Assembling System/Network Reliability

Jason W. Rupe

Jason W. Rupe, PhD
Qwest Communications International, Incorporated
4001 Discovery Drive
Boulder, Colorado, 80303 USA
Internet (e-mail): jrupe@ieee.org

SUMMARY & PURPOSE

This tutorial will provide reliability specialists with an improved understanding of systems modeling concepts, pitfalls, & techniques to better equip them for working with ever increasingly complex & growing systems, which must be assured to be reliable, available, maintainable, survivable, performable, & meet customer demands.

Systems are generally becoming more complex, and large complex systems are becoming more common. And as the fields of engineering & reliability progress, answers to product demands and solutions to reliability problems can add complexity to systems. In addition, with economies becoming more global each day, systems become larger and more complex. Add to this the idea that the rate of change continues to increase in our world, so we have fewer resources to address reliability concerns in ever increasingly complex and growing systems.

While several companies provide tools which address this problem to a certain extent, the sophistication of users must increase with the complexity of these tools, and the complexity & size of the systems being studied. Otherwise, we run the danger of being in over our heads.

The purpose of this tutorial is to better equip reliability specialists to address reliability concerns under these increasingly critical constraints mentioned above. In it, I assume the audience will have a rudimentary understanding of what simulation is, methods such as FMECA and fault trees, and how to solve a simple Markov model. This tutorial will focus on the system as a whole, and reveal that the way to assure meeting customer needs is to recognize and consider the large system, and the users who interface it.

Jason W. Rupe, *PhD*

Jason Rupe received his BS (1989) and MS (1991) in Industrial Engineering from Iowa State University, and PhD (1995) from Texas A&M University. His research interests are in system quality & reliability analysis, reliability optimization, performability modeling of systems, communication network reliability, maintenance models, and applied stochastic processes. He is a Senior Member of IEEE, and of IIE. Jason is an Associate Editor of the IEEE Transactions on Reliability, a Vice-Chair'n for RAMS, is on the advisory board for IIE Solutions magazine, and is active in his local IEEE and IIE chapters. He is the director of the Modeling and Optimization team at Qwest Communications International, Incorporated, in Boulder, Colorado, and continues to lead reliability efforts within the company. Before that, he worked as the lead reliability Member of Technical Staff with the Mathematical and Statistical Modeling team at U S WEST Advanced Technologies. He has published several papers in respected technical journals, reviewed books, and refereed journals & conference proceedings. Jason received the P. K. McElroy Award for 2000 Best Paper.

Table of Contents

1. Introduction 1
2. Context 1
3. Metrics Linked to Use 3
4. Modeling Concepts 4
5. Modeling Approaches 6
6. Math 8
7. Linking the Mathematical Model to the Metric 9
8. Partial Example 9
9. Conclusions and Discussion 10

1. INTRODUCTION

This tutorial will provide reliability specialists with an improved understanding of systems modeling concepts, pitfalls, & techniques to better equip them for working with ever increasingly complex & growing systems, which must be assured to be reliable, available, maintainable, survivable, performable, & meet customer demands.

Systems are generally becoming more complex, and large complex systems are becoming more common. And as the fields of engineering & reliability progress, answers to product demands and solutions to reliability problems can add complexity to systems. In addition, with economies becoming more global each day, systems become larger and more complex. Add to this the idea that the rate of change continues to increase in our world, so we have fewer resources to address reliability concerns in ever increasingly complex and growing systems.

While several companies provide tools which address this problem to a certain extent, the sophistication of users must increase with the complexity of these tools, and the complexity & size of the systems being studied. Otherwise, we run the danger of being in over our heads.

The purpose of this tutorial is to better equip reliability specialists to address reliability concerns under these increasingly critical constraints mentioned above. In it, I assume the audience will have a rudimentary understanding of what simulation is, methods such as FMECA and fault trees, and how to solve a simple Markov model. This tutorial will focus on the system as a whole, and reveal that the way to assure meeting customer needs is to recognize and consider the large system, and the users who interface it.

1.1 Notation and Acronyms

RAMS	Reliability, Availability, Maintainability, and Survivability
HALT	Highly Accelerated Life Testing
HASS	Highly Accelerated Stress Screening
FMS	Flexible Manufacturing System
S	State Space, eg. {0,1,2} or {(0,0),(0,1),(1,0)..}
A	Matrix of transition rates from row # to column #, with the diagonal being the negative of the sum of the remaining row entries. AKA, Infinitesimal Matrix
J	Matrix of 1's
I	Identity matrix, 1's on diagonal, 0 otherwise
X^{-}	Inverse of matrix X
T	Time interval under consideration
Pij	Probability of being in state j at a future time, given starting in state i at time zero
ρ	reward matrix
Wij	cumulative reward

1.2 Organization

In this tutorial, I will present much of the softer content that is so very important to modeling systems & networks, yet is not taught. In fact, much of this content I mention is not acknowledged, practiced, or used by most reliability practitioners, but certainly should be. In this tutorial, I will present some mathematics, but very little - only what is necessary to convey the messages.

The tutorial follows this order (Slide 2):

- Context, why model, what is RAMS modeling, and how does it interface with testing.
- Measurements, measures of effectiveness, performability, customer perspective.
- Elements of a system/network model.
- Approaches to modeling.
- System/network modeling frameworks and other resources.
- Mathematics to measures
- Common Errors

What I will not cover in this tutorial is just as important (Slide 3):

- Statistical Methods for collecting failure information or analyzing model results.
- How to build a simulation.
- Exciting stochastic process research.
- The importance of FMECA, FRACAS, etc.
- How to perform a repairability study.
- What is the best software to buy.

While some of these items are not requirements of performing a useful RAMS study of a system or network, some of these are either necessary, or at least useful.

2. CONTEXT

Mathematical modeling of RAMS, for it to be truly valuable, must meet several business needs. From a cost standpoint, earlier is better. There are also some synergies to exploit between modeling and testing. For modeling to be useful, it is most important to consider the system or network, its intended functions, the service metrics associated with it, and the customer usage.

