SYSTEMS ANALYSIS TO OPTIMIZE RELIABILITY, AVAILABILITY, AND MAINTAINABILITY (RAM) AND REDUCE FAILURE

A Thesis

by

MAZDAK IAN MINA

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	James Holste
Committee Members,	Mahmoud El-Halwagi
	Jeffrey Hart
Head of Department,	James Holste

May 2019

Major Subject: Safety Engineering

Copyright 2019 Mazdak Ian Mina

ABSTRACT

Building a new optimization system that uses a systems approach and combined optimization approach (in the form of surrogate-based modeling) to produce the optimum Reliability (R(t)), Availability (A(t)), and Maintainability (M(t)), a.k.a. RAM, figures, companies figure out specifically how to adjust their industrial model to the "nuts and bolts" (a.k.a. "parts") level, to minimize their failures, thus maximizing their profits. By combining a systems approach with optimization of reliability, availability, and maintainability, a company can create a system that will maximize its profits and limit its cost due to the "Tip of the Iceberg" theory regarding losses in a safety incident. Designing a system that is able to optimize RAM at this level will involve defining the requirements of the system, determining how to find the initial reliability, optimum maintainability, and optimum availability of the program, as well as understand what the predecessor systems of this process were, and how they can be improved upon. Then, to make this system come to life, one must first do a detailed breakdown of the requirements needed to build the system, and then use those requirements to create a visual representation of the system that doubles as both a blueprint for the builder and an easy-to-follow guide through the system for the audience. Overall, by making the following program, companies would be able to reduce failure by updating their RAM optimization models to create a much more reliable RAM optimum and determine what needs to be done, down to the most specific (parts) level, thus maximizing their profits.

DEDICATION

To Dr. Sam Mannan (1954-2018)

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. James Holste, and my committee members, Dr. Mahmoud El-Halwagi and Dr. Jeffrey Hart for their guidance and support throughout the course of this research.

Thanks also to my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professor James Holste and Mahmoud El-Halwagi of the Department of Safety Engineering and Professor Jeffrey Hart of the Department of Statistics.

All work conducted for the thesis was completed by the student independently.

Funding Sources

Graduate study was supported by an assistantship from Texas A&M University. There are no outside funding contributions to acknowledge related to the research and compilation of this document.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

INTRODUCTION

By combining a systems approach with optimization of reliability, availability, and maintainability, a company can create a system that will maximize its profits and limit its cost due to the "Tip of the Iceberg" theory regarding losses in a safety incident. The "Tip of the Iceberg" theory shows that reducing incidents greatly increases the bottom line of companies, because there are often massive indirect (or "below the surface") costs to a safety incident that are not taken into account when calculating the total cost of an incident occurring (Afsahl et al, 2012) (OSHA, 2016). For this reason, a company's most efficient way to maximize its profits is to limit the risk of safety incidents, and optimization is the best way to reduce the risk of failure of a system). The best way to reduce the risk of failure is to optimize the reliability, and the best way to optimize the reliability is to optimize the maintainability and availability, due to their direct effect on reliability. Therefore, by creating a system in which the company can optimize its reliability, maintainability, and availability, a company will also maximize its potential profits due to the indirect costs, as explained below.

"Tip of the Iceberg" Theory

The "Tip of the Iceberg" theory proves that by reducing the number of incidents of failure, the company's bottom line will improve due to the direct and indirect costs saved by avoiding an incident. The "Tip of the Iceberg" theory states that the indirect (or hidden) costs of having an accident oftentimes can be much more (say, 10 times more) than the direct (or "visible") costs of an incident. The theory comes from ship damage estimates in the early 1900's, where damage estimates caused by hitting an iceberg often proved far off from the actual costs, due to the fact that estimates were being taken that only accounted for the "visible" part of the iceberg (a.k.a. the part of the iceberg that was above the surface). Since anywhere from three to 10 times the entire

iceberg was actually below the surface, the calculations using this method proved to be inadequate. By not taking the "hidden" part of the iceberg into account, shipping companies were falsely being lead to believe that the cost of putting in mechanisms to avoid hitting the iceberg were far less than the cost of striking one. Had the "hidden" part of the iceberg been taken into account, it would've clearly showed that hitting the iceberg proved far more costly than almost any avoidance mechanism, and that avoidance was clearly the more cost effective strategy.

The same idea has been proven to apply to safety in the modern world of industrial engineering. As Figure 1 shows, there are often far more costs that aren't considered when calculating the cost of an incident (these are the "hidden costs"), and as a result, many companies do not take an adequate approach to avoiding an incident, due to falsely believing that the incident will be less costly than the mechanism to avoid the incident. In reality, having an incident is almost always far more costly, and avoiding the incident is almost always the correct strategy in terms of being the most cost effective for a company.

Direct vs Indirect Costs

As stated in Afsahl et al, the three kinds of risks that a plant manager might have to face are:

1. Risks that are physically infeasible to fix
2. Risks that are physically feasible but economically infeasible to fix
3. Risks that are both physically and economically feasible to fix (Afsahl et al, 2012, pgs 2-3)

A smart plant manager only focuses on the third kind of risk (risks that are both physically and economically feasible to fix), because focusing on the other two is a waste of time and resources.

That time spent, and those resources allocated, would be better off allocated towards risks which are physically and economically feasible to fix instead. However, how can one determine whether or not a risk is economically feasible?

Due to the fact that, oftentimes, only the direct costs of an incident are considered, the economically feasible to fix risks are often never fixed. To avoid this mistake, the indirect costs must also be included in the calculation used in the final decision (see Table 1 below), as proven by companies such as McWane (Barstow et al, 2003)(Champion et al, 2017) and NASA. When analyzing the effects of improving safety within McWane and NASA, it can be proven that the cost of avoiding incidents is often far more economically feasible.

Incident	**Direct or *Indirect*?**	**Expected Cost Estimation**
Lawsuits	Direct	~$8,000/case
Government Fines	Direct	~$18,000/case
Insurance Payments	Direct	$2,000/case
Worker's Compensation	Direct	$10,000-$20,000/case
Property Damage	Direct	$500,000-$5Million/case
Medical Costs	Direct	$1,000-$3,000/case
Increase in Insurance Rates	*Indirect*	*$5,000/year*
Loss of Production during Incident	*Indirect*	*$4,000/hour*
Turnover of Workers	*Indirect*	*$800,000/worker*
Bad Press/Lost Customers	*Indirect*	*$200-800Million/Bad Press Day*
Loss of Sponsorship	*Indirect*	*$1-15Million/Sponsor*
Cost of Shutdown	*Indirect*	*>$3-5Million/day*
Total Direct Losses	Direct	*$450,000-$2.03Million*
Total Indirect Losses	*Indirect*	*>$1Billion/Incident*

Table 1: Comparing Common Direct (or "Seen") Costs to Indirect (or "Unseen") Costs of an Incident

For example, when the Challenger shuttle imploded, the "seen" costs for NASA were the loss of 8 human lives, the loss of the ship, lawsuits, and worker's compensation. However, there were many "unseen" costs as well, such as a loss of funding caused by public opinion of the disaster, the cost of PR to defend themselves to the public, all of which could've been spent on space research directly had the incident never occurred. In this way, the cost of the implosion was far more than the "seen" costs. (Hubbard, 2010)

Figure 1: Comparing the "Seen" and "Hidden" Costs [Reprinted from (2)]

And NASA is not the only example, as companies like McWane Piping Corporation also noticed a massive difference in product quality after taking drastic action to improve their safety record. When McWane Piping implemented sweeping changes to their safety program in 2011 (ISHN RSS, n.d.), safety incidents dropped, and in addition, the quality of their plant vastly improved, with revenues increasing massively (NY Times, n.d.). While it may be difficult to pinpoint exactly where the improvements in product occurred, the correlation between the reductions in failure of incidents and the revenue and quality of the product was clearly noticeable, giving more credence to the validity of the "Tip of the Iceberg" theory. Overall, the theory itself, combined with the histories of McWane (Companies-Histories.com, n.d.) and NASA, show that improving safety to try to avoid failures, or "avoiding hitting the iceberg", significantly reduces failures. For this reason, failures should be minimized to maximize profits, and the best way to do so is to maximize reliability. (Ayuub, 2003)

