Flexible Modeling and Simulating Mission Availability Within The Operational Framework for Canadian Naval Platforms

By

Scott Cameron Daniel Koshman

A thesis submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Industrial and Systems Engineering

written under the direction of

Professor David Coit

New Brunswick, New Jersey

May 2011

ABSTRACT OF THE THESIS

Flexible Modeling and Simulating Mission Availability Within The

Operational Framework for Canadian Naval Platforms

by Scott Cameron Daniel Koshman

THESIS Director:
Professor David Coit

Availability and reliability metrics have become key in-service performance measures in Canadian defence contracting. Previous implementations have evolved due to challenges in application, and were focused on the Air Force operational environment. With ongoing capital procurement and in-service support contracting, the Navy requires a definition and method of assessing availability appropriate to Naval platforms.

Naval ships are multi-role multi-function platforms. Traditional single function availability metrics are ambiguous for multiple functions / capabilities. Critical systems (e.g. propulsion, power) have an obvious effect on availability, while the loss of other functions (e.g. radar) do not. Non-critical system and capability impact is a function of the requirements of the current mission, thus mission availability must be evaluated.

Mission availability for a multi-function platform was defined as the interval average evaluation of critical system availability, mean capability availability, and mean weighted performance availability. The latter linked engineering performance to expected operational performance. Mission Capability Configuration Reliability Model was introduced to link system performance to capability performance. Using this model, an availability simulation, incorporating failure, maintenance, and logistical models was developed to assess mission availability. The simulation was applied to the project management functions of ship design and specification prototyping, availability assessment for contract management, and in-service performance prediction.

Acknowledgments

I want to express my thanks and gratitude to the following who have helped me and contributed to my thesis experience:

Professor David Coit for guiding me and keeping my perspective less complicated.

Professor Elsayed for providing challenges that helped me learn how to research.

Professor James Luxhoj and Professor Hoang Pham for ideas and concepts from their classes that were either used directly in my Thesis, or to better understand concepts.

Professor Mohsen Jafari for his time guiding me through my first formal simulation.

Professor Mani Lakshminaryanan for generously providing SAS code and for entertaining my many statistics questions during class breaks.

Ayberk Gokseven for teaching me Arena and responding to my questions.

Shaun Wolski, Livonia Mitchell, Isaac Goldberg, Ping Deng, and Yannick Lobe for their direct support.

Stéphane Ricard and Bruce Day for providing source material.

Department of National Defence (Canada), Director General Materiel Equipment Program Management for sponsoring my education.

Lastly, my sincere thanks to all of those who supported me in any respect during the completion of the project.

Table of Contents

Table of Tables

Table of Figures

Glossary

ADM(MAT) - Assistant Deputy Minister (Materiel)

AMT - Active Maintenance Time

CF - Canadian Forces

CM - Corrective Maintenance

Conceptual Model - For the simulation, any model that captures some idea to be later implemented in the simulation model if required

CSH - Canadian Search and Rescue Helicopter

Default Scenario - The main simulation used for experiments

DoD - Department of Defense (USA)

DND - Department of National Defence (Canada)

IM - Imperfect Maintenance

In-service - Refers the active operational life of platforms and systems

ISS - In Service Support

ISSC - In Service Support Contract

ISSCF - In Service Support Contracting Framework

GAO- Government Accountability Office (USA)

RBD - Reliability Block Diagram

M - Maintenance downtime

MCC RM - Mission-Capability-Configuration Reliability Model

MCF - Mean Cumulative Fucntion

MDT - Mean Delay Time, or Mean Downtime, or Maintenance Delay Time

MHP - Maritime Helicopter Project

MMI - Man Machine Interface

MTBCF - Mean Time Between Critical Failures

MTBF - Mean Time Between Failure

MTTF - Mean Time To Failure

MTTR - Mean Time To Repair

PBA - Performance Based Accountability

PBC - Performance Based Contracting

PBL - Performance Based Logistics

Platform - In the context of this Thesis, generally a naval ship

PM - Preventative Maintenance

PWGSC - Public Works and Government Services Canada

SAM - Ship Availability Model

Simulation Model - Refers to both the actual simulation construct and sometimes the models used directly in the simulation construction

SoR - Statement of Requirements

USAF - US Air Force

1 Introduction

This core of the research in this Thesis is the assessment of mission availability for a warship in the Canadian Navy context. This Thesis details the motivation, background, literature review, methodology, research, simulation construction, experimentation, and application of results. The research is expected to be applicable to Canadian procurement programs, where availability has been identified as a key performance factor.

The overall goals of this Thesis were to:

- Study the Operational Naval Environment to determine a practical model of Mission
- Develop applicable interpretations of Mission Availability
- Develop a methodology of applying Mission Availability to Project Management functions of determining acquisition requirements and in-service performance
- Demonstrate Mission Availability by simulation of a Halifax Class Frigate

1.1 Overview

Canada has a Navy of about 30 warships. Each of these warships is designed to perform a number of roles concurrently. These roles allow them to be employed in many different missions in the interest of Canada. To succeed at these roles, each ship has a variety of functions for which systems have been provided. These systems work together, dependent on their collaboration to maximize a ship's performance and survivability. In turn, the warships collaborate together in Task Groups to maximize the success of operations.

This Thesis focused on a single platform/ship, using the Canadian Patrol Frigate[1] (CPF) as a working example. The CPF has over 65 systems installed on board (OT DIV 2003). These systems support over 20 standard core capabilities, from which system requirements are derived. These capabilities are required in different combinations for different missions[2]. Warships often have different configurations to support a capability[3]. System configurations with redundancy on a warship provides improved reliability, but are also designed to provide improved effectiveness and survivability against external threats. This Thesis studied the engineering reliability aspects of ship's system configurations, and its linkage to operations.

Canada has recently updated its In-Service Support Contracting Framework (ISSCF), incorporating a Performance Based Contract (PBC) approach. In any contract, ADM(mat) (2009) states there is a Statement of Requirements (SOR) that should "specify the following: for the Acquisition Program, fleet size and mission/capability requirements; for the ISS Program, the required equipment availability and yearly activity rates". Defining availability in a naval context is the subject of this Thesis.

1.2 Motivation

Canada has the longest coastline (243,042 km) in the world[4] including more than 52,000 islands. Yet Canada has only the 30th largest Navy to defend its maritime assets (Globalfirepower.com 2010). The Canadian Navy is also actively deployed globally with the United Nations, North Atlantic Treaty Organization, and humanitarian efforts.

[1] Also known as the HALIFAX Class Frigate. The class is undergoing modernization, and the old
[2] For example, not all missions require the capability Harbour Defence.
[3] For example, different radars on a ship can be used for safe navigation.
[4] The next longest coastline, is of Indonesia, and it is only a quarter the size of Canada's!

Within practical cost constraints, Canada needs to maximize availability of its platforms in order to maintain its Naval operational tempo[5].

Canada has three major naval capital projects currently underway, the HALIFAX Class Modernization/Frigate Life Extension (HCM/FELEX), Joint Support Ships, and the Arctic Patrol Ships. These programs combined are worth billions of dollars (http://www.forces.gc.ca/site/pri/2/invest-eng.asp n.d.) encompassing both the acquisition of new platforms/systems, and the upgrade to existing platforms/systems (including the aforementioned In-Service Support (ISS) Contracts.

Canada's Departments of National Defence, Industry Canada, and Public Works and Government Services Canada (PWGSC), as well as the Navy are simultaneously involved in Naval contracts. Industry Canada oversees regional benefits from each contract, PWGSC is the contracting authority dealing with contract legal issues, and DND provides the project office. The Navy defines the requirements and provides Naval engineers to DND to work in the project offices. As the project managers and engineers working on large capital acquisitions, the Navy requires intimate knowledge of reliability engineering for ships.

2 Phase I - Background

2.1 Procurement : A Requirement For Availability

The importance and centrality of availability in DND equipment contracting has been evolving over the last few decades. From the background provided (ADM(Mat)

[5] Available personnel may be the limiting factor, but is not the subject of this Thesis

2009) and summarized here, availability[6] has become a focal point of both acquisition and in-service support contracting.

2.1.1 Traditional Approach

DND has traditionally contracted for in-service support activities (repairable item support, spares support, engineering services) separate from the acquisition contracting. DND would provide shipping, spares, technical data, special test equipment, and engineering/configuration management support.

The problems identified (ADM(Mat) 2009) with this approach are:

a. tending small contract fragments to the defence industrial base;

b. this approach including 'prescriptive' Statement of Work (SOW), and a time-and-material basis of payment providing no direct motivation for Industry to improve performance;

c. reduced accountability. A key concern; since this approach is not end-to-end, and many organizations are involved, it is difficult to identify causes of failed delivery of products/services and poor performance;

d. headquarter staff reductions in DND makes it 'untenable' to maintain this resource intensive approach; and

e. **operationally ineffective resulting in *poor equipment availability*.**

The pre-modernization CPF uses an equipment life cycling approach akin to this 'traditional fragmented approach'. Some individual systems (or sometimes groups if from

[6] and eventually unavailability.

the same OEM) have been contracted for In-Service Support contracts. No single contractor has responsibility for the whole suite of systems on a ship, and certainly not the overall availability of the platform.

2.1.2 Updated approach 2002 (Optimized Weapons System Management, and System Support Contracting)

The Canadian Air Force tried an updated approach, the Optimized Weapons System Management (OWSM). Under this approach a single contractor shares with DND the Total System Performance Requirement support responsibility. DND holds responsibility for mostly management functions and interact with the contractor in an Integrated Product Team approach.

Developed for the Canadian Air Force, this strategy is limited to their operational environment. There are differences between the Navy, Army and Air Force that affect using civilians in 'integrated' teams. With the Navy, for example, there is no room on ship for long term civilians, and their presence increases safety risks in emergency situations.

Despite efforts to implement this approach, the anticipated benefits were not realized as envisioned. The number of support contracts was reduced only to 3-5, and referred to as the System Support Contracting approach under the OWSM.

Some limitations to the adoption of this new approach (within the Air Force) were:

a. limited contract scope - It was not trivial to transform the existing system support networks, and thus, the contract scope could not be as inclusive as desired; and

b. culture - DND retained certain activities resulting in contracts that, as before, were based on time and materials, transferred little risk, and provided low accountability to the contractors resulting in the same problems identified in the traditional approach. Companies avoided accepting additional risk, or incentive programs that could jeopardize profit.

Only with a new platform[7] acquisition could an ISS contract with a single contractor supporting the whole platform be realized. The Canadian Search and Rescue Helicopter (CSH) project team tried this approach.

2.1.3 Canadian Search and Rescue Helicopter

In the CSH project, a performance-based contracting approach was adopted. All levels of maintenance and support were contracted to Industry[8]. Only two performance measures were contracted - ***aircraft availability*** **and** ***mission reliability***. These two measures were chosen because all maintenance was contracted out. A penalty was written in the contract for below standard aircraft availability, with the intent of motivating the "contractor behaviour" and "assure the required level of performance" (ADM(Mat) 2009).

[7] In this case, platform refers to aircraft.
[8] Previously noted, in the Navy, maintenance during ship operations cannot be contracted out.

The following shortcomings have occurred:

a. performance is below expectations;

b. the ISS contract was competed separate from the acquisition. Thus, in-service performance shortfalls could come from any of platform supplier, ISS contractor, and DND (*meaning low accountability*); and

c. insufficient initial provisioning (based on immature in-service performance data), and late delivery of spares affected initial performance of the ISS contract, but obviously without contractor accountability.

The next project to try this approach was the Maritime Helicopter Project (MHP).

2.1.4 Maritime Helicopter Project

The MHP contracted both the aircraft acquisition and ISS contract concurrently to the platform supplier. By contracting both at the same time, the Government of Canada avoided a sole source situation (which would be more costly than a competitive bid).

The primary performance measure is "***aircraft non-availability***"(ADM(Mat) 2009). Penalties are applied for contractor attributed *non-availability* beyond a threshold. The concept of non-availability is expanded upon greatly at a Performance Based Accountability (PBA) workshop (ADM(Mat) 2008)[9]. In the case of the MHP, since the contractors have no control over field level support by CF technicians, the project requires a complex tracking/fault attribution system.

[9] The details of 'non-availability' in this article are specific to the CF Air Force contracting environment. While the concept can be applied to the Navy, it requires modification prior to application.

2.1.5 US DoD Experience

The background (ADM(Mat) 2009) also covers lessons learned from the US. The US has, by far, the most experience with Performance Based Contracting (PBC) in the world.

A PBC sustainment contract for USAF aircraft fleet exists that is similar to the MHP project. Field support is similarly the responsibility of the USAF, thus creating a similar situation for measuring accountability. There are a couple differences (ADM(Mat) 2009):

a. the USAF measures availability while the CF Air Force measures non-availability; and
b. the USAF PBC contract is designed to force improvement by gradually increasing the contracted fleet availability requirement. From basic reliability theory there are fundamental problems with this requirement [10].

The USAF has implemented a 'robust' tracking and downtime attribution system, and unsurprisingly has had contract issues when the contract could not be met. Some of the contractors arguments were/are :

a. USAF aircraft are used in a wartime role, and experience greater overall stress than 'normal';

[10] If a bathtub curve describes the failure rate of an item, then after initial burn-in at some point the failure rate is an IFR. The only way to possibly change this is to 'refurbish' the item (make as like new as possible) or mid-life cycle re-engineering of the system; the latter is far from trivial and needs to be approached with correct engineering discipline. For availability, some improvement may be realized with experience due to improved maintenance times. Downtimes might also be improved upon if administrative processes can be improved. However, there is a limit to these improvements, and these can be counteracted by the eventual increased failure rates. Thus, eventually, this type of requirement may ask for something that is not physically possible (i.e. contract dispute).

b. military maintainers are at fault, in part, for aircraft availability shortfalls; and

c. the contractor is questioning everything they do not control.

It should not be surprising that a contractor would take legal action when their profits are threatened. These problems fall under legal game theory. This Thesis does not, however, delve into contract law and contract enforcement issues.

The result is that the USAF is looking at future contracts to not focus on aircraft availability but emphasize contractor controlled activities (ADM(MAT) 2009). The MHP project appears to be doing the same by focusing on non-availability versus availability (ADM(Mat) 2008).

2.1.6 US Government Accountability Office (GAO) Assessment

The GAO compared Industry practices for similar platform support situations. Two observations were noted (CF 2009):

a. Industry relies more on non-PBC contracting vehicles. This includes time-and-material contracts (especially for new systems). Lack of accurate/reliable system performance increases risk on new systems; and

b. Industry emphasized having rights to technical data, in case the service provider arrangements fail.

2.2 Definitions of Availability

There are many definitions of availability, useful for analyzing different situations. The definitions studied for this Thesis are given in this Section.

2.2.1 Instantaneous Point Availability

From Arusand and Hoyland (2004)we have the following common definition of Availability:

Let $X(t)$ be the state of an item at time t

$$X(t) = \begin{cases} 1 & \text{if the item is functioning at time } t \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

Let $A(t)$ represent point availability and $\overline{A}(t)$ represent point unavailability at time t

$$A(t) = \Pr(X(t) = 1) \tag{2.2}$$

$$\overline{A}(t) = \Pr(X(t) = 0) \tag{2.3}$$

Average Availability A_{av} on the time interval $(0,t)$ or (t_1,t_2)

$$A_{av}(\tau) = \frac{1}{\tau}\int_0^{\tau} A(t)dt \tag{2.4}$$

$$A_{av}(t_1,t_2) = \frac{1}{t_2 - t_1}\int_{t_1}^{t_2} A(t)dt \tag{2.5}$$

Equation (2.5) is also referred to as *mission availability* (Arusand and Hoyland 2004). Equation (2.4) can be used to calculate the expected 'lifetime availability' of a component for a planned lifespan τ.

Steady State Availability A:

$$A = \lim_{t \to \infty} A(t) \tag{2.6}$$

Equation (2.6) applies to the situation when "uptimes and downtime are non-lattice"[11] (Arusand and Hoyland 2004).

2.2.2 Availability with Perfect Repair

A perfect repair fixes an item to "as-good-as-new"[12]. A time line analysis shows we have periods when a system is working, and a system is down. Arusand and Hoyland (2004) derive inherent availability as follows:

Let T_i be *i.i.d.* random variables representing the sequence of uptimes

Let D_i be *i.i.d.* random variables representing the sequence of downtimes

The Law of Large numbers gives us:

$$\begin{aligned} &\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}T_i = E(T) = \text{MTTF} \\ &\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}D_i = E(D) = \text{MDT} \end{aligned} \tag{2.7}$$

where MTTF is the Mean Time To Failure and MDT is the Mean Downtime.

The steady state availability is:

[11] non-periodic.

[12] or as good as 'burned-in' depending on how we are applying assumptions and models.

$$\lim_{\tau\to\infty} A_{av}(\tau) = \lim_{n\to\infty} \frac{\sum_{i=1}^{n} T_i}{\sum_{i=1}^{n} T_i + \sum_{i=1}^{n} D_i}$$

$$= \lim_{n\to\infty} \frac{\frac{1}{n}\sum_{i=1}^{n} T_i}{\frac{1}{n}\sum_{i=1}^{n} T_i + \frac{1}{n}\sum_{i=1}^{n} D_i} \tag{2.8}$$

$$= \frac{E(T)}{E(T) + E(D)}$$

$$\lim_{\tau\to\infty} A_{av}(\tau) = \frac{\text{MTTF}}{\text{MTTF} + \text{MDT}} \tag{2.9}$$

Some useful approximations are given by Arusand and Hoyland (2004). For an item with independent failure times, constant failure rate λ, and MTTF >> MDT (for equation (2.10)):

$$\overline{A_{av}} = \frac{\text{MDT}}{\text{MTTF} + \text{MDT}} = \frac{\lambda \cdot \text{MDT}}{1 + \lambda \cdot \text{MDT}} \approx \lambda \cdot \text{MDT} \tag{2.10}$$

$$\text{mean number of repairs in period } t = \frac{t}{\text{MTTF} + \text{MDT}} \tag{2.11}$$

If downtime is only from corrective repair, then MDT=Mean Time to Repair (MTTR) and equation (2.9) becomes:

$$\lim_{\tau\to\infty} A_{av}(\tau) = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}} \tag{2.12}$$

Equation (2.12) is also called the *Inherent Availability* (Elsayed 2011).

2.2.3 Achieved Availability

Achieved Availability A_a can be used in measuring availability from both corrective and non-corrective maintenance. From Elsayed (2011):

$$A_a = \frac{\text{MTBM}}{\text{MTBM} + M} \tag{2.13}$$

where MTBM is mean time between maintenance, and M is the mean maintenance downtime.

2.2.4 Operational Availability

A general definition (in the form of operational unavailability) is given by Arusand and Hoyland (2004) as:

$$\overline{A_{op}} = \frac{\text{Mean total planned downtime} + \text{Mean total unplanned downtime}}{\text{Mission Period}} \tag{2.14}$$

Another definition is given by Elsayed (2011) as:

$$A_o = \frac{\text{MTBM} + \mathit{Ready\ time}}{(\text{MTBM} + \mathit{Ready\ time}) + \text{MDT}} \tag{2.15}$$

MTBM is mean time between maintenance. MDT includes maintenance downtime, delay time from logistics, and other indirect delays. *Ready time* = operational cycle - (MTBM+MDT).

2.2.5 Other Mission Availabilities

Birolini (1985) gives four definitions of availability, specifically for the purpose of assessing availability during a mission (and not to the exclusion of other availabilities).

Two of these, *mission availability* and *work mission availability*, are also discussed by Elsayed (2011).

$$A_{av}(t) = \frac{1}{t} E\{\text{up time in } (0,t)\} \tag{2.16}$$

Equation (2.16) is equivalent to Equation (2.4)

$$A_{Joint}(t, t+\theta) = \Pr(\text{item up at } t \cap \text{item up at } t+\theta) \tag{2.17}$$

$$A_m(T_o, t_f) = \Pr(\forall \text{repair times for failures during a mission of total operating time } T_o \le t_f)$$

(2.18)

$$A_{wm}(T_o, t_d) = \Pr(\sum \text{failure repair times in a mission of total operating time } T_o \le t_d)$$

(2.19)

Equation (2.17) is the probability that an item is operating successfully at the start and end of a mission. Equation (2.18) is the probability that a item failure during a mission is repairable within a specified time. Equation (2.19) is the probability that an item with multiple failures during a mission, will not exceed a specified total repair time.

2.2.6 Non-availability

Identified in ADM(Mat) (2009) is the term non-availability, and is discussed as the more common term unavailability in great detail (ADM(Mat) 2008). Unavailable time is based on Operational Unavailability A_u and is broken into two components, Active Maintenance Time (AMT) and Maintenance Delay Time (MDT).

AMT includes:

- Scheduled/Unscheduled Maintenance
- Modification Programs
- Changes to Configuration[13]

MDT includes:

- Lack of qualified personnel
- Unavailability of spare parts
- Unavailability of tools and test equipment
- Lack of proper facilities
- Unsafe conditions
- Management Delay
- Maintenance Information Systems unavailable

Following a comparison of US DoD PBL and DND A_o PBA approaches, (ADM(Mat) 2008), where DND replaces over a dozen DoD PBL metrics with just unavailability A_u, and A_u is defined as:

$$\begin{aligned} A_u &= A_{u,OEM} + A_{u,DND} \\ &= (A_{u,AMT-OEM} + A_{u,MDT-OEM}) + (A_{u,AMT-DND} + A_{u,MDT-DND}) \end{aligned} \tag{2.20}$$

2.2.7 Availability with Degraded Mission Capability

Any platform (especially a ship) can be in a degraded state of capability. In ADM(Mat) (2008) these are states are referred to as:

[13] Mission changes for aircraft preparing for their next flight.

- FMC - Full Mission Capable
- PMC - Partial Mission Capable
- NMC - Not Mission Capable

From ADM(Mat) (2008) the following relationships apply:

$$\begin{aligned} A_o &= A_{FMC} + A_{PMC} \\ &= 1 - A_{NMC} \end{aligned} \tag{2.21}$$

$$\begin{aligned} A_{FMC} &= \frac{\text{FMC}\,Time}{Total\,Time} \\ &= \frac{Total\,Time - (\text{PMC}\,Time + \text{NMC}\,Time)}{Total\,Time} \\ &= 1 - \left(\frac{\text{PMC}\,Time + \text{NMC}\,Time}{Total\,Time}\right) \\ &= 1 - A_{NFMC} \end{aligned} \tag{2.22}$$

where NFMC is Not Fully Mission Capable.

In this Thesis, the concept of partially capable availability was a key performance metric. There is additional mathematics given at (ADM(Mat) 2008) for estimating Availability from Fully Mission Capable Availability. However, several assumptions were made in the reference's model that won't be applied in this Thesis.

2.2.8 Summary

Table 1 summarizes the availability definitions from the preceding sections.

Table 1- Summary of Availability Definitions

Availability	Formula
Point Availability	$A(t)=\Pr(X(t)=1),\ \overline{A}(t)=\Pr(X(t)=0)$
Average Availability	$A_{av}(\tau)=\frac{1}{\tau}\int_0^{\tau}A(t)dt=\frac{1}{t}E\{\text{up time in }(0,t)\}$
Mission or Interval Availability	$A_{av}(t_1,t_2)=\frac{1}{t_2-t_1}\int_{t_1}^{t_2}A(t)dt$
Asymptotic Average Availability	$\lim_{\tau\to\infty}A_{av}(\tau)=\frac{\text{MTTF}}{\text{MTTF}+\text{MDT}}$
Inherent Availability	$\lim_{\tau\to\infty}A_{av}(\tau)=\frac{\text{MTTF}}{\text{MTTF}+\text{MTTR}}$
Achieved Availability	$A_a=\frac{\text{MTBM}}{\text{MTBM}+M}$
Operational Unavailability	$\overline{A_{op}}=\frac{\text{Mean total planned downtime}+\text{Mean total unplanned downtime}}{\text{Mission Period}}$
Operational Availability	$A_o=\frac{\text{MTBM}+\mathit{Ready\ time}}{(\text{MTBM}+\mathit{Ready\ time})+\text{MDT}}$
Joint Availability	$A_{Joint}(t,t+\theta)=\Pr(\text{item up at }t\cap\text{item up at }t+\theta)$
Mission Availability	$A_m(T_o,t_f)=\Pr(\forall\,\mathit{repair\ times\ for\ failures\ during\ a\ mission\ of\ total\ operating\ time}\ T_o\le t_f)$
Work Mission Availability	$A_{wm}(T_o,t_d)=\Pr(\sum\text{failure repair times in a mission of total operating time }T_o\le t_d)$
Operational Unavailability (CF Air Force)	$A_u=A_{u,OEM}+A_{u,DND}$ $=(A_{u,AMT-OEM}+A_{u,MDT-OEM})+(A_{u,AMT-DND}+A_{u,MDT-DND})$
Operational Availability (multi mission capable state)	$A_o=A_{FMC}+A_{PMC}$ $=1-A_{NMC}$
Fully Mission Capable Availability	$A_{FMC}=1-\left(\frac{\mathit{PMC\ Time}+\mathit{NMC\ Time}}{\mathit{Total\ Time}}\right)$

2.3 Supporting Missions - CF Readiness and Sustainment Program

Policy Document CFCD 129 (2009) describes CF policy Readiness & Sustainment (R&S). This policy covers force generation, force employment and support. It relates capability with equipment in the context of mission requirements.

The general operational cycle of a Canadian warship consists of 1) a Docking Work Period (DWP); 2) Tiered Readiness Program (TRP); 3) a period of Standard Readiness (SR); 4) a period of High Readiness (HR) with expected operational deployment; 5) another period of Standard Readiness; and 6) a period of Extended Readiness (ER) prior to starting the cycle again. This is shown in Figure 2-1.

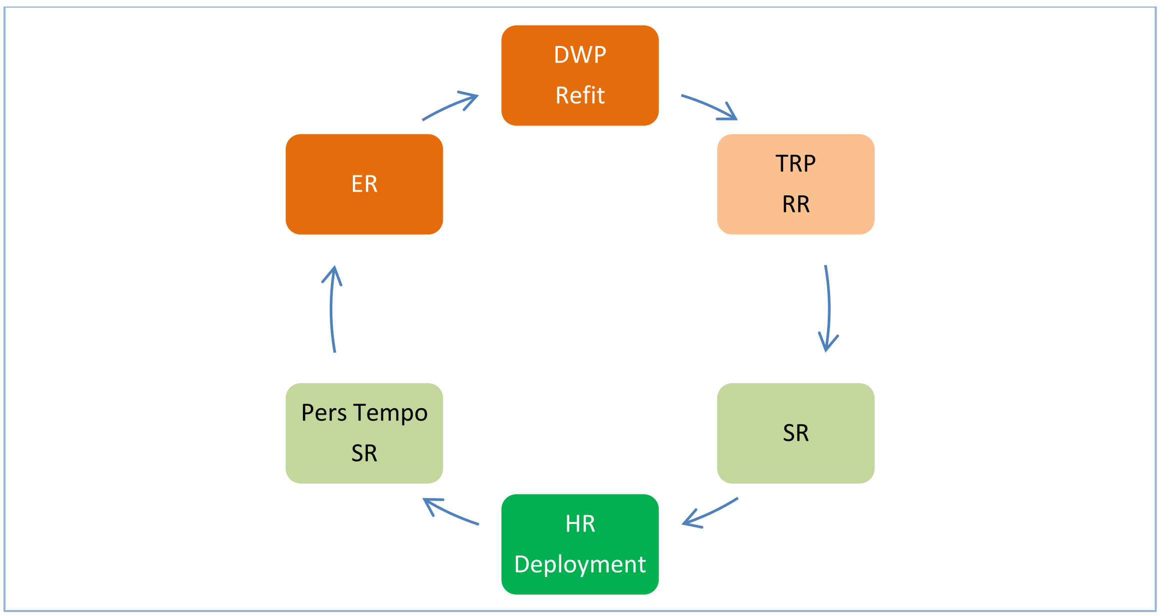

Figure 2-1 - General Ship Operational Cycle

Ships are typically used on missions when in SR. When in HR, ships can be expected to be deployed. Readiness levels dictate the personnel and training levels. Materiel (equipment) state is "determined by the type of mission and resource availability"(CF, CFCD 129 2009). High Readiness ships by default are expected to have complete capability inherent in the ship design, plus mission modifications. This Thesis while not specific to an exact readiness state, considered capabilities using Standard Readiness as a guideline. Core capabilities are identified, in the reference, for Standard Readiness as well as considerations for the ship's intended employment (coastal, continental, international).

3 Phase I - Literature Review

3.1 Mission Reliability of a Combat Tank 1989

Kim (1989) defines mission reliability as "the probability that a system will perform its specified mission". The author notes that, for a given platform: 1) not all components are required for a given mission, and 2) the failures of configurations of components in real systems can overlap. The author presents a combat tank as a working example to demonstrate mathematics of mission reliability.

The author defines two 'missions' moving and firing. Let F be the event where the components for firing are working. Let M be the event where the components for moving are working. Let QM be the event where the combat tank is required to move. Noting that the tank requires a different configuration for firing when it is moving versus when it is not moving, the following relationship is given:

$$\Pr(F) = \Pr(QM)\Pr(F \mid QM) + \Pr(\overline{QM})\Pr(F \mid \overline{QM}) \tag{3.1}$$

Treating $\Pr(QM) = \alpha$ and $\Pr(\overline{QM}) = 1 - \alpha$, the article states that choosing α=1[14] sufficient in practice, since the math predicts a lower bound. The article then breaks-down the 22 major components involved, identifies which are required for each capability and decomposed capability, considers the components as being independent and in series, calculates Pr(F) for calculates the lower and upper bounds.

The concept of a different action requiring a different subset of components is important. For example, assume we are discussing two radars on a ship as if they are capabilities. In order for each radar to function, it requires power from the ship's power

---

[14] ignoring the stated situation where a different configuration is used when not moving.

generation and distribution network. Yet, the physical connections from the power generation and distribution network are different. Many of the main components are the same (diesel generators, switchboards) but there is a variance in the configuration. This is shown in Figure 3-1 where each radar requires its own link. This is far less naive then excluding the link, since if the link fails, the other capability is still available.

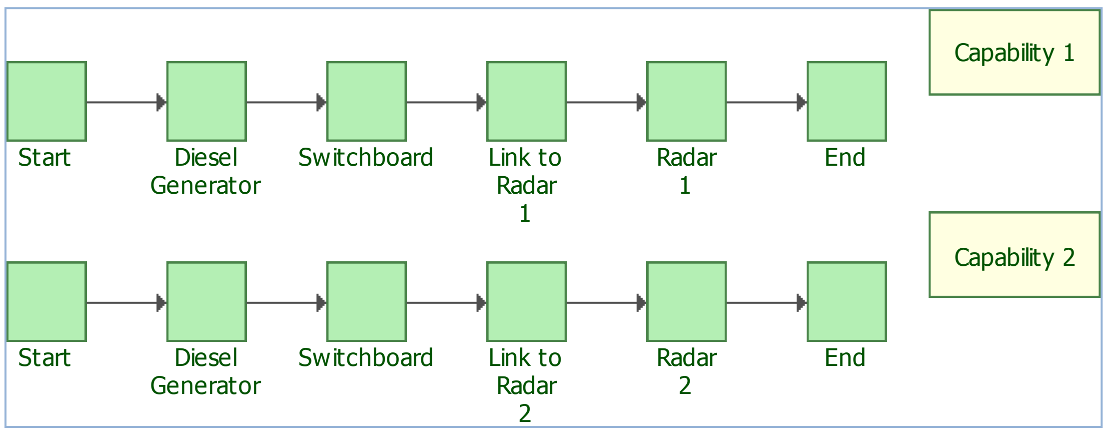

Figure 3-1 - Dependence of System Configuration on Context

Returning to the Combat Tank example (Kim 1989), the model resembles what is proposed in this Thesis. However, this Thesis does not treat mission reliability as described. This formulation confuses an availability concept of phasing the mission. In a actual mission, the tank would be required to move, move and fire, stop, and stop and fire at pseudo-random times[15]. As soon as the tank has fired, realistically, it is time to move again. The logic discrepancy is the assumption that $\overline{QM}$ exists. On a firing mission, the tank might not be moving, but it is still required to move. As soon as the moving components fail, the tank is in a state of failure. The interpretation $\overline{QM} = \varnothing$ simplifies equation (3.1) trivially.

[15] it is noted that the author did not consider a similar breakdown for the 'move mission', where the components needed are understandably different when the tank does not need to move.

This Thesis, using the Mission-Capability-Configuration Reliability Model described later, defines the tank example differently. The mission could still be called a firing mission. It would consist of two capabilities (versus missions), moving and firing. The mission reliability would be the probability of the intersections of these two capabilities working. There would be a reliability model for each capability. Since the firing capability works slightly differently when moving versus not moving, two configurations (in parallel) would be included, and resolved when analyzed.

The point about how the author constructed the formula (3.1) is brought up as a guiding cautionary when constructing phased missions. For the purposes of this Thesis, it is most appropriate to consider each mission phase as a collection of capability requirements which can be met by some set of configurations of a system or in this case, the ship. Attempts to define phases as actions[16] during a mission are problematic, and this Thesis rejects that approach.

3.2 Ship Availability Modeling 1987

Ship availability modeling was explored by the Ministry of Defence (UK) in the 1980s. In White and Venton (1988) practical limitations of a Ship Availability Model (SAM) were discussed. This is the most direct coverage of this Thesis subject found, though further development does not appear in literature. The article references a previous paper from two years prior presented at *Proceedings Reliability '85*[17] that discussed actual models that they were trying to implement, however a source for this paper has not been found. It is not clear how far the work was continued. There are

[16] instead of capabilities or configurations

[17] This appears to be a conference hosted by Ministry of Defence (UK) with Industry. No official website, contacts, or proceedings have been found.

many points within this article based on their experience working on SAM within the UK naval environment.

The focus of the article was on application of SAM for ship design and procurement. The limitations discussed focus on the challenges of the project managers trying to implement SAM. The authors, though, do not believe SAM is actually required at the time of the article.

White and Venton (1988) describe an 'ideal package' of a modeling program. Their 'ideal' is a program that accommodates simple and quick models for the concept phase of procurement, and then gets more detailed as the ship's design gets more detailed. Two approaches to building the models are given.

In the first approach, all the systems are identified along with their relationships. System models are further developed, and specified ship values of availability and reliability apportioned among the systems. The authors argue that this approach is troublesome when the relationships are complex, which causes problems also in the early definition stage. This approach is referred to as "'Apportionment and System Modeling" (White and Venton 1988).

Their second approach consists of building a single model for the whole ship. Referred to as "Ship Modeling", it is claimed it can consider complex systems interactions, and ensures a consistent model among the systems. They acknowledge 'practical difficulties' without expanding on this.

Without specific examples of what they are talking about, it is not entirely clear what the actual distinction between the *Apportionment and System Modeling* approach and the *Ship Modeling* approach actually are. There is an assumption that defining system relationships in the first does not adequately consider the level of complexity that the second can. It appears the authors are suggesting , that in the first approach, there were people doing a 'basic' system design of the ship, assigning availability specifications, then doing the sub-modeling of the systems, and missing the interdependencies of the systems. The capability-configuration approach in this Proposal attempts to avoid this problem by not avoiding configuration overlaps.

The authors state that either choice can be used, depending on the system interaction complexity, but also the "interactions between systems and the sequence of ship activities, e.g., passage, attack submarine, shoot down aircraft, which make up a mission". This is a very revealing statement about the authors' focus of their work. They are concerned with how difficult it is to define sequences of events which can be used to calculate availability. In trying to define a sequence of events that includes responses to events (such as threats), they may have concerned themselves with a too detailed version of phased missions. Their example of 'attacking submarine' is an event that can occur during a mission. The submarine threat requires an anti-subsurface capability of the ship, but the periods surrounding this event do not automatically 'not require' this capability. Thus, the configuration required by the ship does not actually change outside of the event. By trying to model a mission as a sequence of a specific actions taken, one is not measuring mission availability of a capability, one is trying to estimate the measured availability or perceived availability in relation to the activities that required the systems. This Thesis Proposal approaches availability not as 'the system works when used', but as 'the system will work in required periods of time'. The article's assertion about 'extreme difficulty' to define system/system and system/sequence interactions is not immediately convincing without clarification of where they were finding extreme difficulty.

The article then goes on to describe that ship systems are supported by multiple systems and that these multiple systems support each other. The article makes this sound complex, when what is actually being described can be successfully modeled in a RBD by a simple series relationship (that is not to say that there are no other complexities involved). These interrelated relationships are key to considerations for the model for this Thesis.

The overall SAM program characteristics identified by White and Venton (1988) are :

a. **outputs must yield availability and reliability information at different levels[18];**
b. poor values at the output need to be traceable;
c. sensitivity of output values to input value changes need to be identified;
d. handle many small combinations in the concept phase;
e. handle single systems in single and multiphase;
f. handle multiple systems with interactions in single and multiphase;
g. allow for configuration changes in systems and system combinations;
h. changes in mission phase pattern; and
i. changes in the input data.

---

[18] This conclusion is recognized in this Thesis for the problem of simulation output analysis.

3.3 Other Articles

Phased-mission availability analysis appears quite often in literature (Ma and Trivedi 1999), (Wang and Trivedi 2007), (Andrews 2008), and (Remenyte-Prescott, Andrews and Chung 2010). These papers focus on algorithms to solve phased-mission availability. In an early article, Easry and Ziehms (1975) give an example of a fire department response to a fire. In the first phase, several different fire vehicles can suffice to clear the building. In the second phase, special equipment is required thus a different configuration of fire fighting gear and vehicles is needed. In the third phase, once again the configuration changes in order to overhaul the fire. One key observation from the article, is that later phases are not independent of the first. A system not needed in the first phase, but needed in the second, must be available in the first as well[19]. This was especially important for the transformation from phased mission into a single mission RBD.

In some articles, availability given different maintenance strategies is analyzed. References (Zhao 2003), (Tsai, Want and Tsai 2004), and (Galante and Passannanti 2009) consider preventive maintenance while (de Smidt-Destombes, van der Heijden and van Harten 2004) considers an aspect of condition based maintenance.

There are also papers, such as Ke, Huan and Lin (2008) which apply fuzzy logic in the analysis of redundant repairable systems. Fuzzy logic is interesting because it involves training fuzzy sets, not unlike trying to match distributions for component

[19] True for the fire response example. If a system is repairable with the mission parameters then other assumptions can be made. For a mission though, a system may be considered non-repairable if the requirement to repair would remove it from availability in later mission phases.

failures. If a certain type of fuzzy set is used (triangular set) then specific defined fuzzy operations on those sets yield more triangular sets (very fast math).

An overview of 'system effectiveness measures' specific for weapon system analysis are given by Sherif and Kheir (1982). These include "system performance, operational readiness, life cycle costing, design to cost, reliability, availability, maintainability, producibility, operability, capability, adequacy, and logistics".

Using a Re-locatable Over-the-Horizon Radar, Willingham and Forster (1990) discuss the problem of sparing, and the effect additional spares can have on availability. Their definition of availability was:

$$A_o = \frac{\text{MTBCF}}{\text{MTBCF} + \text{MTTR} + \text{MSRT}} \tag{3.2}$$

where MTBCF = mean time between critical failure, and MSRT = mean supply response time. This Thesis assess this definition as one of three availability metrics.

The problem of computing phased mission reliability given variable configurations is investigated by Somani, Ritcey, and Au (1992). In their model, they have a fault tolerant system with multiple redundant configurations, dedicated spares, pooled spares, and random phase durations.

Multifunctional systems have also been studied (Sols and Nachlas 1995), (Sols 1997), and (Sols, Ramirez-Marquez, et al. 2007). One of the common considerations in these papers is whether an item is ready to undertake a mission. This consideration is too restrictive for this Thesis. Additional models are given for Element Availability, Function Availability, Mission Availability, Degraded Availability, Degraded Mission

Availability, Degraded Function Availability. By using the MCC RM in developed in Section 5.2, this Thesis provides a minimum of four levels of availability analysis that encompass these definitions though interpretation and applicability may differ.

3.4 Nonparametric Estimation of Availability

This Thesis must assess availability from empirical data, especially simulation output. Non parametric methods are investigated in this section.

In this section, notation is X for MTTF or MTBF[20] and Y for MTTR.

3.4.1 Limiting Availability

The limiting Availability defined in equation (2.12) and represented here as:

$$\lim_{\tau \to \infty} A_{av}(\tau) = \frac{X}{X+Y} \tag{3.3}$$

Where data is complete, Baxter and Li (1996) provided estimators for the limiting availability by constructing a second order Taylor series expansion of equation (3.3), and noting the conditions where the second derivative terms converge to zero when multiplied by $\sqrt{n}$. Using the assumption of independence of X and Y, and the convergence of $\sqrt{n}(A_{av} - A) \sim N(0, \sigma^2_{A_{av}})$[21], the mean, variance, variance estimator, and confidence intervals were derived and are given in equations (3.4), (3.5), (3.6), and (3.7).

$$A_{av} = \frac{\overline{X}}{\overline{X}+\overline{Y}} \tag{3.4}$$

[20] depending on the model used.
[21] this is commonly referred to as the Delta Method.