2.1 Why Model

Many of the reasons for modeling the RAMS of a system or network may be obvious, but there are some reasons that require more discussion (Slide 4).

Prediction. Perhaps the most obvious reason for modeling RAMS is to predict the performance of the system/network.

Testing Prohibitive. Early in the development phase of a product, or before a system or network is integrated and tested, we can use modeling to predict how the result will serve a desired function.

Incomplete Testing. Most of the time, it is not possible to test a complete system or network until the system or network is complete. Remember, we include the user's perspective and intended use of the system or network. Without some modeling involved, it is difficult to assess this impact.

Guide Testing. Modeling can reveal insight into a system or network that can guide testing. For example, a system can be sensitive to the success of redundant component switchover, which should therefore be thoroughly tested.

Efficient Discovery. Modeling can reveal system or network insight, often more efficiently than testing. Modeling tools such as Monte Carlo simulation of uncertainty can be set up easily in models.

Drive Design. Modeling, by providing insight, can drive design decisions early in the design cycle.

Compare Options. While testing can compare options, the resources required can be large. Modeling can compare options in design, quickly in some cases.

Cost of Reliability. The cost of improving reliability in a design increases significantly as the product matures, or as time elapses. This cost curve might be exponential, but is certainly increasing significantly. Figure 1 (Slide 5) is an example of this curve.

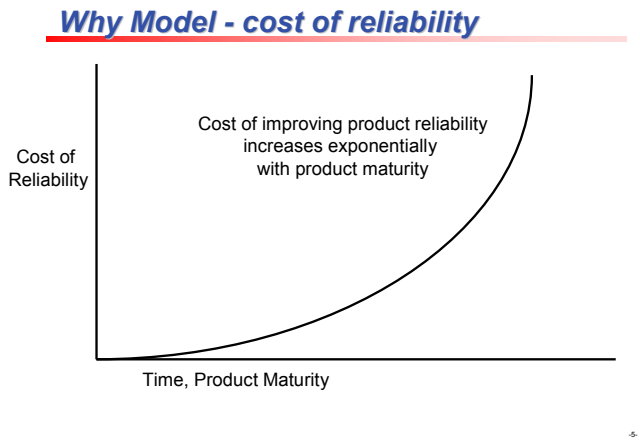


Figure 1: Cost of reliability over time.

2.2 Modeling, and Accelerated Testing & Screening

Modeling helps predict and testing helps improve system or network reliability, but together they can do both much better.

Some HALT/HASS experts are against RAMS modeling because they recognize how easy it is to misuse modeling, or misapply the results. They argue that HALT/HASS techniques are the direct best way to improve reliability, and that is all that matters. However, I contend that some systems and networks should not be optimized for reliability. In addition, while HALT/HASS techniques can be exceptionally inexpensive ways to improve reliability, there are many failure modes of systems and networks that these techniques cannot address.

Instead, modeling can be used where testing cannot. And both can compliment each other. In early design phases, or when systems or networks are first envisioned, modeling can guide the design, and find where to focus early testing to create fast, effective designs. In addition, HALT/HASS can improve the reliability of products so that more is possible at the system and network level (Slide 6).

2.3 What is Systems/Network Reliability Modeling?

By systems and network reliability modeling, we mean applying mathematical techniques to predict performance and effectiveness metrics for large systems or networks. Often these metrics are related to reliability, availability, maintainability, and survivability (RAMS), so we can term it "RAMS Modeling". Everyone has their own approach to this type of modeling; software applications you can purchase utilize various mathematical results (Slide 7).

Hierarchy. A system or network is the largest combination of elements, while a component is the smallest. A network, we say, is composed of systems working together. A system, then, is composed of sub-systems, which, as convenient, can be composed of smaller sub-systems, until we are at the lowest level of detail, the part or component level. For convenience, if the highest level in the hierarchy is not a network but rather a system, then we drop levels to sub-system, part, to (possibly sub-part or) component. This nomenclature is depicted in Figure 2 (Slide 8).

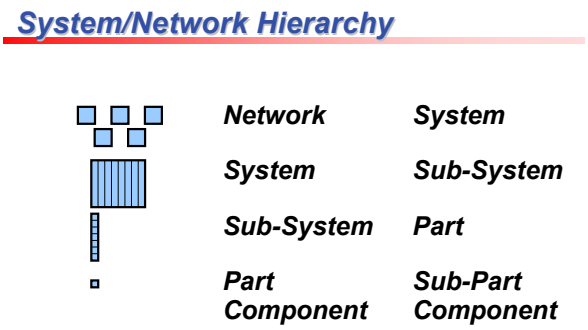


Figure 2: Conducive hierarchy.

States and Transitions. While modeling for RAMS, it may be convenient to think about a network, system, sub-system, or component as existing in states, and transitioning between these states. Petri-nets, stochastic models (including stochastic activity networks), and most modeling techniques applicable to RAMS problems have states and transitions between them.

A system or network exists to perform a function. RAMS modeling is the concern of how effectively the system or network, or a portion of it, performs its intended function. We are concerned with how effective it is at performing its intended function, from the perspective of the user. We model the change in performance so we can predict the measures of performance and effectiveness.

A state represents the condition of the subject, and a transition between states can result in a change in performance

level. We can further divide states to make the modeling more convenient, or for any number of reasons (Slide 9).

RAMS Modeling. RAMS modeling is not always given the attention it needs.

The common approach to answering "the reliability question" in non-life-threatening commercial products (at least in my industry) is over simplified. Reliability engineers are often asked to construct the answer for sales managers to deliver to potential customers. First, the engineer, who possibly is contracted for the task, will find a block diagram, conduct a part-count-method-estimate of component failure rates, and calculate an availability metric from some chosen perspective for the product. Figure 3 (Slide 10) is a depiction of what I often observe.