System's Approach

The best way to reduce failure is to optimize RAM, and the best way to optimize RAM is to take a systematic approach. A systematic approach involves breaking the system down into its hierarchical levels, defines the needs, explores concepts to meet those needs, and that develops a system to address those concepts (Barnard, 2008). For RAM Optimization, the required needs include optimizing R(t), A(t), and M(t) individually, takes into account all the levels of the hierarchy of the system that it is optimizing, and not only solves each level and each of R, A, and M, but does so in the correct sequence, while also adjusting each value as the other values change to create the optimum overall result. Each level of the hierarchy must be addressed, because failures that impact the entire system often are caused by a combination of failures at the lower levels of the system. When enough parts fail, that causes failures of the sub-components, which causes failures of the components, which causes failures of the sub-systems, and the entire system itself. To identify the cause of the failure of the system, one must find the failure of the parts of the system, and by taking a systems approach, one can optimize the system by optimizing every single level to minimize failure, and maximize reliability. (Hubbard, 2009)

System Hierarchy

As Figure 2 shows, the system hierarchy is broken into six levels, and is as follows (from largest to smallest):

1. System of Systems (Ex: Entire Company of Shell)
2. System (Ex: One Shell Plant)
3. Sub-System (Ex: Each building and area within a shell plant)
4. Component (Ex: Each machine within each of those buildings and areas)
5. Sub-component (Ex: Specific sub-sections of each of those machines)

6. Parts (Ex: The parts that those sub-sections are made of)

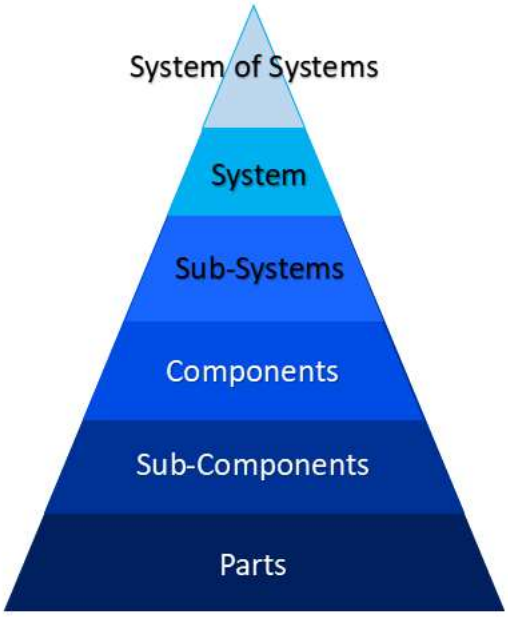

Figure 2: System Hierarchy

It's important that the system is broken down into its smallest parts, because as outlined before, the RAM of the System (or System of Systems) directly depends on the RAM of its sub-systems, which depends on the RAM of the components, and so on. If the individual parts of the system prove to be unreliable, that will affect the reliability of the sub-components involving those parts, which will reflect the reliability of the components, and the ripple effect will continue throughout the entire system and system of systems. To summarize, each lower level will directly affect the levels above it, and the affects will likely grow larger with each larger level, making it so that optimizing the entire system requires optimizing the smallest parts of that system. For this reason, an optimization system must start from the bottom up, and begin by optimizing the parts of the system before trying to optimize the entire system. (Buede, 2016) (Kosiakoff, 2011)

Bayesian Network

Building a model using all 6 layers of the hierarchy, that aggregates the probability to the complete product, will involve building a program that uses a Bayesian network (seen in Figure 3). Bayes theorem is the aggregation of a probability, while a Bayesian Network is a graphical structure of the stated model (Neapolitan, 2004). Using a Bayesian network, one can take the probability and consequence values received in from the FTA and the ETA, and see how they compound over each layer of the system (Christensen et al, 2011) (Mitchell et al, n.d.). The network shows which parts affect which sub-components, and in what way (are they in series or in parallel? Series means they both have to occur for the component to work properly, parallel means that both parts may contribute, but each one is separate from the others and does not depend on the others to maintain the sub-component), and once the numbers for the sub-components are received, how they affect the components in the same way, followed by the sub-systems, the systems, and the system of systems.

Overall, the Bayesian network is crucial, because it is the interaction diagram that allows the user to see exactly how each individual part or component affects each part above it. In doing so, allows the optimization of each part, sub-component, component, etc, by showing where the biggest risks in failure (and drops in reliability, maintainability, and availability) are occurring. This will allow the user to determine where the biggest opportunities for change are to be made, and determine how those changes will ultimately reduce the overall risk and, over several years, what the overall reduction in total accidents and overall reduction in cost will be. Ultimately, creating a Bayesian network is critical, because it is the visual network that the optimization system will be based on for each project (the specifications for each part will be given, followed by the structure and specifications for each sub-component and each layer above that), and in addition, unlike previous models, will allow the user to immediately see where the optimization

improvements need to occur at the lower levels (whereas previous models would give the optimum numbers, but left it up to the user to decide what the best way to reach those numbers were).

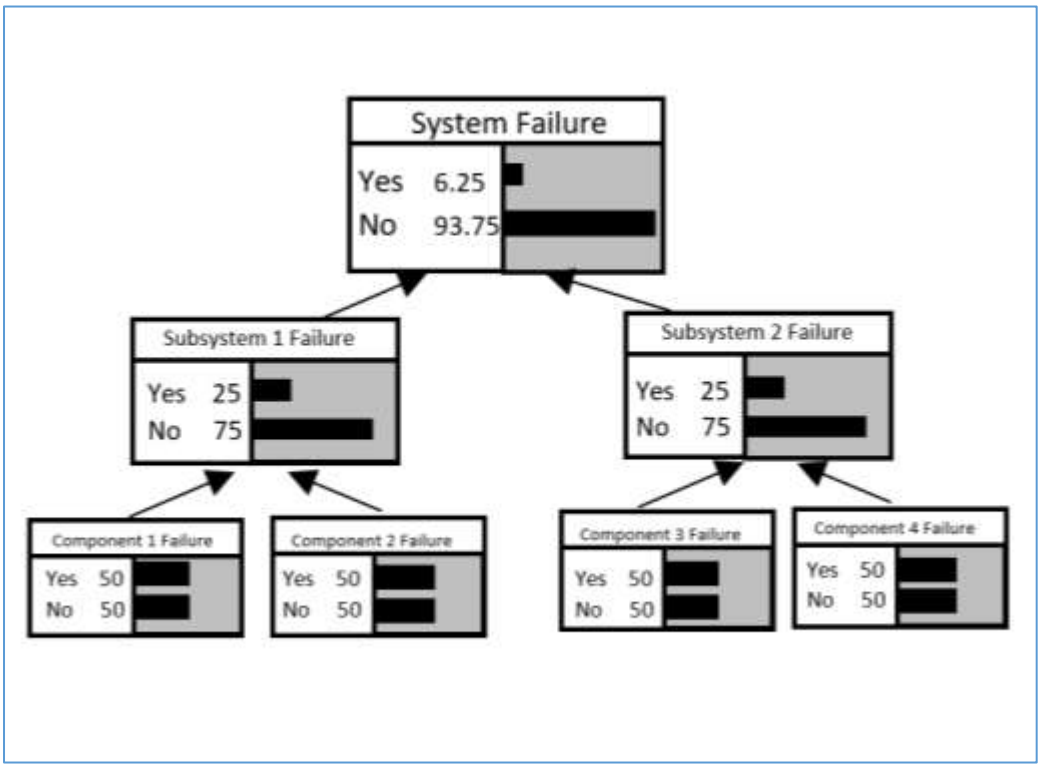

Figure 3: Example of a Bayesian Network

What is RAM?