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$
$$\overline{Y} = \frac{1}{m}\sum_{i=1}^{m} Y_i \qquad (3.5)$$

$$\sigma_X^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})^2$$
$$\sigma_Y^2 = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \overline{Y})^2$$
$$\eta = \lim_{\substack{n\to\infty \\ m\to\infty}} \frac{n}{m}, \quad \eta = \frac{n}{m} \qquad (3.6)$$
$$\sigma_{\overline{A_{av}}}^2 = \frac{\mu_Y^2}{(\mu_X + \mu_Y)^4}\sigma_X^2 + \eta\frac{\mu_X^2}{(\mu_X + \mu_Y)^4}\sigma_Y^2$$

$$\sigma_{\overline{A_{av}}}^2 = \frac{\overline{Y}^2}{(\overline{X} + \overline{Y})^4}\sigma_X^2 + \frac{n}{m}\frac{\overline{X}^2}{(\overline{X} + \overline{Y})^4}\sigma_Y^2$$

$$A_{av} - z_{\alpha/2}\frac{\overline{\sigma_{av}}}{\sqrt{n}} \le A \le A_{av} + z_{\alpha/2}\frac{\overline{\sigma_{av}}}{\sqrt{n}} \qquad (3.7)$$

This limiting availability is built upon failure and repair times. The *n*/*m* factor in equation (3.6) provides a way for adjusting for partial cycles but not for censored data. If we assume availability is independent on each renewal cycle, then multiple replication runs can be added together in a large set of observations. For components this is a safe assumption, but not for systems.

Baxter and Li (1996) also provides for the case of censored observations. Censored data occurs naturally in a simulation at the end of each replication. In Baxter and Li (1994) non-parametric confidence intervals were developed from large sample data for the renewal function and point availability. The assumptions in this paper are based on sequences of independent, identically distributed r.v. The assumption of independence in this Thesis cannot be assumed, or more precisely is assumed false. The

MCC RM models a set of components which entangle creating dependencies that spread across capabilities, and across renewals. There is independence between replications, and this is used to create confidence intervals based on Central Limit Theorem.

3.5 Maintenance Models

Maintenance includes both actions required to restore a system and prevent a system from failing. This section describes key concepts applicable to the Thesis's simulation.

3.5.1 Stochastic Repair Process

Following Manzini et al. (2010) repair process is notated as in Table 2.

Table 2 - Repair process general notation

Function	Representation	Expression
CDF	$G(t)$	$\int_0^t g(x)dx$
Pdf	$g(t)$	$\frac{dG(t)}{dt}$
Mean time to repair	MTTR	$\int_0^\infty xg(x)dx$
Repair 'hazard' function	$\mu(t)$	$\lim_{\Delta t \to 0} \frac{G(t+\Delta t)-G(t)}{(1-G(t))\Delta t} = \frac{1}{\overline{G(t)}}\frac{dG(t)}{dt}$
Initial Condition	$G(0)$	0
Asymptotic Condition	$G(\infty)$	1

While stochastic repair processes are common throughout literature, repair times on a ship may be either stochastic or deterministic. If the fault involves a faulty circuit card, typically this is replaced with a spare, and the time to complete this task is consistent. Other repairs, such as an ambiguous untraceable fault on a radar system, has a non-consistent repair time. Finally there are faults which are not repairable while on ship, and must wait until the ship is either alongside or in dock.

The simulator produced for this Thesis handles either deterministic (i.e. circuit card replacement) or stochastic repairs (complex fault finding and resolution). Repair times are not required to be exponentially distributed, as is too common in literature. The simulation assumes all repairs take place on ship[22].

3.5.2 Maintenance Concepts

Maintenance is not an automatic process. It involves decisions, delays, and proactive behaviour. Research in this area involves determining optimal policies under sets of ideal assumptions. Many of the concepts in modern maintenance theory are found in Manzini et al. (2010) and include:

a. Corrective Maintenance (CM). Action taken to bring a system or component from a non-operational state to an operational state.

b. Preventative Maintenance (PM). Actions "intended to reduce the probability of failure or degradation of the functioning of an item" (Manzini, et al. 2010). The probability of failure of an item can be reduced by restoring the item, replacing it, calibrating it, or simple housekeeping actions intended to keep new failure modes from occurring;

c. 'Good as New'. An item is restored to its initial state, either equivalently new, or as 'Good as Burned In';

[22] This would not be difficult to add, as being alongside can be modeled as a resource required for the repair to take place.

d. Replacement. Instead of fixing a component, a component is replaced, and the broken part returned to the supply system for either disposal or repair by the OEM. This is one way of trying to ensure a 'Good as New' repair[23];

e. 'As Bad As Old'. Condition after maintenance, when the item's condition is not improved though restored to an operational state.

f. Minimal Repair. Minimal repair is the intentional repairing of a system to the 'as-bad-as-old' state. Reasons may include inconvenience to conduct a full repair, intentional deferment policy, system can't be repaired , or other practical reasons;

g. Imperfect Maintenance (IM). Restoring a system to an unintended state between 'as-good-as-new' or 'as-bad-as-old' is possible and is called imperfect maintenance. IM can be applied to both corrective maintenance and preventative maintenance policies;

h. Degradation. After each renewal period, the part suffers accelerated failure. Modeled in different ways such as having the part switching states in each renewal until it becomes non-repairable. Also used in reference to physical monitoring where the observed deterioration of a part is used to set a replacement policy;

i. Opportunistic Maintenance. This policy is to take advantage of system downtime to do preventative replacements. One example of where this happens, if a system breaks down, then opportunity might be taken to maintain other components while restoring the broken component;

[23] Though possible installation errors could yield in a bathtub hazard function curve that ideally would not exist.

j. Maintenance Free Operating Periods (MFOP). Also Maintenance Free Operating Period Survivability (MFOPS). These reliability measures are the performance ratio of systems successfully being operational for a minimum period of time, for each renewal cycle;

k. Worse Maintenance. From Pham and Wang (1996), this is a maintenance action that increases the failure rate despite brining the system to an operational state again . Depending on the nature of the failure, and the possibilities of repair given circumstances, this might be the best possible option (an example would be setting up an emergency bypass for power when the main power feeds on a ship have been severed or damaged);

l. Worst Maintenance. From Pham and Wang (1996), this occurs when a system or item breaks immediately upon repair due to corrective action. This is somewhat ambiguous of a category since technically the system wasn't repaired.

Goals of maintenance policies can include minimizing costs, minimizing downtime, maximizing availability[24], offsetting down time to non-critical schedules, maximizing performance/production.

3.5.2.1 Corrective Maintenance Policies

Corrective maintenance includes immediate maintenance policies and deferred maintenance policies. Immediate polices assumes action is taken to restore a system immediately upon its malfunction. This is not always practical in real life, due to conflicts in resource management, or other distracting events[25]. Deferred policies involve

---

[24] not the same as minimizing downtime.
[25] Such as damage control and ship survivability.

waiting to a determined time before initiating restoration. The Thesis's simulation assumes immediate corrective maintenance actions, as long as spares and human resources are available.

3.5.2.2 Imperfect Maintenance (Human Error)

Imperfect maintenance is the concept that restoration of the component and/or system to working order does not necessarily restore it to the intended state[26]. An imperfect maintenance model has been implemented. Based on the 'p-q' imperfect maintenance model discussed in Pham and Wang (2000) and Manzini et al. (2010), with an adjustment to add in a human factor. In the 'p-q' model there is a probability assigned to a maintenance action whether it restores the item to 'as-good-as-new' (p), or 'as bad-as-old' (q), with p=1-q.

3.5.2.3 Preventative Maintenance Policy

Preventative maintenance policies are widely discussed in literature. An overview can be found in Manzini et al. (2010) Preventative maintenance policies can be categorized in many ways including time based, and use based (run time), based on statistics / reliability and based on condition. For this Thesis, a simpler implementation of PM was considered. PM was a non-replacement action that corresponds to maintenance that prevents new failure modes from affecting systems (such as cleaning dust off of electronics). This corresponds more accurately to maintenance on a ship then a replacement policy.

---

[26] generally new.

4 Methodology

This Thesis was conducted in five phases: background and literature review, research, simulation design and construction, experimentation, and application. Each phase was motivated by the central problem of ship's Mission Availability.

4.1 Phase I - Background and Literature Review

The first phase consisted of establishing the problem background and motivation, and reviewing literature on the key topics of mission availability, and platform/ship availability. These are detailed in Sections 2 and 3.

4.2 Phase II - Research

In phase II, different ideas and topics were researched. The main concept developed in this Phase was the MCC RM (Section 0). This model, created for exploring the relationship of systems to a multi-function platform in the context of availability and performance, is the centerpiece of this Thesis.

4.3 Phase III: Simulation Design and Construction

Using the research from Phase II, a simulation was developed to simulate ship mission availability through the MCC RM. Relevant aspects of the design, and the models incorporated are discussed in Section 6.

4.4 Phase IV: Experimentation

With a working simulation environment, experiments were performed to explore the functionality built into the simulation. The details are found in Section 7.

4.5 Phase V: Implementation and Application

The final phase was to apply the simulation to the motivating problems. Project management functions were discussed, and the application of the simulation was demonstrated.

5 Phase II - Research

This Thesis consists of an study and analysis of Mission Based Operational Availability and related metrics as applied to a Canadian Naval Ship. Preliminary research has included studying the Naval operational environment, the Naval maintenance environment, ship/system configurations, the problem of deployed logistics (Section 5.1), the development of a Mission-Capability-Configuration model and some of its implications (Section 5.2). The evolution of availability as a procurement measure (Section 2.1), the current Canadian contracting requirements, the status/challenges of using availability measures within current Navy capital projects, and relevant definitions of availability (Section 2.2, Section 5.4).

5.1 Operational Availability with multi-leveled sparing system

For an operational ship, there are multiple levels of sources of sparing. For the Navy, these levels can be represented by:

Source 1 - Ship

Source 2 - Task Group Replenishment Ship

Source 3 - Supply System / Ashore Maintenance Facilities

Source 4 - ISS contractor / OEM

These sources represent different methods of supplying a ship. The supply system may have a part in warehouse and ship it out, or it may procure the part local to the Ship. Each of these sources have different logistical overhead associated with it.

Assume each source i, of spares, has an associated supply of spares (n_i) and an average time to deliver supply to the ship (τ_i). Total spares $N = \sum_{i=1}^{k} n_i$ at a given time.

Assumptions for this analysis are:

- Supplies used without replacement
- Supplies do not regenerate
- time to deliver τ_i is constant
- Single repair point located at the ship (*Source 1*)
- Single spare requirement per repair, only one spare type

Factors specifically not accounted for are:

- Multiple platforms requiring spares
- Possibility spare will not repair system
- Possibility no spares are required
- Probability distributions of delivery times
- Scheduled downtime
- Potential failure of spare delivery

Let $Source_i$ represent a source of spares for repairs on a system. The system to be repaired is co-located with $Source_1$. The reliability of the system will be represented by MTBF. Once a spare is available, the time to repair is represented by MTTR.

Based on the common approximations of availability, this model is:

$$A_{o,spares\,multi\,source} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR} + \tau} \tag{5.1}$$

where MTBF is Mean Time Between Failure, MTTR is Mean Time To Repair, and τ is the mean time to deliver/receive a spare.

Let F be a random variable representing number of failures of the system. The system is represented by:

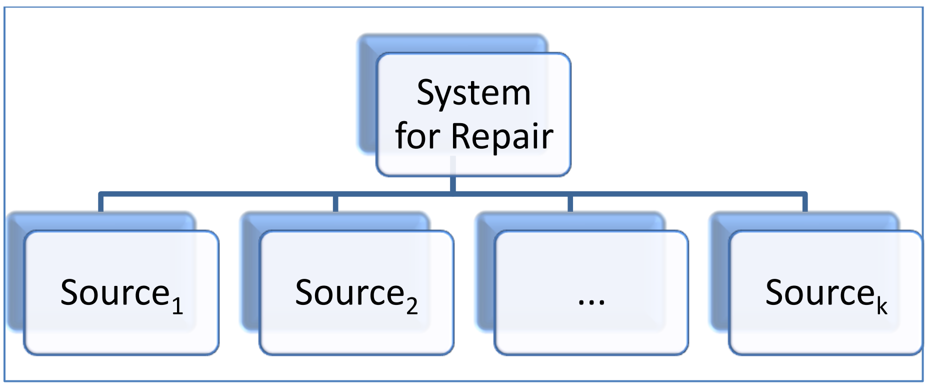

Figure 5-1 - Hierarchal Diagram for System Sparing

Assume that sources are in order such that $\tau_a \leq \tau_b$. Only one spare is sent for a given repair, the spare with quickest delivery time.

If $F < n_1$ the A_o is simply:

$$\begin{aligned} A_{o,spares\,multi\,source} &= \frac{F \cdot \text{MTBF}}{F \cdot \text{MTBF} + F(\text{MTTR} + \tau_1)} \\ &= \frac{\text{MTBF}}{\text{MTBF} + (\text{MTTR} + \tau_1)} \end{aligned} \tag{5.2}$$

If $F > n_1$ AND $F < n_2$ then:

$$A_{o,spares\,multi\,source} = \frac{F \cdot \text{MTBF}}{F \cdot \text{MTBF} + F \cdot \text{MTTR} + n_1\tau_1 + (F - n_1)\tau_2} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR} + \frac{1}{F}n_1\tau_1 + (1 - \frac{n_1}{F})\tau_2} \tag{5.3}$$

For $\sum_{i=1}^{a-1} n_i < F < \sum_{i=1}^{a} n_i$, a ϵ 2 ... k then:

$$A_{o,spares\,multi\,source}(a,F) = \frac{F \cdot \text{MTBF}}{F \cdot \text{MTBF} + F \cdot \text{MTTR} + \sum_{i=1}^{a-1} n_i\tau_i + (F - \sum_{i}^{a-1} n_i)\tau_a} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR} + \frac{1}{F}\sum_{i=1}^{a-1} n_i\tau_i + (1 - \frac{1}{F}\sum_{i}^{a-1} n_i)\tau_a} \tag{5.4}$$

Worst case availability (without running out of spares) is:

$$A_{o,spares\,multi\,source}(k,F) = \frac{F \cdot \text{MTBF}}{F \cdot \text{MTBF} + F \cdot \text{MTTR} + \sum_{i=1}^{k} n_i\tau_i} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR} + \frac{1}{F}\sum_{i=1}^{k} n_i\tau_i} \tag{5.5}$$

Average availability is:

$$E\left[A_{o,spares\,multi\,source}\right] = \sum_{f=1}^{N} A_{o,spares\,multi\,source}(a,f)P(F=f) + 0 \cdot P(F>N) \tag{5.6}$$

Note that for finite spares, A_o approaches zero as the operational period approaches infinity since the spares will eventually deplete.

5.2 Mission-Capability-Configuration Reliability Model (MCC RM)

The central model created for this Thesis, is a Mission-Capability-Configuration reliability model. This model consists of a defined mission, capabilities required for that mission, possible configurations that can be used to meet those capabilities, and links to reliability models for those configurations. A hierarchal representation of this model is shown in Figure 5-2.

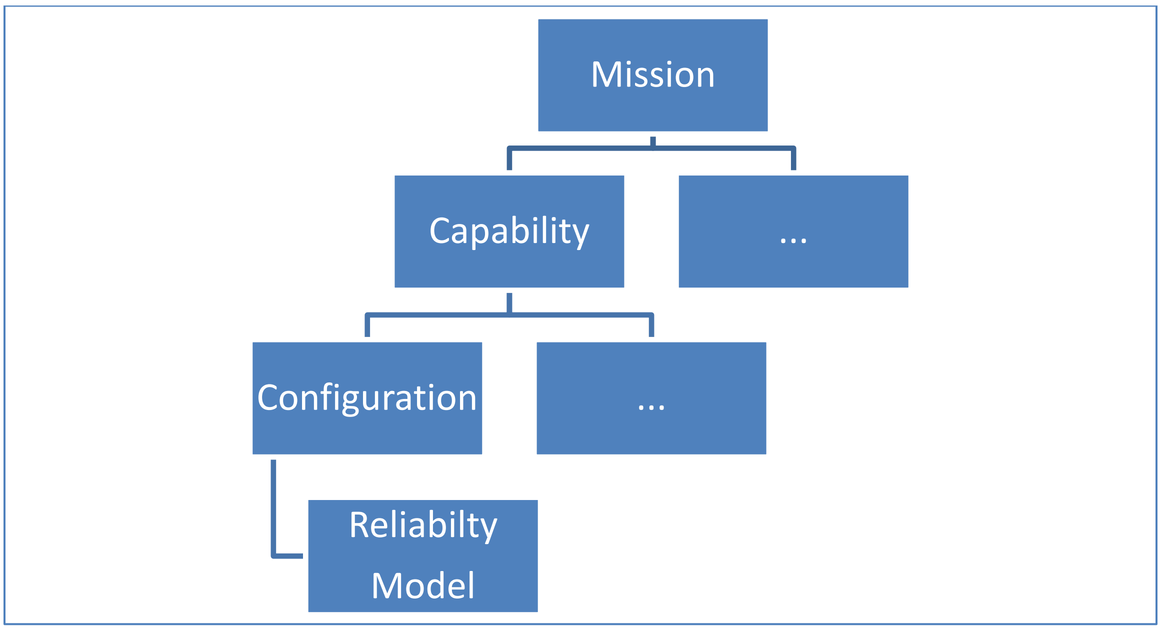

Figure 5-2 - Mission-Capability-Configuration Reliabiliy Model

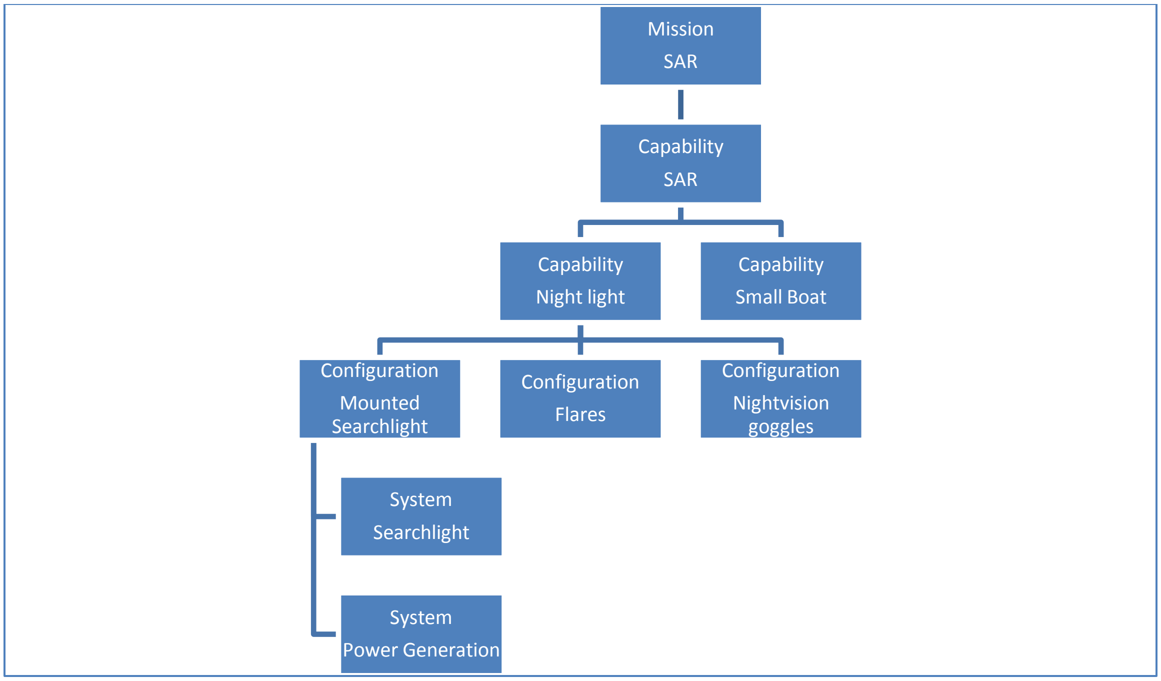

Figure 5-3 - Example of part of a Search and Rescue (SAR) Mission

Search and Rescue (Figure 5-3) is a capability defined for CPFs (CF, CFCD 129 2009). This capability has sub-capabilities. The MCC RM takes this into account by being able to substitute sub-capabilities for a capability. This sub-capability design was not implemented in this Thesis.

5.2.1 Definition of Mission

A mission is a time period of interest when a platform is required to perform various specific functions. It is possible to break a mission into several logical periods referred to as phased missions, or sub-missions. It is also possible to consider a mission at a larger level such as an operation. The logical mission scale for this Thesis was a single ship's deployment with access to a support ship.

This model must be able to adjust the mission definition as required. "The flexibility for Operational Commanders to determine mission requirements and assess the readiness of units to execute a given mission is paramount." (CF, CFCD 129 2009). The MCC is built upon mission requirements, and should provide the desired flexibility in determining requirements.

5.2.2 Phased Mission

Common in literature is the concept of a phased mission. The idea is that at different time intervals, a system may require a different subset of its functions and different configurations to meet objectives.

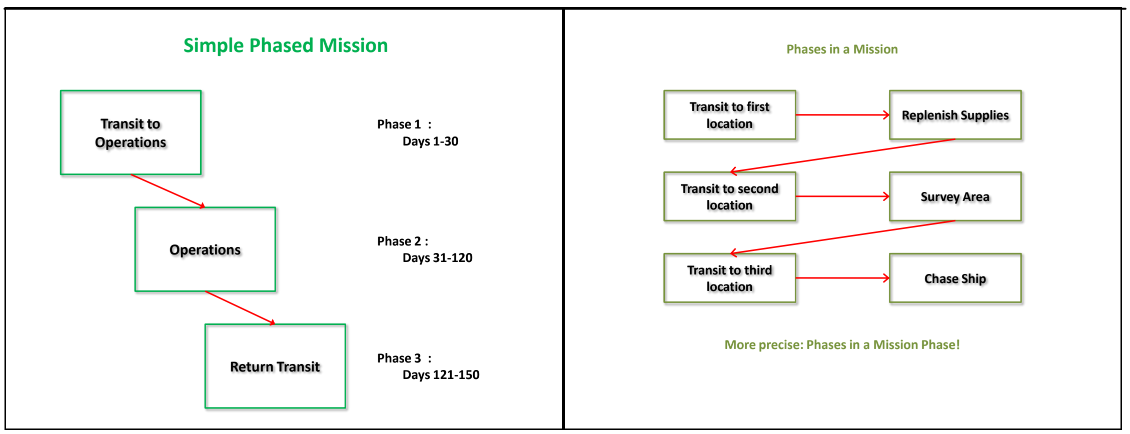

Figure 5-4 - (a) Simple Phased Mission (b) Problematic Phases in a Phased Mission

There needs to be care in defining these time intervals. Consider the first scenario shown in Figure 5-4 (a). In Phase 1 a platform needs to transit to an operational area. In Phase 2 it conducts its operations. Finally, in Phase 3 it transits home. These phases have distinctive requirements on the ship's capabilities. This is the approach for this Thesis.

Another approach is depicted in Figure 5-4 (b). The particular actions that a ship may conduct during operations are estimated and ordered (order and duration affects availability calculations for phased missions). While it may be argued that there exist systems/platforms that this approach works well for, when given a naval warship , the timing, duration, and order of these activities are subject to great 'flexibility'[27] / variance. This activity decomposition is not be used in this Thesis[28].

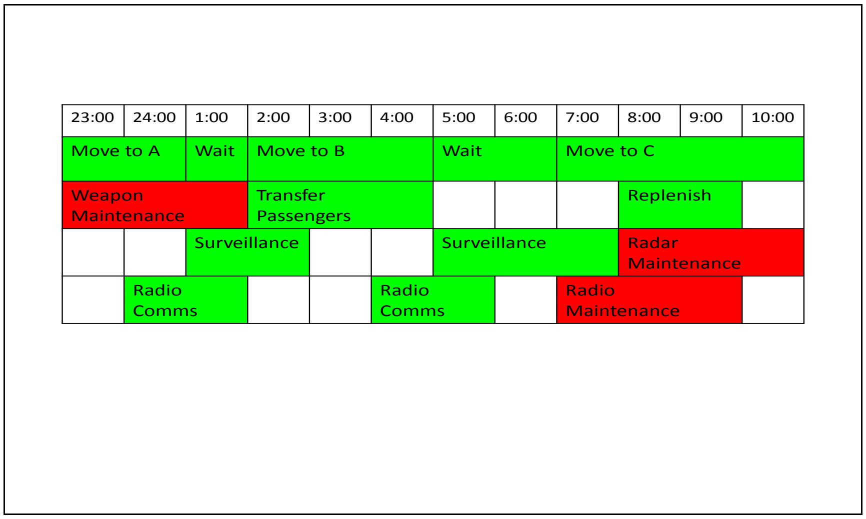

Figure 5-5 - Extension of Phases in a Phased Mission for Multi-function Platforms

In Figure 5-5, there is a more detailed representation of the results of applying the Figure 5-4 (b) approach to mission definition for a system (ship) that performs many distinct functions, and very often performs these functions concurrently. Maintenance downtime was also included for illustration. One notices the following problematic characteristics of this model for calculating phased mission availability:

1) There is no clear distinction between phases. In literature and this Thesis, phased missions are defined as discrete time intervals (though start and end may be subject to variance). To analyze the overlapping activities (subject to their independent variance in time, duration, and order), one would need a continuous approach where

[27] 'Flexibility' in the Naval environment refers to the tendency and necessity for plans to change, often.
[28] This type of decomposition applies to trying to estimate perceived Availability.

mission requirements have an instantaneous definition at a given time. Such an instantaneous definition cannot be well defined in advance of a Navy mission;

2) Maintenance downtime occurs at different times of the day for different systems (a reasonable approach to minimize the impact of the possibility of several systems being down at the same time). If, in a given day, there is a system down for maintenance at every moment, it is unclear whether the ship as a whole is unavailable during that day. Furthermore, if the ship has planned maintenance downtime at every moment during a mission, it is unclear if it is unavailable the entire mission. This is a critical consideration for defining the availability of a multifunction platform such as a Naval ship.

5.2.3 Capability

The platform required capabilities are functions needed to perform a mission. Warships are designed to be multi-role. In addition, it may be called upon to conduct new roles not initially conceived (or included) in its design. When missions are defined in phases, the required capabilities may differ amongst phases. For a ship to be considered available *throughout* a mission, its capability requirements must be met.

5.2.4 Degraded Availability

If a capability is unavailable, this does not mean the ship is unavailable and ceases current operations. Each capability deficiency[29] is assessed by operational authorities and decisions rendered regarding.

---

[29] referred to as operational deficiencies

For this Thesis, not only must the mission availability be assessed, but also the mission-capability availability assessed as well. A single value for ship availability does not provide sufficient insight.

5.2.5 Configuration

There are two methods to provide redundancy for a capability. The first is to design parallelism into key systems, commonly known as redundancy. The second is to have multiple but not identical systems capable of handling a capability. The former method is intended to be captured under the reliability models of the MCC, while the later is captured under having multiple configurations for a capability.

5.2.6 Degraded Capability

The existence of multiple configurations to meet the requirements of a capability does not mean that each configuration provides equal (or even adequate) performance. There may be scenarios where each configuration actually has superior performance over the other. An example is a ship with both a gas engine and a diesel engine to drive the propellers. The ship can move with either, yet the gas turbine engine is capable of higher speeds, while the diesel engine is capable of greater endurance (albeit at a lower cruising speed). Short and urgent trips suggest using the gas turbine engine, while long trips with time flexibility suggest the diesel engine.

5.2.7 Reliability Model

Each reliability model is constructed from the systems required for a configuration, and the applicable relationships. These models may be simple or complex.

For this Thesis, reliability block diagrams were used, and analysis conducted with that model structure as an assumption.

5.2.8 Maintenance Downtime

Maintenance downtime is a commonly used factor in determining system availability. As noted in 5.2.2 overlapping maintenance downtime has an increasingly negative impact on ship availability.

Example

Consider:

$$A = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR} + \text{MDT}} \tag{5.7}$$

And consider for the moment only maintenance downtime for Mean Down Time (MDT). Consider a ship with 20 systems, each system having 1200 hours MTBF, 2 hours MTTR, and 1 hour MDT/day. Then availability of a given system is:

$$\begin{aligned} A &= \frac{1200}{1200 + 2 + 1*(1202/(24-1))} \\ &= 0.9567 \end{aligned} \tag{5.8}$$

If we consider the whole ship, we might calculate the following (based on run time hours and assuming continuous run time when the system is not down):

$$
\begin{aligned}
&\textit{Availability with concurrent system maintenance downtimes}\\
&A=\frac{1200/10}{1200/10+2+1(\frac{1200/10+2}{23})}\\
&=0.9416\\
&\textit{or} \qquad (5.9)\\
&\textit{Availability with consecutive system maintenance downtimes}\\
&A=\frac{1200/10}{1200/10+2+10(\frac{1200/10+2}{14})}\\
&=0.5738
\end{aligned}
$$

Obviously, maintenance scheduling has a profound impact on the ship's calculated availability. The ratio of daily uptime, ignoring failures, 14/24 or .5833. Maintenance downtime becomes the dominate factor for availability. Figure 5-6 shows this relationship for a range of values. Increasing the number of systems to be considered gradually decreases availability, and increasing the number of maintenance hours has a very drastic impact.

This suggests that to attain optimal ship availability, maintenance downtime should be scheduled concurrently. There is an impact of this approach, during the time when all the systems are down for scheduled maintenance, the ship is at its most unavailable state; not all unavailable states are equal. If the downtime is spread then less[30] ship capability is impacted at any one time. The MCC reliability model applied, may imply that it is optimal to schedule *capability downtime* concurrently[31], and distribute the downtime of capabilities. Note that this strategy is most easily considered

[30] And fewer capabilities are impacted. Less is more precise, when considering degraded/partial capabilities.
[31] for capabilities that only have a single configuration.

when there are no systems requiring maintenance downtime, that simultaneously impacts more than one capability[32].

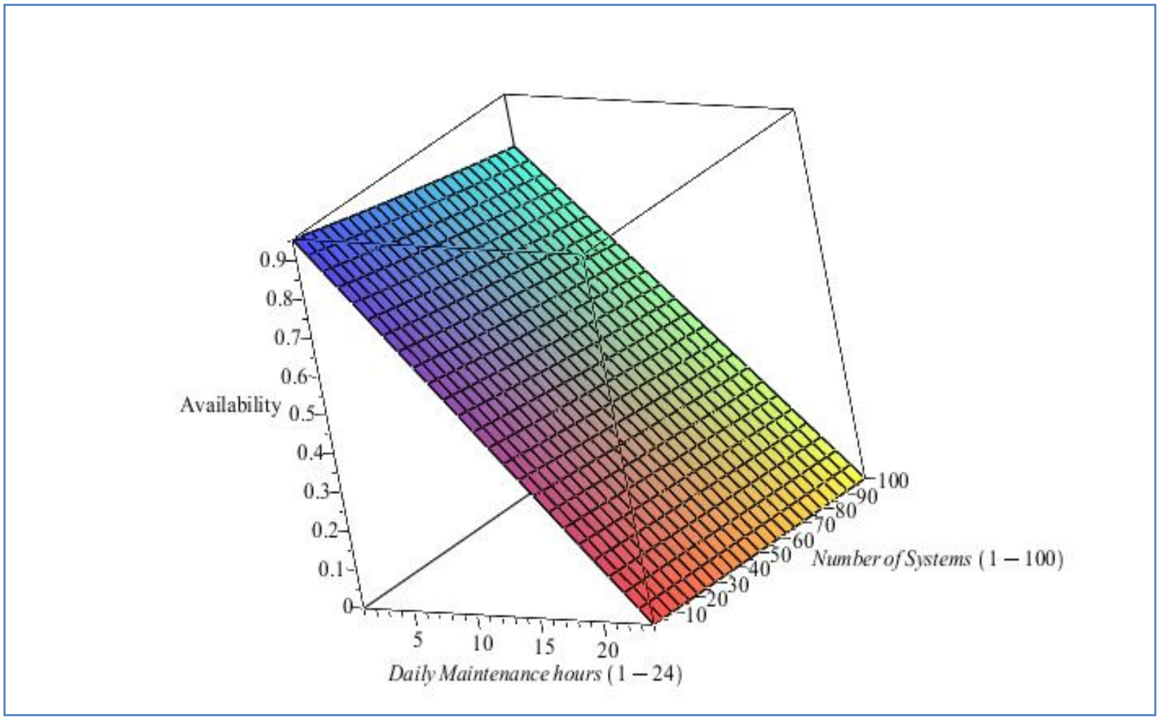

Figure 5-6 - Availability as a function of number of systems, and daily maintenance hours

5.2.9 Function performance measure versus reliability

There is a difference between calculating system reliability / availability with calculating the probability a function will be successful in its task for a given event or time period. Terms such as dependability[33] and effectiveness are sometimes used to discuss these other aspects of performance.

In this Thesis performance weights were applied to different configurations in order to provide an operational view of the implied performance.

[32] There is also a possibility that multiple configurations for a capability may create a similar manageable situation.

[33] reliability is a component of dependability which includes other factors

5.2.10 Evaluating the MCC RM for full capability availability

Given a mission, requiring c capabilities and c_i configurations for each capability Cap_i

$$
\begin{aligned}
\Pr_{Mission}(T > t) &= \Pr(\bigcap_i Cap_i) \\
\Pr(Cap_i) &= \Pr(\bigcup_j Config_{ij}) \\
\Pr(Config_{ij}) &= \text{Reliability Model}
\end{aligned}
\tag{5.10}
$$

$$
\begin{aligned}
P_{Mission}(T > t) &= \Pr(Cap_1)\Pr(Cap_2 \mid Cap_1)\Pr(Cap_3 \mid Cap_1 Cap_2)\ldots. \\
&= \Pr(\bigcup_j Config_{1j})(\bigcup_j Config_{2j} \mid Cap_1)\Pr(\bigcup_j Config_{3j} \mid Cap_1 Cap_2)\ldots \\
&= \left[\sum_{k=1}^{c1}(-1)^{k+1}\sum \Pr(\cap\, k\ Config_{1j})\right]\left[\sum_{k=1}^{c2}(-1)^{k+1}\sum \Pr(\cap\, k\ Config_{2j} \mid Cap_1)\right]\left[\sum_{k=1}^{c3}(-1)^{k+1}\sum \Pr(\cap\, k\ Config_{3j} \mid Cap_1 Cap_2)\right]\ldots
\end{aligned}
$$

(5.11)

Having c capabilities -> c terms in (5.11). Each term has 2^{c_i} probabilities (power set). Assuming c_i=z, a constant, then we have $O(c2^z)$ to compute $P_{mission}$

A sub-problem of (5.11) is evaluating Pr($Config_i$|$Config_j$). Used in $P_{mission}$ above, this term refers to configuration within one capability, conditional on a configuration in another capability. Within the same capability, configurations are in parallel.[34]

$$
\Pr(Config_i \mid Config_j) = \frac{\Pr(Config_i \cap Config_j)}{\Pr(Config_j)} \tag{5.12}
$$

$$
= \frac{\left[\prod_{l=1}^{N}\Pr(System_{il})\right]\left[\prod_{l=N+1}^{\#systems(Config_i)}\Pr(System_{il})\right]\left[\prod_{l=N+1}^{\#systems(Config_j)}\Pr(System_{jl})\right]}{\prod_{k=1}^{\#systems(Config_j)}\Pr(System_{jk})} \tag{5.13}
$$

[34] but likely these configurations are not disjoint.

Equation (5.13) assumes a series relationship amongst systems. While this is not correct for the entire ship, it may be a useful assumption for common systems amongst capabilities. Equation(5.14) finishes the analysis.

$$\Pr(Config_i \mid Config_j) = \frac{\left[\prod_{l=1}^{N} \Pr(System_{il})\right]\left[\Pr(Config_i - common)\right]\left[\Pr(Config_j - common)\right]}{\Pr(Config_j)}$$

$$= \frac{\left[\prod_{l=1}^{N} \Pr(System_{il})\right]\left[\Pr(Config_i - common)\right]\left[\Pr(Config_j - common)\right]}{\left[\prod_{l=1}^{N} \Pr(System_{il})\right]\left[\Pr(Config_j - common)\right]} = \left[\Pr(Config_i - common)\right] \quad (5.14)$$

The continuation of expansion of this form increase in complexity. Instead of a probability of a configuration given a configuration, the probability of a configuration given a capability consisting of the union of configurations is required. Other means of evaluating ship availability are defined in Section 5.4.

5.2.11 Failure modes that affect more than one system

Failure consideration in this Thesis has been for failure modes that are independent from system-to-system (though dependencies may occur within a system, and are captured by the reliability model). Failure modes that affect multiple components, or systems simultaneously has not been investigated in this Thesis.

5.3 Ship System Reliability Modeling (Example)

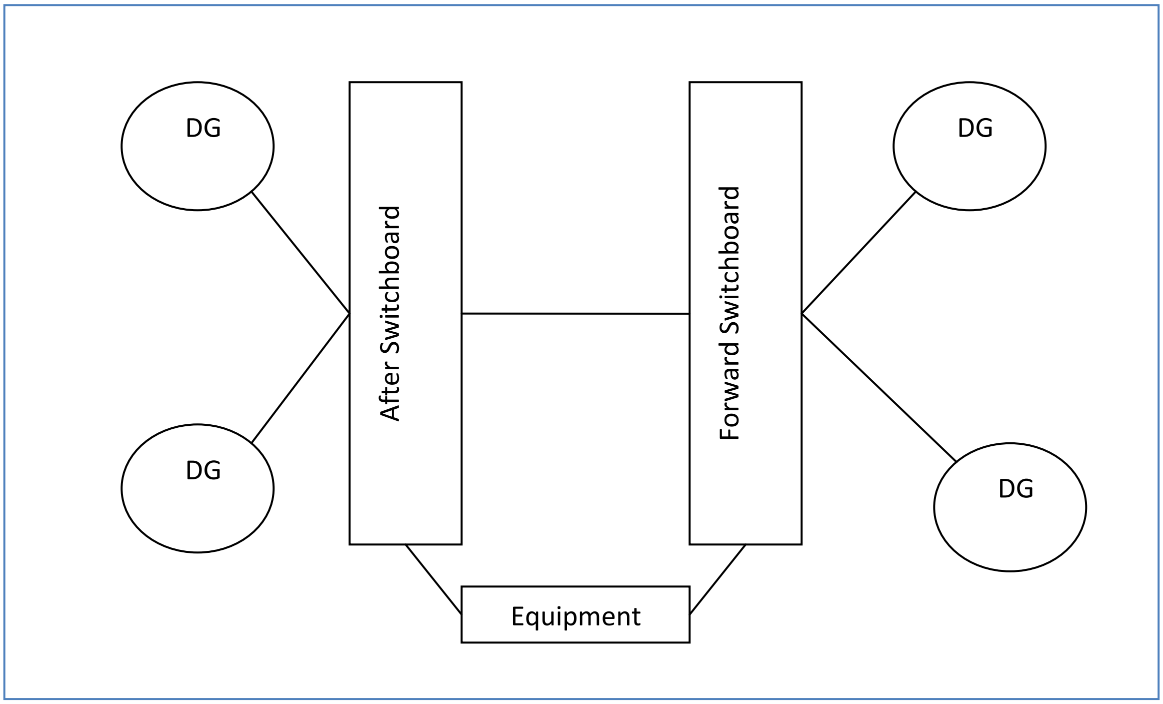

Figure 5-7 - Top Level Diagram of Power Generation

In Figure 5-7 we have the system diagram of the ship's power and distribution system. In Figure 5-8 we have a RBD modeling this system. Four generators feed two switchboards, which in turn feed various loads throughout the ship. There is an option to take on power from shore. A casualty power system exists should it be necessary to 'jury rig' power directly to a generator. Two of the four generators are required, unless shore power is available. Some of the blocks in this example have been mirrored in BlockSim, and they show up more than one location in the diagram. There is also a circular reference loop with the generators[35], as they can feed each other if required. It is notable, that even removing the circular reference loop, BlockSim was not able to provide an exact reliability solution to the power generation and distribution network.

[35] This requires special treatment in BlockSim, if one wishes BlockSim to provide an analytical solution.

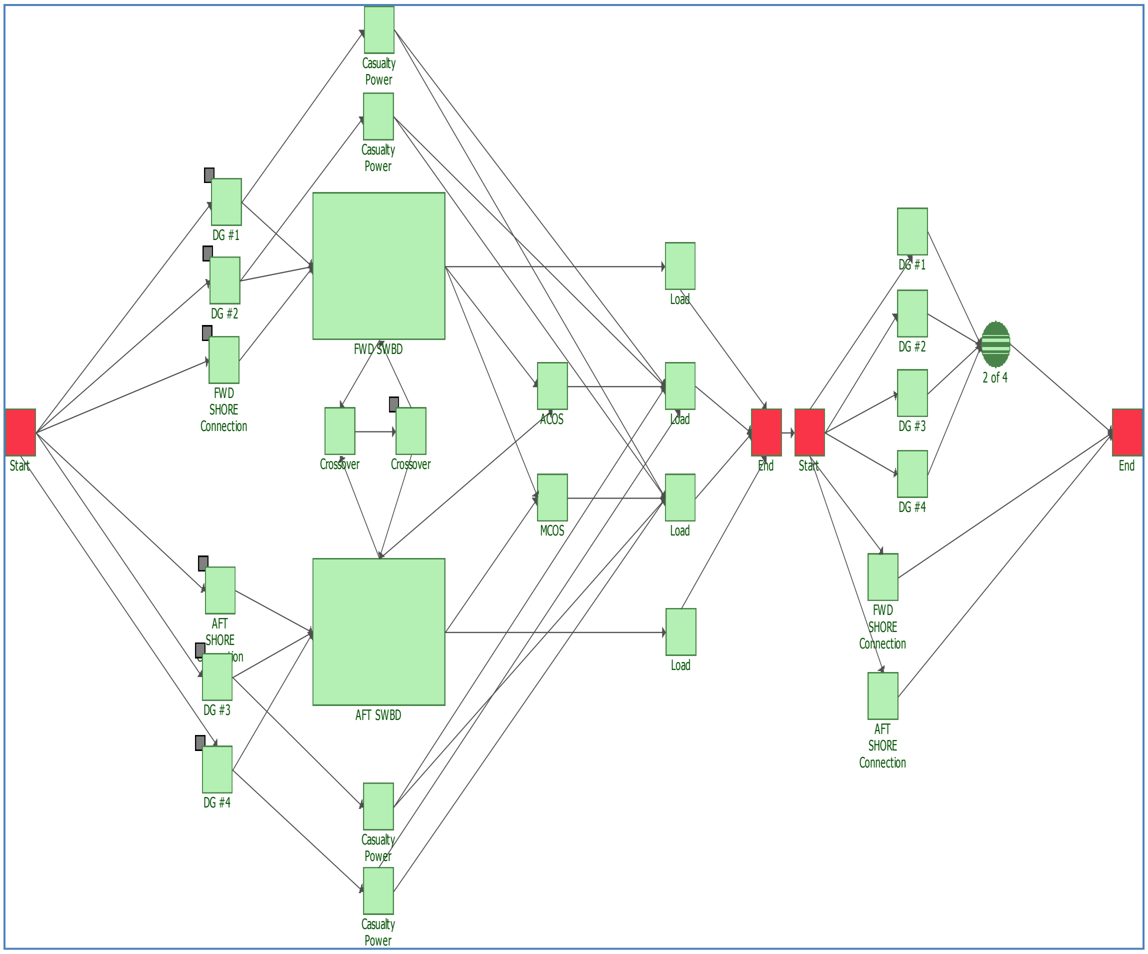

Figure 5-8 - Conceptual RBD of Ship Power Generation & Distribution (BlockSim)

5.4 Multifunction Ship Mission Availability Metrics

In this section, the availability metrics that are used in this Thesis are defined. These metrics are function / capability based.

