question that arises now is: how does one develop the prior epistemic distribution? Saying that it should reflect the assessor's state of knowledge is not sufficient. In practice, the analyst must develop a prior distribution from available engineering and scientific information, where the prior should:

- Reflect what information is known about the inference problem at hand, and
- Be independent of the data that is collected.

An assessor of probabilities must be knowledgeable both of the subject to be analyzed and of the theory of probability. The normative "goodness" of an assessment requires that the assessor does not violate the calculus of probabilities, and that he or she makes assessments that correspond to his or her judgments. The substantive "goodness" of an assessment refers to how well the assessor knows the problem under consideration. It is not surprising that frequently one or the other kind of "goodness" is neglected, depending on who is doing the analysis and for what purpose. The fact that safety studies usually deal with events of low probability makes them vulnerable to distortions that eventually may undermine the credibility of the analysis.

Direct assessments of model parameters, like direct assessments of failure rates, should be avoided, because model parameters are not directly observable. The same observation applies to moments of distributions, for example, the mean and standard deviation. Intuitive estimates of the mode or median of a distribution have been found to be fairly accurate, whereas estimates of the mean tend to be biased toward the median. This has led to the suggestion that "best" estimates or "recommended" values, which are often offered by engineers, be used as medians. In assessing rare-event frequencies, however, the possibility of a systematic underestimation or overestimation ["displacement bias"], even of the median, is very real.

Further, assessors tend to produce distributions that are too narrow. In assessing the frequency of accidents in industrial facilities, it is also conceivable that this "variability bias" could actually manifest itself in the opposite direction; that is, a very conservative assessor could produce a distribution that is broader than his or her state of knowledge would justify.

These observations about the accuracy of judgments are important both when one quantifies his or her own judgment and when he or she elicits the opinions of experts.

The practice of eliciting and using expert opinions became the center of controversy with the publication of a major risk study of nuclear power plants (NPPs). This study considered explicitly alternate models for physical phenomena that are not well understood and solicited the help of experts to assess the probabilities of the models. Objections were raised both to the use of expert opinions (with complaints that voting is replacing experimentation and hard science) and to the process of using expert opinions (for example, the selection of the experts). The latter criticism falls outside the mathematical theory that we have been discussing and is not of interest here; however, the view that voting replaces hard science is misguided. The (epistemic) probabilities of models are an essential part of the decision-making process. Unfortunately, many decisions cannot wait until such evidence becomes available, and assessing the model probabilities from expert opinions is a necessity. (Incidentally, such an assessment may lead to the decision to do nothing until experiments are conducted.)

More details on the utilization of expert judgment can be found in References 6-10 through 6-12. In the NASA Bayesian guide (Reference [6-13]), guidance is provided for prior development. In practice, the analyst must develop a prior distribution from available

engineering and scientific information, where the prior should reflect (a) what information is known about the inference problem at hand and (b) be independent of the data that is collected.

Frequently, beta and gamma distributions are used as conjugate priors. Therefore, two pieces of information are generally needed to select such a conjugate prior. Common information from which the analyst must develop a prior is:

1. A central measure (e.g., median or mean) and an upper bound (e.g., 95th percentile)
2. Upper and lower bound (e.g., 95th and 5th percentile)
3. A mean and variance (or standard deviation).

In some cases, not enough information may be available to completely specify an informative prior distribution, as two pieces of information are typically needed. For example, in estimating a failure rate, perhaps only a single estimate is available. Because the information on which the prior is based may be limited, the resulting prior distribution will be diffuse, encoding significant epistemic uncertainty about the parameter value. The table below summarizes the results for commonly encountered cases.

Information Available	Suggested Prior Distribution
Mean value for lambda in Poisson distribution	Gamma distribution with alpha = 0.5 and beta = 1/(2 × mean)
Mean value for p in binomial distribution	Beta distribution with alpha ≈ 0.5 and beta = (1 – mean)/(2 × mean)
Mean value for lambda in exponential distribution	Gamma distribution with alpha = 1 and beta = 1/mean
p in binomial distribution lies between a and b	Uniform distribution between a and b

6.8 The Method of Maximum Likelihood

The methods for data analysis that have been presented so far are within the framework of the subjective interpretation of probability. The central analytical tool for the updating of this chapter's epistemic model, i.e., the state of knowledge, is Bayes' Theorem. These methods are also called Bayesian methods.

If one adopts the frequentist interpretation of probability, then one is not allowed to use epistemic models. The numerical values of the parameters of the model of the world must be based on statistical evidence only. A number of methods have been developed for producing these numerical values.

A widely used method for producing *point estimates* of the parameters is the method of maximum likelihood. The likelihood function is formed based on the data exactly as they are formed for a Bayesian calculation. Instead of using Bayes' Theorem, however, this example considers the likelihood function as a function of the parameters and finds the values of the parameters that maximize this function. These parameter values are, then, called their maximum likelihood estimates (MLE).

To make the discussion concrete, this section uses Equation (6-22) as an example. To find the maximum, differentiate, i.e.,

$$\frac{dL}{d\lambda} = 80e^{-480\lambda} - 80 \times 480\lambda e^{-480\lambda} = 0 \quad (6-44)$$

Solving Equation (6-44) yields $\lambda_{MLE} = \frac{1}{480} = 0.025 \text{ hr}^{-1}$. More generally, for a total operational time T and r failures, the estimate of the failure rate is

$$\lambda_{MLE} = \frac{r}{T} \quad (6-45)$$

Note that for the first example (no failures in $T = 500$ hrs), $r = 0$ and Equation (6-45) gives the unrealistic estimate of zero. In contrast, the Bayesian posterior mean value was $3.07 \times 10^{-4} \text{ hr}^{-1}$ (Equation (6-42)).

Equations (6-40) and (6-45) lead to an interesting observation. One can get Equation (6-45) from Equation (6-40) by simply setting the parameters of the prior distribution α and β equal to zero. Thus, in Bayesian calculations, when one wishes to “let the data speak for themselves,” one can use a beta distribution with these parameter values. Then, the posterior distribution will be determined by the data alone. Prior distributions of this type are called *non-informative* [6-12].

There is a more general message in this observation that can actually be proved theoretically. As the statistical evidence becomes stronger, i.e., as r and T become very large, the Bayesian posterior distribution will tend to have a mean value that is equal to the MLE. In other words, any prior beliefs will be overwhelmed by the statistical evidence.

6.9 References

- 6-1 L.J. Savage, *The Foundations of Statistics*, Dover Publications, New York, 1972.
- 6-2 G.E. Apostolakis, “A Commentary on Model Uncertainty,” in: *Proceedings of Workshop on Model Uncertainty: Its Characterization and Quantification*, A. Mosleh, N. Siu, C. Smidts, and C. Lui, Eds., Annapolis, MD, October 20-22, 1993, Center for Reliability Engineering, University of Maryland, College Park, MD, 1995.
- 6-3 M.E. Paté-Cornell, “Uncertainties in Risk Analysis: Six Levels of Treatment,” *Reliability Engineering and System Safety*, 54, 95-111, 1996.
- 6-4 R.L. Winkler, “Uncertainty in Probabilistic Risk Assessment,” *Reliability Engineering and System Safety*, 54, 127-132, 1996.
- 6-5 G. Apostolakis and S. Kaplan, “Pitfalls in Risk Calculations,” *Reliability Engineering*, 2, 135-145, 1981.
- 6-6 G.W. Parry, “The Characterization of Uncertainty in Probabilistic Risk Assessments of Complex Systems,” *Reliability Engineering and System Safety*, 54, 119-126, 1996.
- 6-7 G. Apostolakis, “The Distinction between Aleatory and Epistemic Uncertainties is Important: An Example from the Inclusion of Aging Effects into PSA,” *Proceedings of PSA '99, International Topical Meeting on Probabilistic Safety Assessment*, pp. 135-142, Washington, DC, August 22 - 26, 1999, American Nuclear Society, La Grange Park, IL.
- 6-8 B. De Finetti, *Theory of Probability*, Vols. 1 and 2, Wiley, NY, 1974.

- 6-9 A. H-S. Ang and W.H. Tang, *Probability Concepts in Engineering Planning and Design*, vol. 1, Wiley, 1975.
- 6-10 R.L. Keeney and D. von Winterfeldt, "Eliciting Probabilities from Experts in Complex Technical Problems," *IEEE Transactions on Engineering Management*, 38, 191-201, 1991.
- 6-11 S. Kaplan, "Expert Information vs. Expert Opinions: Another Approach to the Problem of Eliciting/Combining/Using Expert Knowledge in PRA," *Reliability Engineering and System Safety*, 25, 61-72, 1992.
- 6-12 T. Bedford and R. Cooke, *Probabilistic Risk Analysis*, Cambridge University Press, UK, 2001.
- 6-13 NASA-SP-2009-569, *Bayesian Inference for NASA Probabilistic Risk and Reliability Analysis*, <http://www.hq.nasa.gov/office/codeq/doctree/SP2009569.htm>, 2009.

7. Modeling and Quantification of Common Cause Failures

7.1 Importance of Dependence in PRA

The significant risk contributors are typically found at the interfaces between components, subsystems, systems, and the surrounding environment. Risk drivers emerge from aspects in which one portion of the design depends on, or interacts with, another portion, or the surrounding environment. Failures arising from dependencies are often difficult to identify and, if neglected in PRA modeling and quantifications, may result in an underestimation of the risk. This chapter provides an overview of the various types of dependencies typically encountered in PRA of engineered systems and discusses how such dependencies can be treated. The focus of the discussion will be on a special class of dependent failures known as Common Cause Failures (CCF).

7.2 Definition and Classification of Dependent Events

Two events, A and B, are said to be dependent if

$$\Pr(A \cap B) \neq \Pr(A) \Pr(B) \quad (7-1)$$

In the presence of dependencies, often, but not always, $\Pr(A \cap B) > \Pr(A) \Pr(B)$. Therefore, if A and B represent failure of a function, the actual probability of failure of both will be higher than the expected probability calculated based on the assumption of independence. In cases where a system provides multiple layers of defense against total system or functional failure, ignoring the effects of dependency can result in overestimation of the level of reliability.

Dependencies can be classified in many different ways. A classification, which is useful in relating operational data to reliability characteristics of systems, is presented in the following paragraphs [7-1]. In this classification, dependencies are first categorized based on whether they stem from intended functional and physical characteristics of the system, or are due to external factors and unintended characteristics. Therefore, dependence is either *intrinsic* or *extrinsic* to the system. The definitions and sub-classifications follow.

Intrinsic. This refers to dependencies where the functional state of one component is affected by the functional state of another. These dependencies normally stem from the way the system is designed to perform its intended function. There are several subclasses of intrinsic dependencies based on the type of influence that components have on each other. These are:

- **Functional Requirement Dependency.** This refers to the case where the functional status of component A determines the functional requirements of component B. Possible cases include:
 - B is not needed when A works,
 - B is not needed when A fails,
 - B is needed when A works,
 - B is needed when A fails.

Functional requirement dependency also includes cases where the load on B is increased upon failure of A.

- **Functional Input Dependency (or Functional Unavailability).** This is the case where the functional status of B depends on the functional status of A. An example is the case where A must work for B to work. In other words B is functionally unavailable as long as A is not working. An example is the dependence of a motor-driven pump on electric power. Loss of electric power makes the pump functionally unavailable. Once electric power becomes available, the pump will also be operable.
- **Cascade Failure.** This refers to the cases where failure of A leads to failure of B. For example, an over-current failure of a power supply may cause the failure of components it feeds. In this case, even if the power supply is made operable, the components would still remain inoperable.

Combinations of the above dependencies identify other types of intrinsic dependencies. An example is the *Shared Equipment Dependency*, when several components are functionally dependent on the same component. For example if both B and C are functionally dependent on A, then B and C have a shared equipment dependency.

Extrinsic. This refers to dependencies that are not inherent and intended in the designed functional characteristics of the system. Such dependencies are often physically external to the system. Examples of extrinsic dependencies are

- **Physical/Environmental.** This category includes dependencies due to common environmental factors, including a harsh or abnormal environment created by a component. For example, high vibration induced by A causes failure of B.
- **Human Interactions.** This is dependency due to human-machine interaction. An example is failure of multiple components due to the same maintenance error.

7.3 Accounting for Dependencies in PRAs

PRA analysts generally try to include the intrinsic dependencies in the basic system logic model (e.g., FTs). For example, functional dependencies arising from the dependence of systems on electric power are included in the logic model by including basic events, which represent component failure modes associated with failures of the electric power supply system. Failures resulting from the failure of another component (cascading or propagating failures) are also often modeled explicitly. Operator failures to respond in the manner called for by the operating procedures are included as branches on the ETs or as basic events on FTs. Some errors made during maintenance are usually modeled explicitly on FTs, or they may be included as contributors to overall component failure probabilities.

Extrinsic dependencies can be treated through modeling of the phenomena and the physical processes involved. Examples are the effects of temperature, humidity, vibration, radiation, etc., in the category of Physical/Environmental dependencies. A key feature of the so-called “external events” is the fact that they can introduce dependencies among PRA basic events. Explicit treatment of the external events such as fire and micro-meteoroid and orbital debris (MMOD) may be a significant portion of a PRA study. (See Chapter 14.)

The logic model constructed initially has basic events that for a first approximation are considered independent. This step is necessary to enable the analyst to construct manageable models. As such, many extrinsic and some intrinsic dependencies among component failures are typically not accounted for explicitly in the PRA logic models, meaning that some of the corresponding basic events are not actually independent. Dependent failures whose root causes are not explicitly modeled in PRA are known as CCFs. This category can be accounted for by introducing common cause basic events (CCBE) in the PRA logic models. A formal definition follows:

Common Cause Failure event is defined as the failure (or unavailable state) of more than one component due to a shared cause during the system mission. Viewed in this fashion, CCFs are inseparable from the class of dependent failures and the distinction is mainly based on the level of treatment and choice of modeling approach in reliability analysis.

Components that fail due to a shared cause normally fail in the same functional mode. The term “common mode failure,” which was used in the early literature and is still used by some practitioners, is more indicative of the most common symptom of the CCF, i.e., failure of multiple components in the same mode, but it is not a precise term for communicating the important characteristics that describe a CCF event.

The following are examples of actual CCF events:

- Hydrazine leaks leading to two APU explosions on Space Shuttle mission STS-9
- Multiple engine failures on aircraft (Fokker F27 –1997, 1988; Boeing 747, 1992)
- Three hydraulic system failures following Engine # 2 failure on a DC-10, 1989
- Failure of all three redundant auxiliary feed-water pumps at Three Mile Island NPP
- Failure of two Space Shuttle Main Engine (SSME) controllers on two separate engines when a wire short occurred
- Failure of two O-rings, causing hot gas blow-by in a solid rocket booster of Space Shuttle flight 51L
- Failure of two redundant circuit boards due to electro-static shock by a technician during replacement of an adjacent unit
- A worker accidentally tripping two redundant pumps by placing a ladder near pump motors to paint the ceiling at a nuclear power plant
- A maintenance contractor unfamiliar with component configuration putting lubricant in the motor winding of several redundant valves, making them inoperable
- Undersized motors purchased from a new vendor causing failure of four redundant cooling fans
- Check valves installed backwards, blocking flow in two redundant lines

CCFs may also be viewed as being caused by the presence of two factors: a Root Cause, i.e., the reason (or reasons) for failure of each component that failed in the CCF event, and a Coupling Factor (or factors) that was responsible for the involvement of multiple components. For example, failure of two identical redundant electronic devices due to exposure to excessively high temperatures is not only the result of susceptibility of each of the devices to heat (considered to be the root cause in this example), but also a result of both units being identical, and being exposed to the same harsh environment (Coupling Factor). Since the use of identical components in redundancy formation is a common strategy to improve system reliability, coupling factors stemming from similarities of the redundant components are often present in such redundant formations, leading to vulnerability to CCF events. CCF events of identical redundant components therefore merit special attention in risk and reliability analysis of such systems. The remainder of this chapter is devoted to methods for modeling the impact of these CCF events.

7.4 Modeling Common Cause Failures

Proper treatment of CCFs requires identifying those components that are susceptible to CCFs and accounting for their impact on the system reliability. The oldest, and one of the simplest methods for modeling the impact of CCFs, is the beta-factor model [7-2].

To illustrate the way beta factor treats CCFs, consider a simple redundancy of two identical components B1 and B2. Each component is further divided into an “independently failing” component and one that is affected by CCFs only (see Figure 7-1). The figure also shows reliability models of the redundant system in FT and reliability block diagram formats. The beta-factor further assumes that

Total component failure frequency = (Independent failure frequency) + (Common cause failure frequency)

A factor, β , is then defined as:

$$\beta = \frac{\lambda_C}{\lambda_T}$$

$$\lambda_C = \beta\lambda_T \quad (\text{common cause failure frequency}) \quad (7-2)$$

$$\lambda_I = (1 - \beta)\lambda_T \quad (\text{independent failure frequency})$$

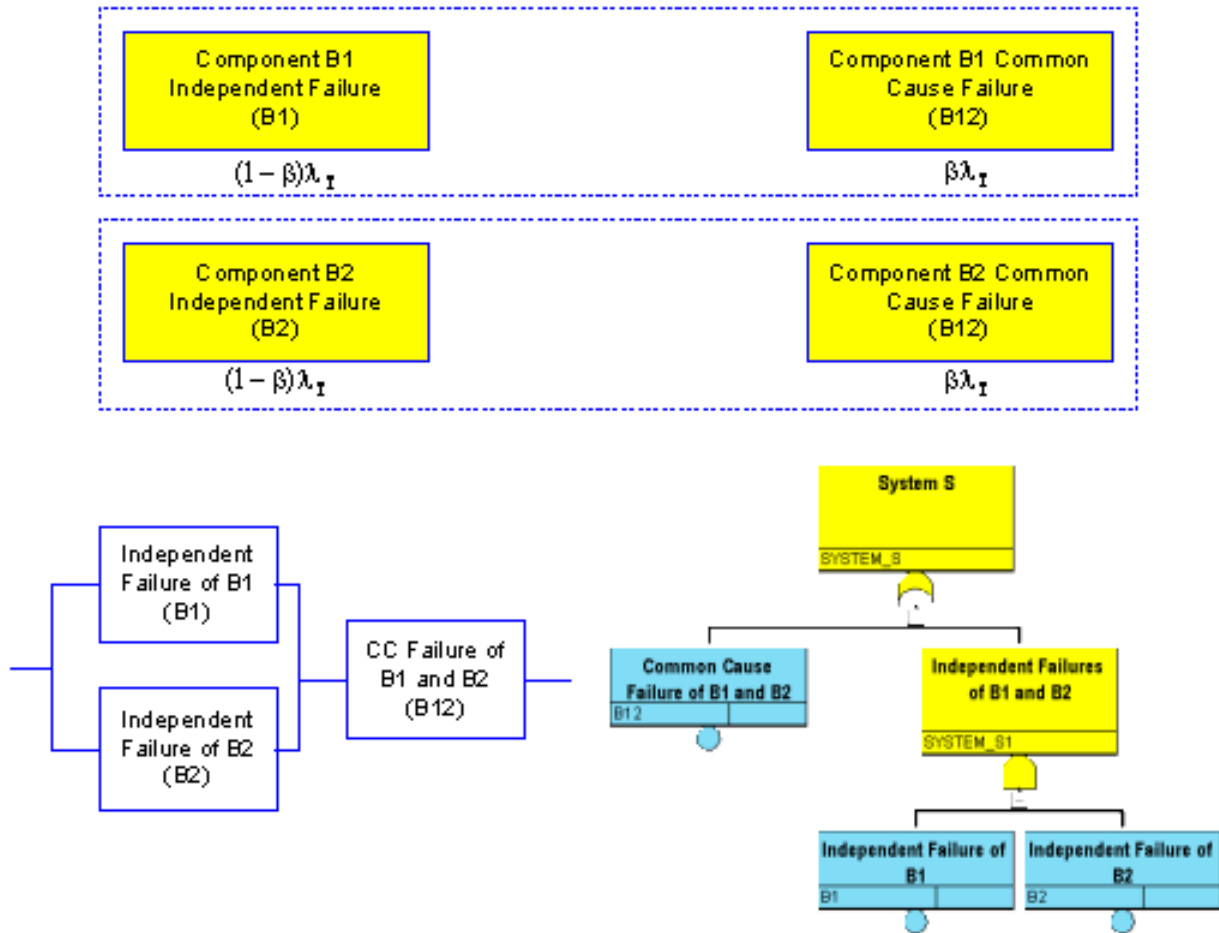


Figure 7-1. Accounting for CCF Events Using the Beta Factor Model in Fault Trees and Reliability Block Diagrams.

Failure probability of the two-unit parallel system of B1 and B2 is then calculated as

$$Q_s = (\lambda_{IT})^2 + (\lambda_{CT}) = [(1 - \beta)\lambda_{IT}]^2 + \beta\lambda_{IT} \quad (7-3)$$

where λt is an approximation for the exponential failure probability model.

A point estimate for beta is given by

$$\beta = \frac{2n_2}{n_1 + 2n_2} \quad (7-4)$$

Where:

n_1 = number of independent failures

n_2 = number of CCFs. Samples of failure events are then used to obtain values of n_1 and n_2 for the specific component of interest. The resulting beta factor value, together with the

total failure rate, λ_T , of the identical redundant components, is then used to calculate the reliability of the redundant formation in the presence of CCF events.

As we can see in the following sections, a generalization of this simple approach forms the basis of a methodology for treating CCF events in PRA models.

7.5 Procedures and Methods for Treating CCF Events

The process of identifying and modeling CCFs in systems analysis involves two important steps:

1. Screening Analysis
2. Detailed Analysis

The objectives of the Screening Analysis are to identify in a preliminary and conservative manner all the potential vulnerabilities of the system to CCFs, and to identify those groups of components within the system whose CCFs contribute significantly to the system unavailability. The screening step develops the scope and justification for the detailed analysis. The screening analysis provides conservative, bounding system unavailabilities due to CCFs. Depending on the objectives of the study and the availability of resources, the analysis may be stopped at the end of this step recognizing that the qualitative results may not accurately represent the actual system vulnerabilities, and that the quantitative estimates may be very conservative.

The Detailed Analysis phase uses the results of the screening step and through several steps involving the detailed logic modeling, parametric representation, and data analysis, develops numerical values for system unavailabilities due to CCF events.

7.6 Preliminary Identification of Common Cause Failure Vulnerabilities (Screening Analysis)

The primary objective of this phase is to identify in a conservative way, and without significant effort, all important groups of components susceptible to CCF. This is done in two steps:

- Qualitative Screening
- Quantitative Screening.

7.6.1 Qualitative Screening

At this stage, an initial qualitative analysis of the system is performed to identify the potential vulnerabilities of the system and its components to CCFs. This analysis is aimed at providing a list of components, which are believed to be susceptible to CCF. At a later stage, this initial list will be modified on quantitative grounds. In this early stage, conservatism is justified and encouraged. In fact, it is important not to discount any potential CCF vulnerability unless there are immediate and obvious reasons to discard it.

The most efficient approach to identifying common cause system vulnerabilities is to focus on identifying *coupling factors*, regardless of defenses that might be in place against some or all categories of CCFs. The result will be a conservative assessment of the system vulnerabilities to CCFs. This, however, is consistent with the objective of this stage of the analysis, which is a preliminary, high-level screening.

From the earlier discussion it is clear that a coupling factor is what distinguishes CCFs from multiple independent failures. Coupling factors are suspected to exist when two or more component failures exhibit similar characteristics, both in the cause and in the actual failure

mechanism. The analyst, therefore, should focus on identifying those components of the system that share one or more of the following:

- Same design
- Same hardware
- Same function
- Same installation, maintenance, or operations people
- Same procedures
- Same system/component interface
- Same location
- Same environment

This process can be enhanced by developing a checklist of key attributes, such as design, location, operation, etc., for the components of the system. An example of such a list is the following:

- Component type (e.g., motor-operated valve): including any special design or construction characteristics, such as component size and material
- Component use: system isolation, parameter sensing, motive force, etc.
- Component manufacturer
- Component internal conditions: temperature range, normal flow rate, power requirements, etc.
- Component boundaries and system interfaces: connections with other components, interlocks, etc.
- Component location name and/or location code
- Component external environmental conditions: e.g., temperature, radiation, vibration
- Component initial conditions: normally closed, normally open, energized, etc.; and operating characteristics: normally running, standby, etc.
- Component testing procedures and characteristics: test configuration or lineup, effect of test on system operation, etc.
- Component maintenance procedures and characteristics: planned, preventive maintenance frequency, maintenance configuration or lineup, effect of maintenance on system operation, etc.

The above list, or a similar one, is a tool to help identify the presence of identical components in the system and most commonly observed coupling factors. It may be supplemented by a system “walk-down” and review of operating experience (e.g., failure event reports). Any group of components that share similarities in one or more of these characteristics is a potential point of vulnerability to CCF. However, depending on the system design, functional requirements, and operating characteristics, a combination of commonalities may be required to create a realistic condition for CCF susceptibility. Such situations should be evaluated on a case-by-case basis before deciding on whether or not there is a vulnerability. A group of components identified in this process is called a common cause component group (CCCG).

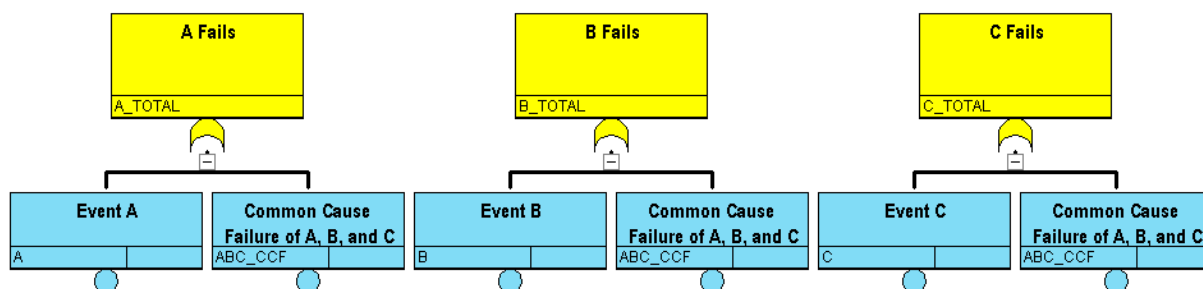
Finally, in addition to the above guidelines, it is important for the analyst to review the operating experience to ensure that past failure mechanisms are included with the components selected in the screening process. Later, in the detailed qualitative and quantitative analysis phases, this task is performed in more detail to include the operating experience of the system being analyzed.

7.6.2 Quantitative Screening

The qualitative screening step identifies potential vulnerabilities of the system to CCFs. By using conservative qualitative analysis, the size of the problem is significantly reduced. However, detailed modeling and analysis of all potential common cause vulnerabilities identified in the qualitative screening may still be impractical and beyond the capabilities and resources available to the analyst. Consequently, it is desirable to reduce the size of the problem even further to enable detailed analysis of the most important common cause system vulnerabilities. Reduction is achieved by performing a quantitative screening analysis. This step is useful for systems FT analysis and may be essential for ESD-level analysis in which exceedingly large numbers of cut sets may be generated in solving the FT logic model.

In performing quantitative screening for CCF candidates, one is actually performing a complete quantitative analysis except that a conservative and simple quantitative model is used. The procedure is as follows:

1. The component-level FTs are modified to explicitly include a “global” or “maximal” CCF event for each component in every CCCG. A global common cause event in a group of components is one in which all members of the group fail. A maximal common cause event is one that represents two or more CCBEs. As an example of this step of the procedure, consider a CCCG composed of three components A, B, and C. According to the procedure, the basic events of the FT involving these components, i.e., “A Fails,” “B Fails,” and “C Fails,” are expanded to include the basic event C_{ABC} , which is defined as the concurrent failure of A, B, and C due to a common cause, as shown below:



Here A_i , B_i , and C_i denote the independent failure of components A, B, and C, respectively. This substitution is made at every point on the FTs where the events “A FAILS,” “B FAILS,” or “C FAILS” occur.

2. The FTs are now solved to obtain the minimal cut sets (MCSs) for the system or accident sequence. Any resulting cut set involving the intersection $A_i B_i C_i$ will have an associated cut set involving C_{ABC} . The significance of this process is that, in large system models or event sequences, some truncation of the cut sets on failure probability must usually be performed to obtain any solution at all, and the product of independent failures $A_i B_i C_i$ is often lost in the truncation process due to its small value, while the (numerically larger) common cause term C_{ABC} will survive.

3. Numerical values for the CCBE can be estimated using a simple global parametric model:

$$\Pr(C_{ABC}) = g \Pr(A) \quad (7-5)$$

4. $\Pr(A)$ is the total failure probability of the component. Typical generic value for “g” range between 0.05 and 0.10, but more accurate generic values that consider different logic configuration (k-out-of-n) can also be used. Table 7-1 lists values of the global common cause factor, g, for dependent k-out-of-n system configurations for success. The basis for these screening values is described in Reference [7-1]. Note that different g values apply depending on whether the components of the system are tested simultaneously (non-staggered) or one at a time at fixed time intervals (staggered). More details on the reasons for the difference are provided in Reference [7-1].

Table 7-1. Screening Values of Global CCF (g) for Different System Configurations.

Success Configuration	Values of g	
	Staggered Testing Scheme	Non-staggered Testing Scheme
1 of 2	0.05	0.10
2 of 2		
1 of 3	0.03	0.08
2 of 3	0.07	0.14
3 of 3		
1 of 4	0.02	0.07
2 of 4	0.04	0.11
3 of 4	0.08	0.19
4 of 4		

The simple global or maximal parameter model provides a conservative approximation to the CCF frequency regardless of the number of redundant components in the CCG being considered.

Those CCGs that are found to contribute little to system unavailability or event sequence frequency (or which do not survive the probability-based truncation process) can be dropped from further consideration. Those that are found to contribute significantly to the system unavailability or event sequence frequency are retained and further analyzed using the guidelines for more detailed qualitative and quantitative analysis.

The objective of the initial screening analysis is to identify potential common cause vulnerabilities and to determine those that are insignificant contributors to system unavailability and to the overall risk, to eliminate the need to analyze them in detail. The analysis can stop at this level if a conservative assessment is acceptable and meets the objectives of the study. Otherwise the component groups that survive the screening process should be analyzed in more detail, according to the Detailed Analysis phase.

A complete detailed analysis should be both qualitative and quantitative. A detailed quantitative analysis is always required to provide the most realistic estimates with minimal uncertainty. In general, a realistic quantitative analysis requires a thoroughly conducted qualitative analysis. A detailed qualitative analysis provides many valuable insights that can be of direct use in improving the reliability of the systems and safety of the mission.

7.7 Incorporation of CCFs into System Models (Detailed Analysis)

The objective of the detailed analysis is to identify the potential vulnerabilities of the system being analyzed to the diverse CCFs that can occur, and to incorporate their impact into the system models. As a first step, the analyst should extend the scope of the qualitative screening analysis and conduct a more thorough qualitative assessment of the system vulnerabilities to CCF events. This detailed analysis focuses on obtaining considerably more system-specific information and can provide the basis and justification for engineering decisions regarding system reliability improvements. In addition, the detailed evaluation of system CCF vulnerabilities provides essential information for a realistic evaluation of operating experience and system-specific data analysis as part of the detailed quantitative analysis. It is assumed that the analyst has already conducted a screening analysis, is armed with the basic understanding of the analysis boundary conditions, and has a preliminary list of the important CCGs.

An effective detailed qualitative analysis involves the following activities:

- Review of operating experience (generic and system-specific)
- Review of system design and operating practices
- Identification of possible causes and coupling factors and applicable system defenses.

The key products of this phase of analysis include a final list of CCGs supported by documented engineering evaluation. This evaluation may be summarized in the form of a set of Cause-Defense and Coupling Factor-Defense matrices (see Reference [7-1]) developed for each of the CCGs identified in the screening phase. These detailed matrices explicitly account for system-specific defenses, including design features and operational and maintenance policies in place to reduce the likelihood of failure occurrences. The results of the detailed qualitative analysis provide insights about safety improvements that can be pursued to improve the effectiveness of these defenses and reduce the likelihood of CCF events.

Given the results of the screening analyses, a detailed quantitative analysis can be performed even if a detailed qualitative analysis has not been conducted. However, as will be seen later, some of the steps in the detailed quantitative phase, particularly those related to analysis and classification of failure events for CCF probability estimation can benefit significantly from the insights and information obtained as a result of a detailed qualitative analysis.

A detailed quantitative analysis can be achieved through the following steps:

1. Identification of CCBEs
2. Incorporation of CCBEs into the system FT
3. Development of probabilistic models of CCBEs
4. Estimation of CCBE probabilities

These steps are discussed in the following sections.

7.7.1 Identification of CCBEs

This step provides the means for accounting for the entire spectrum of CCF impacts in an explicit manner in the logic model. It will also facilitate the FT quantification to obtain top event (system failure) probability.

A CCBE is an event involving failure of a specific set of components due to a common cause. For instance in a system of three redundant components A, B, and C, the CCBEs are

C_{AB} , C_{AC} , C_{BC} , and C_{ABC} . The first event is the common cause event involving components A and B, and the fourth is a CCF event involving all three components. Note that the CCBEs are only identified by the impact they have on specific sets of components within the CCCGs. Impact in this context is limited to “failed” or “not failed.”

The complete set of basic events, including CCBEs, involving component A in the three component system is:

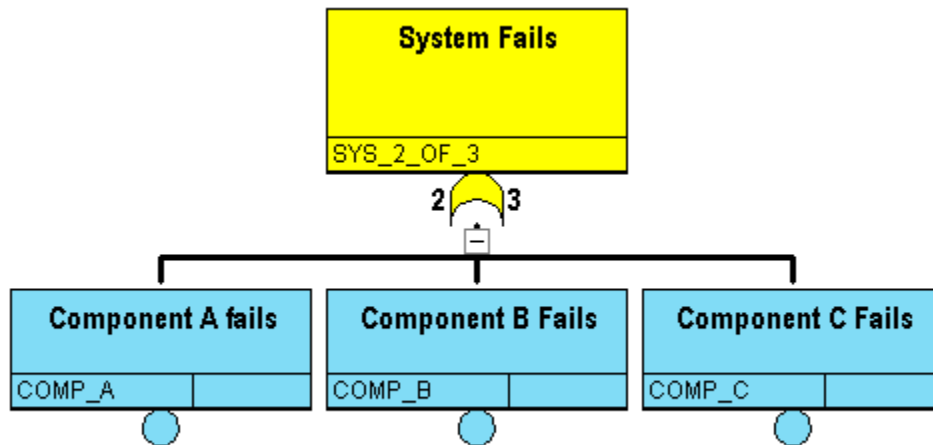
- A_I = Single independent failure of component A. (a basic event)
- C_{AB} = Failure of components A and B (and not C) from common causes
- C_{AC} = Failure of components A and C (and not B) from common causes
- C_{ABC} = Failure of components A, B, and C from common causes.

Component A fails if any of the above events occur. The equivalent Boolean representation of total failure of component A is

$$A_T = A_I + C_{AB} + C_{AC} + C_{ABC} \quad (7-6)$$

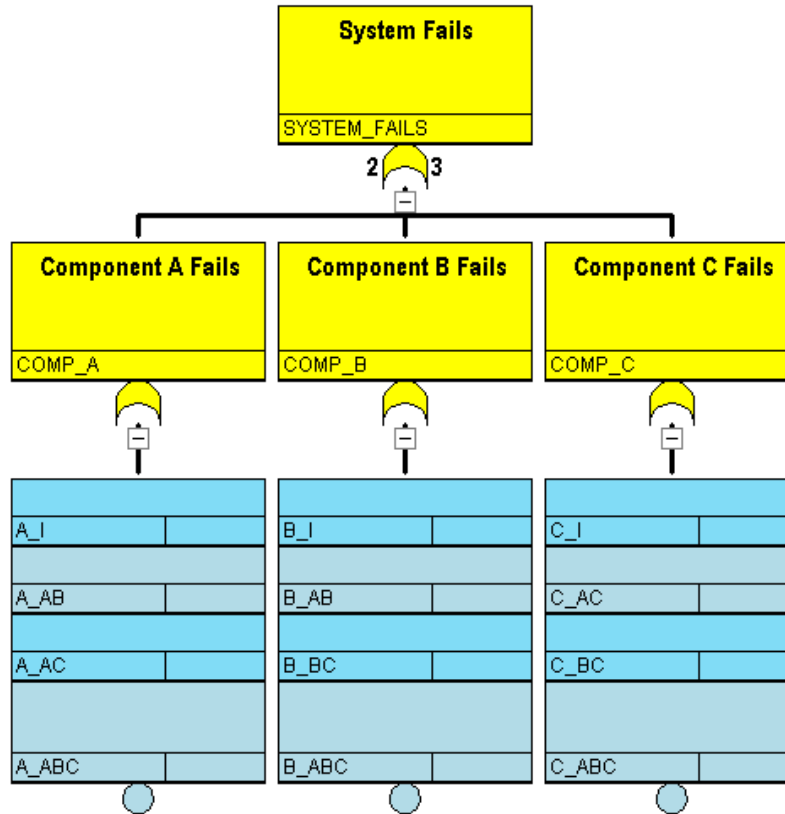
7.7.2 Incorporation of CCBEs into the Component-Level Fault Tree

In this step the component-level FT is expanded in terms of the CCBEs. As an example of this expansion, consider the following system of three identical components, A, B, and C, with a “two-out-of-three” success logic. Also assume that, based on the qualitative and quantitative screening, these three components form a single CCCG. The component-level FT of this system is



Note that the MCSs of this FT are {A,B}; {A,C}; {B,C}.

The expansion of this FT down to the common cause impact level can be achieved by replacing each of the three component basic events by the corresponding set of CCBE events in OR formation, as shown in the following figure:



When the expanded FT is solved, the following cut sets are obtained:

- $\{A_I, B_I\}$; $\{A_I, C_I\}$; $\{B_I, C_I\}$
- $\{C_{AB}\}$; $\{C_{AC}\}$; $\{C_{BC}\}$
- $\{C_{ABC}\}$.

If the success criterion for this example had been only one out of three instead of two out of three, the expanded FT would produce cut sets of the type, $C_{AB} \cap C_{AC}$. These cut sets imply failure of the same piece of equipment due to several causes, each of which is sufficient to fail the component. For example, in $C_{AB} \cap C_{AC}$, component A is failing due to a CCF that fails AB, and also due to a CCF that fails AC. These cut sets have questionable validity unless the events C_{AB} and C_{AC} are defined more precisely. Reference [7-1] discusses the conditions under which these cut sets are valid. However, experience shows that in general the contribution of cut sets of this type is considerably smaller than that of cut sets like C_{ABC} . These cut sets will be eliminated here.

The reduced Boolean representation of the system failure in terms of these CCBE cut sets is

$$S = (A_I \cap B_I) \cup (A_I \cap C_I) \cup (B_I \cap C_I) \cup C_{AB} \cup C_{AC} \cup C_{BC} \cup C_{ABC} \quad (7-7)$$

It can be seen immediately that this expansion results in proliferation of the cut sets, which may create practical difficulties when dealing with complex systems. The potential difficulty involving the implementation of this procedure is one of the motivations for a thorough and systematic screening in earlier steps to minimize the size of the expanded FT. Despite the potential difficulty in implementation, this procedure provides the analyst with a systematic and

disciplined framework for inclusion and exclusion of common cause events, with adequate assurance that the resulting model of the system is complete with respect to all possible ways that common cause events could impact the system.

Another advantage of this procedure is that once the CCBEs are included in the FT, standard FT techniques for cut set determination and probabilistic quantification can be applied without concern about dependencies due to CCFs.

If, after careful screening, the number of cut sets is still unmanageable, a practical solution is to delay the common cause impact expansion until after the component-level FT is solved, at which time those terms in the component-level Boolean expression that had not been expanded would be expanded through a process similar to that in Equation (7-6) and the new Boolean expression would be reduced again. Other techniques include reducing the level of detail of the original component-level tree by introducing “supercomponents,” and assuming that the common cause events always have a global effect. Care, however, must be exercised so that no terms in the expansion of the reduced Boolean expressions would be missed or ignored.

7.7.3 Development of Probabilistic Models of CCBEs

In the previous steps CCF events were introduced into FT models through the CCBE. This section describes the probabilistic models that are commonly used for CCBEs. This is done first by utilizing the same three-component system example, and then generalized to all levels of redundancy.

Referring to Equation (7-7) and using the rare event approximation, the system failure probability of the two-out-of-three system is given by

$$\begin{aligned} \Pr(S) = & \Pr(A_1) \Pr(B_1) + \Pr(A_1) \Pr(C_1) + \Pr(B_1) \Pr(C_1) + \Pr(C_{AB}) \\ & + \Pr(C_{AC}) + \Pr(C_{BC}) + \Pr(C_{ABC}) \end{aligned} \quad (7-8)$$

It is common practice in risk and reliability analysis to assume that the probabilities of similar events involving similar components are the same. This approach takes advantage of the physical symmetries associated with identically redundant components in reducing the number of parameters that need to be quantified. For example, in the above equation it is assumed that:

$$\begin{aligned} \Pr(A_1) = \Pr(B_1) = \Pr(C_1) &= Q_1 \\ \Pr(C_{AB}) = \Pr(C_{AC}) = \Pr(C_{BC}) &= Q_2 \\ \Pr(C_{ABC}) &= Q_3 \end{aligned} \quad (7-9)$$

In other words, the probability of occurrence of any basic event within a given CCG is assumed to depend only on the number and not on the specific components in that basic event.

With the symmetry assumption, and using the notation just introduced, the system failure probability can be written as

$$Q_s = 3(Q_1)^2 + 3Q_2 + Q_3 \quad (7-10)$$

For quantification of the expanded FT,

$$Q_k^m \equiv \text{probability of a CCBE involving } k \text{ specific components in a common cause component group of size } m \text{ (} 1 \leq k \leq m \text{)}$$

The model that uses Q_k^m s to calculate system failure probability is called the Basic Parameter (BP) model [7-1].

For several practical reasons, it is often more convenient to rewrite $Q_k^{(m)}$ s in terms of other more easily quantifiable parameters. For this purpose a parametric model known as the Alpha Factor model is recommended [7-1]. Reasons for this choice are that the Alpha Factor model: (1) is a multi-parameter model which can handle any redundancy level; (2) is based on ratios of failure rates, which makes the assessment of its parameters easier when no statistical data are available; (3) has a simpler statistical model; and (4) produces more accurate point estimates as well as uncertainty distributions compared to other parametric models that have the above properties.

The Alpha Factor model develops CCF frequencies from a set of failure ratios and the total component failure rate. The parameters of the model are:

$Q_t \equiv$ total failure frequency of each component due to all independent and common cause events.

$\alpha_k \equiv$ fraction of the total frequency of failure events that occur in the system and involve failure of k components due to a common cause.

Using these parameters, depending on the assumption regarding the way the redundant components of the systems in the database are tested (as part of the data collection effort), the frequency of a CCBE involving failure of k components in a system of m components is given by:

- For a staggered testing scheme:

$$Q_k^m = \frac{1}{\binom{m-1}{k-1}} \alpha_k Q_t \quad (7-11)$$

- For a non-staggered testing scheme:

$$Q_k^m = \frac{k}{\binom{m-1}{k-1}} \frac{\alpha_k}{\alpha_t} Q_t \quad (7-12)$$

where the binomial coefficient is given by:

$$\binom{m-1}{k-1} = \frac{(m-1)!}{(k-1)!(m-k)!} \quad (7-13)$$

and

$$\alpha_t = \sum_{i=1}^m k \alpha_k \quad (7-14)$$

As an example, the probabilities of the basic events of the example three-component system are written as (assuming staggered testing):

$$\begin{aligned}
 Q_1^3 &= \alpha_1 Q_t \\
 Q_2^3 &= \frac{1}{2} \alpha_2 Q_t \\
 Q_3^3 &= \alpha_3 Q_t
 \end{aligned}
 \tag{7-15}$$

Therefore, the system unavailability can now be written as

$$Q_s = 3(\alpha_1 Q_t)^2 + \frac{3}{2} \alpha_2 Q_t + 3\alpha_3 Q_t
 \tag{7-16}$$

Note that the staggered versus non-staggered assumptions are applicable for parameter estimation as part of the data collection activities. During modeling activities, the typical CCF model to be used will be that of non-staggered testing.

7.7.4 Estimation of CCBE Probabilities

The objective of this step is to estimate the CCBE probabilities or parameters of the model used to express these probabilities. Ideally, parameter values are estimated based on actual field experience. The most relevant type of data would be the system-specific data. However, due to the rarity of system-specific common cause events a search will usually not produce statistically significant data. In almost all cases parameter estimation will have to include experience from other systems, i.e., generic data. In some cases even the generic data may be unavailable or insufficient. Data might be obtained from various sources including:

- Industry-based generic data
- System-specific data records
- Generically classified CCF event data and parameter estimates (reports and computerized databases).

Only a few industries have developed databases for CCF events. These include nuclear power and, to a lesser extent, aerospace.

The problem of data scarcity can be addressed at least in part by applying a method for extracting information from partially relevant data based on using the *Impact Vector Method* and *Bayesian* techniques [7-1]. This is done through a two-step process:

1. **Generic Analysis:** Analysis of occurrences of CCFs in various systems in terms of their causes, coupling factors, as well as the level of impact, i.e., the number and nature of component failures observed.
2. **System-Specific Analysis:** Re-evaluation of the generic data for applicability and relevance to the specific system of interest.

The specific techniques are described in Reference [7-1]. In the following it is assumed that the statistical data needed for the estimation of CCF model parameters are developed by following the referenced procedure or a similar one.

Once the impact vectors for all the events in the database are assessed for the system being analyzed, the number of events in each impact category can be calculated by adding the corresponding elements of the impact vectors. The process results in

$$n_k = \text{total number of basic events involving failure of } k \text{ similar components, } k=1, \dots, m$$

Event statistics, n_k , are used to develop estimates of CCF model parameters. For example, the parameters of the alpha-factor model can be estimated using the following maximum likelihood estimator (MLE):

$$\hat{\alpha}_k = \frac{n_k}{\sum_{j=1}^m n_j} \quad (7-17)$$

For example, consider a case where the analysis of failure data for a particular two-out-of-three system reveals that of a total of 89 failure events, there were 85 single failures, 3 double failures, and 1 triple failure, due to common cause. Therefore the statistical data base is $\{n_1 = 85, n_2 = 3, n_3 = 1\}$. Based on the estimator of Equation (7-17):

$$\alpha_1 = \frac{n_1}{n_1 + n_2 + n_3} = \frac{85}{89} = 0.955$$

$$\alpha_2 = \frac{n_2}{n_1 + n_2 + n_3} = \frac{3}{89} = 0.034$$

$$\alpha_3 = \frac{n_3}{n_1 + n_2 + n_3} = \frac{1}{89} = 0.011$$

Table 7-2 provides a set of estimators. The estimators presented in Table 7-2 are the MLEs and are presented here for their simplicity. The mean values obtained from probability distribution characterizing uncertainty in the estimated values are more appropriate for point value quantification of system unavailability. Bayesian procedures for developing such uncertainty distributions are presented in References 7-1 and 7-4.

Table 7-2 displays two sets of estimators developed based on assuming different testing schemes. Depending on how a given set of redundant components in a system is tested (demanded) in staggered or non-staggered fashion, the total number of challenges that various combinations of components are subjected to is different. This needs to be taken into account in the exposure (or success) part of the statistics used, affecting the form of the estimators. The details of why and how the estimators are affected by testing schedule are provided in Reference [7-1].

7.8 Generic Parameter Estimates

For cases where no data are available to estimate CCF model parameters, generic estimates based on parameter values developed for other components, systems, and applications may be used as screening values. The average value of these data points is $\beta = 0.1$ (corresponding to an alpha factor of 0.05 for a two-component system). However, values for specific components range about this mean by a factor of approximately two.

These values are in fact quite typical and are also observed in CCF data collection efforts in some other industries. A very relevant example is the result of analysis of Space Shuttle CCF events [7-3]. A total of 474 Space Shuttle orbiter in-flight anomaly reports were analyzed in search of Dependent Failures (DFs) and Partial Dependent Failures (PDFs). The data were used to determine frequency and types of DFs, causes, coupling factors, and defenses associated with the Shuttle flights. These data were also used to estimate a generic beta factor that resulted in a value of 0.13.

Table 7-2. Simple Point Estimators for Various CCF Parametric Models.

Method	Non-Staggered Testing*	Staggered Testing*	Remarks
Basic Parameter	$Q_k^m = \frac{n_k}{\binom{m}{k} N_D} \quad k = 1, \dots, m$	$Q_k^m = \frac{n_k}{m \binom{m}{k} N_D} \quad k = 1, \dots, m$	For time-based failure rates, replace system demands (N_D) with total system exposure time T .
Alpha Factor	$\alpha_j^m = \frac{n_k}{\sum_{j=1}^m n_j} \quad k = 1, \dots, m$	Same as non-staggered case	
* N_D is the total number of tests or demands on a system of m components.			

7.9 Treatment of Uncertainties

Estimation of model parameters involves uncertainties that need to be identified and quantified. A broad classification of the types and sources of uncertainty and potential variabilities in the parameter estimates is as follows:

1. Uncertainty in statistical inference based on limited sample size.
2. Uncertainty due to estimation model assumptions. Some of the most important assumptions are:
 - A. Assumption about applicable testing scheme (i.e., staggered vs. non-staggered testing methods).
 - B. Assumption of homogeneity of the data generated through specializing generic data to a specific system.
3. Uncertainty in data gathering and database development. These include:
 - A. Uncertainty because of lack of sufficient information in the event reports, including incompleteness of data sources with respect to number of failure events, number of system demands, and operating hours.
 - B. Uncertainty in translating event characteristics to numerical parameters for impact vector assessment (creation of generic database).
 - C. Uncertainty in determining the applicability of an event to a specific system design and operational characteristics (specializing generic database for system-specific application).

The role of uncertainty analysis is to produce an epistemic probability distribution of the CCF frequency of interest in a particular application, covering all relevant sources of uncertainty from the above list. Clearly, some of the sources or types of uncertainty may be inapplicable, depending on the intended use of the CCF parameter and the form and content of the available database. Also, methods for handling various types of uncertainty vary in complexity and accuracy. Reference [7-1] provides a comprehensive coverage of the methods for assessing uncertainty distribution for the parameters of various CCF models.

7.10 References

- 7-1 A. Mosleh, et al, "Procedures for Treating Common Cause Failures in Safety and Reliability Studies," U.S. Nuclear Regulatory Commission and Electric Power Research Institute, NUREG/CR-4780, and EPRI NP-5613. Volumes 1 and 2, 1988.
- 7-2 K.N. Fleming, "A reliability model for common mode failure in redundant safety systems," General Atomic Report GA-A13284, April 1975.
- 7-3 P.J. Rutledge, and A. Mosleh, "An Analysis of Spacecraft Dependent Failures," Proceedings of the Second International Conference on Probabilistic Safety Assessment and Management, PSAM-II, San Diego California, March 20-25, 1994.
- 7-4 NASA-SP-2009-569, *Bayesian Inference for NASA Risk and Reliability Analysis*, 2009.

8. Human Reliability Analysis (HRA)

The purpose of this chapter is to provide guidance on how to perform Human Reliability Analysis (HRA) in the context of a PRA. In this context, HRA is the assessment of the reliability and risk impact of the interactions of humans on a system or a function. For situations that involve a large number of human-system interactions (HSIs), HRA becomes an important element of PRA to ensure a realistic assessment of the risk. Examples of HSIs include: activities of the ground crew such as the Flight Control Officer (FCO) to diagnose a launch vehicle guidance control malfunction and initiate the Command Destruct System (CDS); flight crew actions to recover from potential system malfunctions; and mechanical/electrical personnel errors during installation, test, and maintenance of equipment prior to start of the mission. The HRA analysts, with support from systems analysts, model and quantify the impacts from these HSIs, which then will be incorporated as human basic events in the PRA logic models (e.g., ETs, FTs). It is noted that in addition to “human interaction,” the terms “human action,” “human error,” and “human failure” have been used in HRA literature and will also be used in this guide, particularly when it comes to the quantification of the impacts of HSIs.

8.1 Basic Steps in the HRA Process

In general, the HRA process has a number of distinct steps, as shown below in Figure 8-1, that will be briefly described in this section.



Figure 8-1. Basic Steps in the HRA Process.

Problem Definition

The problem definition is the first step in the process and is used to determine the scope of the analysis, including what tasks (normal, emergency) will be evaluated, and what human actions will be assessed. These actions need to be identified within the scope of the PRA in terms of the human interactions that are considered in the PRA. For the systems modeled in the PRA, to determine the scope of the human actions that need to be considered, the system's vulnerability to human error needs to be assessed. A NASA space system's vulnerability to human error is dependent upon the complexity of the system (and how the NASA team understands this complexity), the amount that the human interacts with the system (either through maintenance, operation, and/or recovery), and how the human-system is coupled. (A tightly coupled system does not allow the user the flexibility to use alternatives or wait for a repair when there is a failure).

In general, when a system is more vulnerable to human error, then a larger scope and more comprehensive analysis is needed to understand fully and mitigate the human contribution to system risk. During the problem definition phase, determining what type of human actions will be evaluated is very important, because the number and type of errors included in the analysis can lead to an underestimation or overestimation of the impact of the human errors on the

system risk. The subsequent sections provide guidelines to help determine the human interactions that need to be modeled as part of the PRA.

The output of step 1 is a detailed list of the types of human actions that will be evaluated, including nominal and off-nominal, and emergency scenarios.

Task Analysis

The second step in the HRA process is task analysis that identifies the specific tasks and specific human actions that are involved in the human interactions with the system. These tasks can involve physical actions and/or cognitive processes (e.g., diagnosis, calculation, and decision making). Swain [8-1] defines task analysis as follows: "An analytical process for determining the specific behaviors required of the human performance in a system. It involves determining the detailed performance required of people and equipment and the effects of environmental conditions, malfunctions, and other unexpected events on both. Within each task to be performed by people, behavioral steps are analyzed in terms of (1) the sensory signals and related perceptions, (2) information processing, decision-making, memory storage, and other mental processes, and (3) the required responses." The task analysis can be relatively simple or can be complex depending on the type of interactions that are involved. When considering the level of task decomposition, the analyst needs to consider the purpose of the task analysis and the resources available. For an HRA of an initial design, general task definitions may be sufficient. For an HRA of a complex human interaction, a more detailed task analysis may be necessary if the system performance is sensitive to the human interaction. Subsequent sections give examples of task analysis and more detailed descriptions are given in the references.

Error Identification

The third and the most important step in the HRA is human error identification, where human interactions and basic human actions are evaluated to determine what human errors and violations can occur, have potential contributions to hazardous events, and should be included in the PRA.. The analyst must determine what type of human error will occur and the performance factors that could contribute to the error. To accomplish this, the analyst must identify and understand the different types of human errors that can impact the system. Human actions/interactions within a system can be broken down into two main types of elements, a cognitive response or a physical action, and their related errors of omission or commission. Human actions and errors cannot be considered in isolation from the system and environment in which the human works. The system design (hardware, software, and crew habitable environment) affects the probability that the human operator will perform a task correctly or incorrectly for the context and specific situation. Consequently, it is important to evaluate the factors, called Performance Shaping Factors (PSFs) that may increase or decrease the likelihood that these errors will occur. PSF values depend on the specific HRA model used and several examples are subsequently described.

Error Representation (Modeling)

The fourth step in HRA is human error representation, also described as modeling. This step is conducted to help visualize the data, relationships, and inferences that cannot be as easily described with words. Human error modeling allows the analyst to gain insight into the causes, vulnerabilities, recoveries, and possible risk mitigation strategies associated with various accident scenarios. Human errors can be modeled and represented in a Master Logic Diagram (MLD), Event Sequence Diagram (ESD), Event Tree (ET), Fault Tree (FT), or a generic error model and influence diagram. The most appropriate representation and modeling depends on the classification of the human interaction and associated human error (Section 8.2). Alternative

modeling approaches that are amenable for NASA implementations are subsequently described.

Quantification and Integration into PRA

Quantification, the fifth and final step in HRA, is the process used to assign probabilities to the human errors. The human error probabilities (HEPs) are incorporated into the PRA to determine their risk contribution. The method by which quantification is completed is dependent upon the resources available, the experience level of the analyst, and the relevant available data. Quantification data may come from databases, simulations, or expert judgment. The method of quantification also depends on the particular modeling approach used for the HRA and alternative approaches are described.

8.2 Classifications of Human Interactions and Associated Human Errors

To assist in determining the scope of the HRA to be performed, it is useful to classify the types of HSIs and associated human errors that can occur. Many classifications of HSIs and associated human errors have been described in HRA literature. The classifications consider different aspects of HSIs such as their timing with respect to the initiating event (IE), human error type, and cognitive behavior of humans in responding to accidents. Similar to hardware reliability modeling (e.g., failure on demand, running failure, etc.), HSI classification and human error classification is a key step in HRA that supports model development, data collection, and quantification of human actions. Several of the most widely used HSI classifications in HRA are briefly described in the following.

8.2.1 Pre-Initiator, Initiator, and Post-Initiator HSIs

Three types of HSIs, based on their timing with respect to an accident initiating event, (IE), that are useful for initial categorization for HRA are [8-1, 8-2]:

- Pre-initiator HSIs rendering equipment unavailable before it operates or is called upon (e.g., maintenance errors, testing errors, calibration errors);
- Initiator-related HSIs that contribute to the initiation of a potential accident (e.g., a human error causing a loss of system or inadvertent actuation of a system); and
- Post-initiator HSIs that occur during the progression of an accident (e.g., actuating a backup safety system, performing a recovery action).

Post-Initiator HSIs are furthermore broken into two main elements:

- Cognitive response: Detection (e.g., recognizing an abnormal event), diagnosis, and decision making to initiate a response within the time available; and
- Post-diagnosis action response: Performance of actions (or tasks execution) after the diagnosis has been made, within the time available.

A failure of cognitive response or post-diagnosis response involves failure of any of the steps involved in the correct response. Sometimes, failure of cognitive response is simply referred to as diagnosis failure or misdiagnosis. Failure of post-diagnosis action is simply referred to as a diagnosis follow-up failure, or follow-up failure.

8.2.2 Skill, Rule, and Knowledge-Based Response

Rasmussen [8-3] proposed three more specific categories of human cognitive response:

- Skill-based (S): Response requiring little or no cognitive effort;

- Rule-based (R): Response driven by procedures or rules; and
- Knowledge-based (K): Response requiring problem solving and decision making.

Skill-based behavior is characterized by a quasi-instinctive response, i.e., a close coupling between input signals and output response. Skill-based response occurs when the individual is well trained on a particular task, independent of the level of complexity of the task. Skill-based behavior is characterized by a fast performance and a low number of errors.

Rule-based response is encountered when an individual's actions are governed by a set of well-known rules, which he or she follows. The major difference between skill-based and rule-based behavior is in the degree of practice of rules. Since the rules need to be checked, the response is slower and more prone to errors.

Knowledge-based response is characteristic of unfamiliar or ambiguous situations. In such cases, the individual will need to rely on his or her own knowledge of the system and situation. Knowledge-based behavior is the most error prone of the three types of behavior.

8.2.3 Error of Omission and Error of Commission

Two types of human error have further been defined by Swain [8-1, 8-4]:

- Error of Omission (EOM): The failure to initiate performance of a system-required task/action (e.g., skipping a procedural step or an entire task); and
- Error of Commission (ECOM): The incorrect performance of a system-required task/action, given that a task/action is attempted, or the performance of some extraneous task/action that is not required by the system and that has the potential for contributing to a system failure (e.g., selection of a wrong control, sequence error, timing error).

EOMs are often the dominant pre-initiator errors. ECOMs can be important contributors to accident initiators. Both EOMs and ECOMs can be important contributors for post-initiator errors.

8.3 General Modeling of Pre-Initiator, Initiator, and Post-Initiator HSIs in a PRA

General guidelines that are commonly used in modeling HSIs in a PRA are:

- Pre-initiator HSIs are explicitly modeled and are usually included in the system FTs at the component level.
- Initiator HSIs are explicitly modeled and can be included as pivotal events in the ET or in the system FTs at the component level. Post-Initiator HSIs are explicitly modeled and can be included at different levels of the PRA logic model:
 - Errors associated with recovery of component failures are modeled in the system FTs
 - Errors associated with response to an accident initiating event may be modeled in the system FTs or ETs.

8.4 Quantification of Human Interactions (or Errors)

The systems and the HRA analysts may identify a large number of HSIs in a PRA. Detailed task analysis, required for HSI quantification, can be a time-consuming and resource intensive task. It may not be possible, or necessary, to perform detailed quantification for all HSIs.

Therefore, for practical reasons, HSI quantification in HRA is usually performed in two phases:

- Screening analysis; and

- Detailed analysis.

This section describes the basic steps in screening analysis. The detailed analysis that is subsequently carried out depends on the specific HRA modeling approach used and these are described in subsequent sections.

The purpose of the screening analysis is to reduce the number of HSIs to be analyzed in detail in HRA. The screening analysis may be qualitative, quantitative, or a combination of both.

8.4.1 Qualitative Screening

Qualitative screening is usually performed early in HRA to exclude some HSIs from further analysis and, hence, not to incorporate them in the PRA logic models. A set of qualitative screening rules is developed for each HSI type. Examples of commonly used qualitative screening rules are as follows:

- Screen out misaligned equipment as a result of a test/maintenance error, when by design automatic re-alignment of equipment occurs on demand.
- Screen out misaligned equipment as a result of a human error, when a full functional test is performed after maintenance/assembly (for Type A HSIs).
- Screen out misaligned equipment as a result of a human error, when equipment status is indicated in the control room or spacecraft.
- Screen out HSIs if their success/failure has no influence on the accident progression, e.g., verification tasks.
- Screen out HSIs and assume the task is not carried out if there are physical limitations to carry out the task, e.g., time is too short, impossible access due to hostile environment, lack of proper tools.
- Screen out HSIs and assume the action is not carried out if the individual is unlikely or reluctant to perform the action, e.g., training focuses on other priorities/strategies.

8.4.2 Quantitative Screening

Quantitative screening^a is also performed to limit the detailed task analysis and quantification to important (risk-significant) HSIs. Conservative HEP estimates are used in the PRA logic models to perform this quantitative screening. HSIs that are shown to have insignificant impact on risk (i.e., do not appear in dominant accident sequence cut sets) even with the conservative HEPs, are screened out from further detailed analysis. The key elements of a screening analysis are as follows:

- Conservative HEPs typically in the range of 0.1 to 1.0 are used for various HSIs depending on their complexity and timing as well as operators' familiarity with them. Lower values such as 0.01 or 0.005 may also be used as conservative values in certain scenarios when there is an associated rationale or basis. Usually, no recovery factors are considered.
- Complete dependence is assumed among multiple related actions that appear in the same accident sequence cut set, i.e., if an individual fails on the first action with an estimated HEP, then the HEPs on the second and third (and so on) related actions are unity (1.0).

8.5 HRA Models

This section describes HRA modeling approaches that are suitable for use in carrying out human error analysis as part of a PRA, which includes modeling HSIs and the associated human errors. Several modeling approaches are described since they focus on different types of HSIs and involve different levels of task descriptions. Also, it can be useful to apply different models to obtain different perspectives and to check for consistency. The modeling approaches that are selected and described here are based on the reviews of different HRA approaches and their suitability for NASA applications that are described in Reference [8-5]. For each modeling approach described here, overviews of the screening and quantitative analysis capabilities are provided. Reference [8-6] provides additional information on these approaches as well as other HRA approaches.

8.5.1 Technique for Human Error Rate Prediction (THERP)

THERP is comprehensive HRA methodology that was developed by Swain & Guttman [8-4] for the purpose of analyzing human reliability in nuclear power plants. THERP can be used as a screening analysis or a detailed analysis. Unlike many of the quantification methodologies, THERP provides guidance on most steps in the HRA process including task analysis, error representation, and quantification. THERP begins with system familiarization and qualitative assessment (task analysis and error identification). THERP can be used to analyze typical

^a The following Screening Values were used in the Space Shuttle PRA:

Available Time in minutes	Nominal HEP	1 Adverse Condition	2 Adverse Conditions	3 Adverse Conditions	4 Adverse Conditions
T<1	0.3	0.1	1	1	1
1<T<10	0.1	0.3	1	1	1
10<T<30	0.03	0.1	0.3	1	1
T>30	0.003	0.01	0.03	0.3	1

errors of omission and commission. It requires the analyst to construct a HRA ET to model the human error. The analyst then identifies the PSFs that affect human performance. THERP provides a list of three specified PSFs (training level, stress, and experiences) and allows the user to add additional PSFs. THERP allows the analyst to explicitly treat task-error dependencies and human recovery actions. THERP has five levels of dependency that can impact the overall probability of the error. THERP has a large base of experienced analysts in the U.S.; it has been applied to nuclear power plants, off-shore oil drilling, and the NASA Space Shuttle and ISS programs.

8.5.1.1 Task Decomposition

Following are the four phases of operator task analysis using THERP:

1. Familiarization.
2. Gathering plant-specific and event-specific information.
3. Qualitative assessment.
 - A. Performing preliminary task analyses including error identification.
4. Quantitative assessment.
 - A. Estimate the HEPs.
5. Incorporation into system risk and reliability model.
 - A. Perform sensitivity study to determine the impact on the system, and perform detailed analysis on risk-significant HEPs to ensure better estimate
 - B. Incorporate results into system model (e.g., PRA or reliability model).

Following are the ten steps to performing qualitative and quantitative assessment through analyzing the man-machine system:

1. Describe the system goals and functions of interest.
2. Describe the situational characteristics.
3. Describe the characteristics required of the personnel.
4. Describe the jobs and tasks performed by the personnel.
5. Analyze the jobs and tasks to identify error-likely situations and other problems.
6. Estimate the likelihood of each potential error.
7. Estimate the likelihood that each error will be undetected (or uncorrected).
8. Estimate the consequences of each undetected (or uncorrected) error.
9. Suggest changes to the system.
10. Evaluate the suggested changes (repeat steps 1 through 9).

To calculate the HEP for a task, THERP provides a number of activities for the analyst to identify the HEP's existence in the operator tasks being analyzed. Example analyst activities are: assigning screening values, considering PSFs, and quantifying the resultant HEP.

8.5.1.2 Screening

THERP provides guidance for assigning screening values for HEPs in two types of activity: diagnosis and (generally rule-based) action. The screening of diagnosis activity is based on

available time (Figure 8-2 and Table 8-1) [8-1, 8-4]. Screening values for rule-based behavior are shown in Table 8-2.

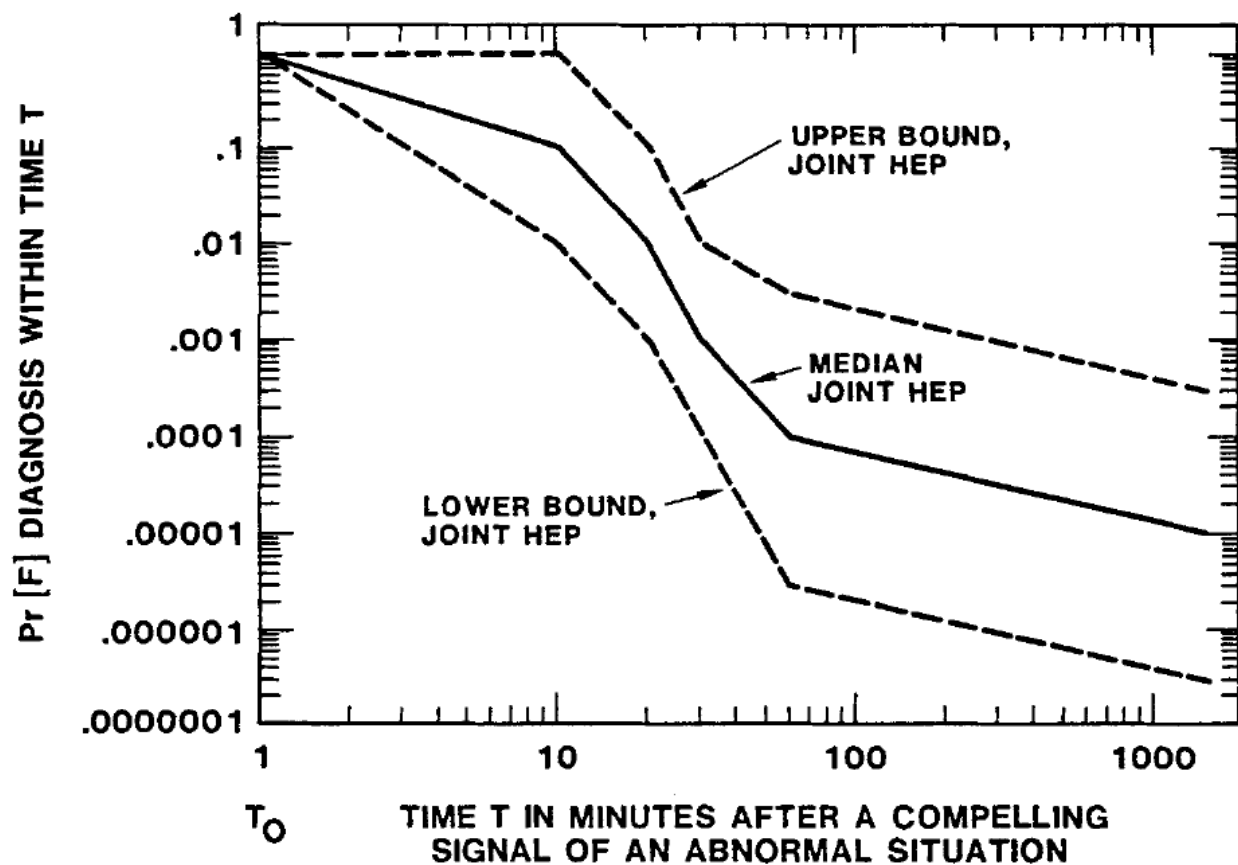


Figure 8-2. Initial Screening Model of Estimated Human Error Probability and Uncertainty Bounds for Diagnosis Within Time T of One Abnormal Event by Control Room Personnel.

Table 8-1. Initial Screening Model of Estimated Human Error Probabilities and Error Factors for Diagnosis Within Time T by Control Room Personnel of Abnormal Events Annunciated Closely in Time.*

Item	T (Minutes** After T ₀ ⁺)	Median Joint HEP for Diagnosis of a Single or the First Event	EF	Item	T (Minutes** After T ₀ ⁺)	Median Joint HEP for Diagnosis of the Second Event**	EF
(1)	1	1.0	--	(7)	1	1.0	--
(2)	10	.5	5	(8)	10	1.0	--
(3)	20	.1	10	(9)	20	.5	5
(4)	30	.01	10	(10)	30	.1	10
(5)	60	.001	10	(11)	40	.01	10
(6)	1500 (≈ 1 day)	.0001	30	(12)	70	.001	10
				(13)	1510	.0001	30

* "Closely in time" refers to cases in which the annunciation of the second abnormal event occurs while CR personnel are still actively engaged in diagnosing and/or planning responses to cope with the first event. This is situation-specific, but for the initial analysis, use "within 10 minutes" as a working definition of "closely in time."

Note that this model pertains to the control room crew rather than to one individual.

** For points between the times shown, the medians and EFs may be chosen from Figure 8-2.

+ T₀ is a compelling signal of an abnormal situation and is usually taken as a pattern of annunciators. A probability of 1.0 is assumed for observing that there is some abnormal situation.

++ Assign HEP=1.0 for the diagnosis of the third and subsequent abnormal events annunciated closely in time.

Table 8-2. Initial Screening Model of Estimated Human Error Probabilities and Error Factors for Rule-Based Actions by Control Room Personnel After Diagnosis of an Abnormal Event.*

Item	Potential Errors	HEP	EF
	Failure to perform rule-based actions correctly when written procedures are available and used:		
(1)	Errors per critical step without recovery factors	.05	10
(2)	Errors per critical step with recovery factors	.025	10
	Failure to perform rule-based actions correctly when written procedures are not available or used:		
(3)	Errors per critical step with or without recovery factors	1.0	--

* Note that this model pertains to the control room crew rather than to one individual.

8.5.1.3 Performance Shaping Factors (PSFs)

THERP provides a list of PSFs but gives no specific rules to assess the states of these PSFs and their effects on HEPs.

8.5.1.4 HEP Calculation Procedure

HEPs are calculated through a number of steps:

1. Analyze the event:

- A. Construct the HRA Event Tree (ET). For each branching point of the HRA ET, identify the likely human errors and the corresponding nominal HEPs as well as the uncertainty bounds.
 - B. Identify factors and interactions affecting human performance: Assess the effect of the performance shaping factors on the HEPs as well as the uncertainty bounds of the HEPs.
2. Quantify effects of factors and interactions:
 - A. Assess the levels of task dependencies based on the five-level dependency scale specified by THERP. Such dependencies would affect the task HEPs.
 - B. Account for probabilities of recovery from errors: Assess the possible recovery branches in the HRA ET and assess the success probabilities.
 3. Calculate human error contribution to probability of system failure:
 - A. Determine the success and failure consequences within the HRA ET and calculate the HEP of the HRA ET. The calculated HEP is used in the PRA model.

8.5.1.5 HEP Quantification

THERP calculates probabilities of the following types of errors:

- Screening and detection of system abnormalities
- Diagnosis and identification of the causes of system abnormalities
- Omitted actions, including actions in procedure preparation, use of a specified procedure (e.g., administrative control), execution of a procedure step, and providing an oral instruction
- Writing down incorrect information
- Acting on a wrong object; includes reading from an unintended display, acting at an unintended control, and unintended control (e.g., turn a control in the wrong direction).

8.5.1.6 Task Dependencies and Recovery

THERP provides five levels of dependency between two consecutive operator activities. These activities are represented by branches of an HRA Event Tree. The five dependency levels are zero dependency (ZD), low dependency (LD), moderate dependency (MD), high dependency (HD), and complete dependency (CD). Although the authors state that "There are no hard and fast rules for deciding what level of dependency is appropriate for any situation; considerable judgment is required" but the "time between tasks" is suggested as a key factor affecting the level of dependency. Recovery, by other crew or by automation, is explicitly covered. The framework therefore allows for explicit accounting of the impact of dependencies and recovery actions on the overall probability of error.

The following quantification model is proposed by Swain and Guttman for the above five levels of dependence:

$$\begin{aligned}
 HEP_N | HEP_{N-1} | ZD &= HEP_N \\
 HEP_N | HEP_{N-1} | LD &= \frac{(1 + 19HEP_N)}{20} \\
 HEP_N | HEP_{N-1} | MD &= \frac{(1 + 6HEP_N)}{7} \\
 HEP_N | HEP_{N-1} | HD &= \frac{(1 + HEP_N)}{2} \\
 HEP_N | HEP_{N-1} | CD &= 1.0
 \end{aligned}
 \tag{8-1}$$

where HEP_N is the HEP for Task N given an error was committed on Task N-1.

8.5.1.7 HEP Uncertainty Bounds

An error factor (EF) is used to represent the HEP uncertainty bounds. In THERP each activity is associated with a median HEP and an EF. The uncertainty bound of this activity is found by multiplying or dividing the median HEP by the EF. For example, assume the median HEP and EF for a certain activity are $1E-3$ and 10, respectively. The uncertainty bounds of this activity are ($1E-4$, $1E-2$). A lognormal uncertainty distribution is the default distribution used based on historical precedence. The lognormal has the feature that it is normal on a log scale, which is the natural scale for failure probabilities. Also most HEP uncertainties are given as lognormal. If there is rational for another distribution it should instead be used.

8.5.1.8 Suitability for NASA Applications

THERP task analysis and quantification schemes are best suited for routine tasks under normal conditions (e.g., proceduralized pre- and post-flight checks). Ground processing activities most closely match the situations for which THERP was developed. THERP does not address human performance in flight, zero gravity, or microgravity environments. THERP is useful for evaluating errors of omission but is difficult to use on continuous feedback HRA types of errors. THERP can also be useful in helping to define the uncertainty bounds of HEPs.

Other factors limiting the applicability of THERP include:

- Difference in typical time windows: The available time windows for action between nuclear power plant operation and aerospace missions are often significantly different. Recovery time windows for nuclear accidents typically vary from hours to days. In comparison, some action time windows and system response times in aerospace scenarios are very short, particularly those in the dynamic phases of space vehicle flight such as ascent, docking, and descent.
- Required Information: THERP quantification relies on specific characteristics of tasks and activities. This requirement limits the usefulness of THERP for application to new aerospace designs for which detailed system information is not available.

8.5.2 Cognitive Reliability and Error Analysis Method (CREAM)

CREAM [8-6] was developed for cognitive error analysis and is based on the Contextual Control Model [8-7]. CREAM can be used as a screening analysis or a detailed analysis. CREAM provides a list of fifteen basic cognitive tasks and their definitions to frame the cognitive error modeling. CREAM requires the analyst to perform task decomposition that breaks the task down into subtasks. Each subtask is matched to one of the pre-specified cognitive activities in

the list. For each subtask, the activity is further classified as an observation, interpretation, planning, or execution activity. Each of these activities has pre-determined error modes from which the analyst can select (e.g., wrong object observed). CREAM specifies 13 specific error modes which include both errors of omission and errors of commission. CREAM provides a basic HEP value and upper and lower uncertainty bounds for each generic error. CREAM provides a list of nine PSFs that can be used to modify the HEP. CREAM has a relatively large U.S. experience base and has been applied to nuclear power plants, off-shore drilling, and the NASA Space Shuttle and ISS programs.

8.5.2.1 Task Decomposition

CREAM identifies fifteen basic tasks (see Table 8-3) to decompose the human activities of interest.

Table 8-3. The Fifteen Cognitive Activities According to CREAM.

Cognitive Activity	General Definition
Co-ordinate	Bring system states and/or control configurations into the specific relation required to carry out a task or task step. Allocate or select resources in preparation for a task/job, calibrate equipment, etc.
Communicate	Pass on or receive person-to-person information needed for system operation by verbal, electronic, or mechanical means. Communication is an essential part of management.
Compare	Examine the qualities of two or more entities (measurements) with the aim of discovering similarities or differences. The comparison may require calculation.
Diagnose	Recognize or determine the nature or cause of a condition by means of reasoning about signs or symptoms or by the performance of appropriate tests. "Diagnose" is more thorough than "identify."
Evaluate	Appraise or assess an actual or hypothetical situation, based on available information without requiring special operations. Related terms are "inspect" and "check."
Execute	Perform a previously specified action or plan. Execution comprises actions such as open/close, start/stop, fill/drain, etc.
Identify	Establish the identity of a plant state or sub-system (component) state. This may involve specific operations to retrieve information and investigate details. "Identify" is more thorough than "evaluate."
Maintain	Sustain a specific operational state. (This is different from <i>maintenance</i> that is generally an off-line activity.)
Monitor	Keep track of system states over time, or follow the development of a set of parameters.
Observe	Look for or read specific measurement values of system indications.
Plan	Formulate or organize a set of actions by which a goal will be successfully achieved. Plan may be short-term or long-term.
Record	Write down or log system events, measurements, etc.
Regulate	Alter speed or direction of a control (system) in order to attain a goal. Adjust or position components or subsystems to reach a target state.
Scan	Quick or speedy review of displays or other information source(s) to obtain a general impression of the state of a system/sub-system.
Verify	Confirm the correctness of a system condition or measurement, either by inspection or test. This also includes the feedback from prior operations.

8.5.2.2 Screening

CREAM provides a two-level approach to calculate HEPs: the basic method and the extended method. The basic method is designed for task screening. It provides simple rules to

determine the HEP range for a task based on the combined PSFs' states. The HEP ranges for the four types of response modes of strategic, tactical, opportunistic, and scrambled are:

- $5E-6 < \text{HEP}(\text{Strategic}) < 1E-2$
- $1E-3 < \text{HEP}(\text{Tactical}) < 1E-1$
- $1E-2 < \text{HEP}(\text{Opportunistic}) < 5E-1$
- $1E-1 < \text{HEP}(\text{Scrambled}) < 1$.

8.5.2.3 Performance Shaping Factors (PSFs)

CREAM PSFs affect HEPs according to the type of basic human function, namely observation, interpretation, planning, and execution. These are shown in Table 8-4 along with their values.

8.5.2.4 HEP Calculation Procedure

The CREAM extended method is used for performing more detailed HEP assessments. The extended procedure includes the following steps:

1. Describe the task or task segments to be analyzed and perform task decomposition that breaks the task into a number of subtasks. Each subtask can be matched to one of fifteen pre-specified cognitive activities (Table 8-3).
2. Identify the type of cognitive activity for each sub-task.
3. Identify the associated human function of each sub-task. Four types of human functions are identified: Observation, Interpretation, Planning, and Execution.
4. Determine the basic HEPs for all sub-tasks. A number of failure modes are identified. Each failure mode is associated with a basic HEP and uncertainty bounds.
5. Determine the PSFs' effects on the sub-tasks' HEPs. Adjust the basic HEPs by multiplying by the adjustment factors based on the identified states of the PSFs.

Calculate the task HEP based on the HEPs of sub-tasks. When using the CREAM process to model cognitive errors, the most likely error to occur should generally be selected with the associated CREAM HEP.

Table 8-4. PSFs for Adjusting Basic HEPs.

PSF	PSF State	Type of Human Function			
		OBS	INT	PLAN	EXE
Adequacy of Organization	Very Efficient	1.0	1.0	0.8	0.8
	Efficient	1.0	1.0	1.0	1.0
	Inefficient	1.0	1.0	1.2	1.2
	Deficient	1.0	1.0	2.0	2.0
Working Conditions	Advantageous	0.8	0.8	1.0	0.8
	Compatible	1.0	1.0	1.0	1.0
	Incompatible	2.0	2.0	1.0	2.0
Adequacy of MMI and operational support	Supportive	0.5	1.0	1.0	0.5
	Adequate	1.0	1.0	1.0	1.0
	Tolerable	1.0	1.0	1.0	1.0
	Inappropriate	5.0	1.0	1.0	2.0
Availability of procedures/plans	Appropriate	0.8	1.0	0.5	0.8
	Acceptable	1.0	1.0	1.0	1.0
	Inappropriate	2.0	1.0	5.0	2.0
Number of simultaneous goals	Fewer than capacity	1.0	1.0	1.0	1.0
	Matching current capacity	1.0	1.0	1.0	1.0
	More than capacity	2.0	2.0	5.0	2.0
Available time	Adequate	0.5	0.5	0.5	0.5
	Temporarily inadequate	1.0	1.0	1.0	1.0
	Continuously inadequate	5.0	5.0	5.0	5.0
Time of day	Day-time	1.0	1.0	1.0	1.0
	Night time	1.2	1.2	1.2	1.2
Adequacy of training and preparation	Adequate, high experience	0.8	0.5	0.5	0.8
	Adequate, low experience	1.0	1.0	1.0	1.0
	Inadequate	2.0	5.0	5.0	2.0
Crew collaboration quality	Very efficient	0.5	0.5	0.5	0.5
	Efficient	1.0	1.0	1.0	1.0
	Inefficient	1.0	1.0	1.0	1.0
	Deficient	2.0	2.0	2.0	5.0

8.5.2.5 HEP Quantification and Uncertainty Bounds

CREAM provides a list of basic human activities along with HEPs and uncertainties by decomposing the analysis into a limited set of sub-tasks as defined by the basic human activities. Table 8-5 gives these values.

Table 8-5. Basic HEPs and Uncertainty Bounds According to CREAM.

Cognitive Function	Generic Failure Type	Lower Bound (5 percentile)	Basic Value	Upper Bound (95 percentile)
Observation	O1. Wrong object observed	3.0E-4	1.0E-3	3.0E-3
	O2. Wrong identification	1.0E-3	3.0E-3	9.0E-3
	O3. Observation not made	1.0E-3	3.0E-3	9.0E-3
Interpretation	I1. Faulty diagnosis	9.0E-2	2.0E-1	6.0E-1
	I2. Decision error	1.0E-3	1.0E-2	1.0E-1
	I3. Delayed interpretation	1.0E-3	1.0E-2	1.0E-1
Planning	P1. Priority error	1.0E-3	1.0E-2	1.0E-1
	P2. Inadequate plan	1.0E-3	1.0E-2	1.0E-1
Execution	E1. Action of wrong type	1.0E-3	3.0E-3	9.0E-3
	E2. Action at wrong time	1.0E-3	3.0E-3	9.0E-3
	E3. Action on wrong object	5.0E-5	5.0E-4	5.0E-3
	E4. Action out of sequence	1.0E-3	3.0E-3	9.0E-3
	E5. Missed action	2.5E-2	3.0E-2	4.0E-2

8.5.2.6 Task Dependencies

CREAM does not provide a specific procedure for identifying and accounting for task or error dependencies. Similarly, error recovery is not explicitly discussed.

8.5.2.7 Suitability for NASA Applications

The CREAM analysis units are “basic human activities” which are generic in nature. As a result, at the level of task description consistent with such basic human activities, the method can be applied to existing aerospace designs for both normal and emergency operations. For new aerospace design, since detailed task information is not available, CREAM's basic method could be used for screening purposes. The CREAM basic HEP calculation method provides HEP ranges for four control modes. The PSFs identified in CREAM may need to be expanded to include the PSFs experienced in the zero gravity and microgravity environments. CREAM has been used in two recent NASA HRAs (Space Shuttle HRA, and International Space Station HRA). The most recent version of the ISS PRA uses a combined THERP-CREAM modeling process for the HEPs not screened out.

8.5.3 Nuclear Action Reliability Assessment (NARA)

NARA [8-8] is a refinement of the Human Error Assessment and Reduction Technique (HEART) [8-9]. Even though it was developed for nuclear plant applications, it can be used for specific types of NASA applications. NARA can be used as a detailed analysis method and does not provide an explicit method for screening. NARA provides basic HEP values that apply to generic tasks, where they are adjusted based on a list of 18 PSFs [called Error Producing Conditions (EPCs)]. NARA covers both short- and long-duration activities by providing EPCs for longer duration tasks. NARA does not explicitly cover task dependencies or error recovery (these are included in the definition of the generic tasks). NARA's parent method HEART has been applied to a number of domains including chemical and weapons manufacturing.

8.5.3.1 Performance Shaping Factors

NARA provides a list of Error Producing Conditions (EPCs) and Effects, that are equivalent to Performance Influencing Factors (PIFs), Weights, or PSFs. Table 8-6 gives a partial listing. No causal model in terms of PIFs, their interdependencies, and other causal factors is provided.

Table 8-6. NARA EPCs and Their Effects (partial list).

NARA EPC ID	NARA EPC Description	NARA EPC Effect ¹
1	A need to unlearn a technique and apply one which requires the application of an opposing philosophy.	24
2	Unfamiliarity, e.g., a potentially important situation which only occurs infrequently or is novel.	20
3	Time pressure.	11
4	Low signal to noise ratio.	10
5	Difficulties caused by poor shift hand-over practices and/or team coordination problems or friction between team members.	10
6	A means of suppressing or over-riding information or features which is too easily accessible.	9
7	No obvious means of reversing an unintended action.	9
8	Operator inexperience.	8
9	Information overload, particularly one caused by simultaneous presentation of non-redundant information.	6
10	Poor, ambiguous, or ill-matched system feedback.	4
11	Shortfalls in the quality of information conveyed by procedures.	3
12	Operator under-load/boredom.	3
13	A conflict between immediate and long-term objectives.	2.5
14	An incentive to use other more dangerous procedures.	2
15	Poor environment.	8
16	No obvious way of keeping track of progress during an activity.	2
17	High emotional stress and effects of ill health.	2
18	Low workforce morale or adverse organizational environment.	2

1. The term "Weight" is used in the equations.

8.5.3.2 HEP Quantification

In NARA, a final HEP is calculated by the following equation:

$$HEP_f = HEP_i \times \prod_{j=1}^N \{ [Weight(PIF_j) - 1] \times State(PIF_j) + 1 \}$$

Where N is the number of applicable PIFs, and $0 \leq State(PIF_j) \leq 1$

The State (PIF) is assigned a value ranging between 0 (best, positive) to 1 (worst, negative). The basic HEP (BHEP) which is the first factor in the above equation is provided in tables. Table 8-7 through Table 8-10 give a partial listing of these basic HEPs.

Table 8-7. The Generic Tasks of NARA (partial list).

	Generic Task	Basic HEP
A1	Carry out simple single manual action with feedback. Skill-based and therefore not necessarily with procedure.	0.005
A2	Start or reconfigure a system from the Main Control Room following procedures, with feedback.	0.001
A3	Start or reconfigure a system from a local control panel following procedures, with feedback.	0.003
A4	Reconfigure a system locally using special equipment, with feedback, e.g., closing stuck open boiler solenoid relief valve (SRV) using special "gagging equipment". Full or partial assembly may be required.	0.03
A5	Judgment needed for appropriate procedure to be followed, based on interpretation of alarms/indications, situation covered by training at appropriate intervals.	0.01
A6	Completely familiar, well designed highly practiced, routine task performed to highest possible standards by highly motivated, highly trained and experienced person, totally aware of implications of failure, with time to correct potential error. Note that this is a special case.	0.0001

Table 8-8. The Generic Tasks of NARA for Checking Correct Plant Status and Availability of Plant Resources.

	Generic Task	BHEP
B1	Routine check of plant status.	0.03
B2	Restore a single train of a system to correct operational status after test following procedures.	0.007
B3	Set system status as part of routine operations using strict administratively controlled procedures	0.0007
B4	Calibrate plant equipment using procedures, e.g. adjust set-point.	0.003
B5	Carry out analysis.	0.03

Table 8-9. The Generic Tasks of NARA for Alarm/Indication Response.

	Generic Task	BHEP
C1	Simple response to a key alarm within a range of alarms/indications providing clear indication of situation (simple diagnosis required). Response might be direct execution of simple actions or initiating other actions separately assessed.	0.0004
C2	Identification of situation requiring interpretation of complex pattern of alarms/indications. (Note that the response component should be evaluated as a separate Generic Task.)	0.2

Table 8-10. The Generic Tasks of NARA for Communication.

	Generic Task	BHEP
D1	Verbal Communication of Safety-Critical Data.	0.006

8.5.3.3 Suitability for NASA Applications

NARA does not require detailed task related information for HEP estimation. This characteristic and the simplicity of use make NARA appealing for application to new aerospace designs. The NARA approach is also suitable for existing aerospace designs if the level of detail

offered by “generic tasks” adequately correspond to the task being analyzed. However, the number of NARA Generic Tasks is limited and most likely inadequate to cover all space mission activities. The challenge is in adapting and extending the generic tasks for NASA applications. Similarly EPCs and weight factors need to be calibrated for space applications.

8.5.4 Standard Plant Analysis Risk HRA Method (SPAR-H)

SPAR-H [8-10] is another method that has been developed for nuclear plant applications, but can be used for specific NASA applications. SPAR-H is a revision of the Accident Sequence Precursor (ASP) HRA screening method. SPAR-H can be used as both a screening method and a detailed analysis method. SPAR-H does not provide specific guidance on how to perform task analysis and error identification, but does tell the analyst to decompose each task to either a diagnosis or an action subtask. The method includes worksheets that allow the analyst to provide complete descriptions of the tasks and capture task data in a standard format. SPAR-H requires the analyst to determine the system activity type (power/operation or low power/shutdown) and then provides HEPs for the four combinations of the error type and system activity type (e.g., one combination is diagnosis and power/operation). The HEP is adjusted based on eight basic PSFs. SPAR-H also adjusts the HEP based on dependency. A dependency condition table is provided that allows the analyst to evaluate the same crew, time (close or not close in time), information cues (additional information or no cues), and location (same or different location). SPAR-H treats restoration and recovery tasks as separate events which are specified and analyzed. SPAR-H has a large U.S. experience base, has been applied to over 70 U.S. nuclear power plants, and has recently been used to help support the Nuclear Regulatory Commission’s Office of Reactor Regulation (NRR) Reactor Oversight Process.

8.5.4.1 Task Decomposition

SPAR-H decomposes a task into subtasks of “diagnosis” and/or “action.”

8.5.4.2 Screening

SPAR-H does not provide a procedure for screening.

8.5.4.3 PSF List

SPAR-H is based on an information processing model of human cognition, yielding a causal model of human error. SPAR-H also provides discussion of the interdependencies of PSFs, which are often ignored in other HRA methods. This being said, the interdependencies are not available to the reader in terms of correlation coefficients. The eight PSFs used by the method are:

- Available time
- Stress/Stressors
- Complexity
- Experience/Training
- Procedures
- Ergonomics/Human machine interface (HMI)
- Fitness for duty
- Work processes.

8.5.4.4 HEP Calculation Procedure

The SPAR-H HEP quantification for a specific activity includes the following steps^a:

1. Determine the plant operation state and type of activity:
 - A. Two distinctive plant states, at-power and low power/shutdown, and two types of activities, diagnosis and action, are modeled. Four HEP worksheets are provided for use in calculating the HEPs of the following four different combinations:
 - 1) At-power operation and diagnosis activity
 - 2) At-power operation and action activity
 - 3) Low power/shutdown operation and diagnosis activity
 - 4) Low power/shutdown operation and action activity.
2. Evaluate PSFs' states to determine the multipliers:
 - A. Tables are provided within the HEP worksheet for the analysts to check the most likely states of PSFs. For each worksheet, the analysis needs to identify the type of activity. Three types of activities are specified: diagnosis, action, and diagnosis-and-action. The base failure rates for these types of activities are identical for all worksheets. An HEP multiplier is assigned to each PSF's state. The HEP multiplier could have different values in different worksheets.
3. Two exclusive equations are provided to calculate the final HEP. The choice of one equation over another is dependent on the number of negative PSFs.

8.5.4.5 Error-Specific HEPs

HEPs are calculated for "diagnosis" and "action" failures. The SPAR-H authors have provided a comparison of the base failure rates with other HRA methods. These comparisons are shown in Table 8-11 to Table 8-13.

Table 8-11. Action Error Type Base Rate Comparison.

Method	Error Type Description	Base Rate (5th – 95 th percentile bounds)
SPAR-H	Action Task	0.001
NARA	D. Fairly simple task performed rapidly or given scant attention	0.09
	F. Restore or shift a system to original or new state following procedures, with some checking	0.003
CREAM	Tactical	0.001–0.1
THERP	Rule based actions of control room personnel after diagnosis, with recovery. EF=10	0.025

^a The steps that follow in 8.5.4.4 were developed for the nuclear power industry, therefore its use for space applications will require some adaptation.

Table 8-12. Diagnosis Error Type Base Rate Comparison.

Method	Error Type Description	Base Rate
SPAR-H	Diagnosis Task	0.01
CREAM	Tactical Control Mode	0.001–0.1
	Opportunistic Control Mode	0.01–0.5
THERP	Screening diagnosis. EF=10.	0.01
NARA	Miscellaneous task category “M,” no description in other tasks (A-H) fits diagnosis tasking as well.	0.03

Table 8-13. Mixed-Task Base Rate Comparison.

Method	Error Type Description	Base Rate
SPAR-H	Task involving both diagnosis and action	0.011
HEART	A. Totally unfamiliar, performed at speed with no real idea of likely consequences	0.55
	B. Shifts or restores system to a new or original state on a single attempt, without supervision or procedures	0.26
	C. Complex task requiring high level of comprehension and skill	0.16
	E. Routine, highly practiced, rapid task, involving a relatively low level of skill	0.02
	G. Completely familiar, well-designed, highly practiced, routine task occurring several times per hour, performed to highest possible standards by a highly motivated, highly trained and experienced person, totally aware of implications of failure, with time to correct potential error, but without the benefit of significant job aids	0.0004
	H. Responds correctly to system command, even when there is an augmented or automated supervisory system providing accurate interpretation of system state	0.00002
	M. Miscellaneous task for which no description can be found (Nominal 5th to 95th percentile data spreads were chosen on the basis of experience available suggesting log normality)	0.03
FRANCIE (5th-95th percentile)	1. Procedural Omission	0.0059
	2. Error of Intent	0.085
	3. Selection Error	0.015
	4. Awareness and Task Execution Related to Hazards/Damage	0.016
	5. Cognitive Complexity or Task Complexity Related	0.033
	6. Inspection/ Verification	0.097
	7. Values/Units/Scales/Indicators Related	0.022
	8. Maintenance/Repair Execution	0.041

8.5.4.6 Suitability for NASA Applications

SPAR-H classifies tasks into two types: diagnosis and action. Such a simple classification makes SPAR-H suitable for new designs. SPAR-H can also be easily applied to existing aerospace designs including both nominal and emergency situations. Before such application, the following concerns need to be addressed:

1. SPAR-H worksheets are designed for nuclear power operations, the worksheets need to be revised regarding the appropriate task description, operating conditions, and scope of PSFs and their corresponding weights. If the current PSFs are to be used, then the assignment of factors such as habitat factors, muscle wasting and bone density factors, cardiovascular factors, and other types of illness and their effects, need to be defined where appropriate. This applies to other HRA methods that use PSFs. The appropriate PSFs for the particular application need to be used.
2. Since SPAR-H does not provide guidelines for task decomposition, the analyst has the responsibility to identify how many diagnosis and/or action activities should be considered for a given task. This consequently affects the HEP of the task. The issue becomes more significant for new aerospace designs, where the allocation of tasks may be in development.