To understand how to optimize RAM, one must first understand what "R", "A", and "M" are individually, how each part of RAM affects the other two, and what the optimization of each of those parts looks like, both individually, and altogether. By understanding the individual aspects of each, one can determine which part has the greatest and most noticeable impact on the other two, and it can be determined in what sequence one can meet the best results for finding the optimum. For example, reliability is directly dependent on the failure rate, which directly influences the time needed for maintainability, which directly influences the availability of the system. (Quanterion Solutions Approaches, 2017) (MITRE, 2017) (DoD, 2005)

Reliability

To optimize system reliability, one must lower the failure rate to the lowest feasible level, maximize reliability at each hierarchical level, as well as increase availability and maintainability to their maximum levels as well. Reliability is the ability of an item or system to continue to operate under certain defined conditions. By lowering the rate of failures, the reliability is increased due to the system being able to continue to operate. As a result, before trying to increase reliability, one must first determine how to decrease the rate of failure within each component of the system to a minimum level. (Ebeling, 2010) (Dictionary.com, n.d.) (Miriam-Webster, n.d.) (Social Research Methods, n.d.) (Military Handbook, 2008)

The first step to minimizing the risk of failure is to minimize the hazards, and by reducing both the frequency and consequence of failures through a HAZOP analysis, fault and event tree analysis, and risk matrix analysis. As mentioned before, reliability is the ability for a component or a system to continue to operate, and the ability to avoid failures is directly quantified in calculating the reliability (see Eq. 1) (Modarres, 2006) (Modarres, 2010)

$$R(t) = 1 - F(t) \quad \textbf{(Eq. 1)}$$

R(t) = System combined reliability (as a function of time, t)

F(t) = System combined failure (as a function of time, t)

$$R(t) = e^{-\lambda(t)} \quad \textbf{(Eq. 2a)}$$

$$R(t) = e^{-\beta/\Theta} \quad \textbf{(Eq. 2b)}$$

$$\lambda(t) = f(t)/R(t) \quad \textbf{(Eq. 3)}$$

λ(t) = Failure Rate

f(t) = Failure of single component

To reach the minimum failure rate, solutions may be offered to either decrease the consequence or the likelihood of each hazard (Pasman, 2015). By decreasing either the likelihood or the consequence, the overall risk is reduced, resulting in a reduction of failure for that particular hazard, and when reducing the combined risks of all the hazards, the overall failure rate can be reduced to a minimum feasible point (in the first iteration), resulting in the maximum feasible (or optimum) reliability rate (Ang, 1990) (Ang, 2007) (AIChE, 1995) (Jordaan, 2005) (DiStefano, 2009).

However, as stated before, reliability also affects, and is affected by availability and maintainability, and by maximizing reliability of each part and component, each part and component lasts longer, fails less, which means that there are both more parts and components available (increasing the availability), and that less time will be needed to be spent on the maintenance of each component when it fails. As a result, to maximize availability and maintainability, as stated many times before, one wants to maximize reliability by minimizing the failure rate.

Maintainability

Once the optimum reliability is found, it allows the user to determine the optimum maintainability by figuring out how to help maintain the components and parts after they fail. (Department of Defense, 2009) Maintainability is known as the ability to retain or restore a component or system to an "acceptable" level of functionality. (BusinessDictionary.com, n.d.) And once the optimal maintainability is found, having already found the optimal reliability, one can begin to look at the availability.

Availability

Finally, to maximize availability, on must maximize the three different aspects of availability, including the inherent availability, the operational availability, and the achievable availability. To maximize each of these aspects of availability, a good way to approach it is to try to reduce the unavailability (Q(t), see Eq. 4) of increasing the reliability (or more specifically, the mean time between failures, MTBF, and mean time between maintenances, MTBM), as well as reducing the estimated SAD.

$$Q(t) = 1 - A(t) \quad \text{(Eq. 4)}$$

Q(t) = Unavailability as a function of time,t

A(t) = Availability as a function of time, t

$$a(t) \sim 1 - \lambda t - \bar{M}/T \quad \text{(Eq. 5)}$$

$$a(t) \sim 1 - [(t/\Theta)^{\beta}] - \bar{M}/T \quad \text{(Eq. 6)}$$

What is Optimization?

The two main optimization approaches used to model RAM are the sequential and simultaneous approaches. Each has its own advantages and disadvantages, but both have flaws that make it difficult to determine whether or not the true optimum has been found. And as stated earlier, optimizing reliability, availability, and maintainability is critical to optimizing revenue, and finding the correct value will greatly increase the revenue of a company. Such examples of this include Marathon Ashland Petroleum, who saved $3 million in a year by avoiding heat exchanger failures, Exxon Mobil, reduced maintenance costs by about $1 billion per year by improving mechanical availability, and Conoco Refinery, who dropped maintenance costs by 21% and unscheduled lost profit opportunities by 47% by improving equipment reliability and streamlining their maintenance practices (Goel, 2004).

METHODOLOGY

Designing a system that is able to optimize RAM at this level, one must understand how to define the requirements of the system, how to find the initial reliability, how to find the optimum maintainability, the optimum availability, and finally, understand what the predecessor systems of this process were, and how they can be improved upon. The system requirements are necessary because one must first understand the goal and what the system is designed to do at the top level, and then breaking it down level by level, so that when building the system, one does not miss any steps or make a mistake. In the current system, that involves finding the initial reliability, which can be done by finding the overall failure rate, which can found by finding the individual failure rates of each node, which can be found by doing HAZOP and FTA analyses that help quantify each individual failure rate that makes up the combined failure rate. Once that is defined, and the initial reliability is found, one can use the initial reliability to determine the MTBF, the MTTF, and SAD for the Maintainability, figuring out what the optimum Maintainability is for this process in its current form (at a particular tier). That then allows the user to find the optimum availability by finding the inherent, achieved, and operational availabilities using the values of MTBF, MTTF, and SAD. Finally, the current model will improve on finding the optimum levels of each value because it will combine the sequential and simultaneous approaches currently used in most RAM optimization models to do a surrogate-based approach, commonly used in aerospace engineering optimization models to eliminate many of the disadvantages those individual models often possess.

System Requirements

To determine what system must be built, one must first define what the system must do. This involves defining the requirements of the system.

Defining the requirements, like the system itself, involves defining the "levels" or "tiers" of requirements, beginning with the top (overall) tiers- a.k.a. the customer requirements- and the bottom (base) tiers- a.k.a. the detailed requirements. The customer requirements involve the System of Systems requirements, which define what is required for the overall system of systems. They involve the Needs Requirements (a.k.a. what the entire System of System's needs) and the Originating Requirements (a.k.a. what needs to be done for the needs to be met). Once the System of Systems requirements are specified, requirements for individual systems, sub-systems, components, and if possible (though not always required), the configuration items (sub-components and parts, which can be planned, but since the requirements are evaluated very early in the process, that amount of detail is not always available at the time). These are called the "Detailed Requirements", and break down the requirements at each level (or "tier") of the overall system to determine how to make sure the Needs Requirements are met (Buede et al, 2016) (CJCSI 3170.01I, 2015).

Customer Requirements

For a good requirement to be written, the requirement must present a need, present information that is verifiable, present something that is attainable, and present the requirement with clarity. If a requirement states *how* instead of *what*, that is a bad it is not presenting a need, but a solution to a need. If the requirement describes operations, that is also a bad requirement because it is not expressing need, but a solution to that need. Any requirement that makes assumptions (or at least, bad assumptions) also fails to meet the criteria for a good requirement, as it is not verifiable. Finally, the requirements should be related to the overall system of systems and the system itself, and not over-specify by describing requirements of the sub-systems or more specific areas of the hierarchy (Kossiakoff et al, 2011) (Hitchens et al, 2007).