First let's consider A_{1F} as a random variable representing availability of a group of items (i.e. ships) at a given point of time. For single function availability, A_{1F} is represented in equation (5.15) as a time-dependent binomial distribution.

$$\begin{aligned} &A_{1F}(t) \sim B(n, p(t)) \\ &\Pr(k) = \binom{n}{k} p(t)^{k} q(t)^{n-k} \end{aligned} \tag{5.15}$$

where k is the number of available items, $p(t)$ is the ratio of working items, and $q(t)$ is the ratio of non-working items (regarding the single function). The function $p(t)$ corresponds to $R(t)$ for non-repairable items, and $A(t)$ for repairable items.

Now consider A_{1F3L}, a random variable of a group of items with three levels of availability (full, partial, not):

$$\begin{aligned} &A_{\mathrm{1F3L}}(t) \sim Multinomial(n, p_1(t), p_2(t), q(t)) \\ &\Pr(n_1, n_2, n_3) = \binom{n}{n_1\, n_2\, n_3} p_1(t)^{n_1}\, p_2(t)^{n_2}\, q(t)^{n_3} \end{aligned} \tag{5.16}$$

where n_{i} represents the number of items in each state. Most general forms of $A(t)$ do not handle this case. This distribution can be converted to a binomial for either the case of requiring full availability, or just requiring partial availability.

Next consider a two function system (F1, F2), where each function has two levels of availability and both are required for a system to be functioning:

$$\begin{aligned} &A_{\mathrm{F1}}(t) \sim B(n, p_{\mathrm{F1}}(t)) \\ &A_{\mathrm{F2}}(t) \sim B(n, p_{\mathrm{F2}}(t)) \\ &\Pr(A) = \Pr(\text{F1 working}) \cdot \Pr(\text{F2 working}) \\ &\Pr(k) = \binom{n}{k} \left(p_{\mathrm{F1}}(t) p_{\mathrm{F2}}(t) \right)^k \left(1 - p_{\mathrm{F1}}(t) p_{\mathrm{F2}}(t)\right)^{n-k} \end{aligned} \tag{5.17}$$

where k is the number of items available, A_{F1} is the availability of the first function, A_{F1} is the availability of the second function. Independence of functions was assumed. This case can be modeled like equation (5.15) by setting $p(t) = p_{\mathrm{F1}}(t) p_{\mathrm{F2}}(t)$ and then general forms of $A(t)$ can be applied.

Now consider two functions, the first, F1, has two levels, and the second, F2, has three levels of availability. Define the system has available if the both functions are fully available or partially available. Independence of functions is still assumed.

$$\begin{aligned}
&A_{\mathrm{F1}}(t) \sim B(n, p_{\mathrm{F1}}(t)) \\
&A_{\mathrm{F2}}(t) \sim \mathit{Multinomial}(n, p_{\mathrm{F21}}(t), p_{\mathrm{F22}}(t), q_{\mathrm{F21}}(t)) \\
&\Pr(A) = \Pr(\mathrm{F1\,working}) \cdot \Pr(\mathrm{F2\,working}) \\
&\Pr(k) = \binom{n}{k} \left(p_{\mathrm{F1}}(t)(p_{\mathrm{F21}}(t) + p_{\mathrm{F22}}(t)) \right)^k \left(1 - p_{\mathrm{F1}}(t)(p_{\mathrm{F21}}(t) + p_{\mathrm{F22}}(t)) \right)^{n-k}
\end{aligned} \tag{5.18}$$

Now a more complicated case to consider with three function s (F1, F2, F3) with three levels of availability each. All functions must have at least partial availability. Dependence is now considered.

$$\begin{aligned}
&A_{\mathrm{F1}}(t) \sim \mathit{Multinomial}(n, p_{\mathrm{F21}}(t), p_{\mathrm{F22}}(t), q_{\mathrm{F21}}(t)) \\
&A_{\mathrm{F2}}(t) \sim \mathit{Multinomial}(n, p_{\mathrm{F21}}(t), p_{\mathrm{F22}}(t), q_{\mathrm{F21}}(t)) \\
&A_{\mathrm{F3}}(t) \sim \mathit{Multinomial}(n, p_{\mathrm{F21}}(t), p_{\mathrm{F22}}(t), q_{\mathrm{F21}}(t)) \\
&\Pr(A) = \Pr(\mathrm{F1}) \cdot \Pr(\mathrm{F2|F1}) \cdot \Pr(\mathrm{F3|F2,F1}) \\
&\Pr(k) = \binom{n}{k} (p)^k (1-p)^{n-k} \\
&\text{where } p = (p_{\mathrm{F11}}(t) + p_{\mathrm{F12}}(t))(p_{\mathrm{F21|F1}}(t) + p_{\mathrm{F22|F1}}(t))(p_{\mathrm{F31|F2,F1}}(t) + p_{\mathrm{F32|F2,F1}}(t))
\end{aligned} \tag{5.19}$$

Equation (5.19) represents multifunction availability of the MCC RM involving entangled systems that support multiple capabilities. The empirical results from either collecting field data or simulation results fit this model. This metric approach applies to critical systems.

Using the case of equation (5.19) but changing the availability requirement to simply having any of the functions available, we get the following:

$$
\begin{aligned}
&A_{\mathrm{F1}}(t) \sim Multinomial(n, p_{\mathrm{F11}}(t), p_{\mathrm{F12}}(t), q_{\mathrm{F1}}(t)) \\
&A_{\mathrm{F2}}(t) \sim Multinomial(n, p_{\mathrm{F21}}(t), p_{\mathrm{F22}}(t), q_{\mathrm{F2}}(t)) \\
&A_{\mathrm{F3}}(t) \sim Multinomial(n, p_{\mathrm{F31}}(t), p_{\mathrm{F32}}(t), q_{\mathrm{F3}}(t)) \\
&\Pr(A) = \Pr(\mathrm{F1} \cup \mathrm{F2} \cup \mathrm{F3}) \\
&\qquad = 1 - \Pr(\overline{\mathrm{F1}} \cap \overline{\mathrm{F2}} \cap \overline{\mathrm{F3}}) \\
&\qquad = 1 - \Pr(\overline{\mathrm{F1}}) \cdot \Pr(\overline{\mathrm{F2}} | \overline{\mathrm{F1}}) \cdot \Pr(\overline{\mathrm{F3}} | \overline{\mathrm{F2}}, \overline{\mathrm{F1}}) \\
&\Pr(k) = 1 - \binom{n}{k} (q)^k (1-q)^{n-k} \\
&\text{where } q = q_{\mathrm{F1}}(t) \cdot q_{\mathrm{F2|F1}}(t) \cdot q_{\mathrm{F3|F2,F1}}(t)
\end{aligned}
\tag{5.20}
$$

Equation (5.20) shows a binomial form of the unavailability for this case. Note that F1, F2, and F3 are not disjoint events (each can occur while the other occurs). Thus application of the inclusion-exclusion principle (or some clever reduction) would be required to evaluate the probability from the union form.

5.4.1 Metrics defined for the Simulation

Three metrics are used in this Thesis for Ship Mission Availability based on the cases for equations (5.19) and (5.20). Many systems on a ship support multiple capabilities, thus dependence exists.

Critical Availability is the availability of all critical capabilities at a given time. This is the intersection of each capability (some may have multiple levels). This value is averaged over the period of interest.

Mean Capability Availability is the average number of capabilities at a given time. Given the condition that at least one capability has at least partial availability, then this definition is in a form related to equation (5.20). This value is averaged over the period of interest.

Mean Weighted Performance Availability is the average of the sum of performance weights associated with each capability. Capability weights have value from 0 to 1, where 1 represents fully available, 0 represents unavailable, and anything in-between represents partial availability. Weights are defined for each capability, and represent the linkage between the simulation developed in this Thesis and operations. Similar to Mean Capability Availability, as long as the ship as one capability working, then this metric represents some level partial performance. Defining these values requires knowledge of the capability impact on mission performance. This value is averaged over the period of interest.

There is no reason for the particular performance metric chosen here, other than illustration. In reality, the linkage between capability availability and performance can be very complicated, and possibly segregated into a set of performance values (aggregating the performance values with each other might not make sense).

6 Phase III - Simulation Construction

The MCC RM was implemented in a simulation. The simulation, described here, was created using the disciplined approach of Modeling and Simulation Life Cycle (Loper 2009) (Jafari 2010). Adapted steps from the Life Cycle are:

a) Problem Formulation / System Design;

b) Conceptual Models;

c) Data Collection / Input Analysis;

d) Simulation Model Building;

e) Model Verification;

f) Model Validation;

g) Experimental Design;

h) Output Analysis; and

i) Documentation.

This section focuses on those details particular to the simulation itself. Information related to the Life Cycle steps, especially problem formulation, research, and results, are kept logically in their own sections.

6.1 Problem Formulation / System Definition

Problem formulation and system definition sets the scope and purpose of the simulation design. It acts as the simulation's top level design document and a scope management tool. The simulation problem definition is given in Table 3.

Table 3- Simulation Problem Definition

Subject Area:	Mission Availability
System:	HALIFAX Class Frigate
Objective:	Assess mission availability of a ship for either determining initial acquisition requirements or assessing in-service performance
Models:	• Ship (Mission-Capability-Configuration Reliability Model) • Ship Capabilities • Ship Systems • Maintenance • Operational Cycle • Mission
Inputs:	• Failure Models • Maintenance Models (CM, PM, IM) • Sparing / Logistic Models • Mission
Outputs:	• Ship Mission Availability • Ship Performance • Decomposed Availability and Reliability (Capability, Configuration, System, Component)
Goals:	Implement a basic and relevant operational cycle of a ship, and extract availability and performance information

6.2 Conceptual Models

Conceptual models are the initial models from which the simulation models are designed and constructed. Conceptual models provide the relationships and logic that are then converted into simulation design and logic. This Section outlines the key conceptual models.

6.2.1 MCC RM

The MCC RM is described in Section 5.2. The reliability model level of Figure 5-2 is split into system and component levels. All levels of the MCC RM are simulated providing five levels of analysis. The purpose of each level are:

a) Mission (Ship). Study ship mission availability and performance;

b) Capability. Study the availability of each function;

c) Configuration. Determine configuration availability for meeting capability requirements;

d) System. Study system availability. PM is controlled at this level; and

e) Component. Study failure modes at lowest level. This is also the repairable level.

6.2.2 Tiered Spares Source Model

This simulation model incorporates a 4+1 tiered supply system. As shown in Figure 6-1, the tiers are Ship, Supporting Supply Ship, Warehouse, and OEM, with an additional tier for backup / emergency ordering.

While delays to access to each level can be set as desired, the general default assumption is one hour to access a spare part from the ship[36], 24 hours to access the part from a support ship, two weeks to access the part from the warehouse, the OEM order is manually set[37], and back-up orders[38] take 4,800 hours.

---

[36] This can be seen as incorporating the processes of initial investigation, selecting a part, finding the part, setting up the component for

[37] Both order interval and order quantity.

[38] Unplanned orders to recover from excessive spare demands or to replace non-repairable systems.

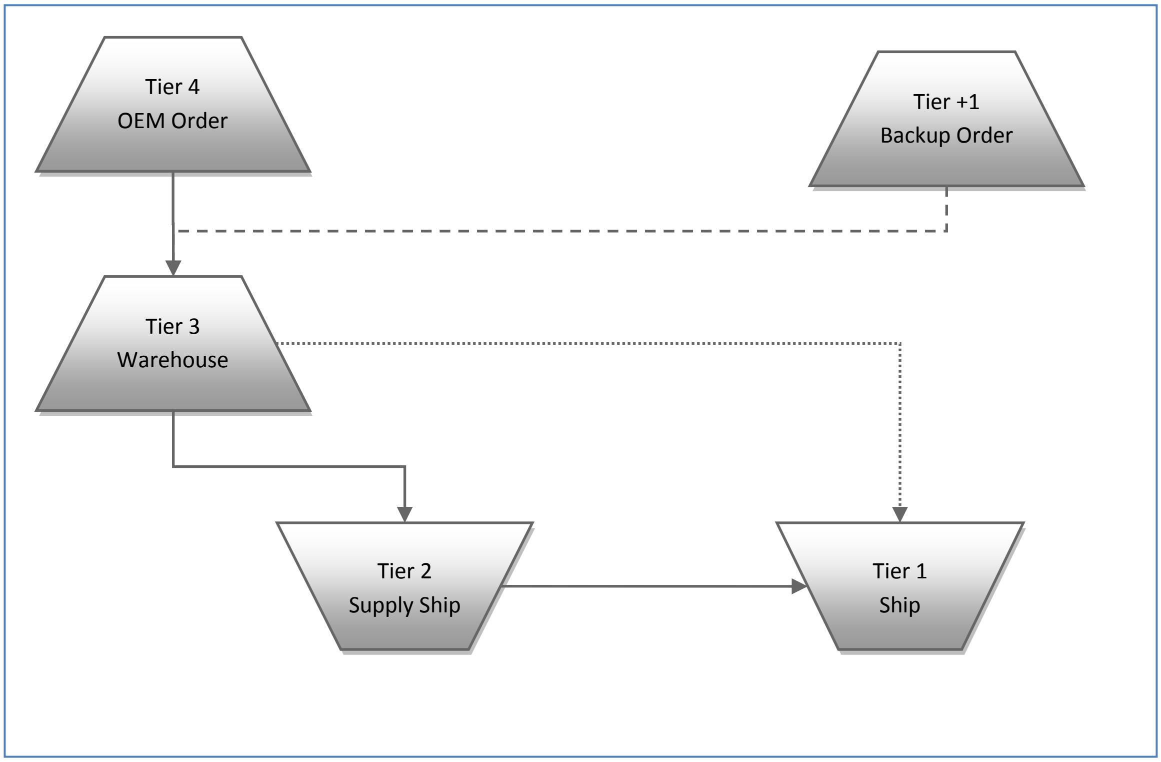

Figure 6-1- Simulation Tiered Supply System

6.2.3 Mission

Mission was defined in sections 5.2.1 and 5.2.2. For the default scenario, the ship has a two year operational period, including a three month deployment/mission with one month transit periods. This is illustrated in Figure 6-2.

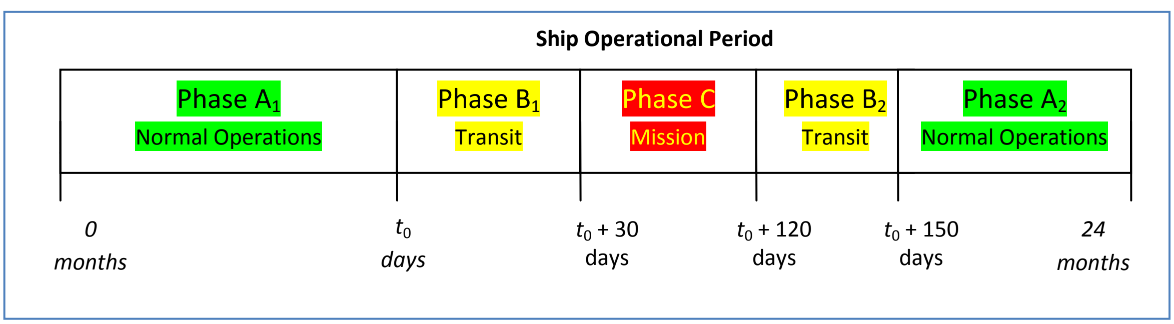

Figure 6-2- Mission Definition for Simulation

Some scenarios did not use a mission profile, and this is noted where applicable. Phase B_1 start time t_0 can either be pre-set or randomly determined within the operational period, as long as there is sufficient time to complete the 150 day mission prior to the end of the operational period.

6.2.4 Maintenance

Corrective, preventative, and imperfect maintenance have been implemented in the simulation. Corrective maintenance is implemented at the component level, preventative at the system level, and imperfect maintenance modifies only the CM[39]. The ability to input expressions into Arena provides flexibility in choosing models for each.

Maintenance concepts and models are described in Section 3.5. The simulation uses immediate corrective maintenance, and block scheduled preventative maintenance. However, no replacement or restoration has been modeled for PM. IM is in form 'p-q' model (Section 3.5), and again only applied to CM. This setup is intended to be representative of actual maintenance policies on a ship.

For discussing the possible range of maintenance related distributions implementable in the simulation, it is relevant to discuss the variables available for these models. For corrective maintenance the variables: 'time', 'component effective age'[40] and 'total number of fails'[41] are available (component level). For preventative maintenance, the variables: 'time', 'system age'[42] and 'number of system failures' are available (system level).

[39] In this simulation, PM does not restore the component to 'new' nor is there a penalty applied for late PM; thus IM has no obvious effect.
[40] Effective age, dependent on maintenance actions.
[41] it's possible to tweak the simulator design to give usefully the number of fails from the system restorations to *age*=0, and to keep recording (and possibly partially restore) whenever imperfect maintenance happens.
[42] System total age from start, not effective age as in the component case.

For determining p or q in the 'p-q' IM model, the simulator assigns each technician a level of skill $Skill_{Tech}$ from 0 to 1, and each component is assigned a level of difficulty $Diff_{component}$ from 0 to 1. The resulting relationship is shown in equation (6.1) and equation (6.2).

$$p_{IM} = Diff_{component} \cdot Skill_{Tech} \tag{6.1}$$

$$p_{IM}(t, a_{component}, nf_{component}) = Diff_{component}(t, a_{component}, nf_{component}) \cdot Skill_{Tech} \quad \text{CM (component level)}$$
$$p_{IM}(t, a_{system}, nf_{sys}) = Diff_{component}(t, a_{system}, nf_{sys}) \cdot Skill_{Tech} \quad \text{PM (system level)}$$
(6.2)

where t is time, a is age, nf is total number of failures[43].

The first form equation (6.1) is the basic 'p-q' IM model described in Section 3.5. The second versions given in equation (6.2) note that in the design of the simulator, there is access to several variables for more complicated imperfect maintenance models[44]. Pham and Wang (1996) review many forms of imperfect maintenance. Those that potentially could be used with this simulator include the 'p-q' rule already stated, the '(p(t),q(t))' rule, and 'improvement factor' methods based on age and total number of failures. For the experiments in this Thesis, only the 'p-q' rule was used.

Maintenance in the real world requires tools, time, manpower, facilities and spares[45]. In the simulation it is assumed that the tools are available, and that the ship is the facility for repair. Time is provided by the user defined CM and PM models. Sparing is provided through the supply distribution system, and it is assumed that only one spare

[43] Refer to footnote 41 .
[44] IM for PM has not actually been implemented but if it was it could make use of this flexibility.
[45] Also appropriate references and technician competency.

part is required. This is not a large assumption as the spare part can be representative of a group of spares. The last assumption made, is that a repair only requires one technician. For some repairs this is true in real life; for other repairs, this is far from true. The one repair, one man assumption, if relaxed would make the drain on human resources more apparent[46].

6.2.5 Component Failure

Failures modeled in the simulation[47] occur at the component level and their affect on other levels (system, configuration, etc.) is evaluated in the simulation itself. The component level can be used to model failure modes, failures of components, assemblies, sub-assemblies, or even a system[48]. The time to failure is calculated at the start of a renewal cycle. Using Arena, there are several different distributions built-in (beta, continuous[49], discrete[50], Erlang, exponential, gamma, Johnson, lognormal, normal, poisson, triangular, uniform, Weibull). Simulation component variables that can be applied to the failure distribution include: 'age'[51], 'number of fails', and 'time'. These distributions combined with these variables can implement run time scale based failures and forms of degradation.

With perfect maintenance, the simulation calculates failure time at the start of a renewal cycle. It is typical to calculate the next failure time from the start of a renewal cycle using a run time based failure distribution (such as Weibull) assuming 'as-good-as -

[46] Making this modification will affect data collection and output analysis in order to provide accurate assessment of the accumulation of maintenance hours.
[47] Not to be confused with the simulation failing!
[48] A single component combined with a system level element essential equates the component with the system. This can be used when the simulation resolution of detail is considered sufficient at the system level.
[49] A special user defined continuous distribution.
[50] A special user defined discrete distribution.
[51] Effective age with imperfect maintenance implemented.

new' condition. When imperfect maintenance is implemented, the method of calculating the next failure time needs revisiting as the component's effective age is no longer zero and must be taken into account. If effective age is not taken into account, than the component experiences erroneous instantaneous failures at probability *F*(*t*<*age*).

$$\begin{aligned} \mathrm{E}[T \,|\, T > t] &= \int_t^\infty \tau \frac{f(\tau)}{R(t)} d\tau \\ &= \int_t^\infty \tau \frac{1}{R(t)} d(-R(\tau)) = -\tau \frac{R(\tau)}{R(t)}\bigg|_t^\infty + \int_t^\infty \frac{R(\tau)}{R(t)} d\tau \\ &= \int_t^\infty \frac{R(\tau)}{R(t)} d\tau + t \end{aligned} \tag{6.3}$$

The mean run time survival time based on effective age is given in equation (6.3) Based on this idea, one can adjust the failure time generation. For a Weibull distribution the adjustment is given in equation (6.4). Note that the exponential special case of Weibull reduces to a exponential generating distribution, the expected result due to its memory-less property.

Weibull distribution modification:

$$\begin{gathered} f(\tau \mid \tau > age) = \frac{\beta}{\eta}\left(\frac{\tau}{\eta}\right)^{\beta-1} \frac{e^{-(\frac{\tau}{\eta})^\beta}}{e^{-(\frac{age}{\eta})^\beta}} = \frac{\beta}{\eta}\left(\frac{\tau}{\eta}\right)^{\beta-1} e^{-(\frac{\tau}{\eta})^\beta + (\frac{age}{\eta})^\beta} \qquad \tau > age \\ R(\tau \,|\, \tau > age) = e^{-(\frac{\tau}{\eta})^\beta + (\frac{age}{\eta})^\beta} \qquad \tau > age \\ R(\tau \,|\, \tau > age) \sim Uniform(0,1) \\ \text{Set } e^{-(\frac{\tau - age}{\eta})^\beta + (\frac{age}{\eta})^\beta} = Uniform(0,1) \\ -(\frac{\tau}{\eta})^\beta + (\frac{age}{\eta})^\beta = \ln(Uniform(0,1)) \\ \tau = \left(age^\beta - \eta^\beta \ln(Uniform(0,1))\right)^{1/\beta} \\ \tau^* = \tau - age = \left(age^\beta - \eta^\beta \ln(Uniform(0,1))\right)^{1/\beta} - age \end{gathered} \tag{6.4}$$

where τ^* is the time until next failure, *age* is the effective age of the component, η is the characteristic life, β is the shape parameter, and τ is the predicted age of the component when it fails.

A similar and possible modification for the gamma distribution is given in equation (6.5). This is problematic to implement since τ cannot be easily isolated[52], and the lower incomplete gamma function must be evaluated during the simulation. With Arena it may be possible to setup a pair of lookup tables; one for the lower incomplete gamma function with effective age, and the other to pick τ. This would be time consuming to create and specific to a single gamma distribution. In the simulation's current configuration, a lookup table method is the only practical option[53].

Gamma distribution modification:

$$\begin{aligned}
f(\tau \mid \tau > age) &= x^{k-1}\frac{e^{-x/\theta}}{\theta^k\Gamma(k)}\frac{1}{1-\dfrac{\gamma(k,age/\theta)}{\Gamma(k)}} \qquad \tau > age \\
&= x^{k-1}\frac{e^{-x/\theta}}{\theta^k}\frac{\Gamma(k)}{\Gamma(k)-\gamma(k,age/\theta)} \\
&= f(\tau)\frac{\Gamma(k)}{\Gamma(k)-\gamma(k,a/\theta)} \qquad \tau > a \\
R(\tau \mid \tau > age) &= \left(1-\frac{\gamma(k,\tau/\theta)}{\Gamma(k)}\right)\frac{\Gamma(k)}{\Gamma(k)-\gamma(k,age/\theta)} = \frac{\Gamma(k)-\gamma(k,\tau/\theta)}{\Gamma(k)-\gamma(k,age/\theta)} \qquad \tau > age \\
R(\tau \mid \tau > age) &\sim Uniform(0,1) \\
\text{Set } \frac{\Gamma(k)-\gamma(k,\tau/\theta)}{\Gamma(k)-\gamma(k,age/\theta)} &= Uniform(0,1) \\
\gamma(k,\tau/\theta) &= -Uniform(0,1)\cdot(\Gamma(k)-\gamma(k,age/\theta))+\Gamma(k) \\
\tau^* &= \tau - age
\end{aligned} \tag{6.5}$$

[52] For gamma distribution where k is not an integer.

[53] Using VBA, it is possible to automatically create the lookup table during simulation initialization.

where τ^* is the time until next failure, *age* is the effective age of the component, θ is the characteristic life, k is the shape parameter, and τ is the predicted age of the component when it fails.

Alternatively the simulation could be redesigned. A VBA method could be created to evaluate the age dependent gamma function. It is also possible to have the VBA method simply run the base gamma distribution (or any base run time failure distribution) until it finds a valid value, though the number of attempts to create this value cannot be predetermined.

Similar to the gamma distribution, the normal distribution modification given in equation (6.6) and the lognormal distribution in equation (6.7) require lookup tables. All formulas are summarized on Table 4. For this Thesis, the modified Weibull distribution is convenient to implement and sufficient for experimentation and research.

Normal Distribution modification:

$$f(\tau|\tau > age) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\tau-\mu}{\sigma}\right)^2} \frac{1}{1-\phi\left(\frac{age-\mu}{\sigma}\right)} \qquad \tau > age$$

$$= f(\tau)\frac{1}{1-\phi\left(\frac{age-\mu}{\sigma}\right)} \qquad \tau > a$$

$$R(\tau|\tau > age) = \frac{1-\phi\left(\frac{\tau-\mu}{\sigma}\right)}{1-\phi\left(\frac{age-\mu}{\sigma}\right)} \qquad \tau > age$$

$$R(\tau|\tau > age) \sim Uniform(0,1)$$

$$\text{Set } \frac{1-\phi\left(\frac{\tau-\mu}{\sigma}\right)}{1-\phi\left(\frac{age-\mu}{\sigma}\right)} = Uniform(0,1)$$

$$\phi\left(\frac{\tau-\mu}{\sigma}\right) = Uniform(0,1)(1-\phi\left(\frac{age-\mu}{\sigma}\right)) - 1$$

$$\tau^* = \tau - age \qquad (6.6)$$

Lognormal distribution modification:

$$\phi\left(\frac{\ln\tau-\mu}{\sigma}\right)=Uniform(0,1)(1-\phi\left(\frac{\ln age-\mu}{\sigma}\right))-1$$
$$\tau^{*}=\tau-age \qquad (6.7)$$

Table 4- Aged Component Failure Time Generation Expressions

Distribution	Generation Expression given *T>age* τ is effective age, τ^{*} is 'run time' until failure
Exponential	exponential(λ)
Weibull	$\left(age^{\beta}-\eta^{\beta}\ln(Uniform(0,1))\right)^{1/\beta}-age$
Gamma	Using lookup tables for $\gamma(k,age/\theta)$, and $\gamma(k,\tau/\theta)$ $\gamma(k,\tau/\theta)=-Uniform(0,1)\cdot(\Gamma(k)-\gamma(k,age/\theta))+\Gamma(k)$ $\tau^{*}=\tau-age$
Normal	Using lookup tables for $\phi(\cdot)$ $\phi\left(\frac{\tau-\mu}{\sigma}\right)=Uniform(0,1)(1-\phi\left(\frac{age-\mu}{\sigma}\right))-1$ $\tau^{*}=\tau-age$
Log normal	Using lookup tables for $\phi(\cdot)$ $\phi\left(\frac{\ln\tau-\mu}{\sigma}\right)=Uniform(0,1)(1-\phi\left(\frac{\ln age-\mu}{\sigma}\right))-1$ $\tau^{*}=\tau-age$

6.2.6 State Model

The simulation incorporates five states for describing components, systems, configurations, and capabilities (but not ship). Not all states apply to these levels; capability and configuration are only defined by 'uptime' and 'downtime'. These states and their possible transitions are shown in Figure 6-3. Uptime state represents an item working and being used. Fail state represents a component that is broken, but does not have the resources in place to start CM. CM state represents a component under repair.

PM state represents a system and component undergoing PM; for a component, the CM state takes priority over the PM state. Downtime state represents an item in a state of non-usage. Off Phase State is used when a system and its components are not required for a particular phase in a phased mission scenario. Simulation events are evaluated by their effect on these states at each level.

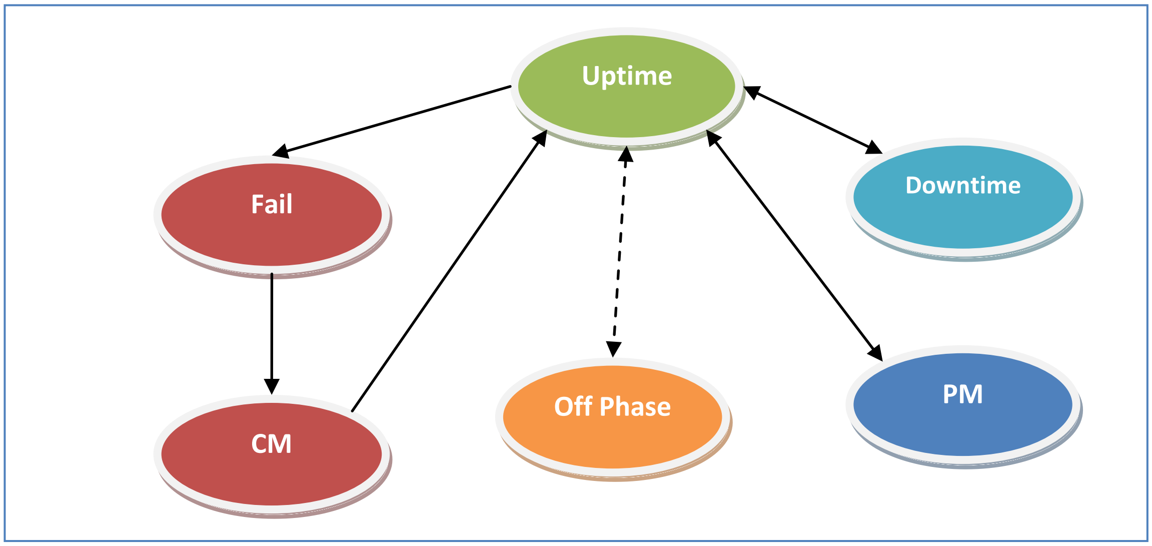

Figure 6-3 - Simulation State Transitions

The state model was the driving logic for the implementation of each level of the MCC RM in the simulation. Each level had to correctly implement the state changes, while recording relevant data for later analysis.

6.2.7 Ship and System Models

The models described in this section are for the generic ship studied. For experiments that do not use this model, it is noted as such in their respective sections.

Table 5 provides the default MCC RM implemented in the simulation for the active mission deployment phase. The Mission capability requirements are based on CFCD129 (2009) and chosen for how they affect ship mission level performance and interact with each other. Main Propulsion is independent of other capabilities and critical to ship's operation. Power Generation and Distribution is a critical capability, that is also required for most other capabilities. Both critical capabilities have graceful degradation[54]. Long Range Surveillance represents a capability that can 'only' be satisfied by a single system. Navigation represents a capability where two configurations can satisfy it equally. Medium Range Surveillance represents a capability that can be met by two configurations, though the secondary configuration, when required, has degraded performance. Finally AAW represents a capability that can be handled at some level by three configurations each by itself, and performance is maximized by the simultaneous availability of each[55]. With six capabilities (two critical), nine distinct configurations (some interrelated), and twelve systems, this is considered a representative of what a full MCC RM might look like for a ship[56].

Table 5 - Default MCC RM for simulation

Mission Level	Capability	Configuration	Systems
Ship	Move (Critical)	Main Propulsion	Main Propulsion Lube Oil DFO
	PG&D (Critical)	Diesel Generators	Power Generation DFO Power
	Long Range Surveillance	Long Range Radar	LRS Cooling

[54] Propulsion has two gas turbines; if one is not available then ship speed is degraded. Power Generation has three generators in a k of N:G configuration. If 2 of N are required, and only one is working, this represents both a critical failure (for the mission) and a degradation for performance (some systems can continue working).

[55] This is different from the previous cases, where only one configuration (the primary, and sometimes the secondary) is needed for maximum performance.

[56] For comparison, a Halifax class frigate might be represented by 20+ capabilities using a mix of 65+ systems.

Mission Level	Capability	Configuration	Systems
			Power
	Medium Range Surveillance	Long Range Radar	LRS Cooling Power
		Medium Range Radar	MRS Cooling Power
	Navigation	Navigation Radar	Nav Radar Power
		Secondary Navigation Radar	Alternate Nav Radar Power
	AAW	Close In Defence	CIWS Med Range Radar Power
		Missile	Missiles FCS Power
		Gun	AA Gun FCS Power

6.2.7.1 Power Distribution

Figure 6-4 represents the RBD of the Power Generation and Distribution system. The system has three generators capable of providing power to either of the two switchboards. The generators exist in a *k*-of-3:G configuration depending on the load requirements of the active capabilities. Two terminals (MMIs) are used to control the system, and a source of fuel for the generators has been modeled.

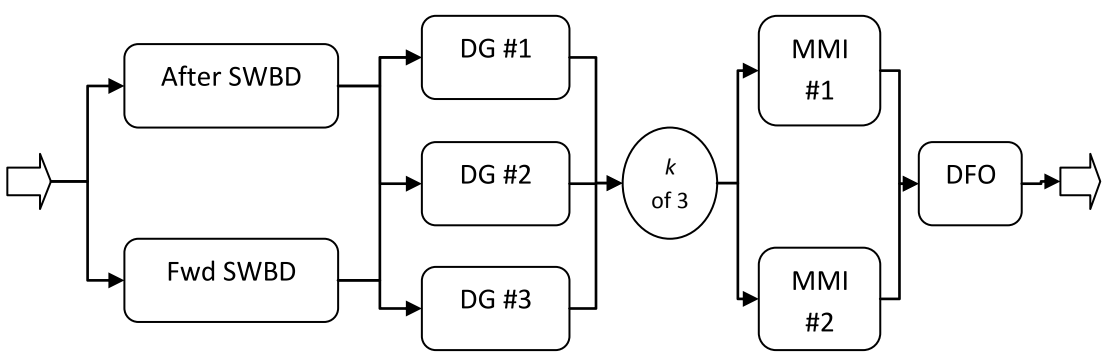

Figure 6-4- Power Distribution RBD

6.2.7.2 Medium Range Sensor

Figure 6-5 represents the Medium Range Radar RBD. All components are in series and consist of a terminal, cooling, power supply, equipment cabinet, and the external sensor (radar dish).

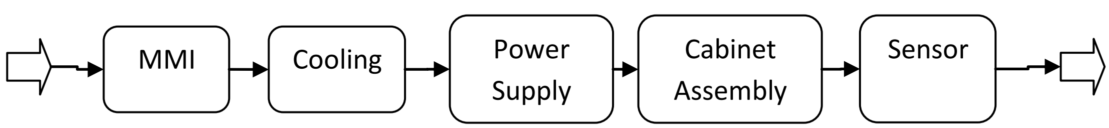

Figure 6-5 - Medium Range Radar RBD

6.2.7.3 Long Range Sensor

Figure 6-6 represents the Long Range Radar RBD. All components are in series, consisting of a terminal, two equipment cabinets, cooling, power supply, and the external sensor (radar dish).

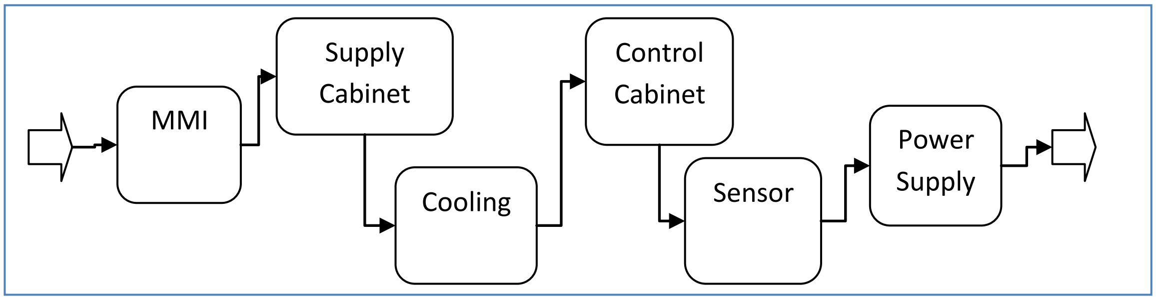

Figure 6-6 - Long Range Radar RBD

6.2.7.4 Navigation

Figure 6-7 represents the RBDs of the two navigation radars. Each has an interface, and an external sensor. The primary navigation radar also has an equipment cabinet.

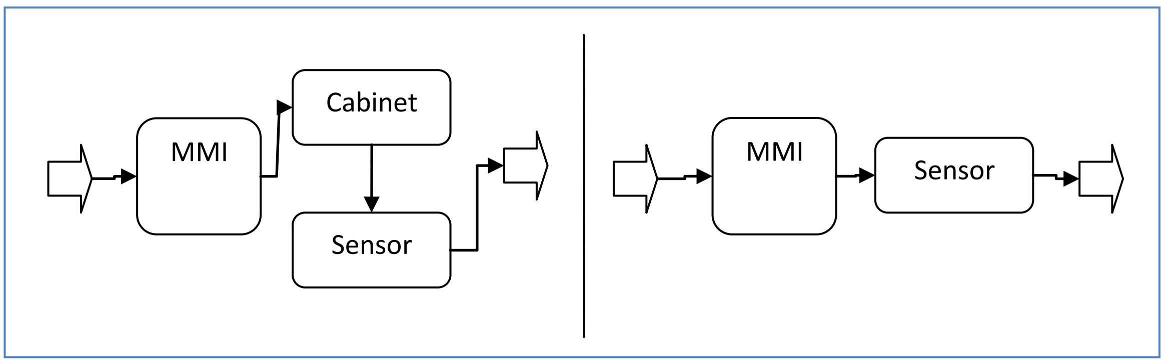

Figure 6-7 - (a) Navigation Radar RBD (b) Alt Navigation Radar RBD

6.2.7.5 AAW

Figure 6-8 represents the combination of AAW RBDs. The Fire Control System feeds is needed for the Missile system and the Gun system. The Close In Weapons System is stand alone.

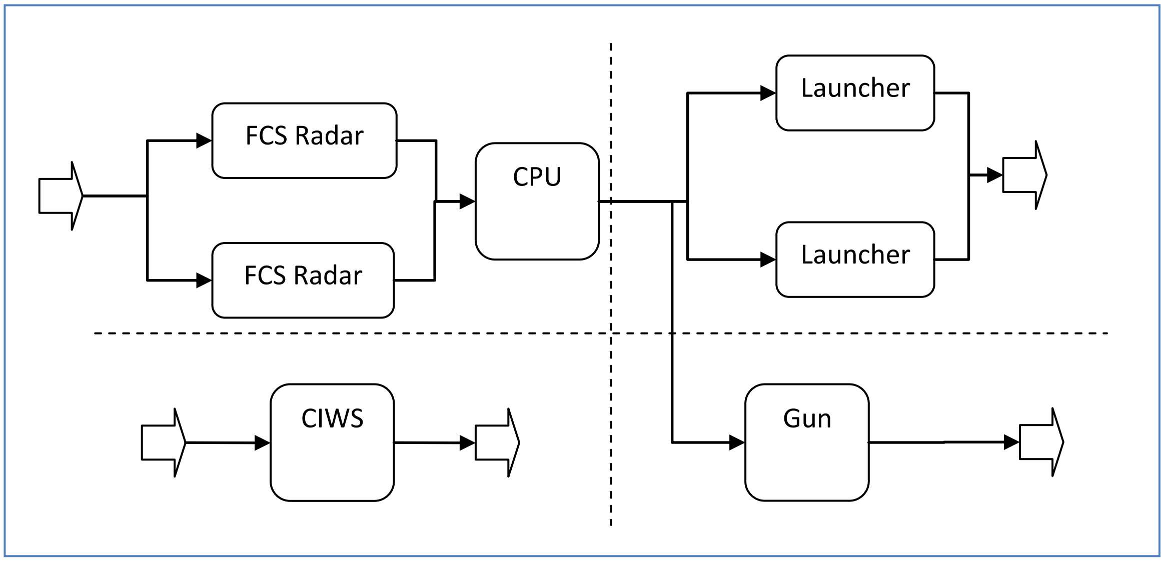

Figure 6-8 - AAW RBDs

6.2.7.6 *Main Propulsion*

Figure 6-9 represents the RBD of the Main Propulsion. Two gas turbine engines work in parallel, and two terminals work in parallel. There is a gearbox, fuel source and lubrication oil source in series.