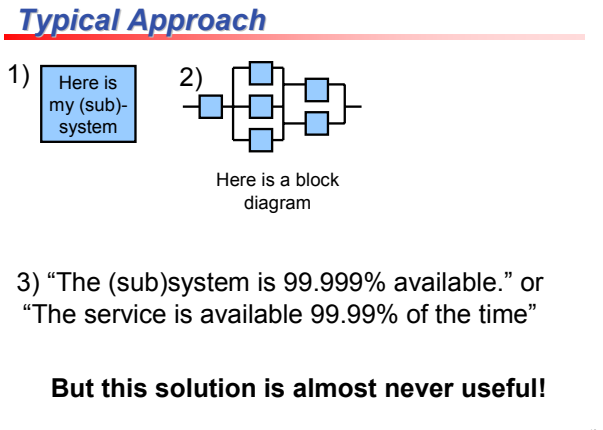


Figure 3: A typical approach to addressing reliability.

This static approach is seldom useful because the model is seldom sufficiently accurate, usually does not answer the important questions, and the knowledge gathered is often ignored.

A more thoughtful approach is to follow these steps, perhaps with iteration as needed (Slide 11):

- Identify the system/network under consideration. If you are primarily concerned with a sub-unit of the system or network, consider how it interfaces with the rest of the system or network components.
- Identify the functions, and intended service. Consider the use by the end user, and their perspective of the functions intended.
- Define service metrics and requirements. The end user should drive theses metrics and requirements when possible. Sometimes the technology will drive the metrics, which must then be translated into requirements.
- Create a design, and model the RAMS performance and effectiveness. This step involves sub-steps: 1) obtain estimates of failure rates, 2) define the state space and transitions between states for the system or network components, and 3) calculate estimates for the required metrics. We will cover more here later in this tutorial.
- Predict the service delivered through the design. In other words, use the model, with appropriate mathematical tools, to determine the performance and effectiveness deliverable through the system or network design.
- Re-evaluate the design. In light of what is revealed through the previous tasks, evaluate the design, determine where to improve, and use a closed loop corrective action process to affect the designs, and redo the above steps as needed.
- Follow up with testing and continue to re-evaluate the designs. Even as testing becomes possible, work with testing results, and drive testing, to affect the designs as needed. For example, testing may reveal that a less expensive option may be sufficient to deliver the desired RAMS performance and effectiveness.

3. METRICS LINKED TO USE

3.1 Perspective and Metrics

The perspective drives the metrics. For example, from the perspective of the component manufacturer, the number of returns is a concern. Therefore, the intended use of the product drives the modes of failure, and the infant mortality is most important. From the perspective of the end customer of the service being provided by the system or network, the service quality is most important. Reliability, availability, and perhaps survivability of the system or network should be measured. Table 1 (Slide 12) shows some additional links between user perspective, and the metrics needed.

Perspective Example

Perspective	Concern
Component Manufacturer	Reduce Returns
Subsystem Manufacturer	Subsystem Availability
System/Network Provider	Repair Rate, Protection Rate, System Availability
Customer	Service Quality/RAMS

Table 1: Various possible perspectives for "reliability".

3.2 Metrics

While many metrics describe various performance characteristics of a system or network, or even parts of these, some are just metrics that describe the performance, while others describe the effectiveness of the system or network at meeting the intended functions for the end user.

There are many metrics typically predicted through various forms of modeling. At the rudimentary level, these include failure rates, failure modes, criticality of failure, repair rates,

protection rates, and other statistics that form inputs to the system or network model.

Measures of Performance. Several measures of performance are commonly considered. Definitions are readily available, and often change according to perspective. Each provides some measure of how the system or network performs, or perhaps how a part of the system or network performs. An example list is in Table 2 (Slide 13) of the tutorial presentation. While these measures of performance are useful for comparisons, they seldom are sufficient to describe the impact to the end user under the perspective chosen. For that, we need a measure of effectiveness.

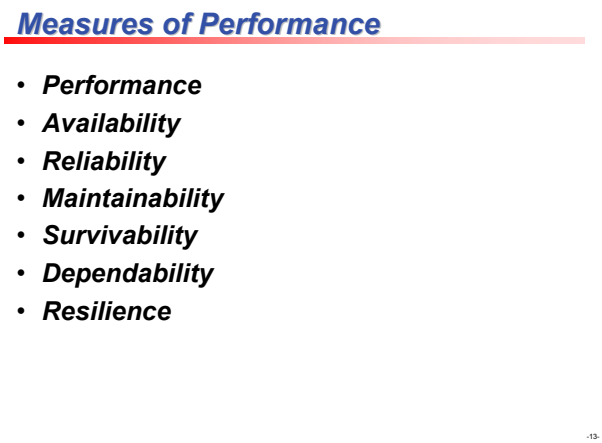


Table 2: Possible Measures of Performance.

Measures of Effectiveness. There are few measures that are complete enough to reach the perspective of an end user, and be what I consider a measure of effectiveness. This is probably because these measures are so broad (Slide 14).

Performability - the probability that a system or network performs at a specified level of service. The result is a performance assignment to each system or network state, and the likelihood that the system or network is in each state. Sometimes we can use this function to derive single measures that characterize effectiveness at meeting the needs of particular missions, or functions, from a user perspective.

Mission Effectiveness - the ability of a system or network to perform some intended function as compared against some standard. This definition is vague enough to capture most any set of measurements that might be sufficient to describe the impact to the end user.

3.3 Defining Metrics

Every RAMS measure has at least 3 dimensions to consider (Slide 15):

1. Perspective - the point of view from which to measure.
2. Time - steady state, or transient.
3. Dimensionality - point, closed interval, or open interval/vector.

Some resources consider the second and third point, but the model, and the metric, is driven by all three.

Metric Definition Process. The dimensions listed above drive a process for defining the measure needed (Slide 16).

First, determine the perspective to be modeled. For the end user, they will almost surely have the perspective of service quality. For the provider of the system or network, they will have the perspective of the system or network, and perhaps for maintenance reasons need metrics around the sub-systems or components. The component or sub-system provider will take the perspective of the component or sub-system quality.