8.6 Guidelines on Uses of HRA Models

Compared to other HRA methods considered, the four HRA methods that have been described are: (1) relatively easy to use; (2) provide an explicit procedure for HEP estimation; and (3) do not require extensive information from task analysis (when used in their respective screening modes). There are significant differences between the selected methods, which may make each better suited to a different type of analyst or question to be analyzed. CREAM and SPAR-H use broader definitions of tasks making them easier to apply to a wider spectrum of aerospace activities. CREAM's set of generic tasks is defined based on human information processing. The set of generic tasks provided in CREAM covers a range of activities while remaining at a level of specificity that is easily adaptable to NASA mission tasks. CREAM's generic tasks may assist an analyst when a well understood task is to be performed in a new setting and, therefore, is a useful tool for screening. The provision of activities could allow the analyst to focus on the incorporation of PSFs that are unique to the NASA mission.

The task types used in SPAR-H, while also based on human information processing, characterize tasks in a much simpler way, by dividing tasks by emphasis on cognitive workload or physical workload (or potentially a combined rate). SPAR-H characterizes tasks as Diagnosis or Action, which can be generalized to any task, but is not as specific as tasks defined in CREAM or THERP. Therefore, SPAR-H is not as effective for screening analyses, but can be extremely powerful to assess the effects of performance-shaping factors on a task. Because NASA may wish to assess the potential for error in new mission tasks and activities (activities that may not have been performed 'for real'), the ability to estimate probabilities without having to specify exact tasks may be useful. In addition, more so than other methods, SPAR-H does not require that the analyst be familiar with human performance, just the task. Through its standardized process, it assures repeatability between analysts who have the same understanding of the task in question. Its results are easily reproduced, because it provides standard worksheets that ensure the analyst uses the same process each time.

NARA combines context characteristics and human tasks, and (like CREAM) defines a set of "generic tasks," also largely based on the human information processing model. These tasks can be generalized to match a subset of aerospace activities. Among the four methods, NARA has the most extensive use of real data to support its HEP quantification. One of the appealing

features of NARA is its use of actual human error data (i.e., CORE-DATA) in most cases. This contrasts with the other methods that are either totally or partially expert judgment based.

Finally THERP, when compared to the other three methods, is highly granular in its task decomposition (e.g., Opening a Valve). Treatment of human performance is much like treatment of the mechanical system, with significant emphasis on actions and much less emphasis (especially when compared to the other selected HRA methods) on cognitive aspects of performance. THERP relies on task granularity and a small number of PSFs. THERP is effective when the task is well understood, but the potential cognitive impacts on performance are not understood. Like CREAM, THERP can assist the analyst in the identification of potential human errors during screening analyses. The tasks used in THERP can be and have been generalized for use in non-nuclear power applications.

Different HRA methods may need to be combined to handle different aspects of the HRA modeling. For example, as was done in the Space Shuttle PRA, CREAM may be used as the general methodology with THERP being used for dependency treatments and uncertainties.

8.7 HRA Examples

Two HRA examples are presented in this section. Section 8.7.1 provides an example for a Post-Initiator HSI, and Section 8.7.2 provides one for a Pre-Initiator HSI.

8.7.1 Example for a Post-Initiator HSI

This example is based on information that was used in the Cassini PRA [8-11, 8-12]. The HSI of interest is defined as follows: the Flight Control Officer (FCO) diagnoses a launch vehicle guidance control malfunction and initiates the manual Command Destruct System (CDS).

The CDS is a part of the Flight Termination System (FTS) that requires a radioed command from the ground to activate destruct ordnance on a launch vehicle. The other part is usually called the Automatic Destruct System (ADS) and is typically activated by pull-lanyards or break wires running along the length of the vehicle. If the vehicle fails structurally and breaks up, the lanyard pulling or wire breaking serves as a trigger for ordnance activation, without the need for a human manual action (i.e., FCO to push the CDS button).

Definition of the HSI

This post-initiator HSI has two main elements:

- Cognitive response: FCO diagnoses system malfunction and initiates manual CDS, and
- Action response: FCO completes response by pushing the CDS button/control.

Since the action part is a simple, single, and fast task, and the CDS control button is well identified, the HEP associated with the FCO action is negligible compared with his or her cognitive failure probability. The HSI is included in the associated event tree. See the circled event in the Event Sequence Diagram (ESD), shown in Figure 8-3. This covers a simple task analysis for this HSI.

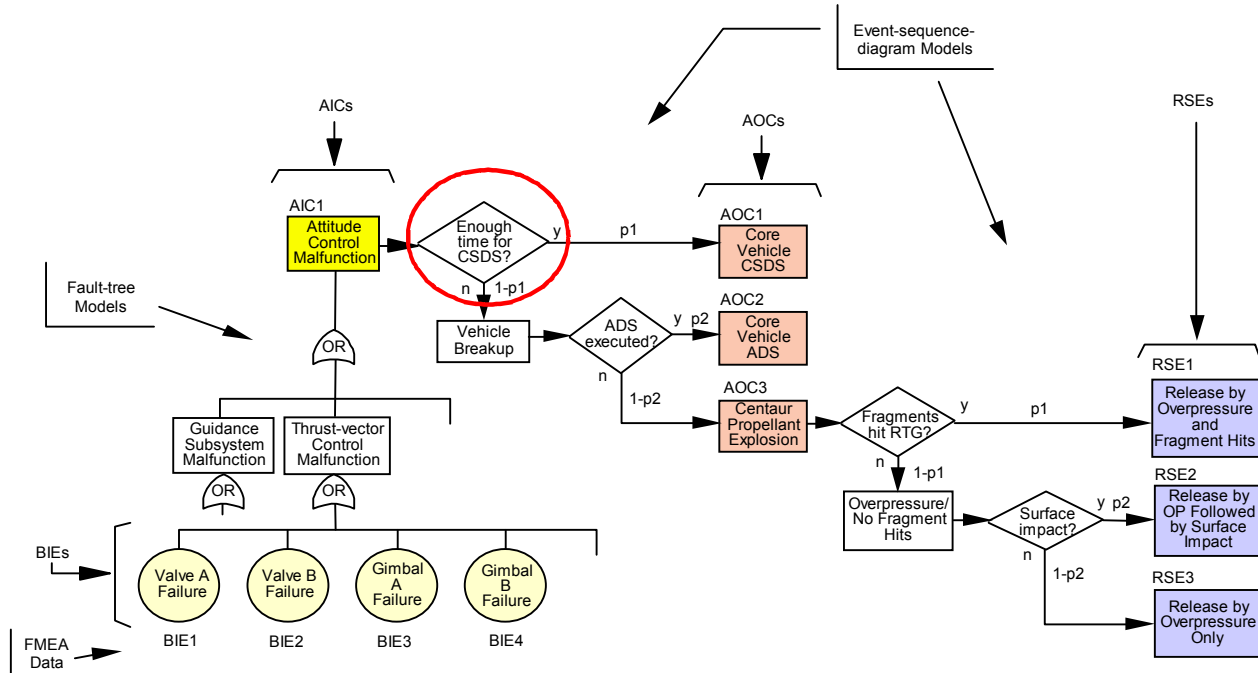


Figure 8-3. Example of Cassini PRA Fault Tree and Event Sequence Diagram Models^a.

HSI Modeling and Quantification

The FCO's cognitive response is time-dependent (i.e., FCO's response has to be completed within a given time window, T_w), hence a time response model is appropriate for HEP quantification [8-4, 8-8]. (Reference [8-13] also describes the general time-reliability model). One therefore needs to know the FCO's response time distribution.

The general time-reliability model, which can be used with any specific HRA modeling approach described earlier, is given as

$$\Pr(\text{Non-response in time } t) = \Pr(T_r > T_w) = \int_0^{\infty} f_{T_w}(t)[1 - F_{T_r}(t)]dt \quad (8-2)$$

where T_r and T_w represent "crew response time" and "available time window" for a specific HSI, and $f_{T_w}(t)$ and $F_{T_r}(t)$ represent the density function and cumulative probability distribution for stochastic variability of the above variables. The term $[1 - F_{T_r}(t)]$, which is the complementary cumulative probability distribution of crew response time, is usually known as a time response curve (TRC).

As a general example, if one assumes a lognormal distribution for T_r with a log median of $\mu = \ln(T_{1/2})$ ($T_{1/2}$ is the crew median response time) and log standard deviation σ , and if one assumes a constant window T_w for action, then

a. The ESD symbols used in this figure are different than the symbols used in this guide.

$$HEP = \Pr(\text{Non-response in } T_w) = \Pr(T_r > T_w) = 1 - \Phi[\ln(T_w / T_{1/2}) / \sigma] \quad (8-3)$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function (CDF). It must be noted that HEPs derived from TRCs with relatively large time windows may be extremely small (i.e., $<1E-5$). In these cases, one has to ensure other non-time dependent human errors are also accounted for, such as errors in executing actions, using the appropriate HRA modeling approach.

For the specific example being illustrated, due to lack of data on FCO's response time, expert judgment and engineering insights need to be used to develop an FCO response time distribution. The FCO cognitive response time can be broken down into two time elements:

- "CDS trigger delay": time at which an objective and observable "trigger condition" for CDS activation by the FCO occurs, e.g., vehicle reaches a certain angle of inclination. Note that this time is, in principle, not a part of the FCO's response time and just reduces the available time window. It varies with the type of failure scenario. An estimate of 0 to 6 seconds, depending on the initiating malfunction, is suggested in the Cassini PRA.
- "FCO response time": time taken by the FCO to observe/confirm system malfunction indications, diagnose the situation, and make a decision to initiate CDS. The FCO's response time depends on clarity of indications, human-machine interface, and guidance provided by emergency response procedures. An important factor in the FCO's decision making can be his or her reluctance to activate CDS before ensuring this is the last resort.

The Cassini PRA provides the following information on the FCO's response time distribution.

- A "best-estimate" FCO's response time distribution (in CDF format) is suggested as follows:

Time (sec)	Success Probability	Function
0 – 2	0	---
2 – 8	1	Linear

The FCO median response time ($T_{1/2}$) is estimated to be 3.5 seconds for the best-estimate distribution.

- An "upper-bound" CDF is suggested as follows to allow for time delays associated with (1) lack of visual clarity (0.5 sec), (2) eye movement and eye adjustment for different level of light (1.5 sec), and (3) flight-specific change in procedure and additional information (0.5 sec):

Time (sec)	Success Probability	Function
0 – 2	0	---
2 – 10	1	Linear

The FCO median response time ($T_{1/2}$) is estimated to be 6.0 seconds for the upper-bound distribution.

- The Cassini PRA also combines the "trigger delay" time and "FCO response time" distributions to come up with the following overall CDF for FCO to activate CDS:

Time (sec)	Success Probability	Function
0 – 2	0	---
2 – 15	1	Linear

The FCO median response time ($T_{1/2}$) is estimated to be 8.5 seconds for the overall FCO's response time distribution.

This distribution is believed to provide a reasonable representation of the overall FCO's CDS activation response time for a variety of failure scenarios and conditions. The graphical CDF for FCO's manual CDS initiation is presented in Figure 8-4. Mathematically, the TRC for CDS activation by FCO is expressed as follows:

$$\begin{aligned}
 HEP(t) = 1.0 - p(t) &= 1.0, & 0 < t < 2 \text{ sec} \\
 HEP(t) = 1.0 - p(t) &= 1.15 - 0.077t, & 2 \leq t \leq 15 \text{ sec} \\
 HEP(t) = 1.0 - p(t) &= 0.0, & t > 15 \text{ sec}
 \end{aligned} \tag{8-4}$$

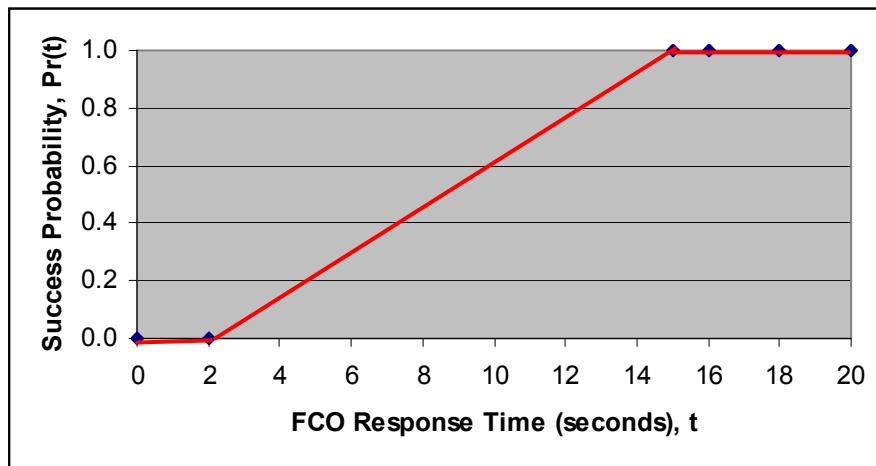


Figure 8-4. FCO's CDS Activation Time Cumulative Distribution Function.

To estimate the HEP for CDS activation by the FCO, one needs an estimate of the available time window (T_w). If, for example, $T_w = 10$ seconds, then

$$HEP = 1.15 - 0.077 * 10 = 0.38.$$

Finally, an uncertainty needs to be added as for all HEPs. In past PRAs, an error factor (EF) of 2 has been generally assigned to HEPs above 0.1, an EF of 3 for HEPs between 0.01 and 0.1, an EF of 10 for HEPs between 0.001 and 0.01, and an EF of 30 for HEPs below 0.001. This reflects the greater uncertainty with lower HEPs. The analyst, however, needs to determine the appropriate EF for the particular case being analyzed.

8.7.2 Example for a Pre-Initiator HSI

This section presents an HRA example for a Type A HSI based on limited available information. This HSI concerns the handling of a piece of optical equipment. The HSI of interest is defined as follows: "optical equipment failure due to human error."

HSI Definition

The potential human errors during the handling process and quality assurance (QA) are first identified. The process is briefly described as follows:

- In a clean room, the optical piece is first cleaned and then placed in a contaminate-free bag. The bag is then purged and sealed. Next, the bag is packed in a protective foam container, with humidity indicators, an accelerometer, and a desiccant. The accelerometer will record potential impacts to the optical equipment. There is an independent QA person who verifies the process using a formal sign-off procedure.
- Upon receiving the package, the box is opened and the package and equipment are examined for damage. The indicators are then examined to see if requirements have been exceeded. An independent QA person verifies the process.
- The item is typically stored in a similar environment or installed on the vehicle, which is in a clean room environment. Again, an independent QA person verifies the equipment storage or installation using a sign-off procedure.

Task Analysis

The THERP model [8-5] is used here for the task analysis. The following three main tasks are identified for this HSI:

1. Cleaning, bagging, purging and sealing bag, and packing optical equipment (i.e., packing task),
2. Examining the package and equipment for damage after opening (i.e., unpacking/examining task), and
3. Storing or installing optical equipment on the vehicle (i.e., storing/installing task).

The second task can be mainly treated as a recovery mechanism from potential human errors made during the first task. The following basic human errors (BHEPs) and recovery factors (RFs) are now defined for the above tasks:

Packing Task

- BHEP1 = Human error for improperly cleaning the optical equipment
- BHEP2 = Human error in bagging the equipment
- BHEP3 = Human error in purging and sealing the bag
- BHEP4 = Human error in packaging the bagged optical equipment
- RF1 = RF for visually inspecting the bag for damage (for BHEP2)
- RF2 = RF for independent QA checker verifying optical equipment cleaning, bagging, purging, sealing, and packing (for BHEP1 to BHEP4)

Unpacking/Examining Task

- BHEP5 = Human error in unpacking the equipment
- RF3 = RF for visually inspecting the bag for damage (for BHEP2, BHEP5)
- RF4 = RF for examining the accelerometer for a record of any impact (for BHEP4, BHEP5)
- RF5 = RF for examining humidity indicators for equipment moisture content requirements (for BHEP2, BHEP3)

- RF6 = RF for independent QA checker verifying optical equipment cleaning, accelerometer record, and humidity indicators' level (for BHEP1 to BHEP5)

Storing/Installing Task

- BHEP6 = Human error in performing equipment storage or installation
- RF7 = RF for independent QA checker verifying equipment storage or installation (for BHEP6)

With regard to performance shaping factors, it is assumed that all personnel are well-trained and have well-written procedures. Additionally, the working conditions and stress level are assumed to be optimal.

HSI Modeling and Quantification

Since the three main tasks (i.e., packing, unpacking/examining, and storing/installing) are in series, one can use the model that the occurrence of an error in any task will result in an error in the handling and quality assurance process. Therefore, the HEP can be expressed as follows:

$$HEP = \sum_{i=1}^I [(BHEP)_i \prod_{j=1}^J RF_{i,j}] \quad (8-5)$$

It is noted that all PSFs are unity in this case (i.e., nominal conditions are assumed). Using the human errors and RFs identified earlier in the task analysis, the HEP for this HSI is expressed as follows:

$$\begin{aligned} HEP = & BHEP1 * RF2 * RF6 + BHEP2 * RF1 * RF2 * RF3 * RF5 * RF6 \\ & + BHEP3 * RF2 * RF5 * RF6 + BHEP4 * RF2 * RF4 * RF6 \\ & + BHEP5 * RF3 * RF4 * RF6 + BHEP6 * RF7 \end{aligned} \quad (8-6)$$

Estimation. For illustration purposes, BHEP and RF estimates in NUREG/CR-4772 [8-1] and NUREG/CR-1278 [8-4] are used to quantify the HEP associated with handling of the optical equipment. The generic values used for BHEPs and RFs are summarized in Table 8-14. Following guidance provided in NUREG/CR-4772, zero dependence is assumed among errors, since the tasks are in series and it is assumed different QA personnel are used during packing, unpacking, and installation tasks (i.e., ZD is assumed). High dependence is assumed for multiple recovery possibilities during the unpacking task assuming the same person performs these tasks closely in time.

To simplify calculations, the estimates for BHEPs and RFs in Table 8-14 are used as mean values to calculate the overall HEP for this HSI. For more accurate estimate the medians in Table 8-14 can be converted to mean values. Using estimates in Table 8-14, the HEP for this HSI is calculated as follows:

$$HEP = 5.0E-4 \text{ (packing \& unpacking/examining tasks)} + 3.0E-3 \text{ (storing/installing task)} = 3.5E-3$$

Table 8-14. Generic BHEP and RF Estimates [8-1, 8-4].

BHEP/RF	Median	EF	Section	Source
BHEP1	0.03	5	8.7.2	NUREG/CR-4772
BHEP2	0.03	5	8.7.2	NUREG/CR-4772
BHEP3	0.03	5	8.7.2	NUREG/CR-4772
BHEP4	0.03	5	8.7.2	NUREG/CR-4772
RF1	0.1	5	8.7.1	NUREG/CR-1278
RF2	0.1	5	8.7.2	NUREG/CR-4772
BHEP5	0.03	5	8.7.2	NUREG/CR-4772
RF3	0.1	5	8.7.1	NUREG/CR-1278
RF4	0.5	< 2	8.7.2 (Assumed HD with RF3)	NUREG/CR-4772
RF5	0.5	< 2	8.7.2 (Assumed HD with RF3/RF4)	NUREG/CR-4772
RF6	0.1	5	8.7.2	NUREG/CR-4772
BHEP6	0.03	5	8.7.2	NUREG/CR-4772
RF7	0.1	5	8.7.1	NUREG/CR-4772

It is noted that the calculated HEP here could be on the conservative side, because the 0.03 estimate for BHEPs is rather conservative, as stated in NUREG/CR-4772. If more information becomes available to the HRA analyst, a more refined task analysis including task-specific PSFs can be performed. Having more information on tasks, procedures, and personnel, and using less conservative BHEP estimates, would result in a more realistic estimate of HEP. Also, human performance data on this and other similar operations (if available) can be used to estimate the BHEPs and RFs for various tasks.

It is observed that the HEP associated with the task of the optical equipment storage or installation on the vehicle dominates. Another recovery mechanism such as a post-installation test (if feasible) or a second QA checker for the installation task would help reduce the HEP. For example, if one considers a second independent QA person (with a 0.1 RF credit), then the calculated HEP would be reduced to 8.0E-4 from 3.5E-3. Finally, uncertainties (error factors EF) need to be assigned to the final estimate. These are obtained by propagating the EFs on the individual BHEPs and RFs.

8.8 References

- 8-1 A.D. Swain, *Accident Sequence Evaluation Program Human Reliability Analysis Procedure*, NUREG/CR-4772, U.S. Nuclear Regulatory Commission, 1987.
- 8-2 A.J. Spurgin, P. Moieni, and G.W. Parry, *A Human Reliability Analysis Approach Using Measurements for Individual Plant Examination*, EPRI NP-6560L, Electric Power Research Institute, 1989.
- 8-3 J. Rasmussen, *On the Structure of Knowledge—A Morphology of Mental Models in a Man-Machine Context*, RIS0-M-2192, RIS0 National Laboratory, Roskilde, Denmark, 1979.
- 8-4 A.D. Swain, and H.E. Guttman, *Handbook of Human Reliability Analysis with Emphasis on Nuclear Power Plant Applications*, NUREG/CR-1278, U.S. Nuclear Regulatory Commission, 1983.

- 8-5 F. T. Chandler, J. Y. H. Chang, A. Mosleh, J. L. Marble, R. L. Boring, D. I. Gertman, Human Reliability Analysis Methods: Implementation Guidance for NASA, NASA/OSMA Technical Report, July 2006.
- 8-6 E. Hollnagel, Cognitive Reliability and Error Analysis Method (CREAM), Elsevier, 1998.
- 8-7 E. Hollnagel, Human Reliability Analysis: Context and Control, Computers and People Series, Academic Press, 1993.
- 8-8 B. Kirwan, H. Gibson, R. Kennedy, J. Edmunds, G. Cooksley, and I. Umbers, Nuclear Action Reliability Assessment (NARA): A Data-Based HRA Tool, Safety and Reliability Journal, Manchester 25(2), 2005.
- 8-9 J. Williams, A Data-Based Method for Assessing and Reducing Human Error to Improve Operational Performance, IEEE Conference on Human Factors in Power Plants, Monterey, California, June 5-9, 1988.
- 8-10 D. Gertman, H. S. Blackman, J. Marble, J. Beyers, L. N. Haney, and C. Smith, The SPAR-H Human Reliability Analysis Method, NUREG/CR-6883, U. S. Nuclear Regulatory Commission, 2005.
- 8-11 C. Smidts, A. Mosleh, S. Swaminathan, S. Bell, K. Rudolph, B. Bream, and R. Mulvihill, CASSINI Accident Probabilistic Analysis Quantification Methods, PSAM-III International Conference on Probabilistic Safety Assessment and Management, Crete, 1996.
- 8-12 PRC Inc, Probability Analysis Validation Report, Prepared for Lockheed Martin Space Launch Systems for the CASSINI PRA, 1995.
- 8-13 R.E. Hall, J.R. Fragola, and J. Wreathall, Post Event Human Decision Errors: Operator Action Tree/Time Reliability Correlation, NUREG/CR-3010, U.S. Nuclear Regulatory Commission, 1982.

9. Software Risk Assessment

Software is a key component of modern space systems. Its widespread use goes hand in hand with the utilization of sophisticated digital and electronic devices whose functions it monitors and controls, and which in turn provide the interfaces through which practically all types of system hardware devices and components, and their associated functions, are monitored and controlled. Thus, in modern space systems software provides the “brains” for the accomplishment of both primary and ancillary launch vehicle, space vehicle, and supporting ground system functions, such as the propulsion of a rocket; the Guidance, Navigation, and Control (GN&C) of a spacecraft; or the data gathering and downlink of a science instrument.

Safety functions are also in large part entrusted to software. In case of incipient accident conditions, when the timeframe available for a needed action is too short to permit a human operator’s decision, the actuation of launch abort and system safety functions is determined by an intervention logic encoded in software. In crewed spacecraft, software is critical to the safety of the humans on board not only under potential accident conditions, but also under normal conditions, through the routine monitoring and control of the environment that permits crew survival and health, e.g., proper cabin pressurization, oxygen content, temperature, etc..

The rate of expansion from limited to more extensive applications in the use of software in space and other complex systems has been quite fast in recent years, but the state of the art in software assurance techniques, including reliability and risk assessment, has not kept up with this fast growth. In a majority of space programs and projects of the recent past, assurance of the proper functioning and safety of software has primarily relied on the line-by-line verification of code against requirements and on some amount of testing across the range of input conditions defined by designers as the “software operational profile.” From an assessment point of view, software reliability has often been assumed to be sufficiently high as not to require specific analysis. In the few cases where a quantification has been attempted, a reliability figure of merit has been typically estimated by means of a “black box model,” as will be discussed in some detail in Section 9.4.6.

Reliability and risk experts know well that complexity produces more opportunity for system anomalous behavior, or possibly even for catastrophic failures. In order to have sufficient confidence in the successful accomplishment of a mission where software plays a crucial role, designers and analysts need to have at their disposal for the assessment and assurance of correct software functionality, tools comparable in analytical and predictive capability to those that are routinely available and used for the evaluation of the typical hardware portions of the system. The perception of the technical community at large, even in the very recent past, has been predominantly driven by skepticism towards approaches that have been proposed by researchers and developers for the analysis and demonstration of software assurance. As a result, many programs and projects have assumed software contributions to mission risk to be negligible in comparison to hardware component contributions without carrying out a formal demonstration of safe performance, or have applied coarse methods of assessment that have

generally offered very limited insight into the level of assurance provided by the adopted flight software designs and implementations.

The material that follows is based on recent developments in software risk assessment techniques that have been successfully applied and validated in actual NASA space program environments. These techniques are linked together via one risk modeling framework which is referred to as “CSRM” (Context-based Software Risk Model). As will be discussed and illustrated in the following sections, this framework can accommodate a range of techniques, to suit the specific needs of the user and the specific characteristics of the system of interest, providing the analyst with the flexibility to adapt his/her approach to the nature of the problem at hand, as well as to the level of information and resources available to tackle it.

9.1 Concept of Software Risk and Related Definitions

Before entering into a technical discussion of the methods of analysis and assessment of software risk, it is useful to clarify the meaning given to the term “software risk” in the context of this chapter and the guidance it provides in the following sections.

In a space system mission we define as “software risk” the possibility that a negative mission outcome may be produced by a system malfunction that is either directly originated by a software fault, or to which a software fault is a determining contributor. “Determining contributor” means that, although other non-software-related malfunctions may be part of the scenario that produces the negative mission outcome, the latter would not occur if a critical software fault were not also present.

The concept of software risk is related to other concepts with which the user may be familiar. However, total equivalence cannot automatically be assumed for certain terms, when these are applied in the software risk domain. For example, one may think that the assessment of “software reliability,” as generally intended within the technical community, covers in practice the assessment of software risk. However this is only partially correct, if referred to the definition of reliability that describes it as the ability of a component to perform in full accordance and compliance with its design specifications (see Section 9.6). In fact, as will be discussed in the following, a majority of major mission failure contributions by software have been the result of erroneous or incomplete design logic and/or functional specifications. That is, in most of these situations the software was “reliable” according to a definition formulated in the terms just described, but its behavior did not represent a correct response to the circumstances that the system and its resident software were facing, and in fact was a determining factor of the mission failure that resulted from such circumstances (see Section 9.2 for more information on this specific subject).

Another way of stating the above is that the study and assessment of software risk and associated “software failures” does not exclude design errors and omissions. Moreover, because flight software is also the means of implementation of some important portion of the operational logic of a system, the investigation of possible software design error risk cannot remain confined to the implementation of software in actual computer code, but needs to be extended into the validation of the system design logic itself. Indeed, as the discussion in Section 9.2 shows, some of the recorded software related mission failures of recent memory would have occurred in the same exact way if the implementation of the faulty design logic had been carried out via hardwired electrical relays and/or analog electrical circuitry. The counterpoint to that, however, is that the implementation of design logic in hardware devices is self-limited in complexity by the nature of such devices, whereas software poses almost no limits to the variety and complexity of system control logic and algorithms that can be incorporated into

the design of a given system. As mentioned earlier, this possibility of unbounded complexity creates at the same time the potential for much broader ranges of system functionality and for more difficult-to-anticipate modes of system malfunction or failure.

9.1.1 Basic Definitions

A definition of *software risk* consistent with the concepts discussed above can be formulated as follows:

“Software risk is the possibility of events by which the software used within a system may fail to successfully execute its mission-critical or safety-critical system functions, under the conditions that are to be covered according to the concept of operation of the system designed to carry out the mission, and under the design envelope that is derived from, and consistent with, that concept of operation.”

The above definition is intentionally given in purely qualitative terms, to make a distinction between the definition itself and the parameters by which software risk may be quantified. These may vary according to the type of mission considered, but in general will include the probability of the events referred to in the above definition, complemented, for those situations where the events involved may have mission impacts of different kinds and magnitude, by a set of parameters that can provide an appropriate appraisal of such potential impacts and their severity if the events of concern do come true.

The definition also excludes from the realm of interest those software related events that do not affect mission-critical or safety-critical functions, i.e., events that have a peripheral impact on the mission. These may of course still be of interest from an overall system reliability viewpoint and accordingly addressed via appropriate analytical and assessment means.

It is finally noted that in the above definition the concept of “covering mission conditions” does not refer to an exhaustive coverage, in a combinatorial sense, of all the possible values of software input data and interacting system parameter states, but to the identification and recognition of all the key dynamic interactions and interfaces between the key mission-critical system states and conditions, and the desired software functional responses to these.

Beyond the software risk definition given above, a set of definitions is provided in Section 9.6, with the practical intent of providing the reader with an unambiguous interpretation of basic terms related to the contents of this chapter. Other somewhat different interpretations or uses of the same terms may be found in the technical literature and in different contexts. This is a necessary caveat, since it is generally recognized that no standard definitions exist for some of the terms frequently used in the software reliability and safety technical areas, as it is also discussed in recent technical journal articles (see for example [9-1]). Thus, no claims may be made as to the general validity and acceptance of these definitions and terminology beyond the context and boundaries of this Guide.

The discussion in Section 9.1.2 (immediately below) is specifically intended to assist the reader in the interpretation of some of the key definitions provided in Section 9.6.

9.1.2 Software Defects and Software Failures

A schematic representation and classification key for how defects or faults can be introduced in flight software during its development and production process is shown in Figure 9-1. The figure identifies the following principal phases of system and software development:

- A. Requirements Development & Analysis
- B. Software (SW) Module Specification Development

- C. SW Module Coding, Initial Verification & Validation (V&V) and Testing
 D. Operational Data Entry, Final V&V and Testing

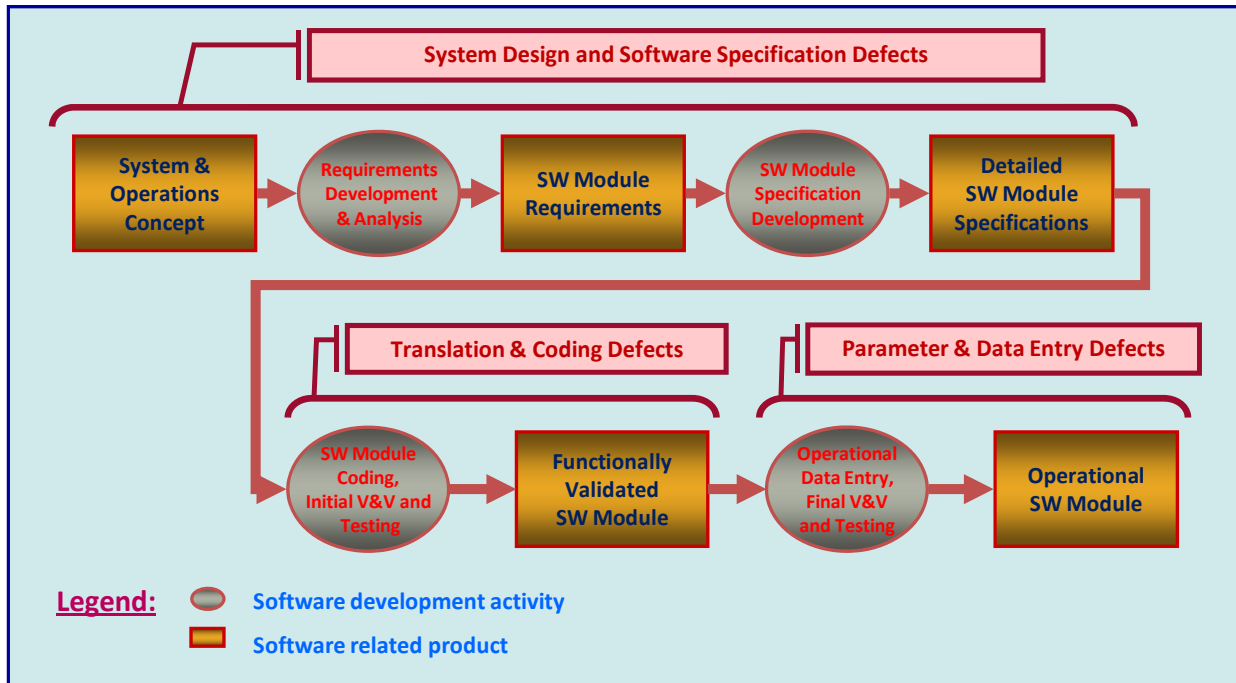


Figure 9-1. Software Defects by Development Phase.

The above phases of development proceed from the initial definition of a System and Operation Concept and incrementally generate software products, i.e.:

- Software module requirements, i.e., definition of software functions, as a Phase A product
- Detailed software module specifications, i.e., definition of specific software capabilities and interfaces, as a Phase B product
- Functionally validated software modules, i.e., functional but not mission-ready flight software, as a Phase C product
- Operational software modules, i.e., flight-ready software, as a Phase D and final product

Actual software developments may follow more complex variations of the above described process. However, the above representation serves the general purpose of clarifying that defects or faults can be introduced into a software module essentially in three basic forms, depending on the particular development phase when this may occur, i.e.:

Type 1 – Design and specification defects, i.e., defects introduced during the definition of functional and/or detailed specifications of a software module (Phase A and Phase B in the above representation of the software development process)

Type 2 – Requirement translation and software coding defects, i.e., defects introduced during the translation of software specifications into actual software code (Phase C of software development process)

Type 3 – Parameter and data entry defects, i.e., defects introduced during the “uploading” of mission specific parameters into a flight software module

With respect to the above the following observations are noteworthy:

- All three types of software defects are the product of “human errors” in corresponding software development activities.
- There is a formal distinction between the definitions of *software defects* and *software faults* on one hand, and *software anomalies* and *software failures* (jointly also generically referred to as “*software errors*”) on the other. The former are deviations from desired form and characteristics of the software specifications or code. The latter are the manifestation and effects of such deviations at execution time, that is, when the software containing the defect is executed in test or during the actual system mission.
- Software defects or faults that are present in a software module may or may not actually result in a software anomaly or failure during system test or operation. Whether this occurs or not depends on whether the defective portion of software is called upon and executed. If not called upon during the test process, a defect will remain dormant and undetected. If not called upon during system operation, a defect will remain dormant and inconsequential to the associated mission.
- The reader should be aware of the possible different meanings of the term “*error*,” as used in various contexts in the software engineering and software assurance literature. In general, the action by which a defect is introduced in software is routinely called an “*error*.” However, the execution of a software defect during system test or operation is also routinely called an “*error*.” In the first case the term refers to the human error committed in design, specification, coding, data entry, etc., whereas in the latter case the term refers to the software error occurring at execution time as a delayed effect of the former error.
- In cause-and-effect reasoning, the chain of events leading to a software failure typically develops as follows:
 1. The software designer or programmer commits an error (a “human error”)
 2. A software defect is introduced as a result of the above
 3. The software portion containing the defect is called upon and executed at mission time, which produces an anomaly or failure (a “software error”).

9.2 Lessons Learned from Software Failures in Space Systems

Table 9-I shows a compilation of major mission failures suffered by NASA in the decade from 1998 to 2007. From the table one can see that four out of a total of seven mission failures, i.e., more than half, were caused by a software fault, or had a software fault as a critically concurrent cause. In the same time period, four additional major space mission failures caused by software faults were suffered by the U.S. Department of Defense, and by major launch vehicle developers in Europe and the United States, bringing the total to eight failures in major space missions.

Table 9-1. Causes of Major NASA Mission Failures*, 1998-2007.

*Software related failures highlighted in color

1998 Lewis Spacecraft	<i>ACS and safe hold design</i>
1999 Mars Climate Orbiter Spacecraft	<i>Software / data specification error</i>
1999 Mars Polar Lander Spacecraft	<i>Landing control logic / software design error</i>
2003 STS Columbia	<i>Wing damage from detaching tank foam</i>
2004 Genesis Spacecraft	<i>Improperly installed gravity switches</i>
2005 DART Spacecraft	<i>GN&C software design errors / test oversights</i>
2006 Mars Global Surveyor Spacecraft	<i>Data uplink operator error / power management software design error</i>

The above data are relative to high visibility missions and do not include possible software failures or anomalies that may have occurred in the same time period in lesser missions, or missions by other nations in the international arena (e.g., China, Japan, etc.). Thus the role of software as a failure agent is not over-emphasized by this data, and may actually even be under-represented. It is also noted that the descriptions of the causes of failure reported in the table strictly reflect the conclusions documented in the official failure investigation reports relative to the listed events.

Two noteworthy pieces of evidence emerge from the data reported in Table 9-1 :

- A. *Software had a critical role in a large share (~ 60% in quantitative terms) of the failures suffered by NASA in high-stakes missions during the 1998 – 2007 decade.*
- B. *All these software related failures were rooted in software design faults, i.e., the software did exactly what it was designed to do, but this was not the correct action to be executed under the specific mission conditions encountered.*

The first piece of evidence dispels that notion that software can be assumed to have a negligible impact on the reliability of space systems, and that therefore the assessment of software risk can be considered unnecessary and not worth including within the scope of a system reliability and mission risk assessment activity.

The second piece of evidence suggests that traditional “software V&V,” i.e., the processes carried out in current software assurance practice to demonstrate software compliance with specifications – may be reasonably effective in identifying and correcting software coding and data faults and preventing associated failures, but not as effective in preventing software design and specification errors. This may be explained by the fact that V&V is in large part directed at verifying software compliance with specifications. In a majority of projects, the validation of the design and specifications themselves, especially the part of this concerning the “system design,” i.e., the logic of how the software is to interact with the balance-of-system, often falls in a gray area in between separate areas of design responsibility and because of this is not given as much attention as arguably it should. To illustrate the point, it is useful to examine the details of the Mars Global Surveyor (MGS) spacecraft failure, which actually occurred after about seven years of successful operation of the spacecraft in its orbit around Mars, i.e., well beyond the two year design life for which it had been designed. The following summary is taken verbatim from the MGS failure investigation report [9-2]:

Mars Global Surveyor Failure Investigation Report Summary

NASA's Mars Global Surveyor operated for ten years, longer than any other spacecraft sent to Mars. It pioneered the use of aerobraking, provided global mapping of the Martian surface, atmosphere, magnetic field and interior, and provided key imaging and communications support for subsequent missions. NASA extended its mission four times before its unfortunate loss. Key events pertaining to the loss of the spacecraft, whose last contact was on November 2, 2006, include:

- A modification to a spacecraft parameter, intended to update the High Gain Antenna's (HGA) pointing direction used for contingency operations, was mistakenly written to the incorrect spacecraft memory address in June 2006. The incorrect memory load resulted in the following unintended actions:
 - Disabled the solar array positioning limits.
 - Corrupted the HGA's pointing direction used during contingency operations.
 - A command sent to MGS on November 2, 2006 caused the solar array to attempt to exceed its hardware constraint, which led the onboard fault protection system to place the spacecraft in a somewhat unusual contingency orientation.
 - The spacecraft contingency orientation with respect to the sun caused one of the batteries to overheat.
 - **The spacecraft's power management software misinterpreted the battery over temperature as a battery overcharge and terminated its charge current.**
 - The spacecraft could not sufficiently recharge the remaining battery to support the electrical loads on a continuing basis.
 - Spacecraft signals and all functions were determined to be lost within five to six orbits (ten-twelve hours) preventing further attempts to correct the situation.
 - Due to loss of power, the spacecraft is assumed to be lost and all recovery operations ceased on January 28, 2007.

From the software reliability and risk perspective the MGS failure sequence can be summarized in the following key points:

- A "trigger event" constituted by an operator error caused some spacecraft control parameters to be corrupted and the spacecraft to eventually enter a "contingency orientation" that exposed one of the batteries to sun radiation heating.
- The spacecraft power management software, having been designed to interpret battery overheating as due to over-charging, shut down the charging of the overheating battery (action highlighted in blue font in the incident summary box above), which eventually caused the spacecraft to completely lose power.

While other faulty conditions contributed to the MGS failure, the power management system design logic was a crucial factor in the spacecraft loss, by its failure to account for external heating as a possible cause of increasing battery temperature. This MGS mission example is actually representative of a pattern that has also been observed in other software failure events recorded in recent years. The pattern presents itself as a scenario by which:

- A. *A mission event occurs, creating a particular operational condition for a system or mission;*
- B. *A software controlled function, having not been properly designed and or tested for the mission condition that has developed, responds in an incorrect or inadequate fashion, which in turn causes an un-recoverable mission failure.*

The "trigger event" referred to in A is sometimes a hardware anomaly or even an operator error as in the MGS case, but may also be a normal, although unanticipated, element of an operational mission sequence. Such was the case for the Mars Polar Lander failure, where,

according to the reconstruction of the failure investigation, the vibrations from the deployment of the landing gear were interpreted by the landing control software as a positive indication that the gear had touched ground, causing the software to command a premature shutdown of the retrorockets when the spacecraft was still at altitude over the Martian surface.

In summary, the review of the space mission software failure history provides us with some key insights for setting up and carrying out software risk assessment activities, as summarized below:

- *Traditional methods of software V&V and testing appear to have been reasonably effective in preventing “random” coding errors, or data entry errors at coding time.*
- *The same methods have not been as effective in uncovering design errors, as shown by the majority of software related mission failures, whereby critical faults had been introduced in design factors concerning the functional interfaces between key system functions and their control software.*
- *A significant fraction of software design failures have occurred during the operation of a mission under off-nominal conditions resulting from an anomaly, or under conditions that, although nominal for the mission, had not been fully anticipated or understood by the system designers.*

9.3 Classification of Software Failures for Risk Modeling

When referring to SW failures a variety of classification schemes are possible. In this section, however, simple classification terminology is introduced that is useful from the point of view of risk modeling, i.e., specifically for the purpose of distinguishing types of SW failures in terms of:

- a) Initiation mode,
- b) Outcome in mission terms.

9.3.1 Conditional vs. Unconditional Failures

In terms of the manner in which they are initiated, software failures can be classified as “unconditional” or “conditional.” The definitions associated with this classification are provided below:

Unconditional SW Failure:

A software failure is “unconditional” when it spontaneously occurs during the execution of a software function under nominal system conditions and after a successful, i.e., failure-free initiation of that software function.

Conditional SW Failure:

A software failure is “conditional” if it is logically associated with an *initiating condition* by which, before the occurrence of the software failure, the balance-of-system has entered a state that requires a non-routine type of control action by a software-implemented function.

With regard to *unconditional failures*, it can be said that these generally correspond to a failure by the software to correctly carry out some routine function. In a mature system design,

routine software functions are usually well understood and specified, thus unconditional failures are rarely caused by a design fault in systems that have been used for repeated missions; however, they have occurred relatively often with respect to other types of failures in the first mission of a new system^a. Unconditional failures caused by other types of faults (e.g., coding or data-entry errors not caught by V&V processes) or by external influences (e.g., a “single event upset” that alters the state of a memory bit) are possible but not common.

With regard to *conditional failures*, it is noted that system initiating conditions usually correspond to critical mission phases when the software has to correctly recognize, and respond to, the occurrence of specific system events or external events. These events generally belong to one of three broad categories, i.e.:

- a) Initiation of critical system maneuvers and actions, under potentially uncertain boundary conditions, e.g., the landing of a spacecraft in a previously unexplored planetary environment;
- b) Switch from one mode of system operation to another, based on mission time and/or other system events, e.g., staging of a launch vehicle, based on mission clock and/or measure of fuel tank depletion;
- c) Occurrence of a system hardware anomaly or failure, or of an anomalous external event affecting the system, by which a special software control action is to be undertaken, e.g., the failure of one thruster in a multi-thruster attitude control system, by which a different control logic and algorithm set has to be activated within the software.

9.3.2 Recoverable vs. Mission-critical Failures

In terms of their effects, software failures can be classified as “recoverable” or “critical.” The corresponding definitions are provided below:

Recoverable SW Failure:

A software failure is “recoverable” when it may not directly result in a system and mission failure, because of built-in fault-tolerance features of the system design by which the software function of concern can eventually be restored to a fault-free condition.

Critical SW Failure:

A software failure is “critical” when it cannot be isolated and controlled by built-in fault-tolerance features of the system design, and, if not compensated by other means, could result in a system and mission failure.

Space software and computer systems are commonly designed to be “fault-tolerant,” so that, if a fault is encountered in the execution of a software function, the software function can be momentarily suspended and/or switched to a redundant computer and software unit, and after some recovery routine/process is executed, the normal functionality is restored. This type of fault tolerance is achieved with relative ease, and is therefore fairly common, in the execution of certain types of system functions. For example, it is very common in the handling of data and communication relay functions by earth-orbiting satellites and planetary spacecraft. Other

^a Several failures of this kind have occurred in maiden flights of launch vehicles.

software-controlled functions that have to be executed in split-second or millisecond timeframes are not as easily designed to be fault-tolerant, and this is usually attempted only for very high-stakes functionality, e.g., the propulsion and attitude control of the Space Shuttle and other functions where human life and/or high value space system integrity is at stake.

9.4 Context-based Software Risk Model (CSRM)

The CSRM model is an approach to risk modeling and assessment that has been designed to address and account for the full range of potential software faults and failures that have been identified and recognized in the recorded history of space missions. More specifically, recognizing that traditional “black box” software reliability models (see Sections 9.4.6 and 9.4.6.4) do not well reflect the nature of software “conditional failures” (as defined in Section 9.3.1 above), the CSRM modeling framework has the flexibility to cover with appropriate logic models and quantification techniques these failures and “unconditional failures,” while identifying which among these failures are “recoverable” or “critical.”

The CSRM modeling approach focuses on the representation and assessment of software risk at the system functional level. It uses proven PRA techniques for systematic identification and analysis of key operational mission scenarios as the basis for developing its software risk models and associated quantitative estimations of software levels of assurance in relation to the mission-critical and safety-critical system functions.

The remainder of this chapter addresses the formulation, application, and practical utilization of the CSRM model and associated processes.

9.4.1 Conceptual Formulation

The CSRM model is based on the concept of “condition coverage,” i.e., an extension of the concept of “fault coverage” which has been widely used in the formulation of other software assurance and reliability assessment approaches. The reader may refer to Section 9.6 for definitions of these terms. Discussion of the concept of “condition coverage” that clarifies what is intended by the term, can be found later in Section 9.4.2.2 and in the example provided in Section 9.4.4.2.

Before discussing the CSRM formulations for software risk, it is useful to briefly discuss earlier attempts at modeling software risk. Typical past formulations have been based on an unconditional reliability model, by which software was assumed to have a failure rate λ_s in the time dimension, exactly like any system hardware component. In this paradigm, the software failure rate may be estimated by use of one of a variety of parametric or test-based techniques (see discussion in Sections 9.4.6.3 and 9.4.6.4). However, once this was done, one single overall unconditional probability of failure would be typically calculated for the entire flight software of a spacecraft, according to the standard reliability/unreliability formulations:

$$R_s = e^{-\lambda_s t} \quad (9-1)$$

$$POF_s = 1 - e^{-\lambda_s t} \quad (9-2)$$

In the above equations t is the mission time, λ_s , is the software failure rate in time, R_s is the software reliability, and POF_s is the software unreliability or probability of failure, which also represents the “software risk” for a mission time t .

The validity of the above formulation of software risk hinges in practice on two key assumptions, i.e.:

- A. The failure of software is driven by mission time, and can be represented by means of a failure rate in time.
- B. The failure behavior of a given piece of software is well represented by one average value of the failure rate, which can be estimated by testing the software across its normal “operational profile,” i.e., the range of normal inputs to the software during its nominal operation.

The above assumptions may be justified for certain types of large scale software applications where a “software failure” is merely an interruption of service with undifferentiated consequences, and, in fact, a seemingly random behavior in time, i.e. the type of failure behavior that lends itself to a failure-rate model, has been reported in the literature as characteristic of certain types of software failures (see for example [9-3]). However, neither of the above assumptions applies well to space system flight software executing critical, and well differentiated, functions whose failure may also often produce well differentiated immediate effects. Indeed, as discussed earlier in Section 9.2, the actual recorded history and experience relative to space missions shows without much doubt that space system software risk is not primarily driven by such semi-random types of failures, but by critical failures deterministically rooted in, and traceable to, software design and specification faults.

The CSRM model of software failure uses a logic and probabilistic formulation that can represent both “unconditional” and “conditional” software failures, as well as “recoverable” and “critical” ones. In the CSRM top-level representation of software risk, a generic software action can be represented as the “logic intersection” of two principal composing events, as expressed below:

$$\{\text{System Enters Condition "i"}\} \text{ AND } \{\text{SW Responds Correctly}\} \Rightarrow \{\text{Successful Software Behavior}\} \quad (9-3)$$

so that the corresponding risk scenario is also represented as the logic intersection of two composing events^a:

$$\{\text{System Enters Condition "i"}\} \text{ AND } \{\text{SW Responds Incorrectly}\} \Rightarrow \{\text{Software Failure}\} \quad (9-4)$$

With the notation :

F_{SW} = software failure event

MC_i = i -th type of mission condition entered by the system

R_{SWi} = risk scenario induced by software in relation to i -th type of mission condition (MC_i)

R_{SW} = overall mission risk induced by software,

the risk scenario in expression (9-4) can be formulated in logic symbolic terms as :

$$R_{SWi} = MC_i \cap (F_{SW} | MC_i) \quad (9-5)$$

^a The term “event” is to be interpreted in a flexible way in the context of this discussion. Either of the two events to which the discussion refers can actually be a combination of more events or conditions. The distinction that matters between the two types of events is between the scenario-initiating “system condition” that requires a certain type of “software response,” and the software response itself.

and the overall software induced risk is accordingly expressed as :

$$R_{SW} = \bigcup_{i=1}^N [MC_i \cap (F_{SW} | MC_i)] \quad (9-6)$$

The probabilistic formulations corresponding to expressions (9-5) and (9-6) are, respectively:

$$P_{SWi} = P(MC_i) \times P(F_{SW} | MC_i) \quad (9-7)$$

and

$$P_{SW} = \sum_{i=1}^N [P(MC_i) \times P(F_{SW} | MC_i)] \quad (9-8)$$

where the terms appearing in the equations are defined as follows:

$P(F_{SW} / MC_i)$ = conditional probability of software failure, given mission condition of type i

$P(MC_i)$ = probability that the i -th type of mission condition is entered by the system

P_{SWi} = unconditional probability of mission condition type- i occurrence accompanied by a software failure

P_{SW} = overall probability of software-induced mission failure.

9.4.2 Key Objectives and Characteristics of CSRM Application

The CSRM risk model and associated formulations are flexible and easily adapted to cover the range of software risk scenarios of potential interest, including the effect that software test processes may have in reducing risk. The following general considerations are relevant in this respect.

9.4.2.1 Software Function and Response Mode Identification Objectives

A key objective of the application of the CSRM conceptual formulation of software risk is to “partition” in a systematic and organized fashion the functional space of the software being investigated, so that its specific differentiated functions and “response modes,” i.e., the intended set of control and/or safety actions that it is designed to carry out in response to a defined set of system conditions, can be clearly identified at the very onset of the risk assessment process. The pursuit of this objective should come as no surprise to the PRA practitioner, as the identification of risk for any given system component requires as a first step the identification and understanding by the analysts of what that component is designed to do within the overall functionality of the system to which it belongs.

9.4.2.2 Condition and System Design Coverage

It was mentioned earlier that the CSRM formulation is conceptually based on the idea of “condition coverage.” This is directly related to the objective of software function and response mode identification discussed above. In the execution of software modeling and analysis, the “condition coverage” modeling approach mirrors the orderly software design process by which the system operational conditions that determine the need for differentiated software function and response requirements are identified as the first step of the design process that ultimately leads to detailed software requirements and specifications. Thus the systematic identification of system conditions and of the corresponding software functionality may be viewed as a “design

validation and verification” (Design V&V) contribution of the software PRA process as implemented via CSRM.

9.4.2.3 *Explicit Representation of Routine Software Functionality*

A possible misinterpretation of the CSRM formulation is that it may force a “conditional scenario” model for software risk onto situations where what is of interest is the possibility of software anomalies or failures that occur under normal system conditions, i.e., which may be viewed as occurring without any preceding trigger condition while the software is performing a routine function. This is not an issue, however, since the conceptual model expressed by Equations (9-5) through (9-8) also includes routine software functionality as a subcategory of a risk scenario, permitting its modeling and inclusion in an overall risk assessment by appropriate means. Such means may include formulations based on failure rate models, if pseudo-random models of software failure, as mentioned earlier in Section 9.4.1, are believed to be well suited to represent the behavior of certain “routine” portions of the software functionality.

In mathematical terms, the risk contribution of a software “routine function” can be expressed, in logic and probabilistic versions respectively, via the equations:

$$R_{swr} = MC_r \cap (F_{sw} | MC_r) \quad (9-9)$$

and

$$P_{swr} = P(MC_r) \times P(F_{sw} | MC_r) \quad (9-10)$$

where the notation retains the same meaning as in Equations (9-5) through (9-8), but the subscript “r” is now used to indicate that the associated entities refer to a “routine” type of mission condition.

The implication of referring to routine conditions is that in any given mission the probability of occurrence of such conditions is 1, i.e.:

$$P(MC_r) = 1 \quad (9-11)$$

so that

$$P_{swr} = P(F_{sw} | MC_r) \quad (9-12)$$

Thus, in practical terms, the portion of the software risk contribution that is modeled as being produced by routine software functionality is no longer conditional, in a probabilistic sense, on any preceding trigger-events, and is to be assessed accordingly, e.g., via models that consider the possibility of anomalies or errors during the cyclical execution of routine software functions.

One type of conditioning that may continue to exist in the execution of routine software functions is that associated with the different phases of a nominal space mission. As an example, different subroutines or modules of the GN&C flight software subsystem of a planetary spacecraft might hypothetically be activated in association with various types of orbital maneuvers in the vicinity of a target planet. This obviously would translate in the identification of more than one “routine condition” of software functionality, each with its own risk contribution per Equations (9-9) and (9-12), and an associated definition of the timeframe to which the risk contribution applies.

9.4.2.4 Time Dependency of Software Failure Probability

Some observations are useful with respect to the way time affects the formulation and quantification of CSRM software risk scenarios. From the discussion in Sections 9.4.1 and 9.4.2.3, it follows that, in the most general cases, the CSRM probabilistic formulation for software risk may be conceptually re-written as

$$P_{sw} = \sum_{r=1}^M [P(F_{sw} | MC_r)] + \sum_{i=1}^N [P(MC_i) \times P(F_{sw} | MC_i)] \quad (9-13)$$

The above formulation reflects the separation of the software risk contributions into two groups, one reflecting the existence of M routine regimes of functionality, and one reflecting the existence of N trigger conditions requiring non-routine software response and functionality. For any particular mission, the number of routine vs. non-routine conditions may vary. The number of routine conditions is typically defined and limited by design, whereas the number of potential off-nominal conditions that the system may enter is, in theory, open-ended. In practice, however, the system designer and the system safety engineers will have to make a deliberate decision with regard to the identification of a closed set of off-nominal conditions and events to be included in the system design basis, and for which, if they do arise in the course of a mission, the software will be called upon to functionally respond in some specified manner.

Once the sets of conditions, routine and off-nominal, which constitute the two groups in Equation (9-13) have been identified, the risk analysts have to decide what type of probabilistic model applies to each of them. In general, the routine function portion of the risk model may be addressed with the more or less traditional approach of pseudo-random software failure rate modeling, such as, for example, embodied in the group of probabilistic quantification models known under the label of "Software Reliability Growth Models" (SRGMs).

In a typical SRGM, the time dependence of the probabilistic formulation is usually included in a "failure rate" parameter estimation, which then translates into probability quantifications for any specified period of time of interest (see also Section 9.4.6.4).

The situation is usually different with regard to the off-nominal portion of the risk contributions. In the terms that appear in the rightmost portion of Equation (9-13), i.e., in each of the terms that are best individually expressed in the form represented by Equation (9-7), the time dependence of the probabilistic scenario is normally contained in the term $P(MC_i)$ that represents the probability of occurrence of the i -th off-nominal trigger-condition during the span of a given mission. On the other hand the remaining part of the contribution, i.e., the conditional probability of an incorrect software response to the condition MC_i , i.e., the term $P(F_{sw} | MC_i)$, is usually in its dominant part time-independent, because it reflects whether the software logic and algorithms are by design suited to respond correctly to the time-dependent chance condition MC_i , or not. In this respect, barring any kind of exotic self-modifying software, and considering individual mission phases during which no software design upgrade is loaded into executable memory, the probability of the software design correctness does not change as mission time elapses. Thus the terms $P(F_{sw} | MC_i)$ are usually to be addressed as time-invariant conditional probabilities, i.e., as a conceptual equivalent of a hardware component conditional probability of successful start of operation "on demand."

9.4.2.5 Models for Identification of Individual Risk Contributors

The formulations provided by Equations (9-6), (9-8), and (9-13) are very useful as a conceptual aid and foundation for effective investigation and analysis of software risk. In practical terms any such process will need to employ appropriate analytical means for the identification and quantification of the individual factors that appear in the equations, i.e., the

systems conditions MC_i for which software is called upon to respond, and the probability values $P(MC_i)$ and $P(F_{SW} | MC_i)$.

In a PRA context, the identification of the software risk terms of interest can be accomplished by seeking the identification of “cut-sets” that contain the SW related elements of interest. In general, given a PRA framework that is inclusive of software models, three types of cut-sets can be identified with regard to the identification of the terms of interest in Equations (9-6), (9-8), and (9-13), namely:

- A. Cut-sets that are not software related and thus do not identify any software risk contribution.
- B. Cut sets that identify a spontaneous software failure event occurring under “routine system conditions,” i.e., corresponding to the risk contributions $P(F_{SW} | MC_r)$ in Equation (9-13).
- C. Cut sets that identify a software failure event triggered by the occurrence of a system “trigger event,” i.e., a contingency, anomaly, or hardware failure condition MC_i that has occurred, and thus corresponding to the risk contributions $P(F_{SW} | MC_i)$ in Equation (9-13).

Once software-related “events” have been identified and included in the PRA framework of models, they will need to be quantified probabilistically if a full quantification of the PRA models is desired, as is the case in a standard PRA process. The CSRSM application process within a typical PRA framework, the choice of techniques for detailed analysis of software-related events, and important considerations for the probabilistic quantification of such events are discussed with examples in the following Sections 9.4.3 and 9.4.4.

9.4.3 Application Process

The general formulation of the CSRSM, as further refined via the observations discussed above, essentially states that the software contributions to overall mission risk may be conceptually classified into two basic categories, i.e., those originated by the unconditional failure of some routine software function, and those triggered by the occurrence of an off-nominal “trigger condition” to which the software is not able to respond correctly. As discussed in Section 9.2, the distinction between the two basic types of risk is important because earlier models of software failure did not properly address the conditional category, whereas the space mission data shows that in general this category accounts for a majority of recent failure events where software was a key contributor.

In practical terms a CSRSM application needs to be carried out as part of an overall PRA/PSA (probabilistic risk assessment / probabilistic safety assessment) process. Initial guidance for application of the process in the execution of Constellation Program PRA analyses is documented in [9-4] and will be followed by a more general NASA application guide which is in the making at the time this PRA guide goes to publication. This section defines the basic steps, and provides examples, of a typical CSRSM application process.

Generally speaking, CSRSM can be applied at different levels of detail, to match the system and software design information that is available at a particular stage of program and system development. In practical terms it is convenient to reduce this to the definition of two basic stages and forms of CSRSM application, which in the following are referred to as “*Specification-Level CSRSM*” and “*Design-Level CSRSM*”. In either case of application, the general steps of execution can be summarized as follows:

1. *Identify the mission-critical software functions.*
2. *Map the critical software functions to corresponding PRA model events.*
3. *Develop a set of associated logic models.*
4. *Identify, from the above models, the software-related cut sets for system and mission failure events.*
5. *Estimate the probability contribution from the software-related cut-sets to the system and mission failure events of interest. [This may include, at the top-level, the contribution to key risk metrics such as Loss of Mission (LOM) or Loss of Crew (LOC).]*

9.4.3.1 Specification Level CSR

A Specification-Level CSR analysis is applied in the early system design phase, and may typically make use of top-level system specification/design information and generic data to provide initial insight and preliminary definition of the risks that the software functions contribute to the overall system risk. In this context “generic data” means that software reliability and risk data gathered from other systems which are similar to the system of concern may be typically used in the CSR analyses, as a surrogate for software test and operational data specific to that system, which are usually not yet available in the early system design stages.

The development of the Specification-Level CSR models to be integrated with an existing PRA/PSA framework and set of system models will normally make use of information developed and documented for the relevant systems as part of the PRA/PSA activities, supplemented with information on software related functionality and software design characteristics, as available in system design documentation. A discussion and illustration of a Specification-Level CSR execution is provided in Section 9.4.4.1.

9.4.3.2 Design Level CSR

As the system definition and development progresses, more resolution is desired in a PRA. Accordingly, a Design-Level CSR analysis is executed at such a matured development phase, when more software design information and data are available.

A Design-Level CSR application builds on the insights gained from a Specification-Level CSR analysis. When more detailed system and software design information becomes available, this information is used in the design-level analysis to validate, augment, and refine the CSR models. The updated models decompose software risk into scenarios that include not only nominal mission operations, but also off-nominal conditions within the system design and mission planning envelope. This type of analysis can produce, in addition to more detailed qualitative and quantitative risk scenario information, risk-informed guidance for the execution of software testing, oriented towards reducing the risk of software failure or anomalies for specific scenarios of interest. To this end, it calls for coordination and cooperation between the PRA team and the software development team. In this mode of execution, CSR analysis results are provided by the PRA team to the software development team to “risk-inform” the software testing activity. The goal is to perform testing that targets specific scenarios identified via the CSR analysis, expanding the level of effort on testing consistent with the level of risk initially identified and associated with the conditions triggering those scenarios. The results of testing are then used to re-assess and quantify the risk scenarios of initial concern, with the ultimate goal of keeping the projected software risk contribution within acceptable bounds.

Risk-informed testing should be carried out based on the risk-priority of the scenarios identified in the updated PRA and CSRM analyses, and the risk-informed test results may validate the software design or even lead to software modifications and corrections if any faults are identified. When the test process has driven the software to an acceptable level of risk and reliability, all the information produced can be integrated back into the PRA and used for final quantification of the specific scenarios of concern.

The illustration of a CSRM Design-Level process is provided in Section 9.4.4.2.

9.4.4 Examples of Application

An explanation of the detailed steps of application of the CSRM process is better provided by example. As stated in Section 9.4.3 the essential steps of the process are the same for the two basic types of application that have been identified. The differences between the two are at a more detailed level of model development and quantification, reflecting the greater amount of information available once the design of the software functions has reached a stage of firm definition, and, even more importantly the ability to transfer information from the software risk assessment into the software testing activities and vice versa. The two following subsections describe, respectively, a typical Specification-Level and a typical Design-Level Application.

9.4.4.1 Specification Level CSRM Example

A Specification-Level CSRM analysis can be carried out at an early design phase to identify potential software risk contributors. The inputs required include general system and software design information, and, for quantification of risk, generic software failure and/or anomaly data from similar systems or projects. The example used here to illustrate its application pertains to the “Pad Abort – 1” (PA-1) test mission, which was analyzed as part of the Constellation Program activities, initially according to a conventional PRA process that contained only “placeholder,” undeveloped software-related events. The CSRM analysis was carried out at the “specification level” because only minimal information about the actual software design was available to the analysts at the time of execution.

The CSRM process has been expressly designed for application within the framework of a traditional PRA, i.e., a set of Boolean event-tree / fault-tree models that are being or have already been developed for the non-software portions of a space system. Consistently with this and with the introduction given in Section 9.4.3, the CSRM application steps may be carried out as follows:

1. Identification of mission-critical software functions.

In this step the analyst identifies and tags the mission-critical software functions in the PRA framework, using a reference mission event tree as a logic-model aid and, if necessary, complementary sub-event-tree and fault-tree structures linked to the former. The level of detail represented in these logic models is to be consistent with the level of available information that describes the software functions and their modes of execution. The objective of this analytical step is the identification, along the mission timeline, of all critical software functions at a consistent first-level of detail and in one documented set of models, without necessarily proceeding to duplicate the level of modeling detail of any pre-existing set of PRA models derived to analyze hardware and human operator functions. If no system-specific software information is available at the time the CSRM development is initiated, the mission event tree representation should by default include the software functions that are typically executed in a NASA mission of similar nature. In order to cover the full spectrum of risk scenarios, it is important that the identification of critical software

functions include all contingency and safe-mode operation that are anticipated within the mission and system design envelope.

Figure 9-2 shows a pictorial representation of the test mission, which consisted of an uncrewed test launch of the Launch Abort System (LAS) of the Orion spacecraft, and Table 9-2 shows the basic actions and maneuvers carried out in the mission sequence.

Figure 9-3 shows a Mission Event Tree that identifies the primary software functions directly associated with the successful execution of the key sequential mission maneuvers and actions. This simple analytical step provides the initial reference frame for the more detailed analytical steps that follow.

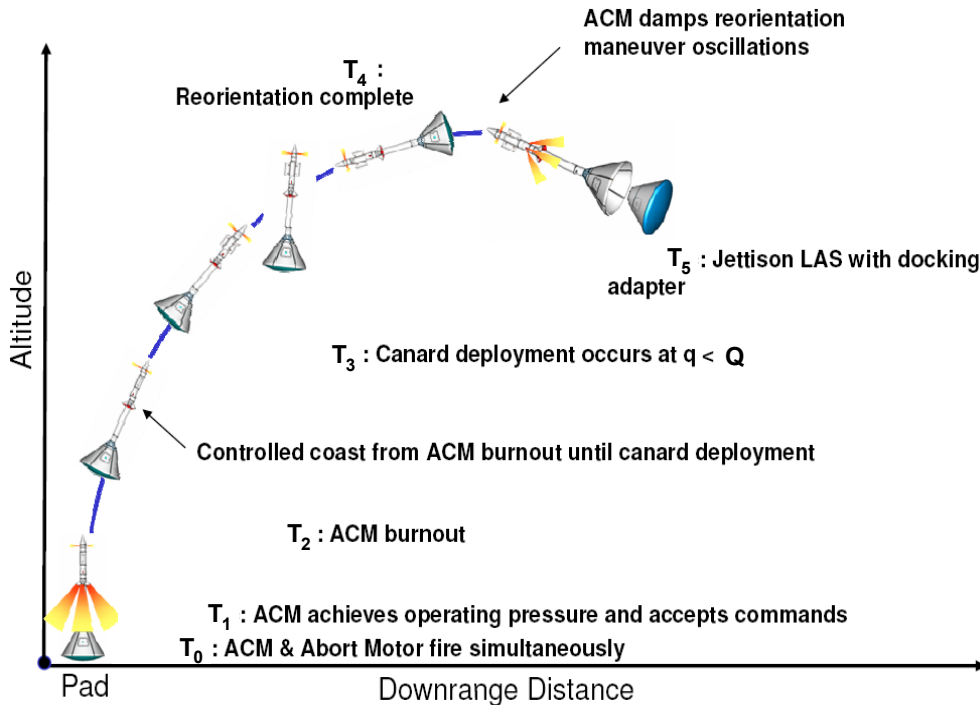


Figure 9-2. PA-1 Mission Sequence Illustration.

Table 9-2. System Actuations and Maneuvers in PA-1 Mission.

	Segment	Mode	Description	Commands Allowed	Discrettes Accepted
			ready to fly (abort) and the Abort discrete will be accepted		
Pad Abort	Abort Initiation	Flight	MM lights abort motor and abort control motor, uses ACM to control attitude	None	None
	Reorientation	Flight	Deploy canards, reorient	None	None
	LAS Jettison	Flight	Jettison the tower	None	None
Descent and Landing	Recovery System Deployment	Flight	Deploy the chutes	None	None
	Landing	Flight	Wait long enough that touchdown is assured, then change phase/segment/mode	None	None
Post-Flight	Chute Release	Chute Release	Cut chutes*	None	None
Shutdown	Safing and Shutdown	Safing and Shutdown	Shut down in an orderly manner	None	None

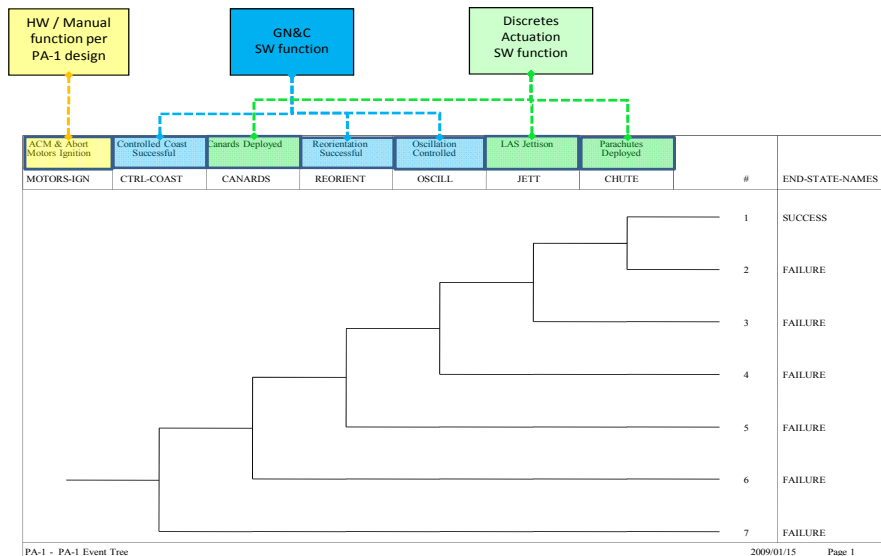


Figure 9-3. PA-1 Mission Event-Tree for Identification of Key SW Functions.

2. Mapping of software-functions to PRA logic-model events.

In this step it is assumed that a set of PRA models has been developed prior to the CSRM process application, i.e., essentially without inclusion of SW models, or with SW events only introduced as top-level “placeholders.” In this pre-existing PRA framework a systematic mapping is then created between the software functions identified and categorized in the preceding CSRM application step and appropriate “CSRM entry-point events” directly related to the execution of such functions. These events:

- a) either already exist and are identifiable in the pre-existing PRA models (e.g., because they are associated with a system function that also has critical non-software components, or because they were introduced in the initial PRA models as top-level “SW-event placeholders”);
- b) or they did not originally appear in such models structures, and may now be appropriately inserted.

Figures 9-4 and 9-5 show examples of entry points identified, respectively, in a detailed sub-event-tree and in a fault-tree model structure of the initially developed PA-1 Mission PRA.

Once an existing set of PRA models has been reviewed and appropriate entry-point events have been identified or added therein, with the aid of the software-function identification reference provided by the top-level CSRM Mission Event Tree (i.e., a mission event tree like the one provided as an example in Fig. 9-3), these events can be further analyzed and/or expanded via dedicated logic models, as discussed below. The quantification portion of the software PRA process is also addressed below, with more discussion and documentation provided in Section 9.4.6.

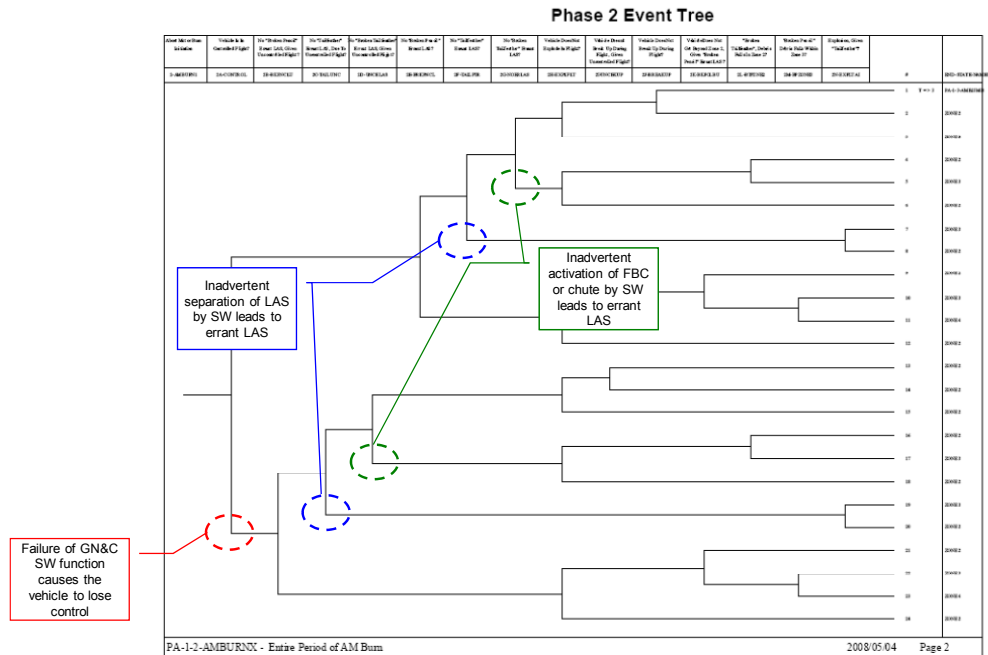


Figure 9-4. CSRM Entry-Point Events Identified in PA-1 PRA Event-Tree Model.

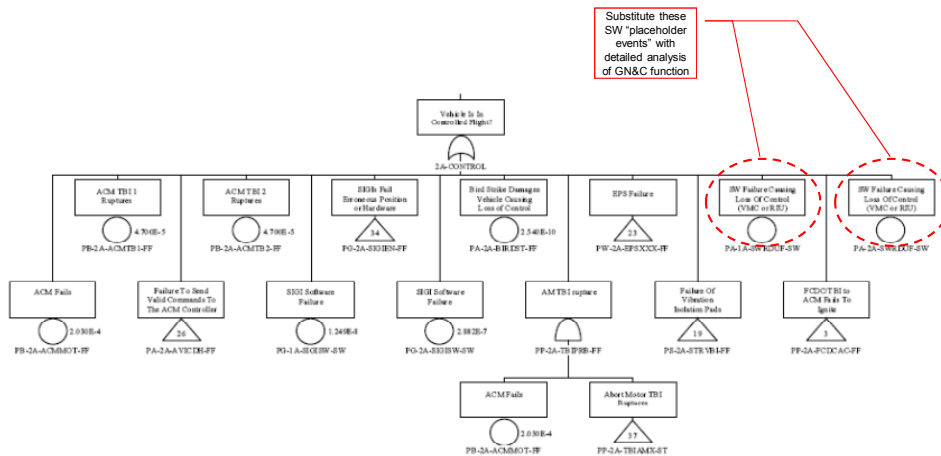


Figure 9-5. CSRM Entry-Point Events Identified in PA-1 PRA Fault-Tree Model.

3. *Expansion of CSRM entry-point events via dedicated logic models.*

In this step appropriate logic models are developed to analyze the critical software “entry-point events.” The level of modeling detail and sophistication employed for this objective may vary greatly, depending on the nature of the event and associated software function(s):

- At one end of the spectrum are events that may be treated as PRA “basic events” and therefore can be directly assessed and probabilistically quantified. Actuation trigger-events controlled by software, i.e., conditional events of the type “if measured variable V exceeds value x , then issue command to actuate hardware device D ,” often fall into this category.

- At the other end of the spectrum are events determined by the outcome of complex, time dependent software functions. The modeling of these events may require the use of special techniques. For example, use of the Dynamic Flowgraph Methodology (DFM) technique [9-5, 9-6] is recommended for the representation and analysis of the dynamic interaction between system and control-software variables and parameters at a level of detail and fidelity appropriate for the identification of the important system failure modes (i.e., the combinations of basic events and conditions that may result in an overall system function failure).
- In the middle of the spectrum are events that are driven by more than elementary software trigger-logic, but which are amenable to satisfactory representation and analysis by means of standard Boolean PRA techniques, such as binary fault-tree models and analyses.

Figure 9-6 shows an example of DFM model created to represent and analyze the GN&C function of the PA-1 system.

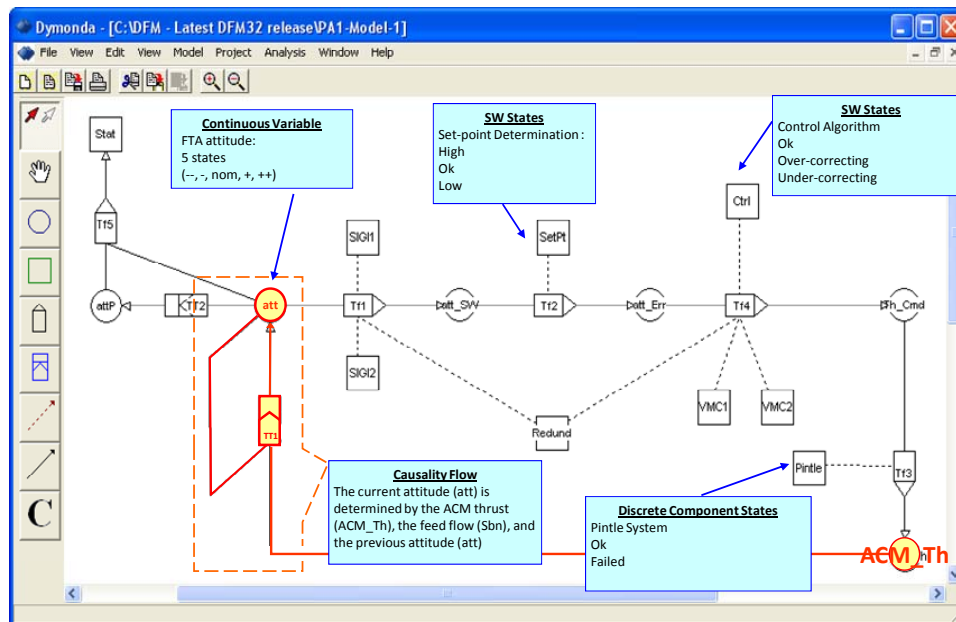


Figure 9-6. DFM Model of Pa-1 GN&C System.

It is beyond the scope of this guide to explain the details of DFM modeling, for which the reader is referred to the existing literature [9-5, 9-6]. In essence, a DFM model represents the key parameters of a system as nodes connected by cause-effect edges, and expresses in multi-valued logic and discrete-timing terms the behavior of a complex system which includes software control functionality. Automated analytical searches executed on a system model permit a systematic and thorough identification of time-dependent “cut-sets”^a for the failure or success of a given system function. Such DFM “cut-sets” represent

^a The term “cut-set,” although normally applicable to binary (Boolean) fault-tree logic is conveniently used here because it is very familiar to PRA practitioners. In multi-valued logic applications, the technically appropriate term is “prime implicant.”

combinations of individual hardware and software events, occurring in some time-sequence and order, which determine the success or failure of the system function of concern.

4. Identification of basic event cut-sets for CSRM entry-point events.

In this step the CSRM entry-point events are analyzed via the models developed for them in the preceding Step 3, in order to identify the corresponding basic-event cut-sets. No special analytical process is needed in the particular cases, mentioned in the brief discussion of Step 3 given above, where an entry-point event can itself be treated as a basic-event. For situations where the entry-point events are expanded into conventional binary-logic PRA models, standard cut-set identification techniques and software aids may be employed, whereas in the cases where the more complex multi-valued, dynamic DFM logic models are utilized, specific deductive analysis techniques are available to the analyst via DFM implementation software such as DYMONDA™ [9-7].

Figure 9-7 shows the principal cut-sets identified by the DFM analysis of the PA-1 system GN&C function (the full list, including conditions determined by more than two simultaneous component failures, is not shown here).

#	Prime Implicant	Probability
1	Pintle system failure	xxx
2	GN&C set-point high AND Control algorithm normal	xxx
3	GN&C set-point low AND Control algorithm normal	xxx
4	SIGII biased high	xxx
5	SIGII biased low	xxx
6	GN&C set-point high AND Control algorithm over-correcting	xxx
7	GN&C set-point low AND Control algorithm over-correcting	xxx

Figure 9-7. DFM-Produced Cut-Set for Failure of Pa-1 GN&C Function.

It is noted that, because the DFM model shown in Fig. 9-6 includes both the software and hardware components of the GN&C subsystem, the cut-sets obtained from it include not only software-related failure scenarios, but also conditions caused exclusively by hardware component failure(s). In fact, generally speaking, a DFM analysis based on a model that represents both software and non-software elements, will yield results that, from a CSRM point of view, can be classified as belonging to the three distinct categories of risk contributions defined and discussed earlier in Section 9.4.2.5, i.e.:

- A. “non-software” cut-sets, i.e., cut-sets that imply only hardware, human operator, or other non-software failure events;
- B. “software only” cut-sets, i.e., cut-sets that correspond to software failures spontaneously occurring under “system routine conditions”;
- C. “non-software-triggered, software-related” cut-sets, i.e., cut sets that represent “conditional” software failures triggered by a non-software event.

The examples of PA-1 system cut-sets shown in Fig. 9-7 are of type A or B. The absence of type C cut-sets is not surprising, because these are typically associated with software contingency modes of operation, and the PA-1 system and mission did not contemplate, at the specification-level of the information that was made available for the CSRM analysis,

any such modes of software functionality triggered by potential “balance of system” anomalies or failures. The cut-sets of strict CSRM relevance generated by the DFM analysis were, therefore, all of type B. With regard to DFM-generated type A cut-sets, although outside the primary focus of a CSRM application, they do provide a check and validation of the hardware-oriented analyses that may have previously been performed via traditional PRA techniques.

It is finally noted that, besides DFM, another process capable of logically expanding a given entry point event and providing its logically equivalent representation in PRA cut-set-compatible format could, if available, be used to carry out the CSRM Steps 3 and 4 described here.

5. Probabilistic quantification of software contribution to system and mission risk.

In this final step of the CSRM application process, CSRM entry-point event probability estimates are obtained via quantification of the corresponding cut-sets, providing, in turn, overall estimates of the software contribution to system and mission risk. The quantification may proceed differently, depending on the nature of the software cut-sets, i.e., whether they are of “type B” or “type C” and on the type of applicable data which are available. Generally speaking, the quantification of potential software failure events in a Specification Level CSRM application cannot count on the availability of system-specific test data, since at the early developments stages when such an application is typically carried out, software modules rarely exist in an executable and testable form. Thus, any risk quantification will usually utilize “surrogate data,” i.e., failure and anomaly records from software systems similar in nature to the one(s) of interest, which have been developed and used in earlier projects and missions, and for which data have been collected and compiled in usable form.

A survey and discussion of the software reliability and risk quantification techniques generally available as possible options to the analyst is provided in Section 9.4.6. For clarity of presentation and explanation, the software probability quantification topic is better handled as one subject, and it is therefore preferable not to disperse it in non-contiguous sections of this chapter. For this reason, the discussion in Section 9.4.6 actually covers both the quantification processes and models that may be available for use at the Specification-Level of an application, i.e., when usually only “surrogate data” are accessible for a given type of software, and those that can be used in the Design-Level stage, when system-specific software test data normally become available.

In this section the discussion deals with aspects of a typical Specification-Level analysis that are significant for the choice of an appropriate quantification model out of the possible options. In this regard, the categorization of risk contributors and system-failure cut-sets first discussed in Section 9.4.2.5, and also addressed immediately above in step 4, produces the following important observation: for type B cut-sets, a pseudo-random failure rate model probability may be applicable for probabilistic quantification of the software event(s) in those relatively common situations when the failure event of concern lends itself to a “failure in time,” rather than a “failure on demand” representation; however, this is normally not the case for type C cut-sets. In the latter, in fact, the time dependence of the risk contribution is normally in the “trigger event” probability, whereas the software response is quantifiable in terms of a conditional probability that is usually not time-dependent. Thus, it can generally be asserted that the software event(s) in type C cut-sets are quantifiable via “failure on demand” conditional probability models, not by unconditional time-based failure rate models.

An additional observation, which is also valid with regard to a Design-Level application, applies to the situations where the analysis and quantification of a CSRM entry-point event is carried out via an expansion of the event itself by means of a dedicated model (i.e., a

DFM, or other type of model and associated analytical process). Under such circumstances, the following options are available, at the analyst's discretion, for the final representation of the expanded event within the PRA framework:

- A. the entry-point event can be treated in the existing PRA as a "basic event," with the associated probability transferred as the results of a DFM analysis (or other equivalent analysis) are separately carried out and documented;
- B. the entry-point event is linked in the existing PRA to an underlying fault-tree model constructed and making use of the cut-sets and associated cut-set probabilities obtained from the DFM or other equivalent analysis.

The above two options are equivalent in risk quantification terms, but Option B transfers more detailed analytical information directly into the structure of the existing PRA models, in the traditional binary logic-model format.

9.4.4.2 Design Level CSRM Example

A Design-Level CSRM analysis is carried out at a stage of the system development when relatively detailed system and software design information are available, and software test data may also have been collected and compiled in a form suitable for use in the estimation of software reliability and risk metrics. This type of analysis may be an iteration and refinement over a specification-level analysis carried out at an earlier stage; or be executed without a preceding interim analysis. The first mode of execution is the preferred one when a mission or project is being assessed from its inception, to maximize any opportunity for positively influencing the system design with the insights gained from the PRA and CSRM activities.

In terms of process flow, a CSRM Design-Level analysis follows the same steps illustrated for the Specification-Level analysis in the preceding section. As will be apparent from the discussion in this section, the principal differences between the two types of application are primarily in Steps 3 and 5, i.e., in the modeling and quantification steps. The more complete and detailed information that becomes available after the initial stages of system development may in fact have a significant impact on how these steps are carried out.

The examples in this section are based on the reference mission of a mini-spacecraft called "Mini AERCam." This system consists of a video-camera and associated recording and video-transmission devices, hosted in a spherical body equipped with thrusters and a GN&C function. The GN&C function utilizes input from a GPS receiver and onboard gyros. This input information is then elaborated by autonomous control software algorithms to actuate the thrusters in such a way as to execute translational/rotational motion and station-keeping necessary to execute video display and recording requests by Space Shuttle or International Space Station astronauts. An illustration depicting the spacecraft and the thruster set-up arrangement is provided by Figure 9-8.

For completeness we discuss below all the Design-Level CSRM execution steps, indicating which ones are essentially identical to the corresponding Specification-Level steps, and which ones differ in some substantial way. A detailed discussion and illustration with the Mini AERCam example system is provided for the latter.

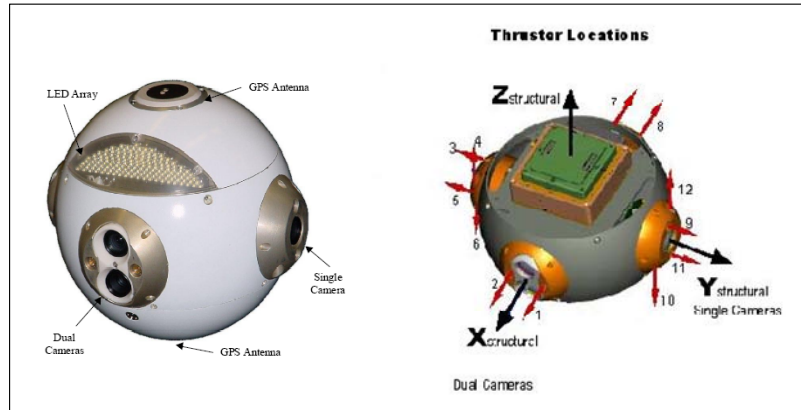


Figure 9-8. Mini AERCam Spacecraft and Thruster Arrangement.

1. Identification of mission-critical software functions.

This step is carried out in a fashion essentially identical to a Specification-Level application. If at the current stage of system development new information has emerged by which key software functionality has been added or modified for the mission of concern, any previously developed top-level CSRM mission event-tree will have to be updated accordingly.

For the Mini AERCam a typical mission consists of the following phases:

1. Release from the docking bay.
2. Autonomous control of the Mini AERCam to reach the vicinity of the target position.
3. Autonomous station keeping to maintain relative position with the target, so as to carry out the video capture and transmission functions.
4. Autonomous control of the Mini AERCam to return to the docking bay.
5. Retrieval of the Mini AERCam into the docking bay.

A top-level mission event tree may be drawn for such a mission as shown in Figure 9-9.

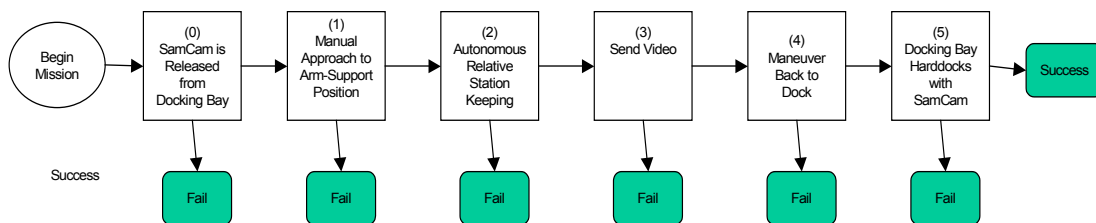


Figure 9-9. Mini AERCam Mission Event Tree.

2. Mapping of software-functions to PRA logic-model events.

This step also closely resembles the corresponding step of a Specification-Level application. The only difference may be that, because of the greater level of detail and maturity of the conventional (i.e., non-software oriented) PRA models already developed for the system of interest, the number of potential CSRM entry-point events identifiable in those models may be higher than in an earlier Specification-Level application. In the Mini AERCam application, given the extent of the software control functions implemented in the system, a number of entry-points would exist in any associated, conventionally developed set of PRA models. However, this particular application was intended from the start to be a “deep-dive,” but limited breadth, demonstration project. Thus, the analysis was directed to concentrate on entry-point events associated with the GN&C system function.

3. Expansion of CSRM entry-point events via dedicated logic models.

As mentioned, the Mini AERCam GN&C function relies on a complex time-dependent interaction of hardware and software components. Therefore it was appropriate to utilize the Dynamic Flowgraph Methodology (DFM) for its modeling and failure analysis. Given its complexity, a full Mini AERCam model including both the GN&C software and hardware components (electronic and mechanical), as well as interfacing subsystem components, was developed in modular fashion, as illustrated by Figures 9-10, 9-11 and 9-12.

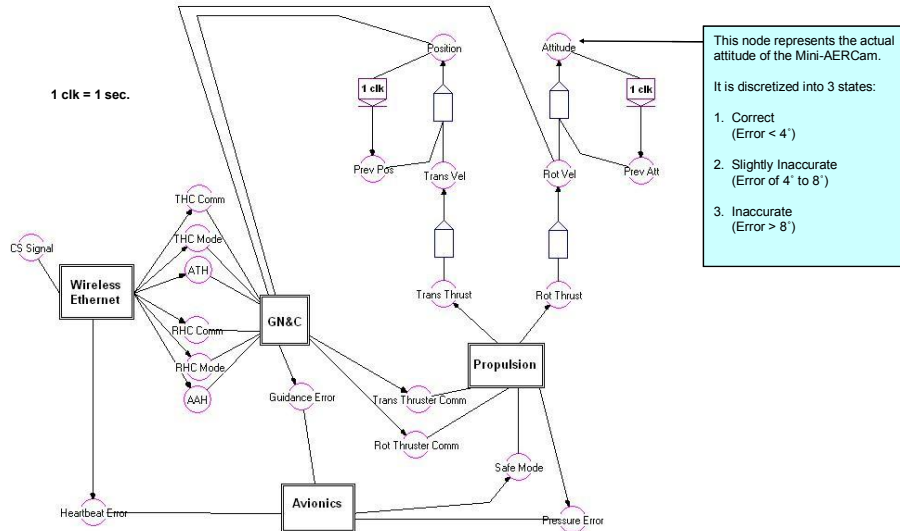


Figure 9-10. Top-Level DFM Model of the Mini AERCam System.

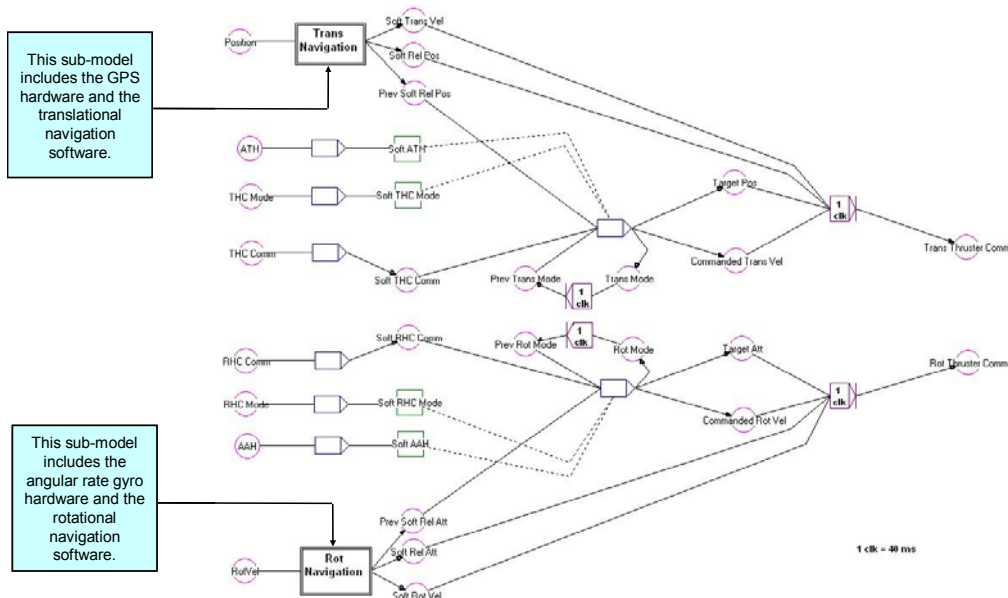


Figure 9-11. Lower-Level DFM Model of the GN&C Sub-System.

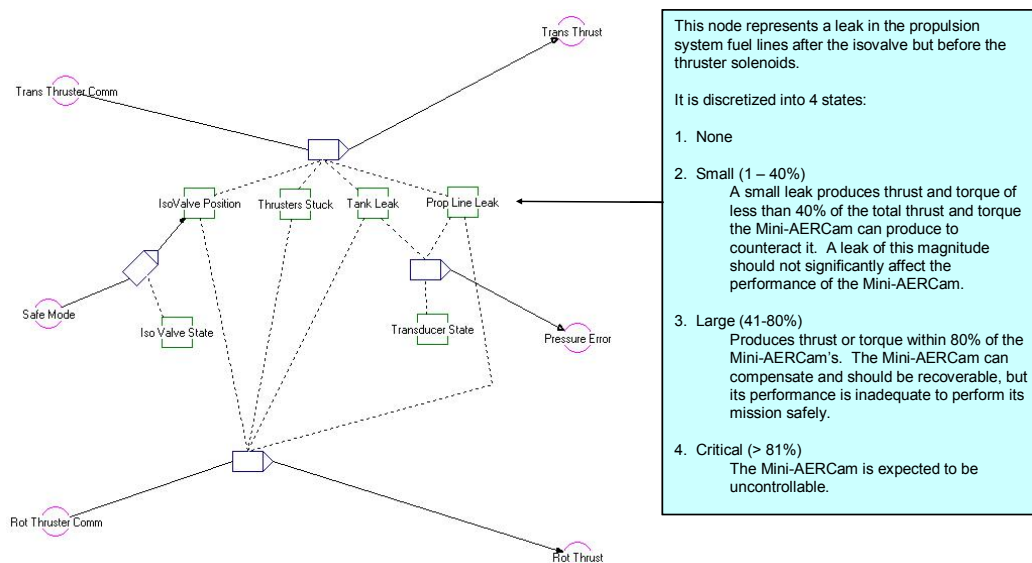


Figure 9-12. Lower-Level DFM Model of the Propulsion Sub-System.