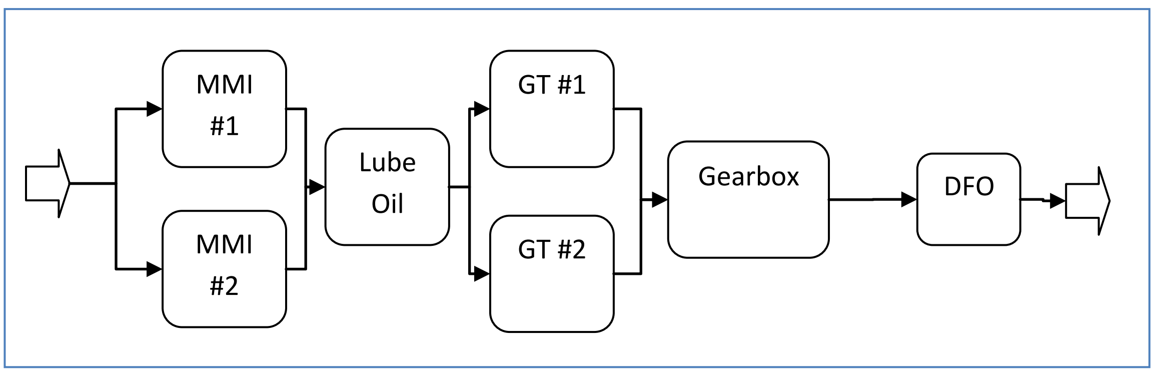

Figure 6-9 - Main Propulsion RBD

6.3 Simulation Inputs

The simulation has inputs to implement the models of section 6.2 and complete the model detail. These inputs are grouped as simulation design, mission, MCC design, system design, component design, supply design, spares design, technician design. The inputs built into this simulation for each group are listed in Table 6.

Table 6 - Simulation Inputs

Input Group	Inputs	Comments
Simulation	Number of Replications	
	Simulation Length	Operational Period
Mission	Number of Phases	
	Phase Durations	
MCC design	MCC Model	Ship, Capabilities, Configurations
	Configuration Weights	Performance Weights
	Configuration priorities	For each capability
System Design	System RBD Models	
	PM Model	
Component Design	Failure Model	
	Repair Model	
	IM Model	
	Technician Type	
	Spare Type	
	Related System	
Logistics Design	Shipment Times	From warehouse, From TG
Spares Design	Spares on Ship	
	Spares in TG	
	Spares in Warehouse	
	Regular Order	Order Interval, Quantity
	Backup Orders	Spare Threshold, Delay
Technician Design	Type and Quantity	
	IM Model	

Default Scenario inputs are listed in Appendix B.

6.4 Data Collection

The simulation output data is collected at five distinct levels. The levels are components, system, configuration, capability, and platform (ship). The purpose of each level are given in Section 6.2.1.

Data times are considered as discrete interval blocks. An interval of [1,10] includes hours one through ten and is equivalent to [1,11); the next interval starts at hour 11. This is different from a continuous time interpretation where consecutive time intervals end and start on the same number, for example (1,10) and (10,20). This difference has implications in data collection and simulation design.

6.4.1 Data Collection Requirements By Analysis Type

The different types of output analysis and the data required from the simulator are iterated in Table 7. The analysis types are described in section 6.8. Time scales can correspond to chronological time (the empirical time scale), and run time (the machine aging time scale). The run time scale is used when information about a machine's inherent performance isolated from unrelated events is desirable. Chronological time gives the actual observed behaviour.

Table 7 - Data requirements for different analysis methods

Analysis	Data Required	Time Scale	Comment
Mean Cumulative Function	Failure Times, Repair Completion Times, Recurrence Censoring times	Chronological Run time	Repair censoring times are derived in analysis

Availability	Up and Down times	Chronological	Downtimes not part of CM are derived[57]
Failure distribution	Failure Times	Run time	Derived from chronological information
Repair / Downtime distribution	Repair Times Maintenance Downtimes Censoring Times	Run time	Repair Censoring times not available in current output[58]

6.4.2 Component Output Data

Component data provides direct information regarding failures, repairs, and logistical delays. Data is recorded for the following events:

a. Periods of uptime. Initiated by changing of state to system downtime or PM. Also recorded at end of a simulation run to capture failure censoring.[59];
b. Failure events. The data recorded is the conclusion of an uptime interval. The failure event is considered to have occurred at the end of the last discrete hour, every hour of the interval recorded for the failure event is an uptime hour; and
c. Repair/downtime intervals. Downtime starts with the failure event and completes with the repair event. The repair event starts when a spare part and a technician are at the component.

---

[57] CM downtime is found at the component level, PM downtime is found at the system level.
[58] An oversight; repair censoring times can be added to the data output however this would affect certain assumptions in the output analysis algorithms.
[59] When a component fails, there is an asynchronous event waiting for a spare part and a tech. When these resources are available, the system proceeds to corrective maintenance. The waiting period is captured in downtime.

PM events are not recorded at the component level. In the chronological availability of a component, PM events can overlap with CM actions and CM occurs concurrently. PM time that does not overlap with a CM event is simply part of the chronological downtime of a component not attributed to CM.

To support the data collection requirements of Table 7, data was exported to an Excel spreadsheet in the following columns:

a. StartTime. The start of a recording interval;

b. StopTime. The end of a recording interval;

c. FailEvent. Indication that a failure has occurred;

d. RepairInterval. Indication that the time interval is a repair interval;

e. Component. Name of the component;

f. System. The associated system of the component. This information combined with component should be a unique identifier of the part for the simulation. With real systems, serial numbers should be used;

g. Replication. The sample replication number.

h. NumberFails. Number of component failures.

i. RunTime. A period of uptime on the given interval. This value is equal to StopTime-StartTime+1 due to the discrete time consideration;

j. RepairTime. Calculated similarly to Runtime. This time period represents downtime;

k. DownTime. This value is recorded during the repair part of a cycle, and represents the additional logistics delays for getting parts, and the technician. It is important to note that unlike RunTime and RepairTime, the downtime does not correspond to the [StartTime, StopTime] interval. It extends back to the initial failure, and represents the sum of downtime considerations throughout its length;

l. Age. Age is the tracked component age. When the component is repaired, its age is reset to zero. For imperfect maintenance, age is needed for 'as-bad-as-old' calculations; and

m. UpOrDown. This is a value indicating whether the [StartTime, StopTime] interval indicates uptime or downtime. The value is implied by the RunTime, RepairTime and DownTime columns, and is included for convenience of output analysis.

6.4.3 System Output Data

The system level provides direct information regarding PM and system failures. Data is recorded at the following events:

a. system uptime. This includes the censoring time at the end of a simulation run;
b. system failure. When a combination of component failures indicates a system failure, this transition is recorded; and
c. PM intervals. As these are controlled at the system level, they are recorded.

System repair time is not been recorded. In systems more complicated than a series system, multiple repairs can take place simultaneously. A repair on a component started before the system failure can restore the system before other repairs are completed. Repair times can be deceptively short. Thus the concept of system repair time in this simulation model is ambiguous. For calculating achieved availability, downtime intervals are used instead.

To support the data collection requirements of Table 7, data was exported to an Excel spreadsheet in the following columns:

a. StartTime. The start of a recording interval;
b. StopTime. The end of a recording interval;
c. FailEvent. Indication that a failure has occurred;
d. PMInterval. Indication that the [StartTime, StopTime] interval is a downtime interval due to PM;
e. System. The associated system;
f. Replication. The sample replication number;
g. NumberFails. Number of fails that the system observed. This is fewer or equal to the total failures observed by the components; and
h. Age. Unlike for the components, age does not reset for the system. It is used to determine the PM schedules.

6.4.4 Configuration Output Data

The configuration level provides information regarding a collection of systems that can be used to perform a mission capability requirement. Since this Thesis has no interpretation of Configuration repair, data is recorded only for uptime intervals.

To support the data collection requirements of Table 7, data was exported to an Excel spreadsheet in the following columns:

a. StartTime. The start of a recording interval;
b. StopTime. The end of a recording interval;
c. FailEvent. An observed failure at the configuration level;
d. Configuration. The associated configuration;

e. Capability. The associated capability;

f. Replication. The sample replication number; and

g. NumberFails. The number of observed fails.

6.4.5 Capability Output Data

The capability level captures not only the uptime related to a mission capability requirement, it also captures which configuration (when there are multiple) is actually used to meet that capability. Knowing which configuration is used, supports determining expected performance. Similar to configuration output data, only uptime intervals are recorded for capability.

To support the data collection requirements of Table 7, data was exported to an Excel spreadsheet in the following columns:

a. StartTime. The start of a recording interval;

b. StopTime. The end of a recording interval;

c. FailEvent. An observed failure at the configuration level;

d. Capability. The associated capability;

e. Replication. The sample replication number;

f. NumberFails. The number of observed fails;

g. CurrentConfiguration. The configuration currently being used to meet the capability requirement; and

h. WeightedPerformance. A performance factor associated with the configuration.

6.4.6 Ship Output Data

At the ship level, availability is a very ambiguous measure. This is the only level where multiple capabilities/functions are combined to provide mission availability information. At all other levels, a single function is assumed. Ship downtime periods only correspond to critical systems, yet unavailability of non-critical systems can have a dramatic impact on the ship's performance.

Data is recorded on every synchronous event that occurs at the component and system levels (the configuration and capability levels do not generate events in this simulation model). As such the ship level contains complete recorded data across the simulation run time.

To support the data collection requirements of Table 7, data was exported to an Excel spreadsheet in the following columns:

a. StartTime. The start of a recording interval;
b. StopTime. The end of a recording interval;
c. FailEvent. The moment when ship experiences a critical failure while in an uptime state. Non-critical failures are not failevents for the ship level;
d. Ship. The associated ship;
e. Replication. The sample replication number;
f. NumberFails. The number of observed critical fails;
g. NumCapWorking. The number of capabilities functioning on an interval;
h. EffectAvg. An average of the capability performance weights;
i. EffectProd. The product of the capability performance weights; and
j. (and other columns) CapX. The weighted performance of each capability is listed in its own column for reference.

6.5 Simulation Tools

Simulation model building refers to the transfer of appropriate information from the conceptual models to the actual simulation environment. There are two main simulation tools that were considered in this Thesis; ReliaSoft's BlockSim and Rockwell's Arena. Other software, such as Maple and Excel was used when appropriate. Arena was selected and used as the simulator for the simulation.

6.5.1 BlockSim

The reliability and maintenance tool BlockSim has been used for initial investigation. This tool was used to model system reliability interactions via reliability block diagrams and fault tree diagrams. Once modeled, BlockSim has both analytical and simulation capabilities that can be applied. Its simulation capabilities are useful for estimating system availability.

BlockSim has two features that made it seem ideal for applying the Mission-Capability-Configuration reliability model. These are sub-diagrams and mirrored blocks. Sub-Diagrams allow for a separation of Mission, Capability, Configuration, and System Reliability Models, and allows the simulation of availability at each level. Mirrored blocks allows a component to be placed in multiple places in the diagram; if it breaks in one location, it breaks in all locations. These features theoretically would makes it easier to apply the model in a verifiable fashion as it removes the need to populate endless connections to single components that interact with many other components. However, these two features cannot be combined; mirror blocks only work within the same sub-diagram and cannot cross between sub-diagrams, and it is not clear how to set up a reliability block diagram (without sub-diagrams) that gives availability of the levels of configuration, capability[60].

BlockSim's fault tree diagrams, as a simulation model, correspond closely to the Mission-Capability-Configuration reliability model. If a reliability model is built upon system failure analysis, then this may be an appropriate approach. This representation yields easy verification of the model's construction. Contrasted to an equivalent reliability diagram, and the problem of verification can be more easily understood with the fault tree setup. The RBD in its simplest form becomes a long series representation that does not in itself reveal the hierarchal relationships trying to be captured. However, a way was not found to analyze availability at the levels other than the complete BlockSim system (the mission) and individual components and another tool had to be considered.

6.5.2 Arena

Compared to BlockSim, Arena has less reliability functionality built into its default packages. In its packaging templates, its 'machine ' block does have reliability built into it, as well as its resource states. Important for this Thesis is that in Arena, it is possible separate the levels of an MCC RM for data collection and post-simulation analysis. Arena was selected as the primary simulation tool.

---

[60] The challenge with acquiring a configuration level availability in BlockSim is that the availability comes from a set of blocks. This is even trickier at the capability level, since its by MCC RM definition a parallel set of configuration, thus a parallel set of likely overlapping blocks.

6.6 Simulation Design and Construction

Several features of Arena were used to great advantage when building the simulation. The detailed simulation design (Arena files and SIMAN code) is not included in this Thesis document, and can only be released to licensed users of Arena.

The full academic version of arena was used. This version had full access to Arena premade template panels, unlimited model size, and the ability for the user to create their own templates. Each of these features were required for this simulations success.

Having full access to built-in template panels simplified the logic design, and notably provided access to VBA. VBA was used to automate data collection, an immense task due to the sheer number of data points. Without the automated data collection, this simulation would have been unwieldy.

Initial design work was completed using a student version of Arena. The student version has a hard limit of number of modules in a simulation. This limit was reached with a single component simulation setup, and attempting to add a second component was futile. By switching to the academic version, the module limit was relaxed, and full simulations could be completed.

The final feature noted here, is the ability to create custom templates. This meant that instead of coding a 60 module logic for each and every component, a single template module could be dragged into the simulation, its parameters set through a convenient interface, and all the logic conventions updated automatically.

6.6.1 Top Level Simulation Design

The top level simulation design is presented in Figure 6-10. It consists of six main sections as described in Table 8 and laid out in Figure 6-10.

Table 8 - Section breakdown of Top Level Simulation Design

Section	Contents
1. Control Logic	Contains the logic which synchronizes the actions of other sections, failure and maintenance logic
2. MCC RM	Implements the MCC RM described in section 5.2 for the simulation.
3. Components	The collection of components, sub-assemblies, failure modes that impact the simulation
4. Supply Distribution	A three tiered distribution, consisting of warehouse, supply ship, and ship stores. Manages spares once they have been added to the supply system
5. Technicians	A waiting area for technicians until they are required to conduct maintenance
6. Spares	Manages the initial creation of spares and procurement of additional spares. Includes both regular order of spares and non-regular procurement (either for emergency or non-repairable systems)

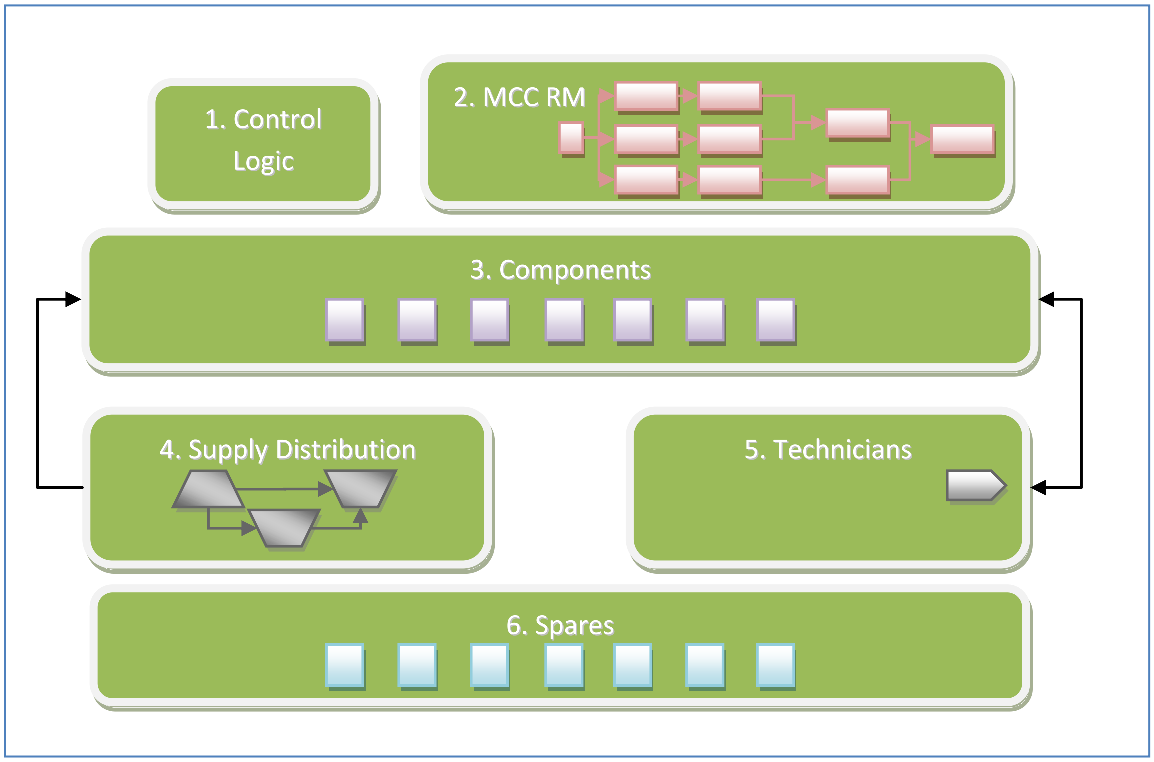

Figure 6-10 - Simulation Design Top Level Layout

6.6.2 Simulation Templates

The simulation consists of 'hard coded' logic[61] and repeatable usable templates specifically developed for this Thesis. The template items include 'component', 'spares', 'system', 'configuration tracker', and 'checkstate'. Each of these templates accelerates the setup of experiments, and automates model input.

6.7 Simulation Verification and Validation

Verification of the simulator was conducted in three ways: test runs with expected results, visual observation of logic while observing key variables, and analysis of output data.

The formal testing against expected results was useful in the early development of the simulation model. In this stage, the six sections outlined in Figure 6-10 were

---

[61] Logic specific to the experiment setup.

individually developed and tested formally before being combined. Formal testing also made use of Arena's entities to track simulation flow. Formal testing becomes more difficult as the size of the simulation grows.

Three areas of visual feedback were designed into the simulation. The components have a coloured circle to indicate their current state. The components also show images whenever a spare part and/or technician is present for CM. The spares system indicates spare levels, a spare plot, spares being procured, and spares used. The MCC RM indicates the current states of each component, and entity flow through the model shows when the entity is representing failed path or a working path. This feedback exists on every simulation run, and was used for verification and validation of the simulation.

The final method used for verification was output analysis. Logic errors in the simulation tended to have very obvious results in the output.

Validation of the simulator has been carried mainly through output analysis. Some of these details are in Section 7, where the experiments conducted are actually a form of validation.

6.8 Output Analysis

The key information required from the simulation is the ship mission availability, and the expected ship performance level. This important piece of data is not sufficient for causal analysis. Data analysis must occur at each level of the simulation (ship, capability, configuration, system, component) to provide traceability on results.

6.8.1 Overview

Mission availability is evaluated in the form of the average interval availability for a period of interest (equation 2.5) applied to multifunction ships (see Section 5.4.1). The three definitions include critical availability, mean capability availability, and mean weighted performance availability.

6.8.2 Methods

The primary methods of output analysis were MCF, availability plots, availability measures with confidence intervals, and probability distribution analysis. The SAS code used for output analysis is listed in Appendix A.

6.8.2.1 Mean Cumulative Function

The mean cumulative function (MCF) is a recommended method for comparing renewal cycles in maintenance (Nelson 2000) and can be applied to number of recurrences or accumulation of recurrence costs (i.e. repair costs or repair hours). Based on recurrences of the renewal cycle, this non-parametric method is used to estimate the mean (absolute) time of each recurrence. It can be applied to instances where recurrences are either independent or dependent on the previous recurrence. As Nelson notes, assumptions of independence between recurrences are "dubious". The MCF can be found in Weibull ++ and in SAS. In SAS, using Proc Reliability, confidence intervals can be generated for dependent or independent assumptions.

In (Nelson 2000), a method of comparing two MCFs is discussed. Essentially, the difference of the two MCFs is the statistic, and confidence intervals are generated for the difference. The comparison is graphical, and looks for time points where the curves are significantly different. This is different from other statistical methods compare all time points simultaneously and must content with the problems of multiplicity and Dunn-Bonferroni corrections. The purpose of Nelson's comparison is investigative, and noting a region where two MCF curves diverge may be indicative of some systematic problem, regardless if the overall difference is not statistically different.

The proper interpretation of the MCF used here is determining the mean time from start of an item working, until the end of its nth repair cycle. This is different from mean time to nth failure, though the data can be rearranged for this purpose.

The MCF is useful at the component level. At higher levels, it becomes less useful as a measure of number of recurrence cycles. For example, at the system level, the number of recurrences becomes ambiguous as the system may have component failures and repairs that do not result in system failure and thus do not result in a recorded recurrence. This ambiguity extends to the configuration level, and becomes more pronounced at the capability level if multiple configurations can provide for a capability. At the ship level, the ambiguity level jumps again, as there is a mixture of different capabilities/functions. Except for critical systems, the failure of a capability does not equate a failure at the ship level.

There is another approach that can be taken with MCFs. Instead of staying strictly in recurrence cycles as described, one can analysis strictly off of the accumulation of maintenance (CM, PM or both). While MCF based on recurrence provides insight into availability, MCF based on maintenance provides insight into the accumulation of repairs, repair costs, and repair hours.

In SAS, Proc Reliability can do MCF plots. Weibull++ also has a MCF plot folio.

6.8.2.2 Availability Graphs and Functions

There are two main types of availability evaluated, point availability and average interval availability (see section 2.2.1). The downtime in the simulation comes from two sources: maintenance actions (CM, PM, and associated delays) and system inactivity. With the combination of these elements, availability from this simulation can also be expressed as operational availability / unavailability (section 2.2.4). When focusing on the system performance independent of external causes of inactivity, then the availability can be referred to as achieved availability (section 2.2.3).

Availability plots were generated using SAS Proc GPlot. SAS Data step, and Proc Means were used to generate the data for the plot. Maple and Excel can also do availability plots.

6.8.2.3 Failure and Maintenance

Analysis of failure and maintenance times can be conducted using the standard tools found in Elsayed (2011). SAS Proc Reliability, Proc LifeTest, and Proc Lifereg support parametric and non-parametric analysis of survival and reliability data.

6.8.2.4 Discrete Fourier Transform

The Fourier transform is used in studying periodic behaviour in various disciplines. Renewal cycles represent a type of periodic behaviour. Fourier analysis was attempted on unavailability data and recurrence data simply to see what information is exposed using this tool. The expectation was that the more regular a failure, it should have a distinctive frequency signature under a Fourier transformation. Fourier analysis was not the main analysis for this Thesis.

An example is given in Figure 6-11 and Figure 6-12. The example system fails every 90 hours and takes 10 hours to repair. The availability representation of the data is essentially a heaviside (rectangular response) function in the time domain, and a sinc function in the frequency domain (see Figure 6-12 for an example). Though it does not appear to do so, the availability analysis actually shows the same information as unavailability (in both complex and frequency plots!; the complex plot is flipped across the imaginary axis), but this is not clear due to the w=0 point in the availability analysis dramatically changing the scale. This occurs because the system is working most of the time, thus the discrete transform accumulates most of the transformation information at w=0. Fourier analysis might be more revealing of regular periodic maintenance than of random failures.

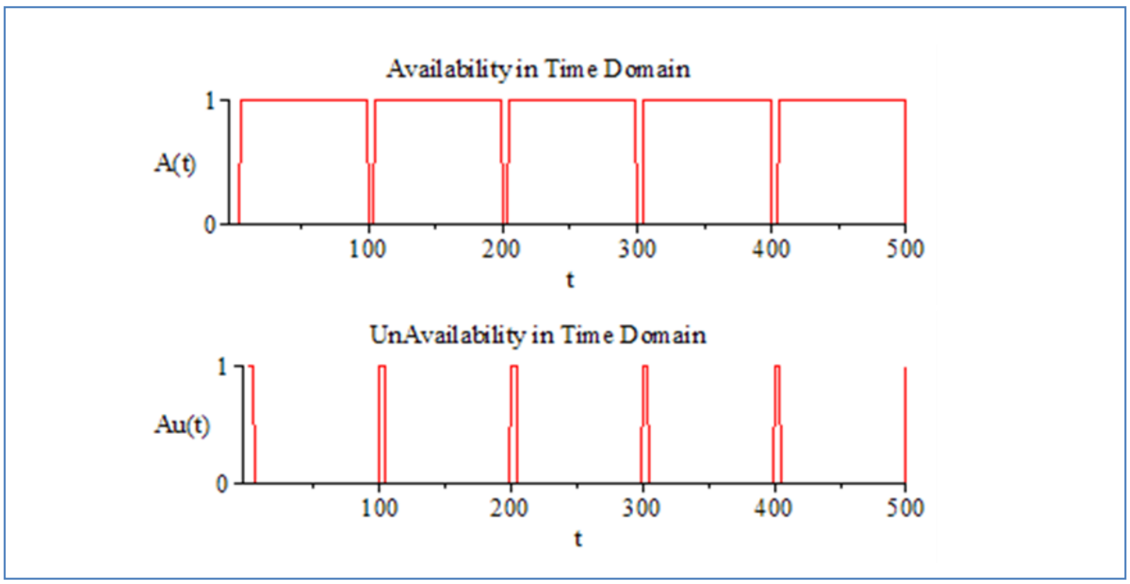

Figure 6-11 - *A*(*t*) and *Au*(*t*) for a periodic repair process (T=100 h, t=10 h)

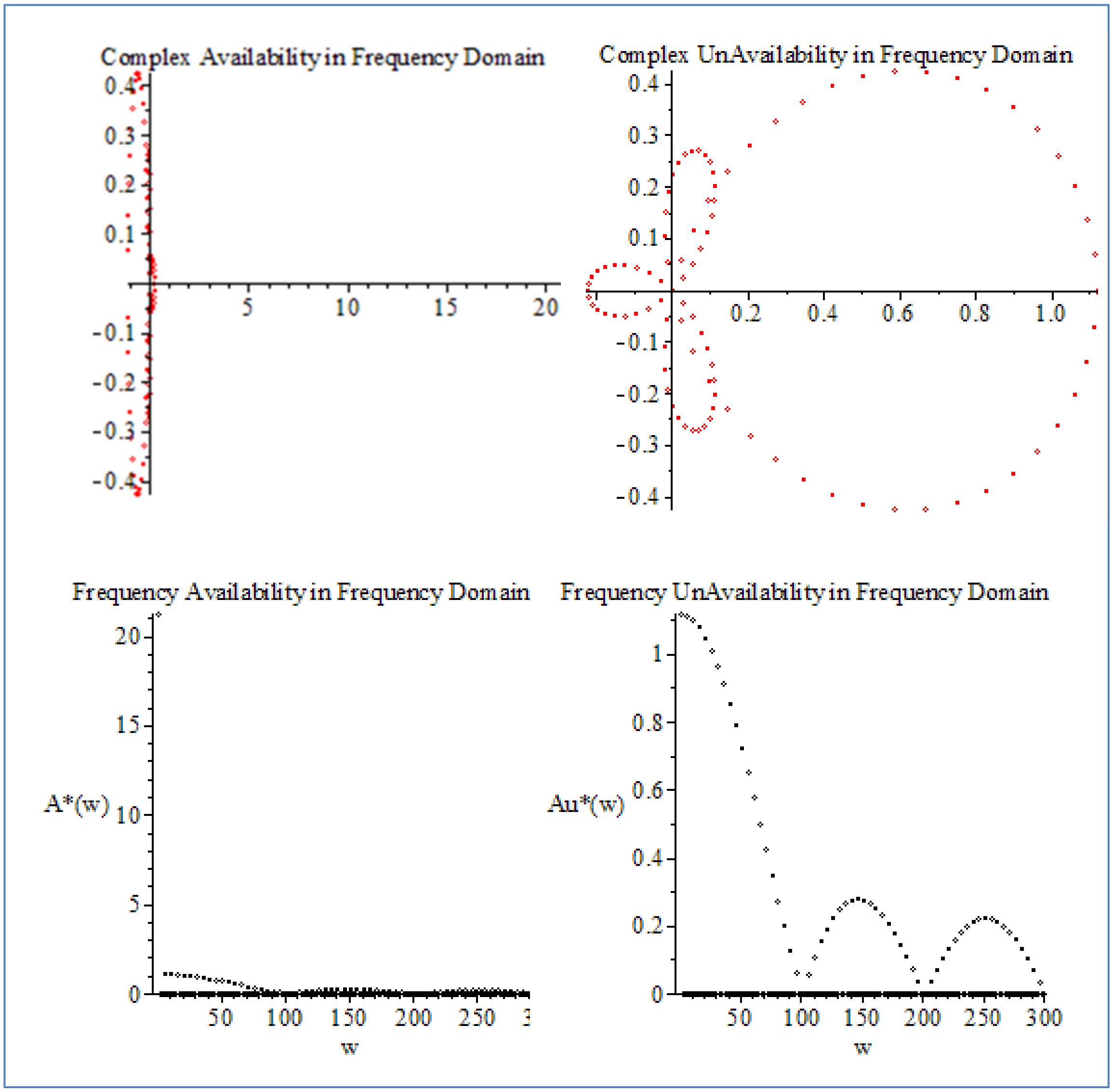

Figure 6-12 - Complex and Power Graphs of Discrete Fourier Transform of *A*(*t*) and *Au*(*t*)

6.8.3 Tools

For output analysis, four tools were utilized: Arena, Excel, SAS, and Maple. There is crossover in the capabilities of each tool. The selection of the tool for output analysis was determined by convenience. Output analysis was conducted in the tool that made the overall process the simplest and most rigorous, by taking advantage of pre-existing procedures and its ability to handle large data sets (more than a million data points). This tool was SAS.

6.8.3.1 Arena

The simulation software, Arena, has the means to process some analysis, especially given its access to VBA. Some of the simulation output consists of first level of output analysis. This includes determining configurations in use, and capability performance weights, and the ship's weighted performance values (see sections 6.4.5 and 6.4.6 respectively). These values were evaluated in Arena since they are based off of decision logic implemented in the simulation.

Arena also has a built-in reporting system for module statistics. Except for purposes of verification, these have been bypassed by processing results through external programs.

6.8.3.2 Excel

Initial simulation data output is written directly to an Excel spreadsheet. Some analysis can be done using Excel; it was found more convenient to do the analysis using SAS avoiding algorithm development for procedures that already exist. Due to nuances of how SAS works, the Excel spreadsheet must be converted to the .xls format; VBA outputs the .xlsx modern format. Alternatively the data can be converted to a data/text file, if there are more data points (in the order of 65000+)than a .xls format worksheet can hold. While technically a step in the output analysis, the Excel file conversion and initial input into SAS are assumed in sections 6.8.4 through 6.8.8.

6.8.3.3 SAS

SAS was the main tool for analysis. It handles large data sets, extracts grouped data automatically, and has both general and specialized procedures useful for availability analysis. Of key interest were Proc Reliability, Proc Lifetest, and Proc LifeReg procedures; these procedures are used in survival/failure analysis and failure/repair recurrence analysis. Other very useful procedures were the general Proc Means, Proc Univariate and Proc GPlot useful for empirical availability analysis.

6.8.3.4 Maple

Maple was used as a general math tool and to investigate discrete Fourier analysis of the unavailability information. It was also used in non-simulation research.

6.8.3.5 Weibull ++

While not used explicitly in the output analysis, Weibull++ was useful for providing an alternative method of checking results, and validating the simulation.

6.8.4 Component Level Analysis

Chronological tracking of up and down times for components causes interpretation problems and comparison problems. There are downtimes for components which correspond to neither CM actions nor logistic support actions. An example would be a component in a system which fails but the system is not restored by end of the mission. The component may have stopped accumulating run hours at 100 hours, yet the mission ended at 1000. The question is does one censor at 100 hours or 1000 hours. For the analysis in this thesis, to resolve this ambiguity censoring is taken at 100 hours; there are no observed values later for that equipment.

With run time analysis, there is no similar ambiguity as the chronological censoring. If a part only runs for 100 hours, that that is its censoring time, and this can be compared unambiguously

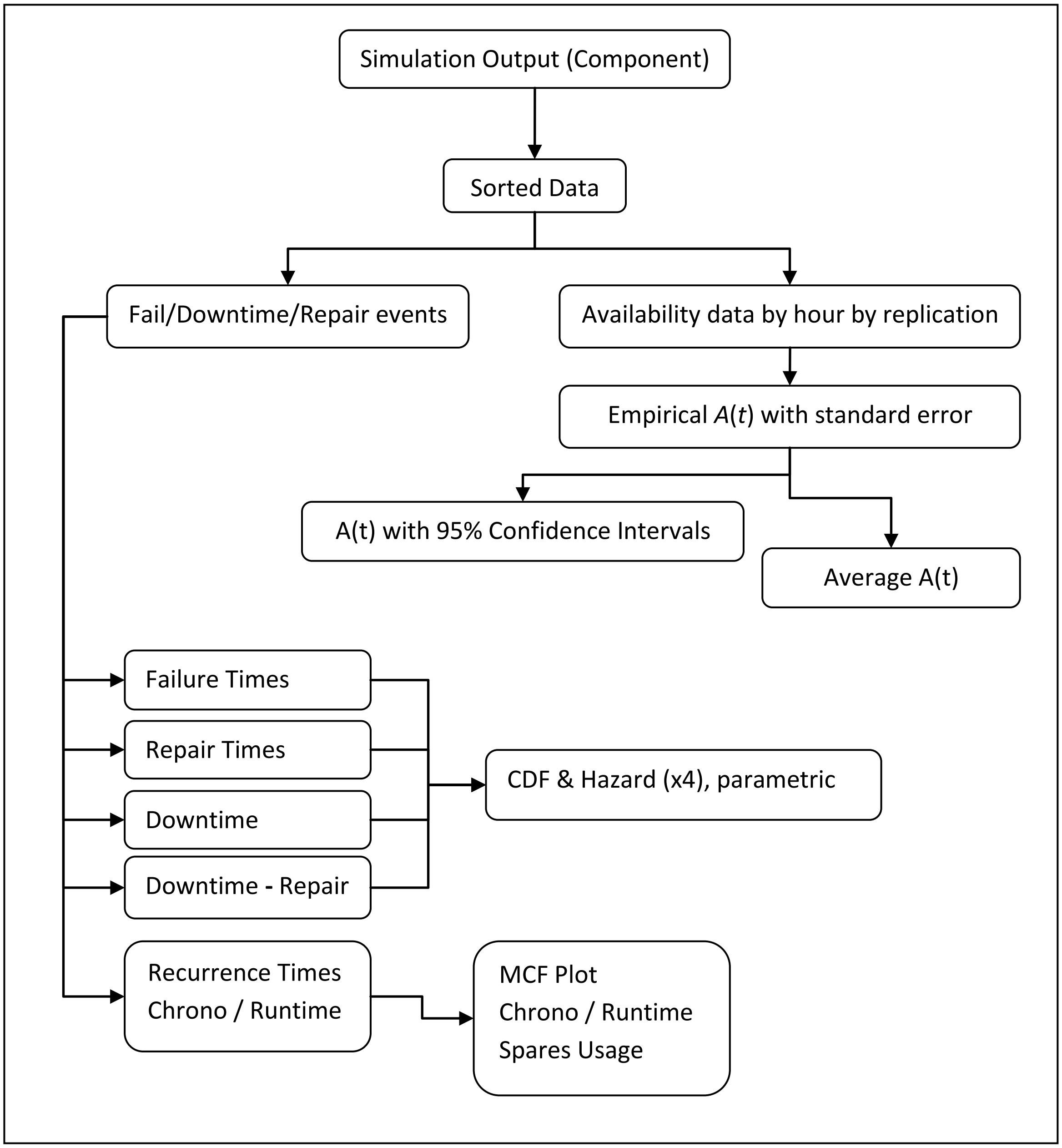

Figure 6-13 - Component Output Data Analysis Map

Figure 6-13 maps the output analysis using SAS. The SAS procedures used were:

a) Data output to Sorted Data. Proc Sort;

b) Sorted Data to Availability by hour/replication. Data step;

c) Availability by hour/replication to Empirical $A(t)$. Proc Means;

d) Availability by hour/replication to $A(t)$ with CI. Proc Means or Data Step;

e) Availability by hour/replication to Aver gate $A(t)$. Proc means or Proc Univariate;

f) Sorted Data to Fail/Repair/Downtime events. Data Step;

g) Fail/Repair/Downtime events to Recurrence Times. Data Step;

h) Fail/Repair/Downtime events to Failure, Repair, Downtime and Downtime less Repair times. Data Step;

i) Recurrence Times to MCF. Proc Reliability; and

j) Failure, Repair, Downtime and Downtime less Repair times to Parametric and Non-Parametric analysis. Proc Reliability, Proc Lifetest, Proc Lifereg.

6.8.5 System Analysis

Figure 6-14 maps the output analysis using SAS. The SAS procedures used were:

a) Data output to Sorted Data. Proc Sort;

b) Sorted Data to Recurrence Times. Data Step;

c) Recurrence Times to MCF. Proc Reliability;

d) Sorted Data to Availability by hour/replication. Data step;

e) Availability by hour/replication to Empirical $A(t)$. Proc Means;

f) Availability by hour/replication to $A(t)$ with CI. Proc Means or Data Step; and

g) Availability by hour/replication to Aver gate $A(t)$. Proc means or Proc Univariate.

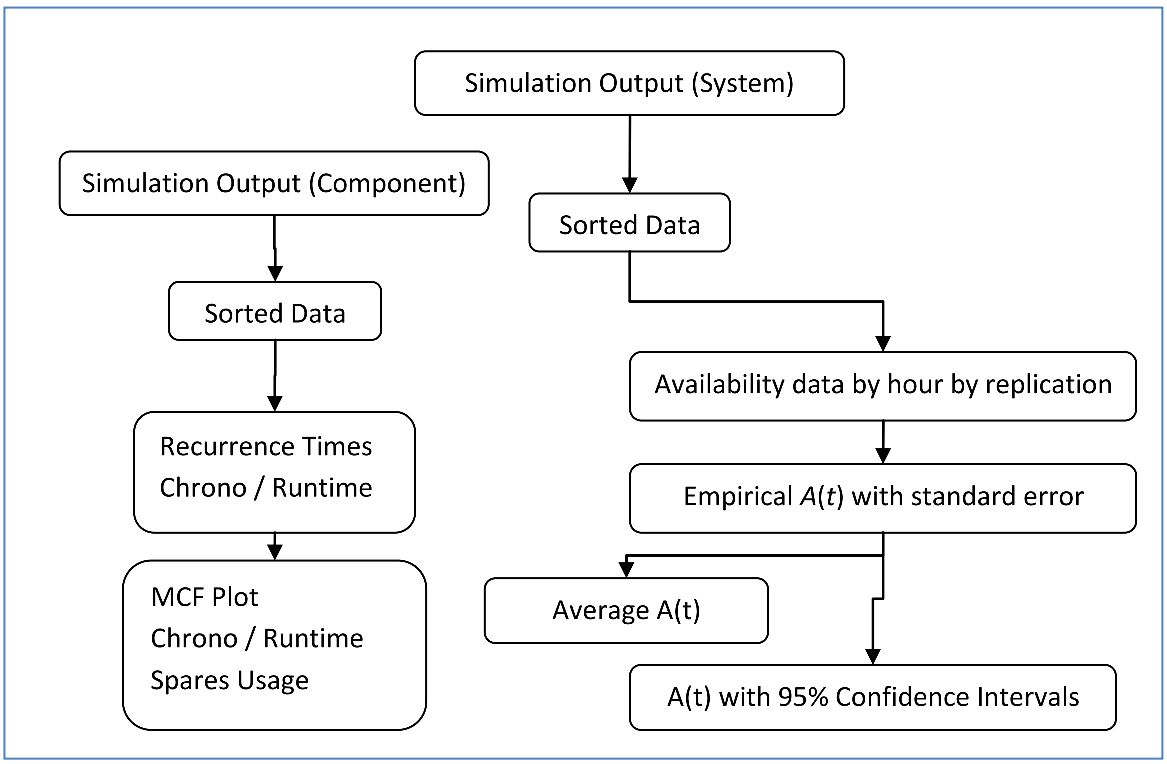

Figure 6-14 - System Level Output Analsys Map

6.8.6 Configuration Analysis

Figure 6-15 maps the output analysis using SAS. The SAS procedures used were:

a) Data output to Sorted Data. Proc Sort;

b) Sorted Data to Availability by hour/replication. Data step;

c) Availability by hour/replication to Empirical $A(t)$. Proc Means;

d) Availability by hour/replication to $A(t)$ with CI. Proc Means or Data Step; and

e) Availability by hour/replication to Aver gate $A(t)$. Proc means or Proc Univariate.