Second, determine the time dimension. If we are interested in an indefinite future time for the mission or function, then a steady state metric is needed. If we are instead interested in a specific time or set of conditions for the mission or function, then a transient solution is needed.

Third, determine the dimensionality of the metric. If the service or function depends on the system or network condition at a point in time, then the effectiveness at that point in time is needed. If the service or function takes place over a period of time, then a closed interval is needed. If the service or function continues indefinitely, or we are interested in knowing how long a function can be met with some certainty, then an open interval or vector is needed.

Literature. Several examples of how others view the selection of metrics exists in literature. Two notable examples are those of Singh & Billington, and Sanders & Meyer, in Table 3 (Slides 17) and Table 4 (Slide 18) respectively.

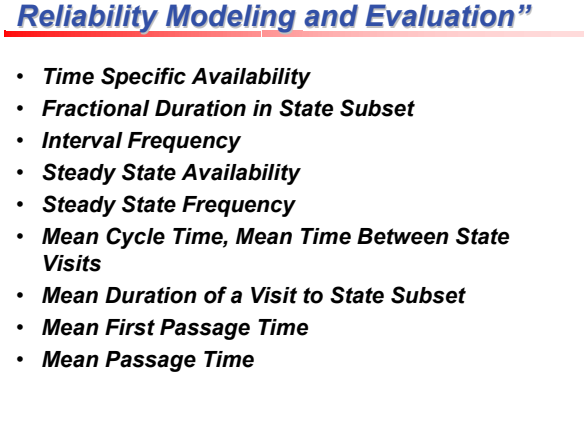


Table 3: One proposed list of metrics.

4. MODELING CONCEPTS

Several system and network concepts must be reflected in modeling concepts.

4.1 Survivability

The term survivability is often used to refer to the ability of a system to continue to perform its intended function after significant failures, with no impact to the amount of service it provides, but possibly at a loss of service in progress. Sometimes this definition allows for a partial loss of service. For example, a survivable communications network can survive a failure of a core switch because there is a redundant

switching system connected and available with enough capacity to fulfill the functions needed from a core switch. Figure 4 (Slide 19) shows this concept.

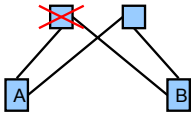
W. H. Sanders, J. F. Meyer - "A Unified Approach for Specifying Measures of Performance, Dependability, and Performability"

- ***Instant of time***
  - **Finite instant of time**
  - **Instant of time as time approaches infinity**
- ***Time-averaged Interval of time***
  - **Finite instant of time, beginning and ending**
  - **Starting time approaches infinity**
  - **Ending time approaches infinity**
- ***Interval of time***
  - **Finite instant of time, beginning and ending**
  - **Starting time approaches infinity**
  - **Ending time approaches infinity**

-18-

Table 4: Another proposed list of metrics.

System/Network Survivability



Survivability is often used to refer to the ability of a system to survive significant failures, with no impact to the amount of service it provides, but possibly at a loss of service in progress.

-19-

Figure 4: Survivability illustrated.

4.2 Detection and Protection Switching

For any system or subsystem to survive, a failure must be properly detected, and a successful protection switch must complete. The purpose of these actions is to alert the repair process to a failure, and to transfer the responsibility for delivering the intended function from a primary to a secondary delivery mechanism. This process can be automatic, as in automatic protection switching (APS), or manual. Protection mechanisms can include (Slide 20)

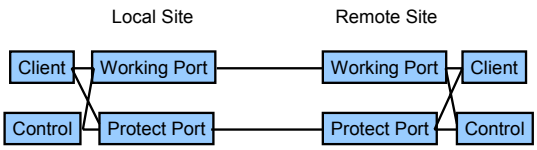
- 1:N, where a single secondary unit awaits to protect the first of N primary devices that fail. The secondary unit may be functioning, but will need to perform some functions before being able to take over for the primary. N can be any positive integer.
- 1+1, where a secondary unit is either shadowing the service provided by the primary, or sharing capacity with the primary unit, and in either case is ready to take over complete service from the primary. In the latter case, the definitions of primary and secondary is arbitrary.

These recovery mechanisms may be software based, hardware based, or a combination.

Of concern in the modeling effort is the probability of successful detection, and then successful protection. Successful protection occurs when a failure is successfully detected, the secondary device is able to take over, the primary device relinquishes control, and the secondary device takes over. Depending on the protection mechanisms, failure modes, and function of the devices, the likelihood of successful detection and protection can differ. We must also consider the likelihood of false detection and false protection switching; and the impact that results on the system effectiveness.

Figure 5 (Slide 21) shows an example physical diagram, while Figure 6 (Slide 22) shows a possible availability block diagram for the same system. Table 5 (Slide 23) describes a cold standby 1:1 protection scheme. Table 6 (Slide 24) describes a hot standby 1+1 protection scheme.

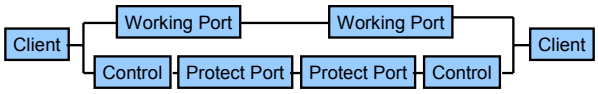
Example Physical Diagram



-21-

Figure 5: An example physical or logical diagram.

Example Availability Diagram



Flow through the system means it works.

-22-

Figure 6: An example availability diagram.

Example 1: Cold Standby, 1:1

- ***Failure - Port Card Laser Drops Signal***
- ***Detection -***
  - **Local - Heartbeat process on control card detects failure**
  - **Remote - Port card detects signal loss, informs control card**
- ***Checking -***
  - **Local - Control card checks status of protect card, and readies**
  - **Remote - Same**
- ***Protecting -***
  - **Local - Control card starts protect card**
  - **Remote - Same**
- ***Alarming -***
  - **Responsible control card signals an alarm**

-23-

Table 5: Cold standby example description.