A DFM model of the scope illustrated by the above figures can be used to analyze a wide range of possible system operational conditions, including success, anomaly, and failure events and scenarios. However, for the purpose of the present example, the discussion will focus on a specific GN&C function entry-point event of interest, defined as the “Failure to Maintain Position” event.

4. Identification of basic event cut-sets for CSRM entry-point events.

The identification of cut-sets for the CSRM entry-point events is carried out in this step in a fashion similar to the corresponding step of a Specification-Level application. For the specific “Failure to Maintain Position” event on which the focus of the example discussion is pointed, the DFM analysis process yielded, as one might expect based on the earlier discussion in Sections 9.4.2.5 and 9.4.4.1 (step 4), the same three types of cut-sets introduced there, i.e.:

- “non-software” cut-sets, e.g. :
“Isolation Valve = Stuck Closed at time -1”
- “software only” cut-sets, e.g. :
“Calculated Target Position = Inaccurate at time -1”
- “non-software-triggered, software related” cut-sets, e.g. :
*“Propellant Line Status = Small Leak at time -1 . AND .
Calculated Thruster Command = Slightly Inaccurate at time -1”*

The type B software related cut-set *“Isolation Valve = Stuck Closed at time -1”* identifies a possible “spontaneous” software failure occurring during the execution of a routine software calculation, and may be accordingly quantified. Usually more than one option is available for doing this, but in the best of circumstances for a Design-Level application, a combination of surrogate data from similar missions and system-specific test data fed into one of the better-validated “SRGMs” (Software Reliability Growth Models) would usually provide the best quantification avenue, as is further discussed in Section 9.4.6.8.

The type C cut-set “*Propellant Line Status = Small Leak at time -1 . AND . Calculated Thruster Command = Slightly Inaccurate at time -1*” identifies, on the other hand, a combination of hardware anomaly condition and software inadequate response to such a condition. The hardware condition is a small leak in one of the propellant lines. The Mini AERCam GN&C software design includes such a condition as one that the software can compensate for by making appropriate algorithmic adjustments to issue commands to the non-faulty portions of the thruster subsystem. Thus a software response failure would be caused in this situation by an algorithmic fault that causes a drift of the attitude control, given the non-nominal thrust condition caused by the propellant line leak trigger event.

Although it is directly implied by the “and” logic, it is worth underscoring that, if only one of the two fault events appearing in the definition of the above type C cut-set were to be true at any given time, the Mini AERCam station-keeping function would not fail. In particular, even if the GN&C software contingency-algorithm had a fault present in it, such a fault would remain in a dormant status indefinitely during mission executions, and would have no mission impact as long as no small propellant-line leak were to occur in any of those missions. Conversely, if a small leak of the type included in the GN&C subsystem design envelope were to occur, but no algorithmic fault were present in the GN&C contingency function, the latter would successfully compensate for the leak by a modified use of the thrusters and the station-keeping function would be successfully accomplished.

5. Probabilistic quantification of software contribution to system and mission risk.

As mentioned earlier, the full range of options that are available to the analyst for the quantification of entry-point event probability is discussed in Section 9.4.6. In an application that proceeds in stages, i.e., from an earlier “Specification-Level” to a later “Design Level,” a combination of quantification techniques would normally be applied, typically using “surrogate data” in the earlier stage, and in the latter superimposing, e.g., via Bayesian “updating” techniques, system-specific software test data to the preliminary results thus obtained.

While Section 9.4.6 presents the general features of the various quantification techniques that can be applied, individually or in combination, in a software risk process like CSR, this section addresses an aspect of the Design-Level quantification that is directly related to the nature of the specific information provided by the logic model analysis and cut-set identification steps discussed above, i.e., an example of quantification driven by “risk-informed testing” applied to “type C” cut sets. A general discussion of the concept of risk-informed testing is the main subject of Sections 9.4.6.6 and 9.5.1.

In the MiniAERCam analysis carried out per step 4, the type C cut-set “*Propellant Line Status = Small Leak at time -1 . AND . Calculated Thruster Command = Slightly Inaccurate at time -1*” was found to be one of the possible software-related causes of GN&C function failure, triggered by a small propellant line leak hardware fault condition. “Risk-informed testing” was, as a result, applied to quantify the cut-set and verify whether this theoretical risk contributor was sufficiently bounded in probabilistic risk terms or not.

The key observation here is that the conditional nature of type C cut-sets dispels the conventional wisdom perception that a prohibitive amount of software testing is necessary to bound risk contributions at some reasonable level. In fact, if for example it is ruled that an individual risk contributor like this should be demonstrated to be at a probabilistic risk level lower than a given value R_i , from Equation (9-7) it follows that this condition is satisfied if:

$$P_{SWi} = P(MC_i) \times P(F_{SW} | MC_i) < R_i \quad (9-14)$$

i.e., if:

$$P(F_{SW} | MC_i) < R_i / P(MC_i) \quad (9-15)$$

In terms of the MiniAERCam risk contribution of interest, the term $P(MC_i)$ in inequality (9-15) is the probability of a small propellant leak during a mission and $P(F_{SW} | MC_i)$ is the time-independent conditional probability of faulty software response to that hardware anomaly condition, which may be used as a risk-informed bounding value to be demonstrated for the software when it operates under such condition. The term $R_i / P(MC_i)$ represents the maximum value of $P(F_{SW} | MC_i)$ that allows the risk contribution limit R_i to be satisfied, and will in the following be referred to with the symbol RB_{SWi} , short for "risk bound for software under condition-i."

As discussed earlier in Section 9.4.2.4, the hardware condition is mission-time dependent and thus can be quantified with data from a HW failure rate database such as NPRD or equivalent data. NPRD-95 suggests a failure rate value of $6.0E-06/hr$ for a small fluid line leak. Given a single MiniAERCam mission duration of 5 hours, but also assuming the more stringent condition that the mini spacecraft is required to function without propellant leaks for a number missions M , we get for the risk-informed conditional software probability bounding value:

$$\begin{aligned} RB_{SWi} &= R_i / P(MC_i) \\ &= R_i / (M \times 5 \times 6.0E-06) \\ &= R_i / (M \times 3.0E-05) \end{aligned} \quad (9-16)$$

Thus, for example, if R_i is set at the quite conservative value of 1-in-100,000 (i.e., $1.0E-05$) for a sequence of 20 MiniAERCam missions, the conditional software probability bounding value is:

$$RB_{SWi} = 1.0E-05 / (20 \times 3.0E-05) = 0.017 \quad (9-17)$$

As a second example, assuming a much more prolonged use of the mini-spacecraft, e.g., in a one-year time-span where no astronaut inspection and maintenance would be possible, an R_i value of $1.0E-05$ would translate into the following conditional software probability bounding value:

$$RB_{SWi} = 1.0E-05 / (5 \times 365 \times 24 \times 6.0E-06) = 3.8E-05 \quad (9-18)$$

Type C cut-sets and associated bounding values for software conditional failure probability provide "risk-informed" criteria for software testing, which is illustrated here by example. It is recalled that the cut-set of interest is for the particular scenario in which the MiniAERCam fails to maintain its station-keeping position in the presence of a small propellant line leak affecting one of its thrusters.

The key consideration for testing the system in realistic conditions is that the leak:

- a) may be of varying magnitude, up to the rate specified as the maximum value that the GN&C software logic and algorithms must be able to compensate for, and
- b) may occur while the mini-spacecraft is in any of its possible station-keeping positions, i.e., it may be in any rotational orientation with respect to a reference coordinate system.

Therefore, the testing should "sample" the combination of leak-rate and spacecraft initial rotational-orientation dimensions in an orderly fashion such as to give reasonable assurance of coverage of the variability of conditions within the given scenario. Ideally, one would want to conduct space-system software tests in a TAYF ("test-as-you-fly") hardware-in-the-loop configuration, but this was beyond the budget and scope of the MiniAERCam PRA

demonstration project. A second-best option was followed instead, by testing the attitude control function under the simulated presence of the entry condition using a Virtual System Integration Laboratory (VSIL) simulation of the hardware (made available to the project by the Triakis Corporation). The VSIL simulation used hardware documentation to produce a realistic software model of the hardware that was interfaced with the actual GN&C software. The use of the VSIL tool allows the software test engineers to easily generate any failure modes of interest in the simulated hardware and observe how the actual software responds under these conditions. The simulation maintains a record of the state of the hardware and the software at any point during the simulation, so the GN&C software variables can be observed and compared to hardware parameters and algorithmically correct representations thereof, to determine the correctness of the GN&C software response.

Repetition of the test, with the spacecraft at different initial rotational orientations, and with the simulated leak at different flow rates, producing different force and torque exerted on the spacecraft itself, provides the statistical basis for an estimate of the probability $P(F_{SW} | MC_i)$ that is of interest for the risk scenario investigated in this example.

A random sampling of the above “test space” with a total of 350 simulated tests resulted in no GN&C software failure to control the spacecraft. This result can be used in a straightforward Bayesian estimation to obtain $P(F_{SW} | MC_i)$. For example, an estimation starting from an uninformative Jeffreys prior (i.e., a beta distribution with parameters $\alpha=0.5$ and $\beta=0.5$) gives the following estimates for $P(F_{SW} | MC_i)$:

5 th percentile	5.61E-06
Median	6.49E-04
Mean	1.42E-03
75 th percentile	1.89E-03
95 th percentile	5.47E-03

The above table thus shows that, with the stated test results, the 1.9E-03 RB_{SW_i} risk-bound established for $P(F_{SW} | MC_i)$, in order to limit the unconditional risk cut-set contribution to less than 1.0E-05, is satisfied not only at mean and median levels of confidence, but up to about a 75% level. Working the Bayesian estimate math in reverse, one can also easily calculate the number of input space sampling tests needed to satisfy a given RB_{SW_i} limit for $P(F_{SW} | MC_i)$ at a desired level of confidence. For example, assuming the same required RB_{SW_i} value of 1.9E-03, a target 95% level of confidence requires 1010 sampling tests if no software response failures occur. That number more than doubles to 2050 if one failure does occur.

The last very important observation to make before concluding this example is that the above considerations on probability and associated confidence level estimation are valid only if the tests are well set up to sample the key dimensions of variability associated with the “trigger event” conditions. Optimistic estimations and false confidence would in fact be generated if any number of successful tests were in practice just the repetition of a previously executed successful test. The means for avoiding falling into such logic trap are not purely statistical, but require good engineering judgment of the factors that may truly introduce variability into the “system-trigger, software-response” process.

9.4.5 CSRM Modeling Detail and Representation of Software Failure Modes

The level of detail to be sought in the development of a PRA logic model is in general a simple question, to which unfortunately there exists no simple answer. This is because there is more than one factor to be considered, depending on the ultimate purpose of the PRA itself.

If the sole purpose of a PRA were to provide a risk metric at the system and subsystem level, then the answer to the modeling detail question would be driven by the availability of quantification data. That is, there would be no point in developing the PRA logic models beyond the level of detail at which the corresponding “basic events” can be directly quantified with the available data. However, such is in general not the case when the PRA is used, as it should be, as a system design and validation aid and a means to anticipate and understand system risk scenarios that could otherwise be overlooked. Under these conditions, a model can be developed in greater detail than suggested by the availability of direct quantification data, and if data are desired at a lower level of detail, they may be derived by sub-allocation techniques, often combined with and facilitated by the use of expert elicitation.

In the case of software, the lack of data argument has been a sort of “catch-22,” i.e., the lack of data to quantify risk has been one of the primary arguments used to explain the high-level nature of software PRA model developments, if any were even attempted; conversely, given the place-holder format of such models, no need has been perceived to exist for better organized test and operational data collection, and therefore software data collection efforts have for the most part also been kept at the high-level, without developing classifications of failures or anomalies by function, or by lower-level categorizations when a software unit is programmed to carry out more than one function and contains identifiable sub-functions.

In general, software data collection has not progressed to the level of organized categorization that is currently standard for hardware data. That is, at the present time software operational data are generally not systematically recorded according to a classification of basic functional characteristics, by which the likelihood of a software fault can be correlated with the type of basic functions that are been executed within a given software module or unit, such as, for example, logic operations, algorithmic implementations of a mathematical calculation, control of data flow, etc.

At the current state of the art, very little exists in the way of a software failure mode classification of the type just mentioned. From a CSRM modeling point of view, however; some steps may be taken that go in the right direction, while at the same time being practically executable without an excessive level of effort .

The basic purpose of a CSRM model development is to permit the integration of both qualitative and quantitative software risk information into the standard framework of a system PRA, as typically carried out in NASA programs and projects. In practical terms the software risk contribution is of interest because the consequences are felt at the material level, i.e. in what happens to the hardware and human components of a mission. Thus, while the “mode” of a fault inside the software may be of analytical interest, the mode of manifestation of the fault at the interface between the software and the “balance of system” is what ultimately determines the eventual effect of a software fault on the system as a whole. This suggests that a recommended minimum level of CSRM modeling detail, and a corresponding definition and identification of “software failure modes,” should be at least be at the level of the software / balance-of-system interface(s). Thus, for CSRM purposes, it is appropriate to focus the modeling and analytical attention on the “software interface failure modes,” and from the identification of these, possibly proceed, if this is judged useful, to identify potential failure-modes that may occur within the software.

An example of the above is provided with the aid of the DFM model in Figure 9-13. The model represents a software control-module that receives as input the value of a tank pressure (TP node), and provides as outputs commands to a valve actuator (VAA node) and to a pump motor (PM node). The non-software portion of the model is shown in blue, whereas the software and interface portions of the model are shown in green. The actual pressure reading goes through a sensor-to-software interface (e.g., an analog-to-digital converter) represented by the box labeled T7 to become an internal software variable (TPSP node). The TPSP value is then elaborated by the software logic and algorithm (box TT2) to determine a pressure “control action” (SPCA node). This in turn determines the specific software commands (VCSP and PCSP nodes) which are addressed to the valve actuator (VAA node) and to the gas pump motor (PM node) via software-to-hardware interfaces represented by the boxes TT3 and TT4.

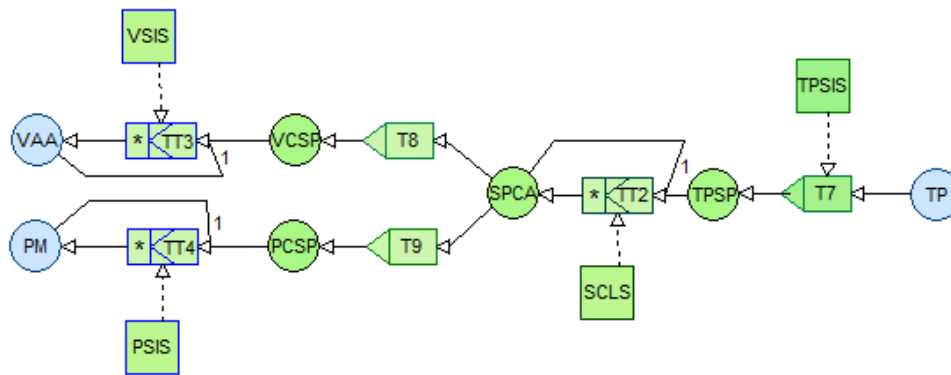


Figure 9-13. DFM Model for Illustration of SW Failure-Mode Representations.

In the above model, relevant software failure modes are identified:

- first by tracking the interfaces between the balance-of-system and the software and accordingly represented in the interface-status nodes TPSIS (on the sensor side at the interface between the “true pressure” TP and its “software image” TPSP), VSIS (on the actuation side between the software valve command VCSP and the valve actuator VAA) and PSIS (on the actuation side between the software pump-motor command PCSP and the pump-motor PM);
- then by proceeding one layer-further, into the basic algorithmic function and logic of the software control, which governs the interaction between the internal software-image TPSP of the controlled pressure variable and the control action selected SPCA by the software itself; the software function failure modes are accordingly represented by the software-status node SCLS.

In the particular example represented in Fig. 9-13, the following software-interface and software-function failure modes were identified following the process illustrated above:

Sensor / software interface failure modes:

- a. reading stuck at high end of pressure scale
- b. reading stuck at low end of pressure scale
- c. reading drifted higher than true pressure value

- d. reading drifted lower than true pressure value

Software / valve-actuator interface failure modes:

- a. command stuck at “close-valve” value
- b. command stuck at “open-valve” value
- c. command frozen at last active value

Software / pump-motor-actuator interface failure modes:

- a. command stuck at “start-motor” value
- b. command stuck at “stop-motor” value
- c. command frozen at last active value

Software internal function failure modes:

- a. function execution frozen at last active execution
- b. function execution logic “reversed” (i.e., response to low pressure selected when pressure is high and vice versa)
- c. spurious activation of low pressure response
- d. spurious activation of high pressure response.

The above definition of software-related failure modes is of course presented as an illustration of the reasoning process that can be applied, and should not be interpreted as a “recipe” to be universally applied in any modeling context. As underscored earlier, the method and format of definition of failure modes of a given system is a decision that the PRA analyst has to make case by case, as it is intimately intertwined with decisions concerning the desirable level of modeling detail that is sought. This is always true, regardless of whether the modeling of concern regards hardware, software, or human aspects of a system.

9.4.6 Software Risk Quantification

The topic of software risk quantification is undoubtedly a complex one. In earlier sections it has been addressed from the point of view of the nature of the data that may be available at certain stages of application of the CSRM process, and of how a certain type of data, e.g., data that may provide an estimate of software failure rate in time, vs. data that fit a conditional “failure on demand” model, may fit the two basic types of CSRM model cut-sets. The discussion in this section, although certainly not exhaustive, provides a broad review of the main types of software reliability data that are typically available for use in a space system risk application, as well as of the types of estimation models that may be associated with these data.

In general terms, before entering any more detailed discussions, it is appropriate to make the reader aware of a general classification of software quantification models that may be encountered in the literature, i.e., the distinction between “black box” and “white box” models:

- A *black box* model of a given software unit or module is a model that assumes no structural or functional information about the software itself.
- A *white box* model of a given software unit or module is a model that relies on, and reflects, structural or functional information about the software itself.

With respect to the above, it is noted that, strictly speaking, in the mathematical world a white box model is defined as a model that reflects complete information about the object being modeled. In reality, in real-world circumstances the information at hand is never “complete,” thus even in the best of cases the modeling effort results, in practice, in a “gray model.” For this

reason, in referring to these subjects, the term “functional model” is used in place of “white box model,” to make a distinction from “black box model” types of modeling approaches.

It is also noted that the distinction between black box and functional modeling approaches is entirely relative to the level of indenture/detail at which a modeling effort is carried out. That is, a functional modeling approach at the system level may become a black box approach at the subsystem or lower level of modeling. In this respect, from the discussion throughout this chapter, it should be clear to the reader that CSRM calls for a functional modeling approach, as a minimum, at the system level and subsystem level.

The discussion that follows addresses the basic types of models that may be used and includes observations and suggestions on how the models and the data on which they rely may be utilized in the CSRM context.

9.4.6.1 Basic Types of Software Reliability Data

The software reliability and failure data that are applicable, or proposed as being applicable, for use in space system software risk quantification generally can be classified as corresponding to one of the following three categories:

- A. Surrogate Raw (SR) Data, i.e., data collected from systems and mission operations judged to be sufficiently similar to the ones being assessed to permit probabilistic estimates extracted from the SR data to be used for the systems and missions of present concern;
- B. Surrogate Parametric (SP) Data, i.e., surrogate data from other systems and operations that are no longer accessible in their original raw state, but have been processed and collapsed into some type of parametric correlation model;
- C. System-specific Debugging (SD) Data, i.e., data collected during a process of software test and fault-correction for the same system and software modules that are of interest;
- D. System-specific Operational (SO) Data, i.e., data collected during software test or operation, but in any case after the end of any planned and systematic fault-correction process, for the same system and software modules that are of interest;

Each of the above four basic types of data may be used in the estimation of reliability and risk parameters of interest during the execution of a software PRA process like CSRM. A common characteristic of these four types of data is that they usually exist externally and independently of the PRA and/or CSRM activities that may be underway.

An additional type of data which, on the contrary, is produced in concurrence with a PRA and CSRM application is:

- E. Risk-informed Test (RT) Data, i.e., data generated via tests specifically formulated and prioritized in accordance with risk information produced in the course of a software PRA and CSRM application.

A survey of the estimation techniques used or proposed in various contexts and applicable in association with each of the above categories of data follows below in Sections 9.4.6.2 through 9.4.6.6. These sections and the following Sections 9.4.6.7 and 9.4.6.8 discuss the relevant aspects of the possible use of the surveyed techniques in the context of a CSRM application.

9.4.6.2 Parameter Estimation with SR Data

Surrogate Raw (RS) Data can be used according to their nature and the nature of the risk parameter that is to be estimated.

In the CSRM context, we have discussed in Sections 9.4.4.1 and 9.4.4.2 how software cut-sets that represent the failure of a routine software function at some point during a time-cyclical execution may be quantified via a pseudo-random failure rate model, whereas software cut-sets that represent the conditional failure of a specific software contingency function upon the occurrence of a system trigger-condition should be quantified using a probability-of-failure-on-demand estimation. The techniques to be applied in the estimation are not different from those that are applied in a corresponding hardware reliability or risk parameter estimation.

An estimation of CSRM probabilities based on SR data is usually satisfactory at the stage of a "Specification-Level" application, when no system-specific software test or operational data are usually available. Of course the predictive value of SR data is dependent on the degree of similarity between the software and system of present concern and the systems and software from which the data were collected. Reference [9-4] discusses two SR databases that have been assembled and used in recent CSRM applications executed within the now discontinued Constellation Program.

9.4.6.3 Parameter Estimation with SP Data and Parametric Models

As implied in the short description given above, Parametric Surrogate (SP) Data are not actually data available to the PRA analyst. The actual data are substituted by a parametric correlation model. This type of model is not strictly speaking a "black box" model, in that it attempts to incorporate certain information about the software system being modeled, but it is not a "functional model" either, since it does not attempt to identify, nor rely on, structural differentiation of functionality within the system being assessed. In this respect, parametric models are closer to a "black box" than to a "functional" approach to parameter quantification.

To use such a model, the analyst has to identify certain characteristics or "factors" pertaining to the software for which he/she is interested to carry out the reliability/risk estimation, typically by answering a questionnaire built into the specific parametric model package being used. These are factors which have been identified in the model built into the parametric tool of choice as factors whose strength or weakness drives software reliability according to a multivariate correlation formula. The correlation model usually provides a "software defect density" prediction, which can be translated into a failure rate or failure probability estimate by the analyst, on the basis of additional information about the specific nature of the software and system of interest. More details on two parametric models of the kind described here can be found in [9-8] and [9-9].

In theory, SP data and the associated parametric model may be used in lieu of SR data, saving the analyst the time and effort that is usually required to organize and categorize SR data into formats amenable to an estimation of the parameters of interest. Unfortunately, however, the original data used to derive the parametric correlation models that generate the desired estimations are generally not available to the PRA analyst. Unlike in the case of SR data usage, the analyst has therefore no insight to judge whether such data have reasonable applicability for the type of estimation of current concern, nor to judge whether the correlation formula contained in the model is being extrapolated beyond its limits of validity in the current application.

A comparison of quantifications for the software risk contribution in the Constellation Program PA-1 mission, using SR versus SP data, found the latter to produce predictions more

“optimistic” by a factor of 100 to 1000 [9-10], whereas the former was in apparent alignment with the historically observed rate of software-related mission failures in the decade up to the year 2007.

9.4.6.4 Parameter Estimation with SD Data and SW Reliability Growth Models (SRGMs)

System-specific Debugging (SD) data are normally associated with a process of repeated software test, fault discovery, and fault removal. This process and the associated SD data that it generates provide the basis for reliability parameter estimation via the use of a Software Reliability Growth Model (SRGM).

SRGMs are “black box” models that provide an idealized description of the way in which the reliability of a software system grows as defects are identified and removed during test. Taking as input a software system’s failure history observed during test, they return the parameters of a probability density function (pdf) for the time to the next failure (or the number of failures in the next test interval), which can be used to estimate and forecast reliability and reliability-related quantities (e.g., failure rate, number of failures to be observed in a future time interval t , etc.).

All SRGMs make the following assumptions about the defect discovery and removal process during testing [9-11, 9-12]:

- A. The software is operated in a similar manner as that in which reliability predictions are to be made.
- B. Every fault within a severity class has the same chance of being encountered as any other fault in that class.
- C. The failures, when faults are detected, are independent.

Assumption A is made to ensure that model results are produced from failure history data that are applicable to the environment in which the reliability estimates and forecasts are to be made.

Assumption B requires the failures or anomalies within a “severity class” to have the same distributional properties. However, each severity class may have a different failure rate than the others, requiring a separate analysis be conducted. This particular assumption has a relevant CSRM-specific implication which is discussed below.

Assumption C makes the models mathematically tractable – by assuming independence of the failures, the joint pdf from which the model parameters are estimated can often be solved analytically, thereby dispensing with computationally-intensive numerical techniques.

It is important to note that although the above assumptions are usually not a strictly accurate representation of the testing process, they are frequently satisfied in sufficient measure to allow satisfactory estimates of software reliability to be produced.

In addition to assumptions A through C, all SRGMs make the more basic implicit assumption that the reliability of the software is increasing during test through the process of observing failures, identifying defects responsible for the failures, and removing these defects. If a software system is not exhibiting reliability growth during test, SRGMs may still yield reliability estimates, but those estimates are likely to be largely inaccurate.

With respect to their use in a CSRM context, two observations can be made with respect to SRGMs. The first observation is that the SD data on which they rely usually become available only in the “Design-Level” CSRM application stage. The second is that the assumptions on which SRGMs rely, and especially Assumption B, make them primarily suitable for the estimation of “routine” software function cut-sets, i.e., those labeled as “type B” in Section

9.4.2.5. This is because SRGMs can make distinctions between failures of different “severity,” as assigned by the personnel running the software tests and detecting defects, but the test themselves are not usually set up to sort “conditional-defect” by the trigger-events that may be causing them.

In summary, the considerations discussed above suggest that SRGMs may be primarily utilized in a CSRM “Design-Level” application context for the assessment of “type B” software routine-function cut-sets.

More detailed information and specific mathematical formulations of SRGMs are easily found in the open literature. In the NASA environment, an SRGM data gathering and parameter estimation environment that incorporates more than one SRGM model, from which an analyst can choose a preferred type of estimation, is available in JPL’s CASRE tool [9-13].

9.4.6.5 Parameter Estimation with SO Data

System-specific Operational (SO) Data usually become available after “debugging test” cycles have been completed, thus estimation of reliability parameters via the use of SO data is generally only possible at the Design-Level CSRM stage.

In practical terms, these data consist of the recorded observations of anomalies or failures, or absence thereof, during the operation of the deployed system of interest, i.e., in their basic nature these are the same type of data that are usually represented in a SR (surrogate raw) database. The estimation techniques that can be applied are therefore generally the same standard estimation techniques that may be applied in the utilization of SR data.

It is noted that, if the operational data collection is preceded by the collection and utilization of SD (system-specific debugging) data via the application of an SRGM, then the SO-based estimation may be combined with the SD-based estimation, for example via a Bayesian updating technique (see also Section 9.4.6.7 on this subject).

9.4.6.6 Parameter Estimation with RT Data

Risk-informed Test (RT) data are data collected as a result of a software and system test effort organized and prioritized on the basis of an initial software risk assessment process. In general this may be driven by the result of an initial Specification-Level CSRM effort, and the results of the RT data estimation processes may become a key part of the basis for a final Design-Level analysis and quantification.

RT data may take a form equivalent to SD or SO data, depending on the nature of the risk-informed test procedure applied, i.e., whether it involves a process of defect removal and software retest, which in turn clearly may depend on the initial level of defects detected at the start of the risk-informed test process.

Additional considerations on the concept of “risk-informed testing” can be found in Section 9.5.1.

9.4.6.7 Parameter Uncertainty Estimation

Standard PRA practice recommends the assessment of uncertainty in risk model parameters and its ultimate effect on top-level risk metrics. To follow this practice, any estimation of parameters from software reliability data should also incorporate an uncertainty analysis. This presents no special problems when a parameter estimation is carried out by the means typically employed in a conventional-PRA context, e.g., a Bayesian estimation that provides the estimated parameter statistics in the form of a full distribution function.

The assessment of uncertainty is a little more difficult when an estimation is carried out by means of an SRGM, since such models usually don't have built-in mechanisms to provide estimated values at varying confidence levels in a statistical sense. However, an analysis of the underlying SD data from an uncertainty estimation point of view can still be carried out by a PRA analyst with sufficient experience and expertise.

The most problematic case for proper inclusion of uncertainty in software reliability parameter estimation would be if a parametric model with pre-collapsed SP data is used. These models do not provide an uncertainty estimate with the output they provide. In addition, in the third-party models that are currently commercially available, the PRA analyst has no access to the underlying proprietary data in order to execute an uncertainty estimation based on the actual data content that is directly or indirectly reflected in the parameter estimation.

9.4.6.8 Use of Multiple Data Sources in Parameter Estimation

Generally speaking, the golden rule of data utilization in PRA is: "use all that are available and applicable, if they carry real information content." This remains true in the use of software reliability data, thus it would generally be expected that a parameter estimation will be based on a combination of data sources.

The data will usually become available in a sequential and progressive augmentation, paralleling the progression of a project and associated system design, which in turn will be reflected in the Specification-Level or Design-Level stage of development of the CSR analysis. In a typical sequence of this kind, one can, for example, envision an estimation that is initiated with SR data at a CSR Specification-level of application, and then is continued at the Design-Level first with inclusion of SD data fed into an SRGM, then with inclusion of RT scenario-related data, and possibly in the end with inclusion of SO mission data.

There are no mandatory prescriptions for how multiple data sources of varying nature can be selected and combined in a probabilistic parameter estimation, but the usual, and most naturally convenient technical route for such an aggregation is the use of a Bayesian assessment framework, which permits successive "updates" of the estimation of a given parameter. In a Bayesian framework the updating process can incorporate new batches of data as these data become available, also automatically producing a new uncertainty assessment as an output of each "update." The degree of estimation uncertainty, as for example indicated by the range of parameter values comprised between the 5th and 95th percentile of the updated parameter distribution function, will normally decrease as new data are added into the estimation process.

9.4.6.9 Quantification of Specific Software Failure Modes

The last sub-topic of the software risk quantification subject that requires some discussion in this chapter is the quantification of lower-level, specific software failure modes, such as those provided as examples in Section 9.4.5.

The pre-existing surrogate data which are available for quantification of software reliability or probability of failure, are usually categorized at a higher level of detail than that reflected in the definitions of the example failure modes of Section 9.4.5. For example, the SR database utilized in the Constellation Program assessments described in [9-4] and [9-10] permitted the direct estimation of more generic and higher level failure modes, such as:

- Failure of continuous control function
- Failure to issue discrete command
- Inadvertent issue of discrete command.

On the other hand, the system-specific test processes that may be applied at a later stage of CSRM application can be more specifically directed at an assessment of lower-level failure modes that are of concern for one reason or another. In such cases, to permit the utilization of both types of data at the lower level which is eventually of interest, the initial estimation with SR data can be brought down to the desired lower level by a process of sub-allocation.

Sub-allocation refers to the process of apportioning the value of the estimated probability of a certain type of failure among the different “sub-modes” identified for such a failure, according to pre-defined ratios that are either assumed, analytically derived via a specific model, or obtained via expert elicitation.

For example, in the case of the software control function represented in Section 9.4.5, Fig. 9-13, the PRA analysts, in preparation for an eventual updating with specific test data, may want to sub-allocate the corresponding probability of failure. This may have initially been derived from an SR database like the one mentioned above, i.e. using data corresponding to a software failure category described as “Failure of continuous control function.” The sub-allocation will thus consist of deciding how the corresponding estimated probability may be apportioned among the five failure-modes identified in the Fig. 9-13 example as sub-cases of the continuous control function failure category, i.e.:

- a. function execution frozen at last active execution
- b. function execution logic “reversed” (i.e., response to low pressure selected when pressure is high and vice versa)
- c. spurious activation of low pressure response
- d. spurious activation of high pressure response.

Although it may appear at first that an objective sub-allocation rationale or basis may not be easy to identify, once specific situations are examined, it is usually possible to find sufficient justification to execute it, given also that the usual objective is that of obtaining a “good enough” starting point for more directed testing and detailed estimation.

9.5 Use of Software Risk Information

A last subject that requires discussion within the topic of “software risk assessment” is the use of the software risk information that is produced, both in qualitative and quantitative terms, and from both a software-specific and general system perspective.

From the start of this specific discussion subject, it is important to note the influence that software testing can have on the actual residual level of operational risk. This is a characteristic of software that is relatively unique and without a close equivalent in the hardware world. The special significance of this characteristic is that it makes for an even stronger argument than in the utilization of a conventional hardware-oriented PRA, for a proactive use of the PRA results, much beyond the objective of providing a “risk number.”

9.5.1 Conditional Scenarios and Risk-informed Software Testing Strategies

From a software design and validation point of view, the most important insight possibly obtainable from a CSRM assessment is the validation of software logic for contingency operations, i.e., the systematic identification and analysis of risk scenarios in which the software is by design supposed to respond with a contingency, or “fault-management,” action to a system hardware anomaly, or to some specific and potentially critical environmental condition encountered during mission execution. As has been illustrated with the discussion of the examples provided in Section 9.4, this type of insight can be achieved in comprehensive fashion

via the combined use of traditional and more advanced models, depending on the nature of the subsystems and functions being addressed in the analysis.

The analysis results can preliminarily be used to focus analytical attention on areas that are identified as potentially critical on the basis of both qualitative and quantitative risk information, i.e., either because the software functions involved are recognized as being characterized by high complexity, or because initial quantification with “Specification-Level” surrogate data indicates that similar functions have experienced anomalies or failures with relatively higher probability in preceding missions.

A risk-informed prioritization or focus on specific software-related contingency scenarios can then lead to the process of “risk-informed testing,” i.e. to a strategy of software tests that are directed at the higher risk “partitions” of the overall theoretical software test space. This can be done with the specific purpose of sampling the specific test partition of concern in such a way as to be able to achieve reasonable confidence of maintaining the probability of software failure for the related scenario(s) below a given risk limit value. This type of risk-informed approach to software testing has been illustrated with the MiniAERCam GN&C test example given in Section 9.4.4.2. As discussed there, this type of risk-informed software testing ideally calls for a “hardware-in the-loop” approach, which can be realized via simulation if a true hardware/software test set-up is impractical and/or too costly.

9.5.2 Integration of Results into Pre-existing PRA Models

The most obvious and standard utilization of a CSRM analysis from an overall system assessment point of view is the completion of the system PRA model with models and quantifications of breadth and depth in the software “dimension” comparable to the breadth and depth of the PRA models developed for the hardware portions of the system.

As discussed in Section 9.4, the CSRM models employed to represent and analyze the system software functions are generally amenable to integration with a conventional PRA framework of event-tree and fault-tree models, and the quantifications thereof can be applied to a corresponding “entry-point event” in one or the other type of conventional PRA binary models.

In addition, even the more sophisticated forms of CSRM logic models, i.e., the DFM models, produce results that can be easily translated into equivalent binary cut-set format. Thus, if it is desired to insert the CSRM information into an existing PRA set of models utilizing a conventional binary logic format, this can be easily done by simply appending to the CSRM entry point event of interest a fault tree structure that represents the cut-sets obtained via the CSRM analysis. Figure 9-14 graphically shows the standard binary cut-set representation of a higher-order failure event as the logic “union” of the underlying cut-sets, which in turn are either basic-events, or “intersections” thereof. Based on this type of equivalence, all quantitative information, including not only cut-set probability, but also the attendant uncertainty information, can be transferred from the CSRM models into the conventional PRA structures. Pilot automated or semi-automated integrations of DFM models and analyses, constructed and carried out via the DFM tool DYMONDA™ [9-7], with widely used PRA software tools (e.g., CAFTA™ [9-14] and SAPHIRE™ [9-15] have been programmed and demonstrated.

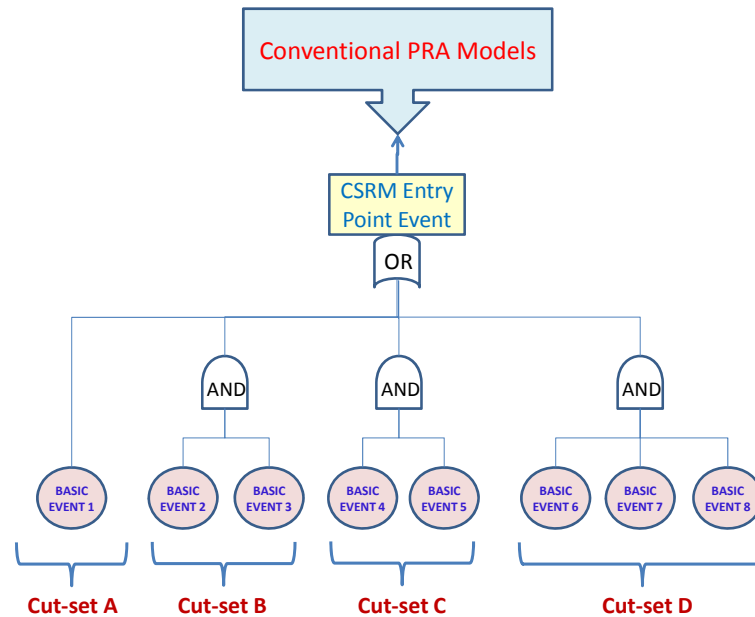


Figure 9-14. Notional Example of Expansion of Entry-point Event into CSRM Cut-Sets.

9.6 Definitions

The following definitions are provided to facilitate the interpretation of terms and language used in Chapter 9. The definitions are listed in a top-down logical/hierarchical, rather than alphabetical order:

Software Reliability:

The probability of failure-free software operation for a specified period of time in a specified environment. [9-16]

Software Defect:

A specific feature or characteristic of a software component, by which such component may deviate or differ from its design specifications.

Software Fault:

A software defect serious enough that, if executed during actual system operation, it will result in a system functional anomaly or failure.

Software Error:

The actual operational execution of a software defect.

Software Anomaly:

The actual operational execution of a software fault that causes a system functional anomaly.

Software Failure:

The actual operational execution of a software fault that causes a system functional failure.

Condition Coverage:

The ability of a modeling or test process to address the key mission operational scenarios, including risk scenarios that are, or should be included in the design basis of a mission.

Fault Tolerance:

The capability built into a system to identify and isolate faults, preventing them from producing system anomalies or failures.

Software Testing:

An organized and formal process of pre-operational software execution carried out for the purpose of identifying and correcting software defects and faults.

Software Operational Fault Coverage:

The degree by which a given software architecture and design is capable of identifying, isolating, and recovering from, functional failures occurring in the software system and supporting computer and network hardware during the operational phases of a mission.

Software Test Fault Coverage:

The degree by which a given type of test process is capable of exercising software functionality and identifying any associated potential faults.

Software Test Fault Coverage Percent Metric:

A quantitative expression in percentage terms of the fraction of software faults that a given test process or procedure is capable of identifying, usually estimated by intentionally injecting faults in some fashion across the functional range of a given software component ^a.

Software Operational Fault Coverage Percent Metric:

A quantitative expression, in probability or “measured” percentage terms, of the fraction of functional failures, among those which may occur in the software system and supporting computer and network hardware during the operational phases of a mission, that a given software architecture and design is capable of identifying, isolating, and recovering from.

9.7 References

- 9-1 J.C. Munson, A.P. Nikora A., J.S. Sherif, “Toward A Quantifiable Definition of Software Faults,” *Advances in Engineering Software*, Volume 37, Issue 5, May 2006, Pages 327-333.
- 9-2 Mars Global Surveyor Operation Review Board, “Mars Global Surveyor (MGS) Spacecraft Loss of Contact,” Preliminary Report, 13 April 2007.
- 9-3 M.C. Hsueh and R. Lyer, "Performability modeling based on real data: A case study", *IEEE Transactions on Computers*, vol. 37 no. 4, April 1988, pp. 478-484.
- 9-4 “Instruction Guide for Integration of the Context-based Software Risk Model (CSRM) with a Cx Probabilistic Risk Assessment (PRA),” ASCA Report for NASA Johnson Space Center, AR 09-03, September 15, 2009.
- 9-5 T. Aldemir, S. Guarro, et al., “A Benchmark Implementation of Two Dynamic Methodologies for the Reliability Modeling of Digital Instrumentation and Control Systems,” U.S. Nuclear Regulatory Commission Report NUREG/CR-6985, Washington, DC, February 2009.

^a **Note:** A quantitative measure of test fault coverage may depend to a non-negligible extent on the very way by which faults are injected and also on the nature of the faults that are injected. For example if a certain type of software functionality is not explicitly recognized and no associated fault is injected, the “measured” fault coverage will not reflect the perceptivity of the test process with regard to that specific type of software functionality. Also, a certain type of injected fault may be more easily detectable by a certain type of test process than another type of fault, although both may be equally severe in mission impact terms.

- 9-6 M. Yau, M. Motamed and S. Guarro, "Assessment and Integration of Software Risk within PRA," International Journal of Performability Engineering, Vol. 3, No. 3, July 2007.
- 9-7 www.ascainc.com/dymonda/dymonda.html
- 9-8 J. McCall and J. Cavano, "Methodology for Software Reliability Prediction and Assessment," Rome Air Development Center (RADC) Technical Report RADC-TR-87-171, volumes 1 and 2, 1987.
- 9-9 <http://softrel.com/>
- 9-10 "Application of Context-based Software Risk Model (CSRM) to Project Orion Pad Abort-1 Flight Software," ASCA, Inc. report for NASA Johnson Space Center, AR 09-01, April 30, 2009.
- 9-11 P.A. Keiller and D.R. Miller, "On the Use and the Performance of Software Reliability Growth Models", Software Reliability and Safety, Elsevier, 1991, (pp. 95-117).
- 9-12 M.R. Lyu , "Handbook of Software Reliability Engineering," McGraw-Hill Publishing, 1995, ISBN 0-07-039400-8.
- 9-13 http://www.openchannelsoftware.com/projects/CASRE_3.0
- 9-14 <http://teams.epri.com/RR>
- 9-15 <https://sapphire.inl.gov>
- 9-16 ANSI/IEEE, "Standard Glossary of Software Engineering Terminology", STD-729-1991, ANSI/IEEE, 1991.

10. Physical and Phenomenological Models

Models that describe physical events and phenomenology of interest are widely used in risk modeling, to complement and help quantify the logic models that constitute the backbone of a typical PRA.

Phenomenological modeling is a physics-based analysis method used to study and characterize complex, interactive systems where the progression of events is governed by physical processes. Phenomenological modeling techniques are used to complement and extend traditional reliability modeling methods by evaluating the nodes in a logic tree that are driven by physical processes. Fault trees and event trees found in probabilistic risk assessments (PRAs) include failure initiators that are statistically generated based on reliability methods. However, the nodes that represent the subsequent events following such failure initiators often require phenomenological models in order to compute the probabilities. This is especially true when the sequence of events and processes depends upon the physical interactions of the system with its current surroundings.

Engineering models based on the fundamental laws of motion, heat transfer, chemical reactions, gas dynamics, structural analysis, and other phenomena can be used to represent faithfully the conditions and state of the environment surrounding the system. These physical processes evolve as a function of time and system state, and are defined using mathematical equations that describe fluid flow, wave propagation, structural fatigue, combustion, crystallization, and so forth. A failure is defined to occur when the system observes physical conditions that violate a subsystem's specific physical limit or capacity.

In phenomenological risk models, the interactions of a complex system are coupled through common physical parameters, and the subsequent responses and consequences are dynamically determined based on the current conditions of the system, its environment, and its design limits. Failure probabilities are then developed by calculating the range of current state conditions, and determining whether they violate a specific design limit or system threshold. Failure probabilities developed by this approach directly and explicitly connect the existing design definition to the physical failure mechanism, providing design teams with actionable engineering information.

At the backend of a PRA, consequences of the system failure events need to be estimated and reported on some form of continuous or discrete scale with a large number of states. Thus non-binary physical and phenomenological models, such as "health effects models" and "casualty expectation models" are also applicable and commonly found in this portion of a PRA.

In some cases, a specific type of scenario that lends itself to modeling via physical and phenomenological models may be addressed as a special subject within a PRA, or as a complementary study. This can be the case, for example, for the analysis of the threat to space missions posed by "micro-meteoroid and orbital debris" (MMOD).

The sections of this chapter discuss the types of physical and phenomenological models more commonly encountered in PRAs. Examples are provided in these sections to show how these models are formulated and applied within the PRA process. Topics covered include:

- Phenomenological modeling during the design process
- Stress-strength formulation of physical models
- Range safety models

- MMOD risk modeling
- Ground-based fire PRA

10.1 Role of Phenomenological Methods in Risk Assessment

This section presents some of the fundamental characteristics and attributes of phenomenological risk models, and how their application differs from and augments traditional reliability analyses. Phenomenological modeling is a powerful tool for improving the understanding of a system under a wide range of nominal and off-nominal conditions. It is used to quantify dynamic failure modes that are driven by physical conditions, interactions, and design limits. While statistical analyses focus on reliability-based failures driven by random faults, anomalies, and part failures, phenomenological models instead focus on assessing failure mechanisms that are due to design and engineering vulnerabilities or unintended physical interactions in an integrated system.

Phenomenological models focus on characterizing unintended physical interactions among systems and/or between a system and its environment. Developing a strong understanding of system-environment interactions is extremely important because the environment is not a directly controllable system and may deviate unexpectedly. Therefore, system capabilities must be explored within a range of conditions, not just nominal conditions, in order to better understand potential vulnerabilities and the levels of safety margins required. Because phenomenological modeling is based on specific conditions and physical interactions, it can be used to characterize off-nominal system behavior and unsteady physical processes in addition to nominal, steady behavior.

When steady-state processes are being studied, it may be appropriate to use statistical methods. Since the environment and system interactions studied in phenomenological methods are highly dynamic, however, traditional reliability methods based on statistical means and steady-state behaviors are often not appropriate in these situations. Instead, physical modeling is needed to understand how potentially small off-nominal events or deviations can cascade into more serious failures. These interactive, unsteady processes do not follow statistical behavior and must be evaluated through time-dependent algorithms that dynamically track the key conditions of the system over time. The physical models that are used to define the processes, whether they are steady or unsteady behaviors, are mathematical engineering models based on deterministic equations.

Because physics-based models depend on the laws of nature, they are not heavily dependent on expert opinion or failure data developed through traditional reliability techniques or data sources such as military handbooks to determine expected outcomes. The mean time to failure cannot be evaluated using handbooks because the system state is constantly evolving and the outcome is dependent on these evolving conditions. Physics-based models also do not specifically address such things as human factors or human error since they are not governed by laws of nature.

10.2 Phenomenological Modeling During the Design Process

A key attribute of phenomenological modeling is that it can be effectively employed throughout the design cycle. An evolving range of physics-based analysis techniques can be employed to address the most relevant, risk-driving performance and safety factors at each point during the design cycle, using different sources and levels of data as they become available. During the early phases of development, high-level physical models can focus on the

general characteristics of a system and gross behaviors of the proposed designs. As the design matures, higher fidelity models can be developed to address more specific characteristics and system details. Each analysis phase requires using methods and data that are of the appropriate precision for the maturity of the system design and the varying level of details that are available at each stage. There is no “one size fits all” solution and understanding which modeling approach to use at each phase is critical to defining an effective analysis process.

The choice of model fidelity and complexity depends on the specific questions being asked at each phase of development. Selecting the proper model or models can be done as an iterative process, where simple, engineering-level models are chosen first to understand the dominant physical processes and how they generally interact. The proper set of models may also be chosen with an adaptive process, where simple models are used to identify local conditions, which then provide the inputs to more sophisticated models. A strong analysis does not necessarily require the incorporation of detailed physics models to answer every question. Rather, a strong analysis requires models that increase the understanding of the interactions between systems, environments, and their changing states. As further information is developed, more sophisticated, higher-fidelity models can be introduced to reflect more accurately the evolution of the processes and answer finer-level questions as needed, but only when the existing models become insufficient.

The amount and quality of the input and system data will also evolve and needs to be incorporated appropriately as the system design solidifies. While data are sparse and less detailed in the early system design phases, the ability to develop useful inputs from other sources is important. Understanding the sources and magnitude of uncertainty in the available data and reconciling them with their potential impact on the overall design choices is a major aim of phenomenological modeling in the early analysis cycle. Detailed analyses with insufficient or placeholder data are much less meaningful than general analyses developed with uncertainty ranges that accurately reflect the state of knowledge and of the design. In fact, the uncertainties that are identified in the early design phases provide a valuable mechanism for developing a clearly defined, risk-based analysis path by logically exposing the next most pertinent set of design issues or weaknesses. As details of the system are defined and understanding of system interactions increases, the associated modeling uncertainties will generally decrease and actively managing them through focused studies or design changes will become less important.

The nature of the dominant sources of failures and errors will also change as the design matures and failure modes are successively driven down. These evolving failure modes span a spectrum of fundamentally different problems, initiators, and solutions. Potential failure modes must therefore be explored and discovered through systematic but flexible techniques that are targeted at exposing the specific, risk-driving features at each design phase. The techniques should come from a variety of methods that complement one another in order to provide thorough coverage and depth of understanding.

Following is a list of different techniques that can be used:

- Top-down parametric – identify general behaviors
- Dispersion studies – identify potential weaknesses, vulnerabilities
- Sensitivity studies – determine which dispersive (and hard to manage) parameters impact the system the most
- Calibration and validation – check model assumptions with pertinent tests
- Refinement – update and improve models and data

10.3 Stress-Strength Formulation of Physical Models

Probabilistic “stress-strength” models were originally formulated to predict the reliability of structural elements in civil and mechanical engineering calculations. The term derives from the fact that they are based on the estimation of the probability distributions of the mechanical “stress” applied to a structural component and of the “strength” of its constituent material. The probability of failure (POF) of the component is then calculated as the probability that the applied stress may exceed the inherent strength. This type of formulation can be generalized to a formulation in which the “strength” of a component is represented by a parameter that describes the component capability in a particular dimension, and this capability is probabilistically compared to a demand parameter, i.e., the “stress” applied to the component. This generalized form is often encountered in the use of physical models, to carry out quantification of standard PRA binary models such as FTs and event sequence diagrams (ESDs).

For example, the ESD in Figure 10-1 describes the possible sequences of events following an attitude control malfunction of a launch vehicle (LV) at lift-off. In this particular ESD, both the branch point that determines whether there is sufficient time for the Flight Control Officer (FCO) to activate the launch vehicle Flight Termination System (FTS) before ground impact occurs, and the branch point that models whether an automated destruct is triggered (as a result of vehicle breakup induced by structural failure) before ground impact, can be quantified using probability values obtained from “stress-strength” models and underlying physical models.

In the first of the two branch points mentioned above, the probability of successful FCO destruct action can be estimated by comparing the time to ground intact impact of the LV, which is the “demand parameter” chosen to represent the effectiveness of the FTS, with the response time of the FCO for FTS actuation, which is the parameter that represents FTS capability. Similarly, in the latter branch point, the probability of successful automated destruct action can be quantified by comparing the time to ground intact impact of the LV (stress, or demand, parameter) with the time to LV structural breakup under the dynamic loading induced by the attitude control malfunction (capability/strength parameter).

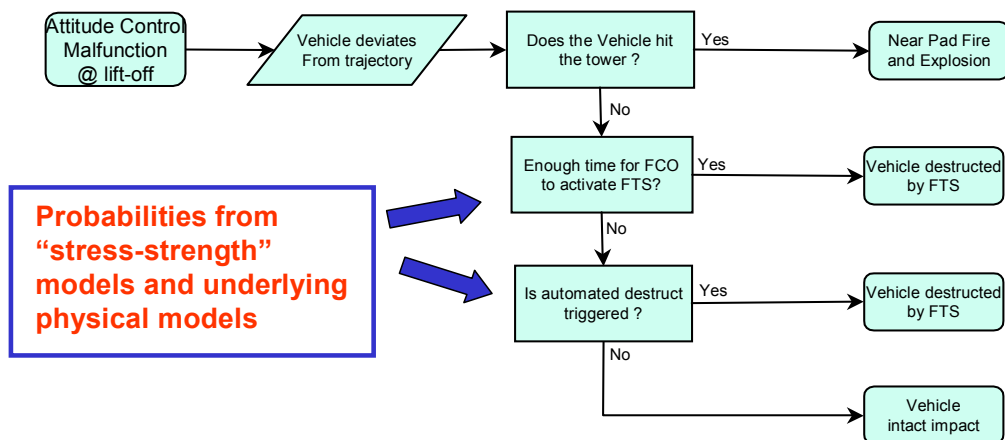


Figure 10-1. Event Sequence Diagram for Attitude Control Malfunction at Lift-off.

Figure 10-2 illustrates how the comparison is carried out in probabilistic terms, using the first of the two previously mentioned branch point estimations as the example. The integral formulation in the box inset uses the probability density functions (pdfs) of the stress and strength parameters and represents the probability that the strength parameter is greater than the stress. This formulation is the instantiation, for the example discussed, of the general expression of the probability of success (POS), or reliability, of a system function expressed in stress-strength terms. A generalized formulation is given by:

$$POS = \Pr(E > \Sigma) = \int_0^{\infty} f_{\Sigma}(\sigma) d\sigma \int_{\sigma}^{\infty} f_E(\varepsilon) d\varepsilon \quad (10-1)$$

where E denotes the strength parameter and Σ denotes the stress parameter.

We note that quantification of stress-strength physical models is often based on Monte Carlo techniques, due to the inherent difficulty of obtaining a closed-form solution for the associated integral formulations, or even of the pdf terms that appear in Equation (10-1), given the complexity of the variable factors that determine their actual forms. For example, the time to intact impact, after a guidance or propulsion system malfunction immediately after LV lift-off, depends on several parameters that are affected by variability and randomness. These include the nature of the guidance or propulsion system failure mode that causes the LV attitude malfunction, the time of the initial failure, and the wind direction and velocity affecting the vehicle trajectory.

A probabilistic physical model for the time to intact impact can be set up as follows:

- Step 1: Assume variability distributions for the above basic parameters.
- Step 2: Use Monte Carlo sampling to draw an input parameter-set to use in the flight dynamics calculation.
- Step 3: Calculate time to impact according to the flight dynamics model applicable to the LV of interest.
- Step 4: Repeat with randomly drawn input parameter sets enough times to obtain a good approximate representation of the time-to-impact distribution.

The process outlined above can be used to obtain a distribution for the time to LV intact impact, T_i , like the one drawn for illustration in Figure 10-2. A probability distribution for the FCO response time and FTS activation, T_a , is also shown in Figure 10-2 and can be obtained by using a human response model, reflecting the human reliability modeling concepts discussed in Chapter 8.

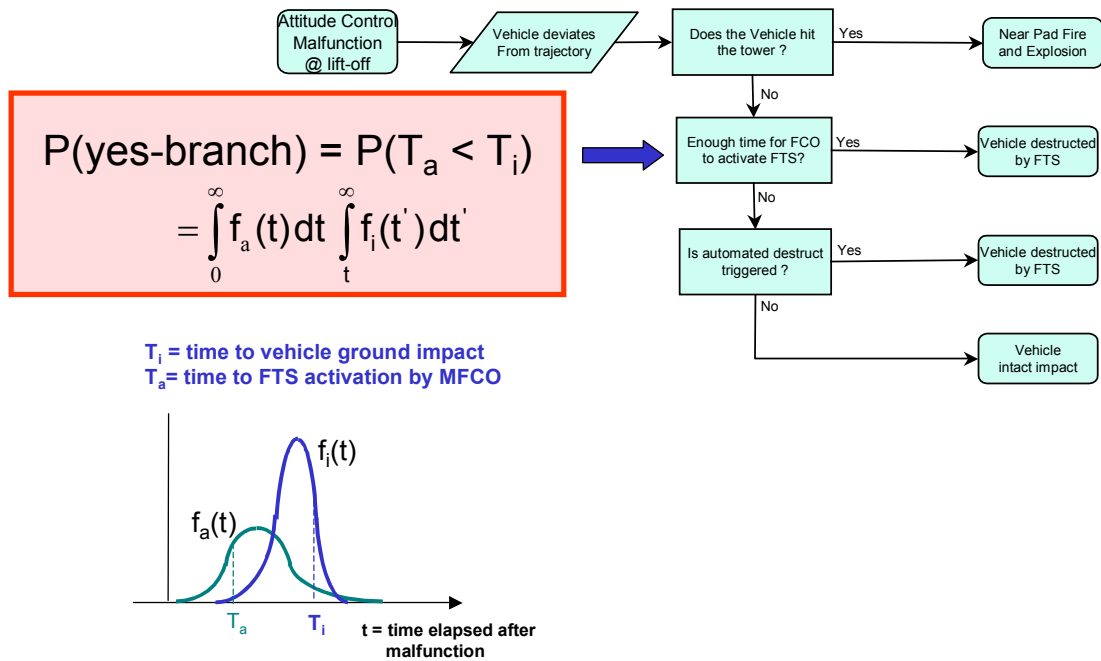


Figure 10-2. Probability Distributions for Time to LV Ground Impact and Time to FTS Activation by FCO.

As outlined earlier, the probability for the yes branch for the question box “Enough time for FCO to activate FTS?” in Figure 10-2 can be obtained using the distributions for T_i and T_a .

Thus, as stated at the beginning of the section, the evaluation of a “Stress-Strength” model can yield probability values for PRA binary models.

10.4 Range Safety Phenomenological Models^a

In space system PRAs, phenomenological models are often encountered in the backend, i.e., consequence evaluation portion of the PRA analysis, where they are often needed to estimate casualty or health effects impacting range safety, e.g., effects associated with launch vehicle accident scenarios. Examples of these “range safety phenomenological models” include:

- “*Inert debris*” phenomenological models: used to carry out probabilistic estimations of injury or loss of life that may be caused by direct launch vehicle/spacecraft debris impacts on populated areas as a result of a launch accident.

a. Some of the launch risk and blast risk materials and examples discussed in this chapter refer to methodology and information originally produced by ACTA Inc. in support of Eastern and Western Range Safety Office activities and of U.S. Air Force and NASA launch activities [10-1,10-2]. The discussion on re-entry risk is based on methodology developed by The Aerospace Corporation in support of U.S. Air Force launch activities.

- *“Blast Impact” models*: used to implement the corresponding estimations for the effects of blasts caused by solid or liquid propellant explosive impacts on populated areas.
- *“Ray Focusing” models*: used to take into account local blast overpressure concentration effects in estimates of the probability of casualties or injuries produced by window breakage induced by launch vehicle explosions.
- *“Plume Dispersion” models*: used to estimate the health effects of toxic plumes produced by launch vehicle propellants during a nominal mission or in an accident.

These phenomenological consequence models all have similar characteristics. They seek to quantify the variability of physical processes that determine the effects of a system failure on a specific area of impact, such as the effects on public health or the environment. They are, at least in part, based on deterministic knowledge of the particular physical processes of interest. In addition, they all use probabilistic methods to assess the aleatory and epistemic uncertainty affecting key physical process parameters that are input variables for the overall model and the effects of this variability on the output parameter(s).

A subset of these phenomenological models will be discussed, and examples will be provided in the following subsections.

10.4.1 Inert Debris Impact Models

Inert debris impact models are the simplest kind of phenomenological consequence evaluation models. They are used to estimate “Casualty Expectation” (E_C) effects. E_C is a measure of collective public risk and is defined as the expected number of casualties in a geographic area for a single launch. The baseline criterion for E_C is defined by the range safety requirements in EWR 127-1, which state that the maximum acceptable E_C , without a waiver, summed over all geographic areas for a single launch is 30 in a million.

An example of an inert debris impact model is the Launch Risk Analysis (LARA) program. LARA is an approach implemented in a software tool that can be used to evaluate the compliance with EWR 127-1 at the Western and Eastern Ranges. The LARA model is used for all Space Shuttle launches.

A synopsis of the LARA approach (Figure 10-3) can be summarized in the following steps:

1. Select a flight time interval and assume a failure occurs.
2. Select a specific failure mode and the resulting vehicle breakup mode.
3. Given the mode of vehicle breakup, focus on a particular fragment and develop the impact point distribution for the selected fragment.
4. Using the fragment impact point distribution and the pdf, estimate the E_C for the fragment.
5. Weight the casualty expectation result with the POF during the selected time interval and sum to obtain the total risk profile.

Since debris impact risk is affected by variability in the physical and launch vehicle parameters, such as vehicle guidance and performance deviations, variability in the vehicle attitude due to control malfunction, wind uncertainties, and variability in the debris aerodynamic characteristics and the fragment perturbation velocities, Monte Carlo techniques can be superimposed on a model framework like LARA. This integrates deterministic and probabilistic models into an overall E_C risk model that takes into account the above described variability and uncertainty factors.

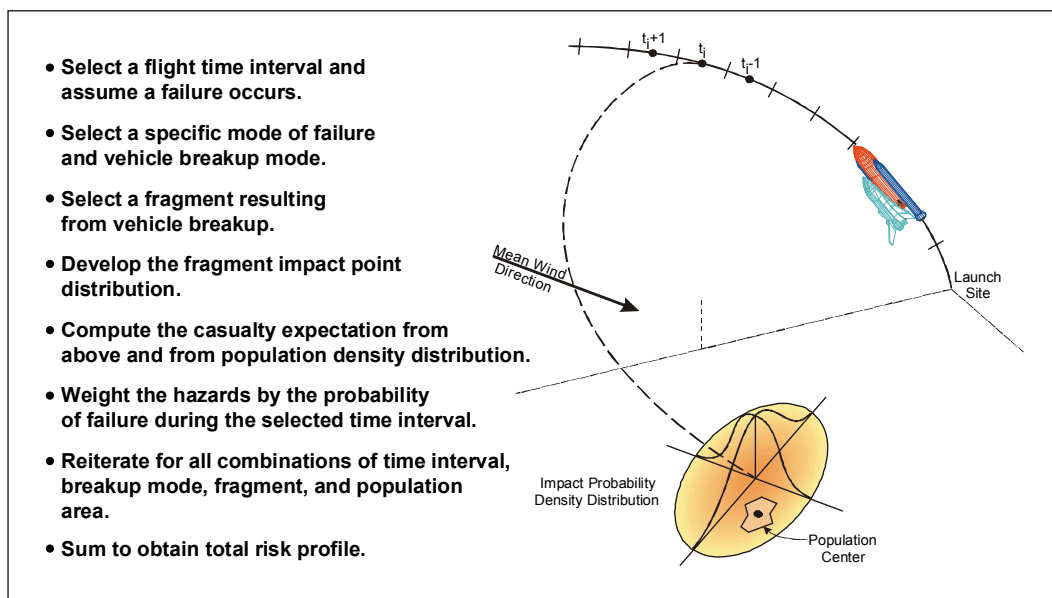


Figure 10-3. Synopsis of the LARA Approach.

10.4.2 Blast Impact Models

Like inert debris impact models, blast impact models also seek to estimate casualty expectation risk. The modeling process for blast impact models is similar to that used for debris impact models but accounts also for the explosive yield of solid propellant fragments. Effects of liquid propellants are not usually considered because of the very low probability of undispersed liquid propellant impacts on public land. Even though re-entry of solid upper-stages from high altitude can also cause blast impact damage, this type of accident is treated with different models because of the typical absence of FTS safeguards against this type of scenario and the different considerations that go into the evaluation of re-entry trajectories. Some specific aspects of re-entry risk models will be discussed in Section 10.4.3.

A typical blast impact model includes the representation of explosive impact and glass breakage risk and is used to evaluate EWR 127-1 compliance. It considers several types of launch vehicle accident / breakup scenarios, mostly concerning low altitude accident initiation and solid fragment effects. The blast impact model calculates E_C induced by solid fragment explosive yield and glass fragments produced by window breakage. The dataflow for a blast impact model is summarized in Figure 10-4. On the input side are:

- The population density descriptions,
- Breakage and casualty versus overpressure tables,
- Explosive yield histogram computed according to an impact model,
- Terrain information, including water and land locations, and
- Wind and temperature covariance information.

These inputs are combined in the blast model with real-time weather data to generate outputs such as casualty statistics, breakage statistics, risk profile, overpressure map, focusing map, breakage map, and sonic velocity profiles. Similar to inert debris impact models, blast models also utilize Monte Carlo simulations to integrate deterministic and probabilistic inputs.

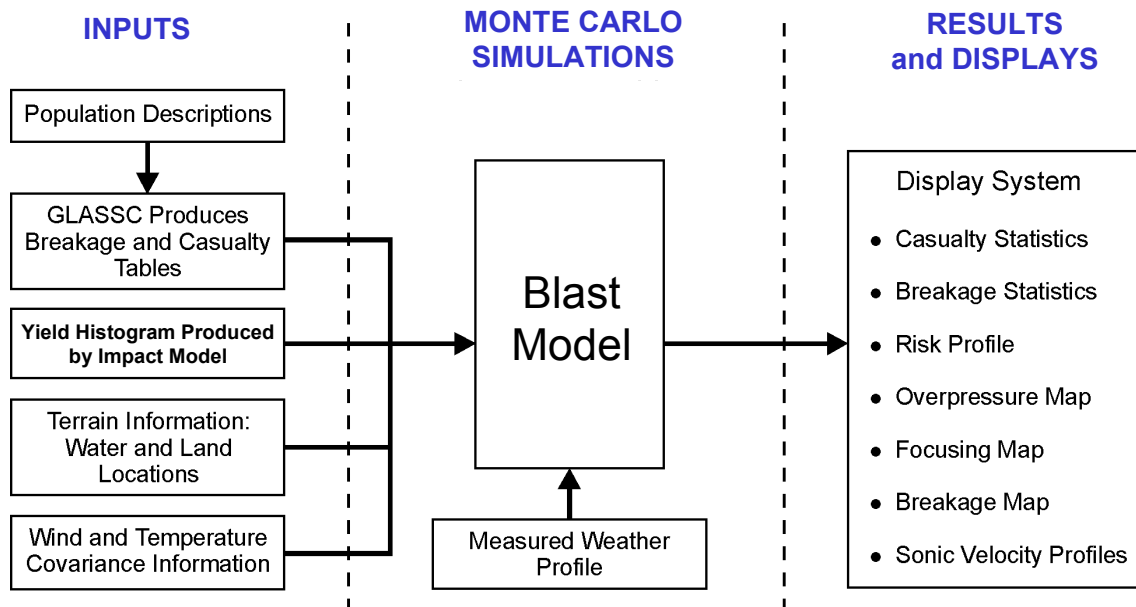


Figure 10-4. Dataflow for Blast Impact Model.

As a prerequisite for the blast model calculation, the impact calculation results are first generated as a yield histogram using an impact model. Within the impact model, Monte Carlo simulation is implemented to compute the explosive yield probability. The simulation procedure can be summarized in the flowchart shown in Figure 10-5. First, a launch vehicle failure scenario is defined. For that failure scenario, the launch vehicle failure condition (failure mode and flight time) is sampled. Given the launch vehicle failure condition, the expected yield and impact location for propellant debris are computed by accounting for the failure mode, the simulated destruct and breakup logic, the ensuing fragmentation scenario, the destruct-induced velocity perturbation, the impact mass and velocity, and the impacted surface hardness. The aggregate results are obtained by repeating the sampling procedure and calculation for other failure scenarios.

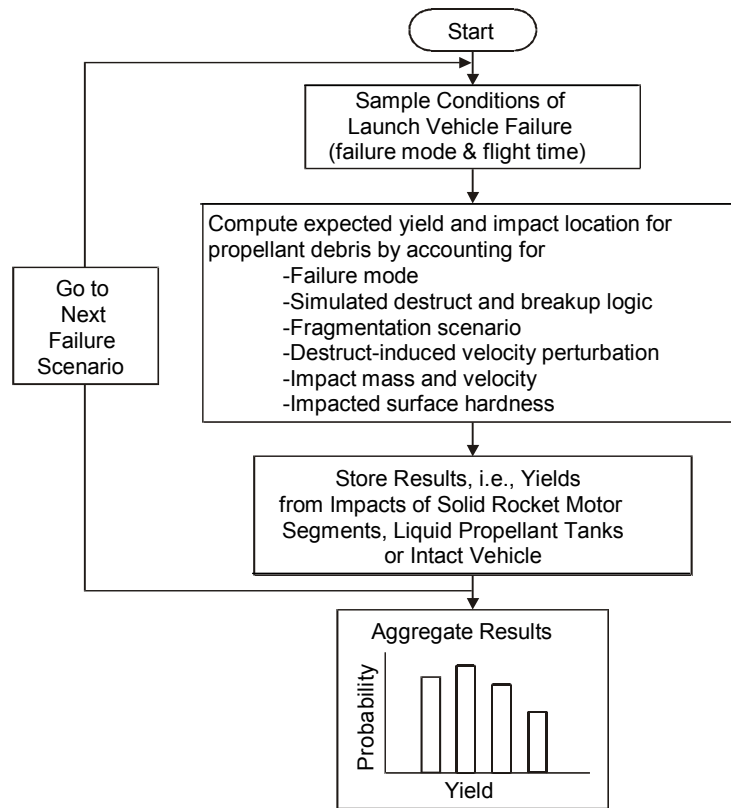


Figure 10-5. Monte Carlo Simulation for Explosive Yield Probability Computation.

For example, to estimate the Titan IV solid rocket motor unit (SRMU) blast risk, failure scenarios (Figure 10-6) are first defined using approaches that combine ESDs (binary logic models) and physical models. These failure scenarios are then analyzed in the impact model to generate yield probability results.

LV Outcome Scenarios and Probabilities

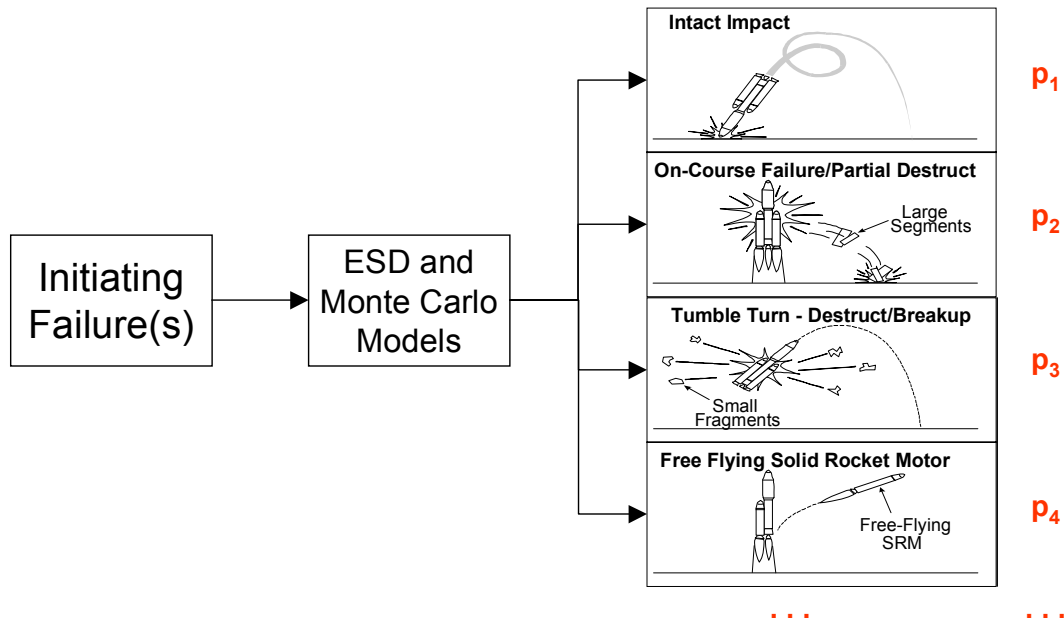


Figure 10-6. Titan IV-SRMU Blast Scenarios.

The yield probability results are in the blast model in combination with breakage and casualty versus overpressure tables (Figure 10-7) to obtain the E_C . Since atmospheric conditions are known to influence strongly the far-field overpressure propagation and subsequent damage potential, real-time weather data propagated through ray-focusing models (Figure 10-8) is a key input for the blast model calculation. Examples of outputs produced are shown in Figure 10-9.

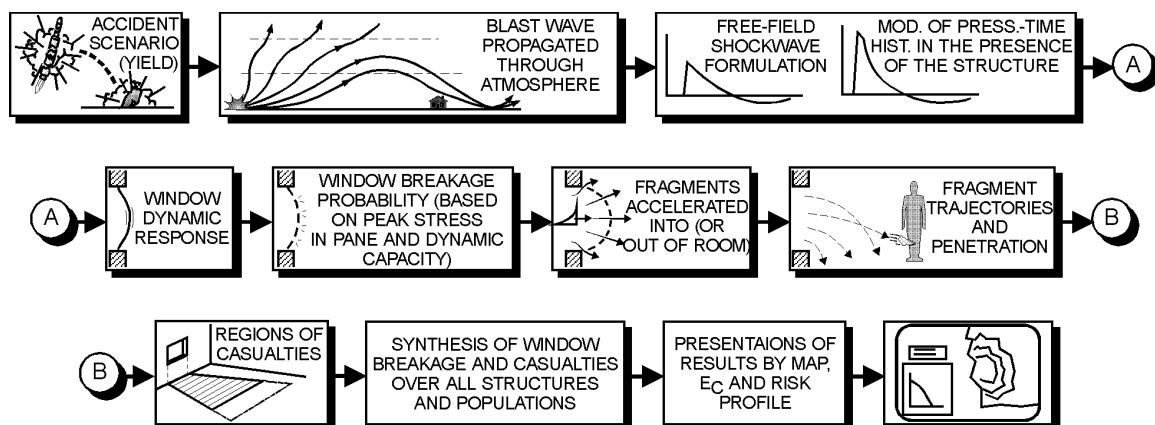


Figure 10-7. Glass Breakage Risk Analysis Modeling Process.

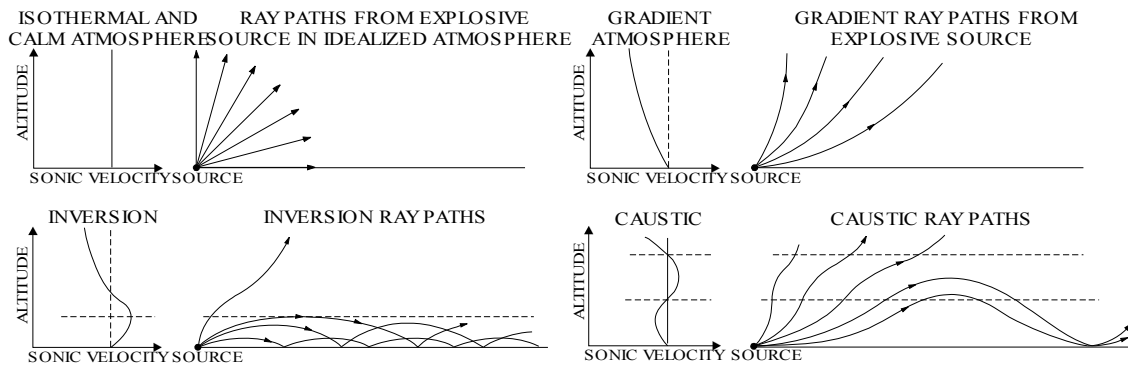


Figure 10-8. Models for Overpressure Propagation.

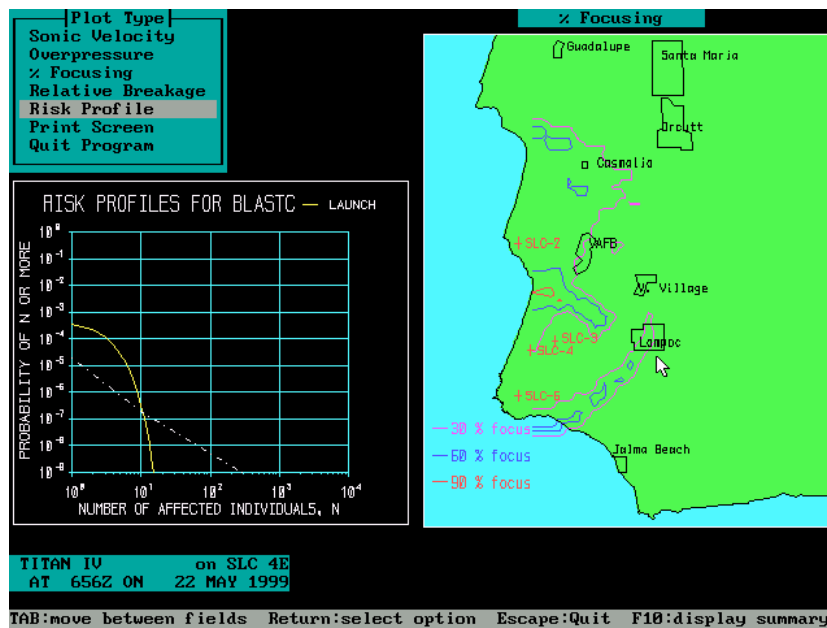


Figure 10-9. Blast Risk Analysis Output.

10.4.3 Re-Entry Risk Models

Re-entry risk models address high altitude accident scenarios resulting in possible ground impact. As mentioned in Section 10.4.2, re-entry of solid propellants from high altitude can cause blast and impact damage similar in nature to that which may result from a launch area accident. However, the area affected by re-entry accidents is much larger. In addition to focusing on the types of accident scenarios and re-entry debris that may result from these, this modeling approach hinges on trajectory simulation for the launch vehicle. For example, for a Titan-IV IUS (Inertial Upper Stage) mission launched from the Eastern Range, the vacuum-IIP (Instantaneous Impact Point) trace (Figure 10-10) crosses the African and Australian continents. Hence, in the case of an IUS solid rocket motor re-entry accident, there are potential risks to the populations in the aforementioned land masses.

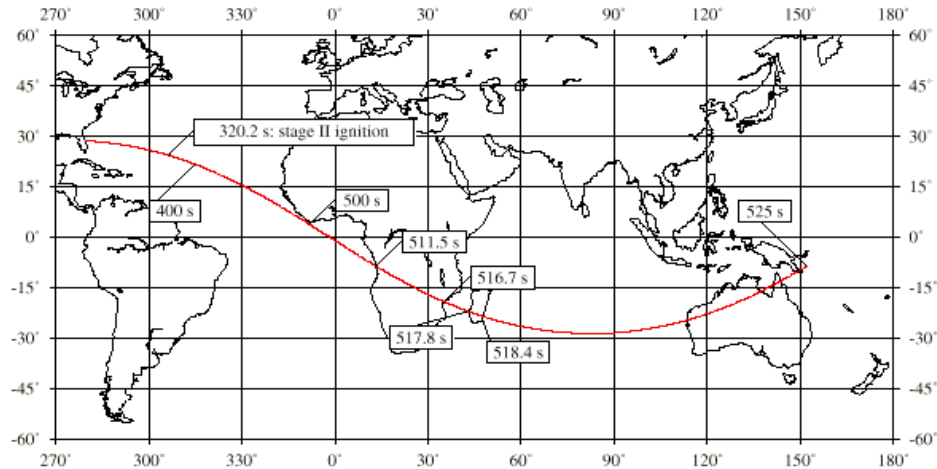


Figure 10-10. Vacuum IIP Trace for a Titan IV/IUS Mission.

By comparing the IIP trace with population density data, the risk areas can be identified. To estimate the risk more specifically in terms of casualty expectation, for the affected areas, phenomenological models combine the time-distribution of IUS failure probability with trajectory and IIP dispersion models in a Monte Carlo simulation to yield the time/space distribution of IUS solid fragment impact probability. This time/space distribution solid fragment impact probability is then combined with population density distribution to yield the E_C distribution (Figure 10-11). In the E_C calculation, explosive yield is also taken into account, typically as a factor defining an equivalent impact surface. This equivalent impact surface can be considerably larger than the area mechanically affected by the fragment impacts.

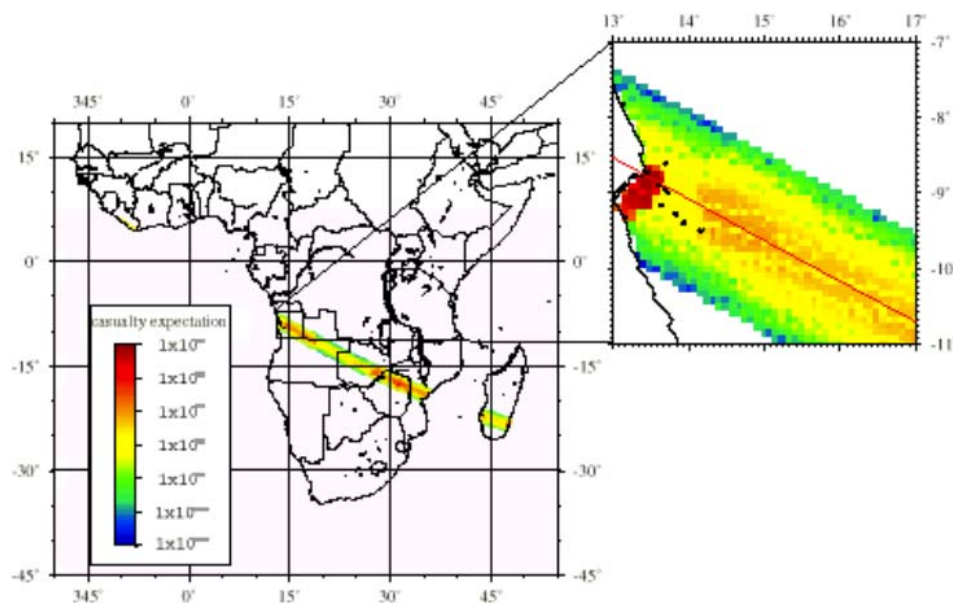


Figure 10-11. Casualty Expectation Distribution in Re-entry Accidents.

10.5 MMOD Risk Modeling

In the term Micro-Meteoroid and Orbital Debris (MMOD) is generally used to refer to any kind of small-size (on the order of 1 cm in diameter or less) body traveling in space outside of the Earth's atmosphere. The term Orbital Debris refers to material that is in orbit as the result of space launches but is no longer serving any function.

Sources of Orbital Debris include discarded hardware, such as Spent Launch Vehicle Upper Stages left in orbit, Space Vehicles left in orbit after useful-life expiration, and deployment and separation hardware, as well as fragments produced by collisions, explosions, or byproducts of Space Vehicle solid rocket motor combustion. Orbital Debris also include degradation products such as paint flakes, insulation particulates, and frozen leaked-fluids (e.g., nuclear reactor coolant leaked from Russian RORSATs).

10.5.1 Risk from Orbital Debris

MMOD generally move at high speed with respect to operational spacecraft. In low Earth orbit (< 2,000 km) the average impact velocity relative to the latter is 10 km/s (~ 22,000 mi/hr). At 10 km/s a 1.3 mm diameter aluminum particle has the same kinetic energy as a .22-caliber long-rifle bullet. At geosynchronous altitude, average impact velocity is lower (~ 200 m/s), but considerable damage can still result. For example, a 1 cm object in geosynchronous orbit has damage potential similar to a 1 mm object in low Earth orbit.

If a relatively large fragment impacts a spacecraft, a "debris cloud" of smaller fragments is generated around the orbit of the impacted spacecraft, spreading progressively in spiral motion until it envelops the entire orbit.

10.5.2 MMOD Risk Modeling Framework

Note that while NASA has developed specialized methods for dealing with MMOD risk, this section treats the issue in a more general fashion. For additional detail, the reader is encouraged to consult NASA documents such as NASA-HDBK 8719.14, Handbook for Limiting Orbital Debris [10-3].

A basic framework for estimating spacecraft damage risk may be set up in a fashion that conceptually reflects the ET shown in Figure 10-12 and the corresponding risk representation given below:

$$\text{Probability of Mission Loss} = P_I \cdot P_{C/I} \cdot P_{D/C} \quad (10-2)$$

The formulation provided by Equation (10-2) is a conditional probability formulation that expresses the MMOD-induced Probability of Mission Loss as the product of the probability of an MMOD impact on the spacecraft or launch vehicle of concern, P_I ; the probability that a critical system component is affected by the impact (given that an impact has occurred), $P_{C/I}$; and the probability that fatal damage of the critical component results (given that such a component has been affected by the impact), $P_{D/C}$.

The following sections discuss typical approaches for the estimation of the probability terms that appear in Equation (10-2).

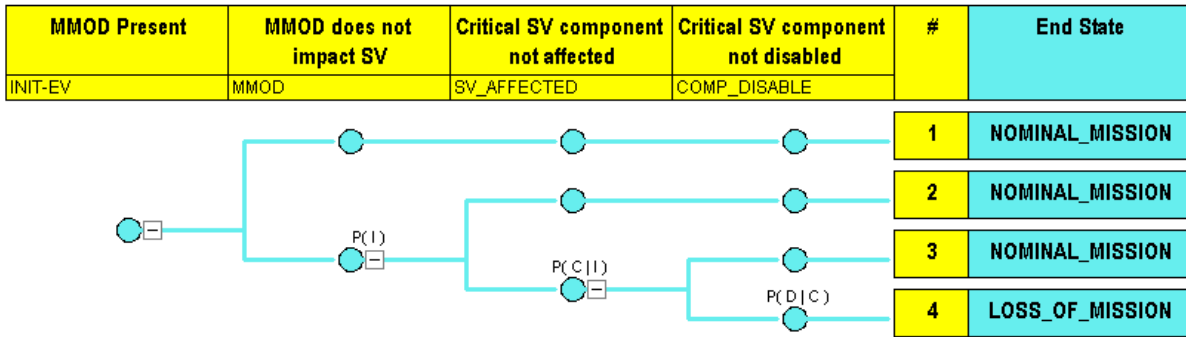


Figure 10-12. Conceptual MMOD Event Tree Model^a.

10.5.3 Probability of MMOD Impact P_I

Computer models like ORDEM (NASA JSC) [10-4] and MASTER (ESA) provide MMOD flux distribution in space, direction, and velocity, as well as average total cross-sectional flux, F , for a specified spacecraft orbit. Although exact formulations of collision probability require complex integral calculations, a simplified calculation of collision rate, CR , can be obtained from the approximate formula:

$$CR = \frac{A}{4} \cdot F \quad (10-3)$$

Where:

- CR = collision rate [impacts/yr]
- A = total spacecraft surface area [m^2]
- F = total cross-sectional flux [particles/ m^2 /yr]

Then an approximate expression for the probability of impact can be simply obtained as:

$$P_I = CR \times MT \quad (10-4)$$

Where:

- MT = mission time duration [yrs].

10.5.4 Probability of MMOD Impact Affecting Critical SV Components, $P_{C/I}$

This conditional probability is essentially a reflection of the spacecraft geometry. A simplified estimation can be based, as illustrated by Figure 10-13, on the calculation of the approximate ratio between the sum of non-overlapping cross-sectional areas of critical components located near the spacecraft outer surfaces and the total spacecraft cross-sectional area, i.e.:

$$P_{C/I} = \sum_i \frac{a_{xi}}{A_x} \approx \sum_i \frac{a_{xi}}{A/4} \quad (10-5)$$

a. Here, SV stands for Spacecraft or launch Vehicle

Where:

- a_{xi} = cross-sectional area of i-th critical component
- A_x = spacecraft cross-sectional area $\sim A / 4$
- A = spacecraft surface area

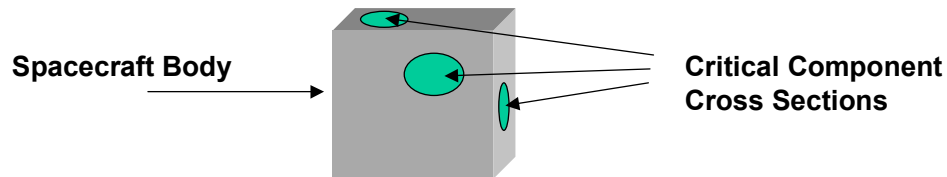


Figure 10-13. Approximate Calculation of Probability of MMOD Impact Affecting a Critical Component.

10.5.5 Probability of Critical Component Damage, $P_{D/C}$

Detailed damage probability modeling can be quite complex due to consideration of component geometry and material non-homogeneity. Complex and laborious hydrocode computer modeling and/or Ballistic Limit Equations (empirically based on ground “hypervelocity” impact tests) are used to model the impact damage dynamics.

Approximate estimations can be obtained in terms of simplified “stress-strength” models, using the “kinetic energy content” of MMOD flux as the stress-term and the “energy absorption capacity” of any materials shielding a critical component as the strength-term.

Assuming that critical component failures are guaranteed to occur if MMOD goes through the shielding material, the conditional probability of damage $P_{D/C}$ —i.e., the probability that fatal critical component damage results, given that an impact has occurred on an area where such a component is located—can be obtained by estimating the integral stress-strength formulation:

$$P_{D/C} = P(E_K > E_C) = \int_0^{\infty} f_C(e) de \int_e^{\infty} f_K(e') de' \quad (10-6)$$

Where:

- E_C = energy absorption capacity of SV shielding material
- $f_C(e)$ = pdf of shielding material energy-absorption capacity
- E_K = kinetic energy of impacting MMOD flux
- $f_K(e)$ = pdf of impacting MMOD flux kinetic energy.

10.6 Ground-Based Fire PRA

The risk due to fires in a ground-based facility is ultimately estimated by applying Equation (6-1). For a particular end state of interest, the frequency at which this end state results from fire, $\Lambda(\text{ESF})$, is:

$$A(ESF) = \sum_{j=1}^J A_j(ESF) \quad (10-7)$$

where $A_j(ESF)$ denotes the frequency at which fires originating in location, j , cause the end state to occur.

The summation in Equation (10-7) implies that fire risk is location dependent. This geometric aspect of the assessment is necessary because the:

- Frequency at which fires ignite;
- Probability the fire propagates; and
- Amount of equipment (or number of personnel) that could contribute to the end state if damaged (or injured) by fire;

are location dependent. Consequently, an initial step in performing a fire risk assessment is to divide the facility into separate regions or fire zones. If U symbolizes the entire space occupied by the facility and Z_j designates the j th fire zone, then:

$$U = \bigcup_{j=1}^J Z_j \quad (10-8)$$

and, in a Boolean context, the fire zones are mutually exclusive. With respect to Equation (10-7), the summation is over the facility fire zones.

Focusing on the j th fire zone, signify by λ_j the frequency at which fires are ignited within the zone. This is primarily dependent upon:

- The quantity and type of combustible material (including transient combustibles); along with
- Ignition sources;

located within the zone. Normally, an inventory of fire zones is performed that identifies the combustible material loadings and potential ignition sources within each. Using this facility-specific information, generic statistical databases are consulted in order to ascertain prior distributions for each λ_j . Posterior distributions are derived by combining these generic distributions with fire records from the facility using Bayes' Theorem. The process utilizes the techniques described in Sections 6.6 and 5.3.

Fire propagation must next be considered. The concern is that if a fire is ignited within a particular zone, no risk impact will result unless the fire causes damage to equipment, or injury to personnel. Note that if personnel injury is a risk assessment end state, the safety of all personnel who enter the zone to fight the fire must also be considered.

Fire propagation is a complex process that usually augments generic statistical data with computer simulations. In addition to modeling the combustion of materials located within the fire zone, a propagation analysis must also evaluate possible detection (by automatic sensors and personnel), as well as suppression (either by automatic fire suppression equipment or facility personnel). If the fire is detected and suppressed before it damages any equipment or injures any personnel, no seriously adverse end states result. However, if the fire:

- Remains undetected; or
- Is detected but not suppressed;

before injury or damage occurs, then it is necessary to determine whether any undesirable end states ensue. If personnel injury is an end state, then occurrence of personnel injury resulting from the fire directly results in that end state. No further analysis is required.

End states not involving personnel injury may require further examination, even if equipment damage is caused by the fire. This is especially important if the facility includes redundant system designs. The issue is illustrated in Figure 10-14.

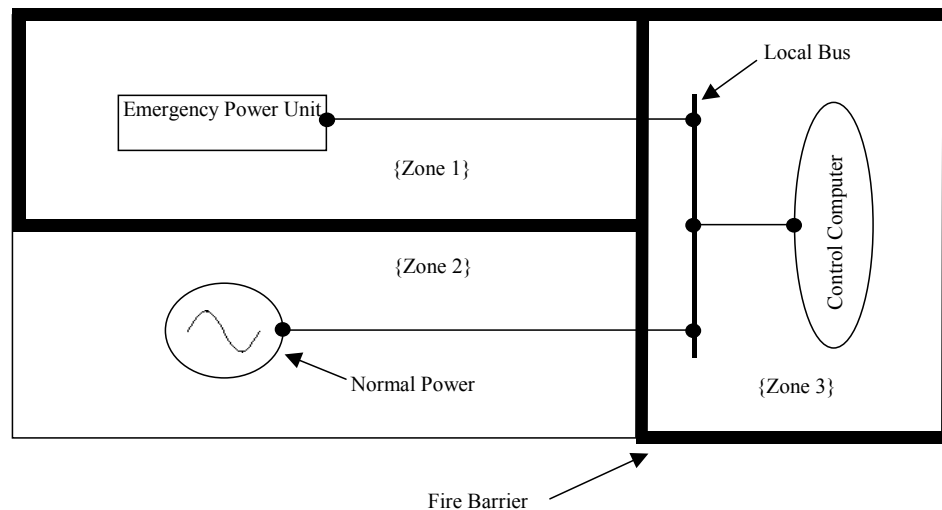


Figure 10-14. Facility Power Schematic.

Figure 10-14 depicts power distribution to a Control Computer. Three fire zones are identified:

1. Zone 1 contains the Emergency Power Unit and is enclosed by a fire barrier. The presence of the fire barrier is significant because it inhibits the spread of fire from one zone to another.
2. Zone 2 is not completely enclosed by a fire barrier. Since it contains the normal power source, the second fire zone could be an outdoor switch yard where the facility connects to the power grid.
3. The facility Control Computer resides in Zone 3. Like the Emergency Power Unit in Zone 1, it is protected by a fire barrier to inhibit zone-to-zone propagation.

If the end state of interest is loss of the Control Computer, Figure 10-15 is the corresponding FT.

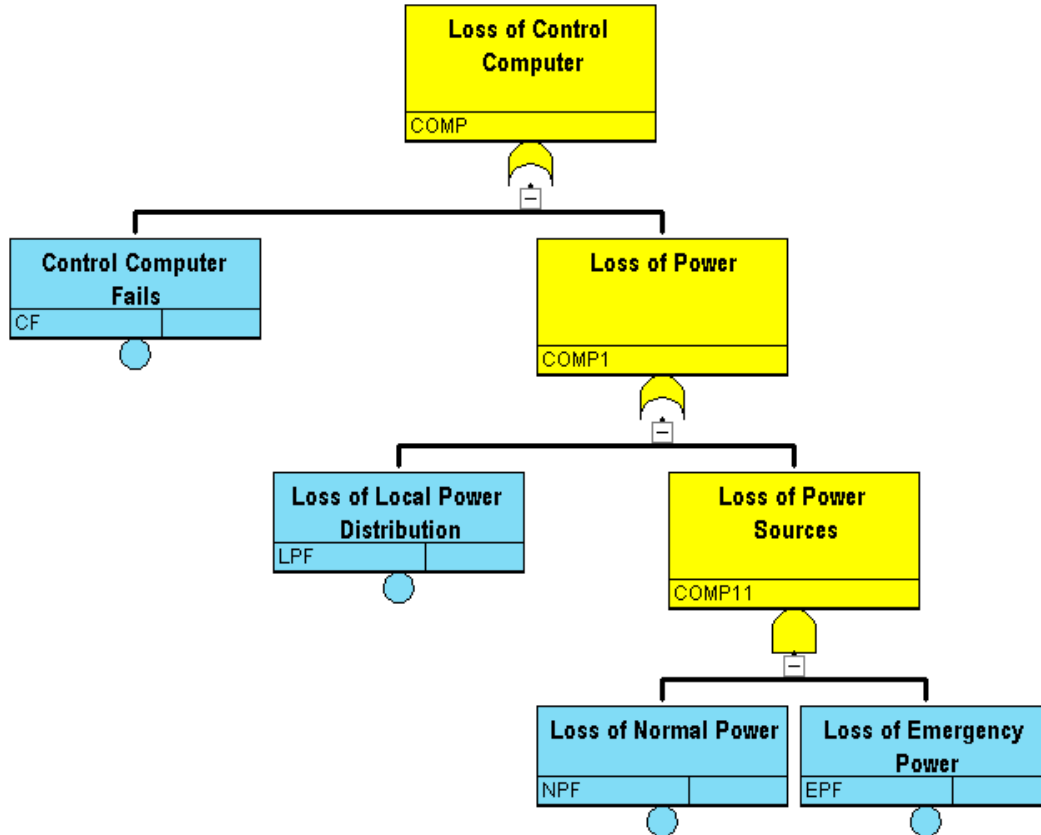


Figure 10-15. Fault Tree for Loss of the Control Computer.