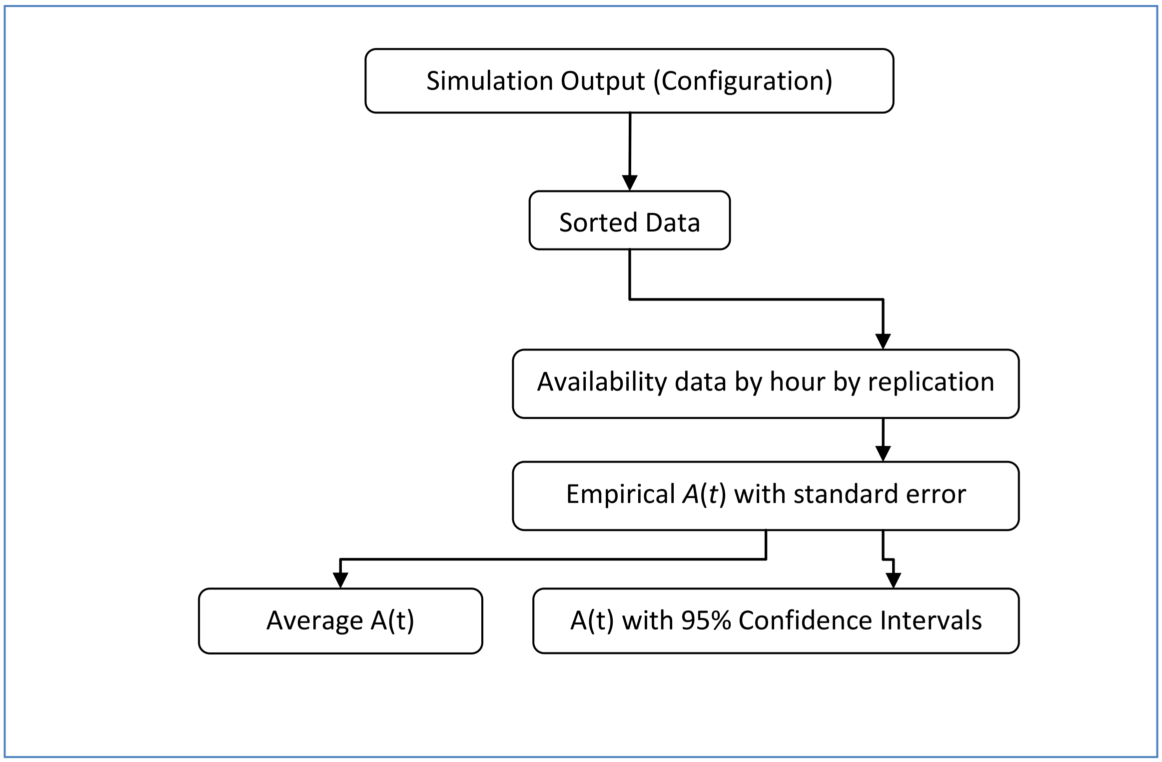

Figure 6-15 - Configuration Output Analysis Map

6.8.7 Capability Analysis

Figure 6-16 maps the output analysis using SAS. The SAS procedures used were:

a) Data output to Sorted Data. Proc Sort;

b) Sorted Data to Availability by hour/replication. Data step;

c) Availability by hour/replication to Empirical $A(t)$. Proc Means;

d) Availability by hour/replication to $A(t)$ with CI. Proc Means or Data Step; and

e) Availability by hour/replication to Aver gate $A(t)$. Proc means or Proc Univariate.

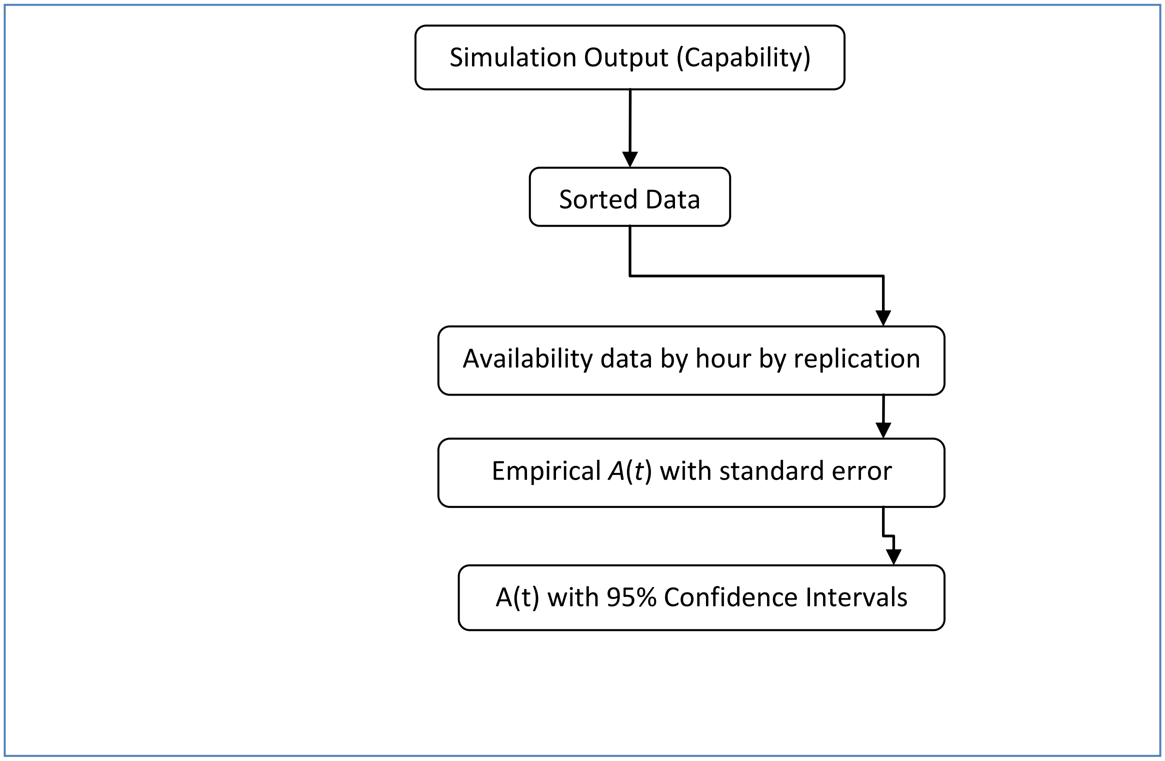

Figure 6-16 - Capability Output Analysis Map

6.8.8 Mission Level Analysis

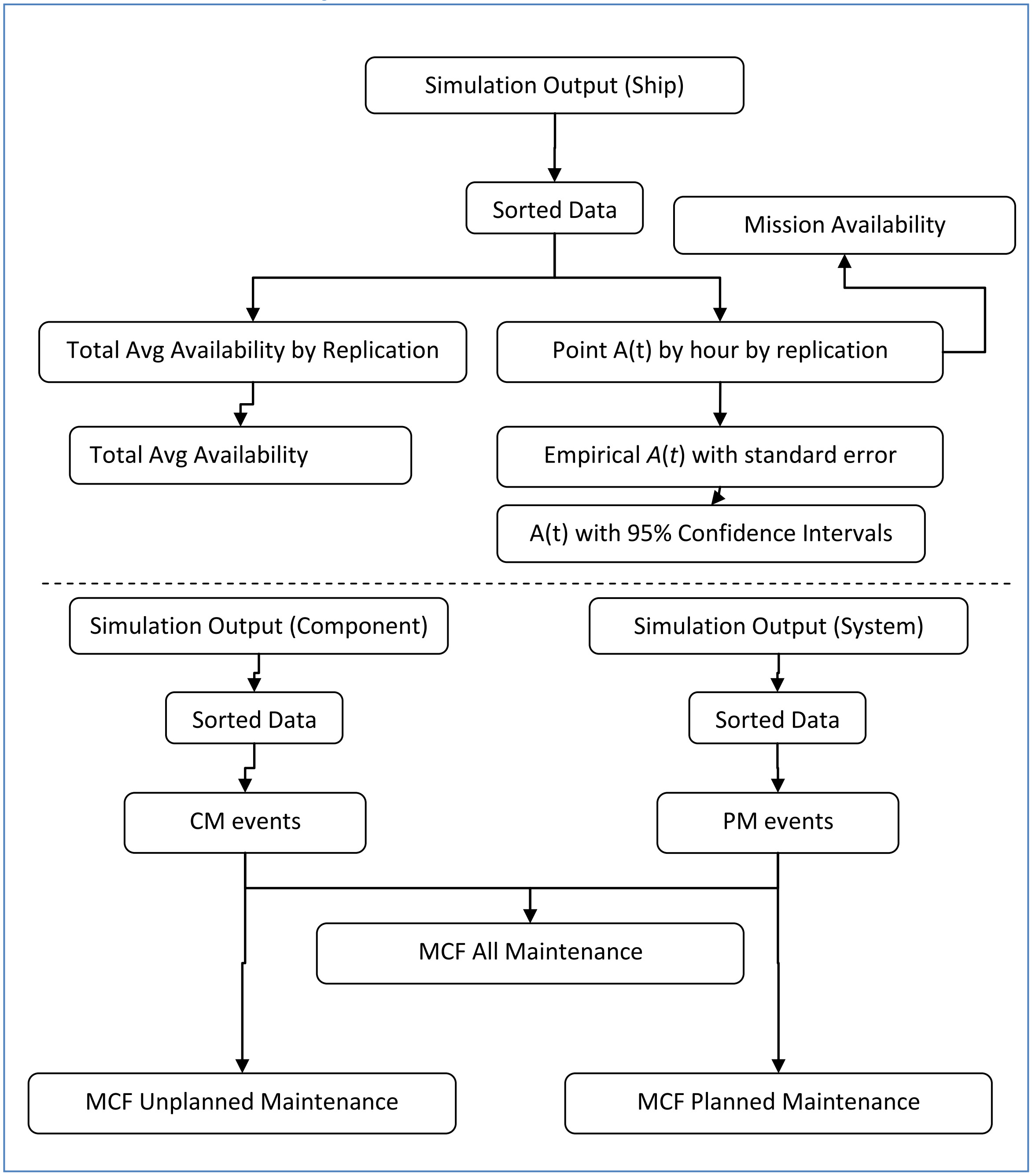

Figure 6-17 - Mission Level Output Analysis Map

Figure 6-17 maps the output analysis using SAS. The SAS procedures used were:

a) Data output to Sorted Data. Proc Sort;

b) Sorted Data to CM, PM times. Data Step;

c) CM, PM times to MCF. Proc Reliability;

d) Sorted Data to Availability by hour/replication. Data step;

e) Availability by hour/replication to Empirical *A*(*t*). Proc Means;

f) Availability by hour/replication to *A*(*t*) with CI. Proc Means or Data Step; and

g) Availability by hour/replication to Mission Availability. Proc Means (using 'where')

h) Sorted Data to Total Average Availability By Replication. Proc Means; and

i) Total Average Availability By Replication to Total Average Availability. Proc Means.

6.9 Simulation Experimental Design

Each simulation experiment, referred to as 'simulation scenarios', is defined by:

a) purpose;

b) scenario design. This refers to MCC RMs and the actual models used at each level;

c) the simulation inputs to the scenario design; and

d) the subset of output analysis of interest and/or modified output analysis required.

7 Phase IV - Simulation Results

The flexibility of the simulation model is demonstrated through the use of various simulation runs. Topics of interest were the implementation of failure, CM, PM, IM, spares models, and phased missions.

7.1 Overview of Default Scenario

A default ship configuration has been established for Section 7 and 8. Table 5 provides the combination of systems, configurations, and capabilities featured in this default scenario. The default scenario has six capabilities, nine configurations, 12 systems, and 38 components entangled with non-trivial relationships. Further, performance values for configurations are assigned in Table 9. Five of the distinct capabilities provide multiple distinct performance levels. Only long range surveillance has performance defined by an indicator level. While this default scenario of a 'complete ship' is used throughout Section 7, when necessary focus is given only to a component or system of interest[62].

Table 9 - Capability Performance Values by Configuration

Capability	Configuration	Performance Value
Move (Critical)	Main Propulsion	1 for both working gas turbines .7 for only one working gas turbine
PG&D (Critical)	Diesel Generators	2 of 3:G -> Value of .4 if only one generator, otherwise full value of 1 3 of 3:G -> Value of 1 for three generators, .7 for two generators and .2 if only one generator
Long Range	Long Range Radar	1

[62] Keeping the component studied entangled with the default scenario provides insight to those interactions.

Capability	Configuration	Performance Value
Surveillance		
Medium Range Surveillance	Long Range Radar	.4
	Medium Range Radar	1
Navigation	Navigation Radar	1
	Secondary Navigation Radar	
AAW	Close In Defence	.4 accumulative with other AAW configurations
	Missile	.35 accumulative with other AAW configurations
	Gun	.25 accumulative with other AAW configurations

The mission for the default scenario is defined by an operational period of 2 years with a five month deployment as per section 6.2.3. Originally the intention was to run the simulation with 100+ replications, taking 1.5+ hours per execution; the software could not handle the full data output transfer, and the maximum replications was reduced to 50. This simulation setup is smaller than a full scale simulation, necessitated by practical considerations. Further, any experiment not requiring the full execution was suitable modified. The time resolution for this simulation is hourly; this is the most reasonable time frame for measuring maintenance actions, failures, and down times. Consideration was given to simulating at a daily or weekly time scale, however, the loss of resolution makes it harder to relate the events to real life (validation).

For single component studies, the diesel generators from the Power Generation and Distribution system were studied. The three generators have a multiplicative effect on spare demand. The generators are also critical to the ship, making the ship sensitive to problematic changes to these components.

The generators are themselves systems, and would not normally be considered components. Flexibility in the simulation allows representation of single failure mode items as either components or systems. When desirable, the generators can be detailed as multiple component systems.

7.1.1 Sparing Levels

Initial sparing in the system was set at 20 spares, with 10 onboard ship, 4 onboard the support ship, and the remainder in a warehouse. Resupply was set at an order quantity of 10 every 1900 hours. Upon excessive depletion of stock, the simulation can be programmed to automatically order additional supplies.

A small experiment was run to compare ship availability given different supply policies for the generators. Order quantity and order intervals were changed to demonstrate the problems of shortages.

Table 10 - Some experiments in Sparing for Generators

Initial Values	Results	Achieved Ship Availability (Two year Simulation run length)[63]
Initial Spares =20 On Ship=10 In TG=4 Order Quantity=10 Order Interval =1900 (Baseline)	Spares in System =80 Spares being procured = 10 Spares Used=30	.921861

[63] Since this is a single replication, there is no stderr that can be estimated for the total result. The reason why some simulation runs have exactly the same availability measured, is because they are all run on the same random value seeds, and the changes in spares did not impact even a single event affecting availability.

Initial Values	Results	Achieved Ship Availability (Two year Simulation run length)[63]
Initial Spares =20 On Ship=10 In TG=4 Order Quantity=5 Order Interval =1900	Spares in System =34 Spares being procured = 5 Spares Used=31	.921861
Initial Spares =20 On Ship=10 In TG=4 Order Quantity=2 Order Interval =1900	Spares in System =9 Spares being procured = 5 Spares Used=31	.921861
Initial Spares =20 On Ship=10 In TG=4 Order Quantity=2 Order Interval =3800	Spares in System =0 Spares being procured = 4 Spares Used=28	.886244
Initial Spares =10 On Ship=10 In TG=4 Order Quantity=2 Order Interval =3800	Spares in System =0 Spares being procured = 6 Spares Used=22	.607877

The results from Table 10 show that the initial sparing values were very conservative, and that the generators were having minimal negative impact on the ship availability. In this situation, achieved ship availability cannot be improved further by increasing the addition of spares into the system. Removing spares still has the normal detrimental effect.

Sparing policy for a ship could be set through various models such as economic order quantity. Using the basic deterministic EOQ model (Nahmias 2009):

$$Q^* = \sqrt{\frac{2K\lambda}{h}} \tag{7.1}$$

where Q^* is the EOQ, K is the fixed cost per order, λ is the demand rate, and h is the holding cost. This corresponds to the basic ordering method built into the simulation. However, when using this and other related models, one must be aware of certain factors.

If penalties are used, then the loss of ship[64] might cost around $1 Billion. The implication from a penalty this large is that one should order an excessive number of spares. Availability considerations are not constructed into the EOQ model.

A simple method to set practical spare levels for the simulation, without analytics, is to run the simulation, track the spares, and adjust. Figure 7-1, has two examples of sparing results from the simulation. The left 'spare' is based on baseline sparing levels described in this section. The spares increased from 20 to 80 in the system, and only 30 were used in repairs. There were two extra backup spares backordered. In the figure's right 'spare', the generators reduced initial spares to nearly zero. Less spares appear to have been used, until you look at the availability in graphs like Figure 7-2 and Figure 7-3. In these two graphs, one can see the times that the power system suffered while waiting for spares[65]. Thus, in the latter case, some remedy is required. While increasing the total spares could alleviate the problem, it is a spare replacement rate deficiency, thus the ordering quantities and/or order interval needs modification.

A simple heuristic for establishing the order quantity and interval from Figure 7-1 is to divide the number of used spares by the simulation run length. This is a very basic consumption rate. Then replace half of the ship's initial sparing at an appropriate time (setting the safety stock at approximately half). In this example, the ship can hold 10 spares, thus we replace 5 spares every $\frac{17520}{30} * 5 = 2920\,\text{h}$. When setting up the initial

---

[64] And people!

[65] Remembering that in the absence of power almost all capabilities are lost, the results may look confusing. The reason why power can be critically failed, yet capabilities not failed is that power critically fails when there is less than two generators. When there is only one generator, there is still power available. A refinement to this model would have particular systems shutdown when power is insufficient, according to a priority.

simulation, it was found useful to run the simulation first with 'infinite' spares, and then use the spares usage information to set ordering policies based on this heuristic.

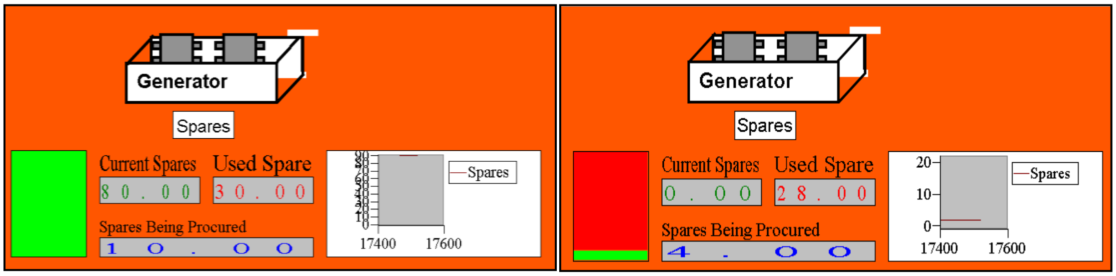

Figure 7-1 - In Simulation Spare Tracking (a) Effective spare management (b) Ineffective Spare management

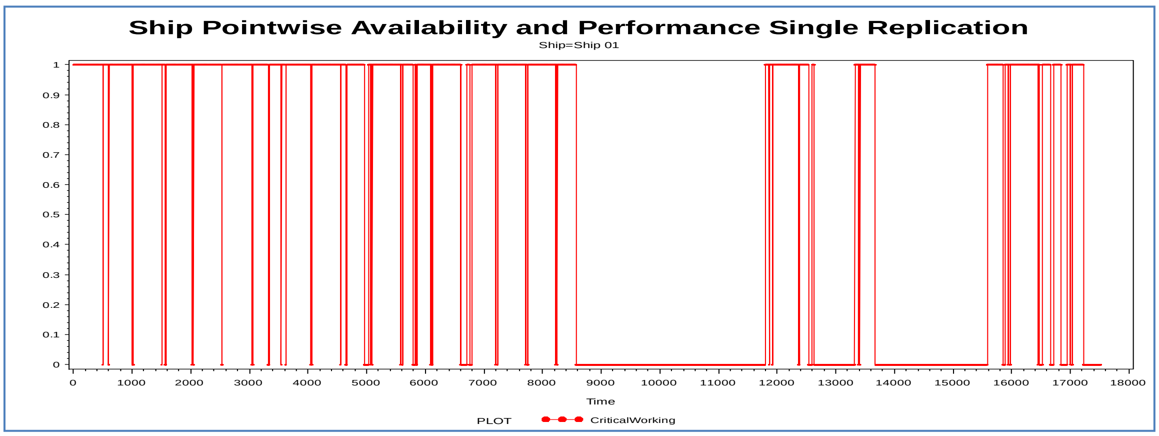

Figure 7-2 - Effect on Critical Capability Availability from poor spare management

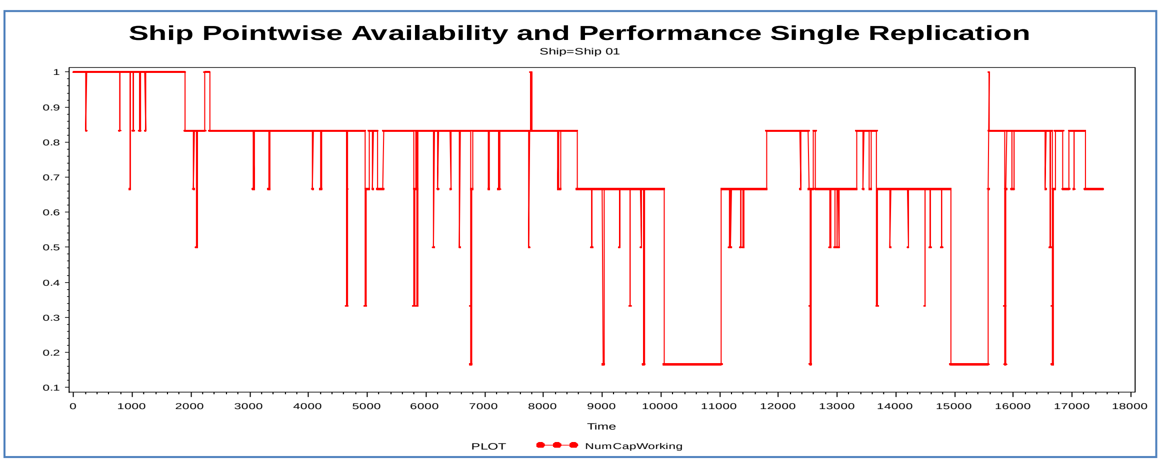

Figure 7-3 - Effect on Mean Capability Availability from poor spare management

Any spare optimization must take into account, ordering costs, holding costs, space limitations, supply distribution limitations, impact on availability, and demand (engineering failures). These are considered normal considerations but 'optimizing' for them can result in subpar performance. The simple reason is that the platform discussed in this Thesis is a warship, and survivability considerations must be taken into account. Consider a system that rarely fails, easy to repair, and spares can be made available quickly through a responsive supply distribution system. Optimizing across the ship might suggest not carrying a spare part for this system (perhaps instead it is carried on the support ship). However, if this system is important when survivability issues arise, then spares may be needed on board to respond immediately to a perilous situation. The ship is designed conditional on those environments and spare policies are adjusted accordingly within resource limitations.

7.1.2 Failure and Maintenance Parameters

Adjusting failure parameters has the effect of changing spares demand, and increasing downtime. To show the affect on ship availability, again the three power generators were modified. In this experiment, ship availability was assessed on both critical capabilities and mean capabilities.

Initial sparing levels for the generators were the same as Section 7.2.1, and the reorder was 5 spares every 2920 hours. This reordering policy was calculated using a heuristic in Section 7.2.1.

Table 11 - Some Experiments in different Failure Parameters

Initial Values	Generator Availability (Avg of all three)	Power Capability Availability	Achieved Ship Availability Critical Capabilities	Achieved Ship Availability Mean Capabilities
Weibull(1680,1.2) (Baseline)	.7193	.9599	.8331	.8063
Weibull(1680,1.8)	.7370	.9610	.8316	.8085
Weibull(1680,2.4)	.7031	.9661	.8386	.8122
Weibull(1680,3.0)	.6704	.9748	.8513	.8169
Weibull(840,1.2)	.6637	.7550	.6126	.7067
Weibull(1260,1.2)	.6650	.9757	.8428	.8182
Weibull(2520,1.2)	.7332	.9630	.8274	.8088
Weibull(3360,1.2)	.8087	.9558	.8260	.8025

The MTTF of the Weibull distribution is:

$$\eta \cdot \Gamma(1+\frac{1}{\gamma}) \tag{7.2}$$

where η is scale parameter, and Υ is the shape parameter. Either increasing η or decreasing Υ results in an increased MTTF. If failures are increased, then there are three effects which can systematically decrease availability as shown in Figure 7-4. The results Table 11 portray only random effect on the availability values except in the one case where failures increase sufficient to causes spare shortages. With sufficient replications and careful design of experiments, the expected trend may be seen; this is less than trivial as it may require isolation from the variability of other systems simultaneously running. The variability of overlap of downtimes between generators also affects the results due to the parallelism of their system design. This table is informative though, suggesting that the greatest impact on availability comes when insufficient resources exist to repair/replace a system.

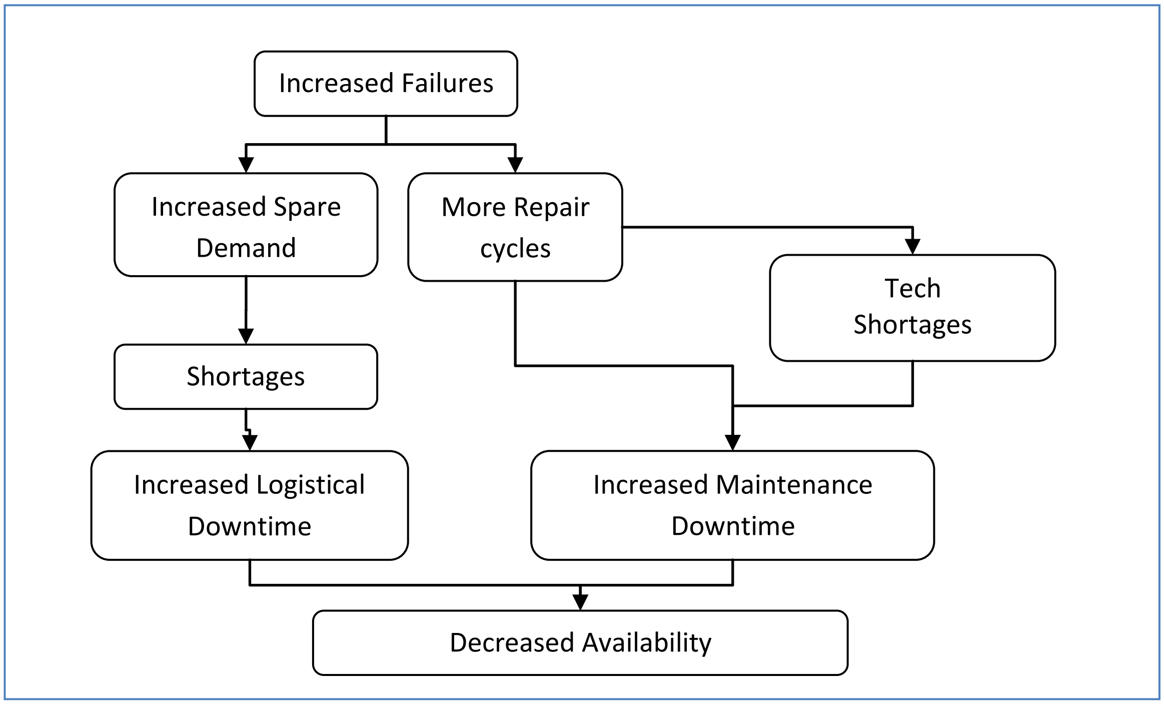

Figure 7-4- Cause and Effect between failures and availability

7.1.3 Imperfect Maintenance

Having imperfect maintenance has the result of increasing the number of failures of a component or system. This is because the component does not completely renew, and from the repaired state, its MTNF (next failure) is typically less than that of brand new. As noted in Section 6.2.4, a 'p-q' imperfect maintenance model was implemented in the simulation. The *p* value is modeled as the product of a difficulty factor, and technician skill, as given in equation (6.1).

Once again, the generators are used for this experiment. Generators 01 and 02 are kept at *difficulty* = 1 for comparison to Generator 03 at with *p=.45* (*difficulty*=0.5 and tech's *skill*=.9) [66]. The simulation was run at 10 replications, and run length was 17520 hours. By fitting a Weibull distribution to the resulting failure times in the run time scale,

[66] Here *difficulty* = 1 means the item is usually restored to a new state (depending on technician skill), while *difficulty* = 0 means the item is restored always to 'as-bad-as-old'.

Table 12 show that generators 01 and 02 perform within expectations. Generator 03 performs equivalently to a Weibull distribution of reduced characteristic life and shape parameter. Interestingly, the fit for Generator 03 closely resembles an exponential distribution (an exponential failure distribution is not affected by 'p-q' IM model).

Table 12 - Results of applying 'p-q' IM model

Generator	Simulation Distribution	SAS Distribution Proc Reliability
#01	Weibull(1680,1.2)	Weibull(1704,1.134) Scale Stderr = 154 Shape Stderr = .0853 Failures = 101 Censored = 8 Mean = 1628
#02	Weibull(1680,1.2)	Weibull(1882,1.178) Scale Stderr = 170 Shape Stderr = .0982 Failures = 93 Censored = 10 Mean = 1779
#03	Weibull(1680,1.2) on a 'p-q' IM policy with *p*=.45	Weibull(1460,1.0257) Scale Stderr = 139 Shape Stderr = .0741 Failures = 113 Censored = 8 Mean = 1445
Critical System Availability = .8409, stderr=.025572		

As a side note, one would normally expect 10 right censored values from 10 replications. However, if a component repaired into an idle state until the end of the simulation run (typically due to maintenance being conducted elsewhere in the system), the right censored value will not occur.

Since Generator 03 is no longer generating independent failure times for each cycle, it makes sense to check its distribution on a Weibull probability plot to check for appropriateness. Figure 7-5 shows the probability plots for Generators 01[67] and 03 respectively. The behaviour is very unusual, Generator 01 does not appear to be a proper Weibull distribution, yet Generator 03 does. Generator 01 does have straight line appearing, but left tail outliers appear to be offsetting the expected line. Using Weibull++ distribution wizard, it still selects Weibull 2-parameter as the best fit.

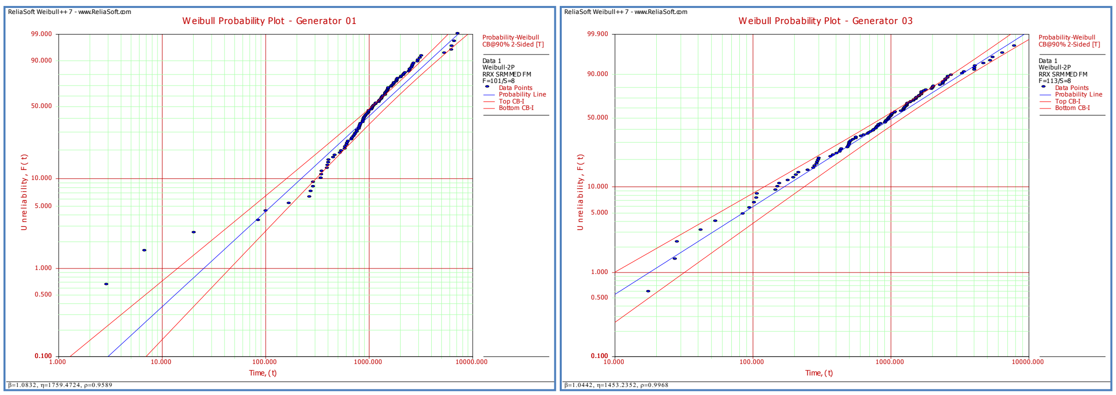

Figure 7-5 - Weibull Probability Plots - (a) No IM (b) IM, using Weibull ++

To check the observed result from Arena's random distributions, successive Weibull output data was collected from Arena and fitted using Weibull++. The results are shown in Figure 7-6. Two methods were used, the built-in Weibull distribution function, and the Uniform distribution modified for generating Weibull values. The modified Uniform appears more Weibull like, but this is hardly a conclusive test with only about 40 data points each and a single random seed value. Weibull++ selects Weibull 3-parameter distribution for Figure 7-6 (a) and Generalized Gamma distribution for Figure 7-6 (b). Random number generation is a difficult challenge. Arena provides the capability to the user to select their own seed for each instance of random number generation. Controlling this seed, in simulation, allows repeatability of exact results. The observations made here, suggest checking random number generators to ensure they

67 Generator 02 had similar results to Generator 01.

are producing appropriate random results; the importance of this suggestion is the need for repeatability when assessing contract availability using real failure data, as discussed in Section 8.

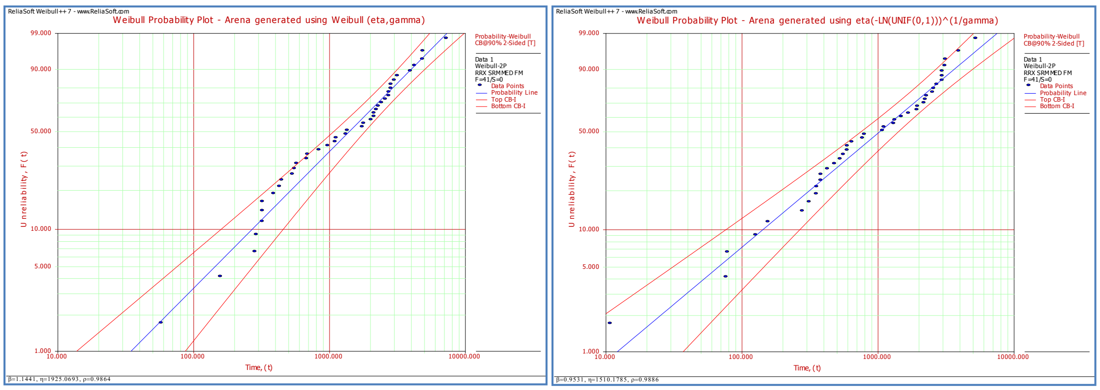

Figure 7-6 - Probability Plots - (a) Arena's built-in Weibull (b) Arena's built-in Uniform

7.2 Full Ship Homogeneous Operations

Under homogeneous operations the default scenario is used without applying phased mission. The ship is considered to be under 'high or standard readiness' and sailing for the duration of its operational period.

7.2.1 Preventative Maintenance

If PM requires a system to be temporarily brought offline, then it contributes to the unavailability of the system. Normally, it is hoped that this downtime is offset by increased reliability of the system or that the downtime is shifted to a more convenient time with less impact on productivity. In an extreme case, inappropriate PM can suffocate the ship as seen in Figure 7-7.

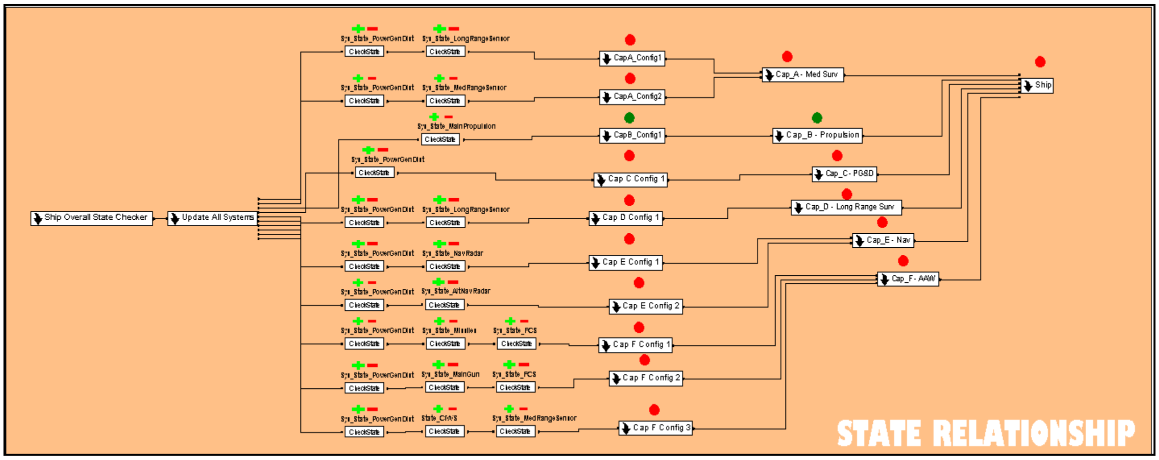

Figure 7-7 - Effect of Power Distribution Preventative Maintenance on MCC RM

In Figure 7-7, we are looking at the simulation's implementation of the MCC RM. On the far right we have the ship state. Moving left, a column of the six capabilities. Then a column of configurations. Finally, the set of boxes with "+-" are the systems (note the +- is merely decorative)[68]. The green and red circles indicate whether a MCC item is available at that point of time. This figure shows all but one capability and configuration as being disabled. The reason is, during this simulation run, Power Generation and Distribution was assigned a PM schedule. Every 2000 hours, all power was lost while maintenance is performed. There was a similar PM schedule on the other critical capability, Propulsion (which doesn't affect any other capability). Figure 7-8 and Figure 7-9 show the result of this PM on ship point availability for critical capabilities, and mean capabilities. When Power Generation PM is carried out, the ship is, in effect, dead.

[68] The remaining boxes are for control, synchronization, and processing all system states before evaluating the MCC RM. The systems in the MCC RM are simply references to the current system state.

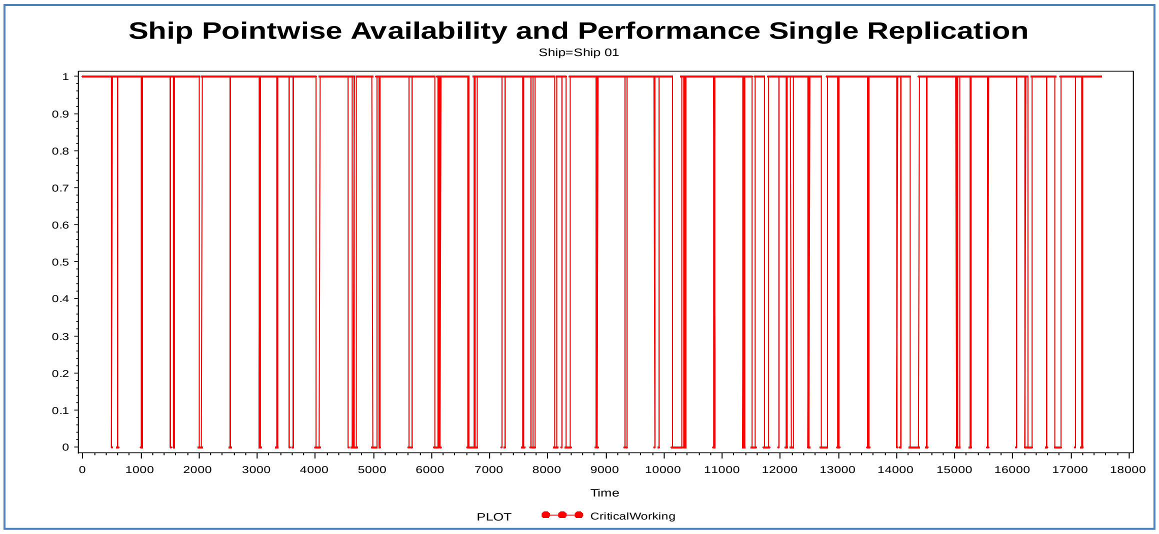

Figure 7-8 - Extreme PM effect on Critical Component Availability

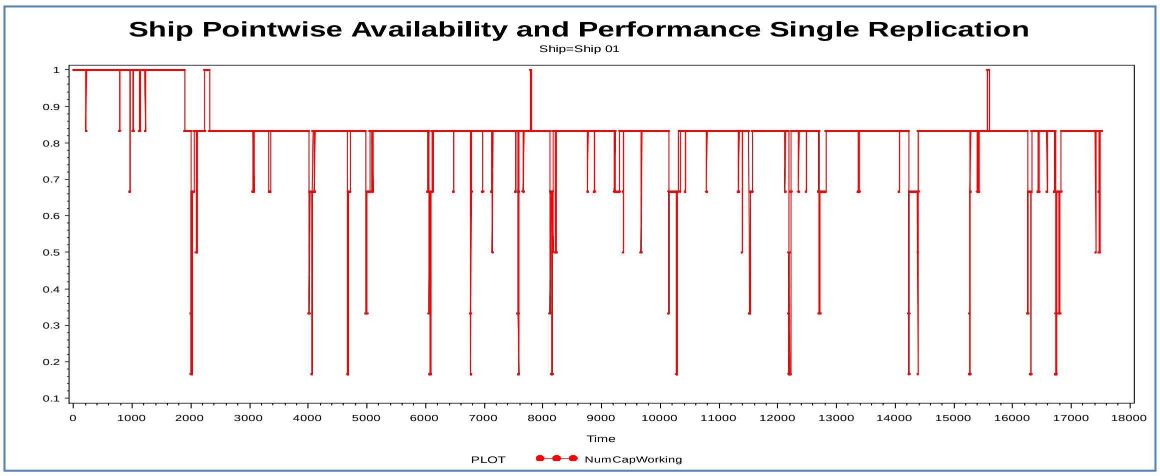

Figure 7-9- Extreme PM affect on Percent of Working

These types of problems can be seen during the simulation by observing the visual feedback. If not spotted during the simulation, then they are extremely obvious during output analysis. The simulation can be used as a guide for adjusting PM policies.

7.2.2 Assessing Mission Availability

Mission Availability is the interval availability over the mission period. For a simulation run without phased missions, Figure 6-2 is adapted by considering a mission as a 150 consecutive day (3600 h) of interest. Since the ship's MCC RM does not change in this time period, this has the interpretation that the ship must be in a state of 'high or standard readiness' throughout its entire operational period, and a particular 150 day period as the mission/deployment.

For this experiment, the default scenario was used with the two year operational period and 50 replications[69]. Mission availability was assessed on an 150 day interval at three distinct time periods. The first mission started at t_m=1. The middle mission started at t_m=6960. The final mission started at t_m=13920. This could be looked at as three missions in an single operational period; in this Section, the three mission periods are compared as different placements of the mission/deployment. There is no PM scheduled on critical capabilities scheduled in the default scenario (see Section 7.2.1). Two simulation runs were generated, the first without IM, and the second with an IM model (system difficulty of .8 and technician skill of .9).

Comparing the results in Figure 7-10 and Figure 7-11, it appears that IM has decreased availability. Figure 7-12 and Figure 7-13 seem to visually confirm. Table 12 however shows that based on univariate testing, for each of the three missions, the null hypothesis that the two cases are identical cannot be rejected.