Example 2: Hot Standby, 1+1

- ***Failure - Port card laser drops signal***
- ***Detection - Remote port card detects loss***
- ***Checking - Remote port card signals to its client to switch to protect port card, which is known to be available***
- ***Protecting - Remote protect card provides function, Local protect card provides function***
- ***Same happens on local side as well***

-24-

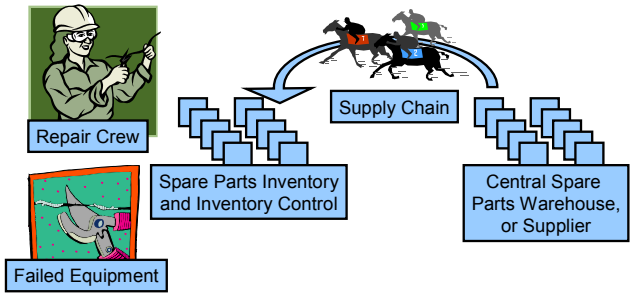
Table 6: Hot standby example description.

4.3 Repairs and Spares

Depending on the purpose for modeling, we may need to consider with some degree of care the repair and spare parts processes. These processes affect repair time. For a repair to complete successfully, all resources needed for the repair must be available. The repair crew must be able to work on the failed equipment, and the spare parts must be on site when needed. If it is possible that either the repair crew or materials such as spare parts may not be available to begin a repair immediately when needed, and if a simple repair distribution is not sufficient for the measures of effectiveness needed, then we must also model the repair crew and spare parts inventory process.

The spare parts inventory process can be a complex supply chain. Spare parts may be kept on site with the systems, and/or clustered at some locations, and/or clustered at a central site, and/or repaired at locations, and/or ordered from the manufacturer. Depending on the process used for spare parts, the model for spares availability can be complex. Figure 7 (Slide 25) shows this concept.

Repairs and Spares



-25-

Figure 7: Depiction of repairing and sparing concepts.

5. MODELING APPROACHES

Here we will discuss what and how to model, and cover some examples of small model descriptions.

5.1 What to Model

Generally, consider what you need to predict, the detail you need, the information you have, and the information you do not have. As we have already covered, the processes, system or network, subsystems and components, services and the perspectives of the customers all define what to model. For sake of simplicity, model what matters and ignore unnecessary details. When there are details that are unknown or not sufficiently known, but are nonetheless needed, then explore the sensitivity of the associated parameters (Slide 26).

5.2 How to Model

For efficient modeling (Slide 27):

- Avoid the "curse of dimensionality", or model size. Direct mathematical approaches often suffer from the problem that the size of the model becomes too weighty to be efficiently solved. While stochastic modeling approaches are known to suffer this problem, simulation is seldom recognized to suffer as well. But it does. Large, complex problems are more difficult to validate & verify, and take longer to solve to a desired level of confidence. In any case, search for ways of dividing the system or network into independent or simply related functions that can be modeled separately.
- Model only the performance measures you need. Do not waste resources on unnecessary modeling details. For example, if you can reduce the number of states in your model structure without sacrificing accuracy, do.
- Model to the right level of detail. Be sure to model all the details you need as well. If you chose to simplify a model, understand what you sacrifice and be sure it is worth it.
- Choose the right modeling approaches. If static probability models are sufficient, they probably will be

easiest to use. If they are not sufficient, then stochastic process modeling methods may be needed. Sometimes, these cannot be solved adequately, so simulation becomes useful. Use the simplest sufficient modeling approach available (Slide 28).

5.3 Example Models, and Frameworks

K-out-of-N. The graphical description of this model is in Figure 8 (Slide 29). State 0 represents the fully functioning state, where all N components are functioning. Each state n represents the state where n components are failed. Transitions to higher states are failures, and transitions to lower states are repairs.

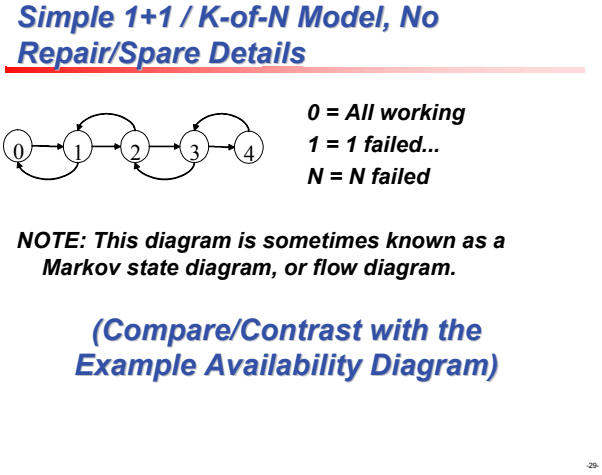


Figure 8: Example state diagram.

This model is useful for representing the behavior of a system of N components or subsystems that share a load, or for N=2, a simplified representation of a working and protect scheme.

Simple Software. The graphical description of this model is in Figure 9 (Slide 30). State 0 represents the fully functioning state, where the software continues to meet its intended function. State 1 represents when the software undergoes a standby update. State 2 represents when the software undergoes a hitless rebuild. State 3 represents when the software undergoes a full rebuild and is down, or is down for other reasons. Transitions represent movements between states as possible by the behavior of the modeled software.

This model is a simple representation of software, and does not differentiate between software functions.

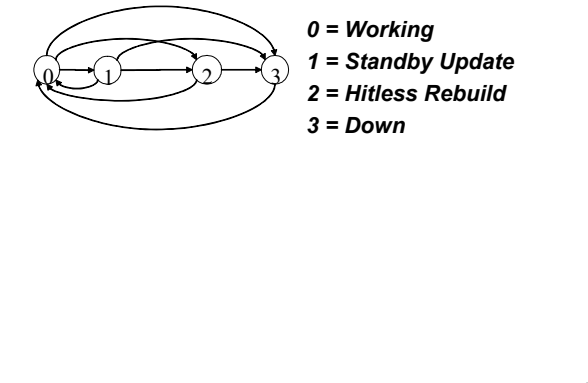


Figure 9: Simple software model.