The customer requirements (seen in Figure 4) are broken into two parts: the mission (needs) requirements, and the originating requirements. The requirements build off each other to give a comprehensive look at the breakdown of requirements for the entire system to operate as planned. The mission requirements involve the *needs* associated with *missions* important to stakeholders, and the originating requirements involve the capabilities the system must have to meet those *needs* and fulfill those *missions*.

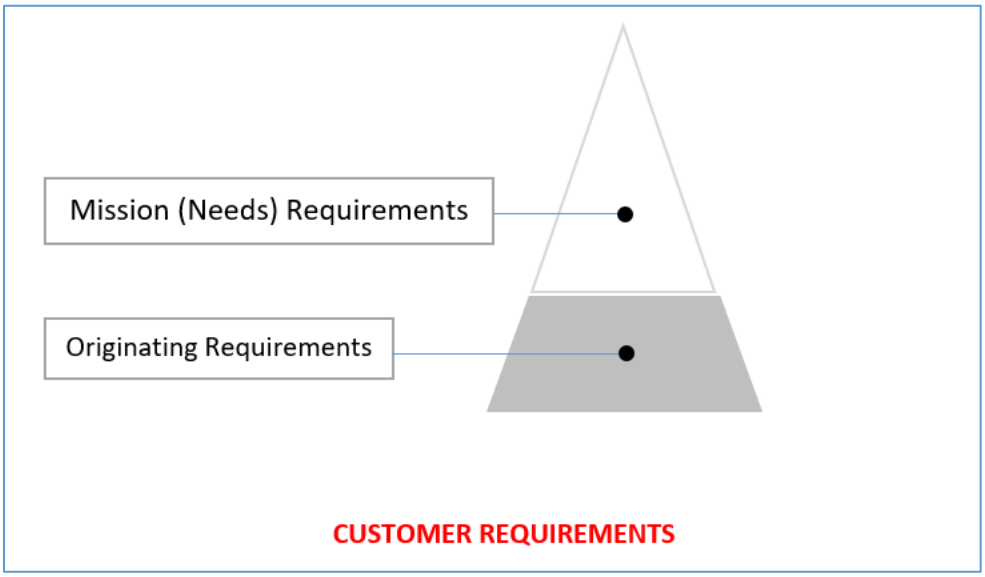

Figure 4: Visual Representation of Customer Requirements

Detailed Requirements

Once the Customer Requirements are defined, the Detailed Requirements (seen in Figures 5, 6, and 7) are written to help break down and understand the requirements of each sub-system, component, and configuration items requirements. Each one helps better understand what is required and what system controls, tweaks, and add-ons are needed to allow the system to meet its overall goal.

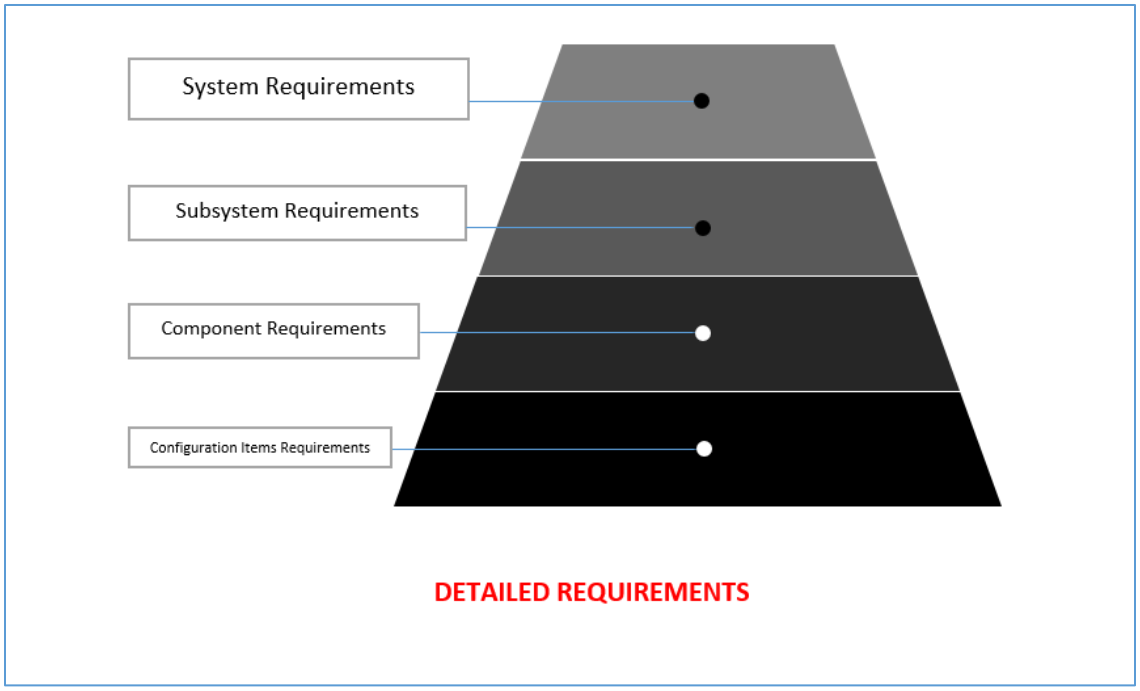

Figure 5: Visual Representation of Detailed Requirements

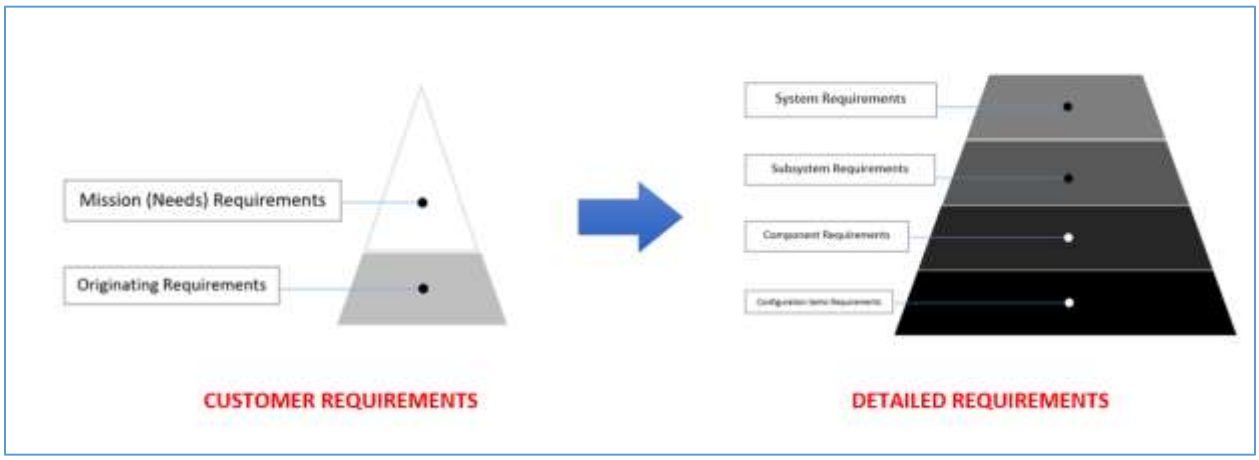

Figure 6: Visual Representation of Requirements, from Customer to Detailed Requirements