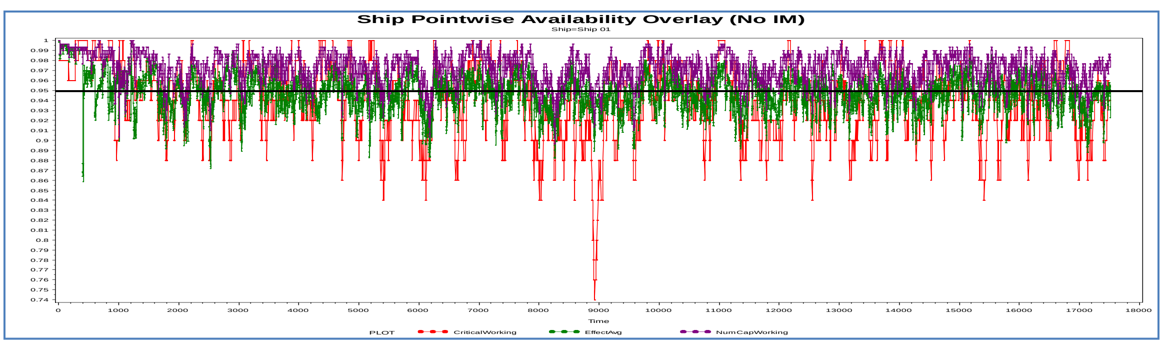

Figure 7-10 - Default Scenario (No IM) Availability of Critical Systems, % of systems, and performance

---

[69] 50 replications appears to be a practical upper limit to the amount of data that could be saved on a simulation run. More replications can be generated if the master seed for random variable generation can be changed and the experiment repeated.

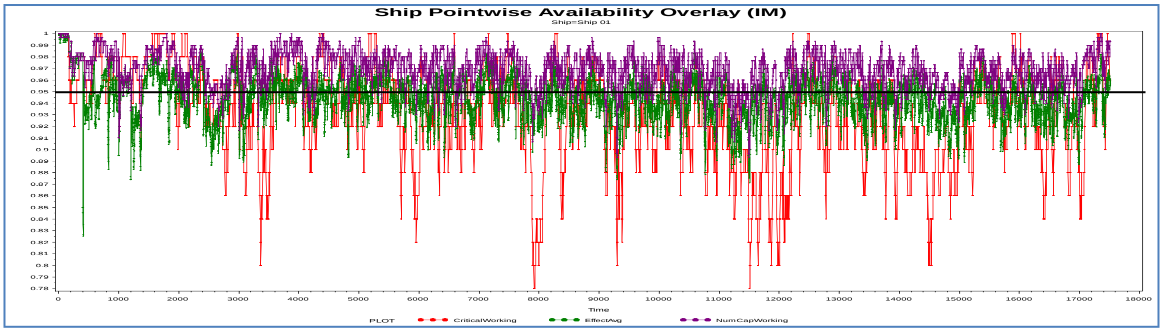

Figure 7-11 - Default Scenario (IM) Availability of Critical Systems, % of systems, and performance

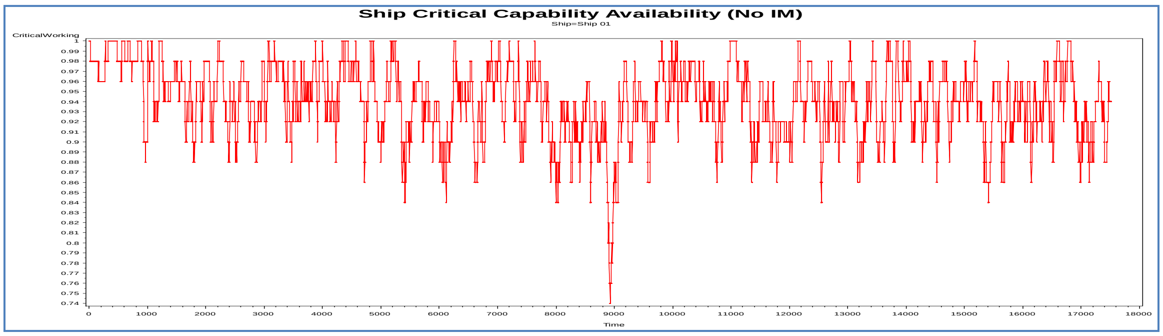

Figure 7-12- Default Scenario (No IM) Critical System Availability

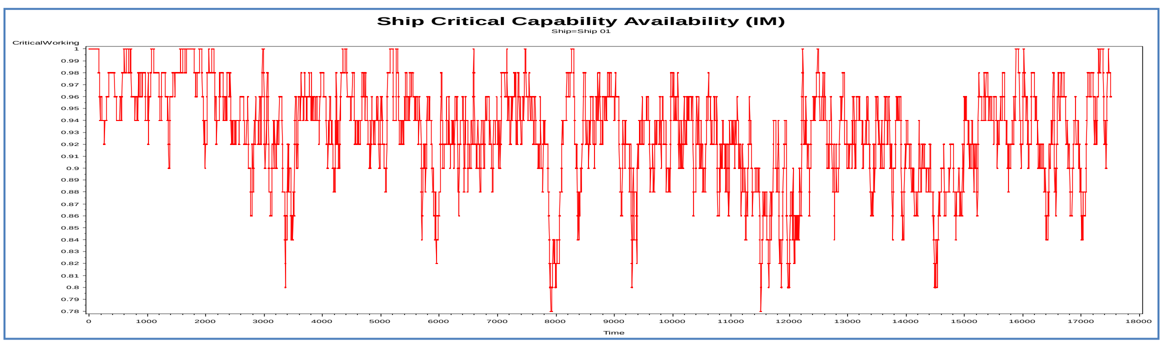

Figure 7-13 - Default Scenario (IM) Critical System Availability

The 'No IM' missions are Figure 7-14, Figure 7-16, and Figure 7-18. The three 'IM' missions are Figure 7-15, Figure 7-17, and Figure 7-19. Their results are in

Table 13.

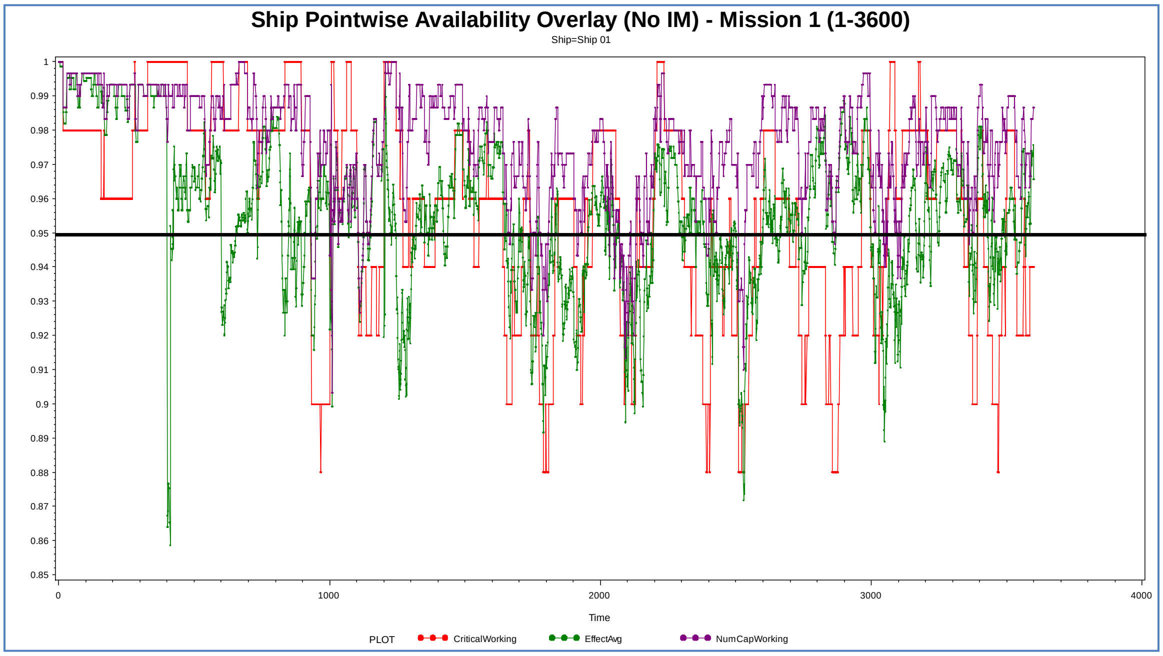

Figure 7-14 - *A*(*t*) Mission 1 (No IM)

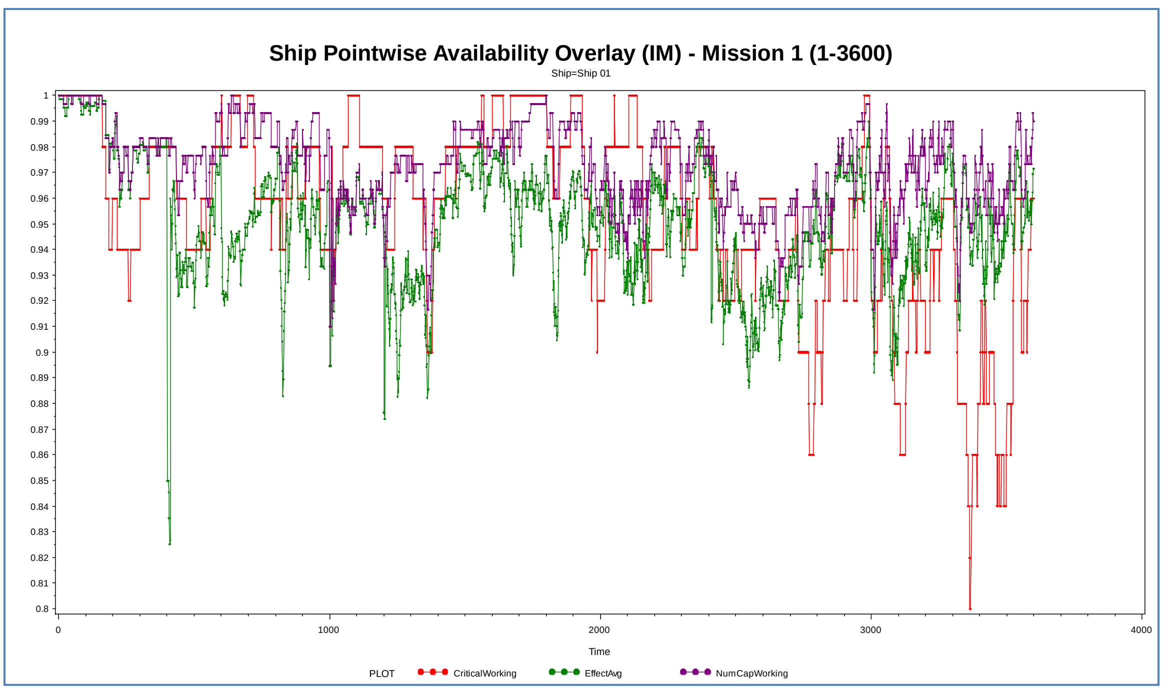

Figure 7-15 - *A*(*t*) Mission 1 (IM)

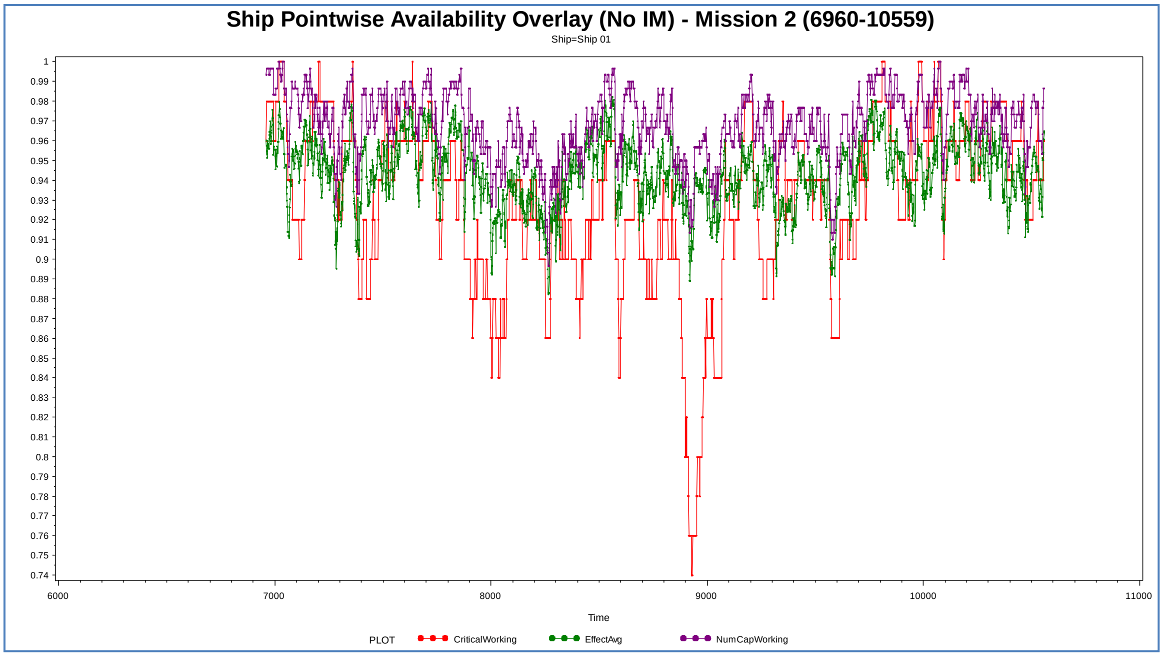

Figure 7-16 - *A*(*t*) Mission 2 (No IM)

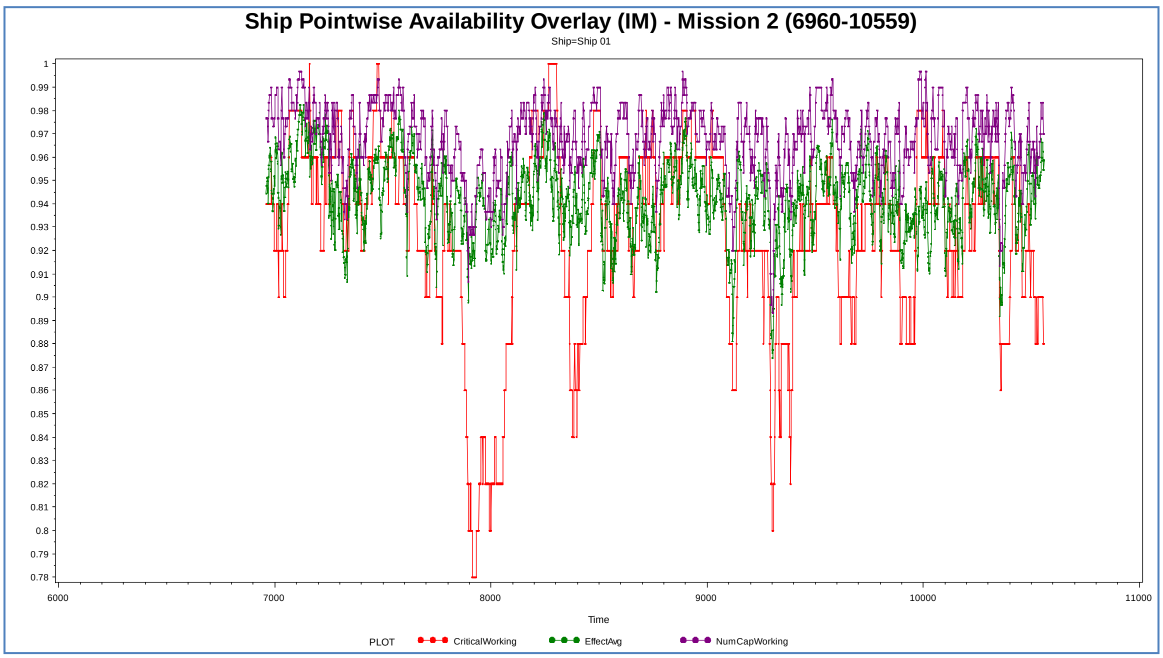

Figure 7-17 - *A*(*t*) Mission 2 (IM)

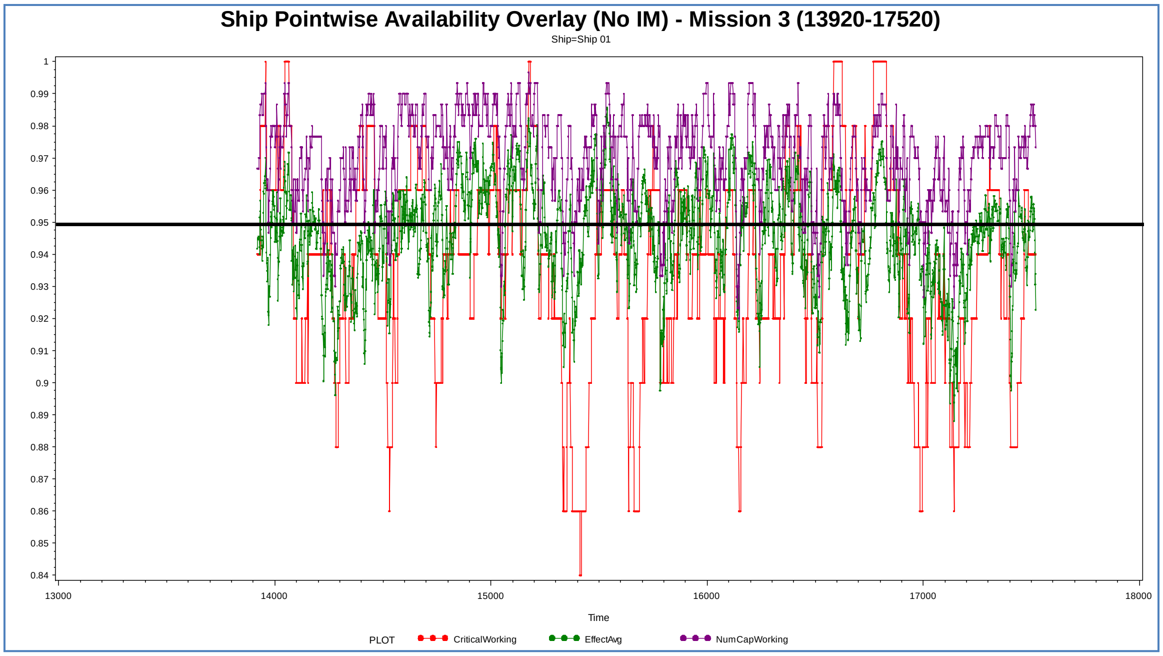

Figure 7-18 - *A*(*t*) Mission 3 (No IM)

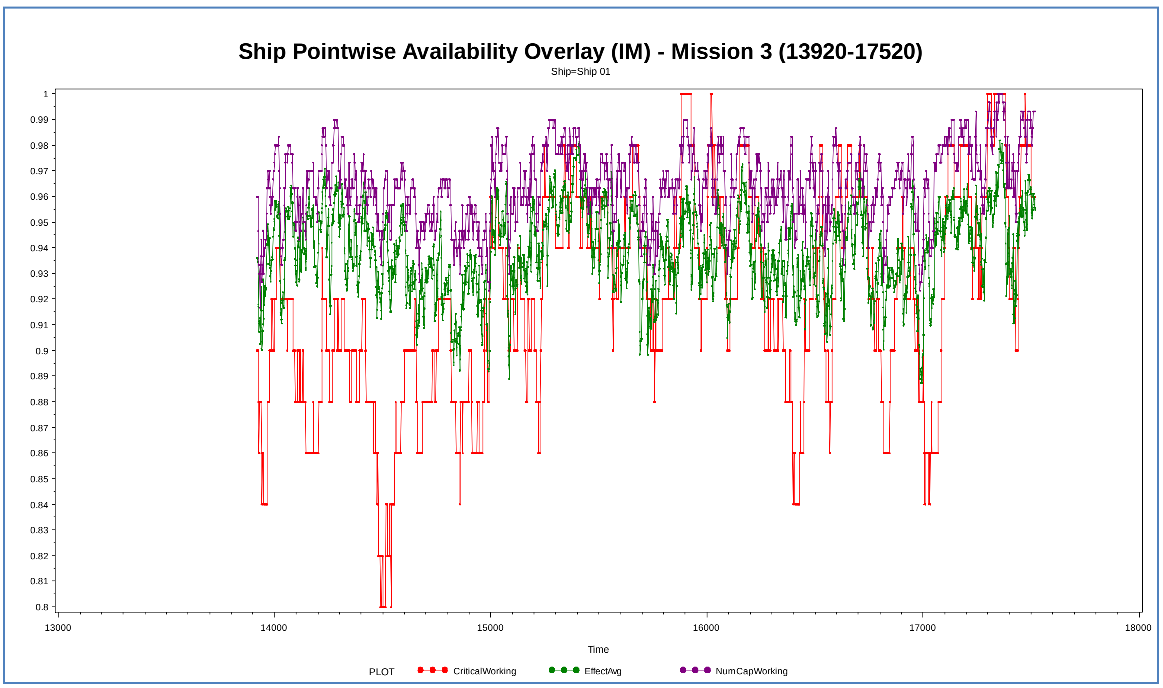

Figure 7-19 - *A*(*t*) Mission 3 (IM)

Table 13 - Comparison of availability metrics from case 'no IM' and 'IM'

Mission	Measure	Statistic	No IM	IM
Mission 1	Critical Capabilities	Mission Availability	.95669	.9551
		Std Dev	.20354	.0508
		Confidence Intervals	(.95575, .9576)*	(.9406,.9695)
	Mean Capabilities	Mission Availability	.97619	.9725
		Std Dev	.09879	.1097
		Confidence Intervals	(.97574, .9767)*	(.9720, .9730)
	Performance	Mission Performance	.95551	.9507
		Std Dev	.10756	.0322
		Confidence Intervals	(.95501, .9560)*	(.9416, .9599)
Mission 2	Critical Capabilities	Mission Availability	.92909	.9268
		Std Dev	.25668	.0486
		Confidence Intervals	(.92790, .9303)*	(.9131, .9404)
	Mean Capabilities	Mission Availability	.96842	.9678
		Std Dev	.11084	.01647
		Confidence Intervals	(.9679, .9689)*	(.9632, .9724)
	Performance	Mission Performance	.94342	.9425
		Std Dev	.11731	.01665
		Confidence Intervals	(.9429, .9440)*	(.9378, .9472)
Mission 3	Critical Capabilities	Mission Availability	.93566787	.9204
		Std Dev	.24534433	.09849
		Confidence Intervals	(.9345, .9368)*	(.8927, .9481)
	Mean Capabilities	Mission Availability	.97046	.9650
		Std Dev	.10106	.02563
		Confidence Intervals	(.9700, .9709)*	(..9578,.9722)
	Performance	Mission Performance	.94555	.9386

		Std Dev	.10950	.02259
		Confidence Intervals	(.9451, .9461)*	(.9323, .9450)

The results in Table 12 show only a small difference in availability despite running the simulation a only a .72 of renewing a part. Two different measures of confidence intervals are also presented for comparison. For the 'No IM' case, the confidence intervals (marked by *) are calculated based on 180000 data points (3600 hours * 50 replications). The 'IM' case has its confidence intervals based on just the 50 replications. In effect, the 'No IM' is capturing variance within and between while the 'IM' case is only capturing between variance. The 'No IM' CIs reject the 'IM' hypothesis of equivalency while the 'IM' case can't reject the null hypothesis. The measurement for 'No IM' CIs is based on normal assumption within the replication, yet this assumption is not correct for the recurrence/renewal cycles. For comparison of a sample set of data with another, sample variance need to be based on only between the replications.

The other observation for Table 12 is that while the first mission performed the best, the worst of the three for availability was the middle scenario. Even if there is a trend in the data, for small differences, randomness can play a large role in concealing the trend.

7.3 Phased Mission

The simulation was modified to implement a phased mission scenario. The phases are described in Section 6.2.3. The start of the deployment t_0 was set to 10,000 hours. The capabilities active for each phase are given in Table 14.

Table 14 - Phase capabilities for phased mission scenario

Phase	Capabilities
A1 (x4 capabilities)	Critical (x2), Navigation, Medium Range Surveillance

Phase	Capabilities
B1 (x3 capabilities)	Critical (x2), Navigation
C (x6 capabilities)	All
B2 (x3 capabilities)	Critical (x2), Navigation
A2 (x4 capabilities)	Critical (x2), Navigation, Medium Range Surveillance

Figure 7-20 shows the point availability of the phase mission scenario over the entire operational period . Each phase is recognizable by a distinctive density of the availability curves. This is caused by a couple of factors. First, the number of capabilities is different, and second the possible values of the performance metric change depending on which capabilities are being used. Figure 7-21 shows the availability for the actual mission , Phase C. Finally, the results of the availability metrics are shown in Table 15. The critical systems had a statistically significant better availability during Phase C, and the other availability metrics were significantly different from the operational period's average availability.

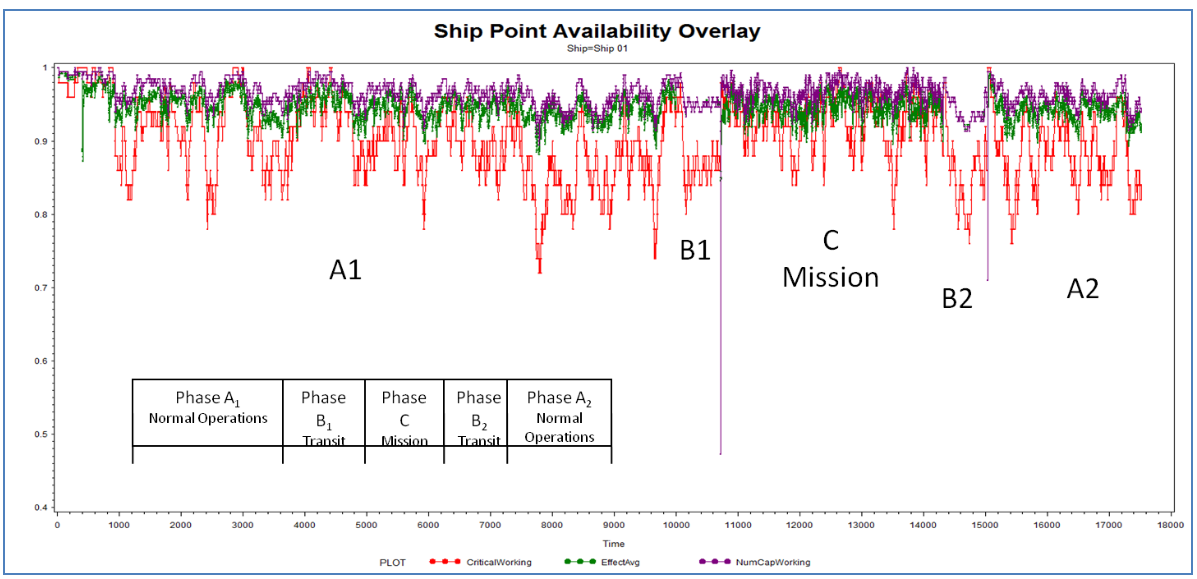

Figure 7-20- Phased Mission Scenario, *A*(*t*)

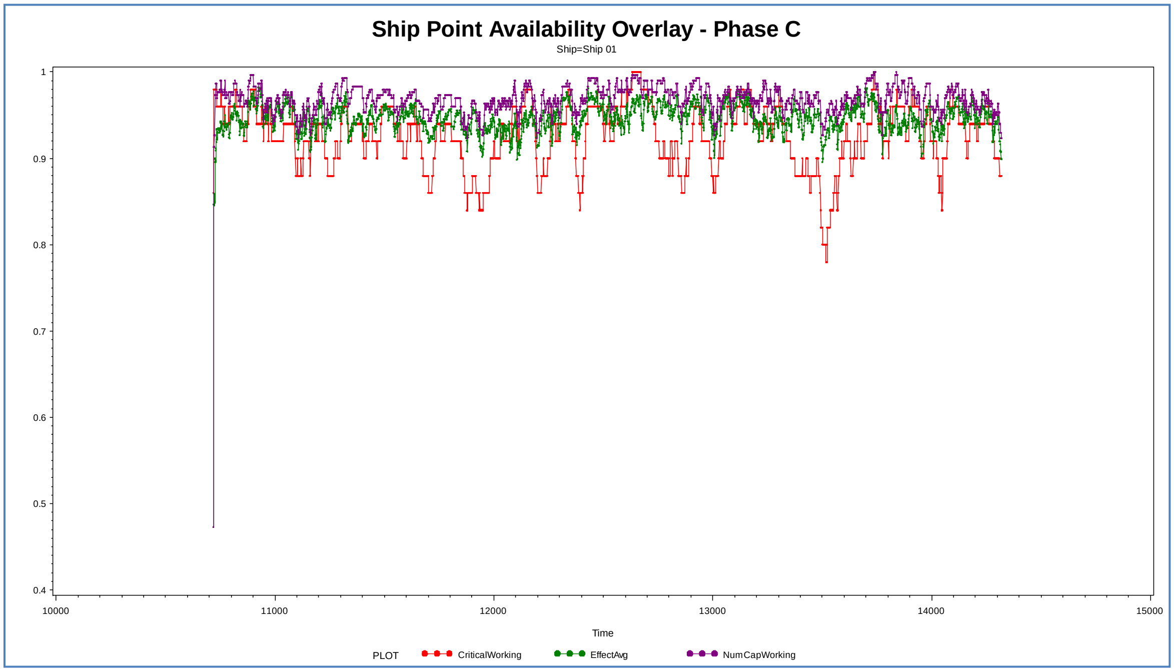

Figure 7-21 - Phased Mission Scenario - Phase C's *A*(*t*)

Table 15 - Phased Mission Availability Performance

Resolution	Statistic	Availability	Std Dev	CI (95%)
Ship	Critical Working	.9018	.0385	(.8909,.9128)
	Performance	.9471	.0146	(.9430,.9513)
	Mean Capability	.9653	.0141	(.9612,.9693)
Phase A1	Critical Working	.9034	.0472	(.8901,.9167)
	Performance	.9487	.0166	(.9441,.9534)
	Mean Capability	.9678	.0163	(.9632,.9723)
Phase B1	Critical Working	.8750	.1586	(.8304,.9196)
	Performance	.9548	.0596	(.9380,.9716)
	Mean Capability	.9546	.0596	(.9379,.9714)
Phase C	Critical Working	.9257	.0574	(.9095,.9418)
	Performance	.9463	.0294	(.9380,.9546)
	Mean Capability	.9683	.0283	(.9604,.9763)
Phase B2	Critical Working	.8534	.2081	(.7948,.9119)
	Performance	.9417	.1030	(.9128,.9707)
	Mean Capability	.9413	.1037	(.9121,.9704)
Phase A2	Critical Working	.8826	.1037	(.8534,.9117)
	Performance	.9410	.0332	(.9317,.9503)
	Mean Capability	.9608	.0361	(.9507,.9710)

8 Phase V - Application and Implementation

The purpose of the simulation developed in this Thesis was to create a methodology of evaluating ship Mission Availability and applying it to specific project management functions. These functions include procurement specifications, contract management, predicting mission availability performance.

8.1 Procurement- Setting Specifications

Setting the reliability and availability specifications for a developing capital procurement program is a difficult task. Simulation provides a way of evaluating and investigating sets of assumptions in the design process.

8.1.1 Optimization and Impact Analysis

Typical optimization problems that could be considered for this section are:

1. Maximize Availability,

   Subject to constraints of budget, and human resources, and capability requirements, and initial state of the ship

   By varying system and logistic choices

2. Minimize Lifetime Costs,

   Subject to minimum availability and reliability requirements, human resource limitations and capability requirements, and initial state of the ship

By varying system and logistic choices

Simulation, with output analysis, can be used for both optimization problems by action as the objective function. Simulation can also be setup to automate searching through the possibilities to find an optimal permutation of choices. The constraint of 'initial state of the ship' emphasizes that these objective functions are not just used at system design, but can be used throughout the ship's life cycle to provide new optimizations based on changes in assumptions. Acting as an objective function, the simulation also provides assessment of the impact of any potential decisions that need to be made. Impact analysis is required since the impact of decisions can have a non-uniform effect on the multi-function platform. Also, 'optimal' decisions are not always considered practical for external reasons, thus information on non-optimal design choices are also required as inputs into decision processes.

8.1.2 Methodology

The following steps might be applied to a brand new ship design and procurement program:

Step 1 - Define missions for the ship

Step 2 - Define capabilities for each mission

Step 3 - Assign a single configuration[70] to each capability and make reasonable failure/repair/supply/logistic assumptions

[70] Or assign a set of initial multiple configurations following standard practices for that capability.

Step 4 - Evaluate availability, reliability, and performance using the simulation. Predict costs[71]

Step 5 - Identify simulated systematic performance deficiencies, and modify the failure/repair/supply/logistic assumptions and policies.

Alternatively, configurations can be added, swapped, or removed

Step 6 - Repeats Steps 4 and 5 as desired

Step 7 - Repeat this process for each Mission defined in Step 1

This process is incremental and while it is time consuming, it provides linkage between availability, cost, and performance. If historical data is available, simulation input can be based on similar systems already in use.

If only adding a new system to an existing ship, the specification process is made simpler by having a existing platform, hopefully with relevant reliability data collected. In this case, the system is added to a simulation using the desired MCC RM, and its effects on availability, performance, and cost are evaluated.

8.1.3 Experiment Setup

The default scenario from Section 7 already represents the design process in 8.1.2. Capabilities were determined for a mission, configurations chosen, and assumptions were made. During some of the experiments in Section 7, deficiencies such as spare shortages were noted, and the design was updated.

[71] Cost estimation includes complete life cycle costs, and not just

Continuing this process, we note in Figure 7-10, near *t*=9000, there is an obvious deficiency in critical system availability. Essentially in 25% of the replications, a critical failure occurred. There are two critical capabilities identified, power and propulsion. Investigation, shows that main propulsion drops to .92 availability in a similarly shaped 'spike'. In a similar time period, power's availability is about .92. The difference is that the 'spike' in propulsion is a departure from its normal behaviour. The two systems can be compared using Figure 8-1 and Figure 8-2.

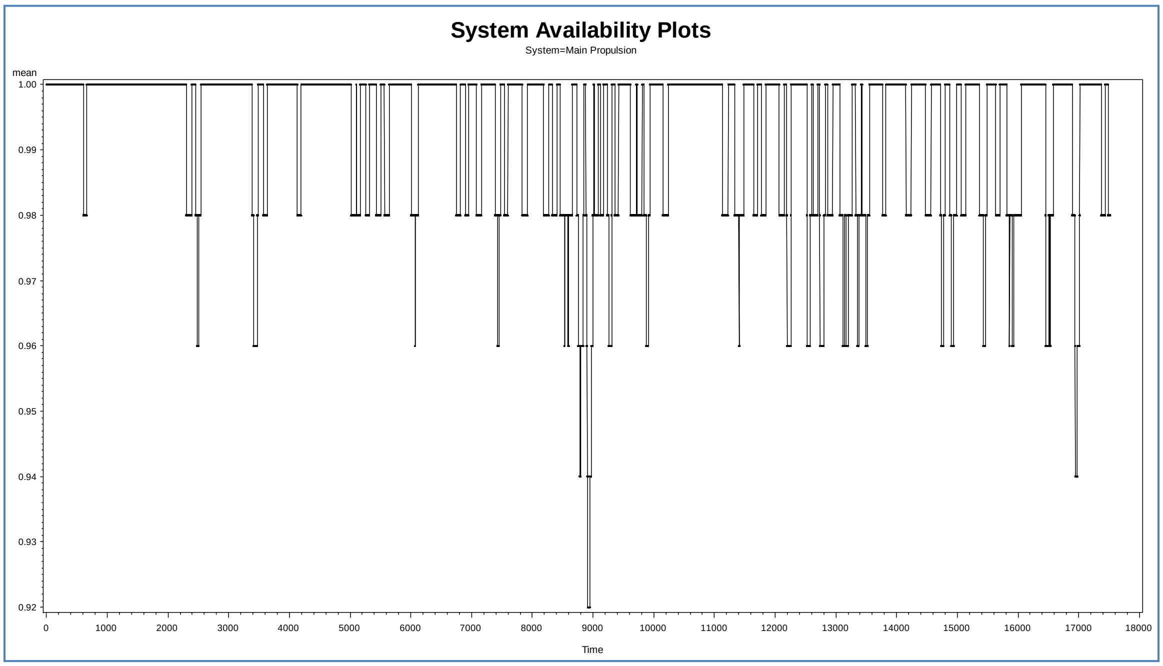

Figure 8-1 - Default Scenario System Analysis Main Propulsion

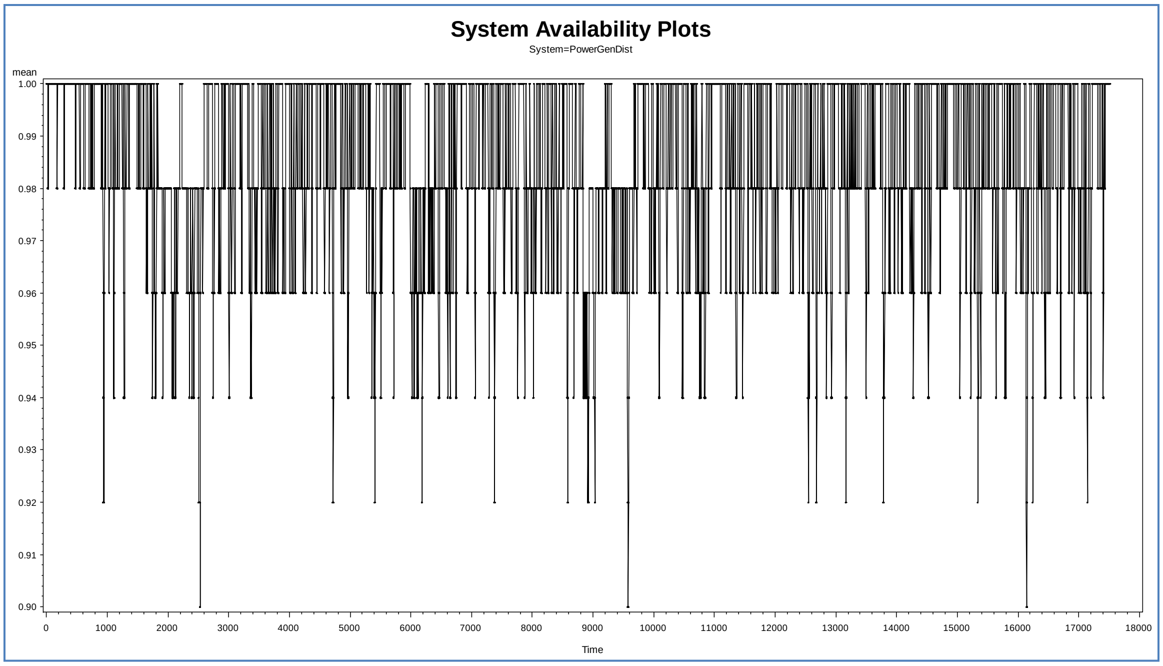

Figure 8-2 - Default Scenario System Availability analysis Power Generation

Investigating the MCF output for recurrence data for propulsion (Figure 8-3), it is noted that no single system accounted for all the failures at that time. The failure distributions modeled for the components have MTTFs which under the right condition may converge in this general region. The gearbox does have five recurrence cycles that might account for most of this 'spike' thus following the systematic methodology, either the assumptions for the gearbox are to be challenged (representing a selection of a different gearbox), or another gearbox might be added in parallel. Ignoring whether a second gearbox makes sense, for demonstration purposes, a second gearbox was added in parallel to the first, and the simulation was run to compare results.

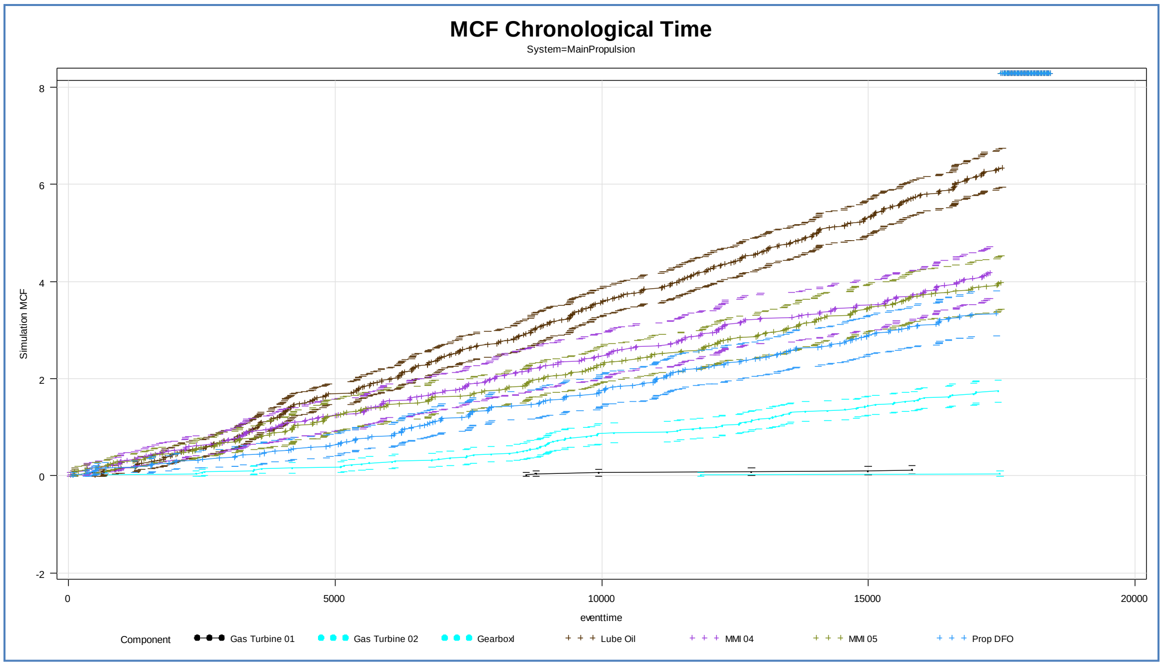

Figure 8-3 - MCF analysis of recurrence data for Propulsion

8.1.4 Results

The critical availability improved from .9190 to .9286 for the entire time period. Figure 8-4 shows the updated availability. It's not clear if the original problem was alleviated and a new concern shows up at t=5800. The process continues with further investigation, checking both if anything was improved from the previous problem, and seeing if the new problem can be mitigated. PM policy could also be considered in this analysis.