Simple 1:1with Imperfect Protection. The graphical description of this model is in Figure 10 (Slide 31). State 0 represents the fully functioning state, where the working and protect components are both functioning. State 1 represents a failure of the primary component that is not (yet) detected. State 2 is the state for a detected primary failure. State 3 is a switch to the protect component meeting the function. State 4 is the state for both the primary and secondary components failed. State 5 is a detected secondary component failure, and state 6 is an undetected secondary component failure. Transitions from an undetected state to a detected state are possible. When both components are down, both are replaced at once in this model.

The likelihood of successful detection of a failure impacts the delivery of the intended function in this model. Also, the likelihood of a successful protection switch is important.

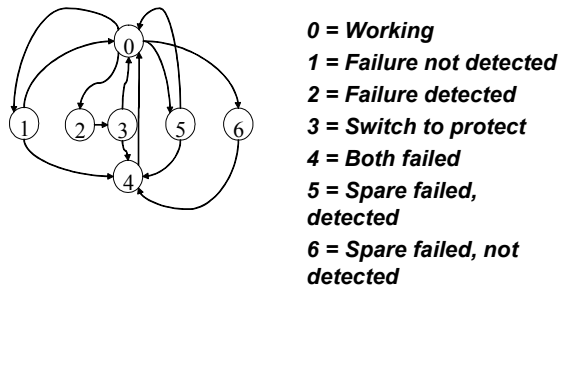


Figure 10: Simple imperfect coverage model.

A System of Machines with Spares and Repair. The graphical description of this model is in Figures 11 & 12 (Slides 32 & 33), with the network being a flexible manufacturing system (FMS).

The top level is the FMS. This FMS consists of several systems of machines. Each system consists of one or more

sub-systems of machine types. Each machine type has a number of machines that are identical, and the machine type has some number of machine part types, with each machine of the same type having the identical machine part types. The machine part types are the components in our hierarchy.

The model represents an individual machine part type. Each part type is modeled for each machine type. The state space is two dimensional (i,j) with i representing the number of spares needed to be replaced, and j being the number of failed machines. A machine can fail, and be repaired with the part type being modeled, or with another part type. A spare part can arrive, or fail to be useful before installed.

This framework is better described in the work by Rupe & Kuo, along with Markov and renewal process models for this framework.

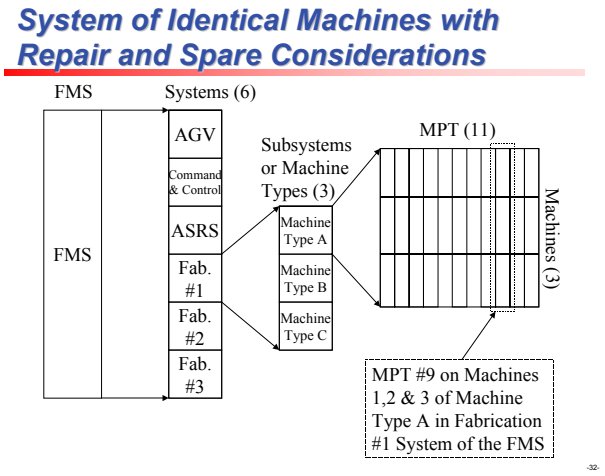


Figure 11: Hierarchical description of FMS.

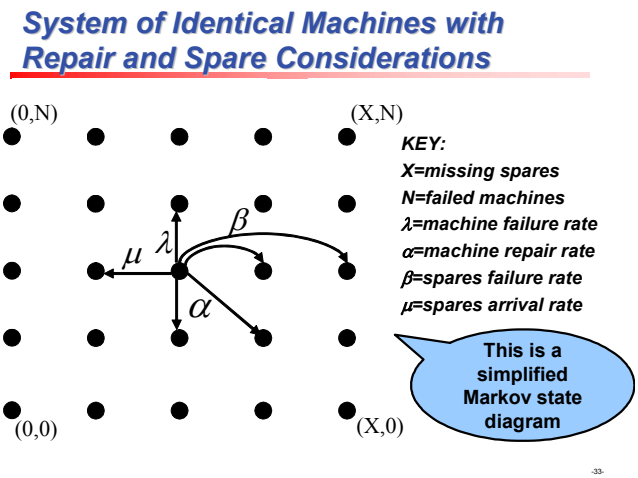


Figure 12: State diagram for system of identical machines.

6. MATH

I classify the application of mathematics to the model description into three categories (Slide 34):

1. Static Probability Methods
2. Stochastic Process Methods - Markov, semi-Markov, and regenerative process models
3. Simulation Methods - discrete versus continuous, event oriented versus process oriented.

In this presentation, I will focus on stochastic process methods, Markov in particular, because this is an introductory tutorial. For static probability methods, refer to the tutorials by Bowles, and by Leemis, or consult an introductory textbook on probability such as those listed in the References. For simulation methods, refer to the tutorial by Dubi, or a textbook on simulation such as those listed in the References.

6.1 Static Probability Methods

One important point worth mentioning about static probability methods is that they are simple and can often be used to simplify otherwise complex models (Slide 36).

- Sometimes, simple static probability methods can be used to simplify a simulation or Markov model. For example, if two states are identical except for the performance characteristic in each state, you may be able to combine the states into one, and use static probability methods to divide the state after solving the larger model.
- Sometimes static probability methods can be used to combine separate Markov or simulation models into system or network level results.

6.2 Simulation Methods

Simulation methods can be used in conjunction with Markov methods.

- Sometimes, Markov methods can replace decomposable parts of the simulation, reducing the uncertainty inherent in simulation methods.
- Sometimes simulation methods, such as Monte Carlo, can enhance Markov models. Monte Carlo methods can be used with stochastic models to explore sensitivity to uncertainty of parameters.