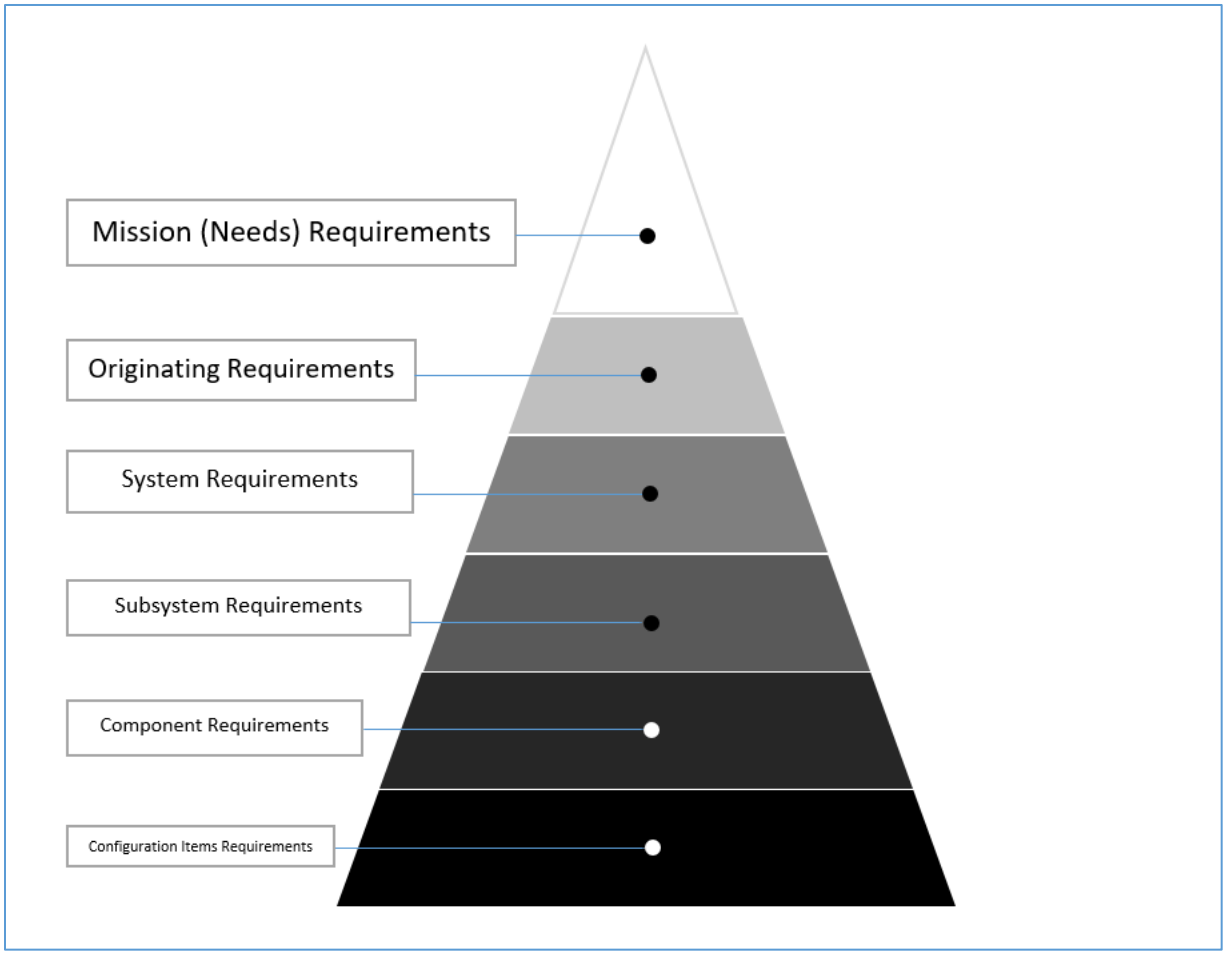

Figure 7: Visual Representation of Overall Requirements

Determining Initial Reliability/Determining Initial Failure Rate

Initial reliability can be found by determining the initial failure rate. The failure rate can be calculated by determining the overall risk, which can be determined by calculating, by definition, the frequency of occurrence for each hazard, multiplied by the consequence that occurs if the hazard results in a failure. To determine that, first, a Hazard Operating Analysis (HAZOP, Table 2 below) must be done, where the system is broken up into nodes, and potential failures for each node (consequence and frequency) are determined. Once all the potential failures are found, the total calculated failure rate is determined by doing both a fault tree analysis (FTA, seen in Figure 8), which breaks each failure into potential causes, gives each cause a value of frequency,

and determines the overall estimate frequency of that specific event (Ugurlu et al, n.d.). Next, an Event Tree Analysis (ETA, seen in Figure 9) may be done, which figures out what each failure may end up resulting in, helping to determine the overall consequence of the failure (Reliability Education, n.d.). Once the consequence and frequency of each failure is determined, it's placed into a Risk Matrix (Figure 10) to determine the quantifiable level of risk of failure (Queensland Government Chief Information Office, 2018).

Item	Study Node*	Possible Consequences	Action Required?
Sensor	Failure	-Crash, hitting an object	-Automatic Landing if sensor malfunction
Absorber	-Overpressure/ Explosion -Solid Object absorbed	-Loss of drone/possible deaths/injuries -Rupture/damage to drone	-Pressure Gauge/Pressure Relief Valve Required -Sensor to detect solid object/expulsion
Separator	-Failure -Rupture	-Drone does not do its job -Total pressure failure and drone could fall out of the sky	-Proper testing must be done -Insulate and protect Separator
Reactor	-Overheating -Overpressure	-Explosion or Fire -Explosion or Fire	-Insulator added -Pressure Gauge/Pressure Relief Valve Required
Engine	Failure	-Drone Falls out of sky	-Back-up engine required
Line	Blockage	-Overpressure/Fire or Explosion	-Back-up Line Required
Storage Tank	Rupture	-Total pressure failure and drone could fall out of the sky	-Insulate Storage Tank
Downward and Back Boosters	Failure	-Drone Falls out of sky	-Emergency boosters
Exterior	Rupture	-Total pressure failure and drone could fall out of the sky	-Reinforce and insulate exterior

*To the right of Study Node should be the columns for ***Process Parameters*** (such as flow, temperature, pressure, concentration, specifically identifying WHAT could cause the failure), and ***Deviations***

Table 2: Example HAZOP Table.