The Boolean equation for Figure 10-15 is:

$$ESF = CF \cup LPF \cup (NPF \cap EPF) \quad (10-9)$$

If phenomenological events were being excluded from the PRA, event LPF is unlikely to be included in the model of the world because local power distribution, comprised of the:

- Local bus; and
- Electrical connection between the local bus and computer;

has such a low failure rate (since the components are passive) that its contribution to risk is negligible. However, because wires and cables are vulnerable to fire, loss of local power distribution is an event in Figure 10-15.

Returning to Equation (10-7):

$$A(ESF) = \sum_{j=1}^3 \lambda_j \Pr(ESF|F_j) \quad (10-10)$$

Here, $\Pr(ESF|F_j)$, is the conditional probability that the end state results, given that a fire starts in Zone j .

An actual ground-based facility will have numerous fire zones. However, even for the simple illustration in Figure 10-14, it is evident that Equation (10-10) can become complicated if zone-to-zone fire spreading is assessed in detail. This is depicted in Table 10-1, which lists the combinations of events needed for a fire initiated in a certain zone to cause end state ESF. Fortunately, the information in Table 10-1 can afford a basis for screening certain combinations of events that will contribute only negligibly to the PRA results.

Table 10-1 (and Figure 10-14) demonstrate that a fire confined to Zone 1 or Zone 2 cannot, by itself, cause end state ESF. In order for end state ESF, to ensue, independent failures of other systems must occur in conjunction with the fire. This is depicted in the Figure 10-16 ET.

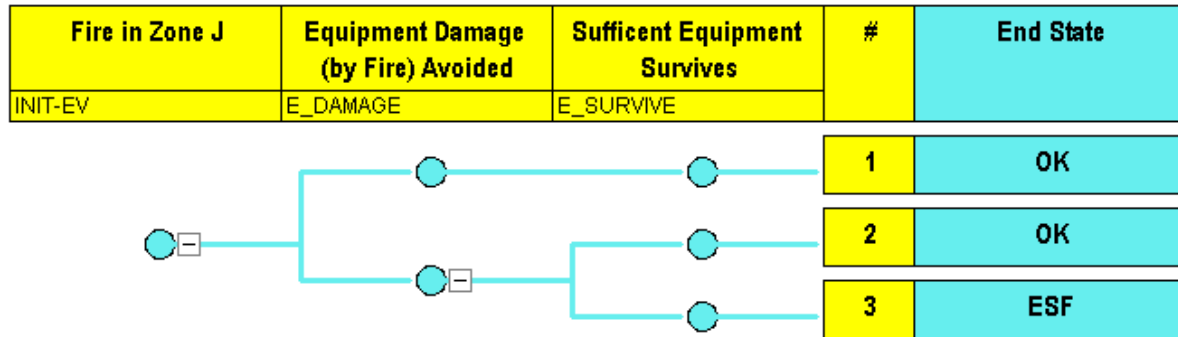


Figure 10-16. Facility Fire Event Tree.

Table 10-1. Fire Progression.

Originating Zone	Propagation to Zone	Equipment Damaged	Other Equipment Failures Needed for End State ESF Occurrence
1	None	Emergency Power Unit	Yes
		Emergency power cable	Yes
	2	Emergency Power Unit	Yes
		Emergency power cable	Yes
		Normal power	Yes
		Normal power cable	Yes
		Emergency Power Unit AND normal power	No
		Emergency Power Unit AND normal power cable	No
		Emergency power cable AND normal power	No
		Emergency power cable AND normal power cable	No
	3	Emergency Power Unit	Yes
		Emergency power cable	Yes
		Local power distribution	No
		Control Computer	No
	2 and 3	Emergency Power Unit	Yes
		Emergency power cable	Yes
		Normal power	Yes
		Normal power cable	Yes
		Emergency Power Unit AND normal power	No
		Emergency Power Unit AND normal power cable	No
		Emergency power cable AND normal power	No
		Emergency power cable AND normal power cable	No
		Local power distribution	No
		Control Computer	No
2	None	Normal power	Yes
		Normal power cable	Yes
	1	Normal power	Yes
		Normal power cable	Yes
		Emergency power	Yes
		Emergency power cable	Yes
		Emergency Power Unit AND normal power	No
		Emergency Power Unit AND normal power cable	No
		Emergency power cable AND normal power	No
		Emergency power cable AND normal power cable	No
	3	Normal power	Yes
		Normal power cable	Yes
		Local power distribution	No
		Control Computer	No
	1 and 3	Normal power	Yes
		Normal power cable	Yes
		Emergency power	Yes
		Emergency power cable	Yes
		Emergency Power Unit AND normal power	No
		Emergency Power Unit AND normal power cable	No
		Emergency power cable AND normal power	No
		Emergency power cable AND normal power cable	No

The IE in Figure 10-16 is a fire originating in Zone j . Given that the fire occurs, the first pivotal event considers whether equipment within the zone is damaged by the fire. If the fire is insufficiently intense to cause equipment damage, there is no loss of the facility Control Computer. Even if the fire damages equipment, end state ESF can be averted if the continued operation of other facility equipment prevents Equation (10-9) from being satisfied. Note that, according to Table 10-1, if the fire ignites in Zone 3 and damages any equipment, end state ESF is inevitable. Relative to Figure 10-16. Facility Fire Event Tree., the conditional probability that sufficient equipment survives to avert loss of the Control Computer is zero.

Propagation between zones must be evaluated as part of a fire PRA. Relative to Figure 10-14. Facility Power Schematic., if the fire barriers have a high, certified rating (e.g., three hours), then it is very unlikely that a fire in one zone can propagate to another. Under this condition, equipment damage by fire is restricted to the zone in which the fire originates. If:

- $\Pr(D_j|F_j)$ symbolizes the conditional probability that equipment in Zone j is damaged, given a fire ignites in Zone j ; and
- $\Pr(ESF|D_j \cap F_j)$ is the conditional probability that end state, ESF, results, given that equipment in Zone j is damaged by a fire initiated in Zone j ;

then

$$\lambda(ESF) = \sum_{j=1}^3 \lambda_j \Pr(D_j|F_j) \Pr(ESF|D_j \cap F_j) \quad (10-11)$$

Techniques for quantifying λ_j and $\Pr(D_j|F_j)$ are described in conjunction with Equations (10-7) and (10-8). Given that certain equipment is damaged by the fire, $\Pr(ESF|D_j \cap F_j)$ can be quantified using Figure 10-15. Fault Tree for Loss of the Control Computer. and Equation (10-9). Beginning in Zone 1, if fire damages any equipment in that zone, event EPF is true. Combining this with Equation (10-9):

$$\Pr(ESF|D_1 \cap F_1) = \Pr(CF \cup LPF \cup NPF) \approx \Pr(CF) + \Pr(LPF) + \Pr(NPF) \quad (10-12)$$

if the rare event approximation is applicable. Similarly, for Fire Zone 2:

$$\Pr(ESF|D_2 \cap F_2) = \Pr(CF \cup LPF \cup EPF) \approx \Pr(CF) + \Pr(LPF) + \Pr(EPF) \quad (10-13)$$

while, for Zone 3

$$\Pr(ESF|D_3 \cap F_3) = 1 \quad (10-14)$$

Table 10-2 lists some illustrative values for λ_j and $\Pr(D_j|F_j)$.

Table 10-2. Illustrative Values for λ_j and $\Pr(D_j|F_j)$.

Zone	Statistic	λ_j (per year)	$\Pr(D_j F_j)$
1	5th percentile	1.1×10^{-7}	0.32
	Median	1.3×10^{-5}	0.62
	Mean	1.2×10^{-4}	0.62
	95th percentile	3.7×10^{-4}	0.90
2	5th percentile	1.2×10^{-7}	0.20
	Median	8.4×10^{-6}	0.55
	Mean	4.2×10^{-5}	0.87
	95th percentile	1.6×10^{-4}	1.00
3	5th percentile	5.3×10^{-7}	0.12
	Median	3.3×10^{-5}	0.45
	Mean	1.5×10^{-4}	0.80
	95th percentile	5.0×10^{-4}	0.48

10.7 A Launch Vehicle Ascent Abort Model

In Chapter 14 we will use a launch vehicle ascent abort scenario to illustrate the analysis process and how phenomenological modeling fits into it and is developed.

The top-level problem is to understand the efficacy of an emergency system, given that an ascent failure requiring abort has occurred. In the example, the emergency system is a launch abort system and we are interested in understanding how well this launch abort system performs under the most likely failure scenarios. The performance of the launch abort system, then, must be understood within the context of the conditions that are produced by these failures, *regardless of how these failures were initiated*. For launch systems, the conditions occurring after a failure during the ascent phase are typically limited to a few, fundamentally different classes of off-nominal situations. We are interested in understanding how the ascent failure physically manifests itself and whether it produces one or more of the following environments: a blast, a debris field, a fireball, or flight conditions that are outside the nominal flight conditions. A blast environment would produce overpressure conditions, which would potentially compromise the structural integrity of the crewed compartment. It could also contribute to extremely severe flight conditions for an abort system. A debris field would produce shrapnel or large fragments that could also compromise the structural integrity of the crewed compartment. A fireball could melt or burn parts of the system that are flammable. Although there may be other resulting failure conditions, these environments, and various combinations of them, cover the majority of resulting failure scenarios.

A time-dependent model is required because the ascent phase of a launch system has a short timeframe and the magnitude of the resulting failure environments and their impact on the system are tightly coupled to the current conditions (such as amount of propellant, vehicle velocity, altitude, mixture ratios, etc.). Failure environments generally develop rapidly, and the analysis of an abort system must be able to reflect these potentially severe changes. Assuming the failure environments are independent of each other, the characterization of each failure

environment may be modeled separately using first-principles physics models. The characterizations should include parameters that can be compared directly to the thresholds of a critical system (e.g., abort system or crew compartment). Any worst-case simplifying assumptions would lead to potentially conservative results. As a result, the methodology naturally identifies the specific areas requiring refined analyses and improved input data, thus providing a well-defined analysis path.

10.8 Summary

As discussed in this chapter, physical and phenomenological effects models are extensively needed and applied as key components of various types of PRAs. Because the nature of these models depends on the type of risk and application of interest, it would be impossible to cover all possible types from a completely generalized point of view. The discussion and examples presented in the preceding sections (and in Chapter 14), however, should provide the reader with the basic understanding necessary to develop the form of model that is appropriate for a specific PRA need.

10.9 References

- 10-1 J.B. Baeker, et al., "Launch Risk Analysis," *Journal of Spacecraft and Rockets*, Vol. 14, No. 12, December 1977, 733-738.
- 10-2 J.D. Collins, "Risk Analysis Methodologies Developed for the U.S. Department of Defense," *Reliability Engineering and System Safety*, Vol. 20 (1988), 87-115.
- 10-3 NASA-HDBK 8719.14, Handbook for Limiting Orbital Debris, July 30, 2008.
- 10-4 D.J. Kessler et al., "A Computer-Based Orbital Debris Environment Model for Spacecraft Design and Observations in Low-Earth Orbit," NASA JSC Technical Memorandum 104825.
- 10-5 P.D. Wilde, et al., "Off-Base Blast Propagation Risk Analysis for Titan IV Launches," JANNAF Joint Propulsion Conference, October 1997.

11. Probabilistic Structural Analysis

This chapter provides guidance for Probabilistic Structural Analysis within the context of PRA. Probabilistic structural analysis entails propagation of uncertainties through deterministic structural analysis models, which can range from simple analytical equations to complex finite element and computational models. Failure is modeled using a limit state that separates the safe and failed regions of the design space, as will be described. An analytical approach that compares applied stress to allowable strength is presented and illustrated for several structural analysis models. Probabilistic methods for solving general structural analysis problems are introduced.

In many cases, numerical approaches such as finite element analysis are required to predict the performance of the system. Practical guidance for solving probabilistic finite element analysis problems is provided. The fracture mechanics limit state is defined for many NASA applications including many Space Shuttle structural and engine components. Guidance for the solution of these types of probabilistic fracture problems is provided. Examples are also provided that illustrate the application of probabilistic structural analysis methods to the solution of realistic structural analysis problems.

11.1 Basic Concepts of Probabilistic Structural Analysis

Structural analysis consists of predicting the response of a structural component or system subjected to external loading. The response (e.g., displacement, stress, strain, vibratory frequency) is dependent on the geometry and material properties of the component/system as well as the values of external loads. Once computed, the structural response can be compared to established limit values (e.g., maximum deflection, yield stress) to determine the performance (i.e., probability of failure or reliability) of a system.

In a deterministic structural analysis, the response is predicted using single-valued descriptions (e.g., typical or conservative values) of the applied loads or stresses, component geometry, and material properties. This calculation generates a single value of the predicted response.

A probabilistic structural analysis is usually based on the same structural model used for the deterministic structural assessment. However, selected input parameters are represented as random variables with specified probability distributions rather than as single values. Probabilistic methods are used to propagate the model uncertainties (the random variables) through a performance model (the underlying structural model) to predict the range and associated likelihood of the response. The performance model may be simple (e.g., a closed-form stress intensity solution for a simple model geometry), or complex (e.g., nonlinear finite element analysis, fracture mechanics).

A schematic representation of probabilistic structural analysis (i.e., uncertainty propagation through a performance model) is shown in Figure 11-1.^a The results include a probabilistic description of the performance measure (reliability function) as well as probabilistic sensitivity factors that characterize the relative contribution of each random input parameter to the variability in the performance measure. This information can be used to guide decisions associated with PRA.

^a Figures in Chapter 11 are reprinted with the Permission Copyright © 2006 Southwest Research Institute®.

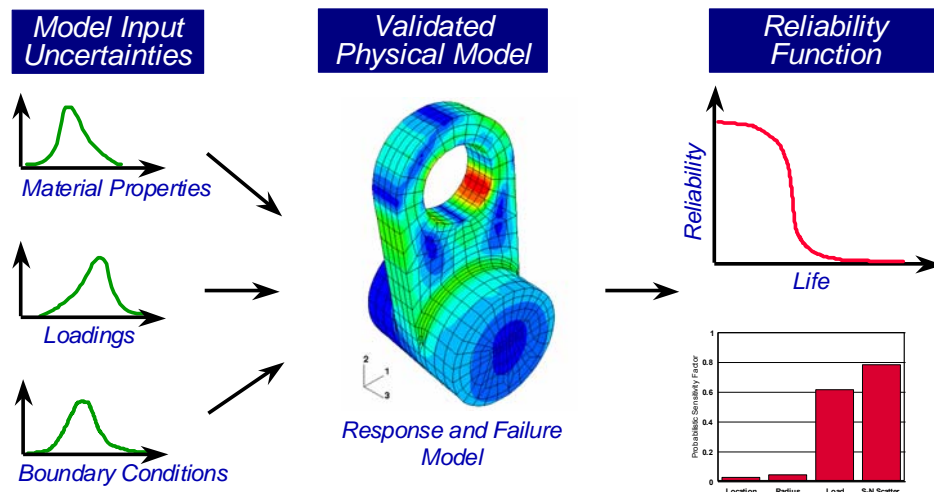


Figure 11-1. A Schematic Representation of Probabilistic Structural Analysis.

When a Probabilistic Structural Analysis is Needed

During the design stage, it is common practice to apply safety factors to deterministic load values to account for the uncertainty associated with the magnitude and duration of applied loads. The dimensions of simple structural components can be identified based on various combinations of load values that include safety factors. This approach is often applied to the assessment of structural systems where the relationship among member strength values is selected to provide a conservative result. However, most structures are modeled as complex systems consisting of many members, failure modes, and post-failure modes. It is difficult to assess the degree of conservatism in the relationships used among load and strength values and among the individual member strengths when such a conservative assessment is performed. This is particularly true when uncertainties are large, which they often are. Once a component or system has been placed in service, it may also experience loads that are different from the design values. The member strengths and structural configuration may change, or the intended use of the structure may change. Even if a component failure history can be established, it may not provide any insight into the reasons for failure or the conditions associated with future failures.

A probabilistic structural analysis can be used to address these problems. It directly accounts for the uncertainties in the relationships among load and resistance and among the individual member resistances. Unlike the factor of safety approach, it provides a quantitative estimate of the risk of failure that can be used for the assessment of both new and existing components and systems. It has been used for the solution of structural problems since the early 1950s, including a number of NASA problems in recent years [11-1 through 11-3].

11.2 Probabilistic Structural Response Modeling

Probabilistic Structural Response modeling involves the basic steps of formulating a limit state function and assigning uncertainty distributions to the variables involved.

11.2.1 Limit State Formulation

In mathematical terms, a model can be represented as

$$Z(\mathbf{X}) = Z(X_1, X_2, X_3, \dots, X_n) \quad (11-1)$$

where $Z(X)$ is a computed response from the model and X is the vector of n random variables affecting the response. In a probabilistic analysis, the objective is to compute the probability of failure. To proceed, the concept of performance must be defined. From Equation 11-1, the performance or limit state function can be defined as

$$g = Z(\mathbf{X}) - z \quad (11-2)$$

or

$$g = z - Z(\mathbf{X}) \quad (11-3)$$

where z is a particular value of $Z(X)$. The condition $g = 0$ defines the limit state, which separates the variable space into failure $g \leq 0$ and non-failure $g > 0$ regions. The alternative forms of writing the limit state function shown in Equations (11-2) and (11-3) provide the means to denote failure as $g \leq 0$ for any situation.

For example, if the model relates the tip deflection of a cantilevered beam to the uncertain load (W), length (L), stiffness (E), depth (b) and width (a), the response function can be written as:

$$Z(\mathbf{X}) = Z(W, L, E, b, a) \quad (11-4)$$

where Z is an analytical or numerical model and $Z(X)$ is the tip deflection, which is also a random variable. If performance is defined as the tip deflection meeting a certain criterion, such as not exceeding 2 cm, then the performance function, g , must be written such that failure is denoted when $g \leq 0$ or $g = 2 - Z(X)$.

The probability of failure, p_f , can be defined as

$$p_f = \Pr[g \leq 0] = \Pr[z - Z(\mathbf{X}) \leq 0] = \Pr[2 - Z(\mathbf{X}) \leq 0] \quad (11-5)$$

where p_f is the probability that 2 cm minus the computed deflection will be less than or equal to zero.

Computing the probability given in Equation (11-5) generally requires a numerical approach. To visualize the calculation it is useful to consider two random variables at a time. If two random variables are plotted on different axes, the resulting combined (or "joint") probability density function (JPDF) can be visualized as shown in Figure 11-2. The $g = 0$ curve (generally not a line) lies in the X_1 - X_2 plane. The intersection of $g = 0$ with the JPDF is shown in the figure with the $g \leq 0$ portion of the JPDF removed for clarity. The probability of failure p_f is the volume under the JPDF in the $g \leq 0$ region, or the portion of the JPDF removed in the figure. The volume under the JPDF in the $g > 0$ region in the figure (i.e., the volume under the remaining part of the JPDF) corresponds to the reliability, or $1 - p_f$. It can be observed in Figure 11-2 that the limit state defines the locations of the failure and safe regions.

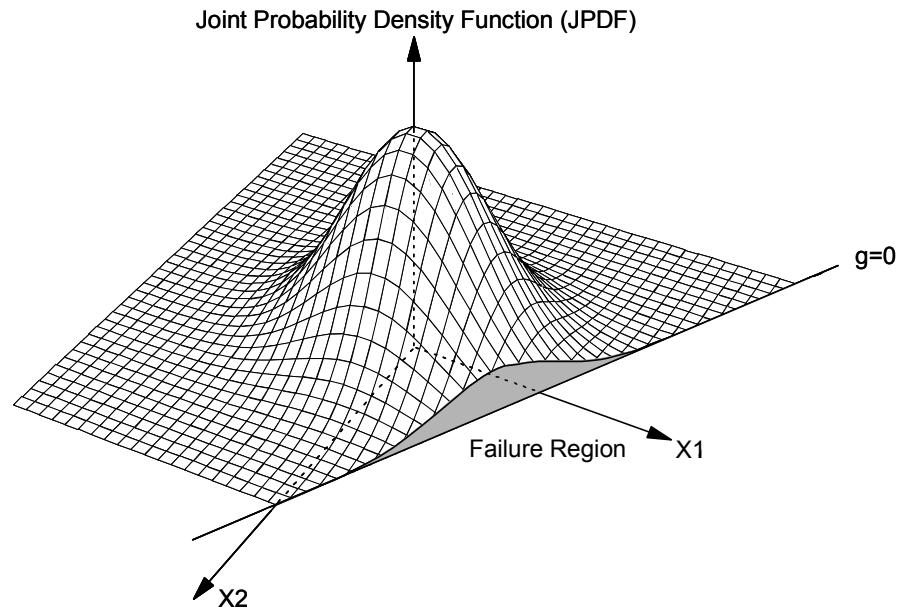


Figure 11-2. Joint Probability Density Function for Two Random Variables showing the Failure Region.

11.2.2 Assigning Uncertainty Models to Random Variables

Uncertainty models must be assigned to the structural analysis model variables that are random in order to perform a probabilistic analysis based on the defined failure condition. Uncertainties can be classified as inherent uncertainties (aleatory uncertainty) or model error (epistemic uncertainty). Inherent uncertainties are irreducible and describe the inherent variations in the system. Examples of inherent uncertainties include variations in system geometry or material properties, loading environment, and assembly procedures. Inherent uncertainties are typically modeled using probability density functions (such as Weibull or lognormal) or by a random process or field, if the statistics vary temporally or spatially, respectively.

Model error describes deficiencies that result from a lack of complete information about the system being modeled. This type of uncertainty is reducible. For example, consider a situation in which limited data are available to estimate the statistical distribution of a geometric property of a population of parts. Even with a large sample size, the statistics and distribution shape will still be in error. The error in the statistics and distribution shape decrease as additional data become available.

One common misconception of uncertainty modeling is that extensive experimental data are required to characterize the uncertainties of all of the variables that contribute to the failure condition. However, in many cases, only a few important variables drive the reliability of the system. A probabilistic analysis can identify the important variables and assist in allocating resources to collect experimental data only for these important variables.

11.3 Stress Versus Strength Modeling

Stress versus strength modeling is a particular form of structural response modeling and is widely used. Therefore a separate section is devoted to it. The classical structural reliability problem considers the potential for stress (S) exceeding strength (R). Strength is the capacity of the system, sometimes called resistance. Stress is the loading on the system, sometimes called load. The system performs adequately as long as the stress is less than the strength.

The system fails when the stress is greater than the strength. Therefore, the problem in probabilistic structural analysis is to compute the probability that the stress is greater than the strength:

$$p_f = P[S > R] \quad (11-6)$$

What makes this classical model probabilistic is the fact that R and S are not deterministic quantities but are instead random variables. To clarify, a deterministic variable is defined by a single value, whereas a random variable is typically defined by a probability density function characterized by a mean value, a standard deviation and a distribution type.

The limit state function (i.e., performance, g, or g-function) divides the design (or variable) space into failure and safe regions. It is defined such that the probability that $g < 0$ is also the probability of failure. For example, g can be defined such that g is less than zero when stress exceeds strength. In equation form,

$$g = R - S \quad (11-7)$$

11.3.1 Normal Distributions

Analytical solutions for the probability of failure are available for specific forms of Equation (11-7). If the function is linear and composed of independent normal random variables then an analytical solution is available. Let Z be the sum of normally distributed random variables

$$Z = a_o + a_1 X_1 + a_2 X_2 + \dots + a_n X_n \quad (11-8)$$

where a_i is a constant and X_i is normally distributed with mean of μ_i and standard deviation of σ_i . Z is then normal with mean of μ_z and standard deviation of σ_z where:

$$\mu_z = a_o + \sum_{i=1}^n a_i \mu_i \quad (11-9)$$

$$\sigma_z = \sqrt{\sum_{i=1}^n (a_i \sigma_i)^2} \quad (11-10)$$

Cumulative probabilities can be calculated using standard approaches for a normal distribution once the mean and standard deviation are determined for the new random variable Z. There is no closed form solution for the cumulative distribution function (CDF) of a normal distribution but values for a standard normal CDF have been extensively tabulated and are available using most engineering calculators and spreadsheets. (A standard normal probability density function is a normal distribution with a mean of zero and standard deviation of 1.) Z can be formulated as standard normal distribution using

$$U = \frac{Z - \mu_z}{\sigma_z} \quad (11-11)$$

where U is now a standard normal variable whose cumulative distribution is defined as

$$F_U(u) = \Phi(u) \quad (11-12)$$

The probability that Z is less than a specific value can be then computed by converting the normal distribution to a standard normal

$$P[Z < z_o] = P\left[\frac{Z - \mu_z}{\sigma_z} \leq \frac{z_o - \mu_z}{\sigma_z}\right] \quad (11-13)$$

Then the value for the standard normal variable for z_o is

$$u = \frac{z_o - \mu_z}{\sigma_z} \quad (11-14)$$

and

$$P[Z < z_o] = \Phi(u) = \Phi\left(\frac{z_o - \mu_z}{\sigma_z}\right) \quad (11-15)$$

Then, if stress and strength are independent normally distributed random variables, the probability of failure can be quantified in closed form by identifying the main descriptors (i.e., mean and standard deviation) of g and computing the probability that g is less than zero. The mean and standard deviation of g are computed using Equations (11-9) and (11-10):

$$\mu_g = \mu_R - \mu_S \quad (11-16)$$

$$\sigma_g = \sqrt{\sigma_R^2 + \sigma_S^2} \quad (11-17)$$

Next, g is converted into a standard normal distribution using Equation (11-15) where $Z=g$ and $z_o=0$. The standard normal variable is then

$$u = -\frac{\mu_g}{\sigma_g} \quad (11-18)$$

and the probability of failure is finally computed as

$$P[g < 0] = \Phi\left(-\frac{\mu_g}{\sigma_g}\right) \quad (11-19)$$

11.3.2 Lognormal Distributions

Another often applied analytical solution is when the model can be expressed as a multiplicative function and the random variables are all lognormally distributed. Assume that $H(X)$ is a multiplicative function of the model parameters

$$H(X) = B \prod_{i=1}^n X_i^{a_i} \quad (11-20)$$

where B and all a_i are constants. Let $Z=\ln(H)$

$$Z = \ln(H(X)) = \ln B + \sum_{i=1}^n a_i \ln X_i \quad (11-21)$$

If all X_i have lognormal distributions, $\ln X_i$ is normal and the formulation becomes identical to the normal formulation. By definition of the lognormal distribution

$$\mu_Z = \tilde{Z} = \ln \tilde{H} = \ln \left[B \prod_{i=1}^n \tilde{X}_i^{a_i} \right] \quad (11-22)$$

$$\sigma_Z^2 = \ln \left[\prod_{i=1}^n (1 + C_i^2)^{a_i^2} \right] \quad (11-23)$$

Where the \sim indicates the median value and C is the coefficient of variation computed as

$$C_i = \frac{\sigma_{X_i}}{\mu_{X_i}} \quad (11-24)$$

and

$$\tilde{X}_i = \frac{\mu_{X_i}}{\sqrt{1 + C_i^2}} \quad (11-25)$$

The distribution of Z is normal and CDF can be computed using Equation (11-15).

If stress and strength are independent lognormally distributed random variables, the limit state function can be expressed as

$$g = R - S = R/S - 1 = 0 \quad (11-26)$$

The probability of failure can then be formulated as

$$p_f = P[g < 0] = P[R - S < 0] = P\left[\frac{R}{S} - 1 < 0\right] = P\left[\frac{R}{S} < 1\right] = P\left[\ln\left(\frac{R}{S}\right) < 0\right] \quad (11-27)$$

The probability of failure is consequently

$$P[g < 0] = \Phi\left(-\frac{\mu_Z}{\sigma_Z}\right) \quad (11-28)$$

For the simpler two variable R-S problem, the main descriptors of g for lognormal variables are given by

$$\mu_g = \ln\left(\frac{\tilde{R}}{\tilde{S}}\right) \quad (11-29)$$

$$\sigma_g = \sqrt{\ln \left[(1 + C_R^2) (1 + C_S^2) \right]} \quad (11-30)$$

where

$$C_R = \sigma_R / \mu_R \quad (11-31)$$

$$C_S = \sigma_S / \mu_S \quad (11-32)$$

$$\tilde{R} = \mu_R / \sqrt{1 + C_R^2} \quad (11-33)$$

$$\tilde{S} = \mu_S / \sqrt{1 + C_S^2} \quad (11-34)$$

The standard normal variable is then

$$u = -\frac{\mu_g}{\sigma_g} \quad (11-35)$$

and the probability of failure is finally computed as

$$p_f = \Phi(u) \quad (11-36)$$

Even though the stress versus strength model given by Equation (11-6) is simple compared to other structural models, it has important applications. The above normal and lognormal formulations are often used for first stage evaluations where information is not detailed enough or precise enough to select other distributions. Where applicable, formulations for other distributions can be handled using the general relationship given by Equation (11-6). In the most general case, Monte Carlo simulations can be carried out.

11.4 Monte Carlo Simulation and Most Probable Locus Approaches

Numerical approaches are often required when the form of the random variables and structural model does not lend itself to an analytical solution described in the previous section. Monte Carlo simulation, a random sampling technique, is a well-known technique for computing the probability of failure [11-4,11-5] for a general reliability problem. In Monte Carlo simulation, random samples are obtained from the input distributions associated with each random variable. The values for the input variables are applied to a structural model to obtain the response. The number of samples is dependent on the probability being computed, with low probabilities requiring a large number of samples. An example of Monte Carlo simulation is shown in Figure 11-3. Monte Carlo simulation can be used to estimate probability of failure or for computing the probability density function of the response.

The advantages of Monte Carlo simulation are that it is simple to implement, will work even if the model response is not smooth and continuous, and the efficiency is not a function of the number of random variables. However, a major drawback of Monte Carlo simulation is the relatively large number of samples required to compute small probabilities. The sampling error and associated confidence bounds for a Monte Carlo estimate, viewed as an average from the Monte Carlo samples, can be determined from the estimate of the standard deviation and associated confidence bounds using the individual values for each of the Monte Carlo samples.

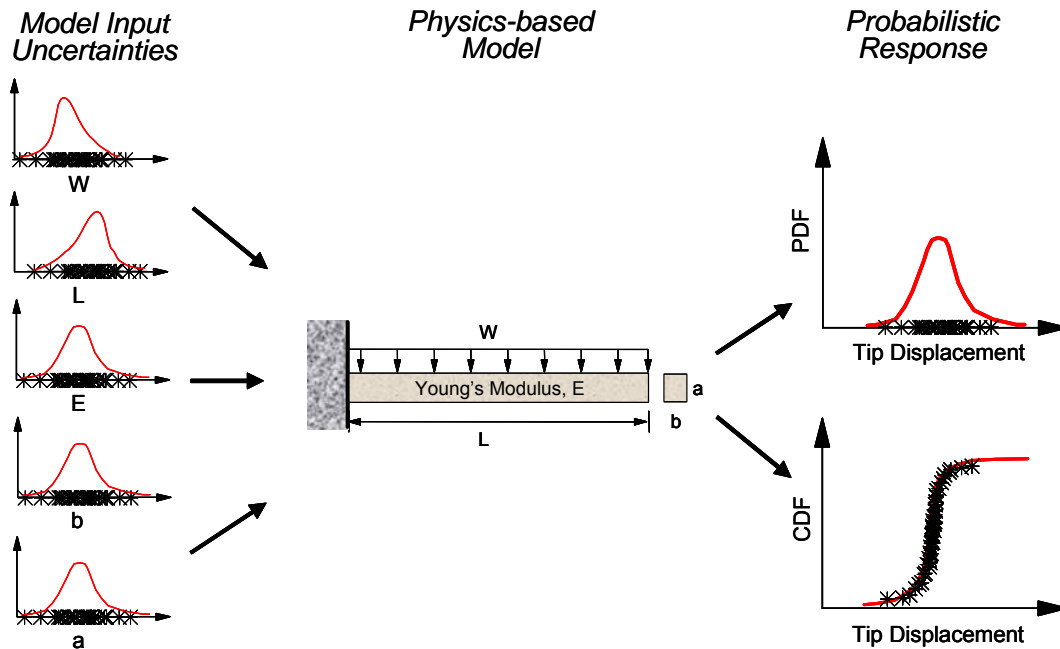


Figure 11-3. Probabilistic Structural Analysis using Monte Carlo Simulation.

In general, to accurately estimate a failure probability, at least 10 failures are required from the number of Monte Carlo samples used. Thus, to accurately compute a probability of 10^{-3} would require approximately 10,000 samples. Probability of failure values much lower than this are not uncommon; in commercial aviation gas turbine engine components, the failure probability must be at or below 10^{-9} failure events per flight.

The primary reason that Monte Carlo simulation is inefficient is that the random samples are clustered near the mean of both input and output values. Recognizing this led to the development of more efficient random sampling techniques such as Latin Hypercube Simulation (LHS) [11-6] and Adaptive Importance Sampling (AIS) [11-7]. The objective of LHS is to distribute the samples so that the sampling density is more evenly spread over the probability distribution of each input random variable. This can produce more failure samples than Monte Carlo for the same number of simulations. In the AIS approach, the initial sampling region is defined by an approximate $g(X) = 0$ function developed at the failure region. The $g(X) < 0$ area is gradually increased by changing the sampling boundary until the sampling region covers the failure region sufficiently.

The inefficiency of random sampling and the computationally-intensive nature of finite element analysis motivated the development of efficient and accurate probabilistic analysis methods that were based not on random sampling, but on the concept of a most probable point (MPP). The MPP is a point on the $g = 0$ limit state that corresponds to the maximum value of the joint probability density function (JPDF) associated with the input random variables. This point is also the minimum distance from the origin to the limit state in a transformed probability space (u -space) as shown in Figure 11-4. There are several unique characteristics about this transformed space. The JPDF is rotationally symmetric about the origin and the origin corresponds to the mean response. Since the MPP is a minimum distance, optimization algorithms can be used to locate the minimum distance. The number of standard deviations from the mean to the MPP is referred to as the reliability index. The MPP is only a single point on the limit state and most MPP methods use an approximation to the limit state, which introduces some error into the probability of failure results. In many cases, this error is small since the majority of the probability is concentrated near the MPP where the approximate limit

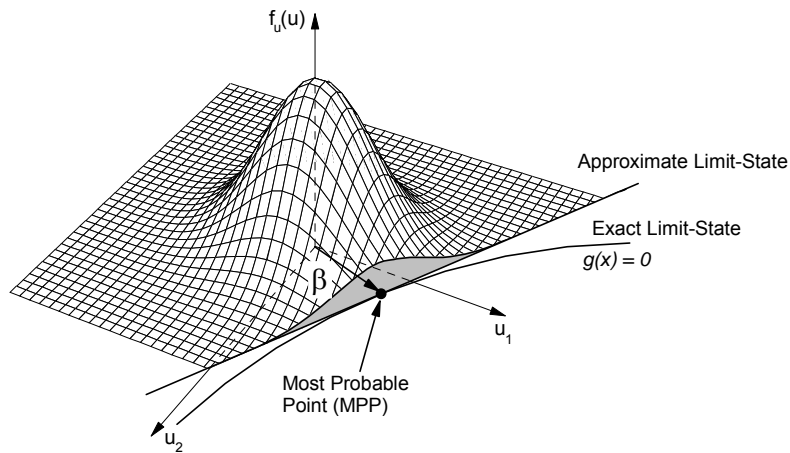


Figure 11-4. Joint Probability Density Function (JPDF), Exact and Approximate Limit-State, and Most Probable Point (MPP) for Two Random Variables in Transformed (u) Space.

state function is usually accurate. Once the MPP is located, different probabilistic methods are used to determine the probability in the failure region.

Although many variations have been proposed, the best-known and most widely-used MPP-based methods include the first-order reliability method (FORM) [11-8 through 11-10] and the second-order reliability method (SORM) [11-11 through 11-13]. These methods describe how the probability integration is performed after locating the MPP. The FORM probability solution is based on the linearization of the g -function at the most probable point (MPP) in the u -space. The concept is shown in Figure 11-5 for two random variables. The linear approximation to the limit state allows the probability to be computed as

$$p_f = \Phi(-\beta) \quad (11-37)$$

where β is the distance from the origin to the MPP. Φ is the standard normal cumulative distribution function (CDF), which relates cumulative probabilities to standard deviations in a standard normal probability distribution. The figure shows that the possibly nonlinear limit state is replaced with a linear approximation for the probability integration. The error in the probability integration will depend on the degree of nonlinearity. FORM can either over- or under-predict the probability depending on whether the surface is concave or convex.

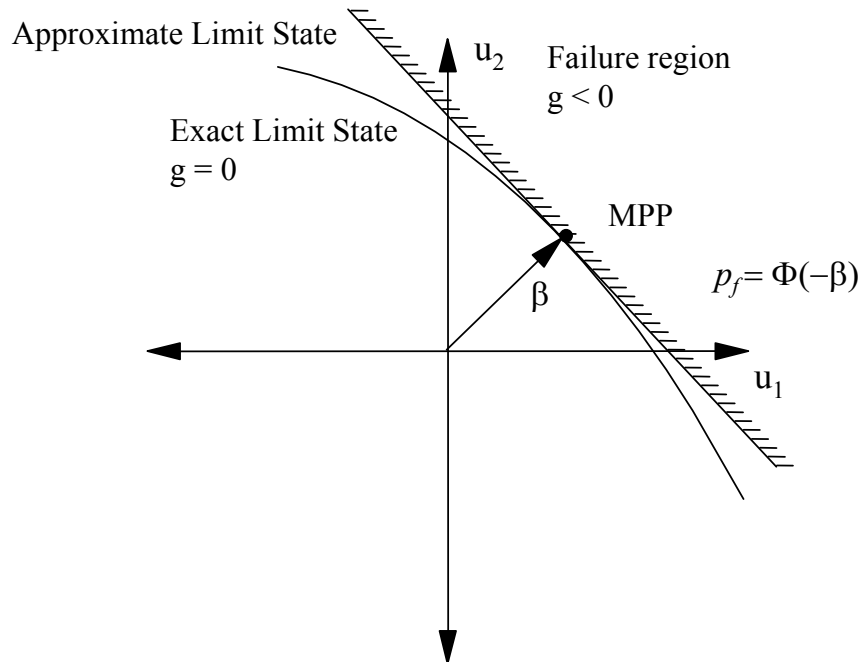


Figure 11-5. Concepts of 1st Order Reliability Method (FORM) for Probability Approximations.

SORM approximates the limit state as a parabolic surface and the probability is estimated using β and the curvatures at the MPP. Approximate and exact solutions to the probability associated with a parabolic surface in the transformed probability space are available [11-11 through 11-13]. The SORM concept is shown in Figure 11-6. The computed probability may be in error if the exact limit state is not approximated well by the parabolic surface, but is usually an improvement over FORM at the additional cost of computing the curvatures.

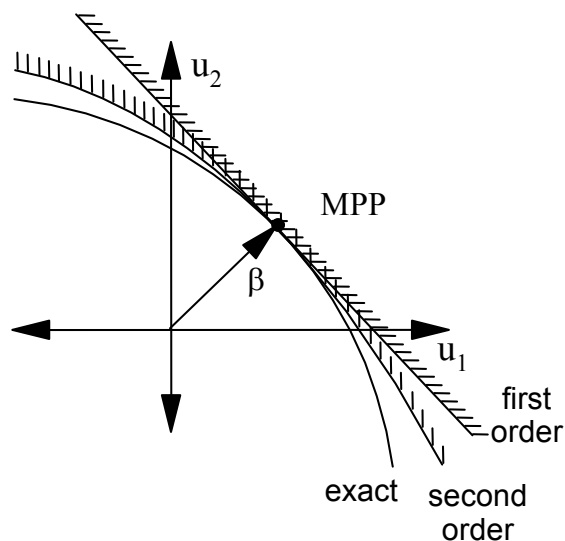


Figure 11-6. Concepts of 2nd Order Reliability Method (SORM) for Probability Approximations.

Advanced mean value (AMV) methods are most suitable for complicated but relatively well-behaved response functions requiring computationally intensive calculations [11-14]. The AMV

methods replace the exact function with a fast running low order polynomial approximation for locating the MPP. The method typically uses successive linear approximations to the limit state until the MPP is located within a user specified tolerance. Once the MPP is found, the probability is usually estimated using the FORM or SORM approach. AMV was developed to efficiently locate the MPP for solving problems where the solution of each response function is computationally intensive. In practice, when a performance function is implicitly defined and is highly nonlinear, it is difficult to assess the error introduced by the polynomial approximation. Therefore, the approximate solutions should be checked using other more accurate methods such as an efficient importance sampling method.

A summary of the advantages and disadvantages associated with common probabilistic methods is provided in Table 11-1. Monte Carlo simulation is the preferred approach when the g -function is analytical because the result approaches the exact solution as the number of samples increases, it can be used with any type of response function, and it quantifies the error associated with the result. The MPP-based methods are approximate, but generally provide sufficiently accurate predictions much more efficiently than many sampling-based methods. Further information regarding the errors and uncertainties associated with these methods is provided in References 11-15 and 11-16. The probabilistic methods described in this section have been implemented in several commercially available software packages as described in Reference [11-17].

Table 11-1. Advantages and Disadvantages of Several Common Probabilistic Methods.

Method	Type	Advantages	Disadvantages	Comments
Monte Carlo	Sampling	Works for any model; Simple to implement; Multiple limit states	Large number of samples required to estimate small probabilities with high accuracy and confidence	Benchmark to verify other methods; Method of choice when computational models allow
Latin Hypercube Simulation	Sampling	Works for any model; Samples distributed more evenly than Monte Carlo for small samples sizes; Multiple limit states	Large number of samples required to estimate small probabilities	Typically used for computing the mean, standard deviation, and distribution of the response
FORM	MPP	Relatively efficient for small probabilities of failure; Exact for linear functions composed of normal random variables	Locating MPP may be difficult for nonlinear response functions; Error in probability prediction for nonlinear limit state; Single limit state only	Accuracy of linear approximation improves for small probability of failure
SORM	MPP	Relatively efficient for small probabilities of failure; Exact for parabolic surfaces	Locating MPP may be difficult for nonlinear response functions; Error in probability prediction for non-parabolic limit state; Single limit state only	Accuracy of parabolic approximation improves for small probability of failure
AMV	MPP	More efficient than FORM since MPP search is performed on a fast running approximate function; Exact for linear functions composed of normal random variables	Locating MPP may be difficult for nonlinear response functions; Error in probability prediction for nonlinear limit state (based on FORM); Single limit state only	Accuracy of linear approximation improves for small probability of failure
AIS	Hybrid	More efficient than standard Monte Carlo since samples are focused in the failure region; Number of samples independent of the probability of failure; Multiple limit states	Must have information of the failure region (MPP) so this method has the same limitations as FORM/SORM for locating the MPP	Provides an efficient error check for MPP methods (FORM/SORM/AMV)

11.5 Probabilistic Finite Element Approaches

Finite element analysis (FEA) has become a popular tool for simulating the behavior and response of complex structures and mechanical systems. Fundamentally, FEA provides a numerical approximation of the response of a structure to applied loadings. Reasonably simple finite element models are used to predict response and behavior when analytical solutions are not possible, for example, when the geometry is nontrivial or nonlinear materials are involved. FEA can also be performed for complete processes and systems with multiple interacting physics such as solid mechanics, dynamics, hydrodynamics, heat conduction, fluid flow, transport, chemistry, and acoustics. For deterministic finite element analysis, the response is predicted using single-valued descriptions for all of the input variables, and the computation yields a single response value.

A probabilistic finite element analysis is based on a deterministic finite element model, except that some of the input parameters are represented as random variables with specified probability distributions rather than as single values. Quantifying the effect of uncertainties in a finite element model provides the analyst with an estimate of the true margin of safety for a particular design and allows alternative designs to be assessed on the basis of quantified reliability. Knowledge of the effect of uncertainties can also lead the analyst to drastically different conclusions regarding which input parameters are most influential. It is for this reason that probabilistic FEA is rapidly gaining widespread acceptance in design.

11.5.1 When Probabilistic Finite Element Analysis is Needed

Deterministic FEA is needed when the structural response cannot be accurately represented using analytical or empirical models. A probabilistic FEA is required when the input variables exhibit significant uncertainty. A probabilistic FEA can be used to quantify the reliability of a component or system, and to identify the design or operating parameters that have the most influence on the performance of the component or system.

11.5.2 Mapping Random Variables to Finite Element Input

When performing probabilistic FEA, a specific (e.g., sample) realization of the random variables must be reflected in the FE input. Random variables that affect a single quantity in the FE input are called scalar variables and random variables that affect multiple quantities are called field variables. Typical examples of scalar random variables include Young's modulus or a concentrated point load. Examples of field random variables are a pressure field acting on a set of elements or a geometric parameter that affects multiple node locations, e.g., radius of a hole. Scalar random variables are directly mapped from the random variable value to the analysis program input. Field variables require a functional relationship between the random variable and the analysis program input. Because different realizations of these field random variables are required, a general approach can be used to relate the finite element input to a change in the random variable value. For example, if the random variable is the radius of a hole, changes to a set of nodal coordinate values will be required each time the radius is changed. This can be accomplished by defining a scaling vector that relates how the FE coordinates change for a given change in the random variable, i.e., radius in this example.

Figure 11-7 shows an example of a field random variable, where a change in the random variable h produces a change in the finite element mesh. This approach can be used for any type of field random variable (e.g., pressure and temperature distributions). If the scaling vector does not change during the analysis, then the relationship between the random variable and the finite element mesh is linear. Nonlinear relationships can also be defined if warranted.

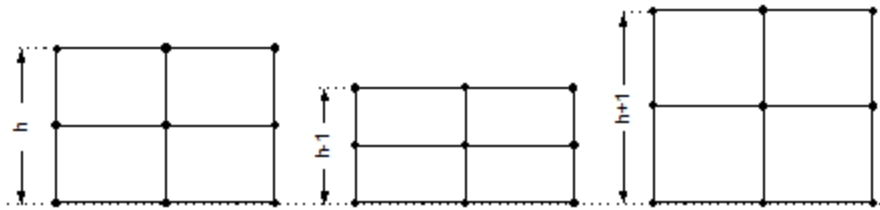


Figure 11-7. A Random Dimension h and its Effects on the FE Mesh.

In most situations, the analyst or engineer will have access to a commercial FEA code. However, in far fewer situations will the commercial FEA code have any built-in capability for performing probabilistic analysis. Fortunately, there are now a number of probabilistic analysis codes [11-17] that provide interfaces to commercial FEA codes. In some cases, these probabilistic analysis codes also have capabilities for interfacing with other software available to the analyst, e.g., internally developed software. A probabilistic analysis algorithm can be incorporated within a FEA code, but it is far more common to “drive” or control the FEA code from the probabilistic analysis code. In this approach, several practical issues arise, and because of their importance in the overall modeling process, are worthy of briefly mentioning here. These include defining performance measures from finite element outputs, mapping random variables into finite element input, and results extraction

A typical FEA produces voluminous amounts of computed results, especially when the results are time varying. In the case of stresses, for example, the probability that the stress exceeds yield at one point in the mesh and at one point in time will seldom be a meaningful performance measure. A more realistic measure would be the condition that the stresses across a net section or along a remaining ligament exceeded yield at any point in time. Therefore, the probabilistic FEA must have the capability of reducing multiple FEA responses into a single meaningful performance measure. Since the types of performance measures are problem dependent, capabilities that are user-definable will typically be needed.

In many cases, in addition to reducing finite element output data, a failure model is required. For example, if the condition of interest is failure by fatigue crack growth, stresses from the FEA must be extracted from the location of interest in the model, reduced or processed, and then passed to a fatigue crack growth model to compute remaining life. The fatigue crack growth model could be analytical or numerical and have random variable inputs associated with it as well. Thus, the probabilistic FEA must have the capability for linking multiple numerical and analytical models such that the analyst can include and describe all important failure modes. For efficiency reasons, most FEA programs store results from the analysis in some type of binary database file. Consequently, a probabilistic code must be able to open and extract specified results quantities from this binary database file. Since each FEA code defines the format of the binary results file, a different results extraction utility must be defined for each FEA code to which the probabilistic software is interfaced.

11.6 Probabilistic Fracture Mechanics

Fracture control is a standard NASA practice for critical structural components. Standard guidelines and templates have been developed to define when fracture mechanics (FM) calculations are required and how they are to be performed [11-18 through 11-20]. Those guidelines and templates address a deterministic FM calculation where inputs to the calculation are either average or conservative values. The “safe life” limit for NASA applications is defined by performing the fatigue crack growth (FCG) life computation based on these nominal inputs and then dividing the life by a factor of four to account for material variability.

The NASGRO® computer program is the standard NASA tool for FM analysis of metallic structures. Fatigue crack growth rate calculations in NASGRO use the so-called NASGRO equation,

$$\frac{da}{dN} = C \left[\left(\frac{1-f}{1-R} \right) \Delta K \right]^n \frac{\left(1 - \frac{\Delta K_{th}}{\Delta K} \right)^p}{\left(1 - \frac{K_{max}}{K_c} \right)^q} \quad (11-38)$$

Here da/dN is the average crack growth rate per cycle; $\Delta K = K_{max} - K_{min}$ is the stress intensity factor (SIF) range (the driving force for FCG), where K_{min} and K_{max} are the minimum and maximum values of the SIF in the cycle; $R = K_{min}/K_{max}$; f is a crack opening function that describes the effect of stress ratio and constraint on growth rate; and C , n , ΔK_{th} , K_c , p , and q are FCG material properties. The SIF is generally calculated according to the relationship

$$K = [S_0 F_0 + S_1 F_1 + S_2 F_2 + S_3 F_3 + S_4 F_4] \sqrt{\pi a} \quad (11-39)$$

where a is the crack size, the S_i terms are the various load or stress quantities (tension/compression, bending, pin loading, etc.), and the F_i terms are geometric correction factors which are themselves functions of the size and shape of the crack and the cracked body. Advanced weight function SIF solutions perform a more sophisticated integration of crack-line stress distributions and geometric factors.

Fatigue crack growth life is calculated by re-arranging and integrating Equation (11-38) between initial and final crack sizes to determine the number of cycles required for the growth to occur,

$$N = \int_{a_0}^{a_f} \frac{da}{g(K_{min}, K_{max}, f, \text{material properties})} \quad (11-40)$$

The final crack size often corresponds to failure by complete fracture (when K_{max} exceeds the fracture toughness K_c). The NASGRO software provides a convenient graphical user interface to facilitate the definition of the crack and component geometry and the applied loads or stresses for a library of SIF solutions, as well as the selection of material properties from a large library database (or the specification of user-defined properties). NASGRO then performs the integration of the crack growth equation and the calculation of the corresponding SIF solutions.

11.6.1 Differences of Probabilistic Fracture Mechanics

In deterministic FM, these calculations (which are based directly on the physics of fracture) employ single-valued descriptions (either typical or conservative choices) of the applied loads or stresses, the crack and component geometry (including the initial crack size), and the material resistance to crack growth. The calculation generates a single value of the predicted FCG life, or predicts whether or not fracture will occur under the specified single-value input conditions.

The foundation of probabilistic fracture mechanics (PFM) is the underlying deterministic FM model employing the physics-based equations. However, selected input parameters are represented as random variables with a specified probability distribution rather than a single value. Appropriate probabilistic methods are used to propagate the model uncertainties (random variables) through a performance model (physics-based equations) to predict the probabilistic response and sensitivity factors of the system performance measures. Typical performance measures in PFM include the number of fatigue cycles to cause failure or the

cumulative probability of failure at some point in the component history. The results include a probabilistic description of the performance measure as well as probabilistic sensitivity factors that characterize the relative contribution of each random input parameter to the variability in the performance measure. The probability density and sensitivity information generated by the PFM analysis provides support for subsequent decision analysis.

11.6.2 When Probabilistic Fracture Mechanics is Needed

The well-established paradigm for deterministic fracture control may be inadequate in certain situations. When significant inputs to the (deterministic) fracture calculation exhibit significant scatter or uncertainty, it is common practice to select conservative (perhaps “worst-case”) bounding values to perform the analysis, and this often gives acceptable results. However, in some cases the resulting calculation of safe life may give an overly conservative (and highly unlikely) result that cannot be sustained due to program cost or schedule factors (for example, when the answer requires frequent, expensive replacement or inspection of the component, but the actual risk of fracture is very low). This outcome is especially likely when more than one independent input variable exhibits large variability and is conservatively bounded, because the probability of two or more independent variables simultaneously exhibiting extreme values is often a negligibly small value. For example, the probability that nondestructive evaluation (NDE) will miss a large crack on a component fabricated from a material with unusually low resistance to FCG, and that same component also experiencing abnormally high load levels, is likely to be very low.

Furthermore, the existing deterministic fracture control paradigm does not produce any quantitative measure of the risk of fracture. In some cases, it may be essential to understand and manage the total risk of system failure, since system failure may have severe consequences for mission success and even loss of human life. Component fracture is sometimes a significant contributor to the total system risk. However, without some probabilistic assessment of the fracture problem, it is not possible to provide a meaningful quantitative input to the system risk summation from the component fracture perspective. In some cases where significant component failure history has been established, a failure rate can be defined empirically. However, this is not possible for new designs, for components with such low failure rates that few or no failures have been observed, or in systems (such as those employed by NASA) where few replicates exist.

A probabilistic fracture mechanics (PFM) analysis can address these shortcomings. PFM is a mature technology that has been applied in numerous industries since the 1970s. Although standard NASA damage tolerance analyses are deterministic, the most recent NASA fracture control requirements [11-21] permit use of probabilistic damage tolerance analysis with the specific approval of the responsible fracture control authority. PFM has been applied to solve several specific NASA applications problems (for example, [11-22 through 11-24]).

11.6.3 Probabilistic Characterization of Input Variables

The primary challenge in a PFM assessment is to identify and characterize the significant random input variables. These variables generally fall into four major classes: describing the initial crack size and state, in-service inspections (if any), applied loads or stresses, and life scatter considerations arising from material variability and life modeling error.

Initial Crack. NASA fracture control assessments typically assume that a single initial crack with a standard aspect ratio is always present in the structure at a critically stressed location. The assumed size of the initial crack is often defined by a nondestructive evaluation (NDE) probability-of-detection (POD) curve, the cumulative distribution of crack detection probability as a function of increasing crack size. The probability distribution of initial crack sizes is taken to be the inverse of the median POD curve, interpreting the POD in terms of the probability of non-

detection to estimate the size of the largest crack remaining in the component following inspection. Alternatively, the initial crack size distribution may also be defined by initial material or manufacturing defect sizes, by the PFM analysis of a structural proof test, or by an “equivalent initial flaw size” (EIFS) distribution that has been back-calculated from final crack sizes associated with known fatigue lifetimes (such as field cracking data). A more comprehensive PFM assessment may also consider the probability of occurrence of the crack, the possibility that more than one crack is present, and the possibility of the crack occurring at other locations in the structure, as well as variability in the initial shape of the crack.

In-Service Inspection(s). In some applications, an in-service NDE inspection is performed in an attempt to detect and remove cracked components from the fielded population. The inspection will alter the distribution of probable crack sizes remaining in the population following the inspection. This process can be treated in at least two different ways. First, the POD curve corresponding to the in-service inspection can be used to define a new distribution of initial crack sizes (as described earlier). An alternative, more rigorous treatment of in-service inspection first determines the distribution of fatigue crack sizes just prior to the inspection by conducting a full PFM assessment of this earlier phase of life. The distribution of prior crack sizes is then altered by applying the in-service POD curve, removing a selection of the larger flaws in the population. The altered distribution then becomes the starting point for a PFM assessment of the next increment of fatigue cycles. In-service inspection also introduces two other potential sources of uncertainty: (1) the time at which the in-service inspection is conducted, and (2) the probability that the in-service inspection is missed entirely.

Loads. Variability and uncertainty in the applied loads or stresses is often one of the most significant considerations in a PFM analysis. Because of the substantial variety of load types that can occur in NASA hardware, it is more difficult to construct a simple conceptual framework to define the proper probabilistic treatment of loads.

If the loading history is best characterized as constant amplitude (the load amplitudes do not vary substantially from cycle to cycle), then the key random variable is the constant load amplitude. If the loading amplitude varies substantially from cycle to cycle, then the characterization can be considerably more complex. The relative distribution of different load amplitudes and different load ratios within the spectrum must be considered. For complex load spectra, the appropriate statistical representation of the spectral content can itself be a complex issue.

Variability in the number of fatigue cycles may be significant. The fatigue loading may have a deterministic occurrence rate (e.g., a start-up/shut-down cycle that occurs once per mission). However, the loading may be driven by uncertain events (e.g., structural vibration during temporary resonance conditions). Both the frequencies of the resulting fatigue cycles and the total time that the structure is subjected to this cycling may be uncertain. The product of the load frequency and the time of cycling will be the total number of cycles occurring in a unit history (e.g., one mission).

In some cases, the available information about loading uncertainty is indexed to remote mechanical or thermal loads applied to the entire structural system, but the fatigue damage occurs at a specific fatigue-critical location subjected to a specific local stress state. The analytical or numerical transfer function used to convert the remote loads to the local stresses (for example, a finite element analysis, or a regression analysis) may introduce uncertainty due to modeling error. Local residual stresses (if any) may introduce additional uncertainties.

Life Scatter. Even if the initial crack size and state and the applied stresses are known exactly, there will still be uncertainty associated with the life prediction analysis. This uncertainty, which is here described as “life scatter,” arises from two fundamental sources. The first source is inherent material variability. This variability will be reflected in the materials data

used to determine the material properties, and so it may be appropriate to perform a statistical analysis of the supporting materials data. Unfortunately, replicate FCG data of sufficient quantity to support a rigorous statistical analysis are rarely available, and so estimates are often required. Since FCG material properties involve multiple parameters (for example, the Paris coefficient and exponent), statistical correlations between the different parameters must be addressed to accurately describe the overall material variability. In some cases, it may be an acceptable approximation to treat one or more FCG property parameters as deterministic and to express all of the variability in the other parameter(s).

The second source is inaccuracy in the FCG life prediction model itself (including the stress intensity factors, the FCG equation, and the load interaction model). No model will be able to address the complex effects of crack and component geometry, load history, temperature, and environment with perfect accuracy. Unfortunately, the uncertainties arising from life modeling errors are difficult to quantify in isolation, because the test data that might be used to characterize the modeling uncertainties also contain inherent material scatter to some degree.

As a result, material scatter and life model error are sometimes described by a single life scatter variable. This parameter may be derived from a statistical analysis of actual and predicted laboratory test results employing the relevant FCG life model. The statistics of the pooled and ordered actual-to-predicted (A/P) values provide a quantitative assessment of both the mean A/P value (describing bias in the life model) and the scatter in the data (expressible as a coefficient of variation (COV) for life scatter). The test data base should comprise an appropriate range of different material sources, load histories, and test conditions relevant to the structural application in question. Finally, if the environment (temperature or chemistry) has a significant impact on material resistance to crack growth, and if there is significant uncertainty in the environment (or in the nature of the environmental effects), then these factors may need to be addressed.

11.7 Probabilistic Structural Analysis Examples

Two probabilistic structural analysis examples are presented in this section. Section 11.7.1 presents an example of a stress versus strength evaluation. Section 11.7.2 presents an example of a probabilistic finite element analysis. More examples as well as expanded discussions of the approaches are given in [11-25].

11.7.1 Example of a Probabilistic Stress versus Strength Analysis

This first example examines a load bearing structure that can be modeled as a cantilever beam (Figure 11-8). The constant load at the tip of the beam will cause maximum deflection at the beam tip and maximum stress on the upper surface of the beam root and the beam can fail either by exceeding a yield stress or by exceeding a deflection limit in this scenario.

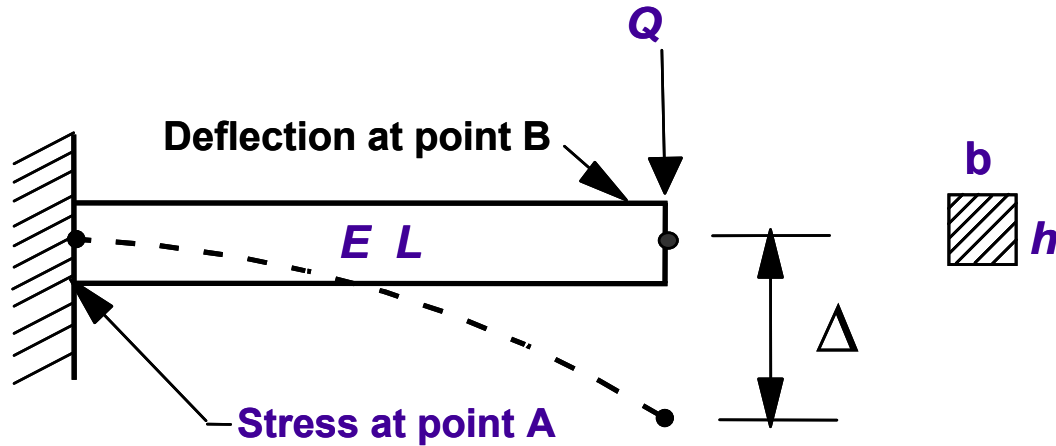


Figure 11-8. Cantilever Beam.

The stress for this component is the bending moment, M , times the distance from the neutral axis to the outer surface, c , divided by the moment of inertia I . The bending moment and moment of inertia can be expressed in terms of the load, Q , beam length, L , the beam base and height, b and h :

$$\text{Stress} = S = \frac{Mc}{I} = \frac{6QL}{bh^2} \quad (11-41)$$

The component fails when the stress, S , exceeds the yield strength, R . The limit state can be formulated as

$$g = R - S = R - \frac{6QL}{bh^2} = 0 \quad (11-42)$$

The nominal, mean values for these variables are given in Table 11-2.

Table 11-2. Parameters for the Stress Limit State.

Variable	Mean
Length, L	18 in
Width, b	1 in
Height, h	1.5 in
Load, Q	1.888 kips
Yield Limit, R	120 ksi
Young's Modulus	30,000 ksi

If any of these variables have uncertainty, then they can be modeled as random variables and the probability of failure is

$$p_f = P[g < 0] = P[R - S < 0] = P\left[R - \frac{6QL}{bh^2} < 0\right] \quad (11-43)$$

The manufacturing process, material quality, and environment will determine which variables have uncertainty. One scenario is that the material is very high quality and the manufacturing process (machining beam dimensions for examples) is very tightly controlled. This may lead to very small variations in the geometry and yield strength resulting in uncertainties only in the load (environment). A probability density function can be defined to describe the variations in the

load. This distribution information may come for measured data, similar components, or expert knowledge. For this example, the load Q is defined as a normal probability density function with a mean of 1.888 kips and a standard deviation of 0.283 kips. This one random variable problem can be solved using approaches developed in the previous section.

$$\mu_g = R - \frac{6QL}{bh^2} = 120 \text{ ksi} - \frac{6(1.888 \text{ kips})(18 \text{ in})}{(1 \text{ in})(1.5 \text{ in})^2} \quad (11-44)$$

$$= 120 \text{ ksi} - 90.624 \text{ ksi} = 29.376 \text{ ksi}$$

The standard deviation of the limit state is

$$\sigma_g = \sqrt{\left(\frac{6L}{bh^2}\right)^2 \sigma_Q^2} = \sqrt{\left(\frac{6(18 \text{ in})}{(1 \text{ in})(1.5 \text{ in})^2}\right)^2 (0.283 \text{ kips})^2} = 13.584 \text{ ksi} \quad (11-45)$$

Finally, the probability of failure is computed from the mean and standard deviation as follows:

$$p_f = \Phi\left[-\frac{\mu_g}{\sigma_g}\right] = \Phi[-2.1625] = 0.0153 \quad (11-46)$$

11.7.2 Example of a Probabilistic Finite Element Analysis

In this example, several uncertainties are propagated through a finite element model of an elastic isotropic simply supported beam. This example is similar to the previous one but the beam is now supported. This second examples shows how more detailed PFM is performed when more accurate and detailed results are to be obtained. A point load, P , acts in the downward direction at the center of the beam. Due to symmetry, only one half of the beam is modeled with finite elements. The geometry, boundary conditions, loading and finite element discretization are shown in Figure 11-9. The response of interest is the maximum stress due to bending, which occurs at either node 3 (tensile) or 9 (compressive).

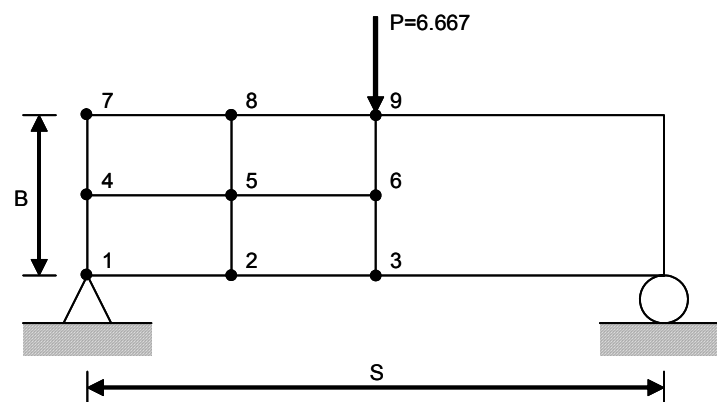


Figure 11-9. Beam Finite Element Example.

The finite element mesh shown in Figure 11-9 is used only to illustrate the process of performing a simple probabilistic FEA. In reality, a much finer discretization of finite elements would be required to resolve the computed stresses with sufficient accuracy.

Uncertainties considered in the probabilistic finite element analysis include the magnitude of the loading and the depth and length of the beam. All uncertainties are characterized using probability distributions. The specific input is listed in Table 11-3. The goal of the probabilistic FEA is to quantify the effect of these input uncertainties on the maximum computed stress in the beam.

Table 11-3. Uncertain Inputs for the Simply Supported Beam Example.

Input Parameter	Mean Value	Standard Deviation	Distribution
Load, P	3.33333 ^a	0.33333	Lognormal
Length, S	8.0	0.8	Normal
Depth, B	2.0	0.2	Normal

^a ½ the load value due to symmetry in the model

A representative finite element input file for the simply supported beam shown in Figure 11-9 is listed in Table 11-4. This input is not intended to be specific to any particular finite element analysis software but rather representative of the input required for any FEA code. The various sections of the input are shown in bold in the table.

Table 11-4. Example Finite Element Input.

```

*NODE
1, 0.0, 0.0
2, 2.0, 0.0
3, 4.0, 0.0
4, 0.0, 1.0      Nodal coordinates
5, 2.0, 1.0
6, 4.0, 1.0
7, 0.0, 2.0
8, 2.0, 2.0
9, 4.0, 2.0

*ELEMENT
1, 1, 2, 5, 4
2, 2, 3, 6, 5      Element connectivity
3, 4, 5, 8, 7
4, 5, 6, 9, 8

*MATERIAL
*ELASTIC      Material properties
30.0E3, 0.3

*BOUNDARY
1, 2, , 0.0
3, 1, , 0.0      Boundary conditions
6, 1, , 0.0
9, 1, , 0.0

*LOAD      Loading
9, 2, 3.3

```

The most straightforward technique of performing a probabilistic FEA is to execute the FEA software from the probabilistic analysis software. The basic functions of the interface are to: 1) modify the FEA input given a specific sample of the input uncertainties, 2) execute the FEA software, and 3) extract the result of interest. By properly interfacing the two software codes,

the probabilistic analysis can be fully automated regardless of the probabilistic method employed.

The last step is the results extraction. The desired FEA output quantity must be returned to the probabilistic analysis software. For efficiency, most FEA software programs store computed results in a binary output file. Therefore, a results extraction program will typically be required to locate the desired output quantity in the file and then return this value to the probabilistic analysis program.

A typical result from a probabilistic analysis of the simply supported beam is shown in Figure 11-10. The figure shows the cumulative distribution function (CDF) of the maximum stress in the beam. Comprehensive uncertainty and sensitivity studies can also be produced showing uncertainty contributions and the most sensitive variables.

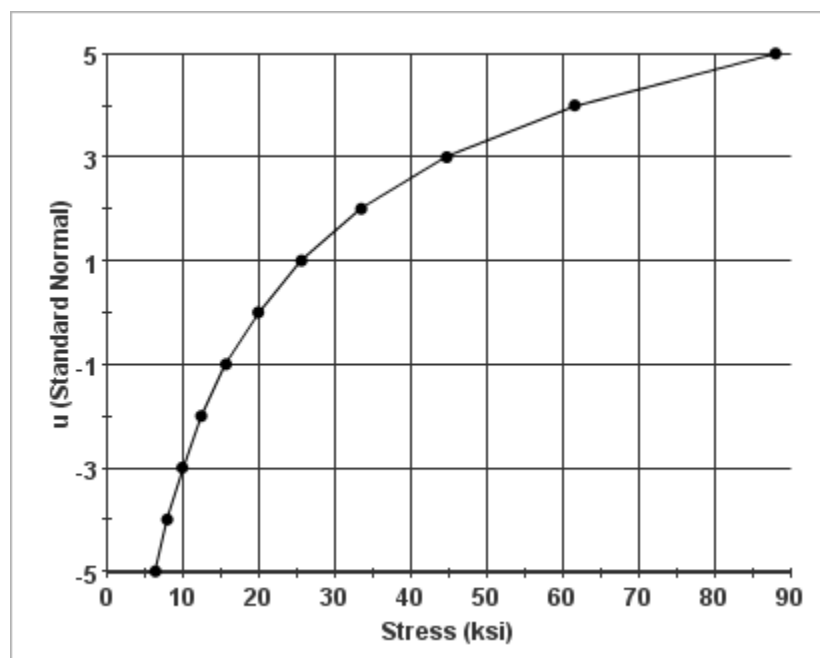


Figure 11-10. CDF of Maximum Stress for the Three-Point Bend Specimen Plot on Normal Probability Scale.

11.8 References

- 11-1 Southwest Research Institute®, *Probabilistic Structural Analysis Methods (PSAM) for Select Space Propulsion System Components*, Final Report, NASA Contract NAS3-24389, NASA Lewis Research Center, Cleveland, OH, 1995.
- 11-2 J. Townsend, and J. Peck, J., “*Probabilistic Structural Analysis of the SRB Aft Skirt External Fitting Modification*,” AIAA Paper No. 99-1577, 40th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, St. Louis, MO., April 12-15, 1999.
- 11-3 L. J. Huyse, C. J. Waldhart, D. S. Riha, B. H. Thacker, C. E. Larsen, R. J. Gomez, and P. C. Stuart. *Space Shuttle Debris Impact Assessment: Probabilistic Analysis and Model Verification and Validation*. Presented at the 48th

- AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Waikiki, HI, April 23-26, 2007.
- 11-4 N. Metropolis and S. Ulam, "The Monte Carlo Method," *J. American Statistical Association*, Vol. 44, 1949, pp. 335-341.
- 11-5 R. Y. Rubinstein, *Simulation and the Monte Carlo Method*, John Wiley & Sons, New York, 1981.
- 11-6 M. D. McKay, W. J. Canover, and R. J. Beckman "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, 1979, pp. 239-245.
- 11-7 Y.T. Wu, "Computational Method for Efficient Structural Reliability and Reliability Sensitivity Analyses," *AIAA Journal*, Vol. 32, 1994, pp. 1717-1723.
- 11-8 R. Rackwitz and B. Fiessler, "Structural Reliability Under Combined Load Sequences," *Journal of Computers and Structures*, Vol. 9, 1978, pp. 489-494.
- 11-9 A. H.-S. Ang, and W. H. Tang, *Probability Concepts in Engineering Planning and Design*, Volume II: Decision, Risk, and Reliability, John Wiley & Sons, Inc., New York, 1984.
- 11-10 H. O. Madsen, S. Krenk, and N. C.Lind, *Methods of Structural Safety*, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1986.
- 11-11 B. Fiessler, H. J. Neumann, and R. Rackwitz, "Quadratic Limit States in Structural Reliability," *Journal of Engineering Mechanics*, Vol. 105, 1979 pp. 661-676.
- 11-12 L. Tvedt, "Distribution of Quadratic Forms in Normal Space Application to Structural Reliability," *Journal of Engineering Mechanics*, Vol. 116, No. 6, 1990, pp. 1183-1997.
- 11-13 K. Breitung, "Asymptotic Approximation for Multi-normal Domain and Surface Integrals," Fourth International Conference on Application of Statistic and Probability in Soil and Structural Engineering, University di Firenze, Pitagora Editrice, 1983.
- 11-14 Y. T. Wu, et al., "Advanced Probabilistic Structural Analysis Method for Implicit Performance Functions," *AIAA Journal*, Vol. 2, No. 9, Sept. 1990.
- 11-15 B. H. Thacker, D. S. Riha, H. R. Millwater, and M. P. Enright, "Errors and Uncertainties in Probabilistic Engineering Analysis," AIAA Paper 2001-1239, Proc. AIAA/ASME/ASCE/ AHS/ASC 42nd Structures, Structural Dynamics, and Materials (SDM) Conf., Seattle, WA, April 2001.
- 11-16 D. S. Riha, B. H. Thacker, and S. H. K. Fitch, "NESSUS Capabilities for Ill-Behaved Performance Functions," AIAA Paper 2004-1832, Proc. AIAA/ASME/ASCE/AHS/ASC 45th Structures, Structural Dynamics, and Materials (SDM) Conf., Palm Springs, CA, April 2004.
- 11-17 Schueller, G.I. (Ed.), "Structural Reliability Software," *Structural Safety - Special Issue*, 28, Nos. ½, 2006, pp. 1-216.
- 11-18 "Fracture Control Requirements for Payloads Using the Space Shuttle," NASA-STD-5003, October 7, 1996.

- 11-19 “*Fracture Control Implementation Handbook for Payloads, Experiments, and Similar Hardware*,” NASA-HDBK-5010, May 24, 2005.
- 11-20 “*General Fracture Control Requirements for Manned Spaceflight Systems*,” NASA-STD-5007, March 13, 2001.
- 11-21 “*Fracture Control Requirements for Spaceflight Hardware*,” Interim NASA Technical Standard NASA-STD-(I)-5019, in review.
- 11-22 S. J. Hudak, Jr., L. Huyse, G. G. Chell, Y.-D. Lee, D. S. Riha, B. H. Thacker, R. C. McClung, B. M. Gardner, and J. B. Pleming, “A Probabilistic Fracture Mechanics Assessment of the Orbiter’s LH2 Feedline Flowliner in Support of Return-to-Flight,” Proc. 9th Joint FAA/DoD/NASA Conference on Aging Aircraft, Atlanta, Georgia, March 2006. See also S. J. Hudak, Jr., et al., “*Probabilistic Fracture Mechanics Analysis of the Orbiter’s LH2 Feedline Flowliner*,” Final Contractor Report, NASA CR-2005-213585, June 2005, available at:

<http://gltrs.grc.nasa.gov/citations/all/cr-2005-213585.html>
- 11-23 “*Probabilistic Assessment of Knife Edge Seal Cracking in Space Shuttle Main Engine High Pressure Oxidizer Turbopumps*,” Final Report to NASA SSME Project Office, February 2007.
- 11-24 D. S. Riha, R. C. McClung, M. P. Enright, C. J. Waldhart, C. F. Popelar, B. M Gardner, S. S. Pai, K. Head, K. Haake, S. Montgomery, G. Prueger, G. Swanson, P. Allen, and E. Mendoza, “Probabilistic Assessment of Knife Edge Seal Cracking in Space Shuttle Main Engine High Pressure Oxidizer Turbopumps,” Submitted to the 10th AIAA Non-Deterministic Approaches (NDA) Conference, Chicago, IL, April 7-10, 2008.
- 11-25 D.S. Riha, M.P. Enright, R. Craig McClung, and Ben H. Thacker, *Probabilistic Structural Analysis*, Southwest Research Institute Report Under Contract NAS3-02142, Prepared for Dr. Shantaram S. Pai, NASA Glenn Research Institute, November 2007.

12. Uncertainty Propagation

Randomness (natural variability) of physical processes modeled in PRAs imposes the use of probabilistic models, which is central to risk analysis. Additionally, the development of scenarios introduces model assumptions and model parameters that are based on what is currently known about the physics of the relevant processes and the behavior of systems under given conditions. Because there is uncertainty associated with these conditions, probabilistic models are also used to represent our state of knowledge regarding the numerical values of the parameters and the validity of the model assumptions. It is important that the uncertainties in the natural variability of physical processes (i.e., aleatory uncertainty) and the uncertainties in the knowledge of these processes (i.e., epistemic uncertainty) are properly accounted for. Epistemic uncertainty associated with phenomenological models will be described in this chapter.

This chapter focuses on the uncertainties regarding the numerical values of the parameters of a given model (parameter uncertainty), rather than on the uncertainty regarding the validity of the model itself. It examines the technique to propagate the uncertainty in the risk model outputs induced by epistemic uncertainties in input parameter values. With *uncertainty propagation* bounding values for output functions of the PRA can be estimated. These bounds are associated with a probability that has bounds, which include the true value of the numerical result predicted by the model. Here, the term “output function” refers to the function corresponding to the risk metric of interest. The term “input parameters” represents the uncertain input to the output function.

As stated in Chapter 7, the widely used framework within which the uncertainty of the model output is calculated is the Bayesian. In this approach the uncertain input parameters are characterized by probability distributions. The approach follows a two-step process:

- Construct a probability density function (pdf) for each input parameter value (a pdf reflects state of knowledge about the value of the parameter); and
- Generate a probability distribution for the output function by mathematically combining the probability distributions on the values of the input parameters using an appropriate mapping technique.

Care should be exercised in interpreting the probability distribution obtained for the output function. The resulting distribution represents only a portion of the uncertainty, which arises from uncertainty in the parameter values. The distribution is predicated on validity of:

- The modeling assumptions made in the PRA model; and
- The distributions assumed for the input parameters.

The uncertainty associated with the risk model assumptions is handled with sensitivity analysis.

The following techniques have been used for propagation of uncertainties [12-1]:

- **Sampling**—The distributions for input parameters are mapped using crude Monte Carlo or Latin Hypercube sampling (LHS) techniques to obtain an empirical distribution for the output function;

- **Moment propagation**—First and second moments of the input parameters are mapped to obtain the mean and variance of the output function using variance/covariance propagation; and
- **Discrete Probability Distribution**—The distributions for input parameters are converted to discrete probability distribution before mapping. The resulting distribution for the output function is empirical.

In this guide only the sampling technique is described. This is because this technique has become the industry standard for propagating uncertainties. Sampling analysis is often supported by PRA codes. In addition, powerful spreadsheet-based sampling codes are now available that allow the PRA analysts to easily set up and execute complex analysis tasks.

12.1 Problem Statement for Uncertainty Propagation

Suppose R is an output of the risk model. Mathematically, R can be represented with a function h with uncertain input quantity x_i :

$$R = h(x_1, x_2, \dots, x_i, \dots, x_j, \dots, x_n) \quad (12-1)$$

The main question to be investigated is the following:

How does R vary when the set of x_i that are uncertain vary according to their assumed probability distributions?

The question of what the confidence bounds and other statistics (e.g., median) are for the output function is closely linked to the question posed above.

How to Interpret x_i in Equation (12-1)

Suppose that the basic event x_i in the logic model is failure of a certain component to perform its mission, and that this failure probability is symbolized by $\Pr(x_i)$. If λ_i is the component failure rate and t_i its mission time, then

$$\Pr(x_i) = 1 - e^{-\lambda_i t_i} \quad (12-2)$$

Typically, the failure rate is predicated upon a Bayesian analysis of available data and has an uncertainty distribution. Let $\pi(\lambda_i)$ be the epistemic pdf for the failure rate in Equation (12-2). For the purpose of illustration, postulate that $\pi(\lambda_i)$ is a lognormal density function with parameters: $\mu_1 = -6.91$ and $\sigma_1 = 0.658$. (Section 6.5 describes the properties of lognormal density functions). This corresponds to a mean failure rate of 10^{-3} per hour, and an error factor (defined as the ratio of the 95th percentile to the median) of 3. The curve shown in the lower left corner of Figure 12-1 depicts $\pi(\lambda_i)$.

Combining Equation (12-2) with the pdf of λ_i results in the pdf for $\Pr(x_i)$ once the mission time is specified. Let it be assumed that t_i is 0.1 hour (i.e., 6 minutes). Then the graph in the upper left corner of Figure 12-1 is the pdf for $\Pr(x_i)$.

Relative to Equation (12-1), $\Pr(x_i)$ is an example of an uncertain input event. This is because the fundamental parameter used to calculate the probability of x_i (i.e., the failure rate λ_i) has epistemic uncertainty.

12.1.1 How Does Sampling Work?

Sampling can be described as a thought experiment in which a system of many components with varying failure rates is to be analyzed to obtain its failure probability. For example, we can imagine that we have thousands of components of type x_i (mentioned above) with different failure rates that follow the lognormal distribution with parameters $\mu_1 = -6.91$ and $\sigma_1 = 0.658$. For a fixed t of 6 minutes, the failures of these thousands of components will give us a set of probability values for x_i . These probabilities will be distributed because λ_i is distributed. Now, in real life, we don't have the benefit of this experiment, but we can simulate it, as long as we are careful enough to select an appropriate set of values for λ_i . The process of selecting the set of possible values for λ_i consistent with its distribution is called sampling.

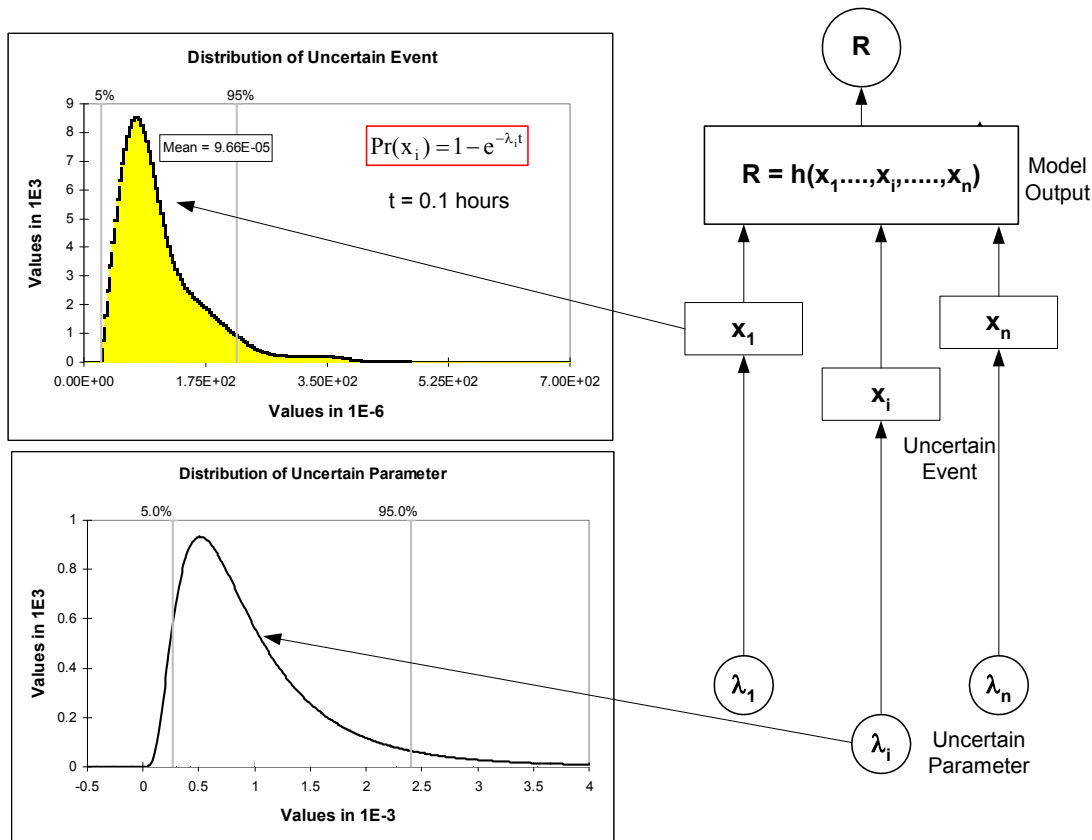


Figure 12-1. Propagation of Epistemic Uncertainties.

The sampling technique uses random numbers (generated by a random number generator) and random (or stratified) sampling of the distributions assumed for the input parameters to obtain an empirical distribution for the output function. A value is drawn at random from the probability distribution of each input parameter. The set of random values, one for each input

parameter, is used to quantify the output function. The quantification of the output function in each simulation trial is referred to as “an iteration.” The process is repeated n times producing n independent output values. These n output values represent a random sample from the probability distribution of the output function. With enough iterations, the sampled values obtained for the probability distribution of each input parameter will approximate its assumed probability distribution.

The two commonly used sampling methods are crude Monte Carlo sampling and LHS. To aid in understanding the difference between these sampling techniques, the reader is encouraged to review the concept of cumulative distribution function (CDF) as discussed in Section 4.3.2.

12.1.2 Crude Monte Carlo Sampling

This is the traditional technique to sample from a probability distribution. In this technique the sampling is completely random. That is, a value is drawn at random from the distribution for each input parameter. Of course, samples are more likely to be drawn in the areas of distribution where the probability of occurrence is higher (Figure 12-2). Because of this property, low probability areas of the distribution (i.e., tail of the distribution) may not be represented adequately in the samples. As a result, a relatively high number of iterations is required to obtain reliable estimates of the output function. This issue is particularly problematic for risk and reliability models that employ skewed probability distributions.^a This problem led to development of LHS technique [12-2].

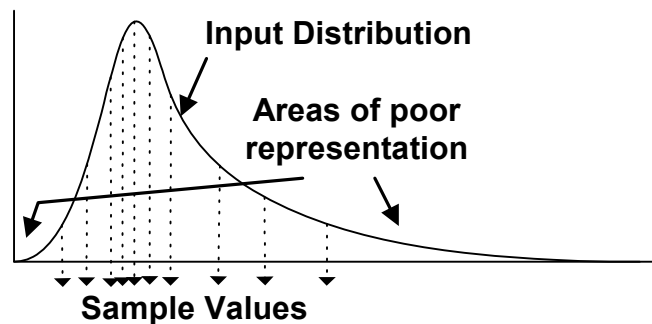


Figure 12-2. Crude Monte Carlo Sampling.

12.1.3 Latin Hypercube Sampling

The sampling technique used in LHS is based on “sampling without replacement.” In this technique the cumulative distribution function for an input parameter is divided up into intervals. A single value is sampled at random from within each interval (or stratification) according to the probability distribution of the input parameter. This sampling technique is illustrated in Figure 12-3. In this illustration, the cumulative curve has been divided into four intervals.

Because the coverage of sampling over the input domain is more uniform, a smaller number of samples is required. For this reason LHS is more appealing than crude Monte Carlo sampling.

a. Skewed distributions have more values to one side of the peak; one tail is much longer than the other.

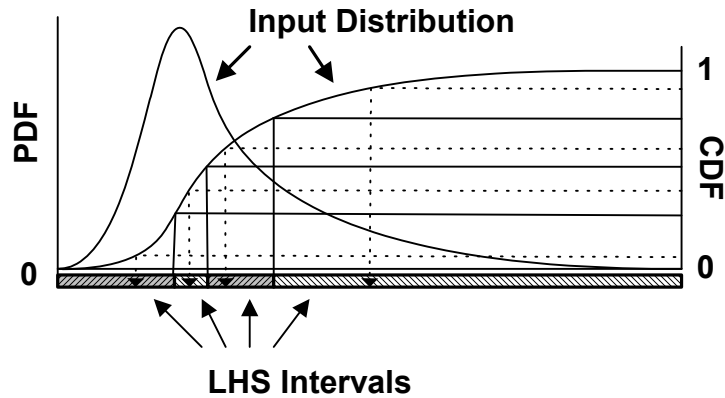


Figure 12-3. Latin Hypercube Sampling (LHS) Technique.

12.2 Achieving Convergence

The precision in the propagated distributions is improved by increasing the number of samples. It is important to run enough iterations so that the statistics generated for the output function are reliable. Therefore, care should be exercised to include enough sample iterations to achieve statistical regularity. By statistical regularity we mean as more iterations are run, the distribution for the output function becomes more stable as the statistics describing the distribution change less and less with additional iterations. The statistics of interest that should be monitored for convergence include the expected value and the standard deviation. With advanced simulation software codes, the analyst has the ability to monitor the change in the statistics of the output function at selected intervals (such as every 100 iterations). The simulation can be programmed to stop automatically once the changes in statistics meet the convergence criteria defined by the analyst. In the absence of any specified convergence criteria, the following steps can be taken to achieve convergence:

1. Set the number of iterations to at least 5,000 and run the simulation.
2. Records the statistics for:
 - mean;
 - standard deviation;
 - 5th percentile;
 - median (50th percentile); and
 - 95th percentile.
3. Perform additional simulations by increasing the number of iterations by increments of at least 1,000.
4. Monitor the change in above statistics.
5. Stop if the average change for each statistic (in two consecutive simulations) is less than a desired value (generally in the range of 1 to 5%). Note that some software performs convergence checking as a part of the uncertainty analysis.

12.3 Example: Uncertainty Propagation for an Accident Scenario Using LHS

Figure 12-4 shows two FTs associated with two standby systems (System A and System B). System A represents a two-train redundant system that consists of two nominally identical devices (A1 and A2). Figure 12-5 shows an ET involving failure and success combinations of the two systems.

In these schematics:

- \bar{A} and \bar{B} denote failure of System A and B respectively;
- Scenario 1 ($IE \cap A \cap B$) represents the success path; and
- Scenario 4 ($IE \cap \bar{A} \cap \bar{B}$) is the risk scenario. Its frequency is the risk metric (R) of interest.

The reduced Boolean equation (rare-event approximation) for Scenario 4 has the following form (here, we use the symbol “+” for the *union* operation and the symbol “.” for the *intersection* operation.)

$$R = IE.ACC.B11 + IE.ACC.B12 + IE.A12.A22.B12 + IE.A12.A21.B12 + IE.A12.A22.B11 + IE.A12.A21.B11 + IE.A11.A22.B12 + IE.A11.A21.B12 + IE.A11.A22.B11 + IE.A11.A21.B11 \quad (12-3)$$

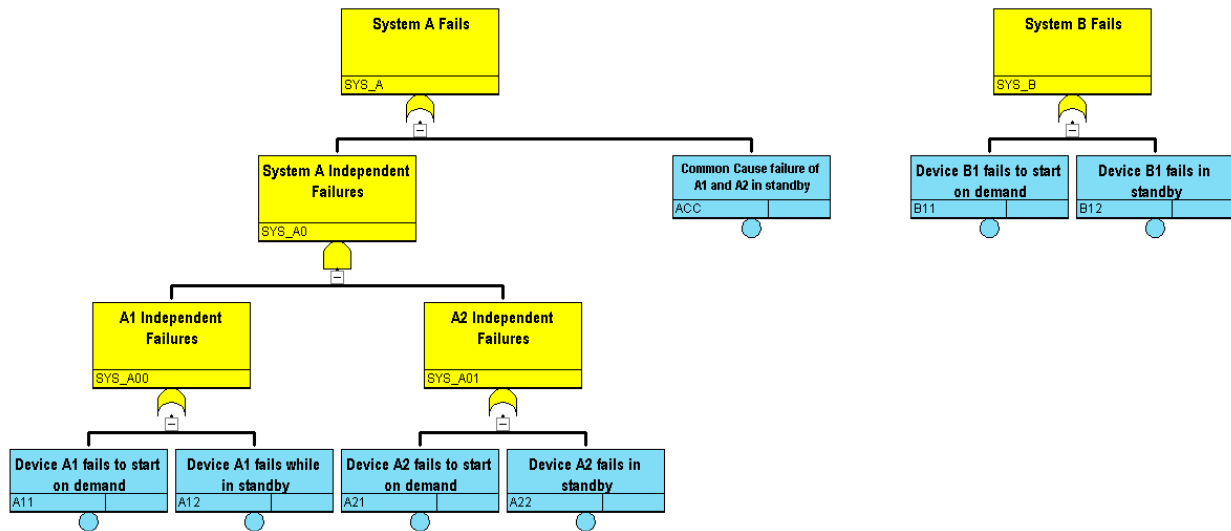


Figure 12-4. Fault Trees for Systems A and B.

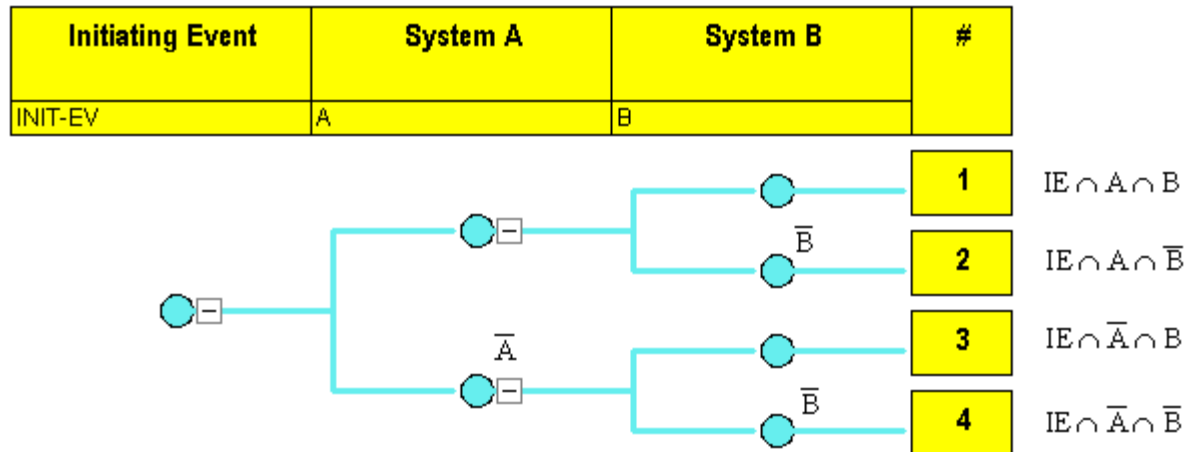


Figure 12-5. Event Tree for Uncertainty Propagation.

The list of the basic events along with their probabilities (unavailabilities) and the expressions used to calculate these probabilities is shown in Table 12-1. The following assumptions are made:

- Each standby device can fail to respond when actuated either due to:
 - Stress-related failures at the time of the demand^a or
 - Time-related failures while in standby (the failure time follows the exponential distribution)
- The average unavailability (Q) of each standby device has two contributions:
 - The probability it fails to start on demand (e.g., γ_{A1}); and
 - Half of the product of the failure rate λ , in standby, and the fault exposure time τ (e.g., $Q_{AII} = \frac{\lambda_{AII} \tau_A}{2}$).

The fundamental parameters used to calculate unavailabilities are identified in Table 12-2. It is assumed that the epistemic pdf for uncertain parameters (e.g., failure rate λ_{AII}) is lognormal. The shape and scale factors of the lognormal distribution assumed for each uncertain parameter are shown in Table 12-2.

a. In practice, distinguishing between stress-induced failures and standby failures may not be possible due to data limitations.

Table 12-1. List of Basic Events and Associated Uncertain Parameters.

Basic Event				Identification of Uncertain Parameters			
ID	Description	Unavailability		Expression Used To Calculate Expected Value	Parameter	Description	Treated as Random Variable?
		Symbol	Expected Value				
A11	Device A1 fails to start once demanded	Q_{A11}	1.0E-2	—	γ_{A1}	Probability of failure to start of device A1	Yes
A21	Device A2 fails to start once demanded	Q_{A21}	2.0E-2	—	γ_{A2}	Conditional probability of failure to start of device A2 given A1 failure	Yes
A12	Device A1 fails independent of A2 (while in standby)	Q_{A12}	3.0E-2	$\frac{1}{2}\lambda_{A1I}\tau_A$	λ_{A1I}	Failure rate of A1 (due to independent causes; per hour)	Yes
					τ_A	Fault exposure time for System A (168 hours)	No
A22	Device A2 fails independent of A1 (while in standby)	Q_{A22}	3.0E-2	$\frac{1}{2}\lambda_{A2I}\tau_A$	λ_{A2I}	Failure rate of A2 (due to independent causes; per hour)	Yes
					τ_A	Fault exposure time for System A (168 hours)	No
ACC	Common cause shock disables two redundant channels of System A (while in standby)	Q_{ACC}	3.0E-3	$\frac{1}{2}\lambda_{ACC}\tau_A$	λ_{ACC}	Failure rate of A1 and A2 (due to common cause; per hour)	Yes
					τ_A	Fault exposure time for System A (168 hours)	No
B11	Device B1 fails to start once demanded	Q_{B11}	2.0E-2	—	γ_{B1}	Probability of failure to start of device B1	Yes
B12	Device B1 fails (while in standby)	Q_{B12}	4.0E-2	$\frac{1}{2}\lambda_{B1I}\tau_B$	λ_{B1I}	Failure rate for device B1 (per hour)	Yes
					τ_B	Fault exposure time for System B (168 hours)	No
IE	Initiating event	—	1.0E-3 frequency	—	f_{IE}	Frequency of initiating event (per mission)	Yes

Table 12-2. Uncertainty Distributions for Uncertain Parameters.