6.3 Stochastic Methods

In the rest of this section on mathematical modeling, I will concentrate on Markov methods, which are simple and useful. To keep it simple, I will assume we can model what we need to with a finite set of states, that all states communicate, and that we can obtain the necessary eigenvalues for the models (Slide 37).

In this section, we will cover simple matrix algebraic methods for solving steady state, transient, and reward rate metrics.

Steady State (Slide 39). First, create the infinitesimal matrix A, then calculate

$$\Pi = I * (A + J)^{-} \qquad (1)$$

which is the steady state probability matrix. Each element is the probability of the model being in a given state at some far future time (Slide 40).

Transient. First, again create the infinitesimal matrix A. Then find the maximum absolute transition rate in A, which will be on the diagonal of A. Define Λ to be larger than the

maximum absolute rate just found. Define PN to be an identity matrix the same size as A, then calculate

$$P = (A/\Lambda) + I \qquad (2)$$

Then calculate recursively Equations 3 and 4 below over k=0 to n until you reach the desired accuracy in the calculated figures:

$$PN = P * PN \qquad (3)$$

$$Pij = (PN \times e^{-T\Lambda} \times (T\Lambda)^k / k!) + Pij \qquad (4)$$

and this is approximately the probability of transitioning from i to end in j by time t, with arbitrary accuracy. This approach is known as uniformization or randomization (Slide 41).

The stopping conditions for this algorithm are to check that the row sums of Pij are all as close to 1 as you need, or the total differences in the row sums are as close to the number of rows as you need.

This algorithm provides the expected proportion of time spent in each state. To find the probability of passing through a given state at least once, make the state absorbing by making the corresponding row all 0's in A.

For cases of large state space, low transition rates, and/or short time intervals, refer to the work of A. P. A. Van Moorsel & W. H. Sanders for adaptive uniformization, and Carrasco for regeneratvie randomization which may perform better.

Rewards (Slide 42). It is simple to apply a reward rate or point reward for each visit to a state within the randomization algorithm. Replace Equation 4 with

$$Wij = (\rho \times PN \times e^{-T\Lambda} \times (T\Lambda)^k / k!) + Wij \qquad (5)$$

or if the reward matrix is, say, a reward on the entering state, then

$$Wij = (\rho j * PN \times e^{-T\Lambda} \times (T\Lambda)^k / k!) + Wij \qquad (6)$$

Renewal Theory (Slide 43). Renewal Theory provides steady state or transient probabilities for a given state space. Reward rates or rewards can be applied to these models, with care. For example, between renewals, the (sub)system may be modeled as a Markov process to which we can apply the above methods, and account for the number of renewals in the time interval for transient calculations.

7. LINKING THE MATHEMATICAL MODEL TO THE METRIC

The metric choice drives the model, as we have said. It also drives the mathematical methods we choose to apply (Slide 44).

7.1 Metric Dimensions, Models, and Mathematical Methods

Recall the three dimensions we proposed for a metric.

- Perspective - this defines the model details such as the state space, transition rates, etc.
- Time - this defines whether we need to calculate the steady state probability distribution, and/or transient solutions.
- Dimensionality (Slide 45) - this defines the reward structure, if needed, and whether you need to define absorbing states. More specifically,
  - Point - use uniformization with no reward rate, but repeat for each absorbing state.
  - Closed Interval - use uniformization with a reward rate, if the model structure allows it.
  - Open Interval/Vector - use uniformization with a reward rate, and steady state with a reward rate for the tail. I have not seen this tried, and I have no reference or proof that it will work. I have not seen a need for this kind of metric either. But I suggest you could use these methods for an approximate solution.

These guidelines are represented in Table 7 (Slide 46).

Linking the Measure Selection with the Calculation Method

	Steady State	Transient
Point	Steady state calculations define initial state probability matrix	Randomization defines initial state probability matrix
Closed Interval	Begin at steady state, use randomization with reward for the interval	Randomization defines initial state probability, randomization with reward for the interval
Vector	Begin at steady state, use randomization with reward for the interval	Randomization defines initial state probability, randomization with reward for the interval*

* (no reference or proof)

-46-

Table 7: Instructions for linking the calculation method with the desired measure.

7.2 Complex System/Network

Chances are we will be applying the above guidelines to models of a set of like components, or at least to subsystems. We will then need to aggregate our results to a system or network level. In this case, rather than using a reward calculation inside the model, we need probability distributions or proportions of time spent for the states, and we combine these states according to the system or network performance characteristics. This is an application of static probability methods to the results of Markov or renewal models (Slide 47).

8. PARTIAL EXAMPLE

8.1 Description of Example

To serve as an example of applying the framework and mathematical approaches provided in this tutorial, consider an example FMS (Slide 48):

- The FMS contains two sets of identical machines.
- Each of the first set of identical machines contains three parts, each with a failure rate of 0.03 per hour. The repair rate of machines is 0.2 per hour. Spare Parts

arrive at a rate of 0.1 per hour, and fail to function as a spare at rate 0.005 per hour.

- Each of the first set of identical machines contains three parts, each with a failure rate of 0.02 per hour. The repair rate of machines is 0.4 per hour. Spare Parts arrive at a rate of 0.1 per hour, and fail to function as a spare at rate 0.01 per hour.
- In both sets of identical machines, each machine is statistically equivalent.
- The time interval for completing the batch is 40 hours. At full capacity, the FMS can complete the batch in 30 hours. At the start of the 40 hours, the entire FMS is functioning.

We seek to determine two things, from the perspective of the FMS user (Slide 49):

1. Do we expect to complete the batch on time?
2. What is the probability that the batch completes on time?

8.2 Solutions

Apply the FMS modeling framework presented above, and the set of equations provided by Rupe & Kuo (Slide 50).

For the first set of machines, the expected proportion of time that any of the three given machines is functioning is 0.7818248.

For the second set of machines, the expected proportion of time that any of the three given machines is functioning is 0.9401467.

Table 8 (Slide 51) presents the application of state probability to the performance at the FMS-level. The bottom of the figure is a performability calculation.