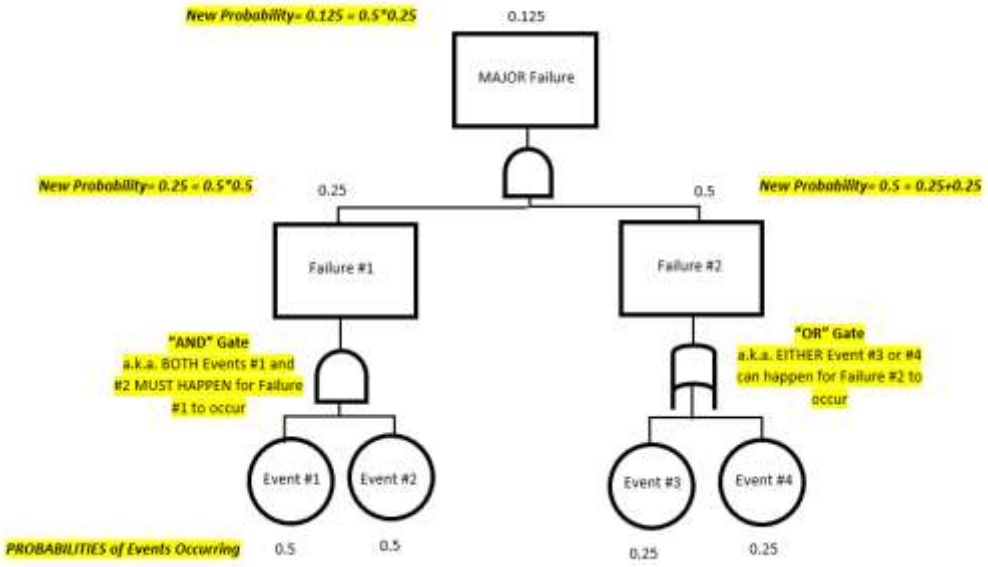

Figure 8: Example of FTA

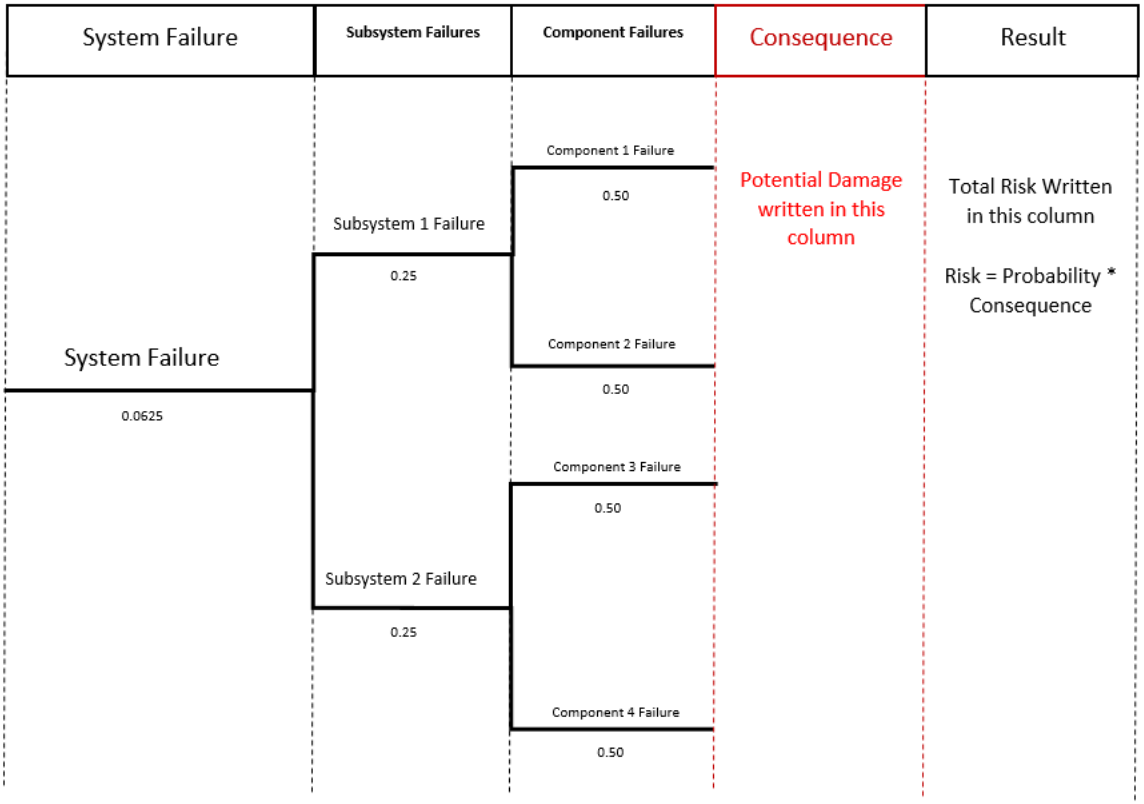

Figure 9: Example of ETA

	Consequence				
Likelihood	*Insignificant*	*Minor*	*Moderate*	*Major*	*Critical*
Rare	LOW	LOW	LOW	MEDIUM	HIGH
Unlikely	LOW	LOW	MEDIUM	MEDIUM	HIGH
Possible	LOW	MEDIUM	MEDIUM	HIGH	HIGH
Likely	MEDIUM	MEDIUM	HIGH	HIGH	EXTREME
Almost Certain	MEDIUM	MEDIUM	HIGH	EXTREME	EXTREME

Figure 10: Risk Matrix

Once the risk of failure is known at a single time, one can find the risk of failure over a distribution of time. Once failure over a distribution of time is known, reliability over that same distribution of time can be found through Eq. 1, which shows that as the failure rate decreases, the reliability increases. As a result, reaching the maximum reliability rate involves reaching the minimum failure rate.

Finding Optimum Maintainability

The optimum maintainability can be found by building a maintenance design program that creates commonality in the system between all parts, components, and subsystems, has a set of stowage requirements, and creates a logistics program/inventory to help make crucial information

regarding maintenance tools and parts easy to find and retrieve, as well as sticking to the renewal process within maintenance (rather than a minimal maintenance). Commonality involves creating and using common parts, procedures, and components between all sub-systems and systems to reduce time, errors, and unnecessary losses during maintenance. Optimizing stowage requirements includes maintaining a mass, volume, re-configurability, and operational space limit that allows for the easiest access to both maintenance tools and spare parts. And finally, inventory and logistics optimization involves logging, storing, and keeping like items in close proximity to where they're most likely to be needed to create a maximum rate of maintenance efficiency. By optimizing the commonality, stowage, and logistics, one can optimize maintainability, due to there being a reduction in the Mean Time Between Failures (MTBF), Mean Time To Failures (MTTF), Scheduled Administration Delays (SAD), and many other factors, which will also reduce the rate of failure ($\lambda(t)$), and as a result, reduce the F(t), which will therefore increase the R(t) and A(t) (ReliaSoft, n.d.).

$$MTBF = t_{med}\, e^{I(s)\wedge 2]/2} \quad \textit{(Eq. 7)}$$

$$T = t_0 + MTTR + SAD \quad \textit{(Eq. 8)}$$

$$MTTF = MTBF = \quad 1/\lambda \quad \Theta[1 + (1/\beta)] \quad \textit{(Eq. 9)}$$

Eqs. 7-9 are the equations for Maintainability to be used in surrogate-modeling program.

Commonality

Commonality can be increased by using the same parts among systems and sub-systems, same procedures across systems and sub-systems, and by creating an inventory and stowage requirement program to keep tabs on the entire operation's parts and procedures, and help gap unnecessary barriers. While perfect commonality will never be reached, the idea is not to create

perfect commonality, but simply to reduce un-commonality in places where it's not needed as much as possible. For example, when traveling across American states, while certain aspects of each state will be different, one can usually assume commonality in language (everyone will still speak English), currency (everyone will still use the American dollar), certain structural requirements (every town will have a governor, a legislature, a court system, a law enforcement system, a health system, an education system, etc), and while, for example, individual foods may be different, one can expect that every item of food is under FDA jurisdiction, and that either the food itself or the company that is providing it has met a certain base level of standards that are designed to make sure people who eat it don't get unnecessarily sick or die. In other words, back in American history when each state was allowed to create its own currency (under the Articles of Confederation), create its own laws involving how vigorous they were in inspecting food with no federal baseline, the lack of commonality caused mistakes to be made across states, such as people moving across states and finding out they no longer have money, or that their food is not safe. By creating commonality in an industry, one can make sure that, at the very least, a certain base level of unnecessary mistakes will not be made.

For example, by creating as much commonality in parts as possible, different components can be more easily diagnosed and have their parts replaced, saving on costs, time, and effort. In addition, if different parts of the subsystem are using different procedures, then when one has to fill in or take on the role of another sub-system, the procedures will change, and there are indirect costs.

Indirect costs include time being lost thanks to time being spent by employees identifying and learning the new procedure, increasing the average maintenance time. In addition, time will need to be spent by another member who knows the other part or procedure explaining it to the

ones learning it for the first time, costing in overtime or forcing the expert to spend time they could have used doing something more productive on helping to get a procedure done that the other members could have done themselves if there was commonality in the system.

Overall, by creating as much commonality as possible within each hierarchy of the system, the MTBF, MTBM (Mean Time Between Maintenance), Maintenance Delays, Scheduled Administrative Delays, etc would all decrease, resulting in a lower failure rate value, leading to higher R(t) and A(t).

Stowage Requirements/Inventory

In addition to having common parts, another way to increase maintainability is through stowage requirements and inventory management, which help reduce logistics problems, save time on retrieving and using maintenance tools and parts, and as a result, also help reduce the MTBF, MTBM, MTTF, and $\lambda(t)$, which all result in a higher maintainability value, higher reliability rate, and higher availability distribution with respect to time.