Random Variable	Parameters of Epistemic Distribution (Lognormal)		Statistics	
	μ	σ	Expected Value	Variance
γ_{A1}	-4.83	0.668	1.0E-02	5.6E-05
γ_{A2}	-4.14	0.668	2.0E-02	2.2E-04
λ_{A1I}	-8.16	0.668	3.6E-04	7.2E-08
λ_{A2I}	-8.16	0.668	3.6E-04	7.2E-08
λ_{ACC}	-10.70	0.978	3.6E-05	2.1E-09
γ_{B1}	-4.14	0.668	2.0E-02	2.2E-04
λ_{B1}	-8.13	0.978	4.8E-04	3.6E-07
f_{IE}	-7.89	1.400	1.0E-03	6.1E-06

The point estimate for the risk metric can be obtained by directly substituting the average unavailability of each basic event in the Boolean expression (Equation (12-3)). Thus

$$\begin{aligned}
 R = f_{IE} \cdot (& Q_{ACC} \cdot Q_{B11} + Q_{ACC} \cdot Q_{B12} + Q_{A12} \cdot Q_{A22} \cdot Q_{B12} + \\
 & Q_{A12} \cdot Q_{A21} \cdot Q_{B12} + Q_{A12} \cdot Q_{A22} \cdot Q_{B11} + Q_{A12} \cdot Q_{A21} \cdot Q_{B11} + \\
 & Q_{A11} \cdot Q_{A22} \cdot Q_{B12} + Q_{A11} \cdot Q_{A21} \cdot Q_{B12} + Q_{A11} \cdot Q_{A22} \cdot Q_{B11} + \\
 & Q_{A11} \cdot Q_{A21} \cdot Q_{B11})
 \end{aligned} \tag{12-4}$$

Substituting the values of basic event probabilities shown in Table 12-1, the point estimate for the risk metric is calculated: $R_0 = 3.04E - 7$ per-mission.

For uncertainty propagation we need to express Equation (12-4) in terms of fundamental parameters^a. Using unavailability expressions listed in Table 12-1, the parametric representation of R (the output function) is obtained as

$$\begin{aligned}
 R = f_{IE} \cdot (& \frac{1}{2} \lambda_{ACC} \tau_A \gamma_{B1} + \frac{1}{4} \lambda_{ACC} \lambda_{B1} \tau_A \tau_B + \frac{1}{8} \lambda_{A1I} \lambda_{A2I} \lambda_{B1} \tau_A^2 \tau_B + \\
 & \frac{1}{4} \lambda_{A1I} \gamma_{A2} \lambda_{B1} \tau_A \tau_B + \frac{1}{4} \lambda_{A1I} \lambda_{A2I} \tau_A^2 \gamma_{B1} + \frac{1}{2} \lambda_{A1I} \tau_A \gamma_{A2} \gamma_{B1} + \\
 & \frac{1}{4} \lambda_{A2I} \lambda_{B1} \tau_A \tau_B \gamma_{A1} + \frac{1}{2} \lambda_{B1} \tau_B \gamma_{A1} \gamma_{A2} + \frac{1}{2} \lambda_{A2I} \tau_A \gamma_{A1} \gamma_{B1} + \gamma_{A1} \gamma_{A2} \gamma_{B1})
 \end{aligned} \tag{12-5}$$

a. Although the PRA codes do not produce the parametric representation of the risk metric as output, they internally generate and process the parametric expressions to perform quantification of the model.

In this example, even though devices A1 and A2 are physically distinct, the assumption that they are identical requires that the same failure rate be used for both devices. Let us assume $\lambda_{A1} = \lambda_{A2} = \lambda_{A1}$. The parametric representation of R can be rewritten as follows^a:

$$\begin{aligned}
 R = f_{IE} \cdot & \left(\frac{1}{2} \lambda_{ACC} \tau_A \gamma_{B1} + \frac{1}{4} \lambda_{ACC} \lambda_{B1} \tau_A \tau_B + \frac{1}{8} \lambda_{A1}^2 \lambda_{B1} \tau_A^2 \tau_B + \right. \\
 & \frac{1}{4} \lambda_{A1} \gamma_{A2} \lambda_{B1} \tau_A \tau_B + \frac{1}{4} \lambda_{A1}^2 \tau_A^2 \gamma_{B1} + \frac{1}{2} \lambda_{A1} \tau_A \gamma_{A2} \gamma_{B1} + \\
 & \left. \frac{1}{4} \lambda_{A1} \lambda_{B1} \tau_A \tau_B \gamma_{A1} + \frac{1}{2} \lambda_{B1} \tau_B \gamma_{A1} \gamma_{A2} + \frac{1}{2} \lambda_{A1} \tau_A \gamma_{A1} \gamma_{B1} + \gamma_{A1} \gamma_{A2} \gamma_{B1} \right)
 \end{aligned} \tag{12-6}$$

The LHS technique was employed to generate a distribution for the risk metric R by propagating the epistemic uncertainties in its parameters. For this example @RISK software [12-3] was used. This software operates in a Microsoft Excel® environment.

The parametric expression of R (i.e., right side of Equation (12-6) shown above) was entered into @Risk as the output function for uncertainty propagation. The parameters γ_{A1} , γ_{A2} , λ_{A1} , λ_{ACC} , γ_{B1} , λ_{B1} , and f_{IE} were declared as input variables whose uncertainties are defined according to Table 12-2.

The numerical distribution for R as generated by @Risk is shown in Figure 12-6. The statistics associated with this distribution are shown in column 2 of Table 12-3.

a. Note that in this expression the terms that reflect the average unavailability of two parallel devices are slightly underestimated (third and fifth). This is because for two components in parallel (A1 and A2 in this example) the average unavailability is $1/3\lambda^2\tau^2$ as opposed to $1/4\lambda^2\tau^2$ [4].

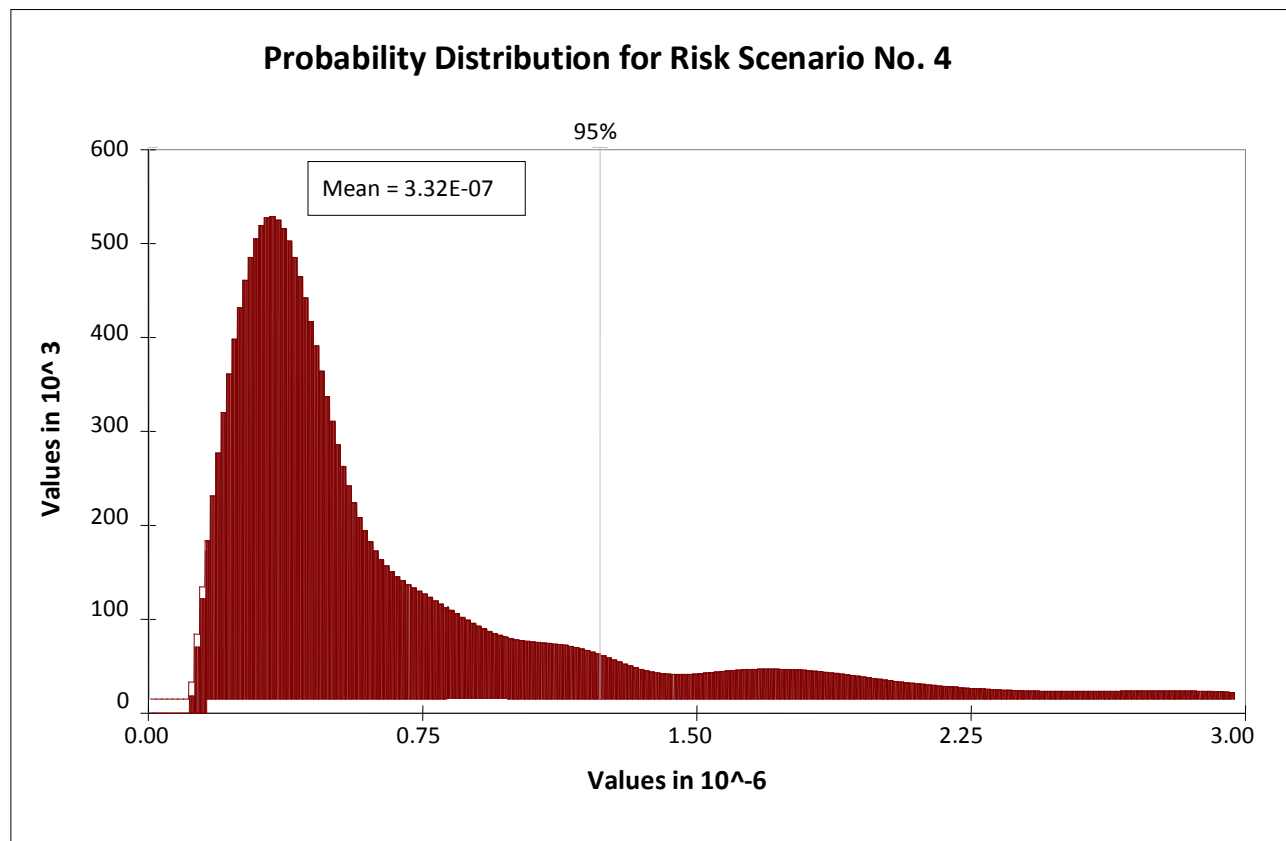


Figure 12-6. The pdf for the Risk Metric R.

Table 12-3. Statistics for Scenario 4 pdf.

Statistic	Value	
	When the Risk Metric is defined by Equation (12-6)	When the Risk Metric is Defined by Equation (12-5)
Mean	3.32E-07	3.06E-07
Variance	2.09E-12	3.00E-12
5% Percentile	4.34E-09	4.48E-09
50% Percentile	7.17E-08	6.91E-08
95% Percentile	1.24E-06	1.15E-06

Column 3 of Table 12-3 reflects the results of a run in which Equation (12-5) was declared as the output function. In this case despite the fact that our current state of knowledge about the failure rate of devices A1 and A2 is the same, the epistemic pdfs for λ_{A1} and λ_{A2} are assumed to be independent. This assumption leads to underestimation of the results as evident in this example (note the mean value is reduced by 5% when Equation (12-5) is used). This treatment of epistemic dependency between variables is the subject of the next section.

12.4 Treatment of Epistemic Dependency

It is important that epistemic dependency among variables of a risk model is correctly accounted for in the uncertainty propagation. The epistemic uncertainty in the failure rates of nominally identical components or basic events must be coupled. Failure to do so will lead to underestimation of the mean value of the results as well as an underestimation of its uncertainty [12-5]. The sources of epistemic dependency include:

1. Use of identically designed and manufactured components—Assigning the same failure rate (λ) to identical devices introduces dependency among the probabilities assigned to the affected basic events; and
2. Organizational factors—Operation, test, and maintenance of a system by the same staff introduce dependency between parameters that are used in the unavailability expression for the system. Examples of such parameters include the component failure rate, the frequency of maintenance, the downtime of system, and the human error probability (HEP).

Dependence among uncertain parameters is often specified in terms of correlation coefficient, which has a value between -1 and 1 . Positive coefficient values indicate a positive relationship between two variables (i.e., uncertain parameters), so that when the value sampled for one variable is high, the value sampled for the second variable will be high as well. Negative coefficient values indicate an inverse relationship, so that when the value sampled for one variable is high, the value sampled for the other variable will be low.

Due to lack of data, the PRA models often assume that the dependent variables are fully correlated. This is accomplished using one of the following techniques:

- Declare dependent variables as fully correlated with correlation coefficient of 1.0 between each pair. In this case each parameter maintains its identity in the output function (for example define λ_{A1} and λ_{A2} as correlated in Equation (12-5)).

- Define one epistemic pdf for all dependent variables similar to the treatment of failure rates λ_{A11} and λ_{A21} in the previous example (this requires that the output function be modified and all dependent variables be replaced by a single variable; similar to Equation (12-6)).

The reader should review the documentation of the simulation software to obtain specific instructions on handling of correlated variables. Chapter 10 provides additional information on the treatment of dependency.

12.5 Epistemic Uncertainty in Phenomenological Models

Risk assessment ultimately involves quantifying models. An example is demonstrating compliance with a requirement that the probability of avoiding contact with the pad (reliability) during launch is at least 0.999. One method for verifying compliance with the requirement is to develop a physics-based model for contacting the pad during launch, and quantifying it using Monte Carlo simulation. By sampling from the input distributions for the model parameters, the probability of contacting the pad could be quantified and compared with the requirement. Mathematically, if x is the minimum distance between a launch vehicle and an imaginary mathematical surface coincident with the pad, the probability of contacting the pad is the probability that x is negative. Although it may require thousands of lines of script to create, ultimately there is some function, $g(\alpha_1, \alpha_2, \dots, \alpha_N)$, which represents the physics-based model and satisfies the relationship:

$$x = g(\alpha_1, \alpha_2, \dots, \alpha_N) \quad (12-7)$$

If there are probability density functions for the input parameters, α_1 through α_N , Monte Carlo simulation can estimate the probability that x is negative. If that probability is 10^{-3} or less, it could claim that the requirement is satisfied. Of course, if only a small sample size is used in performing the simulation the simulation may not have converged to an accurate estimate of the probability that x is negative. Achieving convergence is a very necessary consideration when evaluating the uncertainty in the risk of pad contact, and in any Monte Carlo simulation.

There are, however, other contributors to the uncertainty that the risk requirement is satisfied besides convergence. These other contributors can best be appreciated in the context of epistemic uncertainty described, for example, in Reference [12-6]. The context is illustrated in Figure 12-7.

The universe is vast, and compared to it our Milky Way galaxy is insignificant. Within the Milky Way our solar system is insignificant, and on Earth our launch complex is a mere speck. Thus, compared to the entire universe (i.e., the *World* in Figure 12-7) our launch complex is entirely irrelevant but, conversely, we do not expect most of the *World* to have any appreciable risk impact on our launch complex. Consequently, our model (i.e., the script representing Equation (12-7)) will ignore most of the *World*.

Even within the launch complex our model will exclude much. Most likely, instead of the entire complex our model will be constrained to the launch vehicle, the pad, and support facilities with which there is an interface. The model will also likely include the launch environment (e.g., meteorological conditions). While planets, stars, and galaxies are unlikely to pose any risk to launch, the world in proximity to those portions of the launch complex included in our model could potentially have a non-negligible risk impact. Unless, as Appendix D

recommends, a broad spectrum of candidate events is examined to ensure that the risk model is reasonably

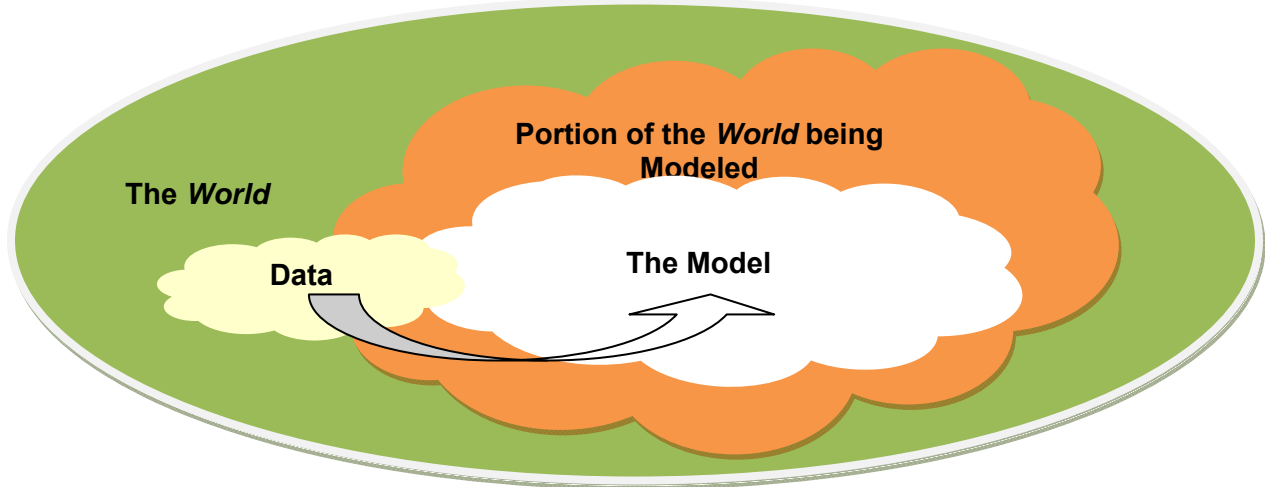


Figure 12-7. A Context for Epistemic Uncertainty in Risk Assessments

complete, verifying that the model results actually satisfy the requirement becomes problematic. In the context of Reference [12-6], this limitation on our ability to verify that the requirement is satisfied constitutes *completeness uncertainty*.

All models contain simplifications and approximations. Hence, even for that portion of the *World* included in Equation (12-7), our model is inexact. Unless we can demonstrate that our risk predictions are insensitive to our simplifications and approximations the associated *model uncertainty* (in the context of Reference [12-6]) limits our ability to verify compliance with the imposed requirement.

Postulate that a sufficiently broad spectrum of events has been examined so that there are compelling arguments to convince reviewers that *completeness uncertainty* is negligible. Hypothesize further that a comprehensive set of sensitivity or other studies conclusively demonstrate that the model (e.g., Equation (12-7)) has so significant uncertainty. Even then there remains the issue Reference [12-6] categorizes as *parameter uncertainty*.

Monte Carlo simulation will quantify the probability that the risk requirement is satisfied by repeatedly sampling from the probability distributions characterizing the model parameters (α_1 through α_N in Equation (12-7)). However, verifying that the result of this quantification is acceptable for comparison with the imposed risk requirement necessitates understanding the uncertainty in these parameter distributions. As examples, if some distributions were derived from laboratory tests where the test environment was fairly benign, the resultant distributions may tend to underestimate harsh launch conditions and overestimate the probability of a successful launch. If other distributions were obtained from field data under conditions far more harsh than is expected during launch, these input parameters will tend to overestimate launch risk.

Relative to *parameter uncertainty*, it is necessary to assess the:

- Probability that a specific model, or collection of candidate models, apply to each input parameter (e.g., there is high confidence that parameter, α_n , can be characterized by a normal distribution, or the probabilities that a Weibull or lognormal distribution describe α_n are similar, so a weighted combination of these two distributions should be applied); and
- Uncertainty in the distribution parameters (e.g., the mean and standard deviation for a normal distribution, or the shape and scale parameters for a Weibull distribution).

Even if there were sufficient resources to ensure that a Monte Carlo simulation had converged to several significant digits, if the:

- input data;
- models (e.g., Equation (12-7)); or
- phenomena ignored by the models;

are uncertain, this uncertainty will translate into uncertainty in the risk prediction. Without addressing these contributors to risk uncertainty, compliance with probabilistic requirements cannot be verified.

Section 2.1 defines risk as including uncertainty, so in order to integrate results from phenomenological models into PRA, some understanding of their uncertainty (i.e., the epistemic uncertainty explained in the context of Figure 12-7) is needed. Without an understanding of uncertainty there is no context within which decision-makers can determine, with confidence, whether quantitative risk requirements are satisfied or one design option is truly superior to another.

Given the computational challenges associated with phenomenological modeling it is unrealistic to apply Monte Carlo simulation techniques such as those described in Section 12 to the quantification of epistemic uncertainty in phenomenological models. The resources required for such an approach would, in most instances, be inordinate. However the use of sensitivity studies involving the statistical data and physics-based modeling assumptions could furnish some insight into how much confidence should be applied to the phenomenological model results. Such an approach has already been recommended to enhance understanding of phenomenological model results, and must serve as a substitute for the more rigorous, quantitative techniques applicable elsewhere. Sections 13.4 and 14.9.2 provide guidance relating to sensitivity studies in the context of PRA.

12.6 References

- 12-1 M.G. Morgan, and M. Henrion., *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*, Cambridge Press, 1990.
- 12-2 R.L. Iman, J.M. Davenport, and D.K. Zeigler, *Latin Hypercube Sampling—A Program Users Guide*, Technical Report SAND79-1473, Sandia National Laboratories, Albuquerque, NM, 1980.

- 12-3 @Risk 4.5 for PC Excel, Palisade Corporation.
- 12-4 G. Apostolakis, and T.L. Chu, "The Unavailability of Systems Under Periodic Test and Maintenance," *Nuclear Technology*, 50, 5-15, 1980.
- 12-5 G. Apostolakis, and S. Kaplan, "Pitfalls in Risk Calculations," *Reliability Engineering*, 2, 133-145, 1981.
- 12-6 Regulatory Guide 1.174, *An Approach for Using Probabilistic Risk Assessment in Risk-Informed Decisions on Plant-Specific Changes to the Licensing Basis*, Rev. 1, November 2002.

13. Presentation of Results

While this chapter discusses the presentation of PRA results as though it is a one-time activity, it should be recognized that in general PRAs are done in stages or are periodically updated, and their evolving results may be presented numerous times at discrete points in the life cycle, rather than just once at the "end." The Space Shuttle, International Space Station, and Constellation are examples of programs wherein the results of "living PRAs" are presented many times over the course of their life cycles.

As discussed earlier, a risk analysis generally consists of the following three analysis tasks:

- Identification of accident scenarios;
- Estimation of the likelihood of each scenario; and
- Evaluation of the consequences of each scenario.

The final step in performing a PRA is to integrate the data obtained in the above analysis tasks and to interpret the results. The integration includes, among other things, development of best estimates for frequencies and consequences, development of distributions reflecting the uncertainty associated with those estimates, propagation of the uncertainties to obtain final results, and development of appropriate displays to communicate the results with their associated uncertainties.

It is also imperative to check the results for accuracy. This will ensure that the model of the world is a technically reasonable representation of the entity being evaluated, and its mission. Documentation related to PRA models whose analysis results are used to make critical decisions regarding design, development, manufacturing, and ground or flight operations that may impact human safety or program-defined mission success criteria should be reviewed. Specific methods and procedures should be used for assessing and communicating the credibility of PRA model analysis results based on factors such as peer review, input pedigree, uncertainty analysis, results robustness, use history, qualifications of the analysts, and the Technical Authority. [13-1, 13-2].

To provide focus for the presentation of results, the results should include identification of system features that are the most important contributors to risk. Insights into relative importance of various features of the system, and the relative importance of various modeling assumptions, may be developed from uncertainty and sensitivity analyses. A discussion of these insights is required to provide the proper interpretation of the "bottom line" conclusions. Such insights should include an appreciation of the overall degree of uncertainty about the results and an understanding of which sources of uncertainty are critical to those results and which are not. In general, many of the insights gained are not strongly affected by the uncertainties. The numerical results need only be accurate enough to allow the decision maker to distinguish risk-significant elements from those of lesser importance.

The level of detail and the style of presentation of risk results depend on the risk assessment objectives. The results section must communicate the project's motivations and objectives and should be done in a way that clearly establishes the appropriateness of the generated results in meeting the risk assessment objective. For example, if the risk assessment is intended for evaluation of alternative design features as in risk-informed decision making (Section 2.3.1), the results should be presented in a structure that allows comparison of various design options according to an appropriate ranking scheme.

One section of the results should be dedicated to highlighting the key characteristics of the PRA that collectively make the results of the PRA credible. This is important to building a strong Safety Case (Section 3.1.5). Types of information that should be presented include:

- Insights into how various systems interact with one another;
- Insights into the relationship between system operability states and accident scenarios;
- Results of activities undertaken to ensure completeness in the types of events that trigger an accident scenario;
- A very clear and concise tabulation of all known limitations and constraints associated with the analysis;
- A very clear and concise tabulation of all the assumptions used in the PRA especially with respect to mission success criteria and omission of certain failure modes;
- Identification of key parameters that greatly influence the numerical results of the PRA;
- Results of activities undertaken (e.g., sensitivity studies) to ensure that the results of the PRA would not be negated if an alternative parameter value or modeling assumption is employed; and
- Results of activities undertaken to ensure technical quality.

13.1 Graphical and Tabular Expression of Results

In general, graphical and tabular displays are effective means for conveying the results of a risk assessment. The suitability of display designs depends on the PRA objective and the experience of the intended audience. Graphical and tabular displays can be used to report the following types of information generated in a typical PRA:

- Total likelihood of various end states;
- List of dominant risk scenarios and quantitative measure of the likelihood of each scenario;
- The relative ranking of each scenario to the total end state likelihood or total mission risk;
- Estimates of scenario consequences in terms of mission loss, payload damage, damage to property, number of injuries or fatalities, and dollar loss;
- Total mission risk and its comparison with reliability and safety goals and thresholds (Section 3.1.5), if specified;
- Importance measures;
- Display of uncertainties associated with various estimates; and
- Risk curves.

List of Dominant Risk Scenarios and the Likelihood of Each Scenario

The description of each dominant risk scenario along with its likelihood should be provided. The narrative should discuss the nature of initiator and system failures involved in the scenarios. The dominant contributors to each system end state should be presented. A consistent presentation scheme needs to be adapted to systematically delineate the progression of the accident starting from the initiator and all system failures and interactions that are captured in the definition of accident scenario. The method of presentation should permit detailed technical review, including recalculation. The following is an example of a “road map” to report dominant accident scenarios in tabular form (this is only an example, the actual headings will differ):

Example of How To Describe a Dominant Scenario (See Example from Chapter 10):

For scenario number 4 ($IE \cap \bar{A} \cap \bar{B}$) the likelihood is 3.04×10^{-7} per mission. This scenario is initiated by the event IE. The FTs for System A and System B are presented in Figure 12-4. More detailed descriptions of these systems and initiators can be found in Section 12.3.

The major contributing minimal cut sets (MCSs) are summarized in Table 13-1 below.

Table 13-1. An Example of Presenting Dominant Risk Scenarios in a Tabular Form^a.

Initiator	Risk Scenario No.	As Defined in Event Tree	Dominant Cut Sets	
			Cut Set	Frequency
IE	4	$(IE \cap \bar{A} \cap \bar{B})$	IE.ACC.B12	1.2E-7
			IE.ACC.B11	6.0E-8
			IE.A12.A22.B12	3.6E-8
Event	Description		Probability	Basis
IE	Initiating event		1.0E-3	Specify appropriate section(s) of the report
ACC	Common cause shock disables two redundant channels of System A (in standby)		3.0E-3	
B12	Device B1 fails (in standby)		4.0E-2	
B11	Device B1 fails to start once demanded		2.0E-2	
A12	Device A1 fails independent of A2 (in standby)		3.0E-2	
A22	Device A2 fails independent of A1 (in standby)		3.0E-2	

13.2 Communication of Risk Results

As stated earlier, it is important that the degree of uncertainty about the results of quantitative risk analysis be communicated clearly. This means it is incumbent on the analyst to find ways to present the uncertainty associated with risk information in a manner that is understandable to those who need these results. This section presents examples of graphic methods that have been used in PRAs to display uncertainties.

13.2.1 Displaying Epistemic Uncertainties

If the consequence of interest is a single undesired end state that either occurs or not (e.g., failure of a system or loss of a mission), the risk metric is defined as the *frequency of the undesired event* (a non-observable quantity). In this case the epistemic uncertainty associated with the numerical value of the frequency can be displayed using one of three methods:

^a This table is intended for the example presented in Chapter 10 and should not be considered as a general template.

- Probability density function (pdf) – Simulation software codes often generate the results in histogram form, which is the discretized version of the density function. A histogram can be fitted with a continuous curve, (see Figure 13-1, Display A).
- Cumulative distribution function – This represents the integral of the pdf (see Figure 13-1, Display B).
- Displaying selected percentiles as in a Tukey box plot^a (see Figure 13-1, Display C).

13.2.2 Displaying Conditional Epistemic Uncertainties

The PRA analyst may want to show how the epistemic uncertainty of the risk metric varies under certain conditions. For example, he or she may wish to communicate the epistemic uncertainty of risk metric R conditional on the value of parameter X_1 (the parameter is assigned a fixed value). In this case, the analyst displays the uncertainty of the risk metric conditional on the value of the parameter. Several representative values of the parameter of interest may be selected, say, $X_1=p1$ and $X_1=p2$. A separate simulation run is performed for each case. The resultant probability distributions $R | X_1=p1$ and $R | X_1=p2$ are superimposed on a single graph as shown in Figure 13-2. As in the single dimensional case, the distributions may be shown as pdfs, as CDFs, or as “band-aid” plots.

^a Named after John Tukey, an American statistician.

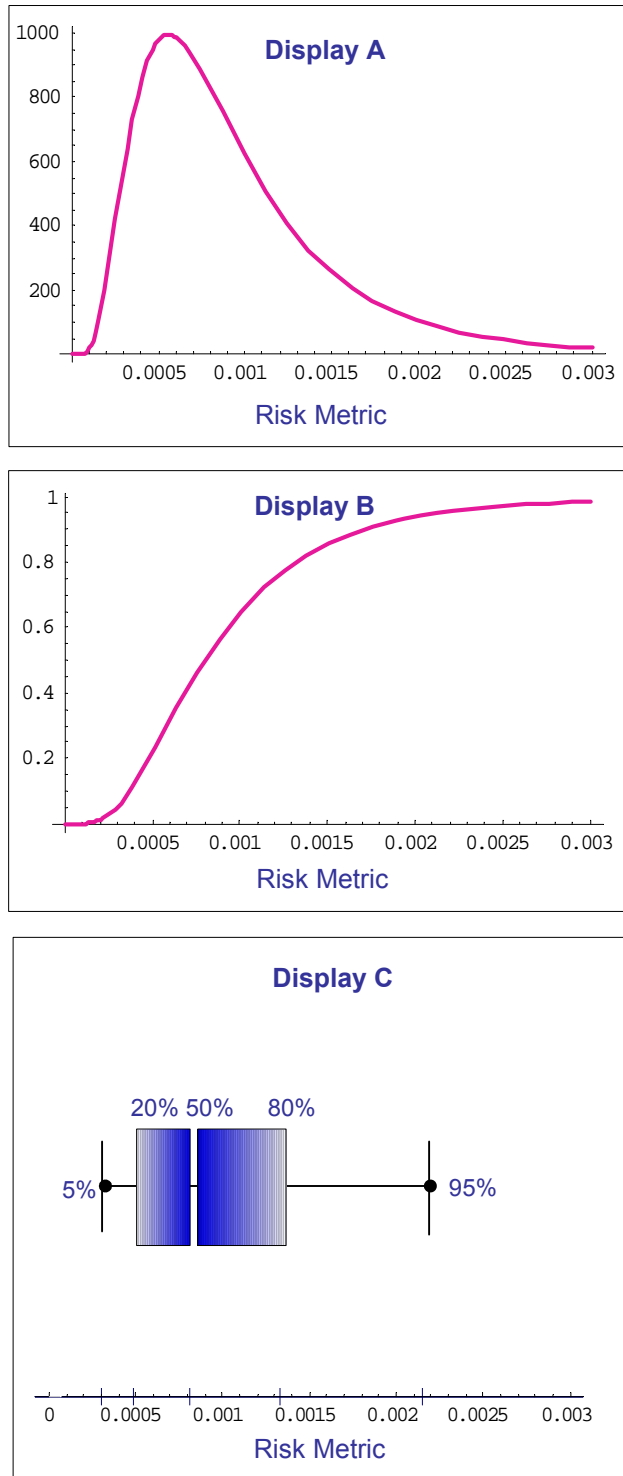


Figure 13-1. Three Displays of an Epistemic Distribution.

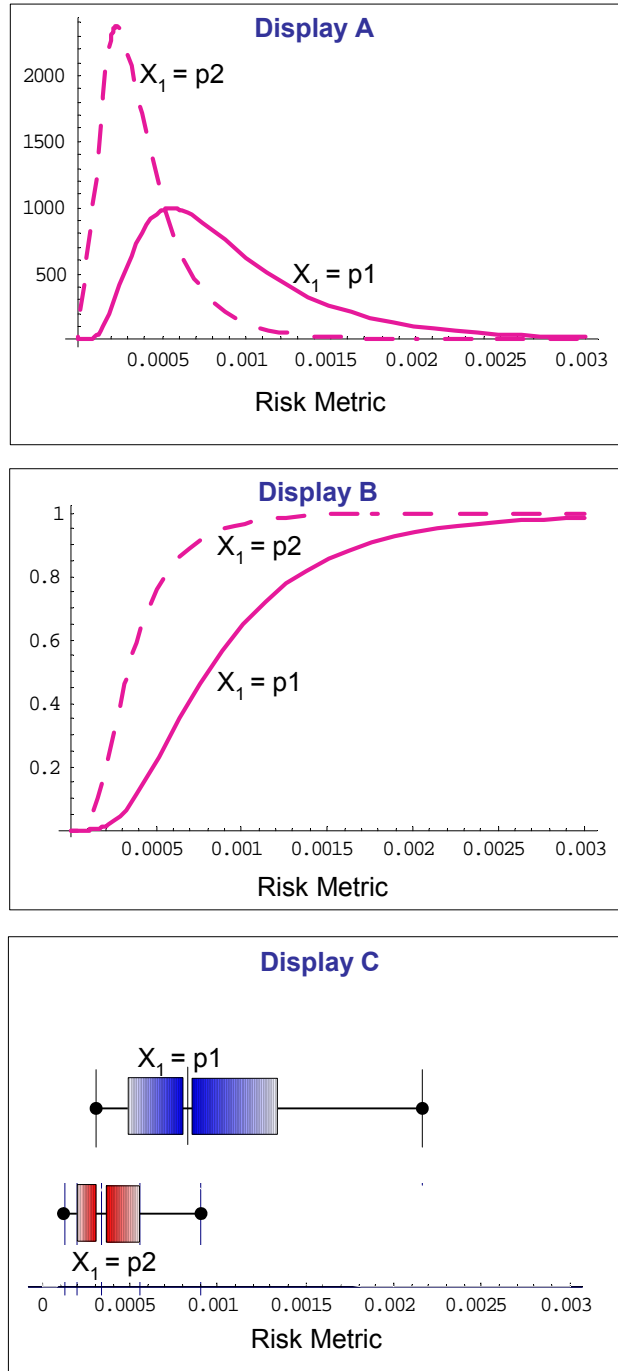


Figure 13-2. Alternative Displays for Conditional Epistemic Distribution.

13.2.3 Displaying Aleatory and Epistemic Uncertainties

If the consequence of interest is a continuous random variable (CRV), such as number of fatalities (an observable quantity), then aleatory and epistemic uncertainty associated with the risk metric is shown using risk curves. An example of such a display is a graph that shows multiple exceedance curves, each of which represents a different confidence level. An exceedance curve provides the frequencies of exceeding a given level of consequence. Since

these curves are often used to communicate uncertainty associated with PRA results, their construction is discussed below.

Construction of Exceedance Curves

The exceedance probability for a given consequence value is the probability of all analyzed accidents whose consequences are greater than or equal to the given consequence value. Figure 13-3 illustrates an exceedance probability versus consequence plot.

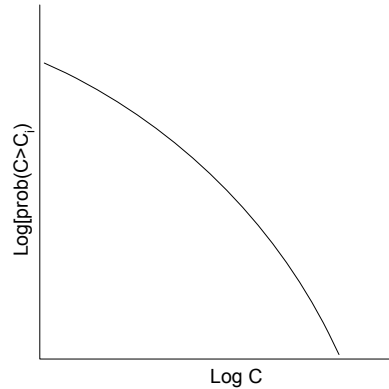


Figure 13-3. A Representative Aleatory Exceedance Curve (Without Consideration of Epistemic Uncertainties).

To obtain an equation for the exceedance probability, let (p_i, C_i) be the probability and consequence of the individual accident scenario, which has been assessed, where the consequences have been ordered by increasing severity (i.e., $C_1 \leq C_2 \leq C_3 \leq \dots \leq C_N$). It is assumed there are N consequence categories (i.e., end states). If P_i is the exceedance probability for a consequence C_i , then

$$P_i = \sum_{k=i}^N p_k \quad (13-1)$$

The individual expressions used to calculate exceedance probability value for various scenarios are shown in Table 13-2.

Table 13-2. List of Scenarios and Exceedance Probabilities.

Scenario	Likelihood	Consequence	$P_i = \sum_{k=i}^N p_k$
S	p	C	
S ₁	p ₁	C ₁	$P_1 = P_2 + p_1$
S ₂	p ₂	C ₂	$P_2 = P_3 + p_2$
S ₃	p ₃	C ₃	.
.	.	.	.
S _i	p _i	C _i	$P_i = P_{i+1} + p_i$
.	.	.	.
S _{N-1}	p _{N-1}	C _{N-1}	$P_{N-1} = P_N + p_{N-1}$
S _N	p _N	C _N	$P_N = p_N$

If we now plot the points (p_i, C_i) , we obtain i discrete points. By drawing a smooth curve through these points and using a logarithmic scale, a probability exceedance curve similar to the curve shown in Figure 13-3 is obtained. Note that when a risk curve is plotted on a log-log scale, it exhibits a concave downward shape. The asymptotes are interpreted as “maximum possible consequence” and “probability of any consequence at all [13-1].”

Example of Constructing an Exceedance Curve

This example will deal with risk to the public. Assume the risk assessment has analyzed seven undesired end states. Also assume the consequence is measured in terms of fatalities. Column 2 in Table 13-3 provides the expected number of fatalities associated with each end state (the values are conditional). The end states are arranged in order of increasing consequence. The point estimates for the frequencies of end states are shown in Column 3. Column 4 gives the exceedance frequency for each consequence. The data in columns 2 and 4 are used to construct the exceedance frequency as shown in Figure 13-4.

Table 13-3. Construction of Exceedance Frequency for the Example Problem.

End State	Consequence (Fatality)	Frequency f	$F_i = \sum_{k=i}^7 f_k$
S ₁	5.0E-03	2.0E-01	1.5E-01+2.0E-01=3.0E-01
S ₂	1.0E-02	1.0E-01	5.1E-02+1.0E-01=1.5E-01
S ₃	5.0E-02	5.0E-02	2.0E-03+5.0E-02=5.1E-02
S ₄	5.0E-01	1.0E-03	1.0E-03+1.0E-03=2.0E-03
S ₅	1.0E+00	1.0E-03	3.0E-05+1.0E-03=1.0E-03
S ₆	3.0E+00	2.0E-05	1.0E-05+2.0E-05=3.0E-05
S ₇	6.0E+00	1.0E-05	1.0E-05

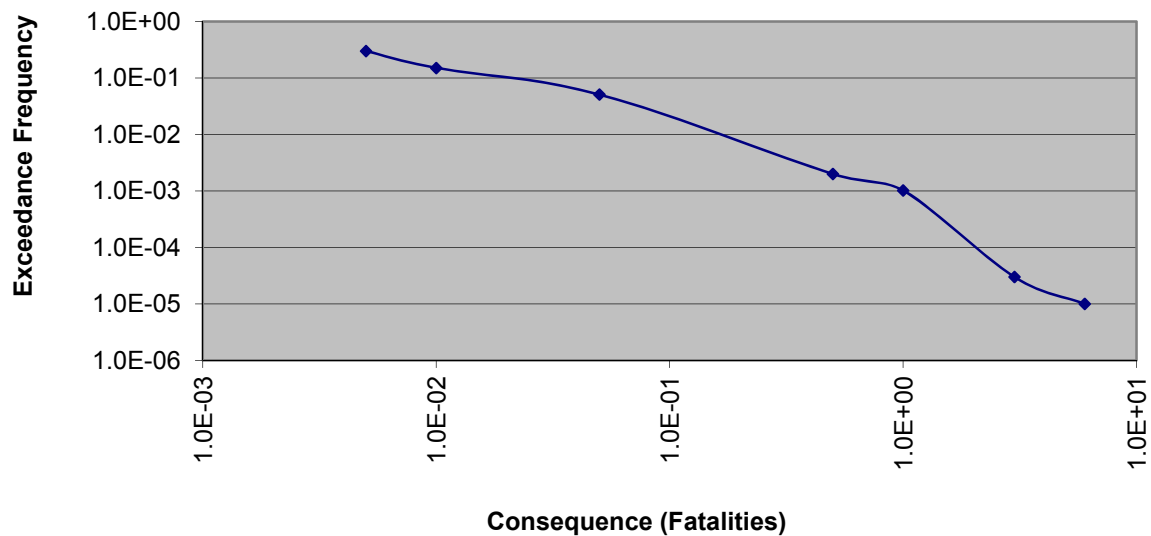


Figure 13-4. Exceedance Frequency versus Consequences for the Example Problem.

Inclusion of Epistemic Uncertainty in Exceedance Curves

If a comprehensive uncertainty analysis were done, it would be possible to produce multiple exceedance curves to reflect different confidence levels. Since both frequency and consequence estimates have epistemic uncertainties, one can produce multi-layered exceedance curves to communicate uncertainty in:

- Frequency estimates;
- Consequence estimates; or
- Frequency and consequence estimates.

Figure 13-5 shows a set of multi-layered exceedance curves developed for a typical space nuclear risk analysis. These curves communicate the uncertainty in the health effects parameters (e.g., cancer fatalities). Each curve represents a level of confidence in the frequency vs. health effects. For example, the curve labeled “95 percentile” reflects an analyst’s view that with 95% confidence the real answer lies on or below that curve. The curve labeled “mean” may be thought of as the “average” confidence level of all possible curves.

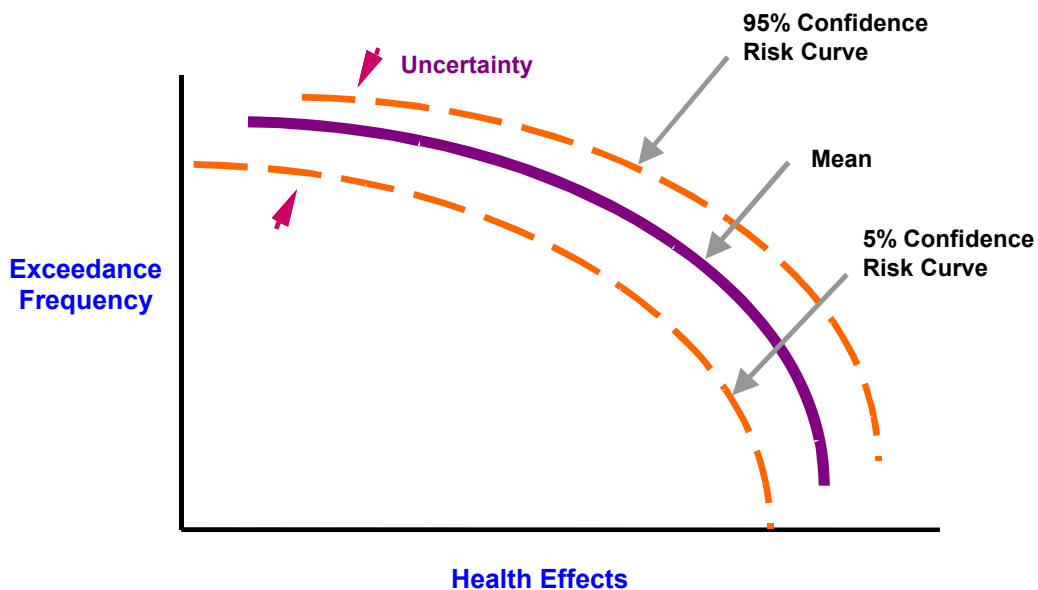


Figure 13-5. Aleatory Exceedance Curves with Epistemic Uncertainties for a Typical Space Nuclear Risk Analysis.

13.3 Importance Ranking

Ranking of risk scenarios based on their frequencies provides limited insight regarding the contribution of individual events such as component failures to the total risk. Scenario ranking provides insights on importance of group of failures, not failure of individual events. An event (say, component x failure) can appear in the structure of many low frequency scenarios, yet it may be absent in the definition of the dominant risk scenarios. If the contribution of low frequency scenarios to the total risk is comparable to that of a few dominant risk scenarios, then scenario ranking will not capture the risk importance of component x. To address this issue, and to provide perspective on importance of individual events or parameters of the PRA model, several quantitative importance measures are calculated. These measures typically determine the change in the quantitative risk metric (e.g., likelihood of a mishap) due to change in probability of an event (or parameter value) in the risk model. Once the importance measures are calculated, the events or parameters of the risk model are ranked according to the relative value of the importance measure. The information generated in the ranking process is often used to support risk-informed decision making (e.g., allocating resources) and to establish guidance for risk mitigation efforts, such as redesign of hardware components, the addition of redundancy, etc.

The following quantitative importance measures are introduced in this section:

- Fussell-Vesely (F-V);
- Risk reduction worth (RRW);
- Birnbaum;
- Risk achievement worth (RAW); and
- Differential.

13.3.1 Importance Measures for Basic Events Only

These importance measures are strictly formulated to assess the sensitivity of the risk metric to changes in the probability of basic events. [13-4] They are designed to handle the importance of basic events when the expression of the risk metric has the following form:

$$R = f(x_1, x_2, \dots, x_i, x_j, \dots, x_n) \quad (13-2)$$

where

$x_i \equiv$ the basic event i , with probability p_i

Fussell-Vesely and Risk Reduction Worth Importance Measures

The F-V importance measure is used to determine the importance of individual MCSs containing basic event x_i to the risk. F-V of event x_i is given by:

$$I_{x_i}^{FV} = \frac{\Pr(\bigcup_j MCS_j^{x_i})}{\Pr(\bigcup_j MCS_j)} = \frac{\Pr(\bigcup_j MCS_j^{x_i})}{R_0} \quad (13-3)$$

Where:

$I_{x_i}^{FV}$ is F-V measure of importance for event x_i ;

$\Pr(\bigcup_j MCS_j^{x_i})$ is probability of the union of the MCTs containing event x_i ; and

$\Pr(\bigcup_j MCS_j) = R_0$ symbolizes baseline expected risk.

The above formulation can be interpreted as the conditional probability that at least one MCS containing event x_i will occur, given that the system has failed. When the expression for the risk is in the sum of the product form, F-V importance is calculated by

$$I_{x_i}^{FV} = \frac{R_0 - R|\Pr(x_i) = 0}{R_0} \quad (13-4)$$

where, $R|\Pr(x_i) = 0$ signifies conditional expected risk when probability of event x_i is set to zero.

The RRW importance is a measure of the change in risk when a basic event (e.g., unavailability of a hardware device) is set to zero. It measures the amount by which risk would decrease if the event would never occur. Mathematically, the RRW measure is calculated by taking the ratio^a of the baseline expected risk to the conditional expected risk when event x_i is set to zero (assuming that the hardware device is "perfect"):

a. Instead of ratio, some PRA codes calculate "Risk Decrease Interval," which is the difference between baseline expected risk to the conditional expected risk when event X_i is set to zero.

$$I_{x_i}^{RRW} = \frac{R_0}{R|\Pr(x_i) = 0} \quad (13-5)$$

where, $I_{x_i}^{RRW}$ is the risk reduction worth for event x_i .

The F-V and RRW measures are related. The right side of Equation (13-4) can be rearranged and be expressed in terms RRW as shown below:

$$I_{x_i}^{FV} = 1 - \frac{R|\Pr(x_i) = 0}{R} \quad (13-6)$$

$$I_{x_i}^{FV} = 1 - \frac{1}{I_{x_i}^{RRW}} \quad (13-7)$$

In practice F-V and RRW measures are used to identify hardware elements that can result in the greatest risk benefit if more resources are allocated to improve their reliability or availability.

Birnbaum Measure (BM) and Risk Achievement Worth (RAW)

The BM is the rate of change of the expected risk as a result of the change in the probability of an individual event. Mathematically, the BM importance of event x_i is

$$I_{x_i}^{BM} = \frac{\partial R}{\partial x_i} \quad (13-8)$$

Because of its formulation, ranking based on BM importance measure does not account for probabilities of events. Highly important but highly reliable hardware equipment (e.g., passive components) exhibit high BM importance measures.

When risk metric has a linear form, the BM can be calculated using the expression below:

$$I_{x_i}^{BM} = (R|\Pr(x_i) = 1) - (R|\Pr(x_i) = 0) \quad (13-9)$$

where $R|\Pr(x_i) = 1$ signifies conditional expected risk when the probability of event x_i is set to unity.

The RAW importance is a measure of the change in risk when the probability of a basic event (e.g. unavailability of a component) is set to unity. Similar to risk reduction worth, the calculation is typically done as a ratio.^a By setting event probability to unity, RAW measures the amount of change in system risk due to assuming the worst case of failing an item.

The RAW measure is calculated using the following expression^b:

a. Similar to RRW, some PRA codes calculate "Risk Increase Interval," which is the difference between the conditional expected risk when event X_i is set to unity and the baseline expected risk.

b. Care should be exercised to ensure that the Boolean expression used to calculate conditional risk is reduced. The RAW is normally calculated by re-quantifying the PRA model with the probability of the given event set to unity.

$$I_{x_i}^{RAW} = \frac{R | \Pr(x_i) = 1}{R_0} \quad (13-10)$$

The RAW measure is useful for assessing which basic events of the risk model are the most crucial for causing the system to have a higher risk. Elements with high RAW are the ones that will have the most impact, should their failure unexpectedly occur.

It can be shown that the BM and RAW measures are also related. By dividing the expression for $I_{x_i}^{BM}$ by the expected risk, R_0 , the following relationship is obtained:

$$\frac{I_{x_i}^{BM}}{R_0} = I_{x_i}^{RAW} - \frac{1}{I_{x_i}^{RRW}} \quad (13-11)$$

The above equation can be rearranged to express BM in terms of RAW and RRW:

$$I_{x_i}^{BM} = R_0 \left[I_{x_i}^{RAW} - \frac{1}{I_{x_i}^{RRW}} \right] \quad (13-12)$$

13.3.2 Differential Importance Measure for Basic Events and Parameters

The importance measures discussed previously are defined to deal with basic event probabilities *one event at a time*. These measures have limitations for use in PRA applications.

- They generally correspond to sensitivity cases in which the basic events values are assigned extreme values (i.e., 0 or 1).
- They are not designed to identify the importance of PRA parameters (they cannot measure the importance of changes that affect component properties or failure modes).
- They do not have additive properties.

Because of these limitations, differential importance measure (DIM) was introduced [13-5].

Definition of DIM

Let R be the risk metric of interest expressed as a function of basic events or fundamental parameters of the PRA model as shown below:

$$R = f(x_1, x_2, \dots, x_i, x_j, \dots, x_n) \quad (13-13)$$

where x_i is the generic parameter such as basic event probability of a component X_i or the failure rate of a component X_i .

The differential importance measure of x_i is defined as

$$I_{x_i}^{DIM} \equiv \frac{dR_{x_i}}{dR} = \frac{\frac{\partial R}{\partial x_i} \cdot dx_i}{\sum_j \frac{\partial R}{\partial x_j} \cdot dx_j} \quad (13-14)$$

DIM reflects the fraction of the total change in R due to a change in parameter x_i .

It can be shown that DIM has the additive property. That is

$$I_{x_i \cup x_j \cup \dots \cup x_k}^{DIM} = I_{x_i}^{DIM} + I_{x_j}^{DIM} + \dots + I_{x_k}^{DIM} \quad (13-15)$$

Calculations of DIM

With respect to calculation of DIM for a parameter of the PRA model, there are two computational difficulties:

- The DIM can be calculated only if the expression for the risk is in parametric form, which is not a standard output form generated by the PRA codes.
- There is no available computer program for use.

However, one can compute DIM for basic events using the F-V and RAW importance measures. The latter measures are often generated by standard PRA codes by applying formulas developed in the previous section on the risk metric that is linear (expressed in disjunctive normal form).

As noted, calculation of DIM deals with change in R (its differential). Since the change depends on how the values assigned to a parameters are varied, DIM is calculated under two different criteria.

- Criterion H1 assumes a uniform change for all parameters (i.e., $\delta x_i = \delta x_j$). Under this criterion, parameters are ranked according to the effect they produce on R when they undergo small changes that are the same for all. This is applicable when parameters of the model have the same dimensions (i.e., the risk metric is expressed in terms of basic event probabilities only). Under the H1 criterion, DIM for parameter x_i is calculated as follows:

$$I_{x_i}^{DIM} = \frac{\frac{\partial R}{\partial x_i} dx_i}{\sum_j \frac{\partial R}{\partial x_j} dx_j} = \frac{\frac{\partial R}{\partial x_i}}{\sum_j \frac{\partial R}{\partial x_j}} \quad (13-16)$$

- Criterion H2 assumes a uniform percentage change for all parameters ($\frac{\delta x_i}{x_i} = \frac{\delta x_j}{x_j} = \omega$.)

Under this criterion, PRA parameters are ranked according to the effect they produce on R when they are changed by the same fraction (ω) from their nominal values. This ranking scheme, which is applicable to all analysis conditions, can be calculated from:

$$I_{x_i}^{DIM} = \frac{\frac{\partial R}{\partial x_i} dx_i}{\sum_j \frac{\partial R}{\partial x_j} dx_j} = \frac{\frac{\partial R}{\partial x_i} \frac{dx_i}{x_i} x_i}{\sum_j \frac{\partial R}{\partial x_j} \frac{dx_j}{x_j} x_j} = \frac{\frac{\partial R}{\partial x_i} x_i}{\sum_j \frac{\partial R}{\partial x_j} x_j} \quad (13-17)$$

The relation between DIM and traditional importance measures (F-V, RAW, and BM) are shown in Table 13-4. These relationships hold only when the risk metric is (1) linear, and (2) expressed in terms of basic events only.

Table 13-4. Relation among DIM and Traditional Importance Measures.

DIM	F-V	RAW	BM
$I_{X_i}^{DIM}$ Under H1	$\frac{I_{x_i}^{FV}}{\Pr(x_i)}$ $\sum_k \frac{I_{x_k}^{F-V}}{\Pr(x_k)}$	$\frac{I_{x_i}^{RAW} - 1}{1 - \Pr(x_i)}$ $\sum_k \left(\frac{I_{x_k}^{RAW} - 1}{1 - \Pr(x_k)} \right)$	$\frac{I_{x_i}^{BM}}{\sum_k I_{x_k}^{BM}}$
$I_{X_i}^{DIM}$ Under H2	$\frac{I_{x_i}^{FV}}{\sum_k I_{x_k}^{FV}}$	$\frac{I_{x_i}^{RAW} - 1}{\frac{1}{\Pr(x_i)} - 1}$ $\sum_k \left(\frac{I_{x_k}^{RAW} - 1}{\frac{1}{\Pr(x_k)} - 1} \right)$	$\frac{I_{x_i}^{BM} \Pr(x_i)}{\sum_k I_{x_k}^{FV} \Pr(x_k)}$

13.3.3 Example of Calculation of Importance Rankings

In this section, the importance measures are obtained for the basic events and parameters of the example problem of Chapter 12.

Ranking of Basic Events

The expression for the risk metric in terms of basic events is as follows:

$$\begin{aligned}
R = & IE.ACC.B11 + IE.ACC.B12 + IE.A12.A22.B12 + \\
& IE.A12.A21.B12 + IE.A12.A22.B11 + IE.A12.A21.B11 + \\
& IE.A11.A22.B12 + IE.A11.A21.B12 + IE.A11.A22.B11 + IE.A11.A21.B11
\end{aligned} \tag{13-18}$$

Substituting the values of the basic event probabilities given in Table 12-1 produces the baseline value for the risk: $R_0=3.07E-7$ per mission. The importance measures at the basic event level are tabulated in Table 13-5. Also shown is DIM at subsystem level (recall DIM has an additive property). Figure 13-6 shows the ranking of basic events with respect to various importance measures. The following observations are made:

- Basic events IE, ACC, B12, and B11 have the highest ranking with respect to all measures.
- At the basic event level F-V ranks individual basic events in the same order as DIM under H2, while RAW rankings are the same as those obtained with DIM under H1.
- Under both H1 and H2, the importance of System A is higher than that of System B.

Table 13-5. Calculation of Importance Measures for the Example Problem.

Importance of Individual Basic Events											
x_i	$\Pr(x_i)$	R_0	$R \Pr(x_i)=1$	$R \Pr(x_i)=0$	$I_{x_i}^{FV}$	$I_{x_i}^{RRW}$	$I_{x_i}^{BM}$	$I_{x_i}^{RAW}$	$I_{x_i}^{DIM}$		
									H1	H2	
A11	1.00E-02	3.00E-07	3.27E-06	2.70E-07	1.00E-01	1.11E+00	3.00E-06	1.09E+01	7.86E-03	2.93E-2	
A12	3.00E-02	3.00E-07	3.21E-06	2.10E-07	3.00E-01	1.43E+00	3.00E-06	1.07E+01	7.86E-03	8.85E-02	
A21	2.00E-02	3.00E-07	2.65E-06	2.52E-07	1.60E-01	1.19E+00	2.40E-06	8.84E+00	6.29E-03	4.69E-02	
A22	3.00E-02	3.00E-07	2.63E-06	2.28E-07	2.40E-01	1.32E+00	2.40E-06	8.76E+00	6.29E-03	7.09E-02	
B11	2.00E-02	3.00E-07	5.20E-06	2.00E-07	3.33E-01	1.50E+00	5.00E-06	1.73E+01	1.31E-02	9.75E-02	
IE	1.00E-03	3.00E-07	3.00E-04	0.00E+00	1.00E+00	Undefined	3.00E-04	1.00E+03	7.89E-01	2.94E-01	
ACC	3.00E-03	3.00E-07	6.01E-05	1.20E-07	6.00E-01	2.50E+00	6.00E-05	2.00E+02	1.56E-01	1.76E-01	
B12	4.00E-02	3.00E-07	5.10E-06	1.00E-07	6.67E-01	3.00E+00	5.00E-06	1.70E+01	1.31E-02	1.97E-01	
Importance of Multiple Basic Events (Selected Cases)											
Subsystem	$x_i \cup x_j \dots \cup x_k$							$I_{x_i \cup x_j \dots \cup x_k}^D$			
	H1	H2									
Train A1	A11+A12							1.57E-02	1.18E-01		
Train A2	A21+A22							1.26E-02	1.18E-01		
System A	A11+A12+A21+A22+ACC							1.84E-01	4.12E-01		
System B	B11+B12							2.63E-02	2.94E-01		

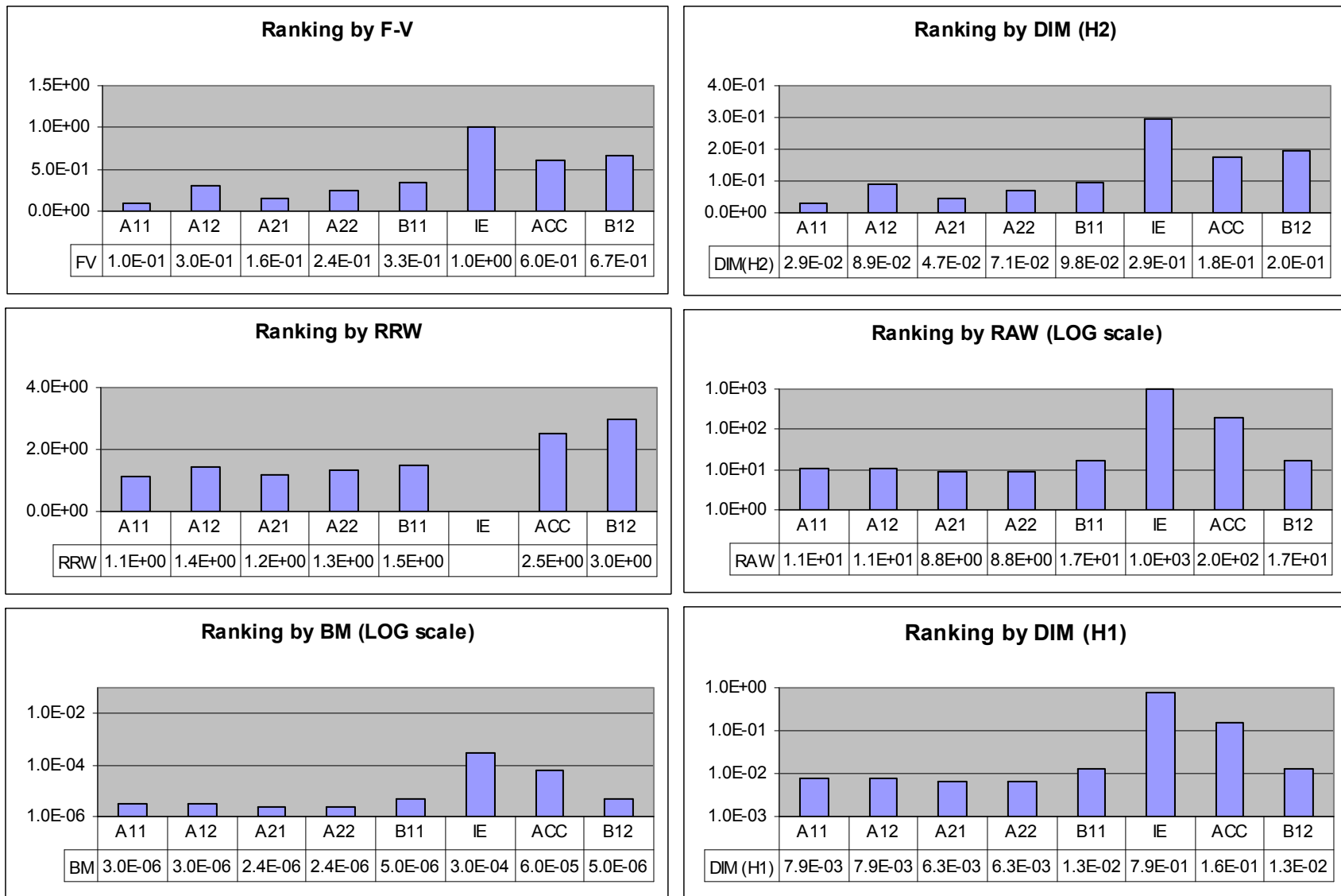


Figure 13-6. Ranking Results for the Basic Events of the Example Problem.

Ranking of Parameters

The risk metric in terms of component parameters is shown below:

$$\begin{aligned}
 R = f_{IE} \cdot & \left(\frac{1}{2} \lambda_{ACC} \tau_A \gamma_{BI} + \frac{1}{4} \lambda_{ACC} \lambda_{BI} \tau_A \tau_B + \frac{1}{8} \lambda_{AI}^2 \lambda_{BI} \tau_A^2 \tau_B + \right. \\
 & \frac{1}{4} \lambda_{AI} \gamma_{A2} \lambda_{BI} \tau_A \tau_B + \frac{1}{4} \lambda_{AI}^2 \tau_A^2 \gamma_{BI} + \frac{1}{2} \lambda_{AI} \tau_A \gamma_{A2} \gamma_{BI} + \\
 & \left. \frac{1}{4} \lambda_{AI} \lambda_{BI} \tau_A \tau_B \gamma_{AI} + \frac{1}{2} \lambda_{BI} \tau_B \gamma_{AI} \gamma_{A2} + \frac{1}{2} \lambda_{AI} \tau_A \gamma_{AI} \gamma_{BI} + \gamma_{AI} \gamma_{A2} \gamma_{BI} \right)
 \end{aligned}
 \tag{13-19}$$

The results of the computation of DIM under H2 (using Equation (13-19)) are shown in Table 13-6 and Figure 13-7. Because the parameters appearing in the expression for R have different dimensions, in this case DIM cannot be generated under the H1 criterion.

Table 13-6. DIM Ranking for the Parameters of the Numerical Example.

PRA Parameter		DIM under H2 Criterion
γ_{A1}		0.0293
γ_{A2}		0.0469
λ_{AI}	λ_{A1I}	0.1595 ^a
	λ_{A2I}	
λ_{ACC}		0.1762
γ_{BI}		0.0975
λ_{BI}		0.1996
f_{IE}		0.2940

a. The failure rates λ_{A1I} and λ_{A2I} are treated as a single parameter because of epistemic dependency.

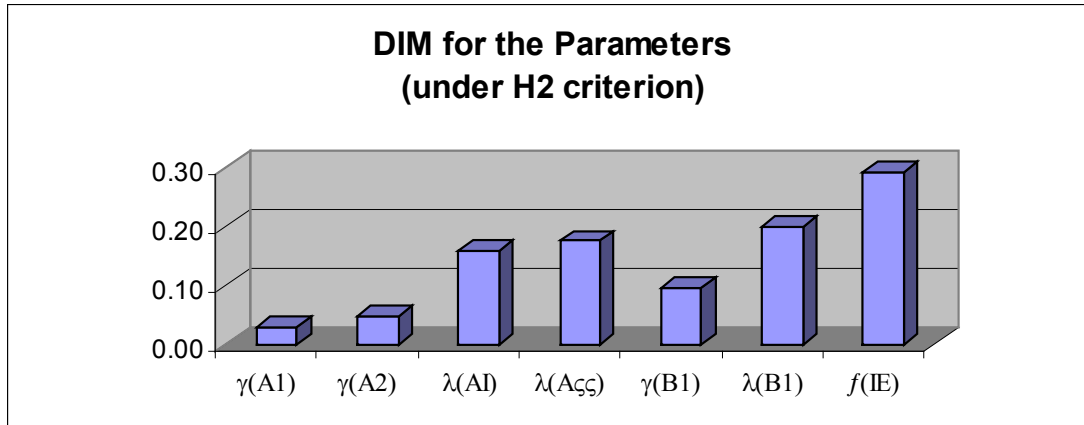


Figure 13-7. Ranking Results for the Parameters of the Example Problem.

13.4 Sensitivity Studies and Testing Impact of Assumptions

As stated earlier, the PRA model is conditional on the validity of its assumptions. The uncertainty associated with the modeling assumptions is usually handled by performing sensitivity studies. These studies are performed to investigate PRA assumptions that are suspected of having a potentially significant impact on the results. Additionally, sensitivity studies are used to assess the sensitivity of PRA results to dependencies among equipment failures.

13.4.1 Impact of Modeling Assumptions

PRA models often use assumptions to overcome data's shortcomings. When information is lacking, heavy reliance is placed on the analyst's judgment. The assumptions made for the mission success requirements and for accident progression can significantly impact the PRA results. The impact of such assumptions needs to be investigated by sensitivity analyses. The results of sensitivity analyses should be reported in tabular form and it should include the original assumption, the alternative assumption and its basis, and the change in the numerical results.

The PRA study of the Galileo mission [13-6] handled uncertainty in the efficacy of the redesign of the solid rocket booster with sensitivity studies. The failure of seals led to the Challenger accident. The extreme cases of a perfect and of a totally ineffective correction were analyzed as bounding cases in the PRA.

13.4.2 Analysis of Impact of Hardware Failure Dependence

Extreme environmental conditions can cause multiple hardware devices to fail simultaneously. (Chapter 10 provides additional information on the treatment of dependency.) Such environmental conditions can be generated either externally to the system by phenomena such as meteoroids; or internally to the system by fires, explosion, etc. Several PRAs have investigated the impact of failure couplings using sensitivity studies as summarized below:

- Examination of risk metric expression (cut sets) to identify dependence-suspect minimal cut sets (DSMCS). DSMCS are minimal cut sets containing failure of components, of which two or more have a common property, which renders them susceptible to dependent failures. DSMCS affected by the following types of coupling:

1. Common environment
 2. Common testing procedure
 3. Common design
- Re-quantification of each DSMCS using the following scheme:
 1. Identify the highest failure probability among the coupled events
 2. Assign the product of the balance of coupled failure probabilities to a high number such as 0.1
 3. Tabulation of results for the risk metric using re-quantified DSMCS one at a time
 - Identification and discussion of DSMCS whose impact on the risk metric is substantially high (say, by a factor of 2 or higher)

13.5 References

- 13-1 NASA/SP-2011-3422, "NASA Risk Management Handbook," November 2011.
- 13-2 NASA-STD-7009, "Standard for Models and Simulations," July 11, 2008.
- 13-3 S. Kaplan and B.J. Garrick, "On the Quantitative Definition of Risk," *Risk Analysis*, 1, 11-37, 1981.
- 13-4 M.C. Check, G.W. Parry, and R.R. Sherry, "Use of Importance Measures in Risk-Informed Regulatory Applications," *Reliability Engineering and System Safety*, 60, 213-226, 1998.
- 13-5 E. Borgonovo and G.E. Apostolakis, "A New Importance Measure for Risk-Informed Decision Making," *Reliability Engineering and System Safety*, 72, 193-212, 2001.
- 13-6 "Independent Assessment of Shuttle Accident Scenario Probabilities for the Galileo Mission," NASA Headquarters, 1989.

14. Launch Abort Models

Please note: While this chapter could for the most part stand on its own, for the purposes of this Guide it should be viewed in the context of specialized modeling that is to be appropriately interfaced and coupled to the overall system risk model discussed in previous chapters.

14.1 Abort Assessment overview

Crew safety remains a primary goal of NASA's continued efforts to implement new means of transporting humans to space. NASA's current human rating requirements call for any new crewed space transportation system to have a crew escape capability [14-1]. Abort capability is crucial because a state-of-the-art booster will be unlikely to meet acceptable safety levels, based on current launcher reliability records [14-2]. The Space Shuttle experience has shown that expecting high reliability from launch vehicles without sufficient analysis is often not justifiable, considering that early ascent flight safety estimates were on the order of one loss for every 100,000 flights, while experience yielded two losses within the first 113 flights. Probabilistic crew safety goals that NASA is considering for any new human spaceflight transportation system acquisition, are more likely to be satisfied by developing effective abort and crew escape capabilities (as well as efforts to reduce the likelihood of failures that can initiate an abort). Figure 14-1 shows an example of loss-of-crew probability versus abort effectiveness rates for different launcher reliability [14-3].

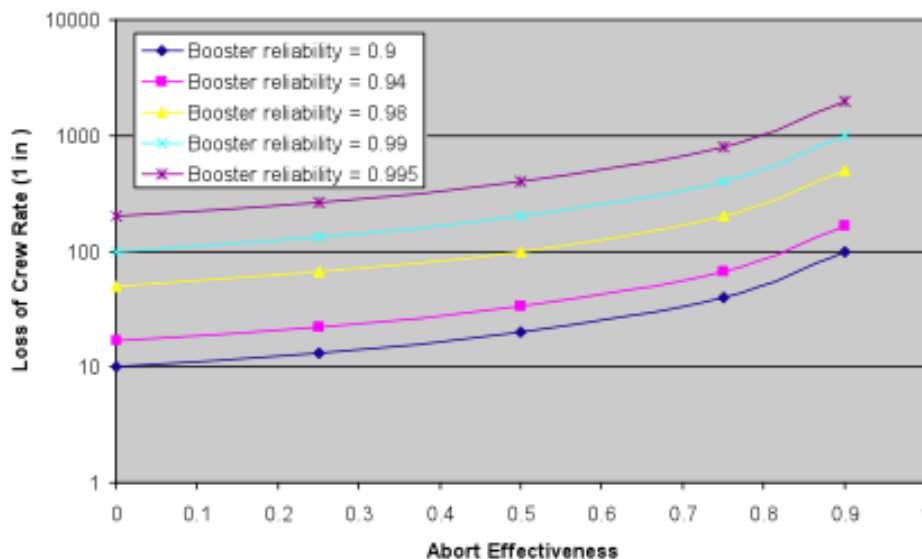


Figure 14-1. Impact of Abort Effectiveness On Crew Risk for Various Booster Reliabilities.

When abort is a cornerstone of crew safety, the ability to meaningfully assess a vehicle's abort capabilities and vulnerabilities is important to understanding overall safety and viability. An effective abort assessment is important both to ensuring that the design meets required safety standards and to designing safer systems up-front by providing actionable insight into key risk drivers and safety tradeoffs as the vehicle is developed.

So, what are the salient attributes of an effective abort assessment? What factors must be considered, what analysis methodologies should be employed, and how should the assessment

be incorporated into the vehicle development process to provide the highest return on information?

This section provides a description of current best practices for assessing abort capabilities. The analysis approach presented extends binary-modeling probabilistic risk assessment techniques through the explicit inclusion of phenomenological models and simulation-based risk models. The abort-modeling process provides an integrated, quantitative estimate of abort risk and provides designers and decision makers with insights into key risk drivers and sensitivities to abort failure.

14.2 Evolution of the Abort Risk Assessment with Program Phases

A successful crew safety risk assessment will inform the vehicle development program of the impact that key design options and decisions will have on crew safety. In this section, we describe the stages of risk assessment and its evolving role throughout the course of a space vehicle program.

The assessment process described here assumes that the program follows a logical series of phases to bring the vehicle from concept studies to initial deployment and, where appropriate, repeated operation. Throughout these phases, the risk assessment is tailored to address the critical questions that arise at each phase or decision point, and evolves as the design develops and matures.

The risk assessment follows a cycle of initial conservatism and successive refinement to produce basic guiding results early and then zero in on the areas that will yield the greatest improvements to crew safety. As more design data become available, the assessment will take on increasingly detailed inputs, focusing on areas with actionable crew safety factors or risk drivers. The assessment should inform the program of the crew safety impacts of requirements, design factors, and operational parameters, particularly in ways that maximize benefit to crew safety for the amount of effort or cost.

NASA's primary vehicle development phases are used here to illustrate how the risk assessment and the questions it addresses evolve through the course of a space systems program. During each of these phases, the decision to proceed is considered in at least one important review.

In the Pre-Phase A period, high-level concepts of the space system and its mission are proposed and compared. At this stage, architectures are compared for mission capability, operational concepts, performance, development effort, cost, feasibility, and safety. At this level, gross attributes of the architectures, such as choice of propulsion technology and arrangement of launch vehicle stages, have great impact on all figures of merit. In support of risk-informed decision making, the decision factors that a risk assessment must address include the relative crew risks of the candidate abort concepts and, perhaps more importantly, the gross attributes driving their risks.

In Phase A, a concept is selected, requirements are formulated, and preliminary system definition and design work are carried out. As for the PRA in general, the abort risk assessment answers questions pertaining to the safety impacts of requirements, system criteria, and high-level design options, especially those that prescribe the performance and features of a launch abort system. For abort modeling, the assessment will typically focus on sensitivity studies and trade analyses that are based on the underlying physics of the most critical failure modes and crew hazards. These studies give insight into how much different variables are likely to impact risk levels, providing a valuable tool to guide and risk-inform decisions regarding the abort system.

In Phase B, the design process is underway, yielding more detailed, component-based design data and failure mode data. Needed technology development is also carried out. The questions addressed by the risk assessment in this phase include identifying the design and technology developments that would best improve crew safety (or least not degrade crew safety in some cases). Again, as for the PRA development in general, the more detailed definition of the abort system in this phase permits the risk assessment to better account for the failure initiators, propagation, and impact to the crew.

In Phase C, the space system design, including the fabrication and integration process, is finalized. Testing of vehicle elements and subsystems also occurs. At this stage, the risk assessment uncertainties are tightened as much as possible with the additional detail in design data and preliminary test data. Where design options are not yet fully finalized, impact on crew safety is still assessed. This can include abort trigger selection and logic or system operation aspects. The risk assessment can also identify any specific new tests or measurements that would best reduce the remaining uncertainties in the crew risk assessment.

In Phase D, the initial system integration occurs, followed by integrated system testing and actual launch tests, so that flight readiness can be assessed. As this phase represents the final opportunity to evaluate crew safety prior to the first crewed flight, it is critical that the risk assessment uncertainties be reduced even further.

With Phase E, the system is put into operation. The risk assessment uses any additional flight data obtained during this phase to inform design, process, and, operational improvements.

14.3 Abort Assessment Process Overview

The general PRA is used to determine the likelihoods of the various ways a vehicle can fail (Chapters 4 through 12). The abort assessment takes these failure “initiators” from the PRA as inputs and evaluates the scenarios and resulting impacts that each one could have on the crew.

Figure 14-2 illustrates the various components of this assessment process, beginning with a specific failure initiation, followed by its propagation to a particular problem manifestation, potential further propagation to other systems, potential development of near-field failure environments capable of directly endangering the crew, the probability of a loss of crew (LOC) given those environments, and finally, the risks associated with the abort process given survival of the near-field environments.

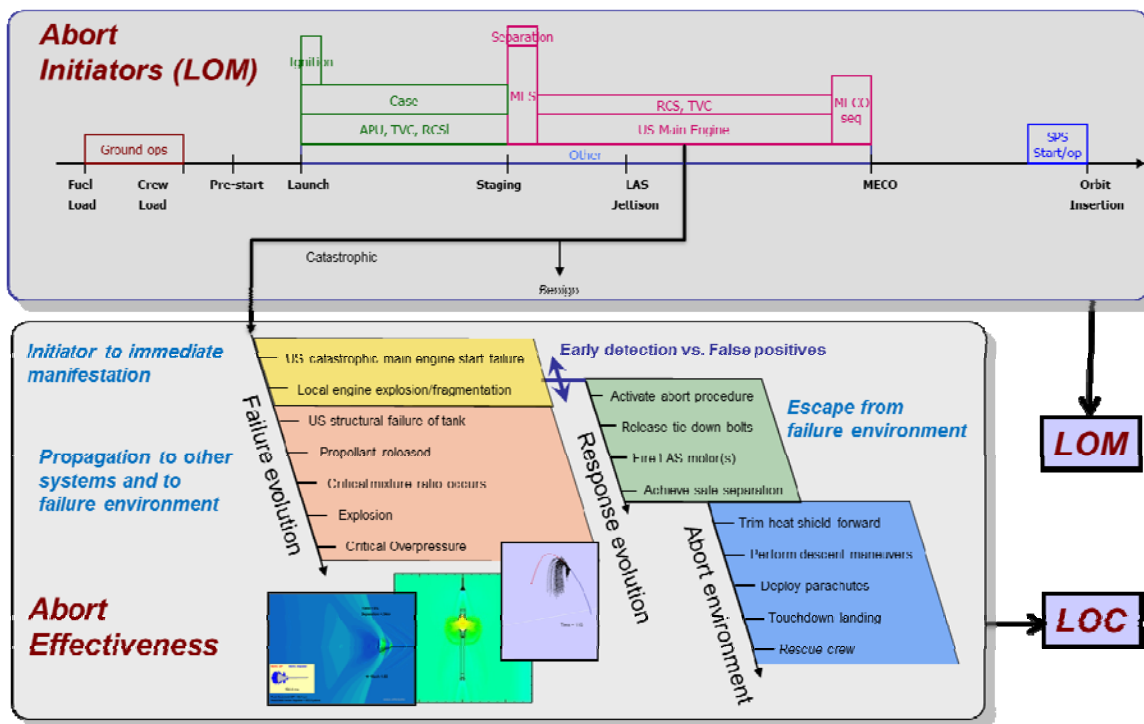


Figure 14-2. Schematic of the Abort Assessment Problem.

Analyzing the failure propagation involves quantifying the relationships between the failure initiators and the ultimate, crew-threatening failure environments. In many cases, a failure can propagate in more than one way. To handle this, the scenario modeling allows for probabilistic branching between the multiple failure evolution scenarios that may stem from a single initiator. In addition, quantifying the timing of the failure evolution and identifying possible methods of detecting it will be critical to evaluating the abort system's ability to escape the resulting failure environments.

To assess the direct crew risks posed by the failure environments, the severities of the potential crew threats are evaluated against the vulnerabilities of the crew escape system. In doing so, the assessment must account for how flight conditions at the time of failure affect the severity and extent of the failure environment and how various abort system parameters affect the vulnerability of the crew abort vehicle. For example, a confined-by-missile (CBM) explosion generates crew threats including blast overpressure waves and high-speed fragment fields. The magnitude of the explosion depends on the amount of fuel remaining at the time of failure, the speed and range of the blast wave and debris depend on the altitude and relative velocities of the vehicle and aborting crew capsule, and the level of overpressure or debris that the abort vehicle can withstand depends on its structural properties and how quickly it can escape the danger zone.

Given survival of the near-field failure environments, the abort system is likely to face a number of remaining risks during descent. For example, a capsule abort system is likely to require reorientation to a heat-shield forward flight attitude, guidance to a suitable location for rescue of the crew, and deployment of parachutes or other deceleration devices. The assessment must quantify the ability to perform these functions, including any dependencies on flight conditions, or mission elapsed time (MET).

Finally, the abort risk assessment integrates all of the components described above to obtain overall loss-of-crew probability and abort effectiveness estimates, and the individual risk contributions from each failure initiator. This integration is typically performed using a Monte Carlo simulation that is effectively coupled with logic-based risk models. The variable, uncertain elements sampled during the simulation may include initiator probabilities, warning time ranges, detection probabilities, branching probabilities for alternate failure propagations, and failure probabilities associated with each specific failure environment.

The overall risk assessment process is highly iterative, evolving to include the effects of any design changes as the vehicle matures. In initially developing the risk scenarios, conservative assumptions are made and bounding cases are assumed to assure coverage of all potential contributions. These assumptions and approximations are then progressively refined as more information becomes available.

In the following sections, the main abort assessment elements summarized above—failure initiators, failure propagation and detection, failure environments, crew escape system capabilities and vulnerabilities, and integrated modeling—are discussed in more detail.

14.4 Abort Failure Initiators

The starting point of the abort analysis is the loss of a critical function required for the mission's successful ascent. Based on the failures identified in the PRA, a list of "failure initiators" is generated, describing each critical vehicle function that could jeopardize the vehicle's ascent if lost and the general state of the launch system at the time of the loss.

To support the abort modeling, the failure initiator information must describe the functional loss in enough detail to identify the component involved, the physical manifestation of the failure, and the severity of the manifestation. These elements are needed to determine the likely progression of off-nominal events that follow. Also required are the probabilities of occurrence for each failure initiator, which are used for Monte Carlo simulation modeling and information about the particular phases of ascent during which each failure initiator could occur. The time of failure is needed to identify the state of the vehicle and ascent environment, and to establish correct initial conditions for any physics-based simulations of the resulting failure environments.

An example of a suitable failure initiator description is lost functionality of structural integrity of a load-bearing upper stage structure while under thrust. This statement identifies the component as the upper stage structure, the lost functionality of structural integrity as the failure mode, and the state of the system as "under thrust." The more information that is provided, including the cause of the loss of functionality, the better the identification of subsequent events and the development of the key conditions associated with the surrounding environment.

Probabilities of occurrence are associated with each failure initiator and uncertainties are modeled with a probability distribution (uncertainty distribution). The probability distributions for the failure initiators are obtained from the general PRA (Chapters 4 through 12).

During the concept design phases, when architecture-level information exists but little detailed design information has been developed, specific knowledge of the failure initiators and the exact mechanisms or physical parameters may not be known. During these early phases, the initial list of failure initiators may cover general characteristics of the failure initiation process or similarities in physical failure outcomes, with the details left as large uncertainties. In the beginning, the failure initiators may be simplified to remove the proximate cause of the failure and list only the effect of the failure on the system as a whole. The initiator list at that stage would consist of a list of functional failures and failure types, such as: loss-of-control failures, explosion failures, heating failures, structural failures, and others. In this case, the probability

distribution for each failure initiator can be derived from proxy sources such as historical launch logs. When proxies are used as a substitute for necessary input data or conditions, the probability distributions must also then include appropriately large levels of uncertainty.

Historical studies of launch vehicles can provide additional baseline failure mechanisms for launch systems that do not involve random failures of components or piece parts. Some of the newer studies have shown that many launch vehicle failures stem from design and process failures that eventually become an ascent problem. These types of failures are typically system-on-system interactions, induced vehicle-environment failures, or process-based errors, while relatively few launch failures are due solely to component or piece part failures. Anchoring the failure initiator set with data from historical studies to define the types of problems commonly encountered in the past is a good approach to developing an understanding of potential system vulnerabilities that can affect the abort system.

A systematic exploration of system vulnerabilities starts by first refining the failure initiators with potential high probability or consequence and/or high uncertainty, and deferring refinement of those with smaller impact until a later time. This approach can give rise to a meaningful design and analysis feedback loop. The initial failure set can expand either in the number of failure initiators considered, or through additional off-nominal information on the existing set of initiators. The failure initiator set can grow because the failure paths themselves are becoming more specific and unique by including additional design information or by removing uncertainty due to less dispersive evolution paths. The failure initiator set can also incorporate additional initial conditions to replace generalized or conservative initial conditions, reducing the uncertainty in the input data. The failure descriptions may become more specific and include information about the timing of events, subsystem geometries and distances, knowledge about flow rates, system constraints, material differences, etc. All of the additional information would enhance the physics-based simulations of the off-nominal conditions immediately following failure initiation.

14.5 Failure Initiator Propagation & Detection

Failure initiators in themselves seldom pose a direct threat to the crew or the abort process. The initiator generates local failure conditions that are exacerbated to the point at which the vehicle state is affected. The local failure conditions generated by the initiator lead to interactions or off-nominal performance beyond the local system boundaries, which eventually result in a direct threat to the crew. The potential for abort failure is therefore strongly dependent on the manner in which the failure progresses to other systems or drives system dynamics. At the same time, an abort will not even commence unless the failure is detected in an abort trigger. Even if a failure is detected, the abort will not be successful if hazardous environments develop more quickly than the abort process. Early detection is desirable, but presents challenges in identifying and verifying a failure early enough to produce a meaningful time margin.

Thus, the three most safety-critical aspects of the failure process are: 1) the severity of the environment or event at the culmination of the propagation, 2) the ability to detect the failure during its evolution, and 3) the speed of the failure's propagation. In this section, a model for failure propagation, abort triggers, and warning times in an abort risk analysis is discussed, and an example is offered to illustrate their roles.

14.5.1 Failure Propagation

Failure propagation describes the progression of a failure from initiator to crew threat. The description may include the intermediate states and final environments to which the initiator may branch, the conditional probability of given branches occurring, the rate of propagation, and the

presence of detectable physical parameters. These features support an integrated abort assessment that accounts for the relative severity and probability of potential intermediate states and subsequent environments.

As for the PRA development in general, the level of detail with which failure propagation is modeled typically begins with higher level modeling and increases with the design and analysis cycles. For instance, in the concept study phase, failure initiators are represented as broad classes of failures that are simply mapped to crew-threatening failure environments (Figure 14-3), allowing one to evaluate the gross sensitivity of the abort system to design parameters and to scope requirements. Because there are generally large design and propagation uncertainties in this early phase, scenarios are developed allowing for worst-case environments and zero propagation times.

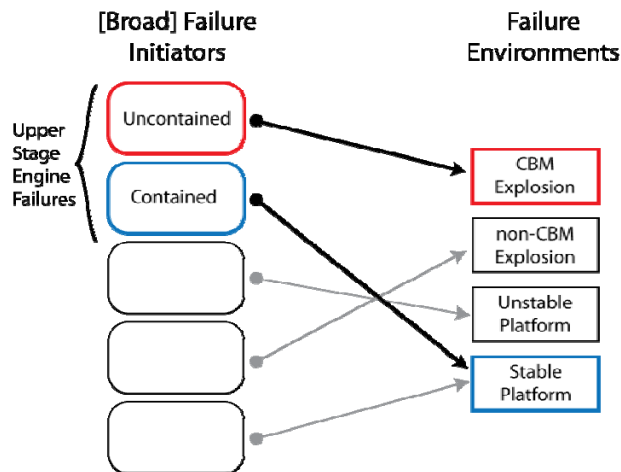


Figure 14-3. Failure Propagation Elements for an Early Concept Crew Risk Model.

With each passing analysis and design cycle, failure propagation mechanisms become better understood and design information becomes more complete. The propagation details can be further refined into a three-stage propagation framework that includes intermediate *failure manifestations* between the previous *failure initiators* and *failure environments* (Figure 14-4).

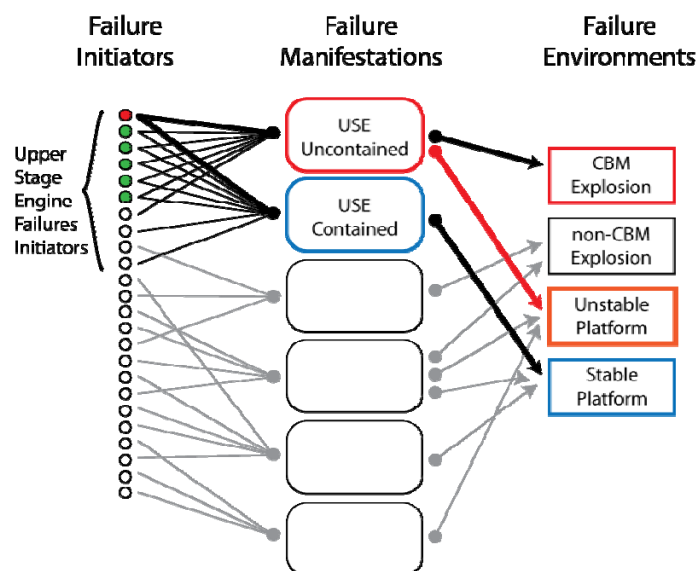


Figure 14-4. Failure Propagation Elements for a More Mature Crew Risk Model.

As the name suggests, a failure manifestation constitutes a critical phase or stage in the failure propagation beyond which the integrity of the mission or vehicle is irrevocably compromised. By allowing propagation paths to branch from the initiator and manifestation stages and converge at the environment stages, the framework efficiently captures a large number of unique failure paths with a relatively small set of common states.

Refinements to the failure propagation model in this context are achieved by adding, when appropriate, more narrowly descriptive failure states (instances of initiators, manifestations, or environments) at each stage. Further refinements come with adding, where applicable, branches between stages to better represent likely outcomes from each failure state. Finally, model refinement is also achieved through better quantification of the likelihood of each state (through initiator and branch probabilities) and the propagation times accumulated in passing between states.

In developing the three-stage failure propagation model, state definition, mapping probabilities, and timing are, again, initially allow for extreme and bounding cases assumptions and then progressively refined. For example, failure manifestations thought to either lead to fast-acting propagation or to have significant potential to interact across system boundaries are initially mapped to worst-case failure environments. Insights from initial analyses and sensitivity studies are then used to identify and focus further refinements to the mappings between failure initiators, manifestations, and environments.

A combination of physics-based analyses, engineering assessments, heritage failure propagation data, and qualitative system analyses can be used to investigate the credibility of the mappings. Physics-based analysis of localized failure phenomena and/or cross-system effects, along with the use of ESDs in certain cases (Chapter 4), can enable quantification of warning times or a better understanding of credible failure manifestations and environments. If significant abort effectiveness improvements are observed as a result of either additional propagation time or re-mapping initiators to alternative failure environments, then further refinements are performed, drawing on physics-based analyses of progressively increasing fidelity.

14.5.2 Failure Detection, Warning Time, and Abort Triggers

Failure detection is a necessary condition for any launch abort. *Warning time* is defined as the failure propagation time from abort initiation to the onset of an immediate threat to the crew. While humans (astronauts or ground crew) may successfully command a manual abort in response to an observable slow-acting failure, automatic failure detection and abort recommendation using an *abort trigger* may be necessary to survive fast-acting, high-consequence failures.

An abort trigger is a conceptual unit containing 1) a physical state, such as a vehicle health parameter, that is subject to anomalies during certain classes of failures, 2) a sensor to monitor the physical state and detect a failure in the event of an anomalous condition, and 3) failure verification and abort initiation logic embedded in avionics software. Abort triggers are intended to induce an automatic abort and reduce the chance of loss of crew by either increasing warning time through early failure detection, or by initiating direct failure mitigation measures.

Developing an understanding of the intermediate physical states through which the launch vehicle's systems pass during various failures allows the effectiveness of abort triggers to be evaluated. Similarly, an understanding of failure propagation paths allows for distinctions to be drawn between the speeds of propagation (fast acting versus slow acting), enabling warning time to be attributed to various failure scenarios.

14.5.3 Failure Propagation and Detection Analysis Example

As an example of failure propagation and detection modeling, consider a bipropellant liquid propulsion system in a crewed launch vehicle. Figure 14-5 shows a sample failure progression for a fuel supply anomaly, following the three-stage failure propagation framework discussed above (failure initiator, manifestation, and environment). The model refinement from a simpler two-stage framework has already occurred. The “fuel supply anomaly” failure initiator may stem from an operational irregularity in the tanks and lines that supply propellant to the engine. The resultant failure manifestations are: “contained” failures that result in an engine shutdown or loss of performance, and “uncontained” failures that propagate beyond the engine system boundaries. Finally, the possible crew-threatening failure environments, from which an abort is attempted, range from stable (but disabled) launch vehicles to detonation-class “confined-by-missile” (CBM) explosions. (The general PRA may only identify the failure manifestations as initiator failures, which is sufficient as long as the more basic initiator or cause of the failure manifestations, shown in the figure as fuel supply anomaly, does not affect the failure environments. If the more basic initiator can affect the failure environments then it needs to be identified.)

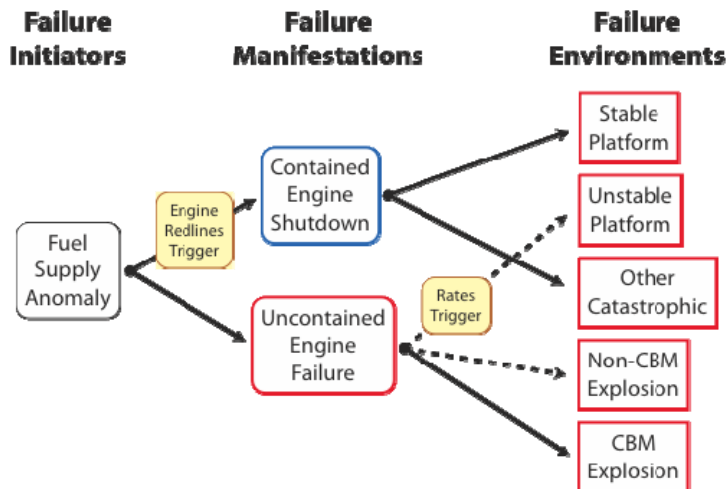


Figure 14-5. Notional Diagram of Engine Failure Progression Through Three Stages of Failure.

The probabilities and warning times allocated to the various failure paths linking the three stages of failure (initiators, manifestations, and environments) are obtained through a more detailed understanding of the intermediate failure propagation and abort trigger processes that occur between each failure stage. For example, the intermediate events occurring in failures due to a fuel supply anomaly might be modeled as shown in Figure 14-6.

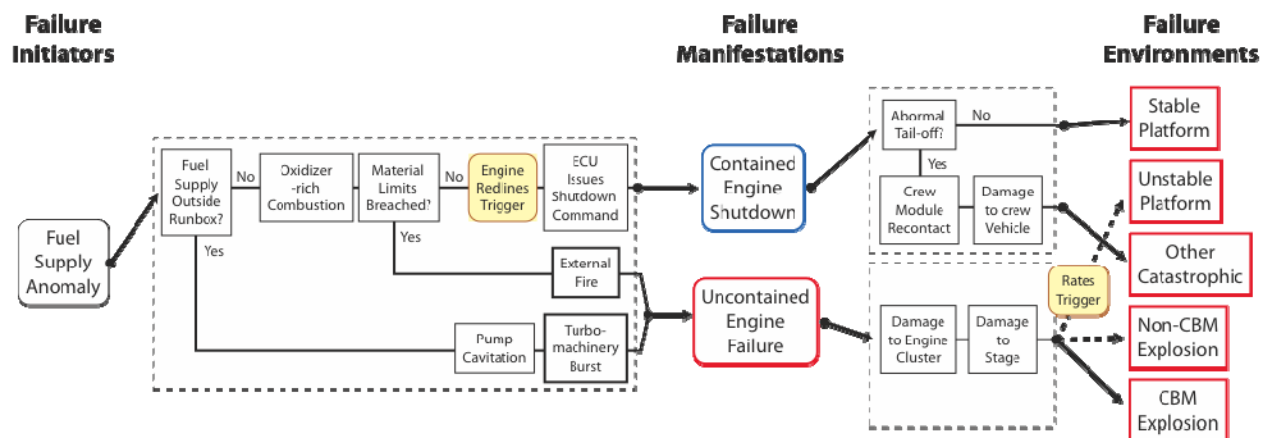


Figure 14-6. Expanded Failure Progression Showing Basis of Failure Path Branching Between Failure Stages. (Dashed arrows indicate mappings added after additional analysis.)

The opportunities available for refining the representation of failure propagation in this example are extensive. Physics-based simulation of the engine's steady-state and transient conditions, along with an understanding of material limits (e.g., melting temperature or burst pressure), can be used to determine the local failure conditions. For example, the off-nominal fuel supply condition results in an intermediate physical state: oxidizer rich combustion in one of the combustion volumes (e.g., gas generator or main combustion chamber). Elsewhere, engine redline sensors, which are mandated for human-rating to initiate engine shutdown under anomalous engine operating conditions, greatly impact the balance between contained and uncontained engine failures through their threshold settings. Engine redline sensors, when used with an abort trigger, could also provide opportunities for gaining warning time ahead of a catastrophic engine failure.

Additional propagation refinements are possible. Detailed failure propagation analysis leads to the identification of a refined set of engine failure manifestations that could result in the uncontained branch of the event sequence. As shown in Figure 14-6, candidates for new failure manifestations might include the events *External Fire* and *Turbomachinery Burst* in place of *Uncontained Engine Failure*. The mappings from each of these new states to the final environments, based on local analysis of each state, would generally be unique, allowing for a richer and perhaps more faithful description of the failure propagation outcomes.