We expect to complete the batch on time. However, this does not tell us the probability that the batch completes on time. We need a complex reward rate and single unified system-level model (Slide 52).

Performability

System 1 State	System 2 State	System 1 Proportion	System 2 Proportion	FMS Performance	State Performability
1 Machine	1 Machine	0.1116456	0.0101040	0.4444444	0.0005014
1 Machine	2 Machines	0.1116456	0.1587087	0.4444444	0.0078752
1 Machine	3 Machines	0.1116456	0.8309729	0.4444444	0.0412331
2 Machines	1 Machine	0.4000788	0.0101040	0.6666667	0.0026949
2 Machines	2 Machines	0.4000788	0.1587087	0.8888889	0.0564409
2 Machines	3 Machines	0.4000788	0.8309729	0.8888889	0.2955152
3 Machines	1 Machine	0.4778904	0.0101040	0.6666667	0.0032191
3 Machines	2 Machines	0.4778904	0.1587087	1.3333333	0.1011271
3 Machines	3 Machines	0.4778904	0.8309729	1.3333333	0.5294853
				FMS Performability	1.0380922

-51-

Table 8: Results for the example.

9. CONCLUSIONS AND DISCUSSION

9.1 Conclusions

I hope you have concluded as have I that (Slide 53):

- Modeling is a useful tool.
- It is important to think carefully about what you want to model, and how.
- It is important to think about the final use of a component or subsystem, in the context of the user, service, and system or network.
- Clever separation helps avoid the dimensionality curse, and improves results.
- Some simple tools work well with Markov modeling.

9.2 Common Mistakes

From the material I presented in this tutorial, you could probably predict many of the common mistakes. A few errors I have seen, some of which were quite significant to the decision making process, are described below (Slide 54).

- Combining independent failures incorrectly. It is highly inaccurate to model a system as experiencing only one component failure at a time when the working components continue to provide service. Instead, each set of like components behaves as a sub-system, and the system can experience failure modes within each sub-system.
- Over simplification of the model. Systems with dubious protection schemes should attempt to predict (through testing or experience) the success of the protection scheme, and account for it.
- Not considering the customer, service, or function. It is not very useful to know the proportion of the time that a sub-system is functioning to some degree. It is more useful to know the performability profile, or the right information to predict the customer's experience.
- Not understanding the product. It is important to correctly model the sub-system being sold as a solution to a system or network design. If a vendor cannot provide accurate models for RAMS metrics, it suggests that the engineers do not know the sub-system well, and it presents doubt that the sub-system will perform well.
- Assuming that a fully redundant system meets any requirements. The real performance and effectiveness of a sub-system is affected strongly by the intended function, use, and perspective of the user. While a fully redundant system may be highly available from some perspectives, it may not be reliable. It may still cause poor service from the perspective of the customer.

REFERENCES

1. B. S. Abbas, W. Kuo, "Stochastic Effectiveness Models for Human-Machine Systems," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 20, No. 4, 1990, pp. 826-834.
2. P. Bratley, B. L. Fox, L. E. Schrage, *A Guide to Simulation*, 1987; Springer-Verlag.
3. J. A. Carrasco, "Computing Transient Dependability/Performability Measures from Irreducible Markov Models using Regenerative Randomization," *IEEE Transactions on Reliability*, to appear.
4. W. Kuo, B. S. Abbas, "Performance Modeling for Multiple Human/Machine Systems," *Journal of Analytical Modeling and Simulation,* Vol. 11, 1993, pp. 57-71.

5. K. W. Lee, J. J. Higgins, F. A. Tillman, "Stochastic Models for Mission Effectivemenss," *IEEE Transactions on Reliability,* Vol. 39, No. 3, 1990, pp. 321-324.
6. J. Meyer, "On Evaluating the Performability of Degradable Computer Systems," *IEEE Transactions on Computers,* Vol. C-29, August 1980, pp. 720-731.
7. J. Meyer, "Closed-form Solutions for Performability," *IEEE Transactions on Computers,* Vol. C-31, July 1982, pp. 648-657.
8. A. L. Reibman, "Modeling the Effect of Reliability on Performance," *IEEE Transactions on Reliability,* Vol. 39, No. 3, 1990, pp. 314-320.
9. J. Rupe, W. Kuo, "Performability of Systems Based on Renewal Process Models," *IEEE Transactions on Systems, Man, and Cybernetics,*Vol. 28, No. 5, 1998, pp. 691-698.
10. J. Rupe, W. Kup, "Performability of FMS Based on Stochastic Process Models," *International Journal of Production Research,* to appear.
11. W. H. Sanders, J. F. Meyer, "A Unified Approach for Specifying Measures of Performance, Dependability, and Performability," *Dependable Computing for Critical Applications,* Vol. 4, Dependable Computing and Fault-Tolerant Systems, Springer-Verlag, 1991, pp. 215-237.
12. W. H. Sanders, J. F. Meyer, "Reduced Based Model Construction Methods for Stochastic Activity Networks," *IEEE Journal on Selected Areas in Communications, Vol. 9, No. 1, January 1991, pp.* 25-36.
13. S. R. Shier, *Network Reliability and Algebraic Structures,* 1991; Oxford University Press, Clarendon Press.
14. C. Singh, R. Billington, *System Reliability Modelling and Evaluation,* 1977; Hutchinson & Co. Ltd. of London.
15. A. P. A. Van Moorsel, W. H. Sanders, "Adaptive Uniformization," *Communications in Statistics: Stochastic Models,* Vol. 10, No. 3, August 1994, pp. 619-648.
16. B. Vinod, S. S. Solberg, "Performance Models for Unreliable FlexibleManufacturing Systems," *Omega, The International Journal of Management Science,* Vol. 12, No. 3, 1994, pp. 299-308.
17. L. Xing, J. Bechta Dugan, "Analysis of Generalized Phased-Mission System Reliability, Performance, and Sensitivity," *IEEE Transactions on Reliability,* to appear.