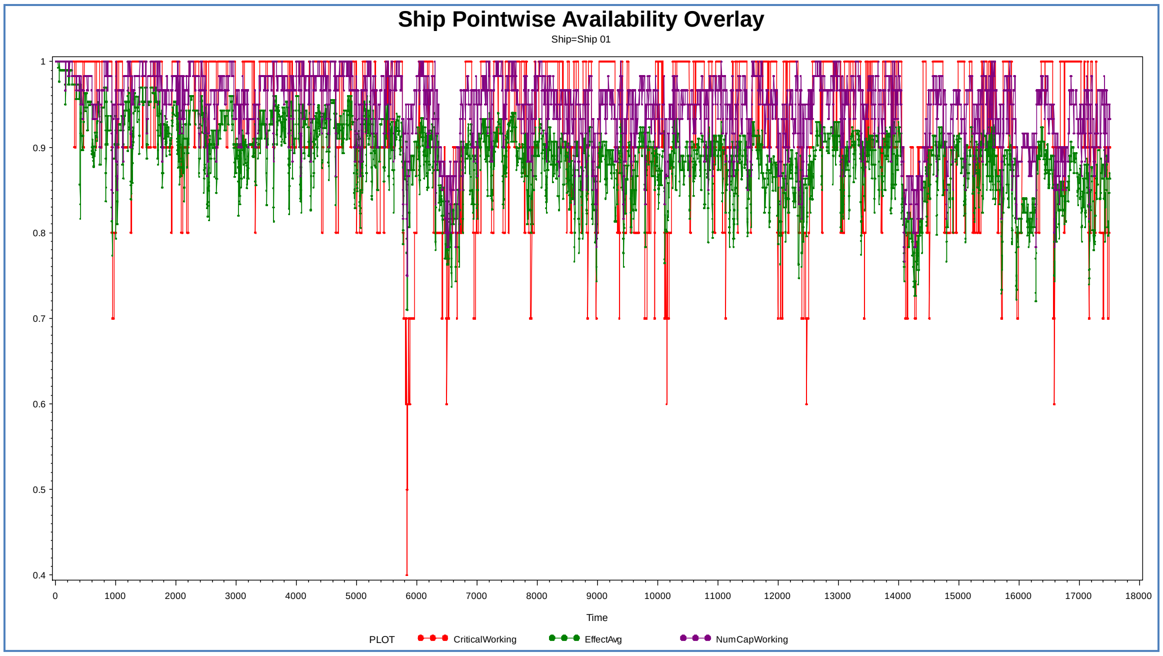

Figure 8-4 - Availability after simulation modification

8.1.5 Discussion

Having a simulation tool that can model and analyze ship mission availability provides a form of fast prototyping of ship designs, and of assessing expected impact on availability given any ship system modifications.

8.2 Contract Management - Comparing Availability

Throughout Section 2.1, the importance of availability / unavailability as a key performance metric in capital acquisition contracts was emphasized. Many challenges in managing this metric were also identified. In this section, an alternative method of evaluating ship performance is identified, described, and demonstrated.

8.2.1 Methodology

The simulation is used as a baseline comparison tool. Real data over an assessment period is used as data input into the failure and maintenance models. The simulation is run using this input to provide mission availability data and compared to the established performance metric. Number of replications is set to the number of ships under study (this can be multiple sampled though). The results of the simulation are compared to the specifications using an one sample t-test (due to small data sets, and variance estimated from the sample). There are three availability metrics are used in this Thesis, any one of them or the combination of the three (taking into account multiplicity concerns), can be used for comparison.

8.2.2 Experiment Setup

The default scenario was again used. All three availability metrics were used. Availability was evaluated on the mission time period t_{mp}=(13920,17520). The simulation was run several times, the first using the baseline assumptions, then using failure scale parameters at 90%, 80% and 70%, and the final run used the scale parameter at 130%. Each simulation run was ten replications, representing ten active ships within the ship class hypothetically studied.

8.2.3 Results

The specification are given in Table 8-1, and the results in Table 8-2. If the confidence intervals of the result includes the specification, then statistically, the 'real' data can't be rejected as 'out-of-spec'.

Table 16 - Specfications for contract management experiment

Specifications (from Table 12 for the case of no IM)	
Critical Availability	.9357
Mean capability availability	.9705
Effective Performance	.9456

Table 17 - Simulated 'real' data for comparison to 'specifications'

Test (sample size =10)	**Measure**	**Mean Std Dev**	**t-test confidence intervals (95%)**
Baseline	Critical Availability	.9424 .0169	(.9303, .9545)
	Mean capability availability	.9708 .00729	(.9656, .9760)
	Effective Performance	.9466 .00442	(.9434, .9497)
10% failure scale parameter reduction	Critical Availability	.9371 .0211	(.9220, .9522)
	Mean capability availability	.9708 .00863	(.9646, .9769)
	Effective Performance	.9479 .00797	(.9422,.09536)
20 % failure scale parameter reduction	Critical Availability	.9297 .03170	(.9071, .9524)
	Mean capability availability	.9706 .009098	(.9641, .9771)
	Effective Performance	.9474 .008144	(.9416, .9532)
30 % failure scale parameter reduction	Critical Availability	.7793 .0764	(.7246, .8340)
	Mean capability availability	.9013 .03563	(.8758, .9268)
	Effective Performance	.8850 .03267	(.8616, .9084)
30 % failure scale improvement	Critical Availability	.9496 .01527	(.9386, .9605)
	Mean capability availability	.9782 .005233	(.9745, .9820)
	Effective Performance	.9549 .006082	(.9506, .9593)

In equation (2.10), unavailability for constant hazard rate and MTTF>>MDT is approximated by $\lambda \cdot \text{MDT}$. While the simulation uses mostly non-exponential forms of Weibull, this may provide a clue why initial results from scale parameter degradation did not show any significant departure from the specifications. The idea is that for a balanced simulation it might not be sensitive to immediate changes in the failure times, and is actually more sensitive to changes in repair times. Eventually though, the increased failure rate invokes the spare shortage problem having dramatic effect on the availability.

8.2.4 Discussion

The simulation model for contract management should be kept relatively simple (the simulation models used in this Thesis are sufficient to this intent). A suggestion is that specifications are built upon a detailed and representative model from the end user, then translated to a simpler model with easily understood and accepted relationships for use in contract management. The idea is that the simpler model has less (but still reasonable) assumptions, and straightforward design. The availability from this simpler model is not expected to match real life, but instead is intended to set a baseline scenario of invariant assumptions for comparison of ship performance based on the observed reliability of the systems and components.

8.3 Operations - Assessing and Predicting Performance

Having specifications does not define the actual performance of the ship. This simulation tool can be used, with appropriate inputs, to predict not only availability/performance, but also accumulation of spares usage, failures, costs[72], and maintenance hours.

8.3.1 Methodology

Generating expected availability and performance measures has already been discussed throughout this Thesis (especially Section 7.2.2). For this Section, emphasis is

[72] Costs have not been specifically modeled into this Thesis.

given on updating the input data periodically using field data. Collecting useful field data is recognized as a challenge. If this data is successfully collected, better simulations and predictions are possible. The other point to be made is that the simulation needs to be able to set its initial conditions to match with current ship status. If a generator is broken on one ship, and a radar on another, the simulation must incorporate this information. This functionality was not programmed into the simulation for this Thesis, and is noted as a requirement for active predication of fleet performance.

Another method of predicting some aspects of performance is MCF. The MCF can be used to predict accumulation of failures, repairs (renewals), maintenance hours, and repair costs. When presented with real data, the MCF can be calculated directly without the simulation. The advantage of the simulation though, is the ability to use changing failure data to provide updated MCF predictions, rather than hindsight evaluations.

8.3.2 Experiment Setup

The data from the default scenario was used. Only MCF plots are studied here, as availability performance is sufficient covered throughout the Thesis. MCF information was processed through SAS's built-in Proc Reliability.

8.3.3 Results

The following MCF plots were generated from the default scenario. Figure 8-5 and Figure 8-6 show the accumulation of recurrence/renewal cycles by the Ship. These cycles are experienced by components within the ship, and rarely represent an actual ship renewal cycle; there are also indicative of spare usage by the ship. Both MCF plots mentioned need to be read with the understanding of 50 replications. For the ship, reading the plot suggests that at 5000 hours, 1.4 renewal cycles will have occurred (on average). In actual fact, there will have been 1.4*1900/50 = 53.2 cycles (on average) or .01064 renewal cycles per hour (per ship) (1900 is the number of 'units' listed on the graph). The difference is simply due to treating a replication as a 'unit' in the SAS code. The MCF plots from the default scenario, at the ship level, at the system level, and at the component level all appear to have a roughly linear accumulation. While many of the components also have linear appearing MCFs, some (Figure 8-7) do show some variance in form when their time scale is converted to 'run time'.

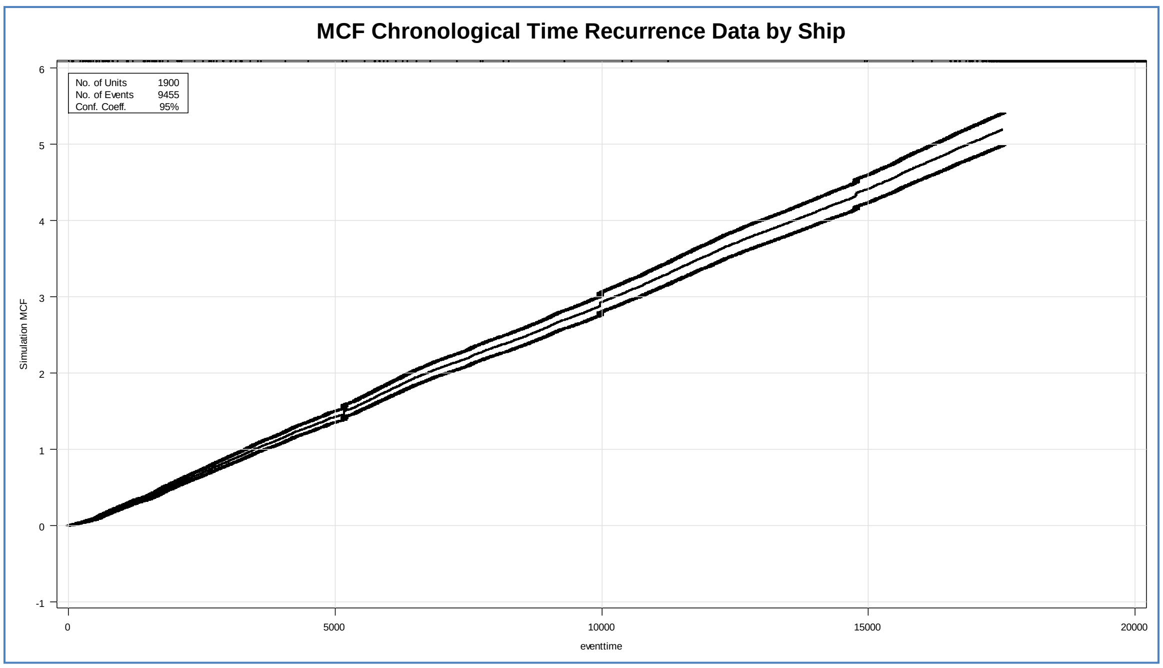

Figure 8-5- Accumulation of recurrences by the ship

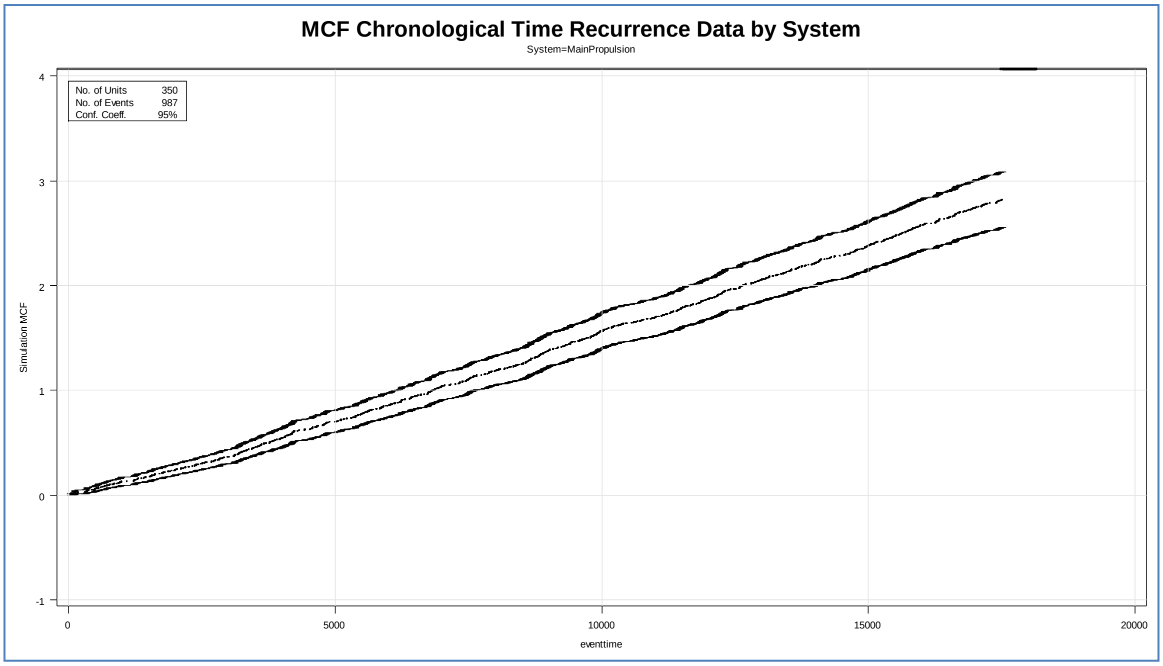

Figure 8-6 - Accumulation of recurrences for main propulsion

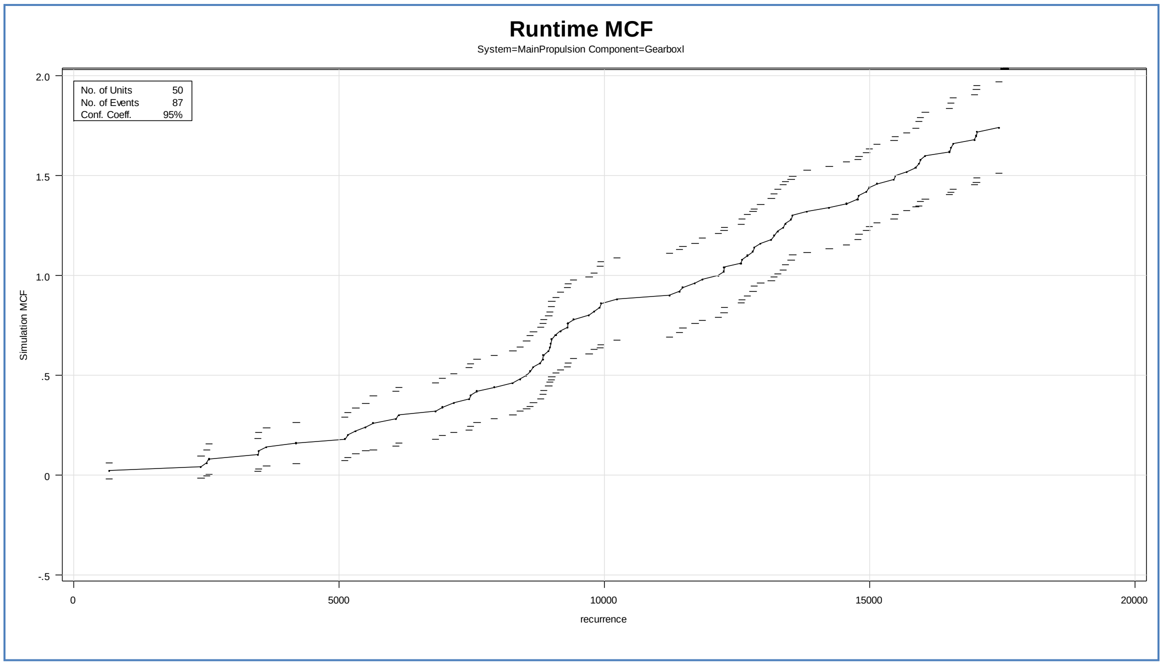

Figure 8-7 Accumulation of recurrences (component)

Figure 8-8 shows the accumulation of maintenance hours for the ship. This plot can be used to predict weekly broken into weeks, months, etc.. to predict the human resource drain of maintenance.

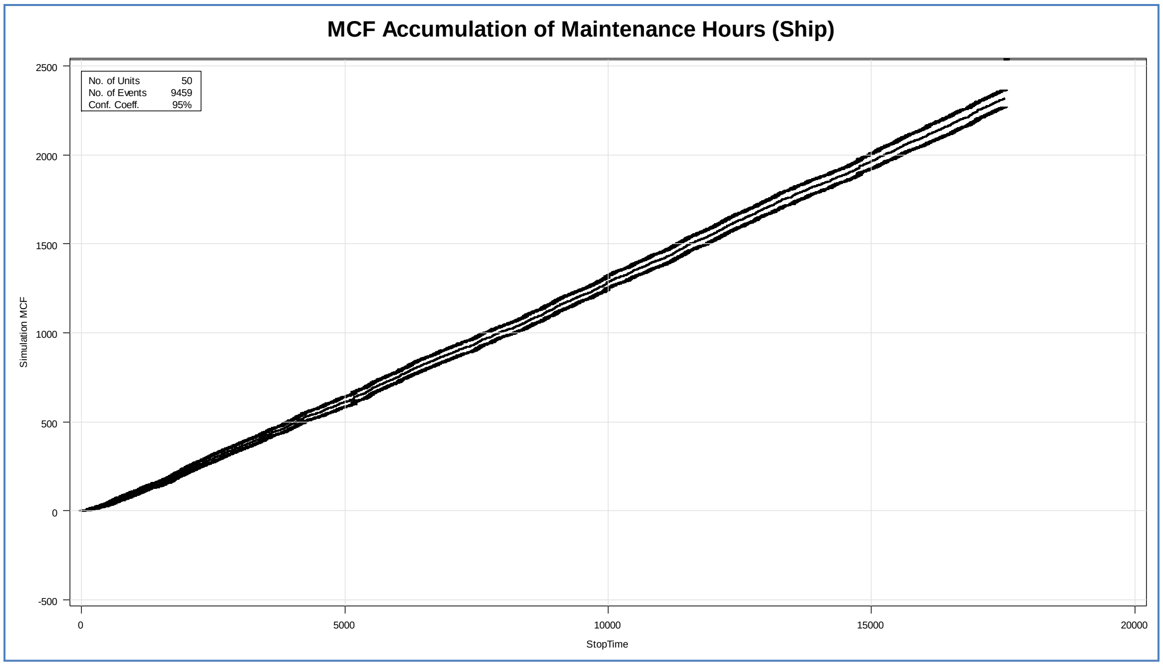

Figure 8-8 - Accumulation of maintenance (CM) hours by ship

8.3.4 Discussion

The plots appear linear, investigation into the simulation found a situation where the technician pool appears to be saturated. This is useful feedback for all discussed applications of the simulation, as saturation of the technicians prevents systems from being repaired as soon as possible, and impacts availability/performance.

9 Conclusion

Availability and reliability are mandated key performance metrics in current DND capital acquisition and in-service support programs. These metrics have been historically problematic to define, specify, and enforce. Different ideas and concepts are being tried; for example current efforts have refocused from managing availability to managing OEM attributed operational unavailability. Past methods have been focused on the Air Force environment.

In the Navy, current programs are underway to procure replacement warships. Warships are multirole, multifunction platforms. They can conduct a variety of different missions using different subsets of the systems installed. To determine their expected effectiveness during deployments, it is required to assess mission availability.

There are many definitions of mission availability, most of them commonly defined in a single function / capability context. The interpretation research and used was average interval availability applied to a multifunction platform. Three performance metrics were chosen to satisfy the extension. First, 'critical availability' as the intersection of critical capabilities. Second, 'mean capabilities availability' as the average of the union of capability states. Third, 'mean weighted performance availability' as the weighted and averaged union of working and partially working capabilities.

Mission Capability Configuration Reliability Model (MCC RM) was introduced to assess mission availability by relating system performance to capability (function) performance. The mission availability then becomes a function of capability instead of systems. The model can be considered a path based approach to assessing the ship's current engineering and operational state.

A simulation tool was developed to implement MCC RM. This simulation modeled a six capability (two critical), nine configuration, ten system and 38 component warship. It used a '4+1' tiered supply chain, with both unique and shared spares, and shared technicians for maintenance tasks. The mission assessed was a five month period of fully active capability within a two year operational cycle.

The simulation incorporated different corrective maintenance, preventative maintenance, imperfect maintenance, and logistical models and assumptions. The output of the simulation was used to assess the three multifunction mission availability performance metrics, and to evaluate spare usage, renewal cycles, and the accumulation of maintenance hours.

Finally the simulation was applied to three management functions. These functions are: platform design specifications / rapid availability prototyping, assessment / management of contract performance, and predicting engineering / operational performance of the existing fleet. The simulation of mission availability, as demonstrated, has application to Naval engineering problems.

.

10 References

http://www.forces.gc.ca/site/pri/2/invest-eng.asp.

ADM(MAT). *An In-Service Support Contracting Framework For Canadian Forces Platforms During the Initial Acquisition Stage.* Ottawa, On, 2009.

ADM(Mat). *ISS PBA Framework for OWSS-Oriented MCP.* 2008.

Andrews, J D. "Birnbaum and criticality measures of component contribution to the failure of phased missions." *Reliability Engineering & System Safety*, 2008: 1861-1866.

Arusand, Marvin, and Arnljot Hoyland. *"System Reliability Theory - Models Statistical Methods, and Applications".* 2nd. Hoboken, NJ: John Wiley & Sons, Inc, 2004.

Baxter, Laurence A, and Linxiong Li. "Non-parametric Confidence Intervals for the Renewal Function and the Point Availability." *Scandinavian Journal of Statisitcs* 21 (1994): 227-287.

Baxter, Laurence A, and Linxiong Li. "Nonparametric Estimation of the Limiting Availability." *Lifetime Data Analsysi* 2 (1996): 391-402.

Birolini, Alessandro. *On the Use of Stochastic Processes in Modeling Reliabilty Problems.* New York, NY: Springer-Verlag, 1985.

CF. *CFCD 129 Readiness and Sustainment.* Ottawa, On, 2009.

CF. *ISSCF v2.1.* Ottawa, On, 2009.

de Smidt-Destombes, Karin S, Matthieu C van der Heijden, and Aart van Harten. "On the avilability of a k-out-of-N system given limited spares and repair capabity under a condition based maintence strategy." *Reliability Engineering & System Safety*, 2004: 287-300.

Elsayed, Elsayed A. *Reliability Engineering I.* Piscataway, NJ: Rutgers, ISE Deptartment, 2011.

Esary, J D, and H Ziehms. "Reliability Analysis of Phased Missions." Edited by Richard E Barlow. *Reliability and Fault Tree Analysis.* Philadelphia: SIAM, 1975. 213-236.

Galante, Giacomo, and Gianfranco Passannanti. "An exact algorithm for preventive maintenance planning of series-parallel systems." *Reliability Engineering & System Safety*, 2009: 1517-1525.

Globalfirepower.com. 2010. http://www.globalfirepower.com/navy-ships.asp (accessed October 2010).

Jafari, Mohsen A. "Course Notes: Simulation of Production Systems." Piscataway: Rutgers, Spring 2010.

Johnston, Gordon. "Reliability Data Analysis in the SAS System." *suppport.sas.com.* http://support.sas.com/rnd/app/papers/reliability.pdf (accessed February 19, 2011).

Ke, Jau-Chuan, Hsin-I Huang, and Chuen-Horng Lin. "A redundant repairable system with imperfect coverage with fuzzy parameters." *Applied Mathematical Modelling*, 2008: 2839-2850.

Kim, Chul. "Analysis for Mission Reliability of a Combat Tank." *IEEE Transactions on Reliability* 38, no. 2 (1989): 242-245.

Loper, Margaret. "Course Notes: Modeling & Simulation for Systems Engineering." Atlanta: Georgia Institute of Technology Professional Education, April 2009.

Ma, Y, and K S Trivedi. "An algorithm for reliability analysis of phased-mission systems." *Reliability Engineering & System Safety*, 1999: 157-170.

Manzini, Riccardo, Alberto Regattieri, Hoang Pham, and Emilio Ferrari. *Maintenance for Industrial Systems.* New York: Springer, 2010.

Nahmias, Steven. *Production and Operations Analysis.* New York: McGraw-Hill Irwin, 2009.

Nelson, Wayne. "Graphical Comparison of Sets of Repair Data." *Quality and Reliability Engineering International* 16 (2000): 235-241.

OT DIV, CF Naval Engineering School. *Phase VI Marine Systems Engineering Requirements.* Halifax, NS, 2003.

Pham, Hoang, and Hongzhou Wang. "Imperfect Maintenance." *European Journal of Operational Research* 94 (1996): 425-438.

Pham, Hoang, and Hongzhou Wang. "Optimal (τ,T) Opportunisitc Maintenance of a k-out-of-n:G system with Imperfect PM and Partial Failure." *Naval Research Logisitics* 47 (2000): 223-239.

Remenyte-Prescott, R, J D Andrews, and P W H Chung. "An efficient phased mission reliability analysis for autonomous vehicles." *Reliability Engineering & System Safety*, 2010: 226-235.

Sherif, Dr Y.S., and Dr N.A. Kheir. "Weapons Systems Analysis Part I: System Effectiveness." *Microelectronics Reliability*, 1982: 531-567.

Sols, Alberto. "System Degraded Availability." *Reliability Engineering & System Safety*, 1997: 91-94.

Sols, Alberto, and A Joel Nachlas. "Availability of Multifunctional Systems." *Reliability Engineering and Systems Safety*, 1995: 69-74.

Sols, Alberto, Jose E Ramirez-Marquez, Dinesh Verma, and Begona Vitoriano. "Evaluation of full and degraded mission reliability and mission dependability for intermittently operated, multi-functional systems." *Reliability Engineering & System Safety*, 2007: 1274-1280.

Somani, Arun K, James A Ritcey, and Stephen H I Au. "Computantionally-Efficient Phased-Mission Reliability Analysis for Systems with Variable Configurations." *IEEE Transactions on Reliability*, 1992: 504-511.

Tsai, Yuo-Tern, Kuo-Shong Want, and Lin-Chang Tsai. "A study of availabilty-centered preventive maintenance for multi-component systems." *Reliability Engineering & System Safety*, 2004: 261-270.

Wang, Dazhi, and Kishor S Trivedi. "Reliability Analysis of Phased-Mission System With Independent Component Repairs." *IEEE Transactions on Reliability*, 2007: 540-551.

Weibull.com. http://www.weibull.com.

White, A. T., and A. O. F. Venton. ""Ship Availability Modelling - Practical Limitations"." *Reliability Engineering & System Safety* 20, no. 3 (1988): 199-217.

Willingham, David G, and John D Forster. "Availability Tradeoffs for Today's Complex Systems." *Proceedings Annual Reliability and Maintainability Symposium*, 1990: 242-249.

Zhao, Y X. "On preventive maintenance policy of a critical reliability level for system subject to degradation." *Reliability Engineering & System Safety*, 2003: 301-308.