Elsewhere, the relative likelihoods of propagating to environments resulting from an uncontained engine failure can be refined with a physical model of the engine explosion and its interaction with the propellant tanks. The original conservative assumption would map all uncontained engine failures to the worst-case CBM explosion environment. The physical analysis would bound the probability of the worst-case branch by the likelihood of engine shrapnel penetrating the tank walls and intervening propellant. This would establish minimum necessary conditions for a CBM explosion and enable the remaining outcomes to be mapped to less severe environments, as indicated by the dashed mapping lines in Figures 14-5 and 14-6.

14.6 Failure environments

The environment that exists at the culmination of the launch vehicle failure propagation process is called the launch vehicle-generated failure environment. For a given failure initiator and propagation path, loss of crew is prevented by survival of this near-field failure environment and successful return to earth and rescue. Survival in this context is meant to include health of the

abort vehicle life support systems as well as other systems needed to successfully complete the remainder of the abort process. Survival of the failure environment requires that either the launch vehicle is not capable of generating a threatening environment at the position of the crew on the stack, or that the crew module is given time and method of propulsion sufficient to escape the environment or that the crew module is sufficiently robust that it can protect the crew from the threatening environment. In order to quantify the probability of failure caused by these near-field environments, the intensity of the environment and its decay with distance must be quantified. In addition, the variation of these characteristics with vehicle flight condition must be represented.

Examples of launch vehicle-generated failure environments include blast overpressure, explosive fragments, fireball, and loss of control. The first three of these are likely to be simultaneously present following a confined-by-missile explosion of a liquid propellant booster.

14.6.1 Explosion Environments

The primary concern with overpressure and fragments is the possibility of damaging the crew module and losing the integrity of the pressurized vessel. Fireball is not typically considered a significant threat to the crew module itself, but the threat of resulting radiant heating to a parachute system deployed after a pad abort must be evaluated. The intensity of explosive environments is known to be strongly dependent on the “blast yield,” or energy of the explosion, often expressed in terms of the equivalent mass of TNT. This factor is dependent on the degree of vaporization and mixing of the liquid propellants prior to ignition. As such, yield tends to be dependent on the detailed specifics of a given failure scenario and generally should be treated as highly uncertain. Early assessments may use a fixed ratio of TNT mass to propellant mass, but better estimates—based on volumes available for mixing, pressures, etc—can be obtained when detailed design information is available.

Although not likely to involve chemical reaction, solid booster case bursts also produce explosive environments including both blast overpressure and fragment fields. The overpressure is considered to be of the type generated by a bursting pressure vessel with a relatively low yield. Fragment fields, on the other hand, are believed to be a significant, far ranging threat due to the internal pressures released and the mechanical properties of the solid propellant itself.

14.6.2 Blast Overpressure

The intensity of the blast overpressure environment is typically measured in terms of two parameters: the peak value of the overpressure and the time-integration of the overpressure distribution, i.e., the impulse. The blast environment intensity is strongly dependent on the speed with which the available, vaporized fuel is burned, i.e., the flame front speed. Supersonic flame fronts (detonation) produce much more intense overpressure environments compared to subsonic flame fronts (deflagration). Scenarios that lead to internal propellant mixing should be considered potentially capable of detonation, with attendant severe overpressure and fragment environments. Unconfined propellant release is unlikely to lead to detonation, except on the pad where the ground and surrounding structures may provide sufficient confinement and congestion to support detonation.

Overpressure environments can be propagated to determine the spatial extent of the danger using either TNT or vapor cloud explosion (VCE) characteristics. Vehicle velocity affects the propagation by impeding the progress of the blast relative to what would be observed in quiescent air. This effect has a significant effect on the separation distance required of the abort system. Interaction of the blast with the abort vehicle’s heat-shield can be modeled assuming

simple shock reflection theory, but improved representation is obtained by performing computational fluid dynamics (CFD) simulations of the interaction of the blast with the abort vehicle outer mold line (Figure 14-7).

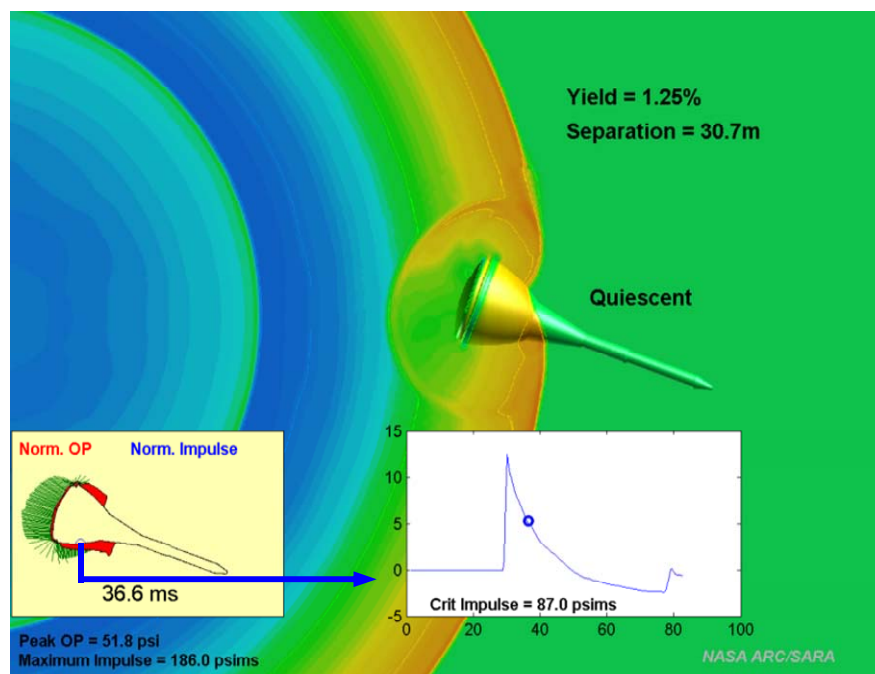


Figure 14-7. Sample Simulation of Blast Overpressure Wave Passing Over Crew Module.

Outputs of the overpressure analysis might include peak overpressure and impulse as a function of distance for various mission elapsed times. Interaction effects also need to be included in these distributions. Specifically, the pressures produced must be consistent with the way the abort system vulnerability is expressed. For example, if vulnerability is expressed in terms of heat-shield-reflected peak overpressure and impulse, then the overpressure environments need to be represented by tables of reflected impulse and peak overpressure.

14.6.3 Fragments

Fragment field environments are strongly dependent on the initial distributions of fragment mass and imparted relative velocity, which are captured in the vehicle's debris catalogs. Vehicle-specific debris catalogs are rarely available early in the design process and initial assessments will require identification of suitable surrogate catalogs from existing vehicles. Initially, catalogs may be assumed to be independent of mission time and blast yield, but potential dependencies should be considered as design information becomes available, especially in the case of solid propellant boosters. A critical factor in the development of debris catalogs is the likely pattern of the debris' outward directions, e.g., surface normal vs. spherically uniform vs. a combination.

Given an initial debris catalog and flight conditions, the fragments may be propagated using simplified (i.e., 3-degree-of-freedom) trajectory dynamics coupled with some randomization process. Monte Carlo methods may be practical if the number of debris pieces is relatively small. Coupling the debris trajectories with the abort system trajectories provides an estimate of the strike probability during the abort. As with the overpressure, the fragment propagation is strongly dependent on flight conditions because of the effect of dynamic pressure on the

fragment drag. The output of the fragment field analysis would be something like debris density as a function of distance for various mission elapsed times.

14.6.4 Fireball

The fireball environment results from a relatively slow burn of propellant, which leads to high local temperatures within the fireball itself and radiated heating for many fireball diameters outward. As previously mentioned, this radiated heat is not likely to pose a threat to metallic components, but parachute designs and/or abort system range must be developed in consideration of the fireball environment. Simple engineering correlations are available for fireball diameter and duration. These can be coupled with point source approximations for the radiant environment decay to produce an initial thermal environment as a function of distance. The initialization of the environment can be improved with energy-volume computational methods, which provide better estimates of size and, especially, duration.

14.7 Loss-of-Control Environments

Loss-of-control environments may arise from failure initiators such as thrust vector control lock-in-place or fail-hard-over, as well as from solid booster case breach failures. Loss-of-control failure environments differ in nature from explosive environments in that they do not spatially extend from the booster. Rather, their hazard is experienced through their effect on the attitude and rate conditions provided to the abort system at abort initiation. Such conditions can exceed structural limits of the abort vehicle or be difficult for the control system to stabilize following separation. As with the explosive environments, however, the conditions generated and the vulnerability of the crewed compartment to those conditions are strongly dependent on flight conditions, especially dynamic pressure and Mach number.

Early assessments may make use of simplified analyses of bounding cases, chosen based on engineering judgment. The data required include: a nominal ascent trajectory, estimates of mass and inertia properties, gimbal authority limits, abort trigger thresholds, and vehicle failure limits (e.g., for structural breakup). In early design phases, many of these parameters will not be defined. In fact, given the expected sensitivity of crew safety to these parameters, the safety assessment should play an important role in their specification.

With design maturity, detailed high-fidelity vehicle dynamics are used to assess the response of the vehicle to various failure modes, including failures of the control system as well as other failures with potential to generate off-nominal forces and moments.

Output from the loss-of-control analyses would include distributions of potential attitude and rate conditions that can be used to initialize abort trajectory simulations. In addition, the amount of warning time between reaching the trigger threshold and reaching the failure limit is needed to assess the ability to escape potential subsequent explosions.

14.7.1 Example of Failure Environment Quantification: Blast Overpressure

This section presents an example of how to quantify the risks posed by a particular failure environment. The example presented here is based on an analysis of blast overpressure environments developed for the Ares I Ascent Abort Risk Assessment, and represents the level of analysis at the Preliminary Design Review (PDR) phase.

In order to apply the blast model pressure propagation information in a risk assessment, the blast overpressure and trajectory must be converted into failure probabilities. Simple engineering-level models are used in this case to model the blast overpressure environment and predict the risk to the crew. These models also provide additional data and insight into the

design parameters that critically impact the ability to survive this failure mode. The components and inputs required to analyze the blast overpressure failure mode for this case are shown schematically in Figure 14-8.

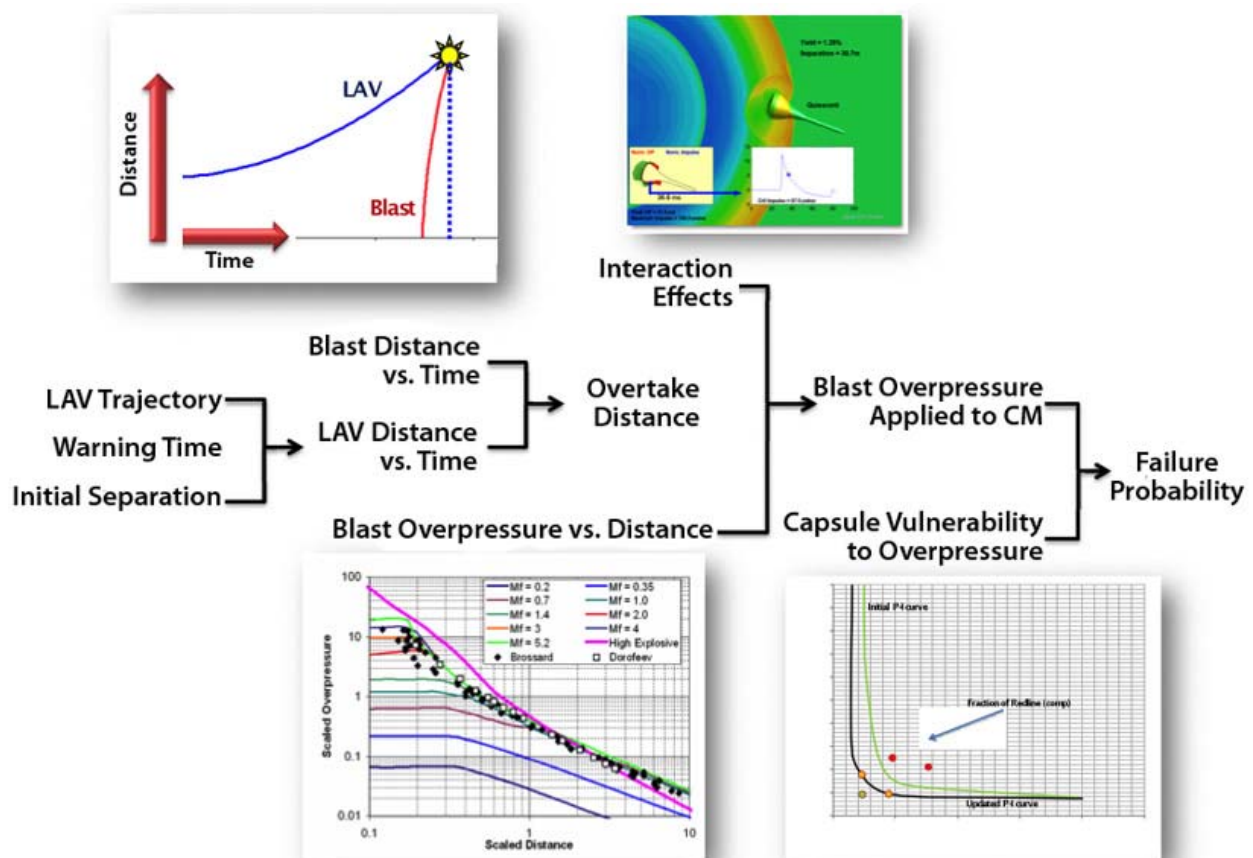


Figure 14-8. Schematic of the Components and Inputs in the Blast Overpressure Analysis.

The model contains a series of component models for explosive blast overpressure propagation and abort vehicle trajectory to determine the overpressure experienced by the escaping crew module. This overpressure is then compared with the capsule's vulnerability to blast overpressure to determine the failure status. Both the blast and abort vehicle propagation modules depend on multiple sub-components and inputs, including blast energy, flight conditions, etc.

This model is then used to produce failure data in the form of tables of failure probability as functions of mission elapsed time (MET) and available warning time. Given a sufficiently efficient implementation, the model could also be used to perform sensitivity analyses with respect to critical design parameters (e.g., abort vehicle thrust profile, crew module overpressure vulnerability, etc.) to inform the development of requirements.

Using the results of the integrated safety assessment, the component models can be enhanced to better reflect the physics involved. For example, a two-parameter Vapor Cloud Explosion (VCE) model was determined to be a closer representation of the liquid propellant explosion than a single-parameter, high explosive model. A simple peak overpressure limit only considers

one crew module failure mode for the blast wave. Using a two-dimensional pressure-impulse (P-I) curve includes a second failure mode involving both the magnitude of the overpressure and the duration of the wave.

The model described in this example clearly extends beyond simply characterizing the failure environment to include abort trajectories and crew module vulnerability. Performing integration at this level may provide significant efficiency benefits relative to performing this integration as part of the overall integrated safety simulation process.

Step-by-step process used to generate failure probability due to blast overpressure in this example:

- 1) Specify blast size – energy (fraction of fuel, pressurized cavity, etc)
- 2) Specify initial blast rate – flame Mach (> 1 for detonation/CBM; < 1 for deflagration/non-CBM)
- 3) Interpolate in Baker, Strehlow, Tang blast families to produce non-dimensional blast characteristic for specified flame Mach (assuming not one of original series)
- 4) Dimensionalize the resulting curve for specific conditions (altitude) and blast energy using Sachs scaling.
- 5) Use relationships between blast overpressure and Mach to integrate a blast trajectory (distance vs. time) for a blast in quiescent air.
- 6) Represent effects of vehicle velocity on the blast overpressure and trajectory (e.g., using a model developed with help of CFD simulations).
- 7) Represent effects of the shock's interaction with the crew module. In the simplest case, for example, a shock reflection relation may be used to account for reflection of the shock off the crew module's heat shield.

At this point, a blast overpressure environment has been generated in the form of a table of peak overpressure, blast impulse, and arrival time as functions of distance from the blast center. To produce a failure probability for a specific situation, the following further steps are used for this case:

- 8) Produce/obtain a trajectory of the abort vehicle for the specific conditions under consideration (altitude, velocity, etc.).
- 9) Adjust the trajectory time and distance to account for warning time and the initial location of the abort vehicle relative to the blast center.
- 10) Determine the point of intersection between the blast (distance vs. time of arrival) and the abort vehicle, i.e., the distance/direction at which the blast overtakes the abort vehicle (if it does).
- 11) Assess overpressure characteristics at the point of intersection (peak overpressure and integrated impulse, including effects of interaction).
- 12) Locate the resulting blast threat as a point specified by the post-interaction impulse and peak overpressure in blast impulse/peak overpressure (P-I) space.
- 13) Assign failure probability based on the relative position of the blast characteristics in P-I space relative to the failure or vulnerability curve of the crew module. Above and/or to the right of the curve represents conditions exceeding the failure criteria (failure probability = 1) and points below and/or to the left indicate conditions that can be withstood by the structure (failure probability = 0). Continuous probabilities between 0

and 1 may be obtained by representing uncertainties in the location of the failure curve, etc. In order to maintain consistency, it is important that this curve be defined with respect to the post-interaction (e.g., reflected) values of the blast characteristics rather than incident values.

14.8 Crew module Capability & Vulnerability

An abort assessment requires data about the crew module and abort system to establish both failure criteria and the severity of the potential failure environments. The design of the crew module will determine, in part, the loss-of-crew failure criteria. For example, the loss of crew due to blast overpressure will depend upon the strength of the crew compartment and its ability to withstand rupturing. Likewise, the seat design will determine the magnitude of crew module acceleration and deceleration that can occur without fatal injuries.

The capabilities of the abort system and crew module are also factors in the environments that the crew will face during an abort. For example, the performance of the abort motor will determine the separation distance between the crew module and launch vehicle during an abort. Similarly, the aerodynamics of the crew module will determine the descent trajectories that can be flown to return the crew to earth, and the descent and landing system will determine the level of risk associated with landing the crew module onto land or water.

The level of detail required depends upon the scope and goal of the assessment and the state of the design. An abort assessment can be used to provide information for design trades during conceptual design, to provide risk contributions of sub-systems during preliminary design, and to provide overall risk of a complete crew module and abort system. An abort assessment during the conceptual design phase will typically focus on sensitivity studies of system design parameters and their effects on crew risk. Designs at this stage are not complete and not all information required for the assessment will be available. In such cases, historical data and conservative assumptions can be used to fill in any data holes. These early assessments can be used to evaluate different design concepts (e.g., top-mounted or side-mounted, liquid or solid stages), to evaluate the effect of a range of system parameter values (e.g., launch abort motor thrust, crew module structural strength) on crew risk, or to highlight sub-systems that present the most risk for design trade studies.

As the design matures, more data specific to the design can be integrated into the assessment. For example, during the conceptual design phase, aerodynamic data needed to compute the descent trajectory would be obtained from existing vehicles of similar shape or computed using simple, generic shapes that approximated the proposed design. Once the outer mold line of the crew module is established, aerodynamic data can be computed based on the specific design. The descent trajectories computed using these new data would better reflect the performance of the crew module design. As more design information gets incorporated into the assessment, the more specific the assessment becomes.

However, it may not be necessary to use all of the design data available. Even with a mature design, it may be insightful to conduct a quick initial assessment of the abort system using low-fidelity models. A low-fidelity assessment could provide a roadmap for further analysis by highlighting sub-systems that contribute the most risk to the crew. For example, an initial analysis of a given design may point to the landing phase as contributing the most loss-of-crew risk. This may be due to assumptions about the reliability of the parachute system, the capabilities of the crew seats, or limitations on where the crew module can land. A second pass of the assessment would refine the assumptions about the parachute reliability and conduct a more detailed analysis of the ability of the crew seats to absorb landing loads. If the crew

module design limitations on where it can land continue to be a large risk driver, this information can be used as part of a follow-up evaluation of that design decision.

14.8.1 Example of Crew Module Vulnerability Assessment

As an example, consider a crew module to be designed for an asteroid mission. For an abort assessment during the conceptual design phase, minimal information about the crew module would be available. At this point, historical data and conservative assumptions would be used to estimate the ability of the crew module to withstand the blast overpressure and debris strikes arising from the destruction of the launch vehicle during an ascent abort. Maximum survivable blast overpressure levels would initially be based on historical data of similar crew module designs. A model used to predict the probability of debris striking the crew module requires as input the characteristics of the launch vehicle debris (e.g., mass, area, imparted velocity) and the abort trajectory of the crew module. The relative position of the debris and the crew module along its abort trajectory is computed and used to determine if the debris can strike the crew module. A conservative assumption would be that any piece of debris striking the crew module, regardless of size and relative velocity, would damage the crew module sufficiently to cause a loss of crew. An example of the debris strike probability, as a function of abort time during ascent, is shown in Figure 14-9.

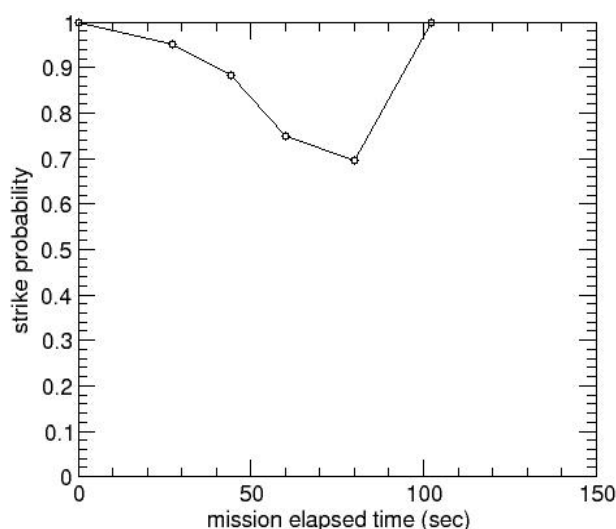


Figure 14-9. Example of Debris Strike Probability as a Function of Abort Time During Ascent.

As the crew module design progresses and information about the structural design becomes available, the failure criteria can be updated. The maximum blast overpressure can be determined using the maximum design strength of the crew module instead of an estimate based on historical data. For debris vulnerability, an analysis can be conducted to determine the impact velocity required for a piece of debris of a given mass to penetrate the crew module's exterior skin. Such an analysis would yield a relationship between the debris mass and the impact velocity required to penetrate the crew module skin, as shown in Figure 14-10.

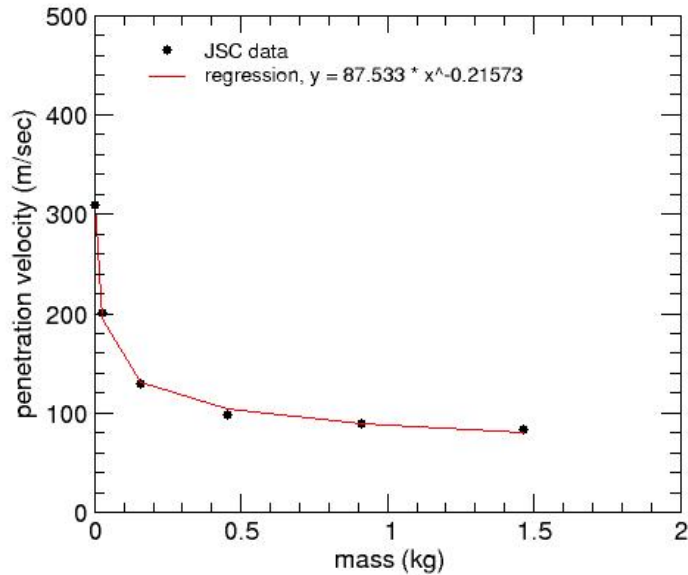


Figure 14-10. Debris Mass and Impact Velocity Required to Penetrate the Crew Module Skin.

Within the debris strike probability model, this penetration criterion would be incorporated by checking whether each piece of debris found to be within striking distance of the crew module has a sufficient impact velocity for its mass. This would eliminate pieces from the debris count, reducing the strike probability by reducing the number of debris pieces that could lead to loss of crew. The resulting reduction in strike probability is shown by the dashed line in Figure 14-11.

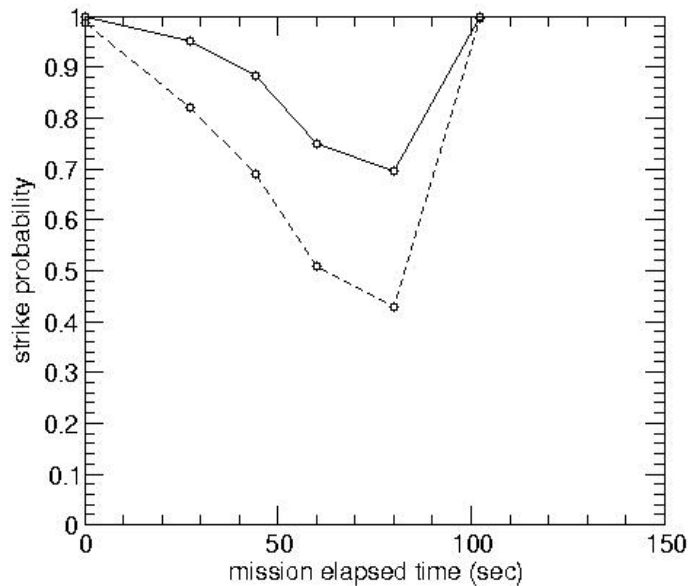


Figure 14-11. Example of Reduction in Debris Strike Probability Due to Imposing Penetration Criteria.

As the designs for both the crew module and launch vehicle evolve and the failure environments become better defined, the failure criteria can become more specific. A general failure criterion

based only the peak overpressure of the blast wave can be replaced by a failure of a structural component obtained from structural analysis of the design. As detailed structural models of the design become available, the structural response of the entire crew module, including the abort hardware, to potential blast waves can be computed. An example is shown in the Figure 14-12.

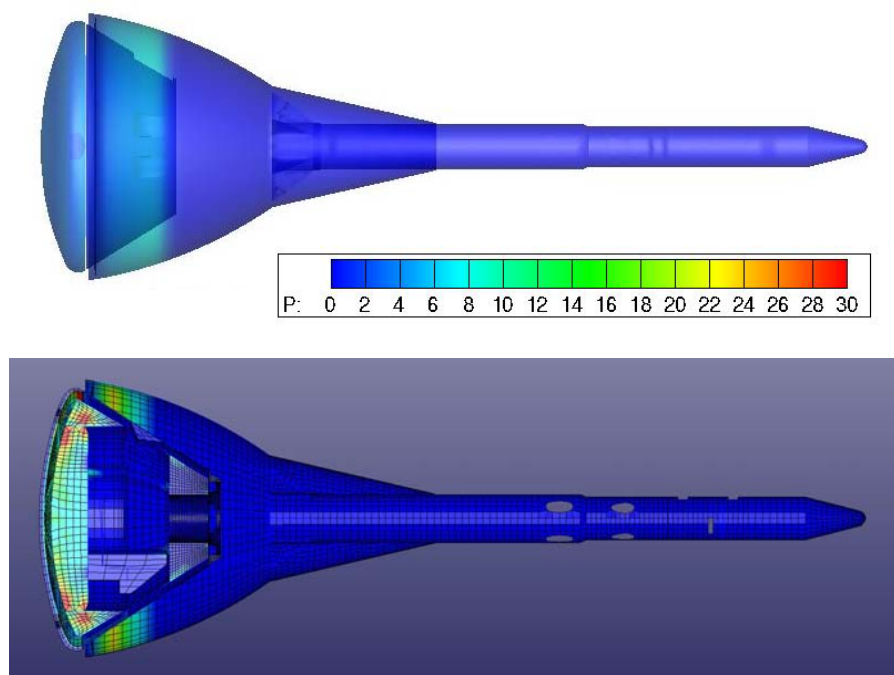


Figure 14-12. Example of Detailed Structural Response Computed for a Crew Module.

Such an analysis evaluates the effect of both peak overpressure and impulse on structural integrity and can highlight potential weak areas that could lead to structural or sub-system failure resulting in loss of crew. Debris penetration criteria could include such parameters as impact angle and debris material properties, and could account for the different wall thicknesses and materials in the crew module design.

Each refinement should provide a better estimate of the impact the vehicle design has on the risks associated with an ascent abort. As the design matures, the structural models can be used as part of the assessment. The strength of the structural design can be used to determine the maximum blast overpressure that the structure can withstand. The wall thickness and design can be used to determine the impact velocity a given piece of debris must have in order to penetrate the crew module. In turn, the results obtained from the risk assessment can be used to refine the design in order to reduce the risk to acceptable levels.

14.9 Integrated Abort Modeling

All of the abort assessment elements discussed in the preceding sections—failure initiators, failure propagation and detection mechanisms, failure environments, and crew module capabilities—need to be integrated into a single stochastic model. Such an integration model links the separate abort assessment pieces with their inputs and supports the assessment of LOM and LOC, as well as any other figures of merit defined for each phase of the project. The model must support sensitivity analyses needed to inform the designer and managers about the relative importance of the modeling parameters and their acceptable ranges. The model also needs to answer questions regarding uncertainty in the outputs or uncertainty added by each

additional step of the process. Since answering these questions requires a great deal of processing power, the integration model must also be light and flexible.

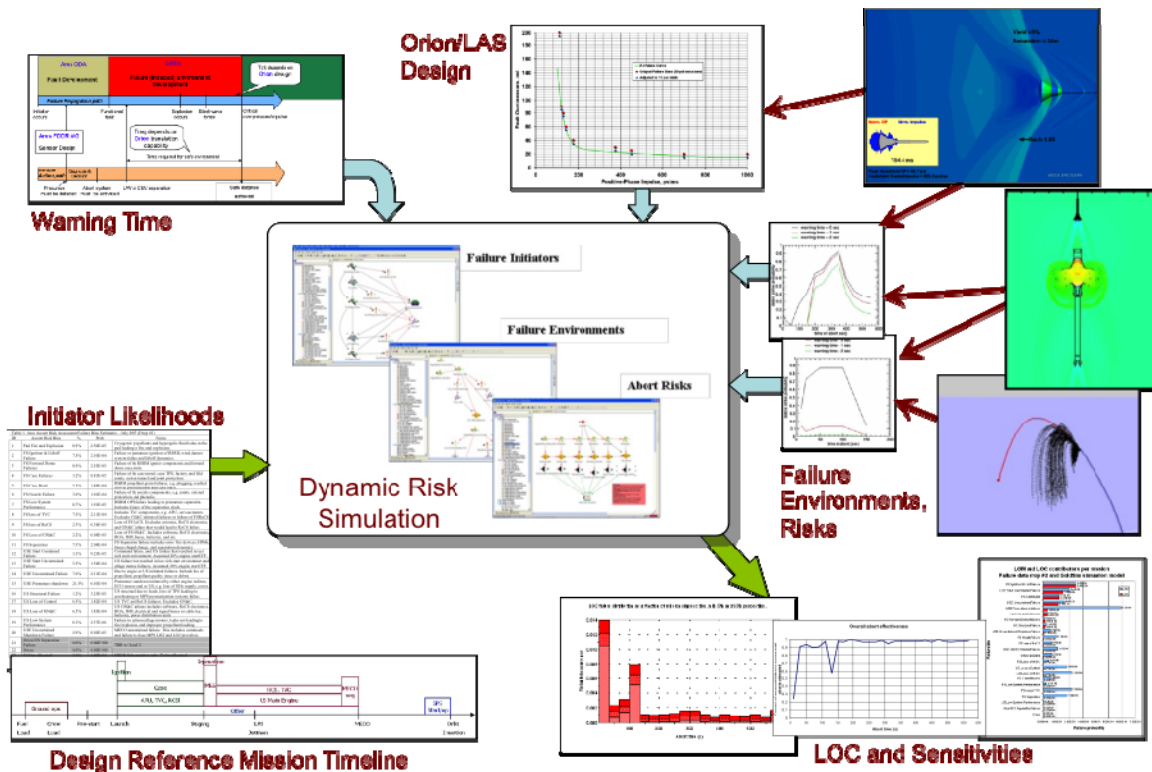


Figure 14-13. Schematic of Integrated Risk Modeling Elements.

14.9.1 Integrated Modeling Evolution

In the early phases of vehicle design and assessment, the integration model may be comprised of very few elements: failure initiators or categories that describe failure manifestation scenarios, probabilities representing the likelihood of each scenario, error factors that describe the uncertainty about each failure probability, and abort effectiveness distributions describing the chances of survival for each failure scenario. To put the elements of this model together, a simple spreadsheet tool may be sufficient. Total LOM probability will be the sum of the individual LOM probabilities due to each failure initiator, and LOC probability will be the sum of each LOM probability multiplied by its corresponding abort effectiveness value. To perform uncertainty analysis, one can perform the above-mentioned process many times using one single sample from the distributions of each of the parameters in each round. The normalized histogram of LOM and LOC probabilities can give a good representation of LOM and LOC distributions.

As the design matures, the risk model becomes more detailed as well—failure initiators get broken down to more specific failure scenarios with more specific mappings, failure probabilities become a function of phase and mission elapsed time, failure detection mechanisms are added to the model, rough abort effectiveness estimates are replaced by hazard environments that are a function of failure time or other flight parameters, abort trigger mechanisms are included to provide earlier warning times, and so on. Integrating a model with all of these details usually calls for a more flexible modeling environment like a Monte Carlo simulation tool.

The integration model needs only to be as detailed as necessary to meaningfully assess the risk with the available information. Details should not necessarily be added just because they are available, but rather because they facilitate superior estimation of the figures of merit defined for that specific phase of the project. For example, if the effectiveness of the failure triggers is one of the metrics that is in question, then trigger mechanisms need to be an explicit part of the model. However, if the overall abort effectiveness is the main focus of the project and triggers are assumed to be part of the vehicle design, then one might decide to integrate triggers with the detection mechanism and thus have a simpler model for assessment.

It may be convenient to integrate the explosive failure environment information with the abort system vulnerability data (including trajectory data) separately from the overall safety assessment integration. Details of the environment propagation are difficult to capture without storing large amounts of data so the process may benefit from considering the abort vehicle vulnerability as part of the environment development. In this case, the output generated consists simply of failure probabilities associated with blast overpressure and with fragments. These may be produced as tabular functions of warning time and mission elapsed time.

14.9.2 Uncertainty and Sensitivity Analyses

Providing decision makers with point estimates for figures of merit without expressing the uncertainty associated with them can be misleading. This is especially true when the outcomes are used to demonstrate that the proposed design meets stated requirements. However, not all of the benefits of uncertainty analysis are for the decision makers; the risk analysts may benefit from this information as well. High uncertainty in parts of the model, especially when there is a wide range of consequences and a big impact on the outcomes, can prompt the analyst to take a more serious look at the problem and try to understand the underlying source of the uncertainty. Uncertainty can be reduced by better investigation of the physical phenomena, higher fidelity modeling, adding more test cases, changing the design to reduce the range of the response to the inputs, and so on.

When uncertainty analysis is a part of the process, every piece of data that is used needs to be examined to determine its precision and accuracy. The uncertainty is usually expressed as a probability distribution function for the possible values of a variable. When computing the uncertainty is prohibitively difficult or costly, sensitivity analyses can be performed to study the effects of changing certain variables and determine the importance of knowing their uncertainties.

14.9.3 Abort Model Review

Like all systems engineering processes, the abort effectiveness assessment process needs to be reviewed. Routine model reviews, discussions with the design teams regarding the findings, and consultation with peers are all part of this process.

Other than these typical tools, the following processes can be useful as well:

- Trying both top-down and bottom-up approaches if possible. A top-down approach uses historical or heritage data while the bottom-up approach is based on the reliability assessment of individual components. These approaches may generate different results, but a comparison between the two can provide analysts with information that can be useful for the review process.
- Comparing the assessment results from one phase to another. As the model becomes more mature, the sources of changes in the assessment results can be defined and studied. For example, if the contribution of a failure initiator to the overall LOC probability

has increased from one revision of the model to another, the reason behind this change can be discussed by all of the parties involved, including designers, managers, and risk analysts. Using this process can help to avoid introducing errors into the model.

14.9.4 Example of Integrated Modeling: Ares I Abort Assessment GoldSim Model

To illustrate the elements and processes involved in integrated risk simulation, this section presents an example of an integrated risk model developed for the Ares I Ascent Abort Risk Assessment. This example represents the risk model at the Preliminary Design Review (PDR) phase of vehicle development, when all the risk model components have been developed and incorporated into the model.

The Ares I ascent abort risk model was built using the GoldSim [14-4] simulation environment to integrate known sources of risk in the ascent abort process and estimate overall abort effectiveness. GoldSim is a Monte Carlo hybrid simulator tool that supports the simulation of dynamic systems with stochastic behavior. Such hybrid simulator tools combine the features of continuous simulators and discrete simulators—they solve differential equations, but can also superimpose discrete events on the continuously varying system. The abort risk simulation model follows the simplified algorithm presented in Figure 14-14.

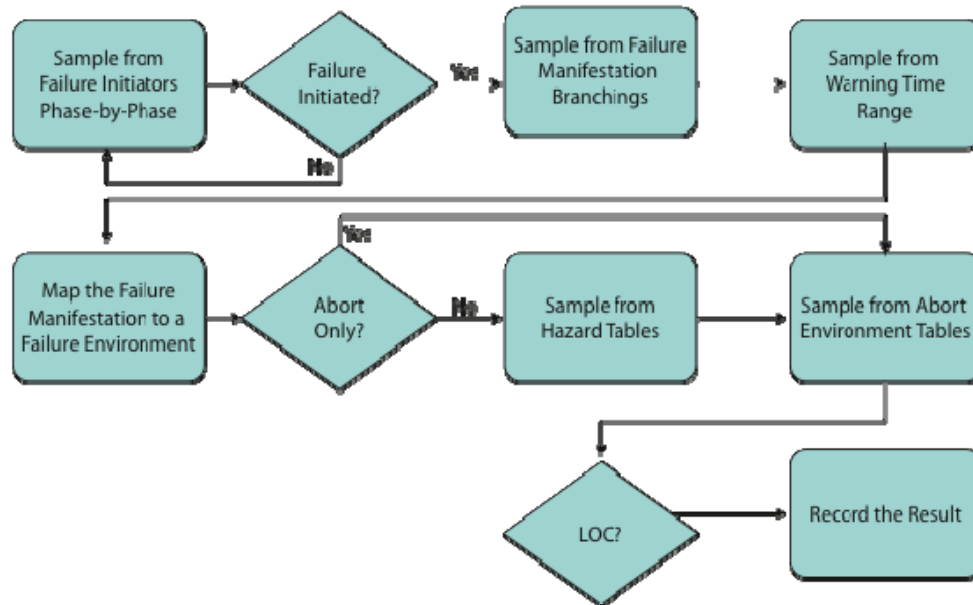


Figure 14-14. Simplified Representation of Risk Simulation Model Algorithm.

The process starts with sampling from the failure initiators at each phase of the mission. When a failure is triggered, the branching probabilities for the triggered scenario are sampled, as is the warning time range for the selected branch. Using the mapping between the failure manifestations and failure environments, a failure environment is selected and the relative hazard tables are sampled to calculate the LOC probability. After sampling from LOC probabilities, Orion and abort-only failure tables are also sampled to find the chance of surviving the abort after observing the Ares-generated potential LOC conditions.

The Ares I GoldSim simulation model consists of five major sections presented as five folders in GoldSim environment: inputs, failure initiators, failure environments (Ares-generated), abort environments (abort-only), and end-states logic and results (Figure 14-15).

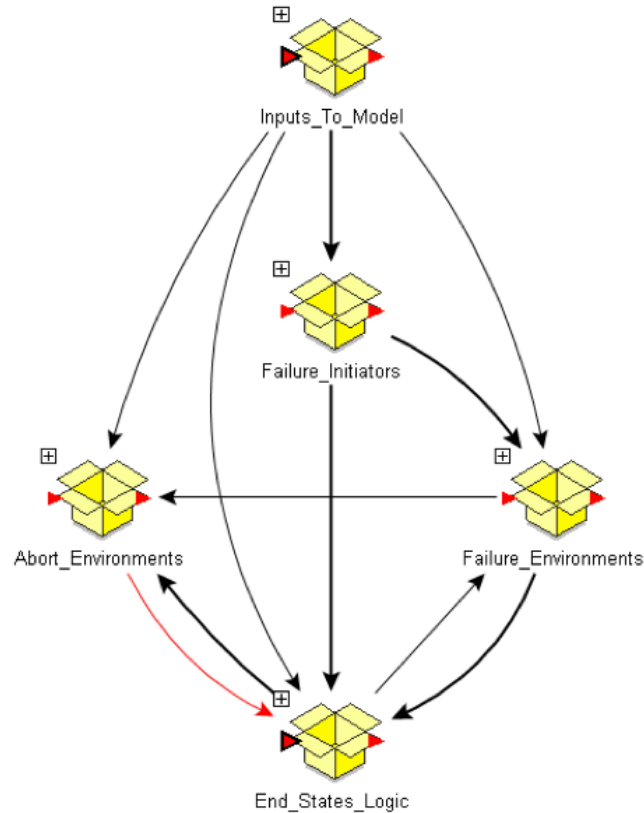


Figure 14-15. Influence Diagram of Main Sections of GoldSim Model.

All of the data used in the simulation are imported from the MS Excel spreadsheets accompanying the model, into the inputs folder at the start of the simulation (Figure 14-16). Imported data include all of the failure initiator information, scenario information, failure manifestation and failure environment data, failure evolution mapping, hazard LOC probabilities, abort environment LOC probabilities, and so on. The inputs folder also includes the simulation elements that trigger the start and end of each mission phase (Figure 14-17). When any piece of these data are needed, a link to elements of this folder provides the needed information to the other parts of the model.

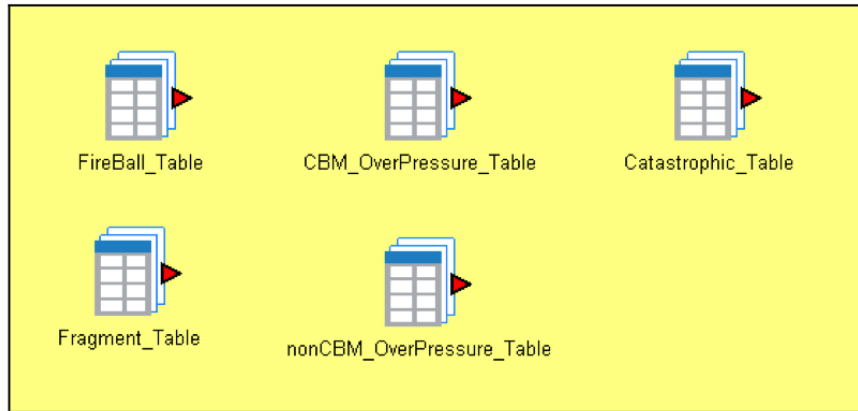


Figure 14-16. Sample GoldSim Inputs That Link to the Excel Spreadsheets.

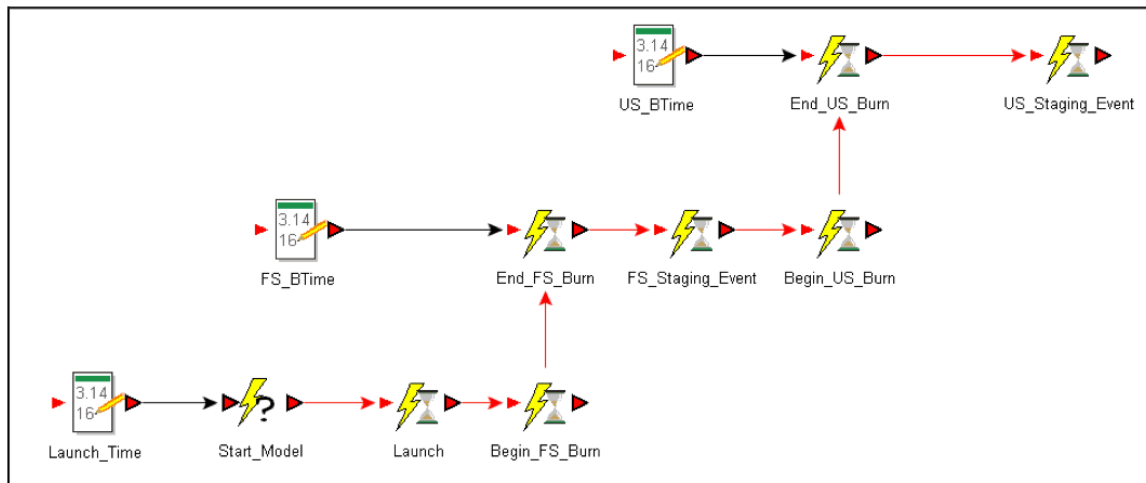


Figure 14-17. Sample GoldSim Inputs That Define the Start and End of Each Phase.

The failure initiators folder simulates the failure behavior of the vehicle during each phase of the mission. Twenty failure initiators can be assigned to each mission phase, each mapping to a unique failure manifestation, and each mission phase can have up to 40 failure scenarios. When a failure scenario of a mission phase is triggered (presenting the failure of that phase), one of the failure manifestations under that scenario is randomly selected and the warning time range of that scenario/manifestation combination is sampled to get a unique warning time for that particular realization. Figure 14-18 shows a screenshot of this folder.

In the failure environments folder, simulation elements needed for mapping between failure manifestations and failure environments are brought together with representatives of each failure environment and the hazard tables (Figure 14-19). When a failure is initiated, a possible failure environment is sampled from the failure mapping table. Each failure environment triggers a subset of hazard tables and the outcome determines whether or not a LOC incident has occurred. After sampling from the failure environments, abort environment is sampled as well, regardless of the outcome. The end-states logic and results folder mostly contains milestone and status elements to track the timing of the events and the outcomes of the random events.

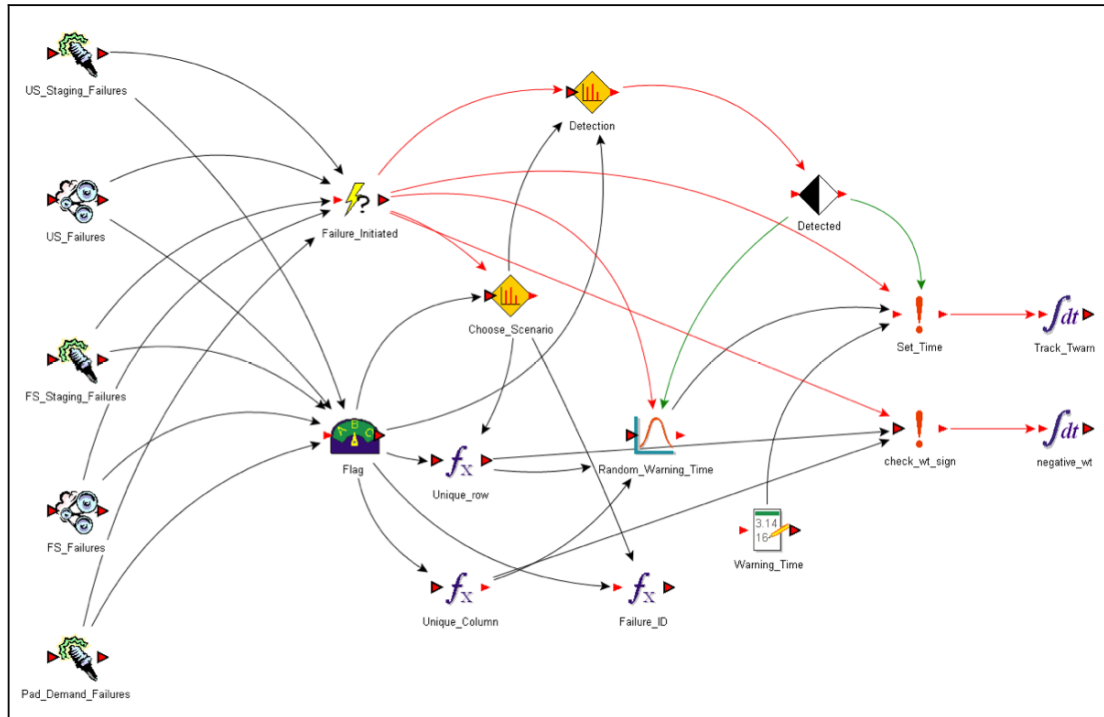


Figure 14-18. Failure Initiation Logic.

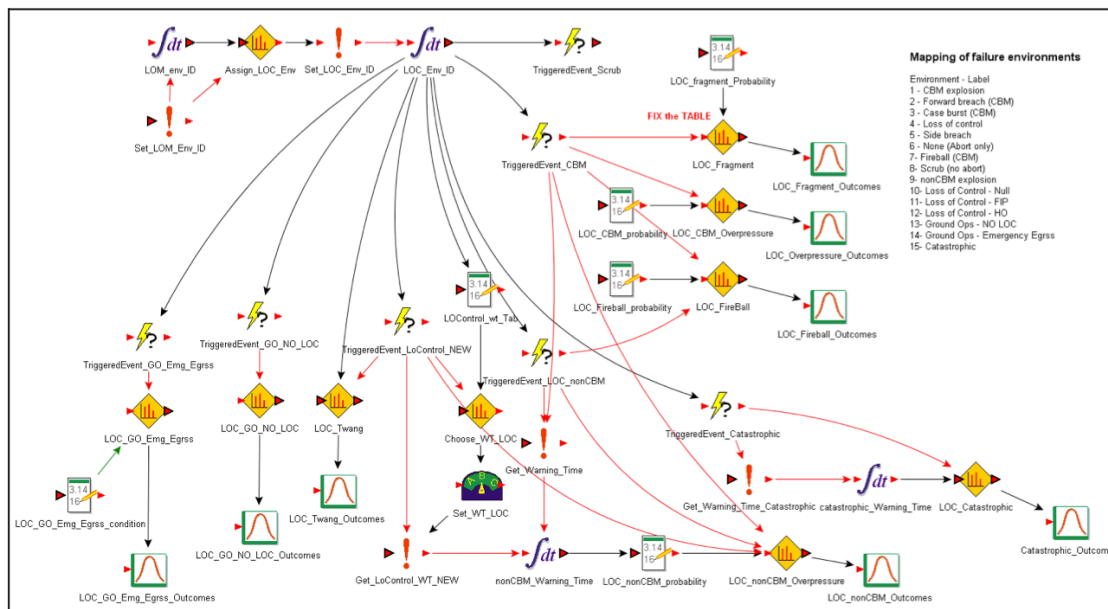


Figure 14-19. Ares-Initiated Failure Environments.

14.10 References

- 14-1 "Human-Rating Requirements and Guidelines for Space Flight Systems," June 2003, National Aeronautics and Space Administration, NPG: 8705.2.
- 14-2 Isakowitz, S. J., Space Launch Systems, Fourth Edition, 2004, Hopkins, J. B., and Hopkins, J. P., AIAA.
- 14-3 Mathias, D. L., Lawrence, S. L., "Launch Architecture Impact on Ascent Abort and Crew Survival," *8th International Conference on Probabilistic Safety Assessment and Management (PSAM)*, May 14-19, 2006, New Orleans, LA.
- 14-4 GoldSim Probabilistic Simulation Environment, GoldSim Technology Group LLC, 22500 SE 64th Place, Suite 240, Issaquah, Washington 98027 USA, <http://www.goldsim.com>.

Appendix A– Probability and its Application to Reliability and Risk Assessment

This appendix reviews elementary probability theory as it applies to reliability and risk assessment. We begin with the logic of certainty, i.e., logical operations with events. The concept of a structure function is introduced along with the important concept of minimal cut sets (MCSs). Elementary probability operations and distribution functions are then presented with examples from reliability and risk assessment. There are several books that cover parts of this material [A-1 through A-6].

A.1 The Logic of Certainty

A.1.1 Events and Boolean Operations

An event is an assertive statement that can be true or false. Thus, “it will rain today” is an event, while the statement “it *may* rain today” is not an event, because it can never be proven to be true or false.

We may assign an *indicator variable*, X , to an event E whose values are 1 or 0 depending on whether the event is true or false, as shown in Figure A-1.

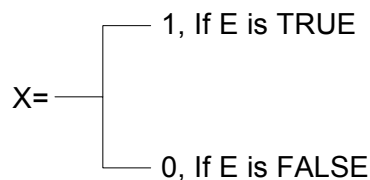


Figure A-1. Definition of an Indicator Variable.

Indicator variables will be useful in performing Boolean operations, as we will see later. At this time, we note that, since these variables are binary, they satisfy the following relation:

$$X^k = X \quad k = 1, 2, \dots \quad (\text{A-1})$$

We now imagine that we perform an “experiment” whose outcome is uncertain. For example, we throw a die; the possible outcomes are $\{1, 2, 3, 4, 5, 6\}$. We call the set of all possible outcomes the *sample space* of the experiment. Another example of an experiment with uncertain outcome is to place a component in operation and wait until it stops functioning. In a generic way, we may imagine the sample space (S) as being represented by all the points inside a rectangle (or any other figure). Each sample point is a possible outcome of the experiment. A collection of points forms an event (E). Of course, such a representation would not be appropriate for an experiment such as the throwing of a die because the outcomes form a discrete set. However, we can work with a continuous set to demonstrate the Boolean laws without loss of generality. Such a representation is called a *Venn* diagram and is shown in Figure A-2.

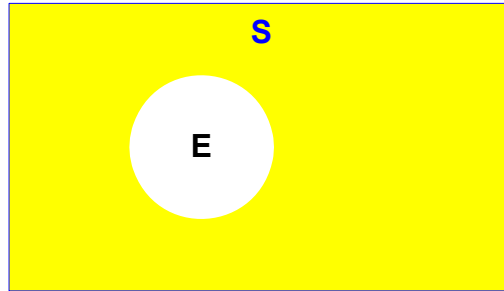


Figure A-2. A Venn Diagram.

We can now discuss the three basic Boolean operations: the negation, the intersection, and the union.

A.1.1.1 Complement of an Event (Negation)

For the event E , we define its complement \bar{E} such that \bar{E} is false when E is true. The indicator variable expression is

$$\bar{X}_E = 1 - X_E \quad (\text{A-2})$$

Figure A-3 shows the Venn diagram for the *NOT* operation, as well as the logic gate “not.”

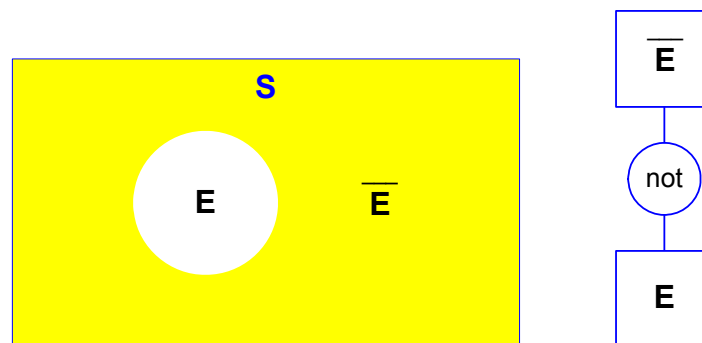


Figure A-3. The NOT Operation.

A.1.1.2 Union

Given two events, A and B , we form a third event C such that C is true whenever either A or B is true. The logic gate OR is shown in Figure A-4. The Boolean and the indicator variable expressions are:

$$A \cup B = C$$

$$X_C = 1 - (1 - X_A)(1 - X_B) \equiv \coprod X_j \quad (\text{A-3})$$

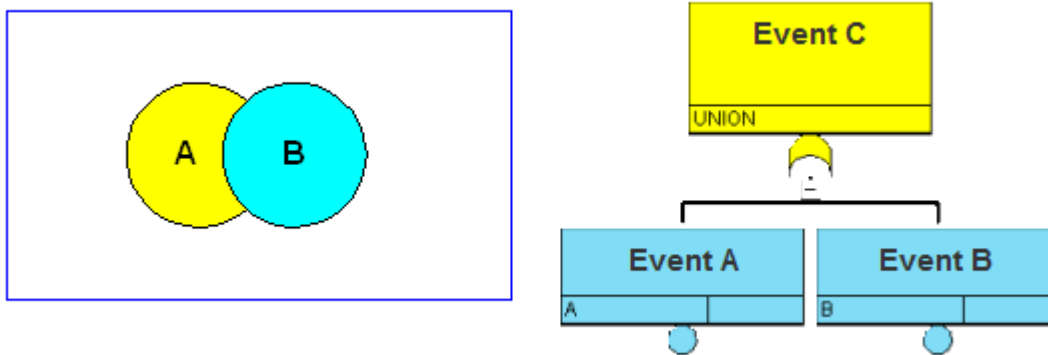


Figure A-4. The Union of Events.

A.1.1.3 Intersection

Given two events A and B, we form a third event C such that C is true whenever both A and B are true. The Venn diagram and the logic gate AND is shown in Figure A-5. The Boolean and the indicator variable expressions are:

$$A \cap B = C$$

$$X_C = X_A X_B \equiv \prod X_j \quad j = A, B \quad (\text{A-4})$$

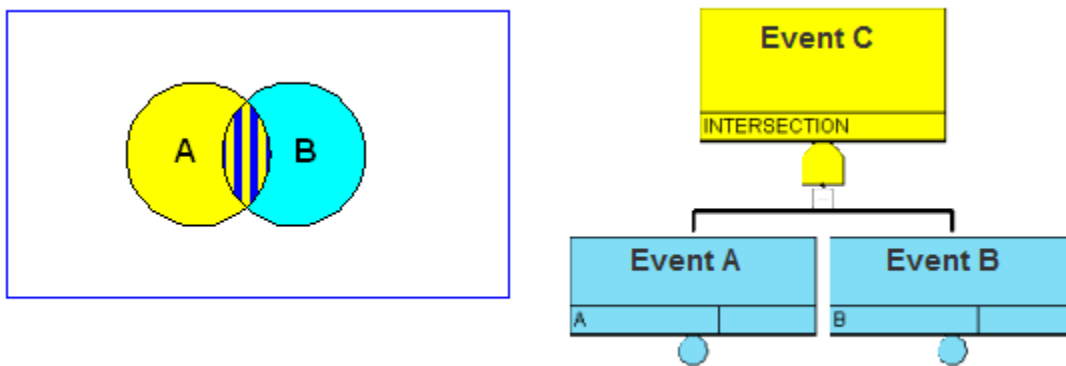


Figure A-5. The Intersection of Events.

Two events are said to be *mutually exclusive* if they cannot be true at the same time. In terms of the Venn diagram, this means that their intersection is empty, i.e.,

$$A \cap B = \phi \quad (\text{A-5})$$

where ϕ denotes the null, or empty set.

A.1.2 Simple Systems

A *series* system is such that all its components are required for system success. Equivalently, the system fails if any component fails. Its block diagram is shown in Figure A-6 where each circle represents one “part” (i.e., a component) of the system. Equations (A-6) and (A-7) show the logical expressions for the indicator variables for failure (X) and success (\bar{X}). In Figure A-7, “X” refers to the event “system failure.” X in Equation (A-6) is the indicator variable of this event.

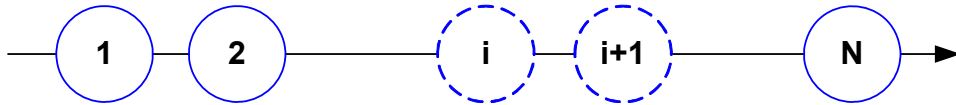


Figure A-6. A Series System of N Components.

$$\text{Failure}^a: \quad X = 1 - \prod_1^N (1 - X_j) \equiv \coprod_1^N X_j \quad (\text{A-6})$$

$$\text{Success:} \quad \bar{X} = \prod_1^N \bar{X}_j \quad (\text{A-7})$$

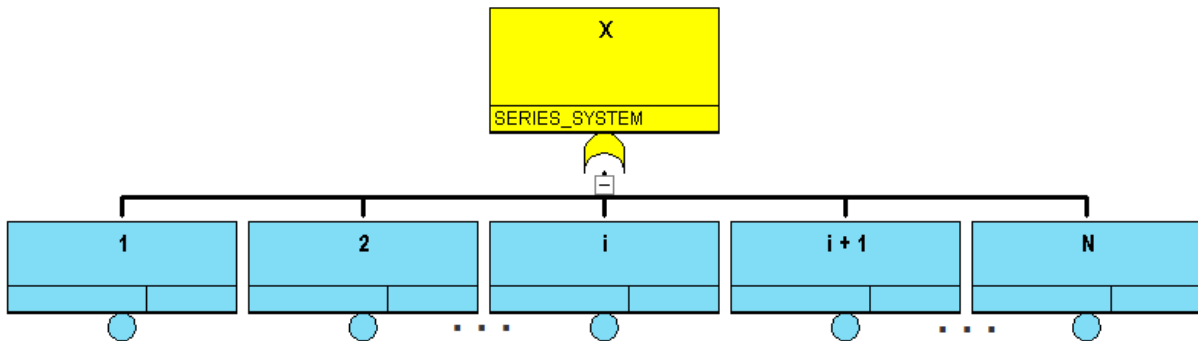


Figure A-7. Pictorial Representation of Equation (A-6).

A *parallel* system is a redundant system that is successful, if at least one of its elements is successful. Equivalently, the system fails if all of its components fail. Figure A-8 shows the system in block-diagram form. Equations (A-8) and (A-9) are the indicator variable expressions for failure and success. Figure A-9 is the corresponding FT.

^a Note that upside down capital pi symbol \coprod represents the coproduct operation (i.e., a disjoint union).

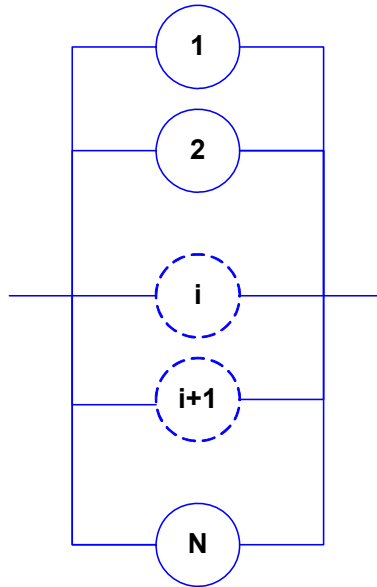


Figure A-8. A Parallel System of N components.

$$\text{Failure: } X = \prod_1^N X_j \quad (\text{A-8})$$

$$\text{Success: } \bar{X} = 1 - \prod_1^N (1 - \bar{X}_j) = \prod_1^N \bar{X}_j \quad (\text{A-9})$$

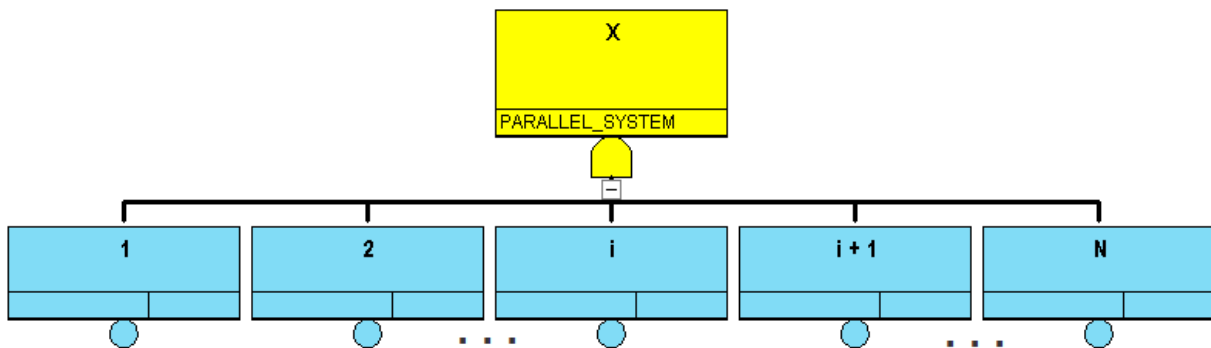


Figure A-9. Pictorial Representation of Equation (A-8).

A.1.3 Structure Functions

Equations (A-6) through (A-9) show that the system indicator variable can be expressed in terms of the indicator variables of the components. In general, the indicator variable of the top event is a function of the primary inputs:

$$X_T = S_F(X_1, X_2, \dots, X_n) \equiv S_F(\underline{X}) \quad (\text{A-10})$$

where $S_F(\underline{X})$ is the *structure or switching function* and it maps an n-dimensional vector of 0s and 1s into 0 or 1.

As an example, consider a *two-out-of-three* system, in which at least two components are needed for success (Figure A-10).

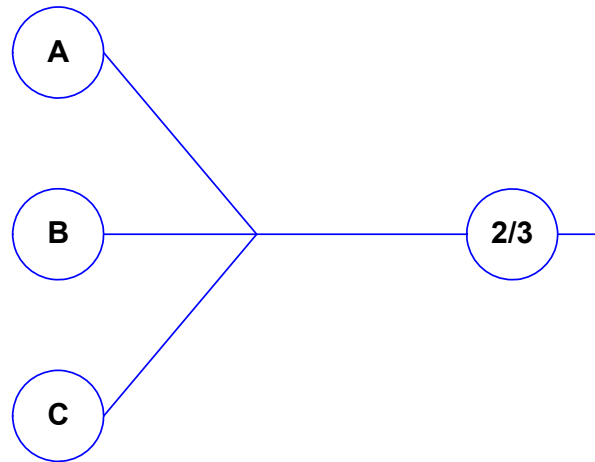


Figure A-10. Block Diagram of the Two-out-of-Three System.

The system fails if any two or all three components fail (OR gate). Thus, we write

$$X_T = 1 - (1 - X_A X_B)(1 - X_B X_C)(1 - X_C X_A)(1 - X_A X_B X_C) \quad (\text{A-11})$$

This is the structure function of this system. We now observe that, if we expand the right-hand side of Equation (A-11) and use Equation (A-1), we get a simpler form, i.e.,

$$X_T = 1 - (1 - X_A X_B)(1 - X_B X_C)(1 - X_C X_A) \quad (\text{A-12})$$

This observation leads us to the concept of cut sets. A cut set is a set of X_i that, when TRUE, they make X_T TRUE, i.e., their failure guarantees failure of the system. Note that $X_A X_B X_C$ is a cut set (Equation (A-11)). However, this cut set does not appear in the simpler Equation (A-12). This leads us to the concept of minimal cut sets (MCSs). A minimal cut set is a cut set that does not contain another cut set as a subset.

The MCSs for the two-out-of-three system are

$$M_1 = X_A X_B, \quad M_2 = X_B X_C, \quad M_3 = X_C X_A \quad (\text{A-13})$$

The indicator variables for the minimal cut sets are M_1 , M_2 , and M_3 .

Equation (A-12) can be written as

$$X_T = \prod_{j=1}^3 M_j \equiv 1 - (1 - M_1)(1 - M_2)(1 - M_3) \quad (\text{A-14})$$

We note that the concept of cut sets applies when the logic represented by the structure function does not contain negations (these are called *coherent* functions). If it does, the corresponding concept is that of *prime implicants*. A discussion of prime implicants is beyond the scope of this appendix (see Reference [A-5]). Equation (A-14) is shown pictorially in Figure A-11.

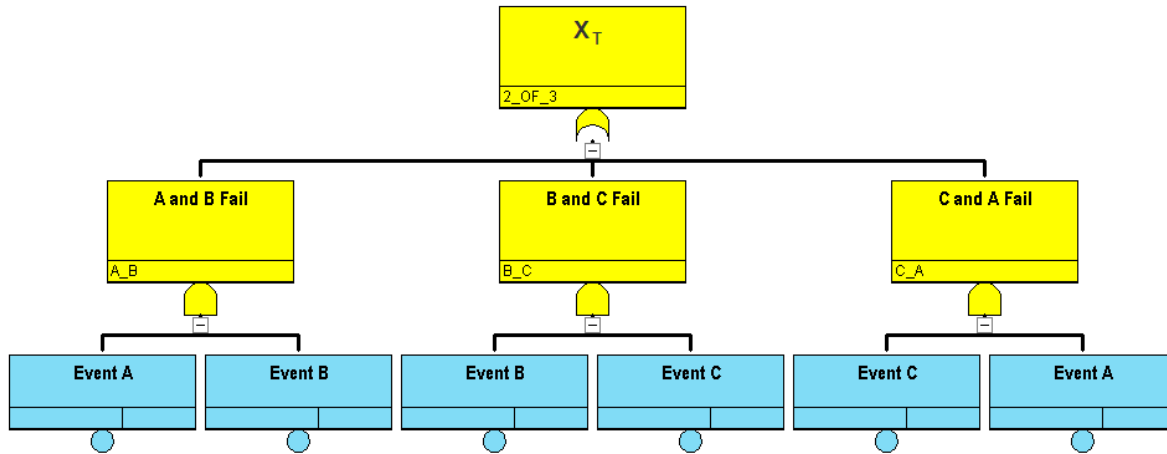


Figure A-11. Pictorial Representation of Equation (A-14).

We can generalize Equation (A-14) to describe a system with any number of MCSs and write

$$X_T = 1 - \prod_{i=1}^N (1 - M_i) \equiv \prod_{i=1}^N M_i \quad (\text{A-15})$$

This is the *disjunctive normal form* (or *sum-of-products* form) of the structure function. Carrying out the multiplication in the above expression yields:

$$X_T = \sum_{i=1}^N M_i - \sum_{i=1}^{N-1} \sum_{j=i+1}^N M_i M_j + \dots + (-1)^{N+1} \prod_{i=1}^N M_i \quad (\text{A-16})$$

Expanding Equation (A-14) or its equivalent Equation (A-12) for the two-out-of-three system we get

$$X_T = (X_A X_B + X_B X_C + X_C X_A) - 2X_A X_B X_C \quad (\text{A-17})$$

where we have used the fact that $X_A X_B^2 X_C = X_A X_B X_C$ (see Equation (A-1)).

Similar expressions can be derived for system success. Then, the equation corresponding to Equation (A-15) is

$$\bar{X}_T = \prod_1^n R_i \quad (\text{A-18})$$

where R_i is the i th *minimal path set*.

Another example is the *two-out-of-four* system. The system works if at least two components work. The MCSs are

$$M_1 = X_1 X_2 X_3 \quad M_2 = X_2 X_3 X_4 \quad M_3 = X_3 X_4 X_1 \quad M_4 = X_1 X_2 X_4 \quad (\text{A-19})$$

And the structure function is

$$X_T = 1 - (1 - M_1)(1 - M_2)(1 - M_3)(1 - M_4) \quad (\text{A-20})$$

or

$$X_T = X_1 X_2 X_3 + X_2 X_3 X_4 + X_3 X_4 X_1 + X_1 X_2 X_4 - 3X_1 X_2 X_3 X_4 \quad (\text{A-21})$$

The identification of minimal cuts for complex structure functions (i.e., large FTs) has been computerized.

A.2 Probability Basics

A.2.1 Definition

In the mathematical theory of probability, the probability of an event A , $\text{Pr}(A)$ satisfies the following Kolmogorov axioms:

$$0 \leq \text{Pr}(A) \leq 1 \quad (\text{A-22})$$

$$\text{Pr}(\text{certain event}) = 1 \quad (\text{A-23})$$

For two mutually exclusive events A and B :

$$\text{Pr}(A \cup B) = \text{Pr}(A) + \text{Pr}(B) \quad (\text{A-24})$$

In PRA, saying that the probability of an event is a mathematical entity that satisfies these axioms is not sufficient; we need to interpret this concept. There are two prominent interpretations of probability:

In the *relative-frequency interpretation*, we imagine a large number n of repetitions of an “experiment” of which A is a possible outcome. If A occurs k times, then its relative frequency is k/n . It is, then, postulated that the probability of A is

$$\lim_{n \rightarrow \infty} \frac{k}{n} \equiv \text{Pr}(A) \quad (\text{A-25})$$

In the *Bayesian interpretation*, there is no need for n “identical” trials. The concept of “likelihood” is taken as being primitive, i.e., it is meaningful to compare the likelihood of two events. Thus, $\text{Pr}(A) < \text{Pr}(B)$ simply means that the assessor judges B to be more likely than A . Probability is a “statement of plausibility,” where distributions are carriers of incomplete information. Definitions in the Bayesian approach include:

Data	Distinct observed (e.g., measured) values of a physical process. Data may be factual or not. For example they may be subject to uncertainties, such as imprecision in measurement, truncation, and errors.
Information	The result of evaluating, processing, or organizing data and information in a way that adds to knowledge.
Knowledge	What is known from gathered information.
Inference	The process of obtaining a conclusion based on what one knows.

Both relative-frequency and degree-of-belief probabilities satisfy the mathematical theory of probability, i.e., the Kolmogorov axioms. The interpretation that is prevalent in PRA is that of degree-of-belief [A-6]. These concepts were presented in more detail in Section 6.3.

A.2.2 Basic Rules

A.2.2.1 Union of Events

For non-mutually exclusive events:

$$\Pr\left(\bigcup_1^N A_i\right) = \sum_{i=1}^N \Pr(A_i) - \sum_{i=1}^{N-1} \sum_{j=i+1}^N \Pr(A_i A_j) + \dots + (-1)^{N+1} \Pr\left(\bigcap_1^N A_i\right) \quad (\text{A-26})$$

For two events:

$$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(AB) \quad (\text{A-27})$$

In PRA, we usually deal with rare events; consequently the intersections of events have very low probabilities. It is very common to approximate Equation (A-26) by

$$\Pr\left(\bigcup_1^N A_i\right) \cong \sum_{i=1}^N \Pr(A_i) \quad \text{rare-event approximation} \quad (\text{A-28})$$

These results can be applied to the disjunctive form of the structure function, Equation (A-16). Thus,

$$\Pr(X_T) = \sum_1^N \Pr(M_i) + \dots + (-1)^{N+1} \Pr\left(\bigcap_1^N M_i\right) \quad (\text{A-29})$$

The rare-event approximation is, then,

$$\Pr(X_T) \cong \sum_1^N \Pr(M_i) \quad (\text{A-30})$$

Note that the intersections of the MCSs must be simplified first (using $X^2 = X$) before probabilities are taken.

As an example, consider the 2-out-of-3 system (Equation (A-17)). Using the rule for the union of events, Equation (A-26), we get

$$\Pr(X_T) = \Pr(X_A X_B) + \Pr(X_B X_C) + \Pr(X_C X_A) - 2\Pr(X_A X_B X_C) \quad (\text{A-31})$$

A.2.2.2 Conditional Probability

We define the *conditional probability* of event A given that we know that event B is TRUE as

$$\Pr(A|B) \equiv \frac{\Pr(AB)}{\Pr(B)} \quad (\text{A-32})$$

Two events, A and B, are said to be *independent* if the knowledge that B is TRUE does not affect our probability of A, i.e.,

$$\Pr(A|B) = \Pr(A) \quad (\text{A-33})$$

Thus, for the two-out-of-three system, assuming independent “identical” components and letting

$$\Pr(X_A) = \Pr(X_B) = \Pr(X_C) \equiv q \quad (\text{A-34})$$

we get

$$\Pr(X_T) = 3q^2 - 2q^3 \quad (\text{A-35})$$

For $q = 0.1$, we get

$$\Pr(X_T) = 0.028$$

The rare-event approximation, Equation (A-30), gives

$$\Pr(X_T) = 3q^2 = 0.030$$

A.2.3 Theorem of Total Probability

Given a set of events, $H_i, (i = 1 \dots N)$, that are mutually exclusive and exhaustive

($H_i \cap H_j = \phi$, for $i \neq j$, $\bigcup_{i=1}^N H_i = S$), the probability of any event E can be expressed as

$$\Pr(E) = \sum_{i=1}^N \Pr(E|H_i) \Pr(H_i) \quad (\text{A-36})$$

A.2.4 Bayes' Theorem

Suppose that evidence E becomes available. What are the new (updated) probabilities $\Pr(H_i|E)$? These are given by Bayes' Theorem as follows:

$$\Pr(H_i|E) = \frac{\Pr(E|H_i)\Pr(H_i)}{\sum_1^N \Pr(E|H_i)\Pr(H_i)} \quad (\text{A-37})$$

The probabilities $\Pr(H_i)$ are the *prior* probabilities, i.e., those that are valid prior to receiving the evidence E . Similarly, the probabilities $\Pr(H_i|E)$ are the *posterior* probabilities. The factor $\Pr(E|H_i)$ is called the *likelihood function*.

From Equation (A-32):

$$\Pr(H_i|E) \Pr(E) = \Pr(E|H_i) \Pr(H_i)$$

using Equation (A-36) we get Equation (A-37).

As an example, suppose that a piece of piping is subjected to an aging mechanism. The probability that this mechanism exists is 0.5. A visual inspection has a probability of 0.6 of identifying the mechanism, if it exists, and a probability of 0.1 of declaring that it is there, if it does not exist (false alarm). What is the probability that the mechanism actually exists when the visual test is positive?

Here, we have two mutually exclusive and exhaustive hypotheses:

- H_1 : the mechanism exists
- H_2 : the mechanism does not exist

Let pvt and nvt denote a positive visual test and a negative visual test, respectively. We can see the various possibilities regarding the test in Figure A-12. The evidence is that a positive test is indeed obtained. Then, Bayes' Theorem gives

$$\Pr(H_1|pvt) = \frac{\Pr(pvt|H_1) \Pr(H_1)}{\Pr(pvt|H_1) \Pr(H_1) + \Pr(pvt|H_2) \Pr(H_2)} = \frac{0.3}{0.3 + 0.05} = 0.86$$

Similarly, the probability that the mechanism actually exists when the test is negative is

$$\Pr(H_1|nvt) = 0.31$$

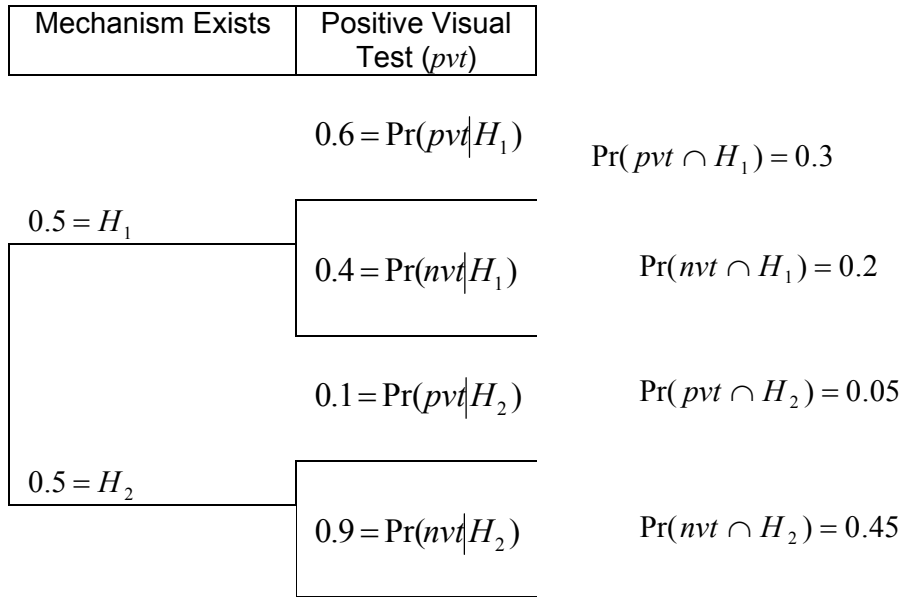


Figure A-12. Various Cases for the Inspection Example.

A.3 Failure Distributions

A.3.1 Random Variables

As we stated in Section A.1.1, events are represented as sets of sample points of the sample space. For example, the event $\{event\}$ is represented by the set $\{2,4,6\}$ of sample points of the die experiment.

A function that maps sample points onto the real line is a *random variable* (RV). For any (one-dimensional) RV, we can represent its possible values on the real number line and then we say that $\{X \leq x\}$ is an event. Figure A-13 shows the real number line and the sample points for the die experiment.

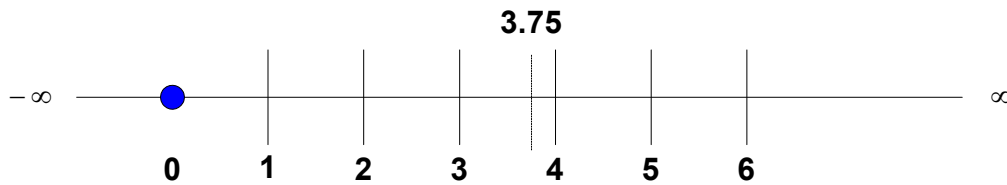


Figure A-13. The Random Variable for the Die Experiment.

For the die experiment, the following are events:

$$\{X \leq 3.75\} = \{1, 2, 3\} \equiv \{1 \text{ or } 2 \text{ or } 3\}$$

$$\{X \leq 96\} = S \quad (\text{the certain event})$$

$$\{X \leq -62\} = \phi \quad (\text{the impossible event})$$

The sample space for the die is an example of a *discrete sample space*. X is a *discrete random variable* (DRV). A sample space is *discrete* if it has a finite or countably infinite number of sample points.

A sample space is *continuous* if it has an infinite (and uncountable) number of sample points. The corresponding RV is a *continuous random variable* (CRV).

A.3.2 Distribution Functions

The *cumulative distribution function* (CDF) of the random variable X is defined as

$$F(x) \equiv \Pr[X \leq x] \quad (\text{A-38})$$

This is true for both DRV and CRV.

A CDF has the following properties:

- $F(x)$ is a non-decreasing function of x ;
- $F(-\infty) = 0$ (the probability of the impossible event) (A-39)

- $F(\infty) = 1$ (the probability of the certain event) (A-40)

Figure A-14 shows the CDF for the die experiment.

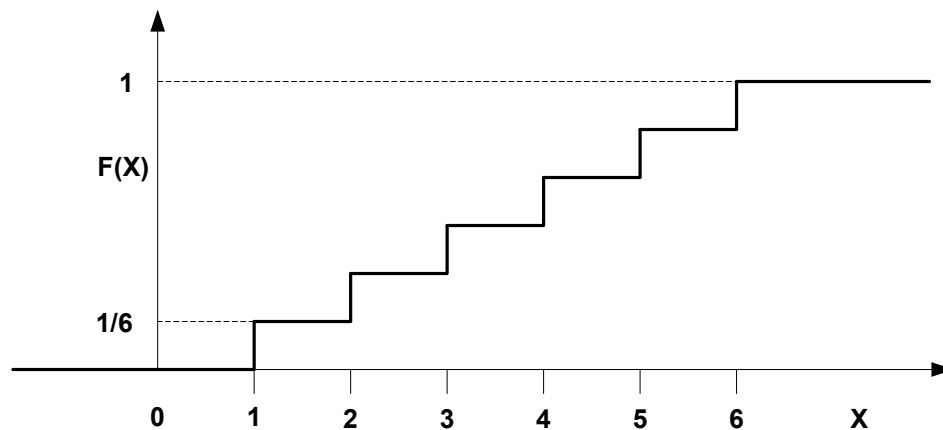


Figure A-14. The Cumulative Distribution Function for the Die Experiment.

As an example, using Figure A-14, we find the following:

$$\Pr[2.1 < X < 4.3] = \Pr[X = 3 \text{ or } 4] = F(4) - F(2) = \frac{4}{6} - \frac{2}{6} = \frac{1}{3}$$

For DRV, we define the probability mass function (pmf) as

$$\Pr(X = x_i) \equiv p_i \quad (\text{A-41})$$

From the definitions, it follows that

$$F(x) = \sum p_i, \text{ for all } x_i \leq x \quad (\text{A-42})$$

Furthermore, property (A-40) of the CDF requires that (normalization condition):

$$\sum p_i = 1 \text{ for all } i \quad (\text{A-43})$$

For CRV, we define the probability density function (pdf) as

$$f(x) = \frac{dF(x)}{dx} \quad (\text{A-44})$$

Then,

$$F(x) = \int_{-\infty}^x f(s) ds \quad \text{and} \quad \int_{-\infty}^{\infty} f(x) dx = 1 \quad (\text{normalization}) \quad (\text{A-45})$$

Example:

Determine k so that

$$f(x) = kx^2, \text{ for } 0 \leq x \leq 1$$

$$f(x) = 0, \text{ otherwise}$$

is a pdf.

Answer:

For this function to be a pdf, it must satisfy the normalization condition, Equation (A-45), i.e.,

$$\int_0^1 kx^2 dx = 1 \quad \Rightarrow \quad k = 3$$

The CDF is

$$F(x) = \int_0^x 3s^2 ds = x^3 \quad \text{for } 0 \leq x \leq 1$$

$$F(x) = 0 \quad \text{for } x \leq 0; \quad \text{and} \quad F(x) = 1 \quad \text{for } 1 \leq x$$

As an example, we calculate the probability that the RV will be between 0.75 and 0.875 (see Figure A-15).

$$\Pr(0.75 \leq X \leq 0.875) = F(0.875) - F(0.75) = 0.67 - 0.42 = 0.25 \quad (\text{A-46})$$

Using the pdf, we calculate

$$\Pr(0.75 \leq X \leq 0.875) = \int_{0.75}^{0.875} 3x^2 dx = 0.25 \quad (\text{A-47})$$

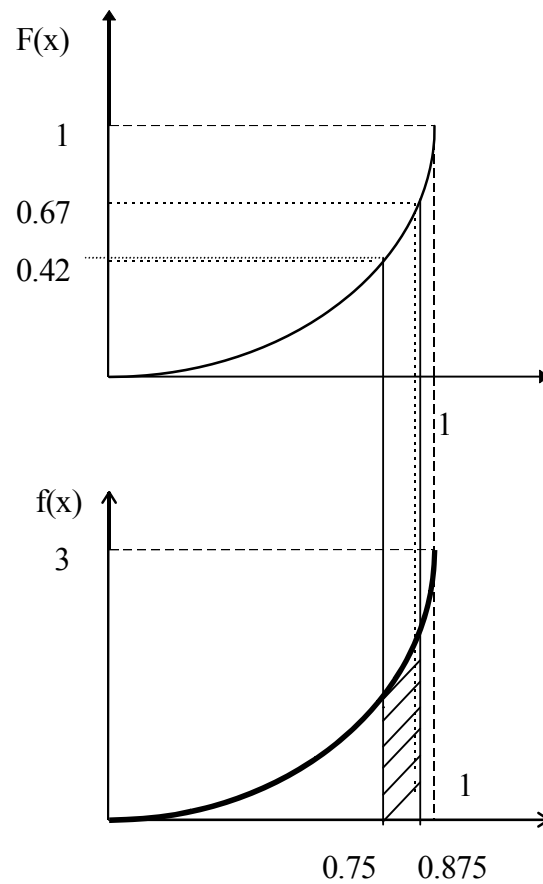


Figure A-15. CDF and pdf for the Example.

A.3.3 Moments

The moments of distributions are summary measures and are useful for communication purposes.

The most common moments are

Expected (or mean, or average) value:

$$E[x] \equiv \alpha \equiv \begin{cases} \int_{-\infty}^{\infty} x f(x) dx & \text{for CRV} \\ \sum_j x_j p_j & \text{for DRV} \end{cases}$$

(A-48)

Variance:

$$E[(x - \alpha)^2] \equiv \sigma^2 \equiv \begin{cases} \int_{-\infty}^{\infty} (x - \alpha)^2 f(x) dx & \text{for CRV} \\ \sum_j (x_j - \alpha)^2 p_j & \text{for DRV} \end{cases} \quad (\text{A-49})$$

Standard deviation:

$$sd \equiv \sigma \quad (\text{A-50})$$

Coefficient of variation:

$$cov \equiv \frac{\sigma}{E[x]} \quad (\text{A-51})$$

Other summary measures include

Mode (or most probable value):

For DRV \Rightarrow value of x_i for which p_i is largest.

For CRV \Rightarrow value of x for which $f(x)$ has a maximum.

Median:

The value x_m for which $F(x_m) = 0.50$

For CRV we define the 100γ percentile as that value of x for which

$$\int_{-\infty}^{x_\gamma} f(x) dx = \gamma \quad (\text{A-52})$$

Example:

For

$$\begin{aligned} f(x) &= 3x^2, \text{ for } 0 \leq x \leq 1 \\ f(x) &= 0, \text{ otherwise} \end{aligned} \quad (\text{A-53})$$

We find:

$$\text{mean value: } E[x] = \int_0^1 3x^3 dx = 0.75 \quad (\text{A-54})$$

$$\text{variance: } \sigma^2 = \int_0^1 3(x - 0.75)^2 x^2 dx = 0.0375 \quad (\text{A-55})$$

$$\text{standard deviation: } \sigma = \sqrt{0.0375} = 0.1936 \quad (\text{A-56})$$

$$\text{coefficient of variation: } \text{cov} = \frac{\sigma}{E[x]} = 0.2582 \quad (\text{A-57})$$

$$\text{mode: } x = 1 \quad (\text{A-58})$$

$$\text{median: } F(x_m) = x_m^3 = 0.5 \Rightarrow x_m = 0.79 \quad (\text{A-59})$$

$$\text{5th percentile: } x_{0.05}^3 = 0.05 \Rightarrow x_{0.05} = 0.37 \quad (\text{A-60})$$

$$\text{95th percentile: } x_{0.95}^3 = 0.95 \Rightarrow x_{0.95} = 0.98 \quad (\text{A-61})$$

A.4 References

- A-1 H-S. Ang and W.H. Tang, *Probability Concepts in Engineering Planning and Design*. Vol. 1: Basic Principles (1975). Vol. 2: Decision, Risk, and Reliability (1984), Wiley, NY.
- A-2 T. Bedford and R. Cooke, *Probabilistic Risk Analysis*, Cambridge University Press, UK, 2001.
- A-3 A. Hoyland and M. Rausand, *System Reliability Theory*, Wiley-Interscience, NY, 1994.
- A-4 S.S. Rao, *Reliability-Based Design*, McGraw-Hill, NY, 1992.
- A-5 H. Kumamoto and E.J. Henley, *Probabilistic Risk Assessment and Management for Engineers and Scientists*, Second Edition, IEEE Press, NY, 1996.
- A-6 G. Apostolakis, "The Concept of Probability in Safety Assessments of Technological Systems," *Science*, 250:1359-1364, 1990.

Appendix B - Event Frequencies and Hardware Failure Models

B.1 Probability of Failure on Demand: The Binomial Distribution

We define:

$$Pr(\text{failure to start on demand}) \equiv q \quad (\text{B-1})$$

$$Pr(\text{successful start on demand}) \equiv p \quad (\text{B-2})$$

Clearly

$$q + p = 1 \quad (\text{B-3})$$

A distribution that is often used in connection with these probabilities is the binomial distribution. It is defined as follows.

Start with an experiment that can have only two outcomes (see Figure B-1):

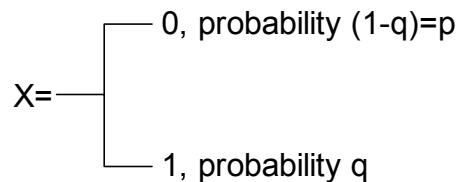


Figure B-1. Binary States of an Experiment.

Consider n independent “trials,” i.e., independent repetitions of this experiment with constant p . These are *Bernoulli trials*.

Define a new discrete random variable (DRV):

X = number of 1’s (failures) in n trials

The sample space of X is $\{0, 1, 2, \dots, n\}$.

Then, the probability of exactly k failures in n trials is

$$Pr[X = k] = \binom{n}{k} q^k (1 - q)^{n-k} \quad (\text{B-4})$$

This is the pmf of the *binomial distribution*. The *binomial coefficient* is defined as

$$\binom{n}{k} \equiv \frac{n!}{k!(n-k)!} \quad (\text{B-5})$$

The two commonly used moments are:

$$E[X] = qn \quad \text{mean number of failures} \quad (\text{B-6})$$

$$\sigma^2 = nq(1-q) \quad \text{variance} \quad (\text{B-7})$$

The pmf satisfies the normalization condition (see Equation (A-43))

$$\sum_0^n \binom{n}{k} q^k (1-q)^{n-k} = 1 \quad (\text{B-8})$$

The probability that n trials will result in at most m failures is (see Equation (A-42))

$$\Pr(\text{at most } m \text{ failures}) = \sum_{k=0}^m \binom{n}{k} q^k (1-q)^{n-k} \quad (\text{B-9})$$

As an example, consider the 2-out-of-3 system. Assuming that the components are independent and nominally identical, each having a failure probability equal to q , we find the failure probability of the system is

$$\Pr(\text{system failure}) = \Pr(2 \text{ or more fail}) = 3q^2(1-q) + q^3 = 3q^2 - 2q^3 \quad (\text{B-10})$$

We can confirm this result by going back to Equation (A-17) which gives the indicator variable for the system as a function of the indicator variables of its components, i.e.,

$$X_T = (X_A X_B + X_B X_C + X_C X_A) - 2X_A X_B X_C \quad (\text{B-11})$$

Since the probability that each X_j , $j = A, B, C$, is true is q , Equation (B-11) gives

$$\Pr(\text{system failure}) = \Pr(X_T = 1) = 3q^2 - 2q^3 \quad (\text{B-12})$$

which is the same as Equation (B-10). Note, however, that the use of the Binomial distribution, Equation (B-10), required the assumption of independent and nominally identical components while Equation (B-11) can be used in general. More discussion on the Binomial distribution, and the other distributions presented in this appendix, can be found in References B-1 and B-2.

B.2 Failure While Operating

We are now dealing with a continuous random variable (CRV), namely, T , the time to failure of a component. Then (see Section A.3.2),

$F(t)$: failure distribution

$$R(t) \equiv 1 - F(t) = \text{reliability} \quad (\text{B-13})$$

$f(t)$: failure density

$$f(t)dt = \Pr(\text{failure occurs between } t \text{ and } t+dt) \quad (\text{B-14})$$

It can be shown that the mean time to failure (MTTF) is given by

$$MTTF = \int_0^{\infty} R(t) dt \quad (\text{B-15})$$

The hazard function or failure rate is defined as

$$h(t) \equiv \frac{f(t)}{R(t)} \quad (\text{B-16})$$

It follows that

$$F(t) = 1 - \exp\left(-\int_0^t h(s) ds\right). \quad (\text{B-17})$$

Note the distinction between $h(t)$ and $f(t)$:

$f(t)dt$ unconditional probability of failure in $(t, t + dt)$.

$h(t)dt$ conditional probability of failure in $(t, t + dt)$ given that the component has survived up to t .

Typical behavior of the failure rate is the “bathtub curve” (called this because of its shape).

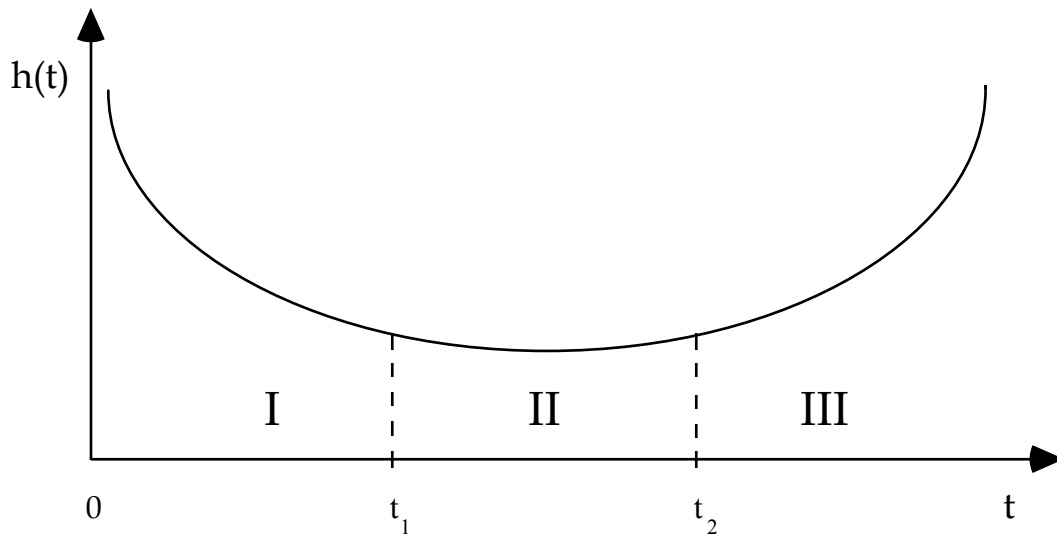


Figure B-2. The Bathtub Curve.

Period I (see Figure B-2) is termed as “infant mortality.” This is when components that have design errors are eliminated. Period II is the “useful life.” The failure rate of the component is nearly constant (although, mechanical components may have a very short such period). Finally, Period III represents “aging (wear-out).” The failure rate is increasing due to mechanisms of cumulative damage during aging.

B.3 The Exponential Distribution

This distribution is used widely in reliability and risk assessment because it is the only one with a constant failure rate. Its probability density function (pdf) is

$$f(t) = \lambda e^{-\lambda t}, \lambda > 0, t > 0 \quad (\text{B-18})$$

where λ is the failure rate.