Renewal Maintenance vs Minimal Maintenance

Another major factor attributing to the overall optimization of maintainability of an item is whether or not the item is fixed to a renewable level ("as good as new") or a minimal level (functional). While fixing to a minimal level is often cheaper in the short term, in the long term, by not taking slightly extra time to fix the part or sub-component to full capacity (or as close to full capacity as possible at this stage of the item's industrial lifespan), one is wasting more time and resources overall, because that same item will fail a lot sooner, and more time an effort is going to be needed to get it back to the minimum level than it would have been to have had it at or beyond that level earlier by fixing it to full capacity. For example, if it takes 5 hours to fix to a

minimum, and 9 hours to fix to full capacity, and full capacity will lead to 1000 more hours of working properly while minimal will only work for another 200 hours, than adding all the maintenance that will needed to be done over the next 1000 hours (including the maintenance already done), the cost of full maintenance over those 1000 hours would've been 9 hours, and that's it. But if every 200 hours, another fix would need to occur, then it would be 5 hours at 0, 10 hours at 200, 15 at 400, 20 at 600, 25 at 800, and 30 at 1000, and an initial 4 hours saved at time zero will ultimately wind up being a 21 hour production loss (if it even stays steady the entire time and multiple maintenance renewals are not resulting in the failures to start to happen more often, which they likely will). And while this may depend on a case-by-case basis, overall, the entire maintenance process will be far more effective if the company sticks to the renewal of the item, not just maintaining it to a minimal.

Finding Optimum Availability

Once the optimum maintainability is found, the optimum availability can be found by determining the inherent availability, the achieved availability, and the operational availability. By adding these three availability components together, one can determine the overall availability, and to maximize the overall availability, one can maximize each of these components individually.

Inherent Availability

The inherent availability is also known as the "steady-state" availability because it is the time a component or part of the system is able to respond to the demands of the system.

$$A_{inh} = \frac{MTBF}{MTBF+MTTR} = \frac{Up\ Time}{Total\ Time} \quad \textit{(Eq. 10)}$$

The idea behind the inherent availability is that scheduled maintenance will not affect it, and that the only effect on this specific kind of availability is the maintenance caused by the failure

of the component. This gives the user a better idea of the reliability of the part or component, as it will allow the user to understand how often the availability of the part of component is failing (as opposed to being removed from the process due to planned removals).

To increase the inherent availability, one must increase the MTBF, which is best increased by reducing failures, which is done by increasing the reliability and focusing on the renewal process of maintenance over the minimum process. As stated before, less failures means less time between failures.

Achieved Availability

The Achieved Availability is calculated as seen below (Eq. 6):

$$\mathbf{A_a} = \frac{MTBM}{MTBM + M(avg)} \textbf{ (Eq. 11)}$$

MTBM = Mean Time between Maintenance

M (avg) = Mean System Downtime

To increase the achieved availability, one must increase maintainability by:

1. Reducing the amount of time in maintenance
2. Increase the time between maintenance by reducing the number of failures

Similar to the inherent availability, by optimizing R(t) and M(t), it will help optimize the achieved availability as well. This will decrease the λ(t), MTTF, and other factors directly involved in the overall failure rate of each part and component, and in the optimization system, will result in a different, more optimized value in the next iteration of the R(t) and M(t) values. The iterations will continue until the overall optimal point (the unchanging optimal point) is found. This is where the program that's being proposed comes in. While different methods have been used to try to find

the optimum point, the current program will go through each of these iterations for each part of RAM in a specific sequence, then from the bottom up, do it for each hierarchy of the system, while incorporating a surrogate-based model to help find the overall maximum point as opposed to a random local maximum.

Predecessor Systems

To build the best system possible, one must first look at previous systems, and understand what their advantages, flaws, and potential improvements to those systems are. By looking at the background of RAM Optimization models, as well as more current versions of optimization, one can see where RAM optimization models are behind and need improvements. Current RAM Optimization Models follow either a Sequential or Simultaneous approach, each of which having its own advantages and disadvantages. However, when looking at more current models in other industries, such as aerospace, models that combine the sequential and simultaneous approaches- to maximize the advantages and minimize the disadvantages- now exist in the form of surrogate-based modeling, and they show that improvements can indeed be made in RAM Optimization systems that could greatly increase a system's overall reliability, and in turn, dramatically increase a company's bottom line.

Models of Optimization

Sequential Approach

The sequential approach involves a step-based approach that optimizes the values in a specific order, then determines what the best values are based on that step-by-step maximizing process. The benefit of a sequential model is that, rather than beginning at a random starting point and finding the nearest local maximum, a value is found (the R(t) value) that makes it most likely to find the largest of the local maxima, helping to find the true optimum of the entire system.

However, there are still flaws, in that when the first iteration is done, and A(t) and M(t) are now different, just like in real life, the maximum calculated R(t) is now different based on the new values of M(t) and A(t), and a new, higher optimum value of R(t) may be found, and many sequential models currently being used either do not iterate or iterate a set number of times and only to the nearest local maximum, but do not have an expansive, mathematical method to finding the optimal maximum point throughout the entire range of the reliability, maintainability, and availability. As a result, there must be a way to potentially find the overall maximum, which is a problem the simultaneous approach tries to solve.

Simultaneous Approach

The simultaneous approach, which is much more prominent and avoids iterations, is being used more in modern industry than the sequential approach because it avoids timely iterations, and because it solves R, A, and M, in conjunction. The advantage to this approach is that that it finds a maximum that isn't secluded to a local area that depends on what the highest value of either R, A, or M are. However, there is a disadvantage, in that the simultaneous approach is generally found based on a generic starting point, and the optimum value that is found may or may not be the maximum optimum value, but another local optimum (and not even the local optimum that's highest based on the peak of R(t)). In addition, both the simultaneous and the sequential approaches are often run as a distribution for the system as a whole, simply because it would be too time consuming to run it on each component individually. As a result, an optimization system is needed that both has a starting point that makes sense, that calculates to the highest value of the local maximum values, and that has a way of iterating not just through R(t), M(t), and A(t), but through the different hierarchies of the system as well, from the bottom level to the top level.

Alternative Concepts

Surrogate-Based Modeling

Unlike the sequential or the simultaneous approaches, surrogate-based models, like space-mapping or kriging, combine both the sequencing and the conjunctive iterations of the two approaches, as well as the multi-layered function optimization calculation runs (Koziel et al, 2013), to find the best value overall for the entire system. Surrogate-based models are, by definition, a combined function (so, the combined functions of R(t), M(t), and A(t), which are put on a y-axis, or a Z-axis, if there are two independent values, acting as the dependent value function, vs one or two common independent values in each function, like λ(t), used as an X-value axis), in which a different independent value is chosen, a regression model is run until a local maximum is reach, and then, depending on how many different random points are selected (Keane, 2010). Several different random points are tested at different parts of the entire range given, and of all the local maxima found, the largest local maxima of the combined z-function is used as the optimum point, with a corresponding RAM-value given to let the user know at what R(t), A(t), and M(t) value he or she wants to keep the system at (and the parts, sub-components, etc., at) to keep the entire system functioning at a maximum possible level with the least amount of failures (Lee et al, 2017) (Hu et al, 2018) (Koziel et al, 2016) (Xu et al, 2013) (Gorisson et al, n.d.) (Iuliano et al, 2016).

Applying Space Mapping to RAM

To apply space-mapping to the current RAM model, one first must create a multi-layered functional program, which goes in a sequence, then uses iterations to find the optimum. The sequence will be R(t) first, then M(t), then A(t), because that is the most logical method. Next a tangential function will be used to determine if the function is at or near a peak (with z'= 0), and if z' is within 0.005 of 0 (+/- 0.005), then the program will terminate, because a peak will have been found. If not, the program will iterate, and continue to iterate, until a peak is found, and once

a peak is found, a two new starting points, each a select distance apart from the original starting point (depending on the number of runs asked for. If asked for 10 runs then each point will be $1/10^{th}$ of the distance from the starting point to the edge of the range, and will be run again, finding 21 peaks, and determining which is highest to determine the fully optimized point.