Appendix A

SAS Code for Output Analysis

```
* File Name and Max Runtime Macro;
%let Filename =PhMn3; * Simulation Output File;
%let TableN=PM.&Filename ; *SAS Library to hold files;
%let MaxRunTime =17520 ; * Simulation Runtime;
%let LeftTimeLimit = 0;
%let RigthTimeLimit = 17520;
%let NumCap = 6; * Max capabilities, for phased mission this value
changes;
ODS RTF File='PhasedMissionResults.rtf'
PATH="C:\Users\scott\Documents\Thesis\SAS\";
ODS RTF CLOSE;

**********************************************************************;
* IMPORT TABLES FROM EXCEL;
proc import out=&TableN.Comp
datafile="C:\Users\scott\Documents\Thesis\Rockwell\&Filename..xls"
dbms=xls replace;
      getnames=yes;
      sheet=Components;
      datarow=2;
run;
proc import out=&TableN.System
datafile="C:\Users\scott\Documents\Thesis\Rockwell\&Filename..xls"
dbms=xls replace;
      getnames=yes;
      sheet=System;
      datarow=2;
run;
proc import out=&TableN.Configuration
datafile="C:\Users\scott\Documents\Thesis\Rockwell\&Filename..xls"
dbms=xls replace;
      getnames=yes;
      sheet=Configuration;
      datarow=2;
run;
proc import out=&TableN.Capability
datafile="C:\Users\scott\Documents\Thesis\Rockwell\&Filename..xls"
dbms=xls replace;
      getnames=yes;
      sheet=Capability;
      datarow=2;
run;
proc import out=&TableN.Ship
datafile="C:\Users\scott\Documents\Thesis\Rockwell\&Filename..xls"
dbms=xls replace;
      getnames=yes;
      sheet=Ship01;
      datarow=2;
run;

**********************************************************************;
* IMPORT TABLES FROM TEXT (TAB DELIMITED);
```

```
* When running many replications (especially >=50) the file sizes
become too large for the 'xls' Excel format.  A txt files is used
instead;
proc import out=&TableN.Comp
datafile="C:\Users\scott\Documents\Thesis\Rockwell\&Filename._component
.txt" dbms=dlm replace;
      delimiter='09'x; * tab;
      getnames=yes;
      datarow=2;
run;
proc import out=&TableN.System
datafile="C:\Users\scott\Documents\Thesis\Rockwell\&Filename._system.tx
t" dbms=dlm replace;
      getnames=yes;
      delimiter='09'x; * tab;
      datarow=2;
run;
proc import out=&TableN.Configuration
datafile="C:\Users\scott\Documents\Thesis\Rockwell\&Filename._configura
tion.txt" dbms=dlm replace;
      getnames=yes;
      delimiter='09'x; * tab;
      datarow=2;
run;
proc import out=&TableN.Capability
datafile="C:\Users\scott\Documents\Thesis\Rockwell\&Filename._capabilit
y.txt" dbms=dlm replace;
      getnames=yes;
      delimiter='09'x; * tab;
      datarow=2;
run;
proc import out=&TableN.Ship
datafile="C:\Users\scott\Documents\Thesis\Rockwell\&Filename._ship.txt"
dbms=dlm replace;
      getnames=yes;
      delimiter='09'x; * tab;
      datarow=2;
run;

**********************************************************************;

* Add frequency information for sas proc univariate;
Data &TableN.Comp;
      Set &TableN.Comp;
      FrequencyD=Runtime + Downtime;
      FrequencyR=Runtime + Repairtime;
run;

proc sort Data=&TableN.Comp out=&TableN.Comp;
      By System Component Replication StartTime;
run;

proc sort Data=&TableN.System out=&TableN.System;
      By System Replication StartTime;
run;

proc sort Data=&TableN.Configuration out=&TableN.Configuration;
```

```
      By Capability Configuration Replication StartTime;
run;

proc sort Data=&TableN.Capability out=&TableN.Capability;
      By Capability Replication StartTime;
run;

proc sort Data=&TableN.Ship out=&TableN.Ship;
      By Ship Replication StartTime;
run;

**********************************************************************;
*Separate fail repair and downtime times;
* Chronological times;
Data &TableN.Comp_fdrChrono replace;
      Set &TableN.Comp;
      By System Component Replication StartTime;
      eventtime=stoptime;
      failcens=failevent;
      repaircens=repairinterval;
      keep system component repaircens replication eventtime failcens;
      if (last.replication | failevent | repairinterval);
run;

* Chronological recurrence times;
Data &TableN.Comp_fdrChrono_recur replace;
      Set &TableN.Comp_fdrChrono;
      By System Component Replication;

      if (last.replication | repaircens);
run;

* MCF plot;
title 'MCF Chronological Time Recurrence Data by Component';
Proc Reliability Data=&TableN.Comp_fdrChrono_recur;
      unitid replication;
      mcfplot eventtime*repaircens(0)=Component /
vaxislabel='Simulation MCF';
      *mcfplot eventtime*repaircens(0)=Component / Interpolate = join
confidence=.95 overlay vaxislabel='Simulation MCF';
      By System;
run;

* MCF plot;
data &TableN.Comp_fdrChrono_recur_sys replace;
      set &TableN.Comp_fdrChrono_recur;
      By system component replication;
      retain extra 0;
      if (first.component) then extra=extra +1000;
      comp_plus_repl =extra+replication;
      if (eventtime>=&maxruntime) then repaircens=0;
run;
title 'MCF Chronological Time Recurrence Data by System';
Proc Reliability Data=&TableN.Comp_fdrChrono_recur_sys;
      unitid comp_plus_repl;
      mcfplot eventtime*repaircens(0) / vaxislabel='Simulation MCF';
```

```
	*mcfplot eventtime*repaircens(0)=Component / Interpolate = join
confidence=.95 overlay vaxislabel='Simulation MCF';
	By System;
run;

data &TableN.Comp_fdrChrono_recur_ship replace;
	set &TableN.Comp_fdrChrono_recur;
	By system component replication;
	retain extra1 0 extra2 0;
	if (first.component) then extra1=extra1 +100;
	if (first.system) then extra2=extra2+10000;
	sys_comp_plus_repl =extra1+extra2+replication;
	if (eventtime>=&maxruntime) then repaircens=0;
run;
title 'MCF Chronological Time Recurrence Data by Ship';
Proc Reliability Data=&TableN.Comp_fdrChrono_recur_ship;
	unitid sys_comp_plus_repl;
	mcfplot eventtime*repaircens(0) / vaxislabel='Simulation MCF';
	*mcfplot eventtime*repaircens(0)=Component / Interpolate = join
confidence=.95 overlay vaxislabel='Simulation MCF';
run;


* Relative to runtime;
* Leftover FrequencyD and FrequencyR are for repair and downtime only;
Data &TableN.Comp_fdrRun replace;
	Set &TableN.Comp;
	By System Component Replication StartTime;
	Retain tempfail 0 failtime 0;

	if (first.replication) then do;
		tempfail=FrequencyD;
		if (FailEvent = 1) then do;
			failtime=tempfail;
		end;
		else do;
			if (last.replication) then
				Failtime=tempfail;
			else Failtime=0;
		end;
	end;
	else if (last.replication) then do;
		tempfail=tempfail+FrequencyD;
		if (FailEvent = 1) then do;
			failtime=tempfail;
		end;
		else if (repairinterval=1) then do;
			tempfail=0;
			failtime=0;
		end;
		else do;
			failtime=tempfail;
		end;
	end;
	else do;
		tempfail=tempfail+FrequencyD;
		if (FailEvent = 1) then do;
```

```
				failtime=tempfail;
			end;
			else if (repairinterval=1) then do;
				tempfail=0;
				failtime=0;
			end;
			else do;
				failtime=0;
			end;
		end;

		if (repairinterval=1) then do;
			repairtime=FrequencyR;
			downtime=FrequencyD;
		end;
		else do;
			repairtime=0;
			downtime=0;
		end;

		repaircens=repairinterval;
		failcens=failevent;
		if (failevent=1 | repairinterval=1 | last.replication) then
			output;

		keep system component repaircens replication downtime repairtime
	failtime failcens stoptime;
	run;

	**********************************************************************;
	* MCF analysis;

	* Runtime recurrence times;
	Data &TableN.Comp_fdrRun_recur replace;
		Set &TableN.Comp_fdrRun;
		Retain temp 0 recurrence 0;
		By System Component Replication;

		recurrence=recurrence+failtime+downtime;
		recurcens=repaircens;

		if (last.replication | repaircens=1) then do;
			output;
		end;
		if (last.replication) then do;
			recurrence=0;
		end;

		keep system component recurrence recurcens replication;
	run;

	*MCF plot;
	*ODS graphics on;
	title 'Runtime MCF';
	Proc Reliability Data=&TableN.Comp_fdrRun_recur;
		unitid replication;
```

```
      mcfplot recurrence*recurcens(0) / Interpolate = join
confidence=.95 overlay vaxislabel='Simulation MCF';
      By System Component;
run;
*ODS graphics off;

**********************************************************************;
* failure analysis;
Data &TableN.Comp_fdrrun_fail replace;
      Set &TableN.Comp_fdrrun;
      Retain tempfail 0;
      By System Component Replication;

      if (first.replication) then
            tempfail=failtime;
      else tempfail=tempfail+failtime; * note that failtime is zero for
repair events;
      failtime=tempfail;

      if ((last.replication | failcens) & (NOT repaircens));

      keep system component failtime failcens replication;
run;

title 'Component Failure Analysis';
ODS graphics on;
proc lifetest data=&TableN.Comp_fdrrun_fail plots=(s lls);
      By System Component;
      Time failtime*failcens(0);
run;
ODS graphics off;

**********************************************************************;
* downtime analysis ;
Data &TableN.Comp_fdrrun_down replace;
      Set &TableN.Comp_fdrrun;
      Retain temp 0;
      By System Component Replication;

      if (repaircens) then do;
            downcens=1;
            output;
      end;
      else if (last.replication & (&MaxRunTime-Stoptime>0)) then do;
            downcens=0;
            downtime=&MaxRunTime - StopTime;
            output;
      end;

      keep system component downtime downcens replication;
run;

*requires independence for analysis;
title 'Component Downtime Analysis';
ODS graphics on;
proc lifetest data=&TableN.Comp_fdrrun_down plots=(s lls);
      By System Component;
```

```
	Time downtime*downcens(0);
run;
ODS graphics off;

*******************************************************************;
* repair analysis, note that there is no repair censor information, it
can't be distinguished from downtime with available data;
* a modification to the simulator would be needed to indicate properly
censored repair times, this would also require modification to other
SAS code to account for the extra downtime data;
Data &TableN.Comp_fdrrun_repair replace;
	Set &TableN.Comp_fdrrun;
	Retain temp 0;
	By System Component Replication;

	if (repaircens) then do;
		repaircens=1;
		output;
	end;
	else if (last.replication & (&MaxRunTime-Stoptime>0)) then do;
		repaircens=0;
		repairtime=&MaxRunTime - StopTime;
		output;
	end;

	keep system component repairtime repaircens replication;
run;

*******************************************************************;
* 'downtime less repair' analysis, this is compared only with known
downtimes and repair times i.e. no censoring;
Data &TableN.Comp_fdrrun_drDiff replace;
	Set &TableN.Comp_fdrRun;
	if (repaircens) then do;
		diffdr=downtime-repairtime;
		output;
	end;
	keep system component diffdr replication;
run;

* Not as useful with the simulator (except for validation), but would
be useful with real data;
title 'Logistical Delay Analysis';
proc univariate data=&TableN.Comp_fdrrun_drDiff;
	By system;
	Class Component;
	Histogram diffdr;
	var diffdr;
run;

*******************************************************************;
* Component Availability Analysis;
Data &TableN.Comp_avail replace;* /debug;
	Set &TableN.Comp;
	By System Component Replication StartTime;

	retain prevStart 0 prevFinish 0;
```

```
if (_N_= 1) then do;
      Do i=StartTime TO StopTime;
            Time=i;
            Availability = UpOrDown;
            MRT= &MaxRunTime ;
            Output;
      end;
end;
else
if (last.replication) then do;
      if (StartTime>prevFinish+1) then do;
            Do i=prevFinish+1 TO StartTime-1;
                  Time=i;
                  Availability=0;
                  Output;
            end;
      end;
      Do i=StartTime TO StopTime;
                  Time=i;
                  Availability = UpOrDown;
                  Output;
      end;
      if (StopTime<&MaxRunTime ) then do;
            Do i=StopTime+1 to &MaxRunTime;
                  Time=i;
                  Availability=0;
                  Output;
            End;
      end;

end;
else do;
      if (StartTime = prevFinish+1) then do;
            Do i=StartTime TO StopTime;
                  Time=i;
                  Availability = UpOrDown;
                  Output;
            end;
      end;
      else do;
            Do i=prevFinish+1 TO StartTime-1;
                  Time=i;
                  Availability=0;
                  Output;
            end;
            Do i=StartTime TO StopTime;
                  Time=i;
                  Availability=UpOrDown;
                  Output;
            end;
      end;
end;

prevStart=StartTime;
prevFinish=StopTime;
keep Time Availability Component System Replication;
```

```
run;

*Empirical A(t) with point standard error;
title 'Component A(t)';
Proc Means Data=&TableN.Comp_avail noprint;
    By System Component;
      Class Time;
      output out=&TableN.Comp_avail_means mean=mean stderr=stderr;
      VAR Availability;
run;

* Confidence intervals 95% (normal);
Data &TableN.Comp_avail_means_CI;
      Set &TableN.Comp_avail_means;
      CI_low=max(mean-1.96*stderr/sqrt(_FREQ_),0);
      CI_high=min(mean+1.96*stderr/sqrt(_FREQ_),1);
run;

title 'Component Availability Plots without CI';
Proc gplot data=&TableN.Comp_avail_means;
      plot mean*time;
      *plot mean*time CI_high*time CI_low*time / overlay;
      By system component;
      symbol1 interpol=join color=black value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;

title 'Component Availability Plots with CI';
Proc gplot data=&TableN.Comp_avail_means_CI;
      plot mean*time CI_high*time CI_low*time / overlay;
      By system component;
      symbol1 interpol=join color=black value=dot height=0.1;
      symbol2 interpol=none color=cyan value=dot height=0.1;
      symbol3 interpol=none color=cyan value=dot height=0.1;
run;

* Availability of components over entire period.  Components downtimes
are not independent, not inherent availability;
title 'Full Time Interval Component Availability (Approx Limiting
Availability)';
Proc Means Data=&TableN.Comp_avail;
    By System;
      CLASS Component;
      VAR Availability;
run;

**********************************************************************;
**********************************************************************;
* Export Component data for import into Maple etc...;
Data TempExport;
      Set FT_7_step2; *Change as required;
      if (Component="Supply Cabinet CC");
      keep time availability;
run;
```

```
proc export data=TempExport
outfile="c:\Users\scott\Documents\Thesis\Rockwell\FullTest_X_Y.txt"
dbms=tab replace;
run;

**********************************************************************;
*This should be valid for component data.  Doesn't require changing
alot from the original data set, just calculating a
frequency=runtime+downtime (mutually exclusive sums);
*Yields inherent availability;
title 'Inherent Component Availability';
Proc Univariate Data=&TableN.Comp;
      By System;
      Class Component;
      Freq FrequencyD;
      Output out=&TableN.Comp_ReportInherAvail min=min max=max sum=sum
N=N mean=mean std=std;
      Var UpOrDown;
run;

**********************************************************************;
**********************************************************************;
**********************************************************************;
* System Analysis;

Data &TableN.System_avail replace;* /debug;
      Set &TableN.System;
      By System Replication StartTime;

      retain prevStart 0 prevFinish 0;

      if (_N_= 1) then do;
            Do i=StartTime TO StopTime;
                  Time=i;
                  Availability = 1;
                  MRT= &MaxRunTime ;
                  Output;
            end;
      end;
      else
      if (last.replication) then do;
            if (StartTime>prevFinish+1) then do;
                  Do i=prevFinish+1 TO StartTime-1;
                        Time=i;
                        Availability=0;
                        Output;
                  end;
            end;
            Do i=StartTime TO StopTime;
                        Time=i;
                        Availability = 1;
                        Output;
            end;
            if (StopTime<&MaxRunTime ) then do;
                  Do i=StopTime+1 to &MaxRunTime;
                        Time=i;
                        Availability=0;
```

```
                        Output;
                  End;
            end;

      end;
      else do;
            if (StartTime = prevFinish+1) then do;
                  Do i=StartTime TO StopTime;
                        Time=i;
                        Availability = 1;
                        Output;
                  end;
            end;
            else do;
                  Do i=prevFinish+1 TO StartTime-1;
                        Time=i;
                        Availability=0;
                        Output;
                  end;
                  Do i=StartTime TO StopTime;
                        Time=i;
                        Availability=1;
                        Output;
                  end;
            end;
      end;

      prevStart=StartTime;
      prevFinish=StopTime;
      keep Time Availability  System Replication;
run;

*Empirical A(t) with point standard error;
title 'System A(t)';
Proc Means Data=&TableN.System_avail noprint;
    By System;
      Class Time;
      output out=&TableN.System_avail_means mean=mean stderr=stderr;
      VAR Availability;
run;

* Confidence intervals 95% (normal);
Data &TableN.System_avail_means_CI;
      Set &TableN.System_avail_means;
      CI_low=max(mean-1.96*stderr/sqrt(_FREQ_),0);
      CI_high=min(mean+1.96*stderr/sqrt(_FREQ_),1);
run;

title 'System Availability Plots';
Proc gplot data=&TableN.System_avail_means_CI;
      plot mean*time;
      *plot mean*time CI_high*time CI_low*time / overlay;
      By system;
      symbol1 interpol=join color=black value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;
```

```
* Availability of systems over entire period.  System recurrence are
not independent, not inherent availability;
title 'Operational Period System Availability';
Proc Means Data=&TableN.System_avail N alpha=0.05 clm mean std;
    By System;
      VAR Availability;
run;


**********************************************************************;
**********************************************************************;
**********************************************************************;
* Configuration Analysis;

Data &TableN.Config_avail replace;* /debug;
      Set &TableN.Configuration;
      By Capability Configuration Replication StartTime;

      retain prevStart 0 prevFinish 0;

      if (_N_= 1) then do;
            Do i=StartTime TO StopTime;
                  Time=i;
                  Availability = 1;
                  MRT= &MaxRunTime ;
                  Output;
            end;
      end;
      else
      if (last.replication) then do;
            if (StartTime>prevFinish+1) then do;
                  Do i=prevFinish+1 TO StartTime-1;
                        Time=i;
                        Availability=0;
                        Output;
                  end;
            end;
            Do i=StartTime TO StopTime;
                        Time=i;
                        Availability = 1;
                        Output;
            end;
            if (StopTime<&MaxRunTime ) then do;
                  Do i=StopTime+1 to &MaxRunTime;
                        Time=i;
                        Availability=0;
                        Output;
                  End;
            end;

      end;
      else do;
            if (StartTime = prevFinish+1) then do;
                  Do i=StartTime TO StopTime;
                        Time=i;
                        Availability = 1;
```

```
                    Output;
                end;
            end;
            else do;
                Do i=prevFinish+1 TO StartTime-1;
                    Time=i;
                    Availability=0;
                    Output;
                end;
                Do i=StartTime TO StopTime;
                    Time=i;
                    Availability=1;
                    Output;
                end;
            end;
        end;

        prevStart=StartTime;
        prevFinish=StopTime;
        keep Time Availability  Capability Configuration Replication;
run;

*Empirical A(t) with point standard error;
title 'System A(t)';
Proc Means Data=&TableN.Config_avail noprint;
    By Capability Configuration;
      Class Time;
      output out=&TableN.Config_avail_means mean=mean stderr=stderr;
      VAR Availability;
run;

* Confidence intervals 95% (normal);
Data &TableN.Config_avail_means_CI;
      Set &TableN.Config_avail_means;
      CI_low=max(mean-1.96*stderr/sqrt(_FREQ_),0);
      CI_high=min(mean+1.96*stderr/sqrt(_FREQ_),1);
run;

title 'Configuration Availability Plots';
Proc gplot data=&TableN.Config_avail_means_CI;
      plot mean*time;
      *plot mean*time CI_high*time CI_low*time / overlay;
      By capability configuration;
      symbol1 interpol=join color=black value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;

* Availability of systems over entire period.  System recurrence are
not independent, not inherent availability;
title 'Operational Period Configuration Availability';
Proc Means Data=&TableN.Config_avail N alpha=0.05 clm mean std;
    By Capability Configuration;
      VAR Availability;
run;

**********************************************************************;
```

```
**********************************************************************;
**********************************************************************;
* Capability Analysis;

Data &TableN.Config_avail replace;* /debug;
      Set &TableN.Configuration;
      By Capability Configuration Replication StartTime;

      retain prevStart 0 prevFinish 0;

      if (_N_= 1) then do;
            Do i=StartTime TO StopTime;
                  Time=i;
                  Availability = 1;
                  MRT= &MaxRunTime ;
                  Output;
            end;
      end;
      else
      if (last.replication) then do;
            if (StartTime>prevFinish+1) then do;
                  Do i=prevFinish+1 TO StartTime-1;
                        Time=i;
                        Availability=0;
                        Output;
                  end;
            end;
            Do i=StartTime TO StopTime;
                        Time=i;
                        Availability = 1;
                        Output;
            end;
            if (StopTime<&MaxRunTime ) then do;
                  Do i=StopTime+1 to &MaxRunTime;
                        Time=i;
                        Availability=0;
                        Output;
                  End;
            end;

      end;
      else do;
            if (StartTime = prevFinish+1) then do;
                  Do i=StartTime TO StopTime;
                        Time=i;
                        Availability = 1;
                        Output;
                  end;
            end;
            else do;
                  Do i=prevFinish+1 TO StartTime-1;
                        Time=i;
                        Availability=0;
                        Output;
                  end;
                  Do i=StartTime TO StopTime;
                        Time=i;
```

```
                        Availability=1;
                        Output;
                  end;
            end;
      end;

      prevStart=StartTime;
      prevFinish=StopTime;
      keep Time Availability  Capability Configuration Replication;
run;

*Empirical A(t) with point standard error;
title 'System A(t)';
Proc Means Data=&TableN.Config_avail noprint;
    By Capability Configuration;
      Class Time;
      output out=&TableN.Config_avail_means mean=mean stderr=stderr;
      VAR Availability;
run;

* Confidence intervals 95% (normal);
Data &TableN.Config_avail_means_CI;
      Set &TableN.Config_avail_means;
      CI_low=max(mean-1.96*stderr/sqrt(_FREQ_),0);
      CI_high=min(mean+1.96*stderr/sqrt(_FREQ_),1);
run;

title 'Configuration Availability Plots';
Proc gplot data=&TableN.Config_avail_means_CI;
      plot mean*time;
      *plot mean*time CI_high*time CI_low*time / overlay;
      By capability configuration;
      symbol1 interpol=join color=black value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;

* Availability of systems over entire period.  System recurrence are
not independent, not inherent availability;
title 'Operational Period Configuration Availability';
Proc Means Data=&TableN.Config_avail N alpha=0.05 clm mean std;
    By Capability Configuration;
      VAR Availability;
run;


**********************************************************************;
**********************************************************************;
**********************************************************************;
* Ship Availability Analysis;
Data &TableN.Ship_avail replace;* /debug;
      Set &TableN.Ship;
      By Ship Replication StartTime;

      retain prevStart 0 prevFinish 0;
```

```
        numcapworking=numcapworking/&NumCap;

        if (_N_= 1) then do;
              Do i=StartTime TO StopTime;
                    Time=i;
                    Output;
              end;
        end;
        else
        if (last.replication) then do;
              if (StartTime>prevFinish+1) then do;
                    Do i=prevFinish+1 TO StartTime-1;
                          Time=i;
                          Output;
                    end;
              end;
              Do i=StartTime TO StopTime;
                          Time=i;
                          Output;
              end;
              if (StopTime<&MaxRunTime ) then do;
                    Do i=StopTime+1 to &MaxRunTime;
                          Time=i;
                          Output;
                    End;
              end;

        end;
        else do;
              if (StartTime = prevFinish+1) then do;
                    Do i=StartTime TO StopTime;
                          Time=i;
                          Output;
                    end;
              end;
              else do;
                    Do i=prevFinish+1 TO StartTime-1;
                          Time=i;
                          Output;
                    end;
                    Do i=StartTime TO StopTime;
                          Time=i;
                          Output;
                    end;
              end;
        end;

        prevStart=StartTime;
        prevFinish=StopTime;
        keep Time CriticalWorking EffectAvg Replication Ship
NumCapWorking;
run;

* this is a single sample of mean, no variance by themselves;
title 'Operational Period Ship Availability by replication';
Proc Means Data=&TableN.Ship_avail noprint;
    By Ship;
```

```
      Class replication;
      output out=&TableN.Ship_avail_repl mean= ;
      VAR CriticalWorking EffectAvg NumCapWorking;
run;

title 'Operational Period Ship Availability';
Proc Means Data=&TableN.Ship_avail_repl (where=(Replication is not
missing )) N alpha=0.05 clm mean std T;
    By Ship;
      VAR CriticalWorking EffectAvg NumCapWorking;
run;

* t-test for comparison with specification, or other population data;
title 'ttest';
proc ttest data=&TableN.Ship_avail_repl (where=(Replication is not
missing )) h0=.9357;
      var CriticalWorking;
run;
title 'ttest';
proc ttest data=&TableN.Ship_avail_repl (where=(Replication is not
missing )) h0=.9456;
      var EffectAvg;
run;
title 'ttest';
proc ttest data=&TableN.Ship_avail_repl (where=(Replication is not
missing )) h0=.9705;
      var NumCapWorking;
run;


title 'Operational Period Ship Availability';
Proc Means Data=&TableN.Ship_avail N alpha=0.05 clm mean std;
    By Ship;
      VAR CriticalWorking EffectAvg NumCapWorking;
run;

title 'Ship A(t)';
Proc Means Data=&TableN.Ship_avail noprint;
    By Ship;
      Class Time;
      output out=&TableN.Ship_avail_means mean=
stderr(CriticalWorking)=CW_std stderr(EffectAvg)=EA_std
stderr(NumCapWorking)=NC_std;
      VAR CriticalWorking EffectAvg NumCapWorking;
run;

*Ship availability confidence intervals, normal assumption;
Data &TableN.Ship_avail_means_CI replace;
      Set &TableN.Ship_avail_means;
      Crit_CI_low=max(CriticalWorking-1.96*CW_std/sqrt(_FREQ_),0);
      Crit_CI_high=min(CriticalWorking+1.96*CW_std/sqrt(_FREQ_),1);
      EA_CI_low=max(EffectAvg-1.96*EA_std/sqrt(_FREQ_),0);
      EA_CI_high=min(EffectAvg+1.96*EA_std/sqrt(_FREQ_),1);
      NC_CI_low=max(NumCapWorking-1.96*EA_std/sqrt(_FREQ_),0);
      NC_CI_high=min(NumCapWorking+1.96*EA_std/sqrt(_FREQ_),1);
run;
```

```
* All three ship A(t) on the same graph;
title 'Ship Point Availability Overlay';
legend1 label=none position(bottom center outside) mode=reserve
shape=line;
axis1 label=none;
Proc gplot data=&TableN.Ship_avail_means;
      plot CriticalWorking*time EffectAvg*time NumCapWorking*time /
overlay legend=legend1 vaxis=axis1;
      By Ship;
      symbol1 interpol=join color=red value=dot height=0.1;
      symbol2 interpol=join color=green value=dot height=0.1;
      symbol3 interpol=join color=purple value=dot height=0.1;
run;

* Separate graphs;
title 'Ship Critical Capability Availability';
Proc gplot data=&TableN.Ship_avail_means_CI;
      plot CriticalWorking*time / overlay ;
      By Ship;
      symbol1 interpol=join color=red value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;

title 'Ship Effective Performance';
Proc gplot data=&TableN.Ship_avail_means_CI;
      plot EffectAvg*time / overlay ;
      By Ship;
      symbol1 interpol=join color=green value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;

title 'Ship % Capabilities Operating';
Proc gplot data=&TableN.Ship_avail_means_CI;
      plot NumCapWorking*time / overlay ;
      By Ship;
      symbol1 interpol=join color=purple value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;

*******************MISSION 1************************;
* All three ship A(t) on the same graph;
title 'Ship Point Availability Overlay (IM) - Mission 1 (1-3600)';
legend1 label=none position(bottom center outside) mode=reserve
shape=line;
axis1 label=none;
Proc gplot data=&TableN.Ship_avail_means (where=(Time<=3600));
      plot CriticalWorking*time EffectAvg*time NumCapWorking*time /
overlay legend=legend1 vaxis=axis1;
      By Ship;
      symbol1 interpol=join color=red value=dot height=0.1;
      symbol2 interpol=join color=green value=dot height=0.1;
      symbol3 interpol=join color=purple value=dot height=0.1;
run;
```

```
title 'Ship Critical Capability Availability (IM) - Mission 1 (1-
3600)';
Proc gplot data=&TableN.Ship_avail_means_CI (where=(Time<=3600));
      plot CriticalWorking*time / overlay ;
      *plot CriticalWorking*time Crit_CI_low*time Crit_CI_high*time /
overlay ;
      By Ship;
      symbol1 interpol=join color=red value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;

title 'Ship Effective Performance (IM) - Mission 1 (1-3600)';
Proc gplot data=&TableN.Ship_avail_means_CI (where=(Time<=3600));
      plot EffectAvg*time / overlay ;
      *plot EffectAvg*time EA_CI_low*time EA_CI_high*time / overlay ;
      By Ship;
      symbol1 interpol=join color=green value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;

title 'Ship % Capabilities Operating (IM) - Mission 1 (1-3600)';
Proc gplot data=&TableN.Ship_avail_means_CI (where=(Time<=3600));
      plot NumCapWorking*time / overlay ;
      *plot NumCapWorking*time NC_CI_low*time NC_CI_high*time / overlay
;
      By Ship;
      symbol1 interpol=join color=purple value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;

*******************MISSION 2*************************;
* All three ship A(t) on the same graph;
title 'Ship Point Availability Overlay (IM) - Mission 2 (6960-10559)';
legend1 label=none position(bottom center outside) mode=reserve
shape=line;
axis1 label=none;
Proc gplot data=&TableN.Ship_avail_means (where=(Time Between 6960 and
10559));
      plot CriticalWorking*time EffectAvg*time NumCapWorking*time /
overlay legend=legend1 vaxis=axis1;
      By Ship;
      symbol1 interpol=join color=red value=dot height=0.1;
      symbol2 interpol=join color=green value=dot height=0.1;
      symbol3 interpol=join color=purple value=dot height=0.1;
run;



title 'Ship Critical Capability Availability (IM) - Mission 2 (6960-
10559)';
Proc gplot data=&TableN.Ship_avail_means_CI (where=(Time Between 6960
and 10559));
      plot CriticalWorking*time / overlay ;
      By Ship;
```

```
      symbol1 interpol=join color=red value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;

title 'Ship Effective Performance (IM) - Mission 2 (6960-10559)';
Proc gplot data=&TableN.Ship_avail_means_CI (where=(Time Between 6960
and 10559));
      plot EffectAvg*time / overlay ;
      By Ship;
      symbol1 interpol=join color=green value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;

title 'Ship % Capabilities Operating (IM) - Mission 2 (6960-10559)';
Proc gplot data=&TableN.Ship_avail_means_CI (where=(Time Between 6960
and 10559));
      plot NumCapWorking*time / overlay ;
      By Ship;
      symbol1 interpol=join color=purple value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;

******************MISSION 3************************;
* All three ship A(t) on the same graph;
title 'Ship Point Availability Overlay (IM) - Mission 3 (13920-17520)';
legend1 label=none position(bottom center outside) mode=reserve
shape=line;
axis1 label=none;
Proc gplot data=&TableN.Ship_avail_means (where=(Time>=13920));
      plot CriticalWorking*time EffectAvg*time NumCapWorking*time /
overlay legend=legend1 vaxis=axis1;
      By Ship;
      symbol1 interpol=join color=red value=dot height=0.1;
      symbol2 interpol=join color=green value=dot height=0.1;
      symbol3 interpol=join color=purple value=dot height=0.1;
run;



title 'Ship Critical Capability Availability (IM) - Mission 3 (13920-
17520)';
Proc gplot data=&TableN.Ship_avail_means_CI (where=(Time>=13920));
      plot CriticalWorking*time / overlay ;
      By Ship;
      symbol1 interpol=join color=red value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;

title 'Ship Effective Performance (IM) - Mission 3 (13920-17520)';
Proc gplot data=&TableN.Ship_avail_means_CI (where=(Time>=13920));
      plot EffectAvg*time / overlay ;
      By Ship;
```

```
      symbol1 interpol=join color=green value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;

title 'Ship % Capabilities Operating (IM) - Mission 3 (13920-17520)';
Proc gplot data=&TableN.Ship_avail_means_CI (where=(Time>=13920));
      plot NumCapWorking*time / overlay ;
      By Ship;
      symbol1 interpol=join color=purple value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;

Data &TableN.Ship_avail_M1;
      set &TableN.Ship_avail;
      if (Time<=3600);
run;
Proc Means Data=&TableN.Ship_avail_M1 noprint;
    By Ship;
      Class replication;
      output out=&TableN.Ship_avail_M1_repl mean= ;
      VAR CriticalWorking EffectAvg NumCapWorking;
run;

title 'Mission 1 Availability Analysis (IM)';
Proc Means Data=&TableN.Ship_avail_M1_repl (where=(replication is not
missing)) N alpha=0.05 clm mean std;
    By Ship;
      VAR CriticalWorking EffectAvg NumCapWorking;
run;

Data &TableN.Ship_avail_M2;
      set &TableN.Ship_avail;
      if (Time>=6960 and Time <=10559);
run;
Proc Means Data=&TableN.Ship_avail_M2 noprint;
    By Ship;
      Class replication;
      output out=&TableN.Ship_avail_M2_repl mean= ;
      VAR CriticalWorking EffectAvg NumCapWorking;
run;
title 'Mission 2 Availability Analysis (IM)';
Proc Means Data=&TableN.Ship_avail_M2_repl N alpha=0.05 clm mean std;
    By Ship;
      VAR CriticalWorking EffectAvg NumCapWorking;
run;

Data &TableN.Ship_avail_M3;
      set &TableN.Ship_avail;
      if (Time>=13920);
run;
Proc Means Data=&TableN.Ship_avail_M3 noprint;
    By Ship;
      Class replication;
      output out=&TableN.Ship_avail_M3_repl mean= ;
      VAR CriticalWorking EffectAvg NumCapWorking;
```

```
run;
title 'Mission 3 Availability Analysis (IM)';
Proc Means Data=&TableN.Ship_avail_M3_repl N alpha=0.05 clm mean std;
    By Ship;
      VAR CriticalWorking EffectAvg NumCapWorking;
run;


* Accumulation of repair hours;
proc sort Data=&TableN.comp_fdrrun out=&TableN.comp_fdrrun_repl;
      By Replication;
run;
data &TableN.comp_fdrrun_ship replace;
      set &TableN.comp_fdrrun_repl;
      By replication;
      if (repaircens) then output;
      if (last.replication) then do;
            stoptime=&MaxRunTime +2;
            repaircens=0;
            output;
      end;
      keep stoptime repairtime reapircens replication;
run;

title 'MCF Accumulation of Maintenance Hours (Ship)';
Proc Reliability Data=&TableN.comp_fdrrun_ship;
      unitid replication;
      mcfplot stoptime*repairtime(0) / vaxislabel='Simulation MCF';
      *mcfplot eventtime*repaircens(0)=Component / Interpolate = join
confidence=.95 overlay vaxislabel='Simulation MCF';
run;


**********************************************************************;
**********************************************************************;
* Phased Mission Analysis for scenario in Thesis;
* Phase A1 - 0     to 9999 hours     4 capabilities ;
* Phase B1 - 10000 to 10719 hours    3 capabilities ;
* Phase C  - 10720 to 14319 hours    6 capabilities ;
* Phase B2 - 14320 to 15039 hours    3 capabilities ;
* Phase A2 - 15040 to end            4 capabilities ;

* (From ship analysis), this modified data step will use number of
working capabilities, and the next data step gives the corrected mean;
Data &TableN.Ship_avail_temp replace;* /debug;
      Set &TableN.Ship;
      By Ship Replication StartTime;

      retain prevStart 0 prevFinish 0;

      if (_N_= 1) then do;
            Do i=StartTime TO StopTime;
                  Time=i;
                  Output;
            end;
      end;
      else
```

```
	if (last.replication) then do;
		if (StartTime>prevFinish+1) then do;
			Do i=prevFinish+1 TO StartTime-1;
				Time=i;
				Output;
			end;
		end;
		Do i=StartTime TO StopTime;
				Time=i;
				Output;
		end;
		if (StopTime<&MaxRunTime ) then do;
			Do i=StopTime+1 to &MaxRunTime;
				Time=i;
				Output;
			End;
		end;

	end;
	else do;
		if (StartTime = prevFinish+1) then do;
			Do i=StartTime TO StopTime;
				Time=i;
				Output;
			end;
		end;
		else do;
			Do i=prevFinish+1 TO StartTime-1;
				Time=i;
				Output;
			end;
			Do i=StartTime TO StopTime;
				Time=i;
				Output;
			end;
		end;
	end;

	prevStart=StartTime;
	prevFinish=StopTime;
	keep Time CriticalWorking EffectAvg Replication Ship
NumCapWorking;
run;

Data &TableN.Ship_avail replace ;*/debug;
	set &TableN.Ship_avail_temp;
	By Ship replication time;
	if (first.replication) then do; * Time=1 correction;
		numcapworking=4;
		EffectAvg=1;
	end;
	else
	if (time < 10000 ) then do;
		numcapworking=numcapworking/4;
		EffectAvg=EffectAvg*&NumCap./4;
	end;
	else
```

```
	if (time >= 10000) & (time <10720) then do;
		numcapworking=numcapworking/3;
		EffectAvg=EffectAvg*&NumCap./3;
	end;
	else
	if (time >= 10720) & (time <14320) then do;
		numcapworking=numcapworking/6;
		EffectAvg=EffectAvg*&NumCap./6;
	end;
	else
	if (time >= 14320) & (time <15040) then do;
		numcapworking=numcapworking/3;
		EffectAvg=EffectAvg*&NumCap./3;
	end;
	else
	if (time >= 15040) then do;
		numcapworking=numcapworking/4;
		EffectAvg=EffectAvg*&NumCap./4;
	end;

	*These two lines are only for correcting a specific simulation
run nuance;
	if (NumCapWorking>1) then numcapworking =1;
	if (EffectAvg>1) then EffectAvg=1;
run;

* DO THIS NEXT: ;
* Create <name>_avail_means and <name>_avail_means_CI as previously
coded in ship analysis;
*;

Data &TableN.Ship_avail_A1;
	set &TableN.Ship_avail;
	if (Time<10000);
run;
Proc Means Data=&TableN.Ship_avail_A1 noprint;
    By Ship;
	Class replication;
	output out=&TableN.Ship_avail_A1_repl mean= ;
	VAR CriticalWorking EffectAvg NumCapWorking;
run;
title 'Mission A1 Availability Analysis';
Proc Means Data=&TableN.Ship_avail_A1_repl (where=(Replication is not
missing )) N alpha=0.05 clm mean std;
    By Ship;
	VAR CriticalWorking EffectAvg NumCapWorking;
run;
title 'Ship Point Availability Overlay - Phase A1';
legend1 label=none position(bottom center outside) mode=reserve
shape=line;
axis1 label=none;
Proc gplot data=&TableN.Ship_avail_means (where=(Time<10000));
	plot CriticalWorking*time EffectAvg*time NumCapWorking*time /
overlay legend=legend1 vaxis=axis1;
	By Ship;
	symbol1 interpol=join color=red value=dot height=0.1;
	symbol2 interpol=join color=green value=dot height=0.1;
```

```
      symbol3 interpol=join color=purple value=dot height=0.1;
run;
title 'Ship Critical Capability Availability - Phase A1';
Proc gplot data=&TableN.Ship_avail_means_CI (where=(Time<10000));
      plot CriticalWorking*time / overlay ;
      By Ship;
      symbol1 interpol=join color=red value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;
title 'Ship Effective Performance - Phase A1';
Proc gplot data=&TableN.Ship_avail_means_CI (where=(Time<10000));
      plot EffectAvg*time / overlay ;
      By Ship;
      symbol1 interpol=join color=green value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;
title 'Ship Mean Capabilities Operating Phase A1';
Proc gplot data=&TableN.Ship_avail_means_CI (where=(Time<10000));
      plot NumCapWorking*time / overlay ;
      By Ship;
      symbol1 interpol=join color=purple value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;

Data &TableN.Ship_avail_B1;
      set &TableN.Ship_avail;
      if (Time>=10000)& (Time<10720);
run;
Proc Means Data=&TableN.Ship_avail_B1 noprint;
    By Ship;
      Class replication;
      output out=&TableN.Ship_avail_B1_repl mean= ;
      VAR CriticalWorking EffectAvg NumCapWorking;
run;
title 'Phase B1 / Transit Availability Analysis';
Proc Means Data=&TableN.Ship_avail_B1_repl (where=(Replication is not
missing )) N alpha=0.05 clm mean std;
    By Ship;
      VAR CriticalWorking EffectAvg NumCapWorking;
run;
title 'Ship Point Availability Overlay - Phase B1';
legend1 label=none position(bottom center outside) mode=reserve
shape=line;
axis1 label=none;
Proc gplot data=&TableN.Ship_avail_means (where=((Time>=10000)&
(Time<10720)));
      plot CriticalWorking*time EffectAvg*time NumCapWorking*time /
overlay legend=legend1 vaxis=axis1;
      By Ship;
      symbol1 interpol=join color=red value=dot height=0.1;
      symbol2 interpol=join color=green value=dot height=0.1;
      symbol3 interpol=join color=purple value=dot height=0.1;
run;
title 'Ship Critical Capability Availability - Phase B1';
```

```
Proc gplot data=&TableN.Ship_avail_means_CI (where=((Time>=10000)&
(Time<10720)));
      plot CriticalWorking*time / overlay ;
      By Ship;
      symbol1 interpol=join color=red value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;
title 'Ship Effective Performance - Phase B1';
Proc gplot data=&TableN.Ship_avail_means_CI (where=((Time>=10000)&
(Time<10720)));
      plot EffectAvg*time / overlay ;
      By Ship;
      symbol1 interpol=join color=green value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;
title 'Ship Mean Capabilities Operating Phase B1';
Proc gplot data=&TableN.Ship_avail_means_CI (where=((Time>=10000)&
(Time<10720)));
      plot NumCapWorking*time / overlay ;
      By Ship;
      symbol1 interpol=join color=purple value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;


Data &TableN.Ship_avail_C;
      set &TableN.Ship_avail;
      if (Time>=10720) & (Time<14320);
run;
Proc Means Data=&TableN.Ship_avail_C noprint;
    By Ship;
      Class replication;
      output out=&TableN.Ship_avail_C_repl mean= ;
      VAR CriticalWorking EffectAvg NumCapWorking;
run;
title 'Phase C / Mission Availability Analysis';
Proc Means Data=&TableN.Ship_avail_C_repl (where=(Replication is not
missing )) N alpha=0.05 clm mean std;
    By Ship;
      VAR CriticalWorking EffectAvg NumCapWorking;
run;
title 'Ship Point Availability Overlay - Phase C';
legend1 label=none position(bottom center outside) mode=reserve
shape=line;
axis1 label=none;
Proc gplot data=&TableN.Ship_avail_means (where=((Time>=10720) &
(Time<14320)));
      plot CriticalWorking*time EffectAvg*time NumCapWorking*time /
overlay legend=legend1 vaxis=axis1;
      By Ship;
      symbol1 interpol=join color=red value=dot height=0.1;
      symbol2 interpol=join color=green value=dot height=0.1;
      symbol3 interpol=join color=purple value=dot height=0.1;
run;
```

```
title 'Ship Critical Capability Availability - Phase C';
Proc gplot data=&TableN.Ship_avail_means_CI (where=((Time>=10720) &
(Time<14320)));
      plot CriticalWorking*time / overlay ;
      By Ship;
      symbol1 interpol=join color=red value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;
title 'Ship Effective Performance - Phase C';
Proc gplot data=&TableN.Ship_avail_means_CI (where=((Time>=10720) &
(Time<14320)));
      plot EffectAvg*time / overlay ;
      By Ship;
      symbol1 interpol=join color=green value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;
title 'Ship Mean Capabilities Operating Phase C';
Proc gplot data=&TableN.Ship_avail_means_CI (where=((Time>=10720) &
(Time<14320)));
      plot NumCapWorking*time / overlay ;
      By Ship;
      symbol1 interpol=join color=purple value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;


Data &TableN.Ship_avail_B2;
      set &TableN.Ship_avail;
      if (Time>=14320) & (Time<15040);
run;
Proc Means Data=&TableN.Ship_avail_B2 noprint;
    By Ship;
      Class replication;
      output out=&TableN.Ship_avail_B2_repl mean= ;
      VAR CriticalWorking EffectAvg NumCapWorking;
run;
title 'Phase B2 / Transit Availability Analysis';
Proc Means Data=&TableN.Ship_avail_B2_repl (where=(Replication is not
missing )) N alpha=0.05 clm mean std;
    By Ship;
      VAR CriticalWorking EffectAvg NumCapWorking;
run;
title 'Ship Point Availability Overlay - Phase B2';
legend1 label=none position(bottom center outside) mode=reserve
shape=line;
axis1 label=none;
Proc gplot data=&TableN.Ship_avail_means (where=((Time>=14320) &
(Time<15040)));
      plot CriticalWorking*time EffectAvg*time NumCapWorking*time /
overlay legend=legend1 vaxis=axis1;
      By Ship;
      symbol1 interpol=join color=red value=dot height=0.1;
      symbol2 interpol=join color=green value=dot height=0.1;
      symbol3 interpol=join color=purple value=dot height=0.1;
```

```
run;
title 'Ship Critical Capability Availability - Phase B2';
Proc gplot data=&TableN.Ship_avail_means_CI (where=((Time>=14320) &
(Time<15040)));
      plot CriticalWorking*time / overlay ;
      By Ship;
      symbol1 interpol=join color=red value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;
title 'Ship Effective Performance - Phase B2';
Proc gplot data=&TableN.Ship_avail_means_CI (where=((Time>=14320) &
(Time<15040)));
      plot EffectAvg*time / overlay ;
      By Ship;
      symbol1 interpol=join color=green value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;
title 'Ship Mean Capabilities Operating Phase B2';
Proc gplot data=&TableN.Ship_avail_means_CI (where=((Time>=14320) &
(Time<15040)));
      plot NumCapWorking*time / overlay ;
      By Ship;
      symbol1 interpol=join color=purple value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;


Data &TableN.Ship_avail_A2;
      set &TableN.Ship_avail;
      if (Time>=15040);
run;
Proc Means Data=&TableN.Ship_avail_A2 noprint;
    By Ship;
      Class replication;
      output out=&TableN.Ship_avail_A2_repl mean= ;
      VAR CriticalWorking EffectAvg NumCapWorking;
run;
title 'Phase A2 Availability Analysis';
Proc Means Data=&TableN.Ship_avail_A2_repl (where=(Replication is not
missing )) N alpha=0.05 clm mean std;
    By Ship;
      VAR CriticalWorking EffectAvg NumCapWorking;
run;
title 'Ship Point Availability Overlay - Phase A2';
legend1 label=none position(bottom center outside) mode=reserve
shape=line;
axis1 label=none;
Proc gplot data=&TableN.Ship_avail_means (where=(Time>=15040));
      plot CriticalWorking*time EffectAvg*time NumCapWorking*time /
overlay legend=legend1 vaxis=axis1;
      By Ship;
      symbol1 interpol=join color=red value=dot height=0.1;
      symbol2 interpol=join color=green value=dot height=0.1;
      symbol3 interpol=join color=purple value=dot height=0.1;
```

```
run;
title 'Ship Critical Capability Availability - Phase A2';
Proc gplot data=&TableN.Ship_avail_means_CI (where=(Time>=15040));
      plot CriticalWorking*time / overlay ;
      By Ship;
      symbol1 interpol=join color=red value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;
title 'Ship Effective Performance - Phase A2';
Proc gplot data=&TableN.Ship_avail_means_CI (where=(Time>=15040));
      plot EffectAvg*time / overlay ;
      By Ship;
      symbol1 interpol=join color=green value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;
title 'Ship Mean Capabilities Operating Phase A2';
Proc gplot data=&TableN.Ship_avail_means_CI (where=(Time>=15040));
      plot NumCapWorking*time / overlay ;
      By Ship;
      symbol1 interpol=join color=purple value=dot height=0.1;
      *symbol2 interpol=none color=cyan value=dot height=0.1;
      *symbol3 interpol=none color=cyan value=dot height=0.1;
run;
```

Appendix B

Default Scenario Inputs

Component Models:

System	Component	F(t)	G(t)	Tech Type	IM Diff
AltNavRadar	AltNavInterface	Weibull(14000,1)	Unfiorm(1,8)	CS	1or .8
AltNavRadar	AltSensor	Weibull(14000,1.1)	Unfiorm(4,24)	CS	1or .8
CloseIn	CIWS	Weibull(3000,1)	Unfiorm(12,36)	CS	1or .8
FCS	FCScomputer	Weibull(7200,1)	1	CS	1or .8
FCS	FCSRadar_01	Weibull(860,3.1)	Unfiorm(4,12)	CS	1or .8
FCS	FCSRadar_02	Weibull(860,3.1)	Unfiorm(4,12)	CS	1or .8
LongRangeSensor	Control Cabinet	Weibull(1400,1.1)	1	CS	1or .8
LongRangeSensor	Cooling	Weibull(4300,1.4)	Unfiorm(2,8)	CS	1or .8
LongRangeSensor	MMI_03	Weibull(4380,1)	Unfiorm(2,4)	CS	1or .8
LongRangeSensor	Power LRS	Weibull(4400,1)	Unfiorm(4,8)	MS	1or .8
LongRangeSensor	Sensor Dish	Weibull(17000,1.4)	Unfiorm(48,96)	CS	1or .8
LongRangeSensor	Supply Cabinet CC	Weibull(3500,1.5)	1	CS	1or .8
MainGun	Gun	Weibull(1200,2.4)	Unfiorm(1,12)	CS	1or .8
MainPropulsion	DFO MP	Weibull(4380,1.2)	Unfiorm(5,20)	MS	1or .8
MainPropulsion	GasTurbine_01	Weibull(72000,1.7)	Unfiorm(24,120)	MS	1or .8
MainPropulsion	GasTurbine_02	Weibull(72000,1.7)	Unfiorm(24,120)	MS	1or .8
MainPropulsion	Gearbox	Weibull(8760,1.8)	Unfiorm(48,96)	MS	1or .8
MainPropulsion	LubeOilPump	Weibull(3000,2.2)	Unfiorm(8,24)	MS	1or .8
MainPropulsion	MMI_04	Weibull(4380,1)	Unfiorm(2,3)	MS	1or .8
MainPropulsion	MMI_05	Weibull(4380,1)	Unfiorm(2,3)	MS	1or .8
MedRangeSensor	Cabinet Assembly	Weibull(3500,1.5)	1	CS	1or .8
MedRangeSensor	Cooling MRS	Weibull(4300,1.4)	Unfiorm(2,8)	CS	1or .8
MedRangeSensor	MMI_06	Weibull(4380,1)	Unfiorm(2,3)	CS	1or .8
MedRangeSensor	PowerSupply MRS	Weibull(4400,1)	Unfiorm(4,8)	CS	1or .8
MedRangeSensor	SensorMRS	Weibull(17000,1.8)	Unfiorm(48,96)	CS	1or .8
Missiles	Launcher_01	Weibull(8200,1.4)	Unfiorm(48,72)	CS	1or .8
Missiles	Launcher_02	Weibull(8200,1.4)	Unfiorm(48,72)	CS	1or .8
NavRadar	NavCabinet	Weibull(6000,1)	1	CS	1or .8
NavRadar	NavInterface	Weibull(1500,1)	Unfiorm(1,4)	CS	1or .8
NavRadar	NavRadarSensor	Weibull(20000,1.3)	Unfiorm(12,24)	CS	1or .8
PowerGenDist	After SWBD	Weibull(1680,1.2)	Unfiorm(10,20)	MS	1or .8
PowerGenDist	DFO for PGD	Weibull(4380,1.2)	Unfiorm(5,20)	MS	1or .8
PowerGenDist	Fwd SWBD	Weibull(1680,1.2)	Unfiorm(10,20)	Tech _MS	1or .8
PowerGenDist	Generator_1	Weibull(1680,1.2)	Unfiorm(12,48)	MS	1or .8

PowerGenDist	Generator_2	Weibull(1680,1.2)	Unfiorm(12,48)	MS	1or .8
PowerGenDist	Generator_3	Weibull(1680,1.2)	Unfiorm(12,48)	MS	1or .8
PowerGenDist	MMI_01	Weibull(4380,1)	Unfiorm(2,3)	MS	1or .8
PowerGenDist	MMI_02	Weibull(4380,1)	Unfiorm(2,3)	MS	1or .8

Preventative Maintenance Models (by System):

System	Duration (hours)	Interval (hours)
Power Generation	N/A	N/A
Main Propulsion	N/A	N/A
Long Range Sensor	5	1000
Medium Range Sensor	12	400
Nav Radar	2	5000
Alt Nav Radar	1	7777
CIWS	4	600
AA Gun	4	600
Missiles	5	1200
FCS	10	1000

Spares Models:

Spare Type	Initial Spares	Max on Ship	Max on Support	Order Quantity	Order Interval	Backup Threshold	Backup Delay
GT	10	2	1	5	4500	0	4800
DFO Prop	10	4	1	5	3900	0	4800
Pump	10	5	1	5	3450	0	4800
Sensor	1	0	0	1	1000	0	4800
Circuit Card C	10	5	1	5	3600	0	4800
Bearings	10	2	1	5	4300	0	4800
PowerPlug	10	5	1	5	3080	0	4800
Cooling Parts MRS	10	4	1	5	4300	0	4800
DFO PGD spare	10	5	1	5	3750	0	4800
Cooling Parts	10	5	1	5	4100	0	4800
ParabolicDish	1	0	0	1	1000	0	4800
Generator	10	10	4	5	2920	0	4800
SWBD	10	2	0	5	3450	0	4800
Circuit Card A	20	5	1	5	3450	0	4800
Circuit Card B	20	5	1	5	3300	0	4800
Terminal	10	5	1	5	2050	0	4800
Launch Part	10	5	1	5	3900	0	4800
FCS Parts	14	5	3	5	3000	0	4800

Spare Type	Initial Spares	Max on Ship	Max on Support	Order Quantity	Order Interval	Backup Threshold	Backup Delay
Circuit Card E	10	5	1	5	4800	0	4800
CIWS Part	10	4	1	5	3900	0	4800
AltNavAE	1	0	0	1	1000	0	4800
AltNavTerminal	3	0	0	1	1000	0	4800
NavAE	1	0	0	1	1000	0	4800
NavTerminal	3	0	0	1	1000	0	4800
Gun Part	10	5	1	5	4000	0	4800
Circuit Card D	10	4	2	1	1000	0	4800

Technician Models:

Technician Type	**Skill**
CS	.9
MS	.9

Logistic Delay Models:

Path	**Delay**
Warehouse to TG or Ship	14 days
TG to Ship	1 day

MCC RM and performance weighting are described in sections 6.2.7 and 7.1.