The CDF is

$$F(t) = 1 - e^{-\lambda t} \quad (\text{B-19})$$

and the reliability

$$R(t) = e^{-\lambda t} \quad (\text{B-20})$$

The hazard function is

$$h(t) = \lambda = \text{constant}$$

and the first two moments are

$$E[T] = \frac{1}{\lambda}; \quad \sigma^2 = \frac{1}{\lambda^2} \quad (\text{B-21})$$

Very often, we use the approximation

$$F(t) \cong \lambda t, \quad \text{for } \lambda t < 0.10 \quad (\text{rare-event approximation}) \quad (\text{B-22})$$

Example:

Consider again the 2-out-of-3 system. Assume independent and nominally identical exponential components with failure rate λ . Then, the *unreliability* of each component as a function of time is

$$q(t) = 1 - e^{-\lambda t}$$

Thus, using Equation (B-10) we find that the unreliability of the system is

$$F_S(t) = 3(1 - e^{-\lambda t})^2 - 2(1 - e^{-\lambda t})^3 \quad (\text{B-23})$$

The MTTF of the system is (Equation (B-15)):

$$MTTF = \int_0^{\infty} [1 - F_S(t)] dt = \frac{5}{6\lambda} \quad (\text{B-24})$$

Recalling that the MTTF for a single exponential component is $\frac{1}{\lambda}$, we see that the 2-out-of-3 system is slightly worse.^a

Let's assume that $\lambda = 10^{-3}$ per hour and $t = 720$ hours (one month). Therefore, the unreliability of the system is

a. Note that this is one of the ways that the performance of the two systems can be compared. For example, we may compare the actual unreliabilities as functions of time. In particular, we are seeking the time τ for which $F_S(t) = 3(1 - e^{-\lambda t})^2 - 2(1 - e^{-\lambda t})^3 < (1 - e^{-\lambda t})$. Solving this inequality leads to the conclusion that the unreliability of the 2-out-of-3 system is smaller than that of a single component for $t < (0.693/\lambda)$. Thus, for reasonable times, the 2-out-of-3 system performs better.

$$F_s(720) = 0.52 \quad (\text{B-25})$$

Note that the rare-event approximation in Equation (B-25) is inappropriate here because $\lambda t > 0.1$.

The system reliability is

$$R_s(t) = 1 - 0.52 = 0.48 \quad (\text{B-26})$$

B.4 The Weibull Distribution

A flexible distribution that is used widely in reliability is the *Weibull distribution*. Its CDF is

$$F(t) = \begin{cases} 1 - e^{-(\lambda t)^b} & \text{for } t > 0 \\ 0 & \text{otherwise} \end{cases} \quad (\text{B-27})$$

where $b > 0$ and $\lambda > 0$.

It can be shown that

$$E[T] = \frac{1}{\lambda} \cdot \Gamma\left(\frac{1}{b} + 1\right) \quad (\text{B-28})$$

$$\sigma^2 = \frac{1}{\lambda^2} \left(\Gamma\left(\frac{2}{b} + 1\right) - \Gamma^2\left(\frac{1}{b} + 1\right) \right) \quad (\text{B-29})$$

where Γ is the gamma function.

The reliability is

$$R(t) = e^{-(\lambda t)^b} \quad \text{for } t > 0 \quad (\text{B-30})$$

and the pdf:

$$f(t) = b\lambda(\lambda t)^{b-1} e^{-(\lambda t)^b} \quad (\text{B-31})$$

The hazard function is

$$h(t) = b\lambda(\lambda t)^{b-1} \quad (\text{B-32})$$

We observe that for $b < 1$, $b = 1$, and $b > 1$, this distribution can be used as a life distribution for the infant mortality, useful life, and wear-out periods, respectively (Figure B-3).

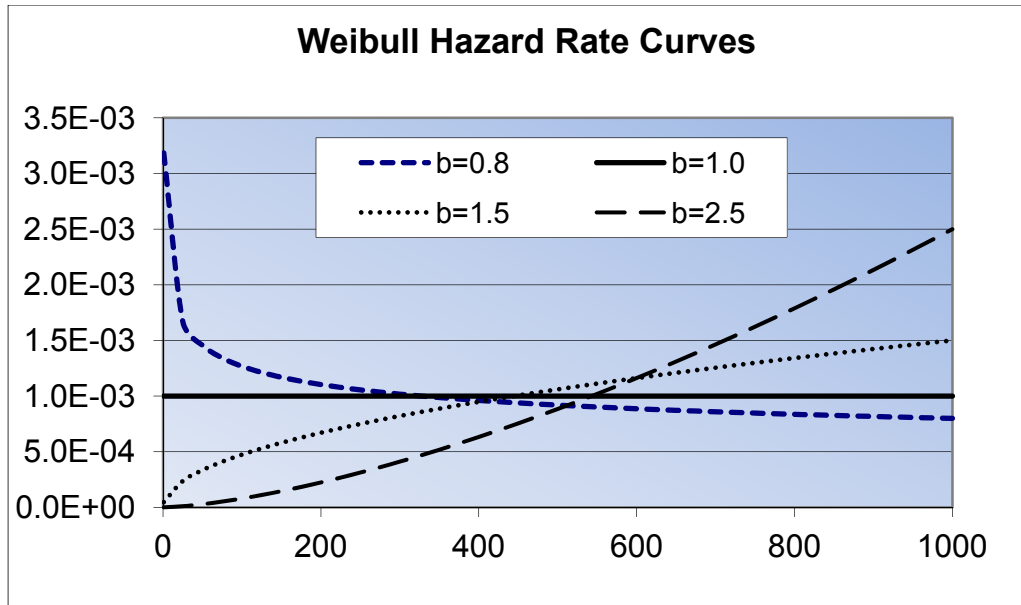


Figure B-3. Weibull Hazard Functions for Different Values of b .

B.5 Event Frequency: The Poisson Distribution

This distribution, unlike the exponential and Weibull distributions but similar to the binomial distribution, deals with DRVs. Events occur over a continuum (time, space) at an average constant rate, λ . The occurrence of an event in a given interval is assumed to be independent of that in any other nonoverlapping interval.

Given an interval $(0,t)$, the DRV of interest is the number of events occurring in that interval. The sample space is $\{0,1,\dots\}$ with a countably infinite number of sample points.

The *Poisson distribution* gives the probability of exactly k events occurring in $(0,t)$

$$\Pr(k) = e^{-\lambda t} \frac{(\lambda t)^k}{k!} \quad (\text{B-33})$$

$$E(k) = \lambda t; \quad \sigma^2 = \lambda t \quad (\text{B-34})$$

This distribution is used to describe the occurrence of initiating events (IEs) in risk assessment.

Example 1

A component fails due to “shocks” that occur, on the average, once every 100 hours. What is the probability of exactly one failure in 100 hours? Of no failure? Of at most two failures?

Answer

$$\lambda t = \frac{1}{100} \times 100 = 1$$

$\Pr(1) = e^{-1} = 0.37$, probability of exactly one failure.

Similarly, $\Pr(0) = e^{-1} = 0.37$, probability of no failure.

$$\Pr(2) = e^{-1} \frac{1^2}{2!} = 0.185 \quad (\text{B-35})$$

$$\Pr(\text{at most two failures}) = \Pr(0 \text{ or } 1 \text{ or } 2) = 0.37 + 0.37 + 0.184 = 0.92$$

B.6 Unavailability

Unavailability at time t is the probability that the component will be down at t , i.e.,

$$q(t) = \Pr[\text{down at } t] \quad (\text{B-36})$$

Note the difference between unreliability (which refers to a time interval, Equation (B-13)) and unavailability (which refers to a specific time).

We distinguish the following cases:

1. Unattended components

$$q(t) = F(t), \quad \text{the CDF} \quad (\text{B-37})$$

Example:

The unreliability of 0.52 that was calculated in the example of Section B.3 is also the unavailability of that system, if we assume that it is unattended.

2. Continuously monitored and repairable components

Each component is monitored so that its failure is immediately known. If the mean time for repair (or replacement) (MTTR) is τ , then the steady-state unavailability is

$$q = \frac{MTTR}{MTTF + MTTR} = \frac{\tau}{\frac{1}{\lambda} + \tau} = \frac{\lambda\tau}{1 + \lambda\tau} \cong \lambda\tau \quad (\text{B-38})$$

since (usually) $\lambda\tau < 0.10$.

This expression for q is an asymptotic result.

Example:

Consider, again, the 2-out-of-3 system (Equation (B-11)). Suppose that $\lambda = 10^{-3}$ per hour and that, upon failure, a component is replaced. The mean replacement time is $\tau = 24$ hours. Then, $\lambda\tau = 0.024$ and $q \cong 0.024$, and $\Pr(X = 1) = 3q^2 - 2q^3 = 1.7 \times 10^{-3}$ is the system unavailability.

B.7 References

- B-1 A. Hoyland and M. Rausand, *System Reliability Theory*, Wiley-Interscience, NY, 1994.
 B-2 S.S. Rao, *Reliability-Based Design*, McGraw-Hill, NY, 1992.

Appendix C - Bayesian Inference Calculations

To illustrate the technique of using Bayesian inference for PRA parameter estimation, this Appendix provides several examples based upon NASA-SP-2009-569 Bayesian Inference for NASA Probabilistic Risk and Reliability Analysis.

C.1 Inference for Common Aleatory Models

We begin with the most commonly encountered situations in PRA, which meet the following three assumptions:

- The aleatory model of the world (corresponding to the likelihood function in Bayes' Theorem) contains a single unknown parameter.
- The prior information is homogeneous^a and is known with certainty.
- The observed data are homogeneous and are known with certainty.

We treat the three most common **aleatory models** in this section:

- Binomial distribution for failures on demand
- Poisson distribution for initiating events and failures in time
- Exponential distribution for random durations, such as time to failure or time to recover

Lastly, we close this section with guidance on selecting prior distributions for single-parameter problems.

Binomial Distribution for Failures on Demand

This model is often used when a component must change state in response to a demand. For example, a relief valve may need to open to relieve pressure upon receipt of a signal from a controller that an over-pressure condition exists. The following assumptions underlie the binomial distribution:

- There are two possible outcomes of each demand, typically denoted as success and failure.
- There is a constant probability of outcome (which would be the probability of failure in PRA, and the probability of success in reliability engineering) on each demand, denoted herein as p .
- The outcomes of earlier demands do not influence the outcomes of later demands (i.e., the order of failures/successes is irrelevant).

The unknown parameter in this model is p , and the observed data are the number of failures, denoted by x , in a specified number of demands, denoted as n . Both x and n are assumed to be known with **certainty** in this section. Cases in which x and n are uncertain are not discussed here.

Note that the binomial distribution describes the aleatory uncertainty in the number of failures, x . The Bayesian inference describes how the epistemic uncertainty in p changes from the prior distribution, which describes the analyst's state of knowledge about possible values of

^a A set of information made up of similar constituents. A homogeneous population is one in which each item is of the same type.

p before data are collected, to the posterior distribution, which reflects how the data have altered the analyst's state of knowledge.

Binomial Inference with Conjugate Prior—The simplest type of prior distribution from the standpoint of the mathematics of Bayesian inference is the so-called **conjugate** prior, in which the prior and posterior distributions are of the same functional type (e.g., beta, gamma), and the integration in Bayes' Theorem is circumvented. Not every aleatory model will have an associated conjugate prior, but the four most commonly used models do. **For the binomial distribution, the conjugate prior is a beta distribution.**

Two parameters are needed to describe the beta prior distribution completely, and these are denoted α_{prior} and β_{prior} . Conceptually, α_{prior} can be thought of as the number of failures contained in the prior distribution, and the sum of α_{prior} and β_{prior} is like the number of demands over which these failures occurred. Thus, small values of α_{prior} and β_{prior} correspond to less information, and this translates into a broader, more diffuse prior distribution.

With the **data consisting of x failures in n demands**, the conjugate nature of the prior distribution and likelihood function allows the posterior distribution to be determined using arithmetic. The posterior distribution is also a beta distribution, with newly adjusted (labeled "post") parameters given by:

$$\alpha_{\text{post}} = \alpha_{\text{prior}} + x$$

$$\beta_{\text{post}} = \beta_{\text{prior}} + n - x.$$

From the properties of the beta distribution, the prior and posterior mean of p are given by:

$$\text{Prior mean} = \alpha_{\text{prior}} / (\alpha_{\text{prior}} + \beta_{\text{prior}})$$

$$\text{Posterior mean} = \alpha_{\text{post}} / (\alpha_{\text{post}} + \beta_{\text{post}})$$

Credible intervals (e.g., see Figure C-1) for either the prior or the posterior can be found using the BETAINV() function built into modern spreadsheets.

Credible Interval Bayesian inference produces a probability distribution. The "credible interval" consists of the values at a set (one low, one high) of specified percentiles from the resultant distribution. For example, a 90% credible interval ranges from the value of the 5th percentile to the value of the 95th percentile.

Percentile A percentile, p, is a specific value x such that approximately p% of the uncertainty is lower than x and (100-p)% of the uncertainty is larger than x. Common percentiles used in PRA include the lower-bound (5th percentile), the median (50th percentile), and upper-bound (95th percentile).

In summary, for conjugate distributions (e.g., beta prior when using a binomial aleatory model), we can solve the Bayesian inference problem by:

1. Knowing that the posterior distribution is the same type, but with “updated” parameters, as the prior.
2. Numerically integrating (via available software tools, e.g., Mathematica, Maple) with the applicable prior distribution and aleatory model. Note that when using a conjugate prior, numerical integration is not needed since the posterior can be found directly (using the equations above), but numerical integration is a general method for Bayesian inference.
3. Numerically simulating using Markov Chain Monte Carlo (MCMC) techniques (via software tools, e.g., OpenBUGS) with the applicable prior distribution and aleatory model. Note that when using a conjugate prior, numerical simulation is not needed since the posterior can be found directly, but numerical simulation is a general method for Bayesian inference.

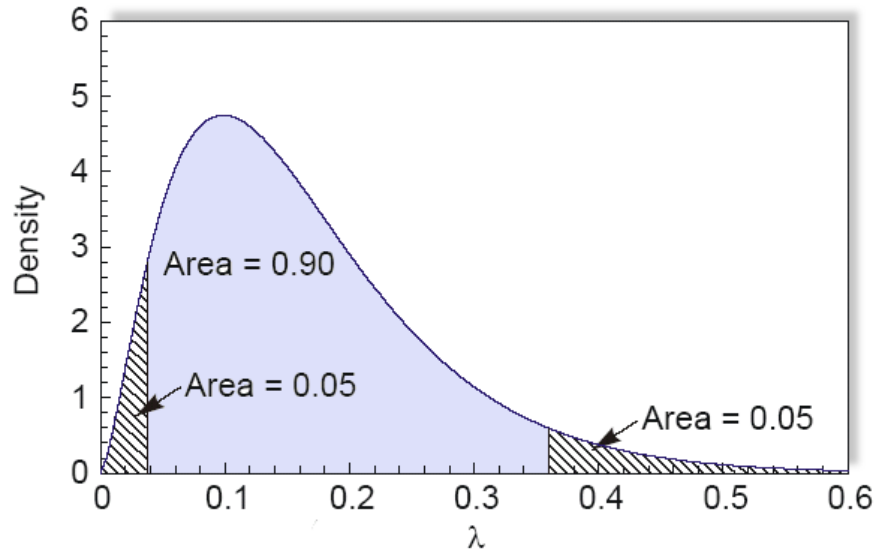


Figure C-1. Representation of a Probability Distribution (epistemic uncertainty), Where the 90% Credible Interval (0.04 to 0.36) is Shown.

We demonstrated (above) method #1 for the beta/binomial case by noting that the posterior is a beta distribution with parameters $\alpha_{\text{post}} = \alpha_{\text{prior}} + x$ and $\beta_{\text{post}} = \beta_{\text{prior}} + n - x$.

Example 1. Relief valve fails to open (binomial model and beta prior).

The prior distribution for failure of a relief valve to open on demand is given (from an industry database) as a beta distribution with:

- $\alpha_{\text{prior}} = 1.24$
- $\beta_{\text{prior}} = 189,075$.

Assume two failures to open have been seen in 285 demands. Find the posterior mean of p , the probability that the valve fails to open on demand and a 90% credible interval for p .

Solution

We begin by noting that the mean of the beta prior distribution is $1.24/(1.24 + 189,075) = 6.56 \times 10^{-6}$. Because α_{prior} is relatively small, the prior distribution expresses significant epistemic uncertainty about the value of p . This can be quantified by calculating a 90% credible interval for p based on the prior distribution. We use the BETAINV() function to do this. The 5th percentile of the prior distribution is given by $\text{BETAINV}(0.05, 1.24, 189075) = 5.4 \times 10^{-7}$ and the

95th percentile is given by $\text{BETAINV}(0.95, 1.24, 189075) = 1.8 \times 10^{-5}$, a spread of almost two orders of magnitude.

With two failures to open in 285 demands of the valve, and the assumption that these failures are described adequately by a binomial distribution, the posterior distribution is also a beta distribution, with parameters $\alpha_{\text{post}} = 1.24 + 2 = 3.24$ and $\beta_{\text{post}} = 189,075 + 285 - 2 = 189,226$. The posterior mean of p is given by $3.24/(3.24 + 189,226) = 1.7 \times 10^{-5}$. The 90% posterior credible interval is found using the $\text{BETAINV}()$ function, just as was done for the prior interval above. The posterior 5th percentile is given by:

$$\text{BETAINV}(0.05, 3.24, 189226) = 5.0 \times 10^{-6}$$

and the 95th percentile is given by:

$\text{BETAINV}(0.95, 3.24, 189226) = 3.5 \times 10^{-5}$. Note how the epistemic uncertainty in the prior distribution has been reduced by the observed data. This is shown graphically in Figure C-2, which overlays the prior and posterior distribution for this example.

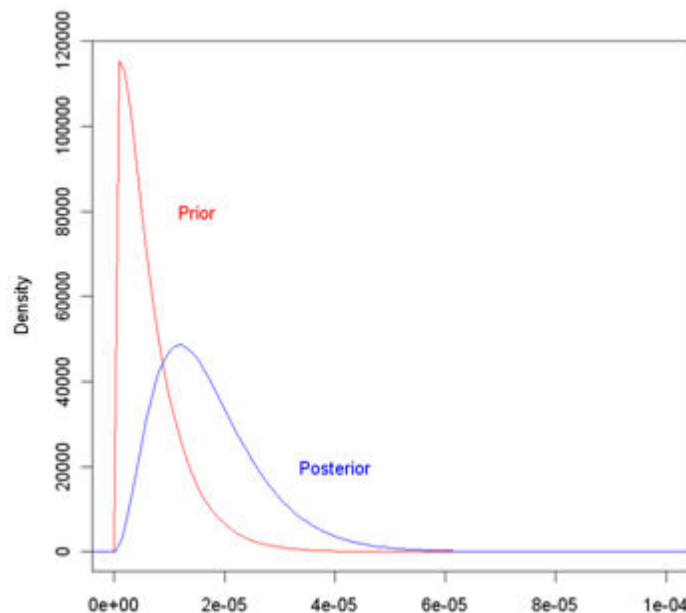


Figure C-2. Comparison of Prior and Posterior Distributions for Example 1.

Inference for conjugate cases can also be carried out using MCMC approaches (such as with OpenBUGS). Script 1 implements the binomial/beta conjugate analysis. For problems such as this, where there is only one unknown parameter to be estimated, the analyst should use 100,000 iterations, discarding the first 1,000 to allow for convergence to the posterior distribution. **Monitoring node p** will display the desired posterior results.

Within Script 1 (and the remaining scripts), the following notation is used:

- “~” indicates that the variable to the left of “~” is distributed as the *distribution* on the right of “~.” Examples of distributions include binomial (dbin), beta (dbeta), gamma (dgamma), Poisson (dpois), normal (dnorm), and lognormal (dlnorm).
- “#” indicates that the text to the right of “#” is a comment.

- “<-” indicates that the variable to the left of “<-” is equivalent to the expression on the right of “<-.”

```

model {                                # A Model is defined between { } symbols
x ~ dbin(p, n)                          # Binomial dist. for number of failures in n demands
p ~ dbeta(alpha.prior, beta.prior)    # Conjugate beta prior distribution for p
}

data
list(x=2, n =285)                        # Data for Example 1
list(alpha.prior=1.24, beta.prior=189075) # Prior parameters for Example 1

```

Script 1. WinBUGS script for Bayesian inference with binomial likelihood and beta conjugate prior.

A directed acyclic graph (DAG) is a common way of displaying a Bayesian inference problem and is the underlying model used by WinBUGS (in script form). In a DAG, observed aleatory variables (such as x for the binomial inference problem above) are displayed as ovals that contain no children nodes (i.e., the “lowest” level in the diagram). Uncertain variables that influence x are shown at a higher level in the DAG, and are connected by arrows to the variables they influence (i.e., they are parents of the node they influence). Constant parameters (such as n above) are also shown in the DAG as diamonds. We will display the DAG associated with each WinBUGS script in this document; however, it is not necessary to develop the DAG, as WinBUGS uses the script representation for its analysis.

For relatively simple problems, a DAG can be an aid in understanding the problem, particularly for an analyst who is new to WinBUGS. However, as the complexity of the problem increases, most analysts will find that the script representation of the problem is clearer. We will use the following conventions for DAGs in this document. Note that all variables, which are referred to as *nodes* by WinBUGS, can be scalars, vectors, matrices, or arrays.

- Ovals represent stochastic variables whose uncertainty (either aleatory or epistemic) is represented by a probability distribution.
- Diamonds represent constant parameters (no uncertainty).
- Rectangles represent calculated parameters. As such, their probability distribution is not specified by the analyst but is calculated by WinBUGS from an equation within the script.
- Dashed lines are sometimes used for clarification when certain parameters are entered or calculated in the script as part of other nodes.
 - In cases where the node is used as inference, the arrow will be connected to the dashed symbol.
 - In cases where the parameter within the node is used as inference, the arrow will be connected to the symbol within the dashed node.

Figure C-3 shows the DAG corresponding to the WinBUGS Script 1. This DAG illustrates that x is the observed variable, because it is a node with no children node. This node (x) is an uncertain variable, indicated by its **oval** shape. Its value is influenced by p (p is a parent node to x), which is the parameter of interest in this problem; we observe x (with n specified), and use this information to infer possible values for p . The dashed region at the top of the DAG, labeled “Beta Prior,” clarifies the type of **prior** distribution used for p , and indicates that the parameters of this distribution (alpha and beta) are entered by the analyst.

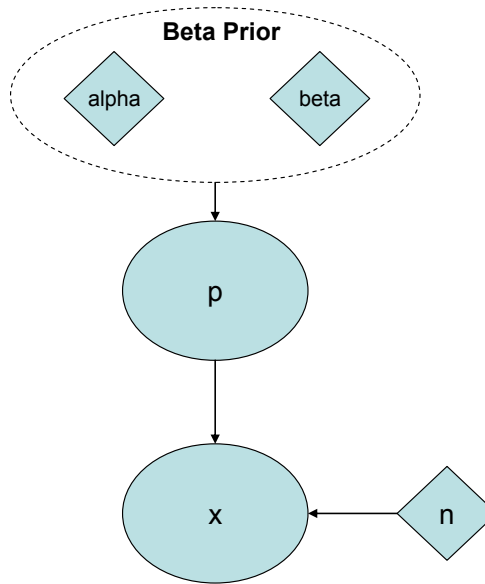


Figure C-3. DAG representing Script 1.

Binomial Inference with Noninformative Prior—As the name suggests, a **noninformative** prior distribution contains little information about the parameter of interest, which in this case is p . Such priors originated in a (continuing) quest to find a mathematical representation of complete uncertainty. This has led some to conclude that they should be used when one knows nothing about the parameter being estimated. As discussed in earlier sections, this is almost never the case in practice, and use of a noninformative prior in such a case can lead to excessively conservative results. Therefore, there are two situations in which a noninformative prior may be useful:

1. The first is where the observed data are abundant enough to dominate the information contained in any reasonable prior, so it does not make sense to expend resources developing an informative prior distribution.
2. The second is where the analyst wishes to use a prior that has little influence on the posterior, perhaps as a point of reference.

The most common noninformative prior for single-parameter inference in PRA is the **Jeffreys** prior.

The Jeffreys functional form is dependent upon the likelihood function, so there is not a single “Jeffreys prior” for all cases. Instead, there is a different Jeffreys prior for each likelihood function. For the case here, where the likelihood function is the binomial distribution, the Jeffreys prior is a beta distribution with both parameters equal to 0.5. Thus, inference with the Jeffreys prior is a special case of inference with a beta conjugate prior. Using the Jeffreys prior with the binomial model leads to a posterior mean of $(x + 0.5) / (n + 1)$.

Note that if x and n are small (sparse data), then adding “half a failure” to x may give a result that is felt to be too conservative. In such cases, a possible alternative to the Jeffreys prior is a beta distribution with both parameters equal to zero (the “zero-zero” beta distribution). This is

not a proper probability distribution, but as long as x and n are greater than zero, the posterior distribution will be proper and the posterior mean will be x / n .

Conceptually, adjusting the beta prior so that α_{prior} and β_{prior} both have small values (in the limit, zero) tends to reduce the impact of the prior and allows the data to dominate the results. Note, though, that when α_{prior} and β_{prior} are equal, the mean of this beta prior is 0.5. The prior should reflect what information, if any, is known independent of the data.

Binomial Inference with Nonconjugate Prior—A nonconjugate prior is one in which the prior and posterior distribution are not of the same functional form. In such cases, numerical integration is required for the denominator of Bayes' Theorem. In the past, this has been a limitation of Bayesian inference, and is one reason for the popularity of conjugate priors. However, cases often arise in which a nonconjugate prior is desirable, despite the increased mathematical difficulty. As an example, generic databases often express epistemic uncertainty in terms of a lognormal distribution, which is not conjugate with the binomial likelihood function. In this section, we describe how to carry out inference with a lognormal prior, which is a commonly-encountered nonconjugate prior, and with a logistic-normal prior, which is similar to a lognormal prior but is more appropriate when the values of p are expected to be closer to one.

Although spreadsheets can be used to carry out the required numerical integration for the case of a single unknown parameter, another way to deal with nonconjugate priors is with WinBUGS. We illustrate the case of a lognormal prior with the following example.

Example 2. Relief valve fails to open (binomial model and lognormal prior).

Continuing with the relief valve from Example 1, assume that instead of the conjugate prior in that example, we are using a generic database that provides a lognormal prior for p .

Assume the generic database lists the mean failure probability as 10^{-6} with an error factor of 10. As in Example 1, assume that our observed data are two failures in 285 demands. The WinBUGS script shown below is used to analyze this example.

```

model {
x ~ dbin(p, n)                # Binomial model for number of failures
p ~ dlnorm(mu, tau)          # Lognormal prior distribution for p
tau <- 1/pow(log(prior.EF)/1.645, 2) # Calculate tau from lognormal error factor
# Calculate mu from lognormal prior mean and error factor
mu <- log(prior.mean) - pow(log(prior.EF) / 1.645, 2) / 2
}
data
list(x=2, n=285, prior.mean=1.E-6, prior.EF=10)

```

Script 2. WinBUGS script for Bayesian inference with binomial likelihood function and lognormal prior.

Solution

Running this script for 100,000 iterations, discarding the first 1,000 iterations to allow for convergence to the posterior distribution, gives a posterior mean for p of 4.7×10^{-5} and a 90% credible interval of $(1.9 \times 10^{-6}, 1.8 \times 10^{-4})$. Note that when the prior distribution is not conjugate, the posterior distribution cannot be written down in closed form. In such cases, an analyst may replace the numerically defined posterior with a distribution of a particular functional form (e.g., lognormal), or may use the empirical results of the WinBUGS analysis to construct a histogram.

Generic databases may not always describe the lognormal distribution in terms of a mean value and an error factor; quite often the median (50th percentile) is specified rather than the mean value. This may also be the case when eliciting information from experts as an expert may be more comfortable providing a median value. In this case, the analysis changes only slightly. In Script 2, the line that calculates μ from the lognormal prior mean and error factor is replaced by the following line:

```
mu <- log(prior.median)
```

and `prior.median` is loaded in the data statement instead of `prior.mean`.

Cases may arise where the value of p could be approaching unity. In such cases, using a lognormal prior is problematic because it allows for values of p greater than unity, which is not meaningful since p is a probability (either failure probability or reliability, depending on the context). In such cases, a logistic-normal prior is a “lognormal-like” distribution, but one that constrains the values of p to lie between zero and one. The WinBUGS Script 3 uses the lognormal mean and error factor (e.g., from a generic database), but “constrains” the distribution to lie between zero and one by replacing the lognormal distribution with a logistic-normal distribution.

```
model {
  x ~ dbin(p, n)                # Binomial distribution for number of failures
  p <- exp(p.constr)/(1 + exp(p.constr)) # Logistic-normal prior distribution for p
  p.constr ~ dnorm(mu, tau)
  tau <- 1/pow(log(prior.EF)/1.645, 2) # Calculate tau from lognormal error factor

  # Calculate mu from lognormal prior mean and error factor
  mu <- log(prior.mean) - pow(log(prior.EF)/1.645, 2)/2
}

data
list(x=2,n=256, prior.mean=1.E-6, prior.EF=10)
```

Script 3. WinBUGS script for Bayesian inference with binomial likelihood function and logistic-normal prior.

Poisson Distribution for Initiating Events or Failures in Time

The Poisson model is often used for failures of normally operating components, failures of standby components that occur at some point in time prior to a demand for the component to

change state, and for initiating events. The following assumptions underlie the Poisson distribution:

- ☑ The probability of an event (e.g., a failure) in a small time interval is approximately proportional to the length of the interval. The constant of proportionality is denoted by lambda.
- ☑ The probability of simultaneous events in a short interval of time is approximately zero.
- ☑ The occurrence of an event in one time interval does not affect the probability of occurrence in another, non-overlapping time interval.

The unknown parameter in this model is lambda, and the observed data are the number of events, denoted x , in a specified time period, denoted t . Both x and t are assumed to be known with certainty in this section. (Cases in which x and t also have epistemic uncertainty are treated in Reference [C-1].) Note that the Poisson distribution describes the aleatory uncertainty in the number of failures, x . The Bayesian inference describes how the epistemic uncertainty in lambda changes from the prior distribution, which describes the analyst's state of knowledge about possible values of lambda before empirical data are collected, to the posterior distribution, which reflects how the observed data have altered the analyst's prior state of knowledge.

Poisson Inference with Conjugate Prior—As was the case with the binomial distribution, a conjugate prior is sometimes chosen for purposes of mathematical convenience. For the Poisson distribution, the conjugate prior is a gamma distribution. Two parameters are needed to describe the gamma prior distribution completely, and these are denoted α_{prior} and β_{prior} . Conceptually, α_{prior} can be thought of as the number of events contained in the prior distribution, and β_{prior} is like the period of time over which these events occurred. Thus, small values of α_{prior} and β_{prior} correspond to little information, and this translates into a broader, more diffuse prior distribution for lambda.

With the observed data consisting of x failures in time t , the conjugate nature of the prior distribution and likelihood function allows the posterior distribution to be written down immediately using simple arithmetic: the posterior distribution is also a gamma distribution, with new (adjusted) parameters given by:

$$\alpha_{\text{post}} = \alpha_{\text{prior}} + x$$

$$\beta_{\text{post}} = \beta_{\text{prior}} + t.$$

From the properties of the gamma distribution, the prior and posterior mean of lambda are given by $\alpha_{\text{prior}}/\beta_{\text{prior}}$ and $\alpha_{\text{post}}/\beta_{\text{post}}$, respectively. Credible intervals for either distribution can be found using the GAMMAINV() function built into modern spreadsheets.

Example 3. Circulating pump fails to operate (Poisson model and gamma prior).

The prior distribution for the circulating pump is given as a gamma distribution with parameters $\alpha_{\text{prior}} = 1.6$ and $\beta_{\text{prior}} = 365,000$ hours. No failures are observed in 200 days of operation.

Find the posterior mean and 90% interval for the circulating pump failure rate. Use the posterior mean to find the probability that the pump will operate successfully for a mission time of 1,000 hours.

Solution

Because the gamma prior distribution is conjugate with the Poisson likelihood function, the posterior distribution will also be a gamma distribution, with parameters $\alpha_{\text{post}} = 1.6 + 0 = 1.6$ and $\beta_{\text{post}} = 365,000 \text{ hours} + (200 \text{ days})(24 \text{ hours/day}) = 369,800 \text{ hours}$. The posterior mean is the ratio of α_{post} to β_{post} , which is $4.3 \times 10^{-6}/\text{hour}$.

The 90% credible interval is found using the gamma inverse (GAMMAINV) function. Note that most spreadsheet software uses the reciprocal of beta as the second parameter. This can be dealt with either by entering $1/\beta$ as the argument, or entering one as the argument, and dividing the overall result of the function call by beta. Thus, the 5th percentile is given by either GAMMAINV(0.05, 1.6, $1/369800$) or [GAMMAINV(0.05, 1.6, 1)]/369800. Either way, the answer is $5.6 \times 10^{-7}/\text{hour}$. Similarly, the 95th percentile is given by either GAMMAINV(0.95, 1.6, $1/369800$) or [GAMMAINV(0.95, 1.6, 1)]/369800. The answer either way is $1.1 \times 10^{-5}/\text{hour}$.

Using the posterior mean failure rate of $4.3 \times 10^{-6}/\text{hour}$, the probability that the pump operates successfully for 1,000 hours is just $\exp[-(4.33 \times 10^{-6}/\text{hour})(1000 \text{ hours})] = 0.996$.

The plot below shows how little the prior distribution has been affected by the relatively sparse data in this example.

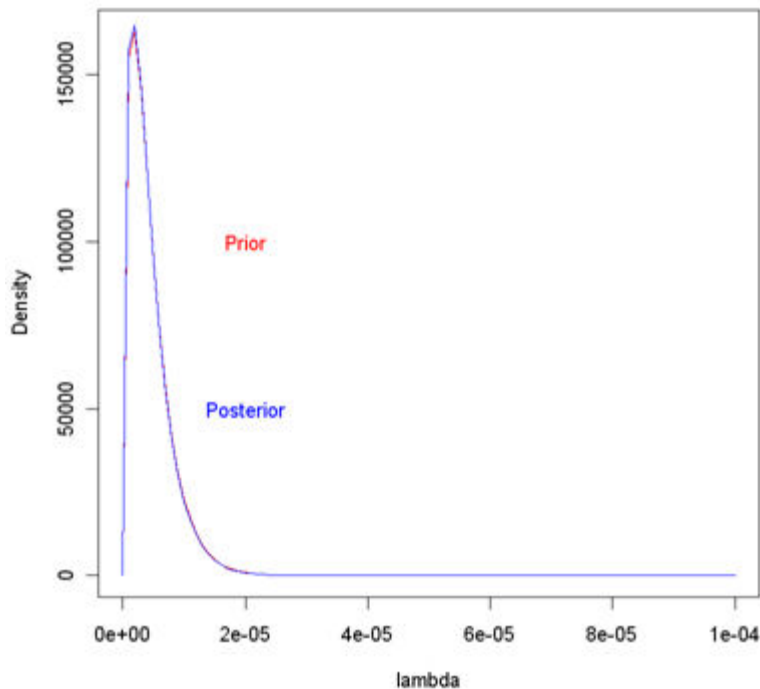


Figure C-4. Comparison of Prior and Posterior Distributions for Example 3.

Bayesian inference can also be carried out using WinBUGS. The script below implements the analysis. Monitoring node lambda will display the desired posterior results.


```

model {
x ~ dpois(mean.poisson)           # Poisson likelihood function
mean.poisson <- lambda*time.hr    # Parameterize in terms of failure rate, lambda
time.hr <- time*24                # Convert days to hours
lambda ~ dgamma(1.6, 365000)      # Gamma prior for lambda
}
data
list(x=0, time=200)

```

Script 4. WinBUGS script for Bayesian inference with Poisson likelihood and gamma conjugate prior.

Poisson Inference with Noninformative Prior—As was the case for the binomial distribution, there are many routes to a noninformative prior for lambda, with the most commonly used one in PRA being the Jeffreys prior. In the case of the Poisson, the Jeffreys noninformative prior is like a gamma distribution with $\alpha_{\text{prior}} = 0.5$ and $\beta_{\text{prior}} = 0$. This is not a proper distribution, as the integral over all values of lambda is not finite. However, it yields a proper *posterior* distribution, with parameters $\alpha_{\text{post}} = x + 0.5$ and $\beta_{\text{post}} = t$. Thus, the posterior mean of lambda is given by $(x + 0.5)/t$.

Note that if x and t are small (sparse data), then adding “half an event” to x may give a result that is felt to be too conservative. In such cases, a possible alternative to the Jeffreys prior is a gamma distribution with both parameters equal to zero. This is not a proper probability distribution, but as long as x and t are greater than zero, the posterior distribution will be proper and the posterior mean will take on the value (x / t) .

Poisson Inference with Nonconjugate Prior—As was the case for the parameter p in the binomial distribution, a lognormal distribution is a commonly encountered nonconjugate prior for lambda in the Poisson distribution. The analysis can be carried out with WinBUGS, exactly as was done for p in the binomial distribution. Here, however, there is no concern about values of lambda greater than one, because lambda is a rate instead of a probability, and can take on any positive value, in principle.

Example 4. Circulating pump fails to operate (Poisson model and lognormal prior).

Assume that the prior distribution for the failure rate of the circulating pump is lognormal with a median of 5×10^{-7} /hour and an error factor of 14. Again, assume the observed data are no failures in 200 days. The WinBUGS script below can be used to find the posterior mean and 90% interval for lambda.

```

model {
x ~ dpois(mean.poisson)           # Poisson distribution for number of events
mean.poisson <- lambda*time.hr    # Poisson parameter
time.hr <- time*24                # Convert days to hours
lambda ~ dlnorm(mu, tau)          # Lognormal prior distribution for lambda
tau <- 1/pow( log(prior.EF)/1.645, 2) # Calculate tau from lognormal error factor
mu <- log(prior.median)           # Calculate mu from lognormal median
}

data
list(x=0, time=200, prior.median=5.E-7, prior.EF=14)

```

Script 5. WinBUGS script for Bayesian inference with Poisson likelihood function and lognormal prior.

Running this script for 100,000 iterations, discarding the first 1,000 iterations to allow for convergence, gives a posterior mean for lambda of 1.6×10^{-6} /hour and a 90% credible interval of (3.5×10^{-8} /hour, 6.5×10^{-6} /hour).

Exponential Distribution for Random Durations

There are cases where the times at which random events occur are observed, instead of the number of such events in a specified period of time. Examples are times to failures of components, times to suppress a fire, etc. If the assumptions for the Poisson distribution are met, then the times between events are exponentially distributed with unknown parameter lambda; this is the same lambda that appears as the unknown parameter in the Poisson distribution.

The following assumptions underlie the Exponential distribution:

- The probability of an event (e.g., a failure) in a small time interval is approximately proportional to the length of the interval. The constant of proportionality is denoted by lambda.
- The probability of simultaneous events in a short interval of time is approximately zero.
- The occurrence of an event in one time interval does not affect the probability of occurrence in another, non-overlapping time interval.
- The random event that is observed is the time to an event.

Because the observed data consist of n number of failure times (with n specified), the form of the likelihood function changes from a Poisson distribution to a product of n exponential distributions. However, much of the analysis is very similar to the analysis done for the Poisson distribution. In this section we treat only the case in which all failure times are observed and known with certainty.

Exponential Inference with Conjugate Prior— As was the case for the Poisson distribution, the conjugate prior for the exponential likelihood is again a gamma distribution, with parameters denoted α_{prior} and β_{prior} . Once again, β_{prior} has units of time, and these units must match the units of the observed times that constitute the data. The posterior distribution will again be a gamma distribution with parameters $\alpha_{\text{post}} = \alpha_{\text{prior}} + n$ (the

number of observed times), and $\beta_{\text{post}} = \beta_{\text{prior}} + t_{\text{total}}$, where t_{total} is the sum of the observed times. From the properties of the gamma distribution, the prior and posterior mean of lambda are given by $\alpha_{\text{prior}} / \beta_{\text{prior}}$ and $\alpha_{\text{post}} / \beta_{\text{post}}$, respectively. Credible intervals for either distribution can be found using the GAMMAINV() function built into modern spreadsheets.

Example 5. Circulating pump fails to operate (exponential model) and gamma prior.

The following seven times to failure (in hours) have been recorded for ATCS circulating water pumps: 55707, 255092, 56776, 111646, 11358772, 875209, and 68978. Using the gamma prior for lambda from Example 3 find the posterior mean and 90% credible interval for the circulating water pump failure rate lambda.

Solution

The prior distribution was given in Example 3 as gamma with $\alpha_{\text{prior}} = 1.6$ and $\beta_{\text{prior}} = 365,000$ hours. In this example, we have $n = 7$ and $t_{\text{total}} = 12782181$ hours. Thus, the posterior distribution is gamma with parameters $\alpha_{\text{post}} = 1.6 + 7 = 8.6$ and $\beta_{\text{post}} = 365000 \text{ hours} + 12782181 \text{ hours} = 13147181 \text{ hours}$. The posterior mean is given by $\alpha_{\text{post}}/\beta_{\text{post}} = 6.5 \times 10^{-7}$ /hour. The 5th percentile is given by $[\text{GAMMAINV}(0.05, 8.6, 1)]/13147181 \text{ hours} = 3.4 \times 10^{-7}$ /hour. The 95th percentile is given by $[\text{GAMMAINV}(0.95, 8.6, 1)]/13147181 \text{ hours} = 1.1 \times 10^{-6}$ /hour.

WinBUGS can also be used for this example. The WinBUGS Script 6 shows how to do this.

```

model {
  for(i in 1:n) {
    time[i] ~ dexp(lambda)      # Exponential likelihood function for n failure times
  }
  lambda ~ dgamma(alpha, beta) # Gamma prior for lambda
}

data                                # Note the nested () for the time array
list(time=c(55707, 255092, 56776, 111646, 11358772, 875209, 68978), n=7, alpha=1.6,
beta=365000)

```

Script 6. WinBUGS script for Bayesian inference with exponential likelihood and gamma conjugate prior.

Exponential Inference with Noninformative Prior—The Jeffreys noninformative prior for the exponential likelihood is like a gamma distribution with both parameters equal to zero. This might seem odd, given the relationship between the exponential and Poisson distributions mentioned above. In fact, it *is* odd that the Jeffreys prior changes, depending on whether one counts failures or observes actual failure times. However, we will not delve into the reasons for this difference and its philosophical implications here. Again, the Jeffreys prior is an improper distribution, but it always results in a proper posterior distribution. The parameters of the posterior distribution will be n and t_{tot} , resulting in a posterior mean of n/t_{tot} . This mean is numerically equal to the frequentist maximum likelihood estimator (MLE), and credible intervals will be numerically equal to confidence intervals from a frequentist analysis of the data.

Exponential Inference with Nonconjugate Prior—Again, the lognormal distribution is a commonly encountered nonconjugate prior for a failure rate. The only thing that changes from the earlier discussion in is the likelihood function, which is now a product of exponential distributions. We again use WinBUGS to carry out the analysis.

Example 6. Circulating pump fails to operate (exponential model and lognormal prior).

Using the prior distribution from Example 4 and the failure times from Example 5, find the posterior mean and 90% interval for the failure rate lambda.

Solution

The WinBUGS script for this example is shown below.

```

model {
  for(i in 1:n) {
    time[i] ~ dexp(lambda)      # Exponential likelihood function for n failure
times
  }
  lambda ~ dlnorm(mu, tau)      # Lognormal prior for lambda
  tau <- 1/pow( log(prior.EF)/1.645, 2) # Calculate tau from lognormal error factor
  mu <- log(prior.median)      # Calculate mu from lognormal mean
}

Data                          # Note the nested () for the time array
list(time=c(55707, 255092, 56776, 111646, 11358772, 875209, 68978), n=7,
prior.median=5.E-7, prior.EF=14)

```

Using 100,000 iterations, with 1,000 burn-in iterations discarded to allow for convergence to the posterior distribution, the posterior mean is found to be 5.5×10^{-7} /hour, with a 90% credible interval of $(2.6 \times 10^{-7}$ /hour, 9.2×10^{-7} /hour).

C.2 Reference

C-1 Bayesian Inference for NASA Probabilistic Risk and Reliability Analysis, NASA/SP-2009-569, June 2009.

Appendix D - Logic-Based PRA Modeling Examples

Two examples of the Boolean logic-based (i.e., using cut sets generated from fault and event trees) PRA modeling process are presented in Sections D.1 and D.2. The first example pertains to a Lunar Base, while the second example is a science mission to another planet.

D.1 PRA Example 1 Problem Description

The first example is intended to demonstrate the classical PRA technique of using small ETs to depict perturbations to the system and failure scenarios. Since much system information is modeled in the FTs, this technique involves a small ET/large fault approach.

The methodology is most suited for steady state type situations, such as a Lunar Base, orbiting space station, or Earth-based facility. The unique characteristic of such applications is that maintenance activities ensure that system components eventually achieve a steady-state availability. In the three situations cited, humans are present and can perform necessary maintenance.

Example 1 addresses:

1. PRA objectives and scope;
2. mission success criteria;
3. end states;
4. system familiarization;
5. initiating event (IE) development;
6. MLDs;
7. other IE development methods;
8. IE screening and grouping;
9. risk scenario development;
10. ESD analysis;
11. system success criteria;
12. ET analysis;
13. FT analysis;
14. data analysis; along with
15. model integration and quantification.

These are the subjects of Sections D.1.1 through D.1.15, respectively.

D.1.1 PRA Objectives and Scope

The mission objectives for the Lunar Base are not the same as the PRA objectives. Mission objectives are:

- Operate a Lunar Base in the Sea of Tranquility with a crew of six for 20 years; and
- Perform science studies.

Success criteria for these objectives are the topic of Section D.1.2.

There are two PRA objectives:

1. Support decisions regarding crew health and safety; plus
2. Identify and prioritize the risks to the Lunar Base.

Crew safety is paramount, although accomplishing science objectives is important to justify the program. It is these two objectives that the PRA will examine.

The extent of the examination is constrained by the programmatic scope imposed on the PRA. For Example 1, the emphasis is on hardware failures, so this is the risk assessment scope. Software failures are not considered. Regarding phenomenological hazards, only energetic internal events (e.g., storage tank or battery explosions and associated missile generation) are assessed.

D.1.2 Mission Success Criteria

Four success criteria are needed to satisfy the mission objectives cited in Section D.1.1. To operate the Lunar Base it is necessary to:

- Maintain a habitable environment for crew working and living on the lunar surface; as well as
- Provide a rescue vehicle (for returning to Earth) in case of catastrophic failure.

Relative to performing science studies, the corresponding mission success criteria are:

- Maintain the science instruments; and
- Transmit data back to Earth.

The baseline science mission time is 20 years.

D.1.3 End States

Section D.1.1 specifies two PRA objectives. With respect to ensuring crew safety, loss of crew (LOC) is a separate end state. Any scenario that results in crew injury or demise engenders the end state, LOC.

The two other end states are:

1. Loss of mission (LOM); and
2. Mission success (OK).

End state, LOM, pertains to the second objective (the science mission). Failure to achieve the baseline mission objectives results if the science program is terminated before its 20-year lifetime expires (e.g., due to an inability to repair or replace vital instrumentation, or loss of capability to transmit data to Earth). If the 20-year mission is accomplished without crew injury, the OK end state ensues.

D.1.4 System Familiarization

This step in the PRA process is the one most often taken for granted. It is extremely difficult to analyze a system without understanding its:

- Composition;
- Operation;
- Design objectives; and

- Failure modes.

This information cannot be obtained solely from a functional diagram or design report. It requires considerable effort to truly understand a system and to be able to model it adequately for the PRA. Often, extensive interaction with mission specialists and system designers is necessary to understand a system in detail sufficient to develop a comprehensive PRA model.

The Lunar Base in Example 1 is intended to illustrate the application of PRA concepts without introducing pedagogically unnecessary complexity. Hence, the Lunar Base being considered has a minimalist design and mission. It consists of seven systems needed for:

1. Environmental control and life support (ECLS);
2. Power generation, storage, and distribution (PW);
3. Command and control (CC);
4. Communication (CM);
5. Fire suppression (FS);
6. Emergency escape (EE); plus
7. Science (SC - e.g., data collection and instrumentation).

The Lunar Base operates continuously with a crew of six. Two people specialize in maintaining the base, leaving four to perform science activities. Normally, a ship from Earth re-supplies the base every 60 days, although base stores are sufficient for 120 days without replenishment. Periodically, a re-supply ship brings a vehicle that will return lunar samples as cargo. The rescue vehicle has a 5-year storage life. As stated in Section D.1.2, the baseline mission duration is 20 years.

Table D-1 is the Lunar Base dependency matrix. Beginning with the ECLS, it is supported by:

- PW (which is required for the system to operate);
- CC (which furnishes overall system control); and
- FS (since the system is vulnerable to fire).

Note that:

- ECLS;
- PW; and
- CC;

afford vital support to all base systems.

The CM System supports only:

- CC; and
- SC.

Support to CC is essential so that the CC System can interface with other base systems and the crew. Relative to SC, the principal dependency on CM is among those crew members performing the science experiments and sample collections.

Only the ECLS depends upon the FS System. Although fire in other areas of the base is a hazard, the designers limited automatic fire suppression to just the ECLS.

Table D-1. Lunar Base Dependency Matrix.

This → Supported by ↓	ECLS	PW	CC	CM	FS	EE	SC
ECLS		X	X	X	X	X	X
PW	X		X	X	X	X	X
CC	X	X		X	X	X	X
CM			X				X
FS	X						
EE							X
SC							

Emergency escape is for the crew, so it is listed as a support for SC in Table D-1. Of course, all crew members would be evacuated and returned to Earth in an emergency. No systems are supported by SC.

D.1.5 Initiating Events Development

Initiating Events (IEs) are the start of a sequence. They are the perturbation or failure that begins the scenario. An important aspect of IE development is that a broad spectrum of candidate events must be considered. This spectrum should extend from likely perturbations to extremely unlikely failures, and the impact of candidate IEs on the system should range from the relatively benign to the catastrophic. The breadth of considerations is essential for ensuring completeness of the IE development process and differs from other types of analyses such as Failure Modes and Effects Criticality Analysis (FMECAs) or Hazard and Operability studies (HAZOPs).

Once a broad list of candidate IEs is developed, the derivation of PRA IEs proceeds through an iterative process involving screening and grouping. The screening process eliminates candidate IEs from further consideration if:

- Their likelihood of occurrence is low (e.g., a candidate IE could be eliminated if its probability of occurrence is negligible relative to other IEs with similar impacts on the system);
- Their impact on the system is too benign to perturb the system into another state; or
- They exceed the scope of the risk assessment.

Grouping (also referred to as binning) combines different IEs into a single, representative IE group if they induce a similar response from the system. When different IEs are combined into a representative group, the frequency or probability of occurrence for the representative group is the sum of the individual frequencies or probabilities of each group member. Since not every member of the group will cause the exact same response from the system, typically the impact of the representative group is modeled as the most severe perturbation caused by individual group members. This technique is conservative.

The primary challenge in developing IEs for an aerospace application is preparing the preliminary list of candidate IEs. For the nuclear industry, standardized lists of IEs have evolved for each plant type. Hence, these can serve as the preliminary list of candidates. However, such lists are unavailable for aerospace applications and, due to the diversity among missions,

it is conceptually difficult to envision a standardized list analogous to those available for nuclear power applications.

It is important to remember that the PRA process is iterative. The list of IEs may not be initially complete, but as the analysis develops it should become complete, exhaustive, and lead to a mutually exclusive set of scenarios.

Due to the importance of considering a broad spectrum of candidate IEs, significant experiential research should be performed. This could include:

- Consulting with individuals or groups who have experience with similar systems or missions;
- Researching background experience;
- Brain-storming;
- Eliciting expert opinion; and
- Performing system simulations.

The vital requisite is to develop a comprehensive list of candidates before the IE screening and grouping process begins. Table D-2 is a perfunctory list of candidate IEs that resulted from a brain-storming session for the Lunar Base example.

Table D-2. Perfunctory List of Candidate IEs.

IE Number	Description
1	Primary O ₂ generation failure
2	Primary CO ₂ removal failure
3	Waste management subsystem failure
4	Power generation failure
5	False alarms (e.g., fire or ECLS failure)
6	Failure to maintain a pressurized environment

D.1.6 Master Logic Diagram for IE Development; Pinch Points

Master logic diagrams (MLDs) (see also section 3.3.1) are graphical representations of system perturbations. They are useful IE development techniques because they facilitate organizing thoughts and ideas into a comprehensive list of candidate IEs. An MLD resembles an FT, but it lacks explicit logic gates. An MLD also differs from an FT in that the initiators defined in the MLD are not necessarily failures or basic events.

Specifically, MLDs are a hierarchical depiction of ways in which system perturbations occur. Typically, these perturbations involve failure to contain (which is especially important for fluid systems), failure to control, and failure to cool or otherwise maintain temperatures within acceptable ranges. An MLD shows the relationship of lower levels of assembly to higher levels of assembly and system function. The top event in each MLD is an end state (e.g., one of the end states established in Section D.1.3). Events that are necessary but not sufficient to cause the top event are enumerated in even more detail as the lower levels of the MLD are developed. For complex missions it may be necessary to develop phase-specific MLDs since threats and initiators may change as the mission progresses.

A key concept in MLD development is the pinch point. Obviously, without some termination criterion an MLD could be developed endlessly. The pinch point is the termination criterion applied to each MLD branch. A pinch point occurs when every lower level of the branch has the same consequence (relative to system response) as the higher levels. Under such conditions, more detailed MLD development will not contribute further insights into IEs capable of causing the end state being investigated. Figure D-1 illustrates the conceptual characteristics of an MLD.

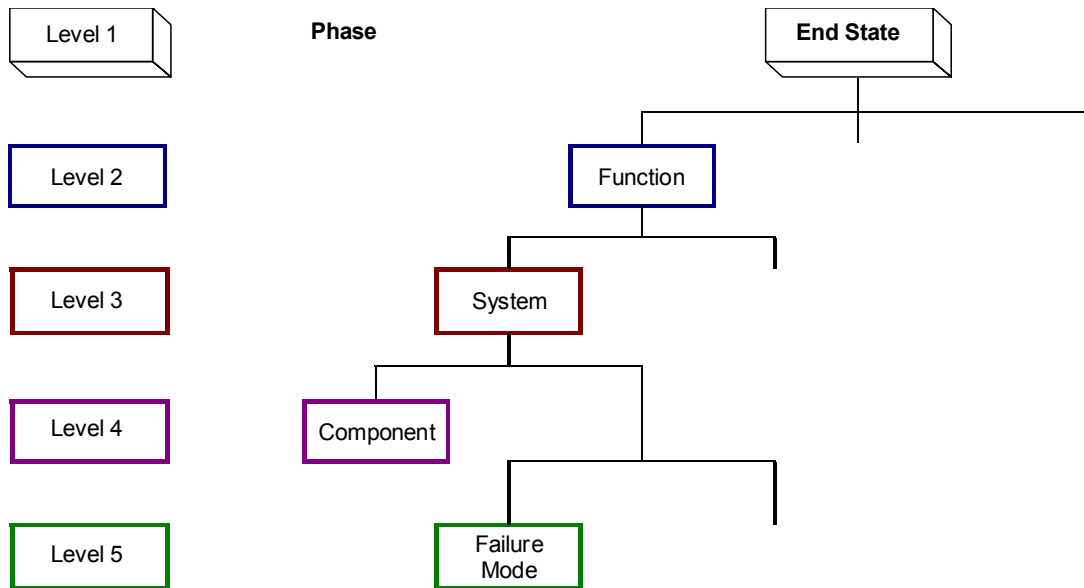


Figure D-1. Conceptual Characteristics of an MLD.

Examination of Figure D-1 discloses that the top event in the MLD is an end state. Typically, the end states of interest are those associated with mission failure (e.g., LOC, or LOM, as discussed in Section D.1.3). At the next level the MLD considers those functions necessary to prevent occurrence of the top event. Relative to the Lunar Base example, end state, LOM, will occur if the lunar base is irreparably damaged. Crew injury would be excluded from an MLD for LOM because this function relates exclusively to end state, LOC (Section D.1.3).

Table D-1 lists the base systems, as they would appear in subsequent levels of a Lunar Base MLD. Below them are the components, while failure modes are addressed in the lowest MLD level. Of course, human errors, phenomenological events, and software errors must be included in the MLD as appropriate. Applying this MLD development technique to the Lunar Base example, Figure D-2 results.

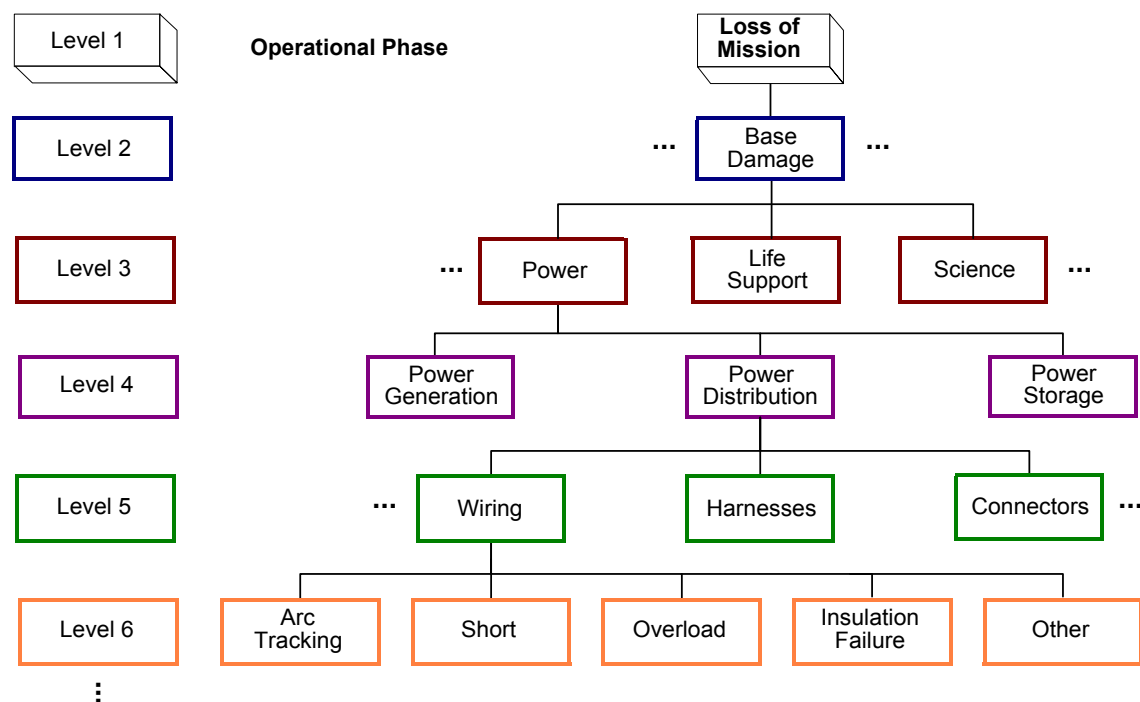


Figure D-2. Lunar Base MLD Extract.

Figure D-2 is only an extract from a larger MLD. The extract suggests that MLDs can be relatively large. This is a valid inference, especially for MLDs during their initial stages of construction.

The only component developed completely in Figure D-2 is the power distribution wiring. Five failure modes are identified:

1. Arc tracking;
2. Short;
3. Overload;
4. Insulation failure; and
5. Others.

The last failure mode is generic and serves as a placeholder in the MLD. Ultimately, it will be necessary to quantify the frequency of wiring failures by examining applicable operating data. If the first four failure modes identified in the MLD dominate wiring faults, the contributions from other failure modes can be ignored as insignificant. However, if the data indicate that there are other significant contributors, they should be included and their impact on the system assessed as part of the iterative process used in MLD development.

An important admonition in IE identification and grouping is that the final set of IEs should be mutually exclusive. Although theoretically IEs with common contributors are amenable to Boolean reduction, many PRA computer codes lack this capability. By ensuring that the IEs are all mutually exclusive, logical inconsistencies internal to the software are avoided.

D.1.7 Other IE Development Methods

The use of FMECAs to support reliability assessments is a fairly standard practice in aerospace and other industries. Although there are a large number of techniques besides MLDs that can be used to develop IEs, FMECAs are emphasized because of their broad utilization. To illustrate IE development using a FMECA, consider the Power Subsystem batteries. Typical battery failure modes include:

- short;
- low voltage;
- rupture or explosion;
- no power; and
- electrolyte leakage.

Table D-3 is an excerpt from a battery FMECA. Relative to candidate IEs, a short is a potentially important failure mode because it can cause loss of one entire side of the Power Subsystem. If retained as an IE, it should not appear in the PW FT to avoid duplicating its contribution to risk. However, if a battery short is not included with the final IE groupings, then it should be incorporated into the PW FT as a basic event.

An alternative to considering a battery short as an IE or FT basic event is to cite the failure cause instead of the failure mode. This has no impact on the model because, as Table D-3 demonstrates, the effects of a battery short and its cause (overcharging, temperature, or wear) are identical.

Rupture is another candidate IE. Although less likely to occur than a short, it is more severe because it is energetic and corrosive enough to cause collateral damage. This illustrates the need to consider both likelihood and consequences when assessing candidate IEs.

Table D-3. Battery FMECA Excerpt.

Item	Phase	Failure Mode	Failure Cause	Failure Effect			Frequency	Severity	Compensating Provisions
				LRU	System	Base			
28 V DC Battery Pack	All	Short	Overcharging, temperature, wear	Loss of 1 battery side, possibly both	Possible loss of second side if protection fails	Possible disruption of service	D	1	Short protection through diodes and breakers, system redundancy
		Low voltage	Loss of cell, under charging	Under voltage condition of 1 side	Possible low power to base	Possible loss of ECLS, CC, and CM	D	1	Low voltage condition should be detected, system redundancy
		Rupture	Overcharging, temperature, wear, physical damage	Battery ruptures	Possible collateral damage	Possible collateral damage to critical components	E	1	Physical separation of critical equipment, barriers around the batteries

D.1.8 IE Screening and Grouping

IE screening and grouping (or binning) should be performed as high in the MLD structure as possible. Ultimately, the binning level depends upon the PRA goals and system response to the candidate IEs. Therefore, some IEs may correspond to relatively high levels in the MLD, while other IEs are low-level pinch points.

An example of binning is afforded by considering a rise in O₂ partial pressure inside of the Lunar Base. This could result from:

- An increase in O₂ flow; or
- A decrease in N₂ flow.

An increase in O₂ flow could be caused by a faulty:

- O₂ sensor (fails low); or
- O₂ regulator (fails high).

Similarly, a decrease in N₂ flow might be due to a faulty:

- N₂ sensor (fails high); or
- N₂ regulator (fails low).

Since all four of these faults cause the same response (a rise in O₂ partial pressure), this event is the pinch point being illustrated, and the sum of the frequencies at which:

- An O₂ sensor fails low;
- An O₂ regulator fails high;
- An N₂ sensor fails high; and
- An N₂ regulator fails low.

is the frequency assigned to the IE group.

D.1.9 Risk Scenario Development

Section 3.3 establishes that a risk scenario begins with an IE that perturbs the system, then progresses through a series of pivotal events to an end state. As was illustrated in Sections D.1.5 through D.1.8, preliminary scenario considerations are fundamental to the IE identification and grouping process. Without such considerations, the relationship between system perturbations (the candidate IEs) and end states cannot be understood adequately to bin the IEs.

Because PRA is an iterative process, these preliminary scenario considerations are revised as the assessment proceeds until a final set of ETs is developed and quantified. After the preliminary IE identification and grouping, the next step in scenario development is to construct an ESD for each IE. This furnishes a more rigorous linkage between the IE and end states. As a result of insights from the ESD construction, some revision to the IE binning may be required. Subsequent to this step, the ESDs are converted into the ETs that are used to quantify event sequences and end states.

D.1.10 ESD Analysis

Four illustrative ESDs are presented for the Lunar Base. They are initiated by:

1. An energetic event;

2. Electrolyte leakage;
3. A smoldering event; and
4. Atmosphere leakage.

The corresponding ESDs are displayed in Figure D-3 through Figure D-6 and addressed in Sections D.1.10.1 through D.1.10.4, respectively.

D.1.10.1 Energetic Event

The energetic event is phenomenological in that it is capable of causing collateral damage (Table D-3). If this hazard causes crew injury, Figure D-3 depicts the event sequence ending with LOC. If there is no crew injury resulting directly from the IE, the subsequent concern is whether critical base equipment survives.

Loss of critical equipment will force a crew evacuation (since a habitable environment cannot be maintained). The end state associated with a successful evacuation is LOM. An unsuccessful evacuation results in LOC.

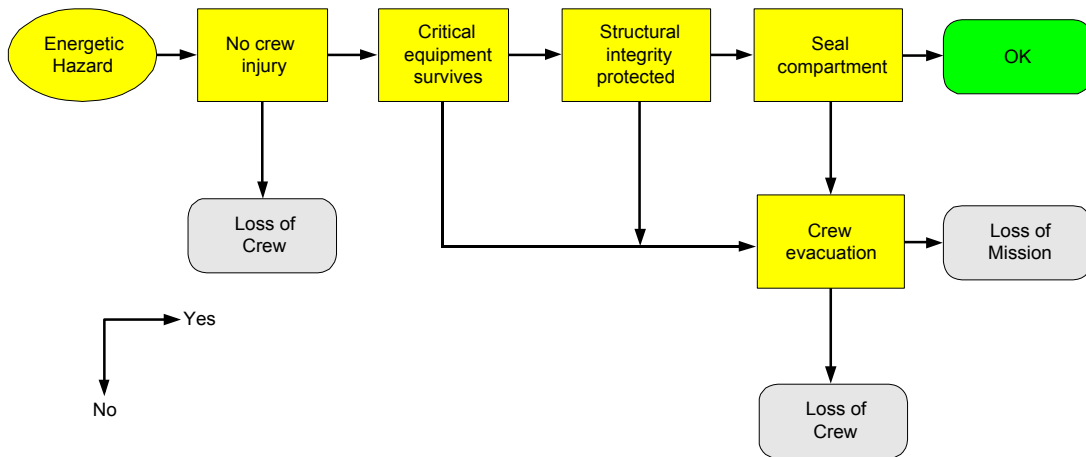


Figure D-3. Energetic Event ESD.

Even if critical equipment survives the IE, structural damage may ensue (equipment and structures are evaluated separately in Figure D-3). Structural damage would engender significant loss of the exterior surfaces, rendering the base uninhabitable. Therefore, it is again necessary to consider crew evacuation with the same end states as described previously.

Should all critical equipment and the base structure survive, damage to the compartment housing the component experiencing the energetic event could entail atmospheric leakage beyond the capacity of the ECLS. If the compartment can be sealed, the base and crew are no longer in jeopardy, so the end state is OK. If the compartment cannot be sealed, crew evacuation once more becomes necessary.

D.1.10.2 Electrolyte Leakage

Electrolyte leakage (Figure D-4) impacts the Power Subsystem, which is a vital support for Lunar Base operation (Table D-1). If electrolyte leakage occurs and critical base equipment fails, crew evacuation is mandated. Successful crew evacuation entails end state LOM, while a failed evacuation causes LOC.

Even if all critical equipment survives the IE, damage to irreplaceable science instruments engenders LOM. If neither critical equipment nor irreplaceable science instruments are damaged, the mission continues. However, if replaceable science instruments are damaged by the IE, a mission delay could be experienced until the instruments are restored.

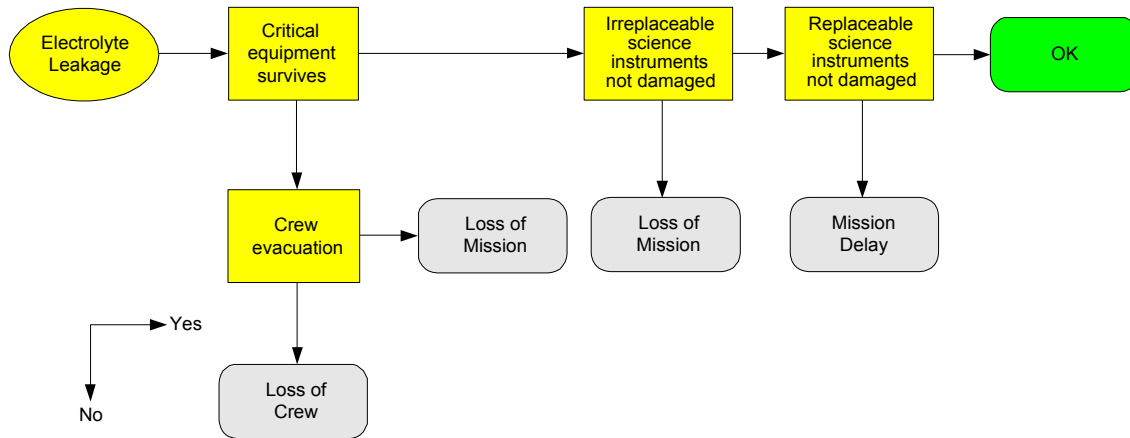


Figure D-4. Electrolyte Leakage ESD.

D.1.10.3 Smoldering Event

A salient consideration with smoldering events is the creation of gaseous toxics. If these toxics are automatically removed from the base atmosphere by ECLS and no shorts occur (which could impact PW), the mission is OK. However, if the toxics are not automatically removed by ECLS, the crew may be able to detect their presence by olfactory sensations or detection equipment. If the toxics are detected and no shorts occur, it is postulated that the crew can remove the toxics and the scenario end state is OK.

Inability to detect the toxics jeopardizes crew health and safety. Depending upon the type and concentration of the toxics, the impact on crew health may be negligible. If not, the end state is LOC. Even without deleterious impacts to the crew, shorts resulting from the IE could cause a loss of base power (either directly, or in combination with other failures). If the shorts are severe enough that critical equipment is unable to function, evacuation is required. If critical equipment remains operational, the mission can continue.

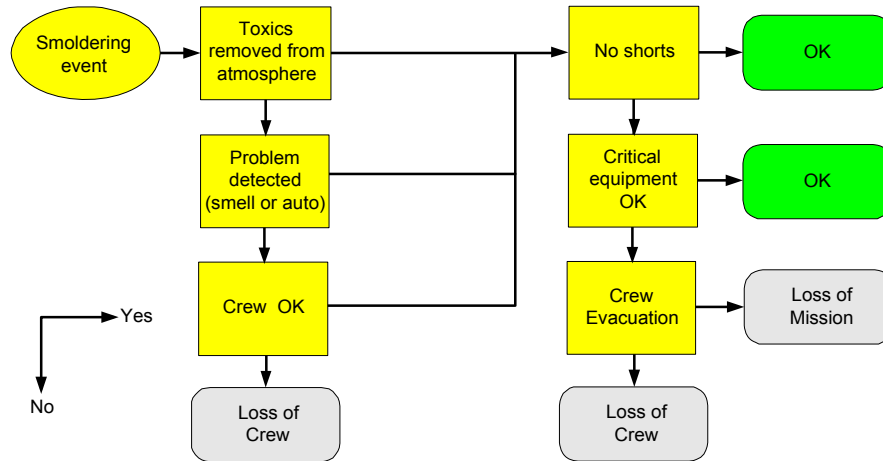


Figure D-5. Smoldering Event ESD.

D.1.10.4 Atmosphere Leakage

Failure to detect atmosphere leakage is conservatively postulated to result in an LOC. If the leak is detected, the planned response is to seal the leaking compartment. Sealing the compartment engenders end state, OK. Inability to seal the leaking compartment requires crew evacuation.

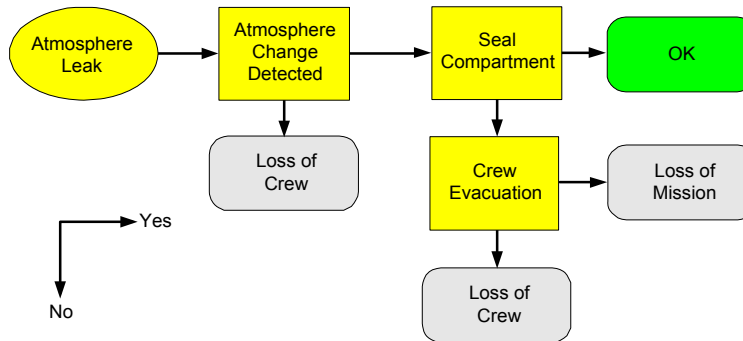


Figure D-6. Atmosphere Leak ESD.

D.1.11 System Success Criteria

Two types of system success criteria are required for a PRA:

1. The minimum number of trains in redundant systems necessary for the system to satisfy its functional requirements (this may depend on the IE); and
2. The time the system must operate.

For the Lunar Base example it is hypothesized that all redundant systems contain two 100% capacity trains. Consequently, both trains must fail in order for the system to fail. This supposition is independent of IE.

The time each system must operate in response to an IE is conservatively modeled as one year. This is not the Lunar Base mission time because once the base recovers from the IE, it returns to an operational state. Instead, the operating time for systems needed to respond to an

IE is predicated upon consideration of how much time is required to recover from the IE and its consequences (i.e., return the base to an operational state).

A standard assumption applied to continuously operating, ground-based facilities is that the system operating time subsequent to an IE is 24 hours. This is predicated upon the knowledge that within 24 hours:

- Personnel from three working shifts will be available at the facility;
- Support personnel (e.g., the engineering staff) can be summoned and will arrive;
- Emergency responders can be summoned if necessary;
- Support utilities (e.g., electrical power and water) are available if required; and
- Within limits, needed replacement equipment can be obtained.

For the Lunar Base, the re-supply interval is two months. Thus, even in an emergency it could require several weeks to a month before additional personnel arrive. Because the re-supply ship has limited capacity (as compared to highway transportation at a ground-based facility), the number of personnel that can be transported per trip is limited. There are no emergency responders, public utilities are unavailable, and any needed equipment must be shuttled to the base from Earth.

Predicated upon these considerations, the base systems may have to operate for several months before it can be assumed, with high confidence, that a normal operating state is restored. Rather than determine whether the number of months is three, four, or six, an entire Earth year was adopted. If such a supposition causes excessively high risk estimates to ensue, a less conservative operating time could be developed and applied.

D.1.12 ET Analysis

Figure D-7 through Figure D-10 are the ETs corresponding to Figure D-3 through Figure D-6, respectively. Relative to the event progression, the ETs and ESDs are equivalent. They each have the same IE, pivotal events, end states, and event sequences. The only difference is in the graphical presentation. As such, the ET descriptions are identical to those for the ESDs in Section D.1.10.

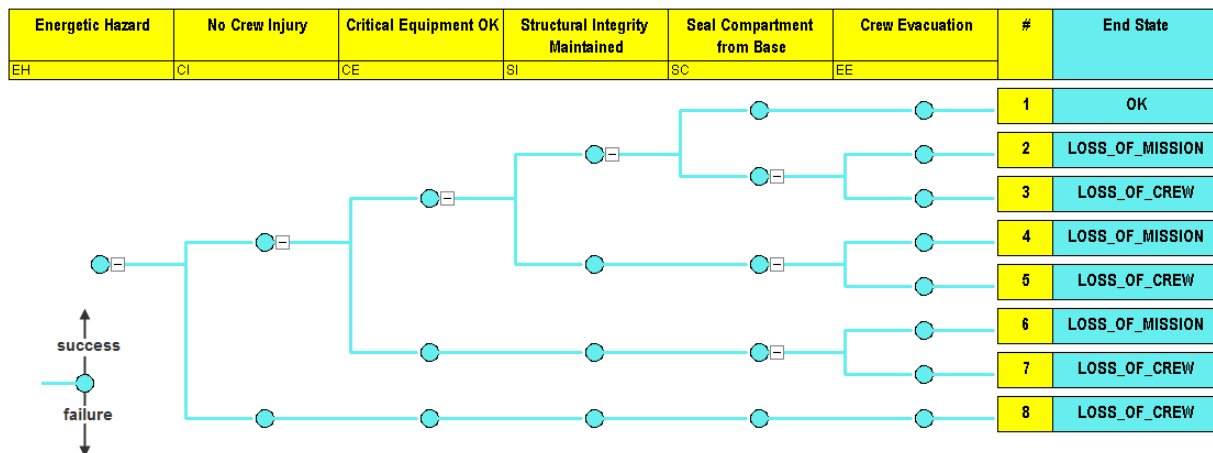


Figure D-7. Energetic Hazard Event Tree.

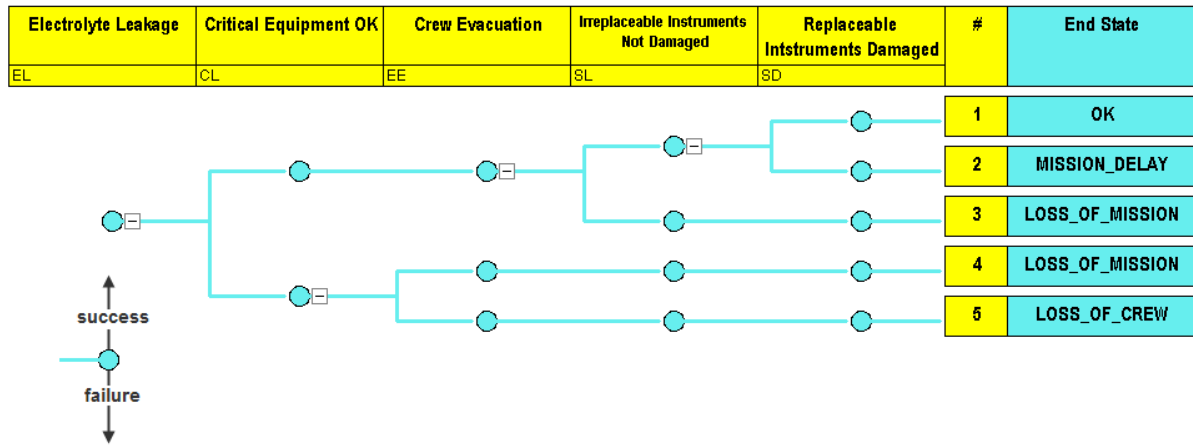


Figure D-8. Electrolyte Leakage Event Tree.

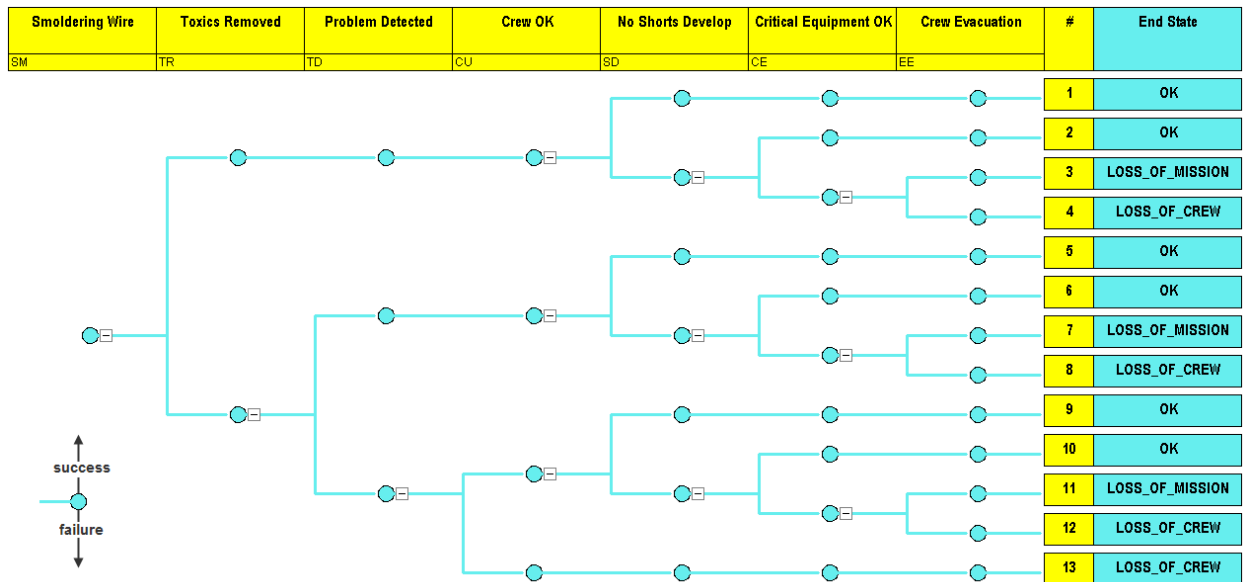


Figure D-9. Event Tree for the Smoldering IE.

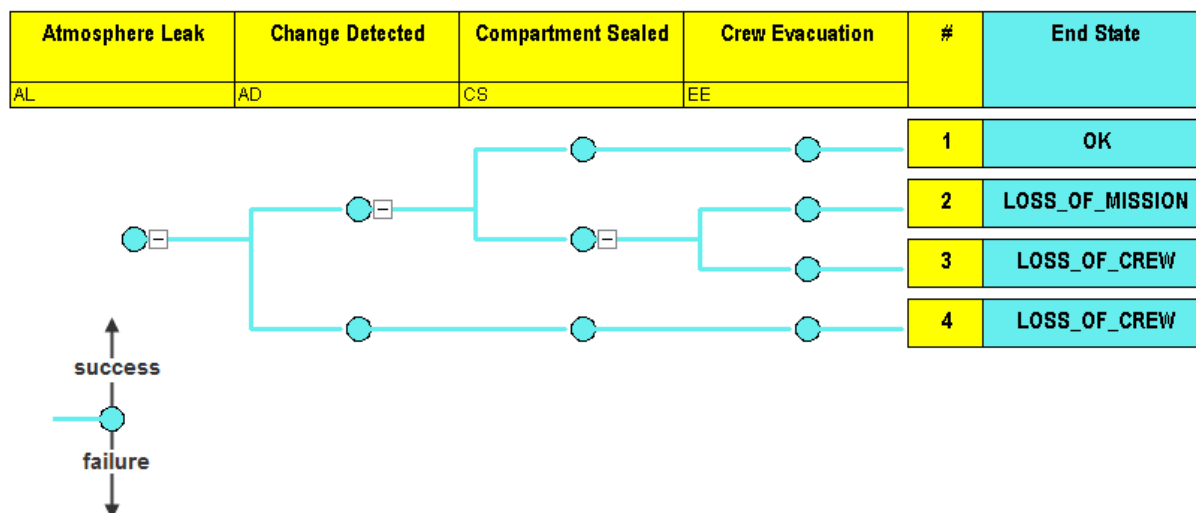


Figure D-10. Atmosphere Leakage Event Tree.

D.1.13 FT Analysis

A key element in the FT construction process is establishing success criteria for the mission or, if the success criteria vary during the mission, determining the success criteria appropriate for each mission phase. For the Lunar Base example, the success criteria are the topic of Section D.1.11.

Top event failure logic is established from the Boolean complement of the success criteria. For instance, if at least one of two power distribution subsystems or antennae must be available to satisfy the success criteria, then failure of both power distribution subsystems or antennae is necessary for system failure. Once the top event logic is established, the FT is constructed by identifying all significant faults that can cause the top event to occur. Typically, this involves failure to contain (which is especially important for fluid systems), failure to control, and failure to cool or otherwise maintain component temperatures within acceptable ranges. Basic events are given specific names to facilitate Boolean reduction and numerical quantification.

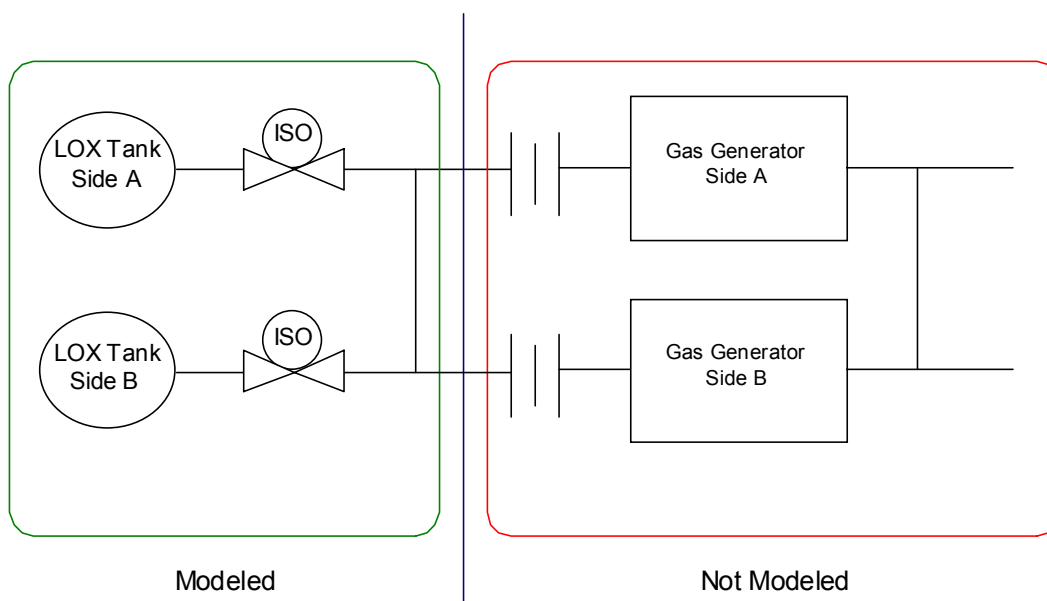
Basic event naming conventions need to be established to ensure consistency among all members of the PRA team, and for compatibility with software requirements. It is also important to establish a naming convention for intermediate gates in the FTs. The basic event naming convention for Example 1 uses:

- Two characters to signify the system;
- Five characters for the identification of a component or collection of components; while
- Two additional characters symbolize the failure mode.

Table D-4 is an example applied to the Lunar Base. The salient admonition is that a convention must be established early in the risk assessment and applied uniformly by all members of the PRA team throughout the analysis. Otherwise, the results can be invalid due to inconsistencies that cause faults in the Boolean reduction or quantification process. In order to illustrate the FT construction process, the Oxygen Supply System will be considered. The assessment will focus on the liquid oxygen tanks and isolation valves, as depicted in Figure D-11. Figure D-11 is part of the Atmosphere Replenishment Subsystem of the Environmental Control and Life Support System.

Table D-4. Naming Convention Example for the Lunar Base.

	Item	Designation
Subsystem ID	Environmental Control and Life Support System	ECLS
	Science Instrumentation System	SI
Component ID	Partial pressure of oxygen sensor number 1	PPO21
	Partial pressure of oxygen sensor number 2	PPO22
	Side A isolation valve	ISOVA
	Side B isolation valve	ISOVB
Failure Mode ID	General failures	FF
	Fails closed	FC
	Independent failure	IF
	Common cause failure	CF
	Independently fails closed	IC

**Figure D-11. Lunar Base Oxygen Supply System.**

The FT for inability to replenish the base atmosphere is exhibited in Figure D-12. This top event occurs if either there is a failure to replenish the:

- O₂; or
- N₂;

(although the example will not examine N₂ replenishment in detail).

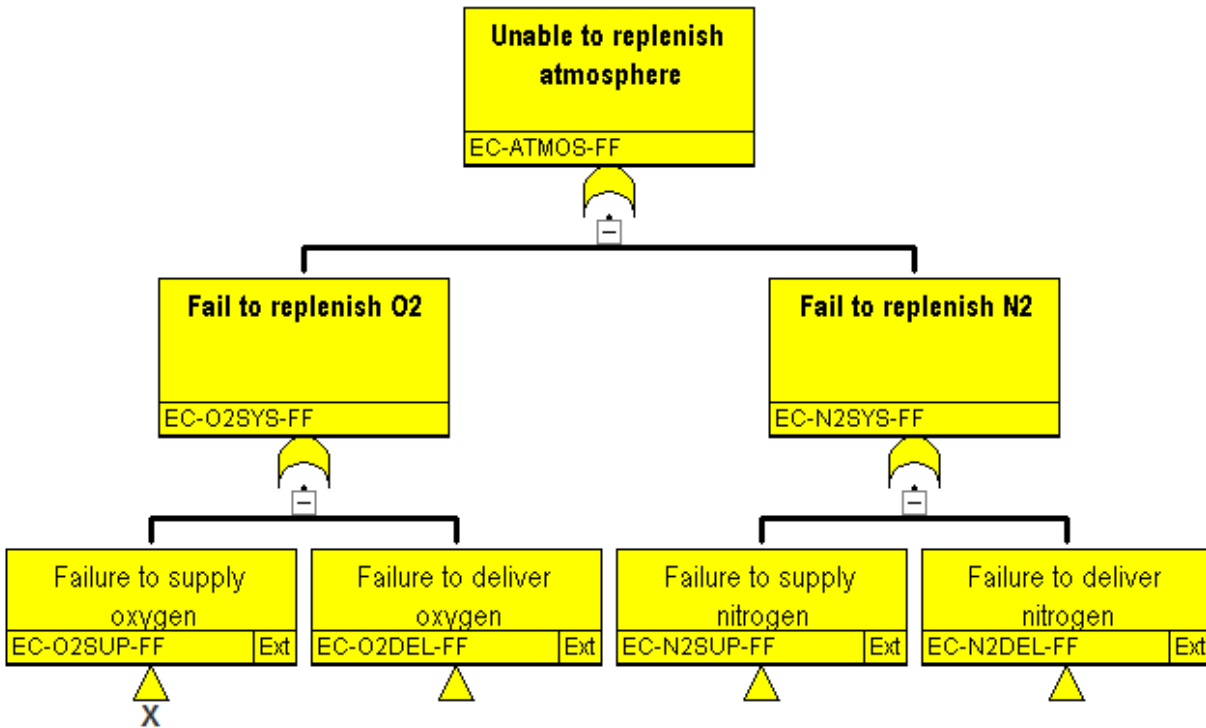


Figure D-12. Fault Tree for Inability To Replenish the Base Atmosphere.

Each of these intermediate events has two causes, either a:

- Supply; or
- Delivery;

failure. These intermediate events, in turn, are connected to other FT logic modules through the triangular off-page connectors.

Following off-page connector X to Figure D-13, notice that failure to supply O₂ requires failure of both sides of the Oxygen Supply System shown in Figure D-11. This, of course, corresponds to a success criterion where either side can fully supply the base O₂ needs. Relative to the portion of the Oxygen Supply System being examined in this example, failure of a particular side results if either the tank or isolation valve fails.

Because the system has redundancy, common cause failures must be considered. Components comprising common cause groups could include:

- Sensors;
- Valves;
- Control computers;
- Pumps;
- Seals; and
- Others, including storage tanks.

Since there are two redundant sides to the system, the beta factor model will be applied (see Chapter 7).

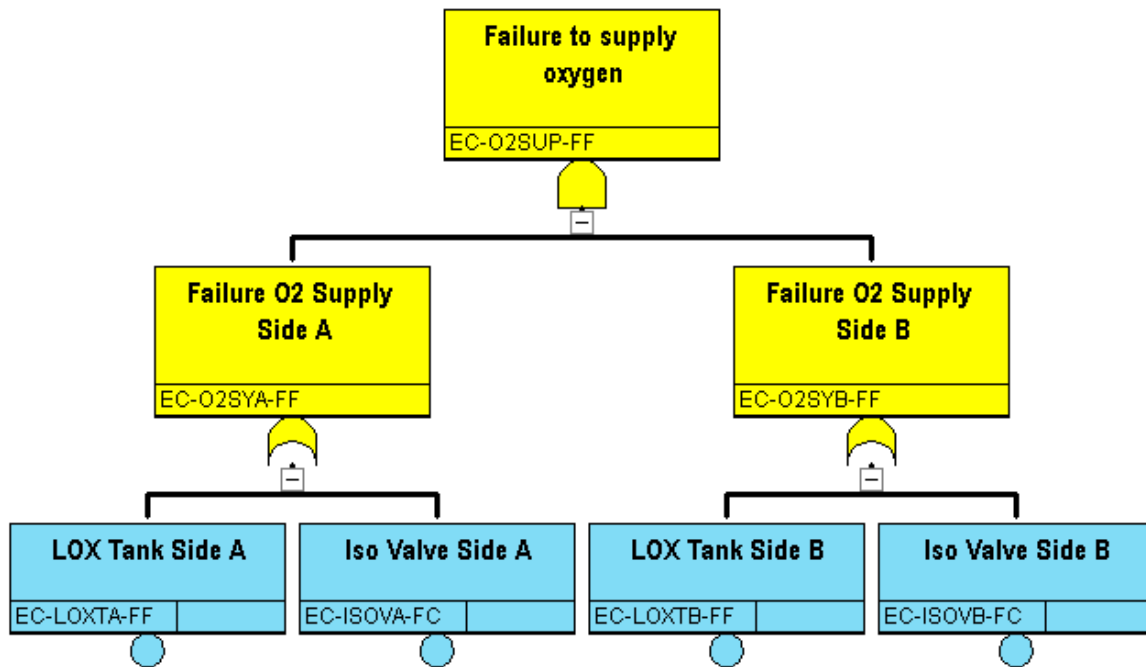


Figure D-13. Fault Tree for Failure To Supply Oxygen.

Before continuing with the tanks and isolation valves in the Oxygen Supply System, it is instructive to digress a moment and examine the partial pressure of oxygen sensors. The intent of this digression is merely to illustrate the construction of FTs pertaining to redundant components other than the mechanical components used to contain or control fluids. If the partial pressure sensors are redundant and either can provide the required sensing function, then both must fail before the capability to sense the oxygen partial pressure is lost. This is symbolized by the AND gate in Figure D-14. Since the sensors comprise a common cause component group, each sensor is modeled as having an independent and common cause failure mode. Because the common cause will fail both sensors, it has the same basic event name in the FT (ECLS-PPO2X-CF).

The Boolean expression for this logic can be derived by noting that

$$EC - PPO21 - FF = (EC - PPO21 - IF) \cup (EC - PPO2X - CF) \quad (D-1)$$

and

$$EC - PPO22 - FF = (EC - PPO22 - IF) \cup (EC - PPO2X - CF) \quad (D-2)$$

which reduces to:

$$EC - PPO2S - FF = [(EC - PPO21 - IF) \cap (EC - PPO22 - IF)] \cup (EC - PPO2X - CF) \quad (D-3)$$

Applying this same concept to off-page connector X in Figure D-12, the resultant FT is depicted in Figure D-15.

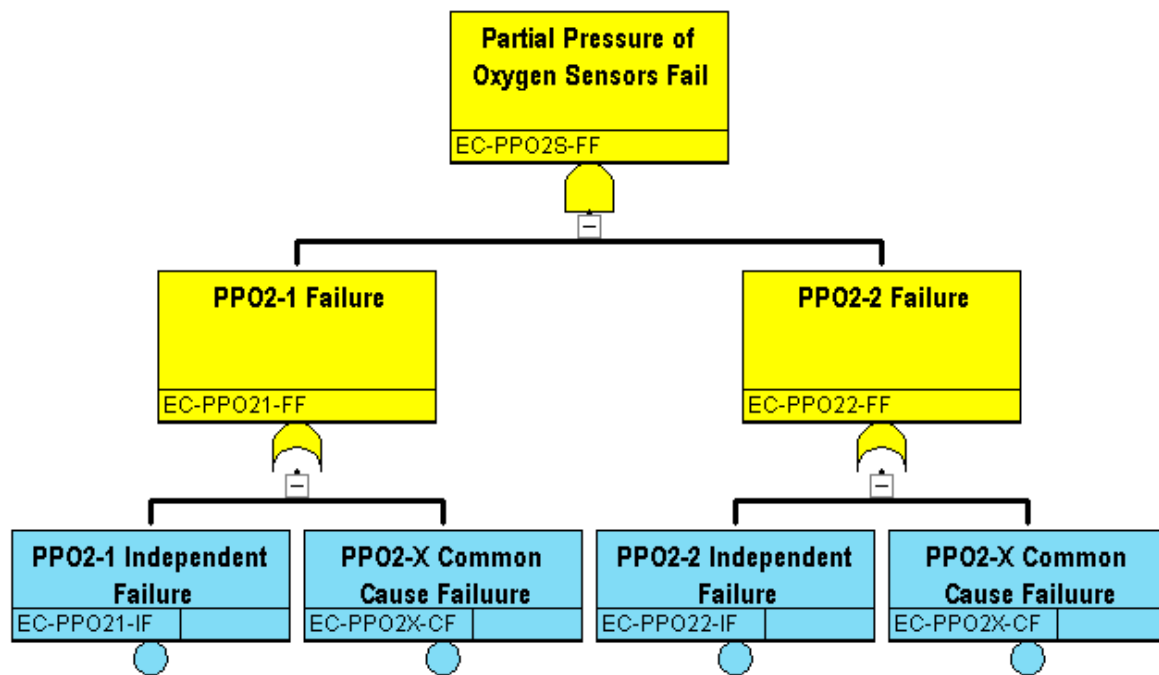


Figure D-14. Fault Tree for Loss of the Partial Pressure of Oxygen Sensors.

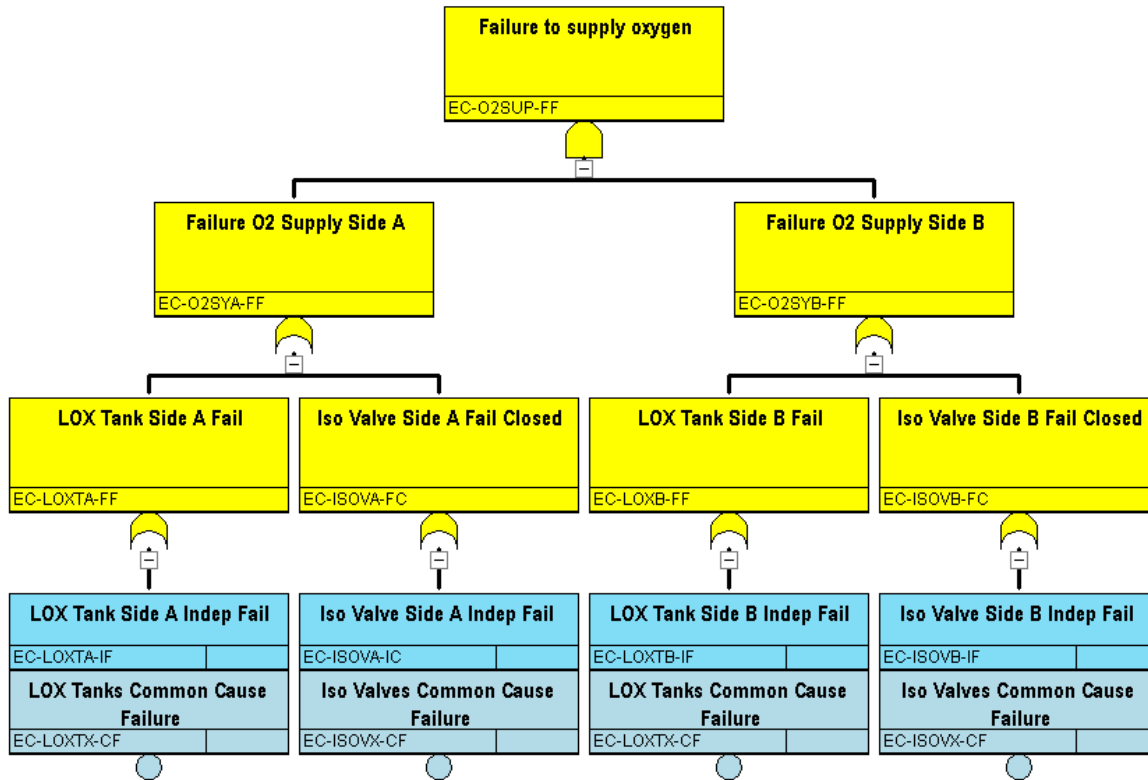


Figure D-15. Final Fault Tree for Failure To Supply Oxygen.

D.1.14 Data Analysis

The fundamentals of data analysis are the subject of Chapter 7. Consequently, the only illustration that will be furnished with the Lunar Base example will focus on common cause failure modeling. Returning to Figure D-14, recall that sensor failure is modeled as resulting from either an:

- Independent; or
- Common;

cause. If their total failure rate is 10^{-6} per hour and beta (the common cause model being applied) is 0.1, then the independent failure rate is 9×10^{-7} per hour, while 1×10^{-7} per hour is the common cause failure (CCF) rate.

D.1.15 Model Integration and Quantification

Once the logic models and database are completed, model integration and quantification can begin. If only a limited scope reliability analysis of one system is being conducted, numerical FT reduction is performed to obtain the minimal cut sets (MCSs). The probability of system failure is then quantified from the union of the MCS. However, if a complete PRA quantification is desired, the MCSs for individual FT top events are obtained and stored. These, of course, correspond to pivotal events in the mission ETs. Boolean algebra and probability theory are then used to quantify the likelihood of each individual event sequence. End state logic is used to determine the likelihood of each particular end state being quantified for the mission.

Figure D-16 illustrates the process. Ultimately, the process involves using Boolean logic to develop expressions for event sequences and end states, then quantifying the event sequences and end states using probability theory.

Suppose that the top event in the FT for failure of pivotal event, TE1, involves basic event, A, combined with either basic event:

- B;
- C; or
- D.

then

$$TE1 = (A \cap B) \cup (A \cap C) \cup (A \cap D) \tag{D-4}$$

if

$$TE2 = (K \cap A) \cup (K \cap D) \tag{D-5}$$

and the reduced Boolean expression representing event sequence 4 is

$$IE \cap TE1 \cap TE2 = (IE \cap A \cap B \cap K) \cup (IE \cap A \cap C \cap K) \cup (IE \cap A \cap D \cap K) \tag{D-6}$$

Once the input data are combined with the MCSs, the basic quantification is complete. This is generally a point estimate (i.e., without uncertainty). At this stage in a PRA, all that remains is to check and interpret the results, and then perform an uncertainty analysis along with the quantification of any other risk metrics (e.g., importance measures) required.

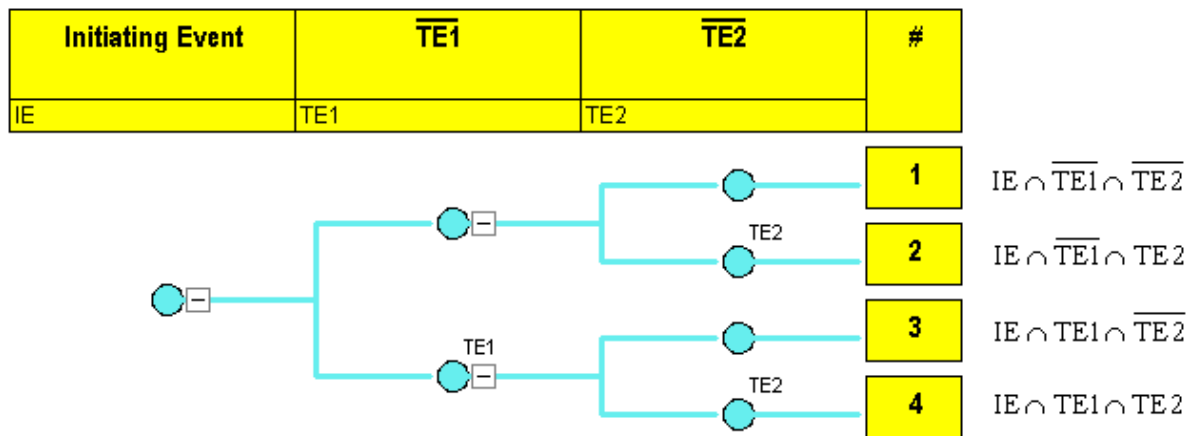


Figure D-16. Quantification of Linked ETs/Fault Trees.

SAPHIRE [D-1] software was used to evaluate the Lunar Base example. Table D-5 is an extract from its input data. The primary name includes the basic event identifier and a brief description. For a Type 1 calculation, SAPHIRE merely uses the input probability value. In a Type 3 calculation, SAPHIRE calculates the basic event probability using an exponential distribution with the failure rate and operating time (Section D.1.11).

Two MCSs resulted from quantifying the FT for failure of the partial pressure of oxygen sensors:

1. CCF; and
2. independent failure;

of both sensors. Table D-6 is the SAPHIRE quantification report for that FT. Since these are the only two MCSs identified, their individual probabilities sum to unity. From Table D-5, note that the probability of a common cause failure equals the input value for the basic event, while the probability for independent failure of both sensors is the product of their independent failure probabilities.

Returning to event sequence 4 in Figure D-9, its occurrence frequency is 1.2×10^{-3} per year (Table D-7). It is initiated by a smoldering wire event, in conjunction with:

- Development of short circuits;
- Critical equipment failure; and
- Crew failure to escape.

Note that the event sequence frequency is the product of the IE frequency combined with the probability of the other basic events in the cut set.

A ranking of the four most dominant contributors to a loss of crew (end state LOC) is exhibited in Table D-8. The interpretation of this SAPHIRE output report is analogous to the previous example. Notice that since the four most dominant sequences have a total frequency of 1.13×10^{-2} per year, they comprise over 99.6% of the total frequency for end state LOC.

The final step in the PRA process involves organizing and interpreting the results. It is imperative to check the results for accuracy. For example, if the independent failure probabilities reported for partial pressure of oxygen (PPO2) sensors 1 and 2 differ, this is indicative of a data entry error.

Similarly, certain groups of failures should be symmetric. Remember that failure of liquid oxygen tank A and failure of isolation valve B will cause an Oxygen Supply System failure. A symmetric failure combination, involving liquid oxygen tank B and isolation valve A, should also appear in the quantification results and have an equivalent probability.

Sanity checks should be performed to determine whether the results are technically reasonable. In the Lunar Base example, the dominant IE relative to a loss of crew is a smoldering wire. It has an assigned frequency of two occurrences per year. A sanity check could confirm whether this is a reasonable value for the event in the Lunar Base environment.

Recalling that PRA is an iterative process, these various checks are simply the last step in the iteration. Of course, if the results disagree with prior expectations, it may not be because the PRA has errors. What it does require is that the basis for this difference be investigated and thoroughly understood.

Due to the large number of cut sets resulting in a PRA, it is advisable to focus primarily on the risk drivers. Also, addressing the risk drivers is often an effective technique for managing mission risks.

Table D-5. Input Data Extract.

Primary Name	Calc Type	Mean Probability	Lambda
ECLS-ARFL1-FF Air filter failure	1	5.000E-002	+0.000E+000
ECLS-ARFL2-FF Air filter failure	1	5.000E-002	+0.000E+000
ECLS-PPO21-IF Partial Pressure of O2 sensor ind. failure	3	7.858E-003	9.000E-007
ECLS-PPO22-IF Partial Pressure of O2 sensor ind. failure	3	7.858E-003	9.000E-007
ECLS-PPO2X-CF Partial Pressure of O2 sensors common cause failure	3	8.762E-004	1.000E-007
EX-CREQE-FF Critical equipment failure	1	3.000E-001	+0.000E+000

Table D-6. SAPHIRE Quantification Report for Failure of Partial Pressure of Oxygen Sensors.

FAULT TREE CUT SETS (QUANTIFICATION) REPORT					
Project		: PRA-COURSE		Analysis	: RANDOM
Fault Tree		: AD		Case	: CURRENT
				Mincut Upper Bound	: 9.379E-004
Cut No.	% Total	% Cut Set	Prob/Freq.	CURRENT CUT SETS	
1	93.5	93.5	8.8E-004	ECLS-PPO2X-CF	
2	100.0	6.5	6.2E-005	ECLS-PPO21-IF, ECLS-PPO22-IF	

Table D-7. Cut Set Report for Event Sequence 4.

Sort/Slice Cut Set Report						
Project-> PRA-COURSE		Event Tree-> SM		Seq-> 04		
Mincut Upper Bound -> 1.200E-003			This Partition -> 1.200E-003			
Cut No.	Total %	Cut set %	Prob/Freq.	Basic Event	Description	Event Prob.
1	100.0	100.0	1.200E-003	SM	Smoldering Wire	2.000E+000
				EX-CREQE-FF	Critical equipment failure	3.000E-001
				PW-SHRTS-FF	Short circuits develop	2.000E-001
				UE-CRESC-FF	Crew fails to escape lunar base	1.000E-002

Table D-8. Cut Set Report for Loss of Crew.

Sort/Slice Cut Set Report						
Project-> PRA-COURSE		End State-> LOC				
Mincut Upper Bound -> 1.134E-002			This Partition -> 1.134E-002			
Cut No.	Total %	Cut set %	Prob/Freq.	Basic Event	Description	Event Prob.
1	44.1	44.1	5.000E-003	SM	Smoldering Wire	2.000E+000
				ECLS-ARFL1-FF	Air filter failure	5.000E-002
				EX-CRTOX-IN	Crew injury due to toxic exposure	5.000E-001
				SI-MASPC-CF	Failure to calibrate mass spectrometer	1.000E-001
2	88.2	44.1	5.000E-003	SM	Smoldering Wire	2.000E+000
				ECLS-ARFL2-FF	Air filter failure	5.000E-002
				EX-CRTOX-IN	Crew injury due to toxic exposure	5.000E-001
				SI-MASPC-CF	Failure to calibrate mass spectrometer	1.000E-001
3	98.8	10.6	1.200E-003	SM	Smoldering Wire	2.000E+000
				EX-CREQE-FF	Critical equipment failure	3.000E-001
				PW-SHRTS-FF	Short circuits develop	2.000E-001
				UE-CRESC-FF	Crew fails to escape lunar base	1.000E-002
4	99.6	0.9	1.000E-004	EH	Energetic Hazard	1.000E-003
				EX-CREWX-EI	Crew injured by energetic debris	1.000E-001

Uncertainty analyses can be performed for:

- FTs;
- Event sequences; and
- End states.

Table D-9 has the uncertainty analysis results for end state LOC in the Lunar Base example. Predicated upon the convergence criteria proposed in Section 12.2, a sample size involving 50,000 iterations was needed. The uncertainty analysis was performed using the SAPHIRE software.

Importance measures can be calculated for:

- FTs;
- Event sequences; and
- End states.

Table D-9. Uncertainty Results for Loss of Crew.

5th Percentile	Median	Mean	95th Percentile
1.5×10^{-3}	6.9×10^{-3}	1.1×10^{-2}	3.5×10^{-2}

Importance measures for the Lunar Base example are displayed in Table D-10. They are the:

- Fussell-Vesely (F-V);
- Risk reduction ratio; and
- Risk increase ratio;

importance measures SAPHIRE calculated for end state LOC. Sorted by F-V importance, the smoldering wire IE ranks the highest. However, IEs characterized by an occurrence frequency should not be examined using conventional risk importance measures. This is because conventional risk importance measures rely on sensitivity studies in which failure probabilities are increased to unity or decreased to zero. Although zero is also the lower bound of an occurrence frequency, IE frequencies have no upper bound. Hence, setting the value of IE event, SM (smoldering wire), to unity actually decreases the IE occurrence frequency. Consequently, even though SAPHIRE includes all basic events in its importance ranking, IEs should be ignored unless they are characterized by a probability of occurrence (instead of a frequency).

Table D-10. Lunar Base Importance Measures.

IMPORTANCE MEASURES REPORT (Current Cut Sets)			
Project	: PRA-COURSE	EndState	:LOC
	Analysis	: RANDOM	
	Case	: CURRENT	
(Sorted by Fussell-Vesely Importance)			
Event Name	Num. of Occ.	Probability of Failure	Fussell- Vesely Importance
SM	5	2.000E+000	9.856E-001
EX-CRTOX-IN	4	5.000E-001	8.797E-001
SI-MASPC-CF	2	1.000E-001	8.796E-001
EC-ARFL2-FF	2	5.000E-002	4.387E-001
EC-ARFL1-FF	2	5.000E-002	4.387E-001
UE-CRESC-FF	7	1.000E-002	1.096E-001
EX-CREQE-FF	2	3.000E-001	1.052E-001
PW-SHRTS-FF	1	2.000E-001	1.049E-001
EH	5	1.000E-003	9.595E-003
EX-CREWX-EI	1	1.000E-001	8.731E-003
AL	4	1.000E-002	4.650E-003
ST-DRSEL-LK	2	3.549E-001	3.408E-003
ST-LMECH-FF	2	8.393E-002	8.060E-004
EC-PPO2X-CF	1	8.762E-004	7.650E-004
EX-STRCT-FF	1	2.500E-001	2.183E-004
EC-PPO22-IF	1	7.858E-003	5.391E-005
EC-PPO21-IF	1	7.858E-003	5.391E-005
SI-MASPC-FF	2	1.000E-006	8.731E-006

Table D-10 (cont.). Lunar Base Importance Measures.

IMPORTANCE MEASURES REPORT (Current Cut Sets)			
Project	: PRA-COURSE	EndState :LOC	
	Analysis	: RANDOM	
	Case	: CURRENT	
(Sorted by Fussell-Vesely Importance)			
Event Name	Num. of Occ.	Risk Reduction Ratio	Risk Increase Ratio
SM	5	6.942E+001	5.080E-001
EX-CRTOX-IN	4	8.309E+000	1.875E+000
SI-MASPC-CF	2	8.308E+000	8.718E+000
EC-ARFL2-FF	2	1.782E+000	9.336E+000
EC-ARFL1-FF	2	1.782E+000	9.336E+000
UE-CRESC-FF	7	1.123E+000	1.179E+001
EX-CREQE-FF	2	1.118E+000	1.245E+000
PW-SHRTS-FF	1	1.117E+000	1.420E+000
EH	5	1.010E+000	1.050E+001
EX-CREWX-EI	1	1.009E+000	1.079E+000
AL	4	1.005E+000	1.460E+000
ST-DRSEL-LK	2	1.003E+000	1.006E+000
ST-LMECH-FF	2	1.001E+000	1.009E+000
EC-PPO2X-CF	1	1.001E+000	1.872E+000
EX-STRCT-FF	1	1.000E+000	1.001E+000
EC-PPO22-IF	1	1.000E+000	1.007E+000
EC-PPO21-IF	1	1.000E+000	1.007E+000
SI-MASPC-FF	2	1.000E+000	9.512E+000

D.2 PRA Example 2 Problem Description

The Lunar Base example demonstrated how a system operating in a steady state may be modeled using a set of multiple ETs with different IEs. Since the ETs are relatively small, system details are modeled with large FT constructs. This is a conventional PRA technique. The unique characteristic of such applications is that maintenance activities ensure that system components eventually achieve a steady-state availability. In these situations, humans are present and can perform necessary maintenance.

The second example addresses a science mission to another planet. Since there is no crew, maintenance is precluded. From a reliability perspective, without maintenance all components will eventually fail, so the reliability of individual components monotonically decreases with time. Consequently, the probability that any individual component is available at the beginning of a

mission phase is time dependent. Such conditional events are difficult to model with most software using the small ET approach. Therefore, a large ET model, using linked ETs, will be illustrated. Also, not every failure results in complete system failure. Some failures may result only in system degradation (e.g., loss of one side of a redundant system), or failure to completely satisfy mission objectives.

A complete application of the PRA process to the science mission in the second example is not provided. This is because many of the PRA steps are analogous to those demonstrated previously. Since the salient difference between the two examples is use of a large ET approach (instead of the more conventional small ET technique), the second example will only proceed through the ET model for the science mission.

D.2.1 PRA Objectives and Scope

The mission objectives for Example 2 involve placing an Orbiter Module in orbit around an object identified as Planet X. The orbiter will collect atmospheric information and deploy a Landing Module to the surface. The Landing Module will collect surface data and soil samples. Besides these science objectives, the mission must also ensure planetary protection and public safety. As always in a PRA, the first step is to define the assessment objectives and scope.

There are three PRA objectives:

1. determining the risk to the public during the launch;
2. determining the biggest risk contributors; along with
3. suggesting ways to improve the mission architecture and operations.

The scope of Example 2 includes:

- An expected casualty analysis of the launch vehicle;
- Assessing the spacecraft subsystems and science instrumentation; as well as
- Illustrating human reliability.

Chapter 8 addresses HRA aspects of the expected casualty analysis and possible instrumentation faults resulting from human errors.

D.2.2 Mission Success Criteria

The baseline science mission length is three Earth years, although minimum science objectives can be achieved with one year of operation.

D.2.3 End States

Six end states have been identified for the PRA, three for the launch phase alone. The launch phase end states involve loss of vehicle (LOV):

1. Before land clear (LOV-BLC);
2. After land clear (LOV-ALC); and
3. With no solid rocket motor (LOV).

They are needed in order to satisfy the objective of determining risk to the public.

The other three end states are:

1. Mission success (OK);

2. Loss of mission (LOM); and
3. Minimum mission (MIN).

Loss of vehicle signifies that the lander was not successfully deployed to the surface of Planet X. Hence, it can occur during the final phase of launch or any subsequent mission phase prior to initiation of the science mission.

Loss of mission designates those situations where the lander was successfully deployed to the surface of Planet X, but the science mission duration is less than one Earth year. If the science mission duration exceeds one year but terminates before the three-year objective is satisfied, the associated end state is MIN. Satisfying the three-year science objective corresponds to mission success (OK).

It is important to identify transition states in transferring between the linked ETs. Transition states are used to transfer information from one phase to another. They differ from end states, which are used to terminate an event sequence. Typically, transition states designate the status of critical systems (e.g., whether they are fully functional or have experienced a loss of redundancy) as the mission progresses from one phase to another.

D.2.4 System Familiarization

The mission profile is comprised of:

- Launch and separation;
- The cruise phase;
- The approach phase;
- Deceleration and orbit insertion;
- Landing module decent and landing; plus
- The science mission.

A two-stage, expendable vehicle serves as the launcher. The first stage has a cryogenic liquid oxygen/liquid hydrogen main stage plus two solid boosters. The second, or upper stage, is also cryogenic.

The spacecraft vehicle has the:

- Orbiter Module;
- Lander Module; and
- Deceleration Module.

The Orbiter Module contains an ion-engine for low thrust acceleration and correction of the approach trajectory. A chemical engine using hydrazine powers the Deceleration Module.

Table D-11 shows the launch phase timeline.

It is assumed that trajectory course maneuvers are unnecessary during the cruise phase. There are no planetary fly-bys, but the vehicle remains in communication with Earth to provide telemetry and other data. These are critical functions the spacecraft must perform. Also, it must complete the cruise phase with sufficient hardware available to perform the remainder of the mission.

Table D-11. Launch Phase Timeline.

Main stage boost	0 – 120 s Mission Elapsed Time (MET)
Solid rocket booster (SRB) burn	0 – 90 s MET
SRB separation	91.5 s MET
Upper stage (US) separation	121 s
US first burn	122 – 165 s MET
US first coast stage	165 – 550 s MET
US second burn	550 – 650 s MET
US second coast stage	650 – 9900 s MET
US third burn	9900 – 10,000 s MET
Spacecraft separation	10,000 s MET

Trajectory course maneuvers during the approach phase involve communication with Earth. In addition, the spacecraft must finish the approach phase with sufficient hardware available to perform the remainder of the mission.

The Deceleration Module is used to insert the spacecraft into an elliptical orbit around Planet X and subsequently to circularize the orbit. After achieving a circular orbit, the Deceleration Module is separated from the Lander Module/Orbiter Module stack.

One week after the circular orbit has been achieved, the Lander Module separates from the Orbiter Module and initiates a descent burn. A heat shield protects the Lander Module during its descent, which is controlled using braking thrusters and a parachute.

The science mission has a desired length of three Earth years. However, minimum objectives can be achieved with one year of operation. Both the Lander and Orbiter Modules are required during the science mission because the Orbiter not only relays Lander data back to Earth, but it also performs its own science experiments.

The spacecraft systems are:

- Command and control (CNC);
- Power generation, storage, and distribution (PWR);
- Attitude and orbit control (AOC);
- Ion Propulsion System (ION) (maintains orbit inclination);
- Chemical deceleration engine (CHM, performs deceleration and orbit circularization maneuvers);
- Communications (COM); and
- Pyro (PYR); Thermal control (THC); along with
- Science instrumentation (SCI).

Some of these systems (e.g., CNC, PWR, COM, and SCI) have subsystems dedicated solely to the Lander or Orbiter.

D.2.5 Initiating Events Development

IE categorization is the third step in the PRA process. Unlike the Lunar Base example, the mission to Planet X does not have a series of IEs in the literal sense. Instead, it begins with an entry point into the ET—initiation of launch. Since launch must be initiated for the mission to begin, the probability of this entry point event is unity. All other pivotal event probabilities are conditionally dependent on launch initiation.

D.2.6 Risk Scenario Development (Including ESD and ET Analysis)

Once the IE is selected, top level scenarios are then developed. The technique used with large ETs is to analyze the mission phases separately, then couple them using linked ETs with the appropriate transition states. Sections D.2.6.1 through D.2.6.6 address each phase of the mission to Planet X.

D.2.6.1 Launch Phase

Supporting the expected casualty analysis requires dividing the launch phase into three segments, as explained in Section D.2.3. Depending upon the PRA objectives, it may be possible to assess the probability of launch phase failure from historical data (if available). However, since this example includes an expected casualty analysis, the launch phase must be evaluated in detail. As indicated in Figure D-17, the mission begins with launch. If the launch is successful, the event sequence transfers to the cruise phase. A failure before land clear results in a loss of vehicle and end state, LOV-BLC. If the launcher fails after land clear but before 91.5 seconds transpire (recall that this is the time of solid rocket booster separation), the end state is LOV-ALC. If the launcher fails after the solid rocket booster separates but before spacecraft separation, LOV is the resultant end state. Figure D-18 is the corresponding launch phase ET. Alternatively, if the PRA objectives permit launch to be modeled as a single event, the launch phase ET can be simplified. Figure D-19 exhibits a simplified, single ET model for launch.

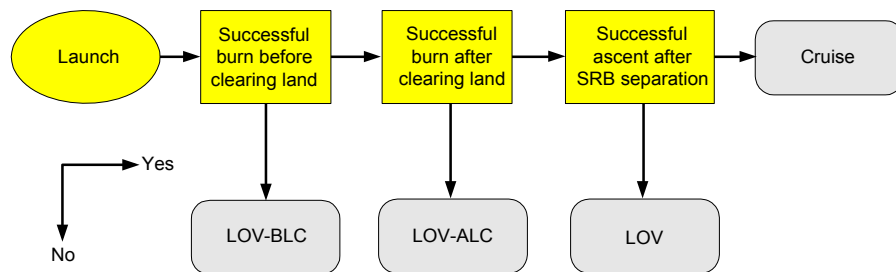


Figure D-17. Event Sequence Diagram for Launch Phase.

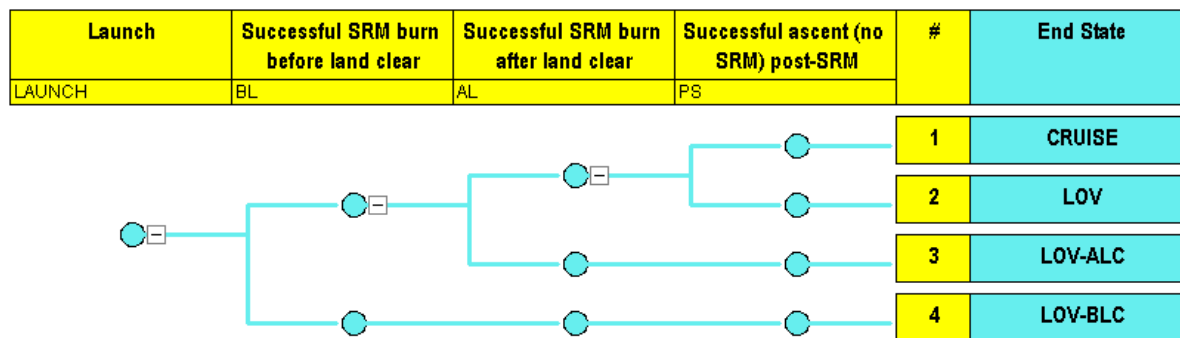


Figure D-18. Event Tree for Launch Phase

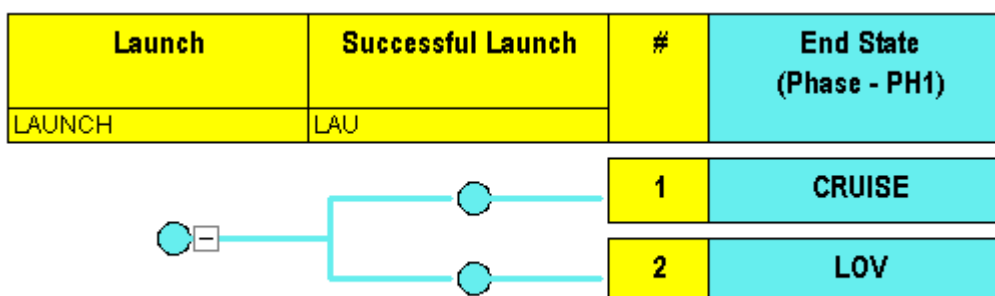


Figure D-19. Simplified Event Tree for Launch Phase.

D.2.6.2 Cruise Phase

Three topics are addressed in conjunction with the cruise phase of the mission to Planet X:

1. Basic ET models;
2. Quantifying redundant system probabilities; and
3. ET models for redundant systems.

They are explained in Sections D.2.6.2.1 through D.2.6.2.3.

D.2.6.2.1 Basic Event Tree Models

The success paths in both Figure D-18 and Figure D-19 result in transitions from the launch to the cruise phase. Since there are no trajectory course maneuvers, the spacecraft only needs to survive the cruise phase. This means the spacecraft cannot experience:

- System failures;
- MMOD hits (the spacecraft is assumed vulnerable to MMOD); or
- Excessive radiation (e.g., from solar flares).

Furthermore, the spacecraft must successfully respond to any nuances.

During the cruise phase only the:

- Thermal Control;
- Command and Control;
- Power; and
- Communications;

Subsystems are operating. Because failure rates for dormant systems tend to be very small, failures of dormant systems can be ignored as highly unlikely. Consequently, only MMOD hits or excessive radiation pose significant threats to the dormant systems. Figure D-20 is a preliminary ET for the cruise phase.

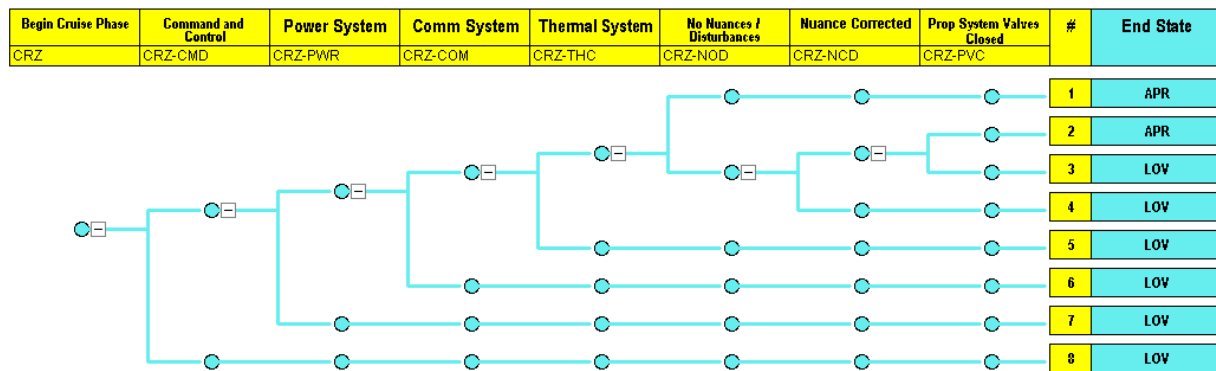


Figure D-20. Preliminary Event Tree for Cruise Phase.

Figure D-20 depicts cruise phase survival as requiring successful operation of the:

- Thermal Control;
- Command and Control;
- Power; and
- Communications;

Subsystems, in conjunction with either:

- The occurrence of no nuances; or
- Successful response to the nuances (which includes propellant valve closure).

If any of the operating subsystems fail or there is an unsuccessful response to a nuance, the end state is LOV. Otherwise, the event sequence enters a transition state and transfers to the approach phase (labeled “APR” in the ET).

Neither MMOD hits nor excessive radiation appear in D-20. This is because in Example 2, their contribution to mission failure is negligible. However, to determine the probability that they cause a mission failure, the process is similar to that used with the small ET approach. Let Λ_P be the frequency at which a phenomenological event (e.g., a MMOD hit or excessive radiation) impacts the spacecraft. This frequency is the IE frequency that would be used for the initiator in a small ET model. The difference between the small and large ET techniques is that for a large ET, the probability that the phenomenological event occurs during a particular mission phase, P , is

$$P = 1 - e^{-\lambda t} \quad (D-7)$$

where t is the duration of the mission phase being assessed. If P is negligible compared to other causes of mission failure, it can be ignored (as in Example 2). If it is not a negligible contributor to mission failure, its contribution is quantified using the same methods that are described in Chapter 10.

Figure D-20 can be simplified if modeling individual subsystems is unimportant. A simplified ET is illustrated in Figure D-21. If modeling individual subsystems is important (e.g., to track loss of redundancy events), techniques for such modeling are required.

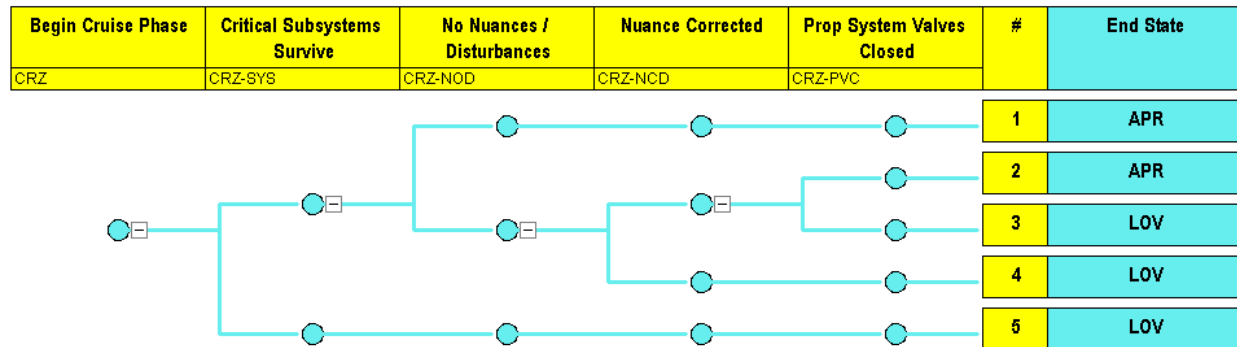


Figure D-21. Simplified Event Tree for Cruise Phase.

D.2.6.2.2 Quantifying Redundant System Probabilities

Loss of system redundancy is an important issue in the mission to Planet X. Table D-12 lists the status of the redundant batteries as a function of mission phase. It is postulated that both batteries are available prior to launch, and that the common cause beta-factor applicable to the batteries has a value of 0.1. Notice that as the mission progresses, the probability that both batteries remain available continuously decreases, while the probability that only one side or neither side is available increases. Nevertheless, because the failure rate for the batteries is small relative to the overall mission duration, the probability that both batteries fail is only 2.91×10^{-4} at the end of the planned, three-year science mission. Figure D-22 displays the probability that:

- Both;
- Only one; or
- No;

batteries are operational as a function of the product, λt . Here:

- λ is the total failure rate for an individual battery; and
- t represents the operating time.

For any given failure rate, the probability that both batteries are available monotonically diminishes with time, since repair is precluded. Independent failures transfer the system to the state where only one battery is available. For this reason, the probability that only one battery

train is available initially increases with time. However, since repair is precluded, the probability that even a single battery train remains available eventually decreases to essentially zero. Consequently, the probability that neither train is available monotonically increases with time. Both Table D-12 and Figure D-22 were derived using the methodology from Chapter 7.

Table D-12. Probability of Battery Status (per Mission Phase).

End of	Both Batteries Available	Only Side A Available	Only Side B Available	Both Batteries Fail
Launch	~1	8.60×10^{-8}	8.60×10^{-8}	9.56×10^{-9}
Cruise	~0.999	3.66×10^{-4}	3.66×10^{-4}	4.08×10^{-5}
Approach	~0.999	3.66×10^{-4}	3.66×10^{-4}	4.08×10^{-5}
Deceleration and Orbit Insertion	~0.999	3.68×10^{-4}	3.68×10^{-4}	4.10×10^{-5}
Descent and Landing	~0.999	3.68×10^{-4}	3.68×10^{-4}	4.11×10^{-5}
Minimum Mission (1 year)	~0.998	1.10×10^{-3}	1.10×10^{-3}	1.23×10^{-4}
Desired Mission (3 year)	~0.995	2.55×10^{-3}	2.55×10^{-3}	2.91×10^{-4}

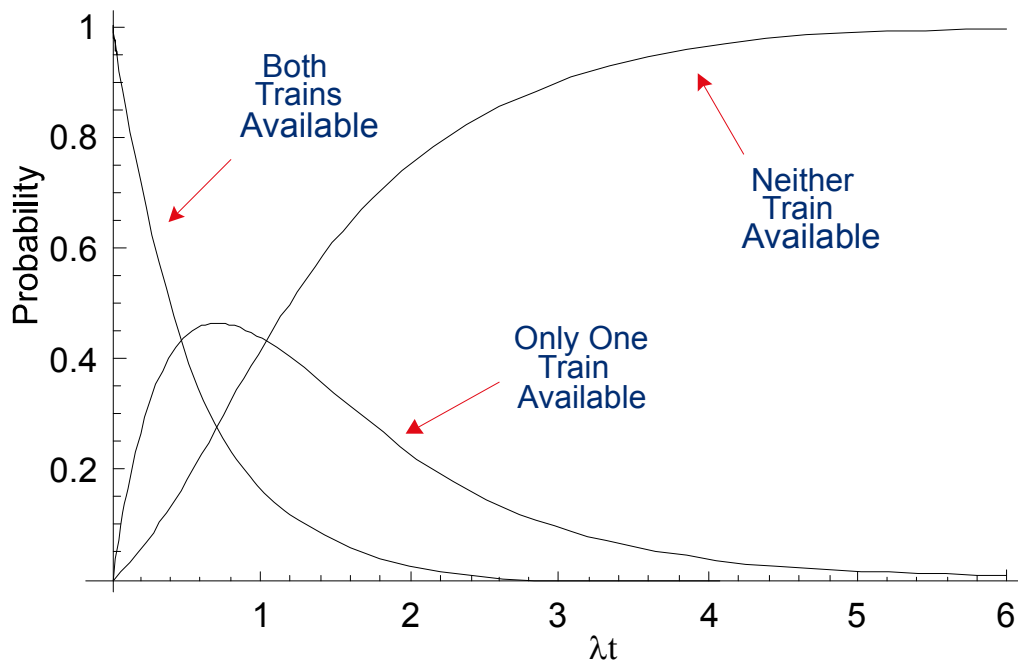


Figure D-22. Probability of Battery Status (as a Function of λt).

D.2.6.2.3 Event Tree Models for Redundant Systems

Lack of maintenance prevents the probability that a system is in a particular state (e.g., completely available or degraded) from achieving a time independent value. Hence, it becomes necessary to consider how to model this time dependence. Although in theory this problem is independent of modeling technique, in practice most PRA software is more amenable to modeling this time dependence in ETs rather than FTs. This is because ETs conceptually display event progressions over time, while FTs tend to represent a “snap-shot” in time.

A system with two redundant trains must be in one of three states:

1. Both trains are available;
2. Only one train is available; or
3. Both trains are unavailable.

Figure D-23 and Figure D-24 are alternative techniques for modeling system redundancy in ETs.

Beginning with Figure D-23, if there is no total system failure then the system must have:

- Both trains available; or
- Just one train available.

Given that there is no total system failure and no loss of redundancy, the only state is that both trains are available. However, if there is no total system failure but a loss of redundancy occurs, then only one train can be available (in this case the end state is labeled “LOR”). If there is a total system failure, no trains are available. A total loss of the system would lead to an LOV end state.

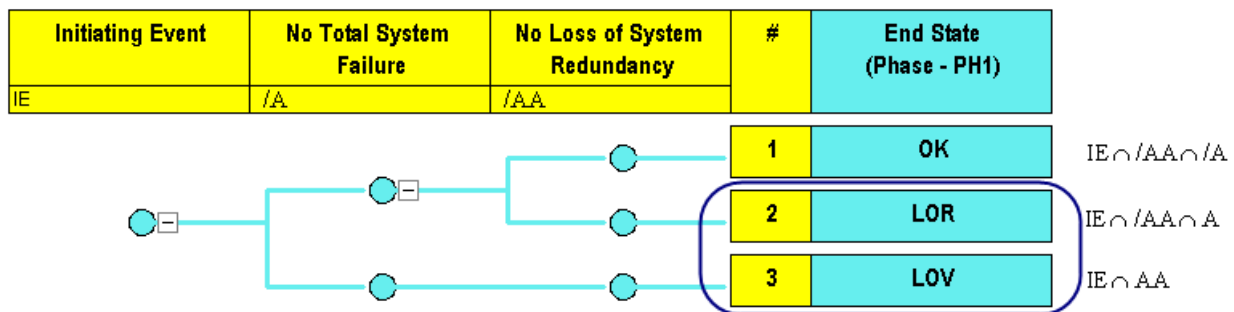


Figure D-23. Event Tree Model of System Redundancy.

Alternatively (Figure D-24), these two pivotal events can be reversed. The event, “no loss of system redundancy,” signifies that we are not in a state where just one train is available. Consequently, the system must be in a state where either:

- Both trains are available; or
- Neither train is available.

Combining this with the requirement that there is no total system failure, the system has both trains available. If we are not in a state where just one train is available and total system failure

occurs, both trains are unavailable. Finally, if we are in the state representing a loss of redundancy, then just one train is available.

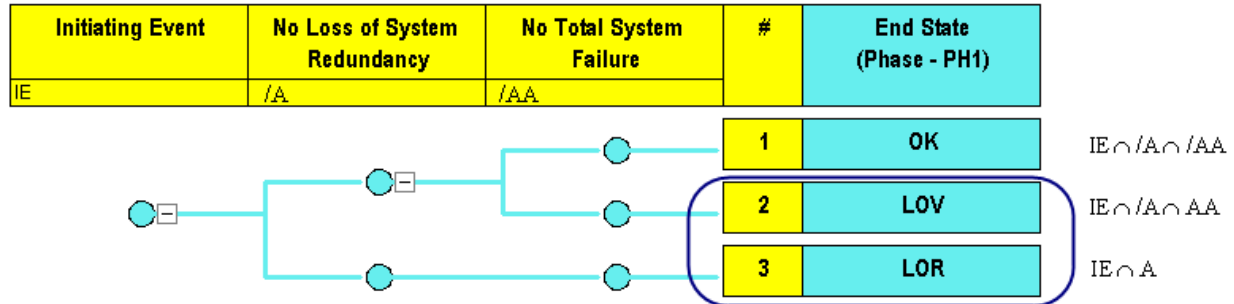


Figure D-24. Alternative Event Tree Model of System Redundancy.

Mathematically, both models are equivalent (they reduce to the same set theoretic end states). However, there are two practical considerations in modeling loss of redundancy. The approach selected must be:

- Easy for the analyst to apply (in order to minimize the introduction of human errors into the assessment); and
- Compatible with the PRA software being used.

Experience indicates that the first technique (in Figure D-23) is conceptually easier for PRA analysts to apply and has no known software incompatibilities.

D.2.6.3 Approach Phase

There are several possible states for entering the approach phase:

1. OK, meaning that the spacecraft is fully functional; or
2. with various losses of redundancy.

However, the basic modeling techniques are the same as those described previously. The key consideration is that the remaining system redundancy at the beginning of approach can be known explicitly from the transition states linking the cruise and approach phases.

The process of modeling the approach phase begins by reviewing the mission events. They indicate that it is necessary to:

- Power up the spacecraft;
- Communicate with Earth; and
- Perform the entry turn.

Naturally, the spacecraft systems must also survive since they are needed in subsequent mission phases.

D.2.6.4 Deceleration and Orbit Insertion

Deceleration and orbit insertion begin by firing the chemical propulsion engine. If this fails, the vehicle is lost. If the initial firing is successful, it is next necessary to circularize the orbit.

After this final firing of the chemical propulsion engine, it must be separated from the spacecraft. Failure to separate results in loss of vehicle.

If a circular orbit is achieved and the chemical stage separates, the Lander descent phase can begin. If a circular orbit is not achieved but the chemical stage separates, it may be possible to circularize the orbit using the ion-engine. However, this may result in an abbreviated science mission.

D.2.6.5 Landing Module Decent and Landing

The basic modeling approach to be applied for entry, descent, and landing is analogous to those given previously. Basically, it begins by reviewing the event list, then developing an appropriate ESD and ET. When this is finished, the science mission is assessed.

D.2.6.6 Science Mission

Figure D-25 is an ET depicting the Lander portion of the science mission. Since the Orbiter must also collect data and communicate with Earth, it could be appended to Figure D-24 to evaluate which of those successful Lander end states ultimately result in science mission success.

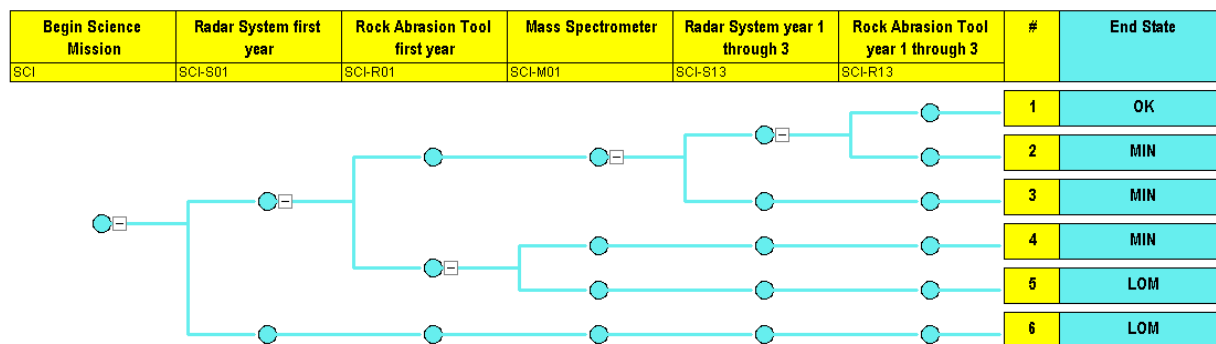


Figure D-25. Event Tree for Lander Science Mission.

Figure D-25 replicates the Lander success criteria. Basically, the Lander consists of three instrument packages:

- Radar System;
- Rock abrasion experiment; and
- Mass spectrometer.

The Radar System is vital for the science mission, so if it fails essential data are lost.

The rock abrasion experiment collects samples and analyzes them inside of the Lander. However, if this experiment fails, remote sensing can be accomplished using the mass spectrometer. Two limitations of the mass spectrometer are:

1. It has less analytical capability than the rock abrasion experiment; and
2. due to calibration drift, its operating life is one Earth year.

Relative to minimum mission requirements, Figure D-25 demonstrates that if the Radar System fails to operate for at least a year, the mission fails (i.e., LOM is the end state). If the

Radar System and rock abrasion experiment both operate for at least one Earth year, minimum mission requirements are satisfied. However, if the Radar System operates for at least one Earth year but the rock abrasion experiment fails during that year, minimum mission requirements can still be satisfied by the mass spectrometer.

Relative to total mission requirements, they can only be satisfied after the minimum mission is completed. Thus, if the Radar System fails during the second or third year, the end state is MIN. If the Radar System and rock abrasion experiment both operate for two additional Earth years, the total science mission is a success. However, if the rock abrasion experiment fails during the second or third year, only minimum mission requirements are fulfilled because the mass spectrometer lifetime is too short.

D.2.7 Remaining Tasks

Remaining PRA tasks are those addressed in Sections D.1.13 through D.1.15. They are not elaborated in Example 2 because the methodology and techniques employed are analogous to those already presented.

D.3 Reference

- D-1 Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE), a computer code developed at the Idaho National Laboratory, <https://saphire.inl.gov>

Appendix E - PRA Simulation Example

To demonstrate the use of simulation for risk assessment, we consider a couple of examples in this Appendix.

E.1 Example 1: Leaks at a Lunar Base

First, to illustrate the technique of using simulation to model an ESD, consider the ESD developed for Atmosphere Leakage in section D.1.

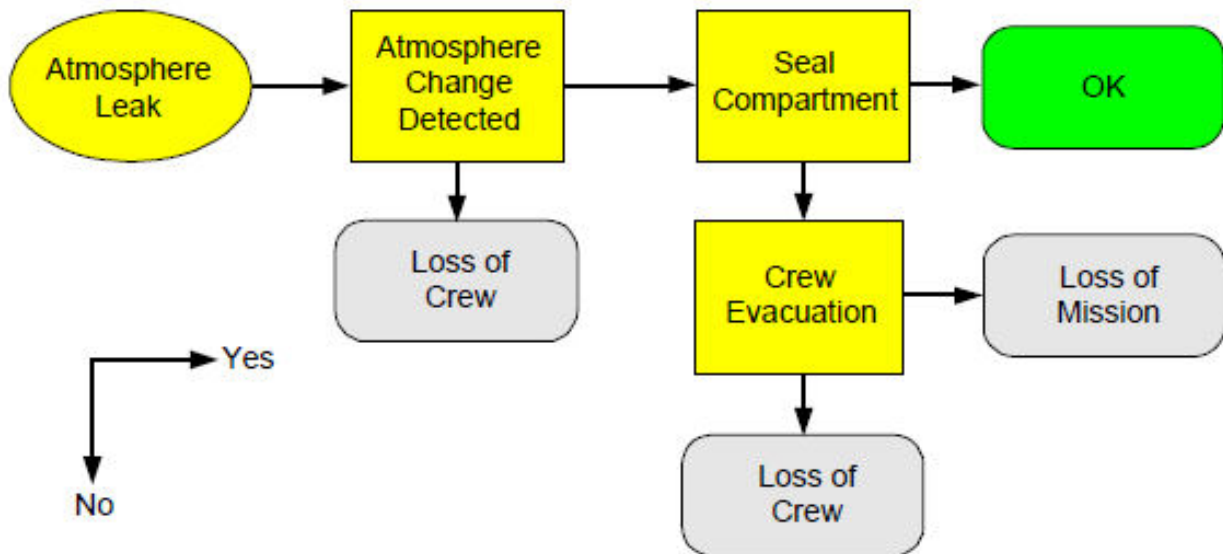


Figure E-1. Atmosphere Leak ESD.

The system success criteria are the same for simulation as they are for the fault and event-tree based approach to PRA:

1. The minimum number of successful trains in redundant systems necessary for the system to satisfy its functional requirements.
2. The time the system must operate.

For the Lunar Base, the atmosphere of a habitat is not a redundant system. Although multiple chambers most likely would be used in a lunar base habitat, for illustration, one chamber is modeled without redundancy in the atmospheric system. The meteoroid flux and micro-meteoroid flux experienced on the lunar surface is evident in the visible craters present.

One study^a points out that a 1 m² surface would experience a flux roughly a factor of 4 less than that of low earth orbit:

Table E-1. Lunar Surface Micrometeoroid Flux.

Estimated flux: Crater Diameter (µm)	# craters / m ² / yr
0.1	3 X 10 ⁵
> 1	1.2 X 10 ⁴
> 10	3 X 10 ³
> 100	6 X 10 ⁻¹
> 1000	1 X 10 ⁻³

The energy associated with a meteoroid is reliant on its velocity at impact. Not all meteoroids would penetrate the lunar habitat wall, which would be specifically designed to survive impacts.

A simulation model representing the ESD was constructed in order to model the scenario-based events. In this case, the hypothetical events in the scenario are:

1. the initiating event of a leak
2. the detection of the leak
3. the repair of the leak
4. evacuation from the lunar base (if necessary)

The ESD is decomposed into states and events and frequencies. The first diagram shows the "Simulation Objects," which in this case handle events simultaneously. The Simulation Objects are Compartment, Leak Detection, Maintenance, and Escape Craft.

^a Introduction to the Moon, Paul D. Spudis, Lunar and Planetary Institute, presented at NASA Johnson Space Center, 4 June 2008

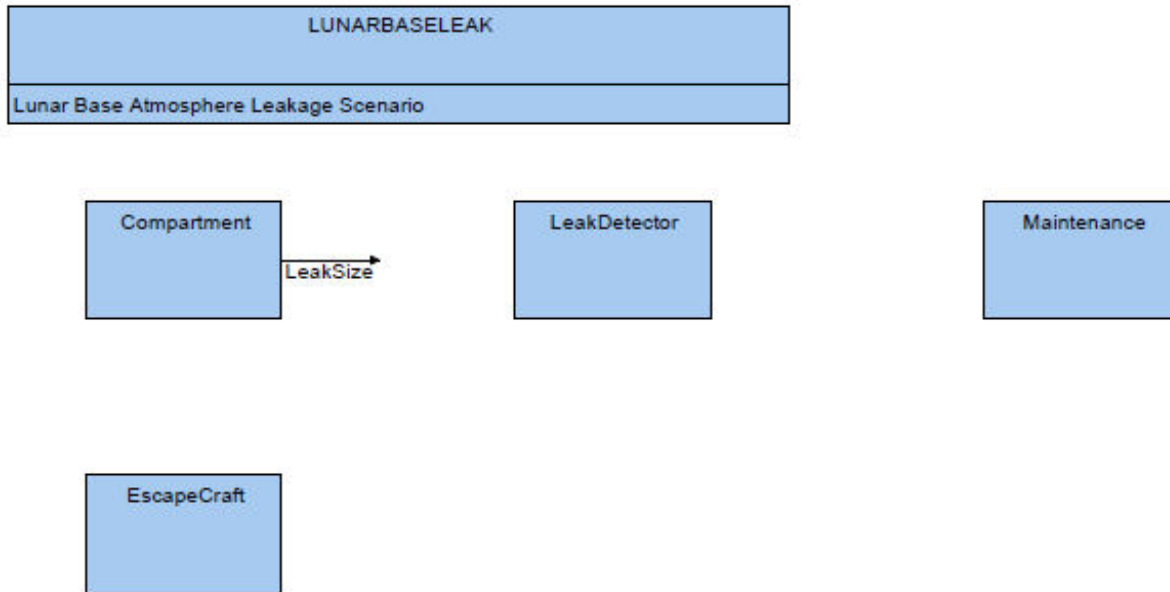


Figure E-2. Lunar Base Atmospheric Leak Simulation Objects.

The Compartment object is where the **initiating event** of a leak will occur. When simulating an atmospheric leak, the magnitude of the initiating event needs to be considered before its detection. In this case, the size of a hole caused by a meteoroid is the initiating event and was split into the three magnitudes of small leak, medium leak, and large leak.

- “SmallLeak” is defined as one which can be repaired while the habitat is still occupied.
- “MediumLeak” is one that would cause the habitat to be evacuated.
- “LargeLeak” is modeled as causing loss of crew before successful detection or repair could be performed.

Probabilities of an occurrence of the initiating event over the operational period of the system are entered into the simulation for each of its leak type. Considering published reports of micrometeoroid flux experienced at the ISS, the probabilities were calculated by using exponential distributions with the mu for a “SmallLeak” at a mean time to occur of 240 days, “MediumLeak” at 55,000 days, and “LargeLeak” at 270,000 days, respectively. The block diagram of the habitat compartment is shown in Figure E-3 and shows the consequences associated with each leak size. When the simulation is started, all of the block diagrams are at their initial states. With “MissionStart” arc, the Compartment node changes to the “Occupied” state and time to failure is calculated for the first Small, Medium, and Large Leaks along the mission timeline. The simulation then handles the first event (which is typically “SmallLeak” since it has the shortest mean time to occur).

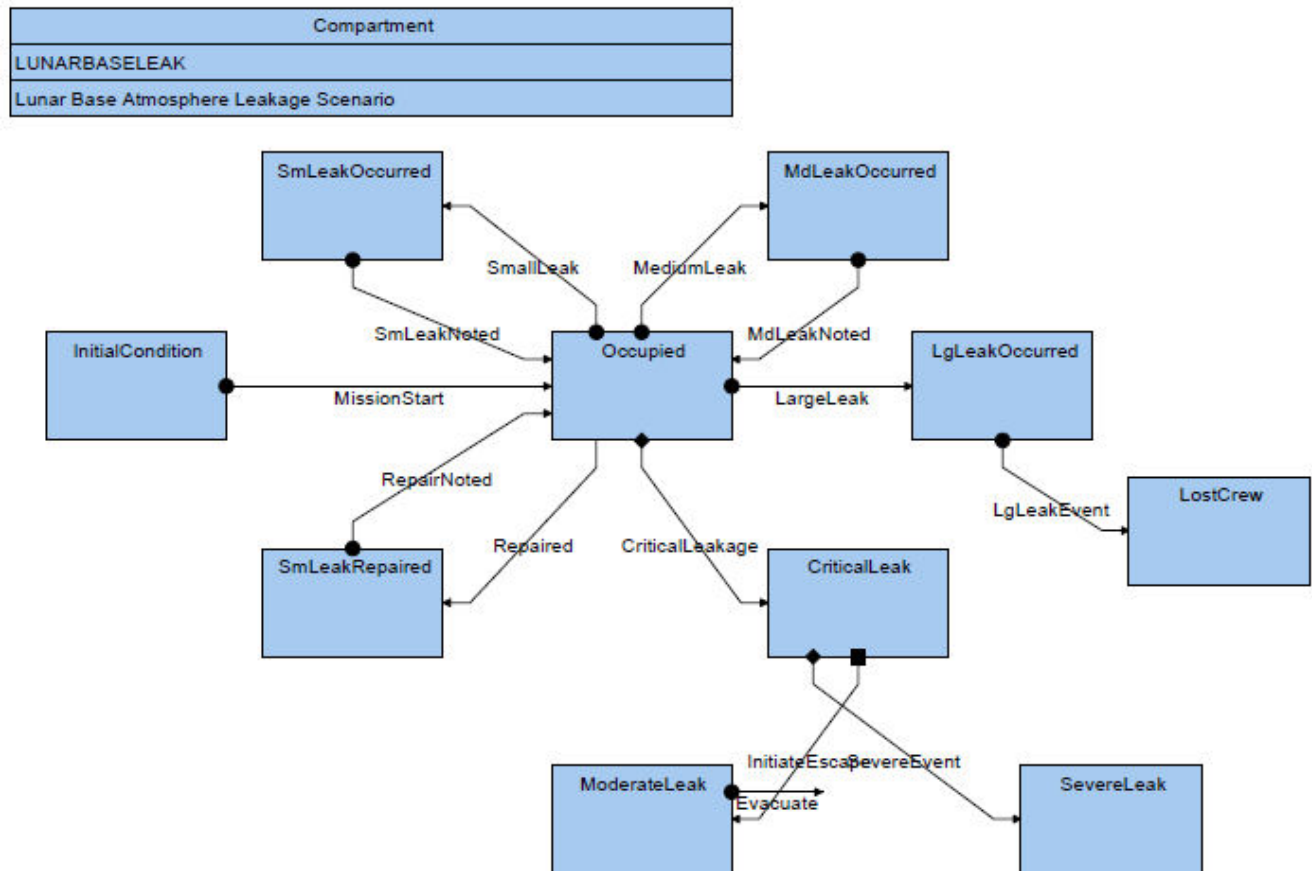


Figure E-3. Compartment Block Diagram.

The event “SmallLeak” (when it occurs) activates the node “SmLeakOccured” on the Compartment diagram and “DetectLeak” on the Leak Detector diagram (Figure E-4). The Event “SensorFailure” is checked using a probability of 0.001, which leads to the State of “Undetected” and loss of crew (LOC). In the very likely Event of “NeedRepair,” the State of “RequestRepair” is entered.

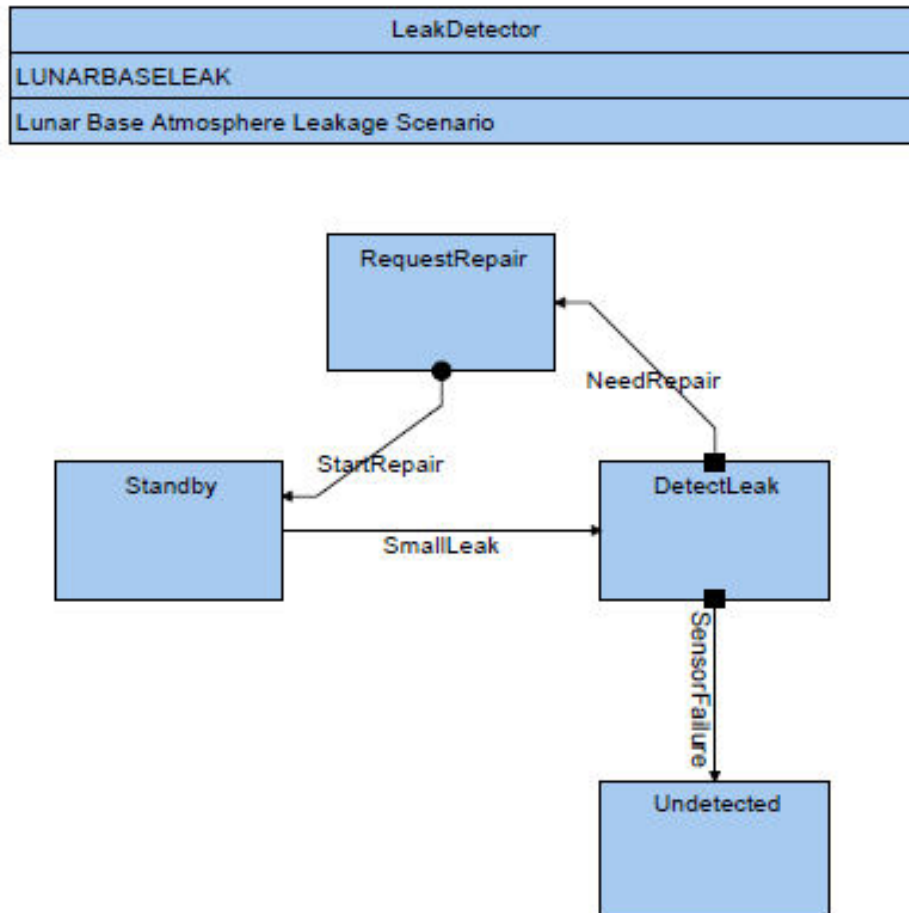


Figure E-4. Leak Detection Block Diagram.

The event “StartRepair” returns the Leak Detector diagram back to the “Standby” state and moves the Maintenance diagram state of “Oncall” to “Repairing.” Event “HumanError” is checked which includes repair failure and if it does not occur (meaning no failure to repair), then the event “Repaired” is generated, which has a mean time of one day before returning to the “SmallLeak” repaired state.

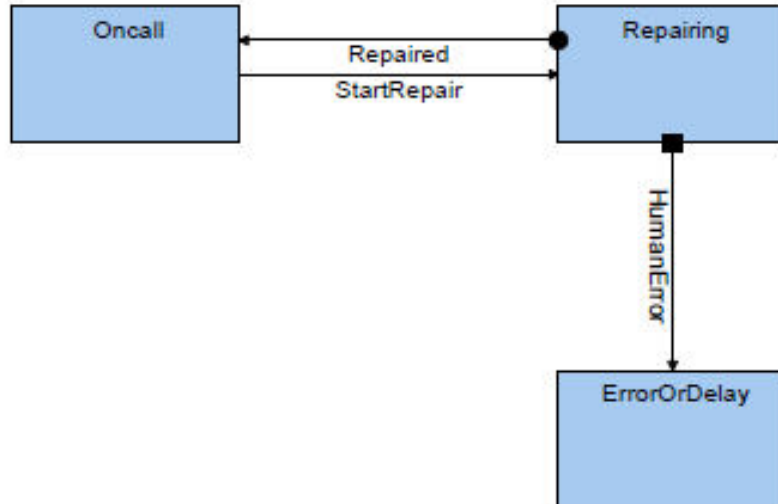


Figure E-5. Maintenance Block Diagram.

The way the simulation keeps track of the size of a leak is to give a small leak a value of 1, a medium leak would be 2, and a large leak 3. By using this state variable, if another “SmallLeak” occurs before the first one is repaired, the leak value (“LeakSize” variable within the simulation) increases to 2 and becomes the equivalent of a “MediumLeak” event. If the variable “LeakSize” is 2 or larger, the event CriticalLeakage is generated and the model moves into the state of “CriticalLeak” on the Compartment diagram. Then, the value of “LeakSize” is checked again. If the value is 3 or greater, LOC is output through the state “SevereLeak.” If the value is 2, an evacuation is requested through the state “ModerateLeak,” which generates an event “Evacuate” and moves the Escape Craft diagram from the state “OnStandBy” to the state “AttemptEscape.” The Escape Craft block diagram checks for successful launch, cruise, and landing of the escape vehicle on Earth through similar logic.

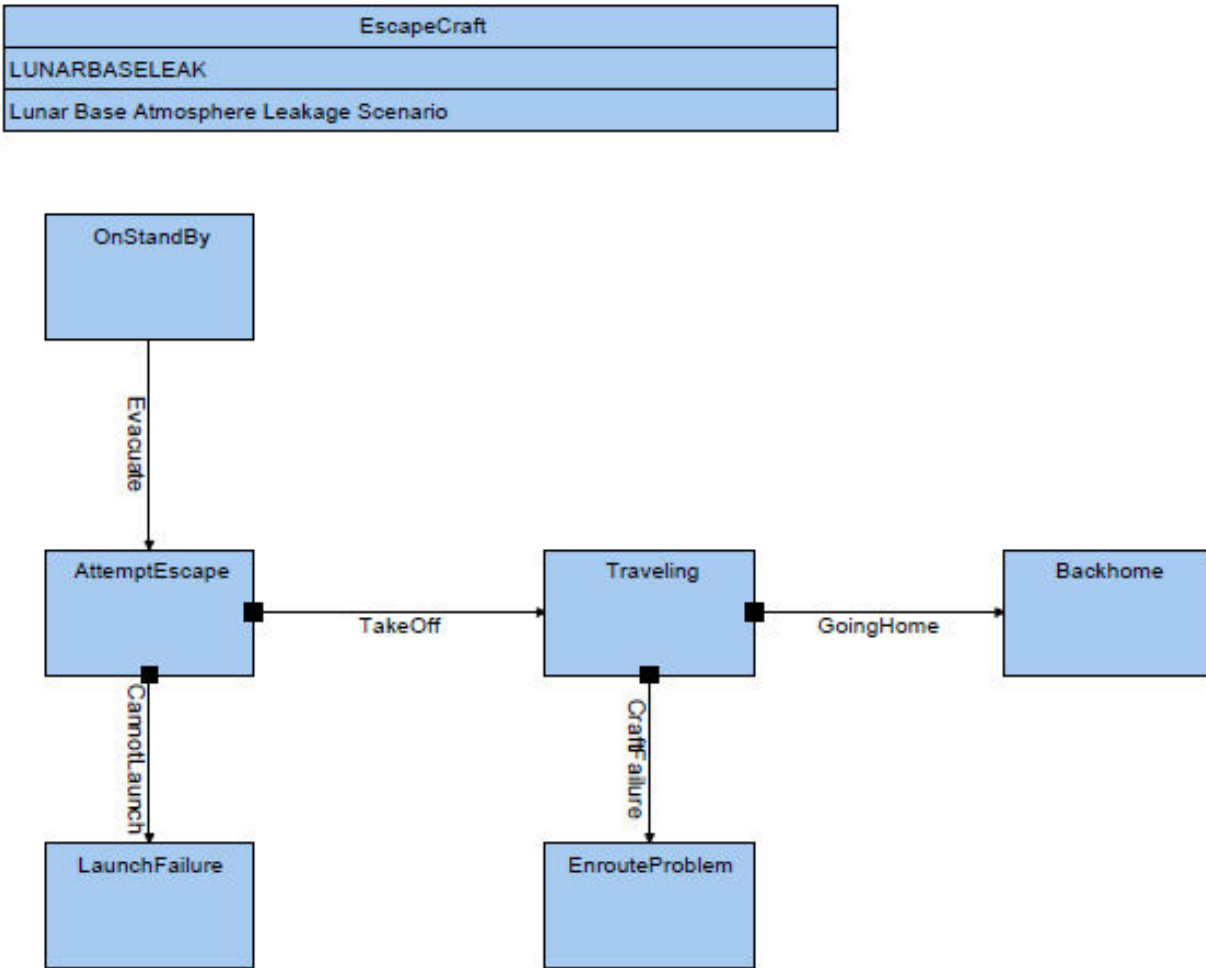


Figure E-6. Escape Craft Block Diagram.

Note that there are exits from the simulation in nearly every block diagram and that the “Compartment” block diagram, despite sending and returning status to it, is not a “master” diagram, but just a part of the overall simulation.

The simulation runs along the mission timeline, in this case 7,300 days (i.e., 20 years), using the transition rates described. The initiating event of a leak checks the other events for follow-on actions at that “date.” Once a leak is initiated, the probabilities of the detection, successful repair, or successful evacuation, if necessary, are checked to see if they occur before the leak is fixed. The repair time was set to 4 days.

Once the leak is fixed, the next leak on the timeline is handled in the same manner, and then the next, until the mission time has been reached. This constitutes an iteration of the simulation. By running multiple iterations of the simulation, the simulation results converge to a state value (either occupying the base, evacuation leading to a loss of mission (LOM), or LOC for each day along the 7,300 days run for the simulation. The resultant probabilistic outcomes of LOM and LOC along with the summation of an unsuccessful mission (either LOM or LOC) are presented in Figure E-7.

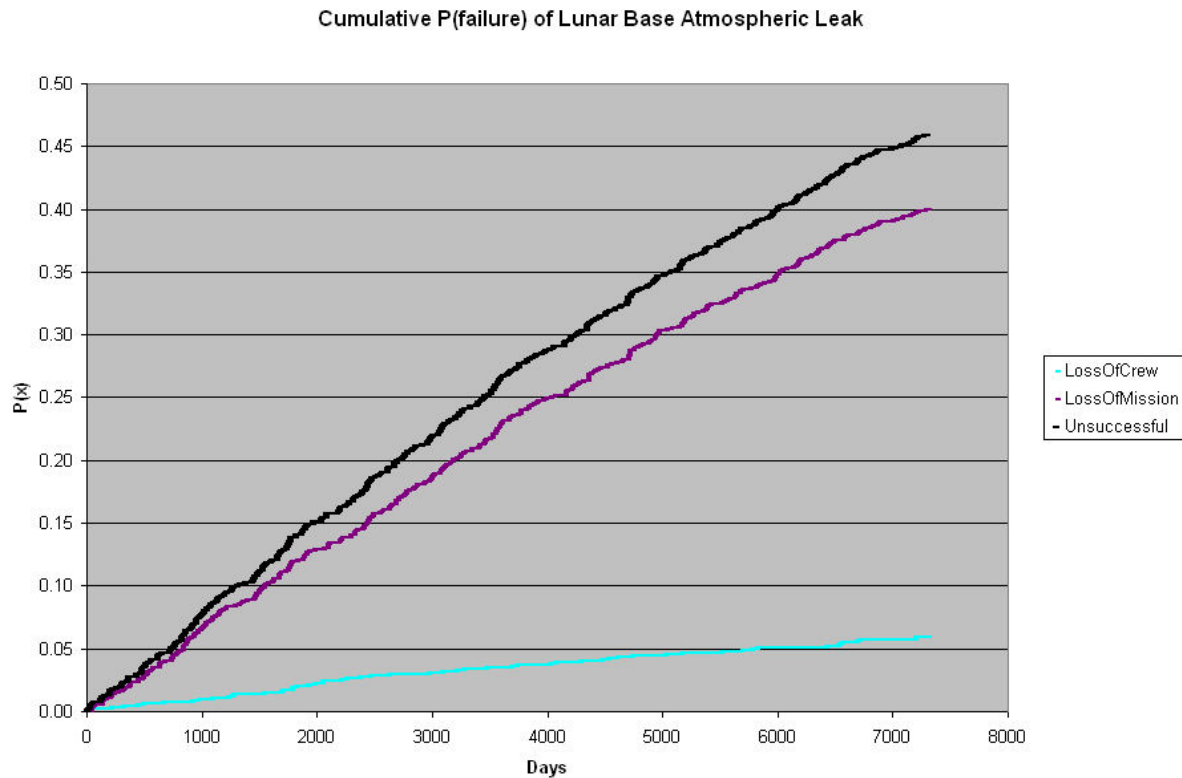


Figure E-7. Lunar Base Atmospheric Leak Simulation Results.

The output of the simulation is a cumulative total versus time based upon 1,000 iterations. One of the limitations of simulation is the difficulty in modeling uncertainty. To do so would require sampling from epistemic distributions for each parameter, and rerunning the simulation many more times.

E.2 Example 2: Expanded Atmospheric Leak Model

In the first example, it would be difficult, but not impossible, to perform the analysis using traditional fault and event tree logic models. To demonstrate further the capabilities of simulation, we modified the Example 1 model to include periods of time where the crew is removed from the habitat on outside missions such that there is not an immediate repair capability if a small leak occurs. Further, if the crew is outside the habitat when a medium or large leak occurs, they are already suited up and just go to the escape craft. These complications, if treated using traditional methods, would require complicated conditional calculations and convolution approaches. We will, instead, demonstrate the approach using simulation.

A simulation model was created representing this scenario. Hypothetical events in the scenario are:

1. The scheduling of missions outside the habitat
2. The availability of a repairman based on the external mission
3. The initiating event of a leak caused by MMD

4. The detection of the leak
5. The repair of the leak
6. Evacuation from the lunar base (if necessary)

States and events were modeled along with frequencies that those states and events have taken place. The simulation objects for the new model are now MMD Events, Containment, Leak Detector, Maintenance, Mission, and Escape Craft.

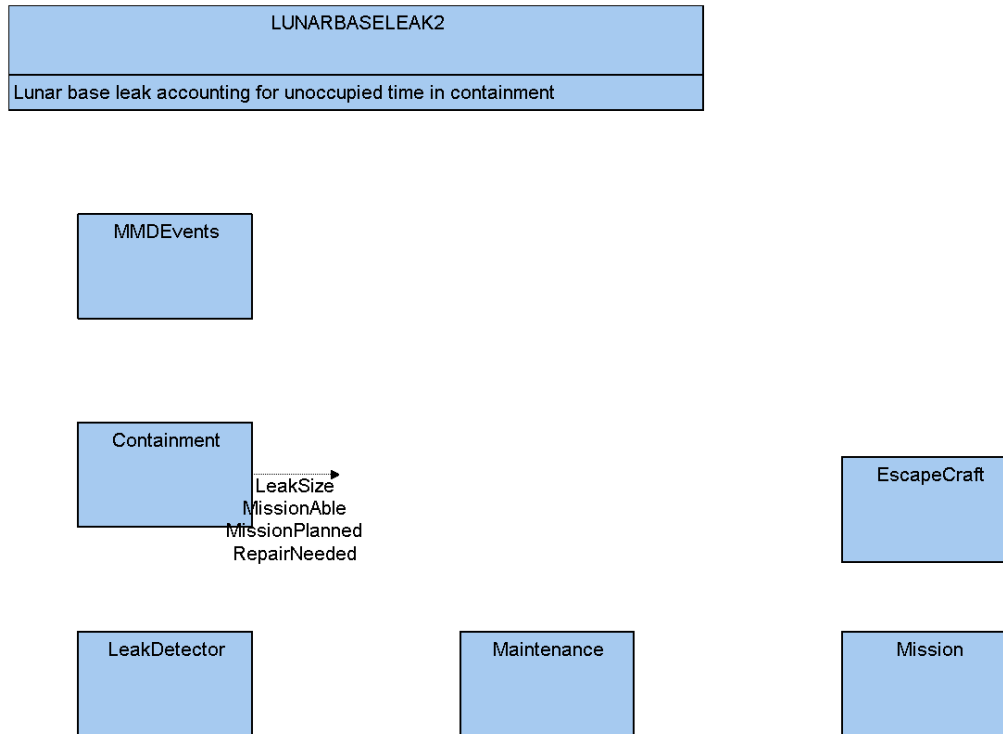


Figure E-8. Lunar Base Atmospheric Leak Objects with External Missions

There are multiple outcomes of the leak initiators now because we have the possibility of the crew being out of the habitat and suited up. To handle these multiple (conditional) outcomes a leak event or MMD Event generator block diagram was created external to the Containment diagram with the outcomes SmallLeak, MediumLeak, and LargeLeak generated for use in the Containment and LeakDetector blocks. StartMMD is initiated when the simulation starts and generates leaks in GenerateMMD based on the frequencies assigned for small, medium and large leaks. Small leaks return to the scenario to see if the leak can be fixed in time. Medium and large leaks are associated with the termination of the simulation.

MMDEvents
LUNARBASELEAK2
Lunar base leak accounting for unoccupied time in containment

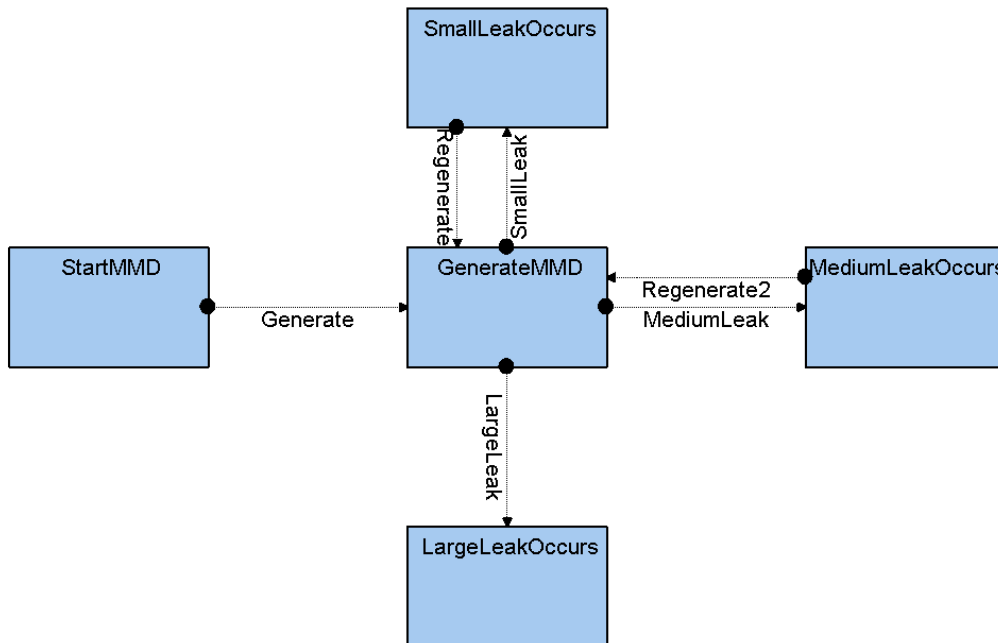


Figure E-9. MMD Event Generator

The Containment diagram is modified from the original model by adding the blocks (states) necessary to account for the external missions. The *MissionPlanning* state was added which generates a mission based on availability of the crew. The crew is available if they are not tending to a small leak.

Once a mission is generated it places the lunar base in an *Unoccupied* state where it stays throughout the mission time until *Returns* places the model back in the *Occupied* state. If a leak happens while in the *Unoccupied* state, the *AbortNoted* is generated through the Mission block diagram and the state is returned to *Occupied* via the *AbortNoted* path rather than the *Returns* one. In this way the en route time is added on to the repair time which increases the possibility of a second leak event causing a loss of mission. Medium and Large leaks while on a mission are handled in the Mission block diagram where an evacuation attempt is made without occupying the habitat again (the base is abandoned).

Containment
LUNARBASELEAK2
Lunar base leak accounting for unoccupied time in containment

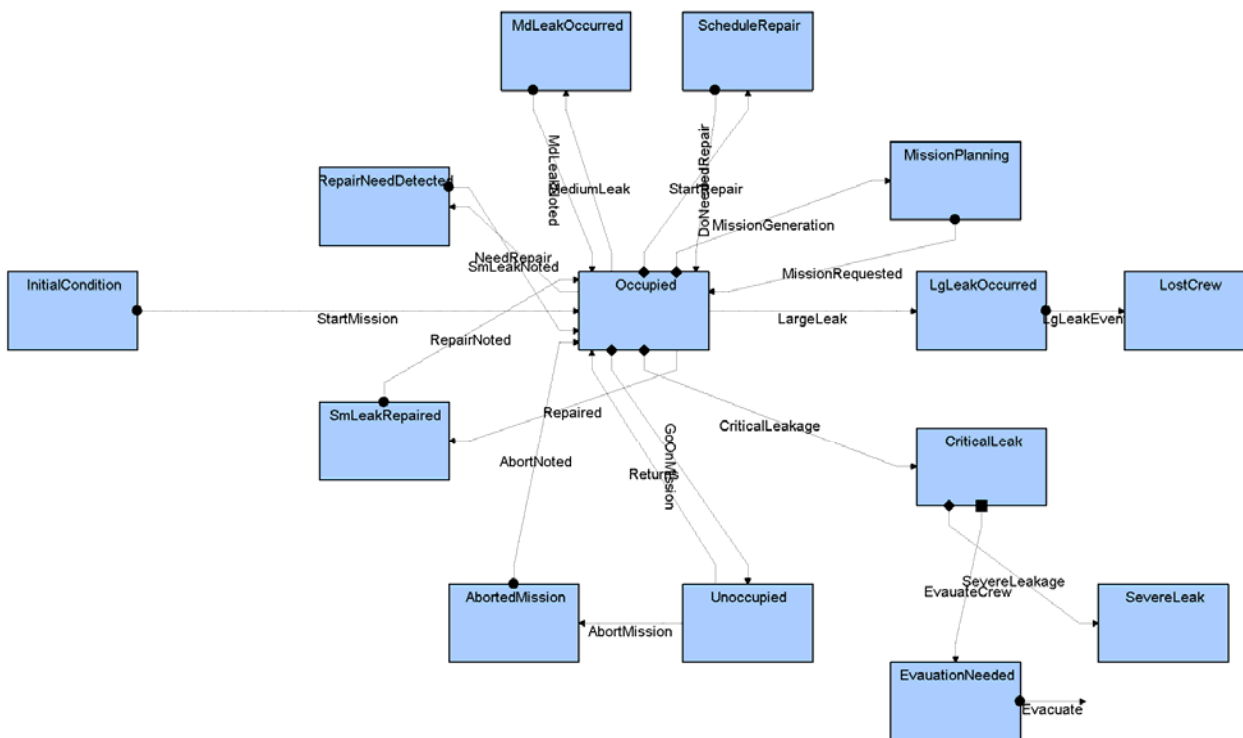


Figure E-10. Containment Objects Block Diagram with External Work

The Leak Detector diagram and function are the same as the original model.

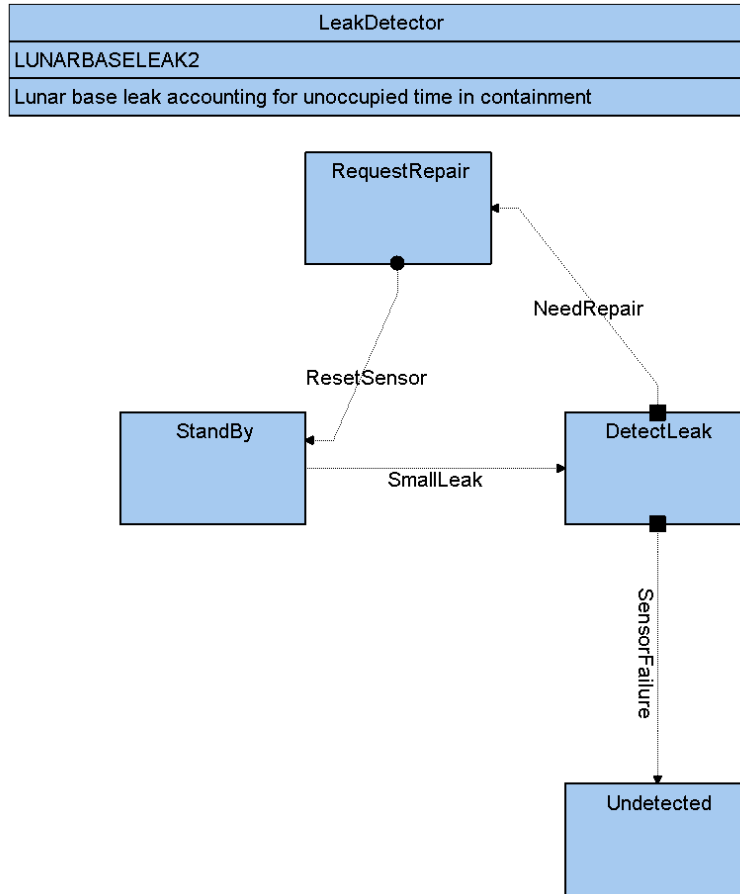


Figure E-11. Leak Detector Block Diagram for External Work Model

A state of *RequestRepair* was added to the Maintenance diagram to allow the repair to start based on the availability of the crew.

Maintenance
LUNARBASELEAK2
Lunar base leak accounting for unoccupied time in containment

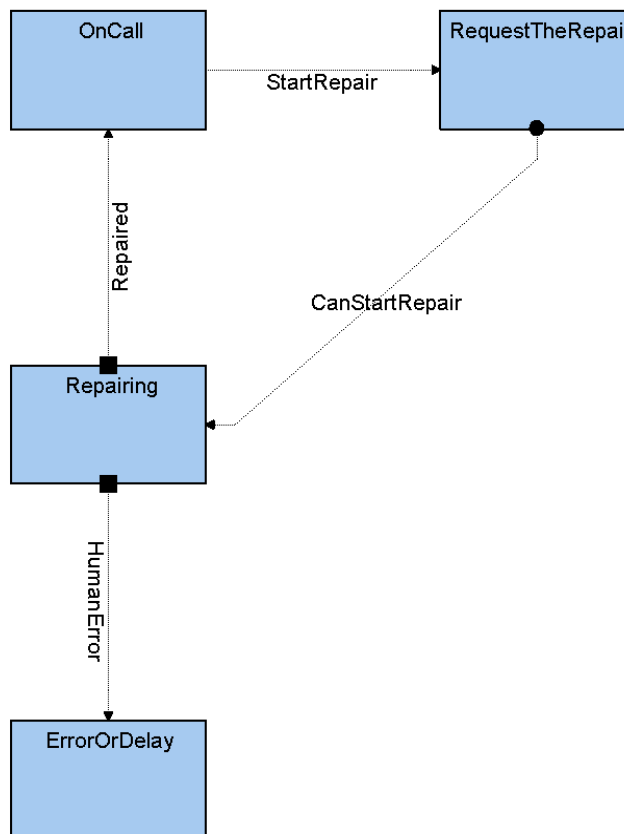


Figure E-12. Maintenance Block Diagram used with External Work

The Mission diagram is new to this model and sits in a *Pending* state until *GoOnMission* is generated in the Containment diagram. Once *GoOnMission* is generated the state changes to *Working* for the duration of the mission. Barring a leak, the state returns to *Pending* and notification of completion is generated with the *Returns* action. If there is a small leak while the mission is in progress, *SmallLeak* is generated from the MMDEvents diagram and changes the state to *SmLeakEmergency* which notifies to the external crew to return to the habitat to repair the leak. This is modeled by the *AbortMission* action, with the time delay for returning to the habitat in the *SmLeakEmergency* state. *AbortMission* acts within the Containment diagram as noted above to set the state in that diagram to *Occupied* and to start a repair action.

Mission
LUNARBASELEAK2
Lunar base leak accounting for unoccupied time in containment

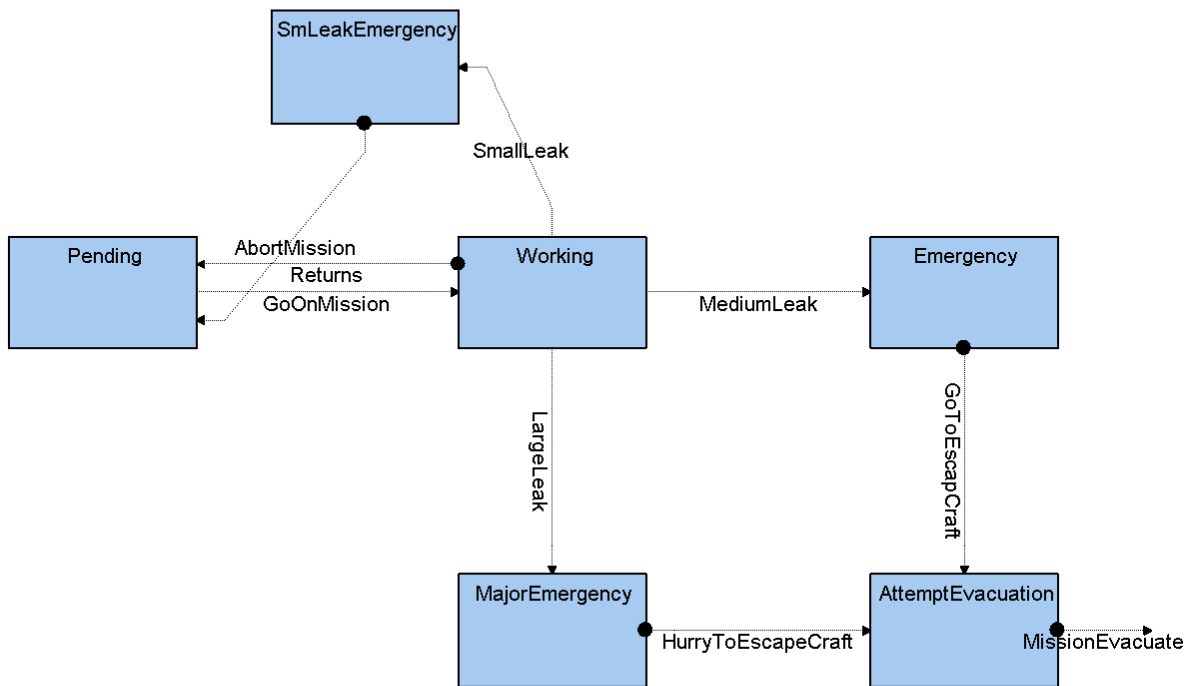


Figure E-13. Mission Block Diagram for External Work

The Escape Craft diagram has not changed from the original model.

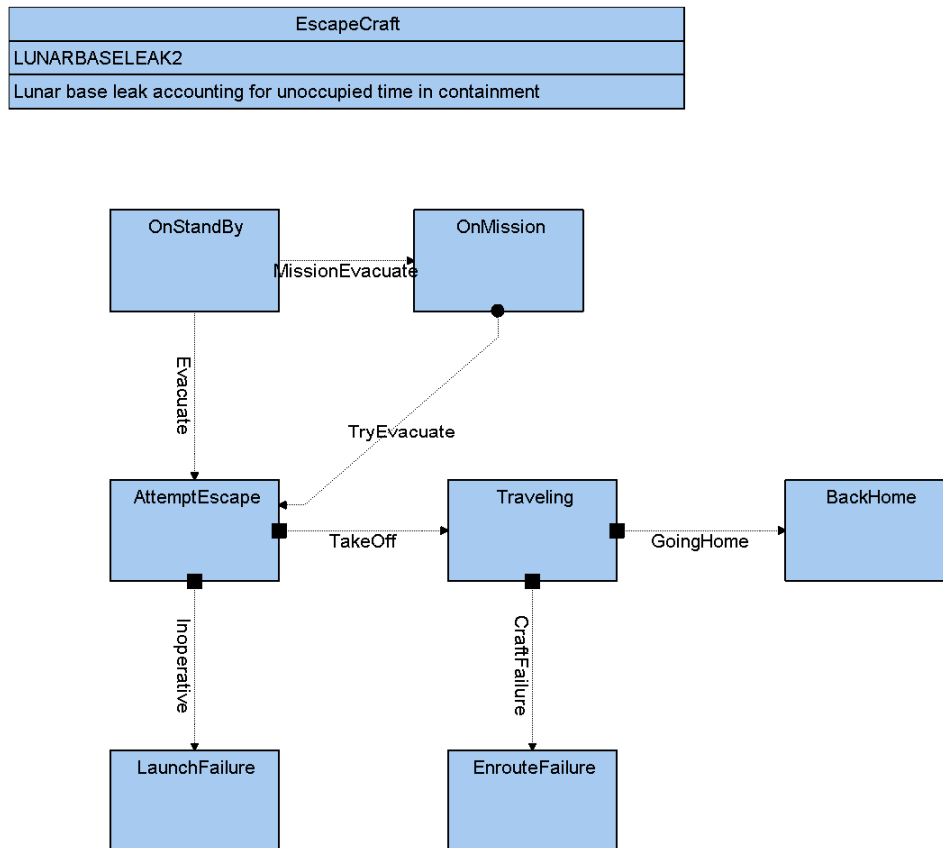


Figure E-14. Escape Craft Block Diagram used with External Work

The variates used for the advanced model are listed in the table below. Note that the repair time of a small leak has been changed from the original model. It has been lowered from 4 days to 0.5 day.

Table E-13. Lunar Base with External Mission Variates List

Variable	Description	Distribution	Parameters (unit default = days)
SmLeakMTF	Small Leak MTF	Exponential	$\mu = 240$
MdLeakMTF	Medium Leak MTF	Exponential	$\mu = 5.50E+04$
LgLeakMTF	Large Leak MTF	Exponential	$\mu = 2.70E+04$
MissionMT	Mean time of an external mission	Normal	$\mu = 2.50E-01$ $\sigma = 2.50E-02$
RepairReturnTime	Time required to return to habitat	Exponential	$\mu = 1.25E-01$
HumanMistake	Human error in repair causes Large Leak	Constant probability	0.001
EscapeMT	Escape time	Exponential	$\mu = 2.00E+00$
RepairMT	Time required to repair a Small Leak	Exponential	$\mu = 0.500$

The same number of iterations were used as in the original simulation model. The results are presented in Figure E.15. The output of the simulation is a cumulative total versus time based upon 1000 iterations.

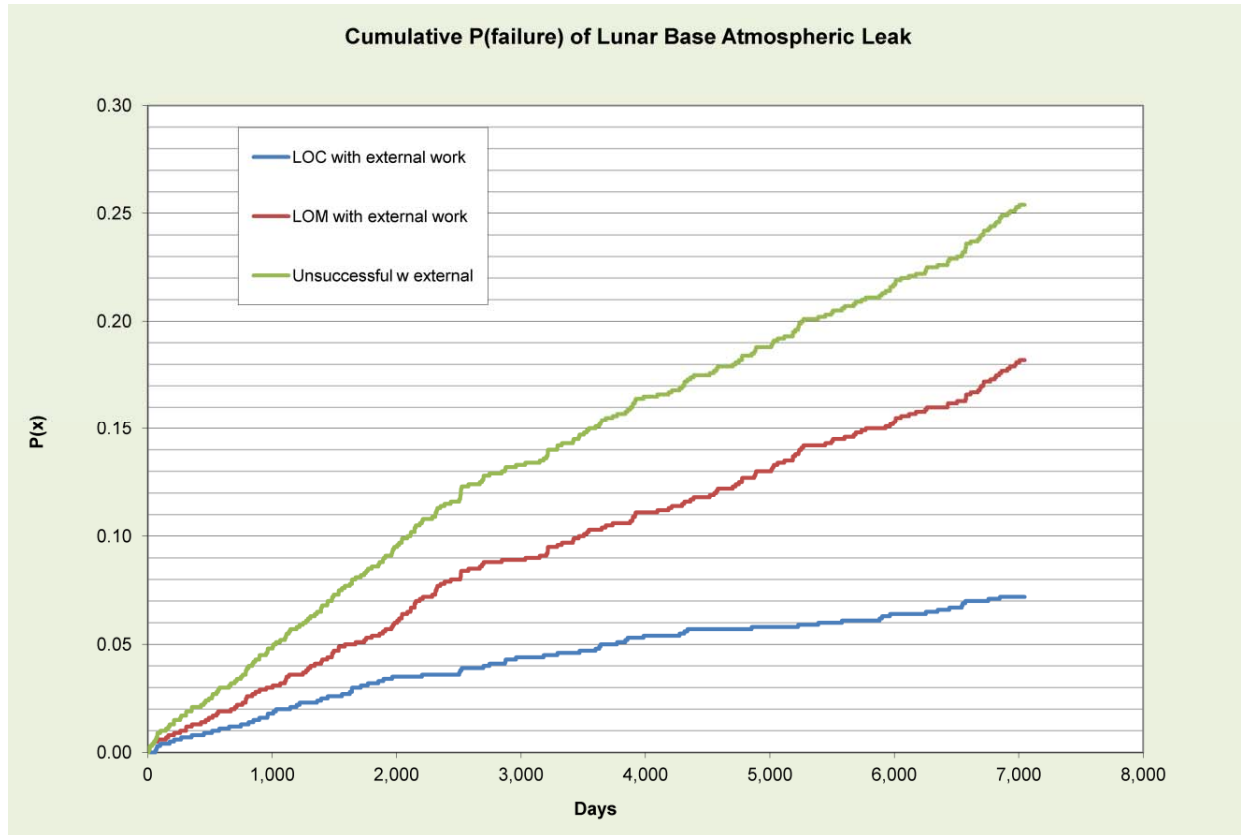


Figure E-15. Results of Lunar Base Atmospheric leak with External Missions added

E.3 Conclusions

Advantages to modeling via simulation:

- Provides a powerful modeling method able to represent highly-time dependent or unique situations
- Intermediate results (e.g., scenarios that are “close” to undesired outcomes) are readily available
- Adding large degree of complexity to the model may only slightly increase analysis time

Limitations to modeling via simulation:

- Uncertainty quantification and “importance-type” measures may be time consuming to model due to need to repeat the calculations
- Representing rare events may be difficult when using “naïve” discrete event simulation
- Modeling tool sets are not widely in use nor accepted



**National Aeronautics and Space Administration
NASA Headquarters
Office of Safety and Mission Assurance
300 E. Street SW
Washington, DC 20546-0001**