The x-variable will be $\lambda(t)$, as the failure rate is not only directly correlated to the optimum value of RAM, but because the goal is to reduce the failure rate as much as possible, and finding the optimum $\lambda(t)$ value on the same optimization function in which the optimum RAM is being found allows the user to "kill two birds with one stone."

In addition, there will be six function values, with one for each level of the hierarchy, and while ideally, each layer will produce results, that might be expensive and time-consuming, so only the parts and the full system results, at the very least, will need to be released, with a 3-D graph being produced, showing how the failure rate directly affects both. However, again, sequentially, the program shall run the optimization from the bottom-up, calculating the optimum RAM for the parts first, then for the sub-components, then components, all the way up to the system of systems (or system, can work as a 5 layer as well) (Leifsson et al, 2006).

Types of Surrogate Models

Kriging

Kriging is a regression-based model that interpolates a value across a function to find the optimum, finding several specific points and determining at what point the maximum function is. There are two kinds of kriging, regular and Gaussian-enhanced, with Gaussian enhanced determining the optimum value based on the tangential function (once the tangent is within a specified limit approaching 0, or a peak value/change in direction, the kriging function determines that the optimum point, or either maximum or minimum has been reached, and assigns the z-

function, or output on the dependent axis, as the optimum). It also has the ability to be done at multiple layers, making it perfect for an optimization model like the one that's needed to solve the entire system layer hierarchy for RAM. Unfortunately, there are some flaws to kriging, such as that if enough points aren't chosen, it may end up finding the wrong value or even a local minimum value. To avoid this problem, another method that is useful is space-mapping, which specifically combines both sequential modeling and simultaneous modeling to determine which arrives at the highest local maximum. (Cressie, 1989)

Space-Mapping

Space mapping adds a sequential function and adds another level of depth (to make it a multi-scale model) to kriging, which allows it to create multiple 2-D functions over a plane to determine where the overall optimum of the entire system is. Unlike kriging, space mapping is a surrogate-model that covers the entire range of the system (Crevecoeur et al, 2006), which calculating optimums from equally separated and spread out intervals, which will make it far more likely that the optimum will be found. In addition, space-mapping includes a third function layer, which allows the system to compare each tier of the system to the next and find the optimum at every tier (or the most important tier, based on the other tiers) (Crevecoeur et al, 2008). That is exactly what is required out of the system that is being built, and as a result, space-mapping is undoubtedly the most relevant system to use as an alternative concept to the current systems. Finally, and most importantly, space-mapping, like kriging and other surrogate models, is also relatively inexpensive, and plots its results, allowing the user to see the entire three-dimensional slope plane of the function and determine whether or not the optimum points have been found, adding an extra degree of simplicity to the user's experience in using the model.

Validation

Finally, once the concept has been developed to address all the requirements, one must ensure that the final system is valid in meeting the requirements, and have a method to test the system to ensure its validity. (Social Research Methods, n.d.) To do so for this system, like with any optimization system, one must test the result of the system against a known setting (with a known optimum), and calculate the difference in the results to determine how often (or how close) the system comes to meeting the optimum each run. Doing so involves a statistical analysis that involves finding the z-value of the optimized model compared to a real-life sample, and then comparing those z-values and their accuracy against the current RAM models being used.

As with validation for any system, one must run a statistical check (Social Research Methods, n.d.) to determine the degree of confidence to which the results obtained by the system translate to real life settings. Since no model is every going to be 100% accurate 100% of the time, one must determine which has a higher percentage of accuracy a higher percentage of the time. The goal of the validation process is to prove that the current model has a higher percentage of accuracy, and does so consistently (a.k.a. precisely) more often than current RAM models. To do so, one must determine the confidence intervals, a.k.a. the intervals that predict how close the models are to reaching the desired results, of each optimization system (both the new one, and the current ones).

To create a confidence interval, one must first compare the results from each optimization model must to the results from an actual system. The number of runs for both (n) will be pre-selected to ensure that the average (μ) and standard deviation (σ) are within the margin of error. The equations to find the average and standard deviation are as follows.

$$\boldsymbol{u = \sum \frac{yi}{n}} \textbf{ (Eq. 12)}$$

$$\sigma^2 = \sum \frac{(\mu - yi)^2}{n} \textit{ (Eq. 13)}$$

Where y_i equals each individual y-value for the model, n is equal to the total number of values, μ is equal to the average, and σ is equal to the standard deviation.

Once the calculated values are found, one must then find how close the μ of each model compares to the actual μ by find the average z value for each model.

$$\textit{Confidence Interval} = (y(avg) - u) \pm z\alpha(\frac{\sigma}{n^2}) \textit{ (Eq. 14)}$$

$$\mathbf{D} = y(avg) - u \textit{ (Eq. 15)}$$

$$\mathbf{E} = z(\frac{\sigma}{n^2}) \textit{ (Eq. 16)}$$

Where the z value is the confidence standard deviation for the error of the plot (for a 95% confidence range, $z_\alpha = 1.96$ if $\alpha = 0.05$; $1\text{-}\alpha = 0.95$ for confidence error from the mean). To determine which model is the most accurate and precise model, one must find the model with the lowest D and the lowest E. In addition, current RAM models will also have to be tested as a control group, testing the D and E values to see how accurate and precise they are. The ultimate goal will be to create a model that is more accurate (has a lower D), and more precise (has a lower E) than the most prominent current RAM models.

It is hypothesized that this current model, if built, will indeed be much more accurate and precise than current RAM models, because current RAM models often don't find the overall optimum. Current RAM models will often find local maxima (or minima), resulting in their y-values being far more spread out in Eqs. 14-15, making them far less accurate and far less precise than the new, more robust model. By using space-mapping to find the optimum, the y-values will

be much more likely to be evenly spaced, to find the optimum almost every time, and as a result, will have values that are much more consistently in line with the overall average, and much more consistently in line with the real-life values. As a result, their accuracy will be higher, their precision will be higher, and if these prove to be true, the next step of optimization will likely have been achieved.

RESULTS

As stated in the experimental section, to make this system come to life, one must first do a detailed breakdown of the requirements needed to build the system, and then use those requirements to create a visual representation of the system that provides both a blueprint for the building of the system, as well as gives the audience a visual guide to be able to follow how to use the system to their benefit.

Customer and Detailed Requirements

Once the customer requirements are defined, and the detailed requirements are broken down from the customer requirements, it is necessary to find a way to validate the system by finding a solution to the problems represented in the requirements. The requirements are designed to specify the problems and what needs to be done to meet the goals of the experiment.

Designing the Requirements

To design the requirements, first, the tiers must be defined:

System of Systems *= Entire Plant*

System *= Optimization Program*

Sub-Systems *= Each Individual Function Program*

Components *= Each Individual Task of Each Function Program*

Sub-Components *= Each Individual Line of Command*

Parts *= Each Individual Portion of Commands*

And once the tiers are defined, the requirements can be written:

Customer Requirements

Needs Requirements

1 System of Systems shall minimize the number of deaths, injuries, and structural damage on the job (maximizing profits)

Originating Requirements

1.1 System of Systems shall minimize the probability and consequence (a.k.a. risk) of the average failure incident

Detailed Requirements

System Requirements

1.1.1 System shall maximize reliability for all 6 tiers.

1.1.2 System shall be able to calculate failure rate based on provided information.

Sub-System Requirements

1.1.1.1 Sub-system shall be able to maximize maintainability for all 6 tiers

1.1.1.2 Sub-system shall be able to maximize availability for all 6 tiers

1.1.2.1 Sub-system shall provide user with an interface where HAZOP information is still provided

1.1.2.2 Sub-system shall have a method of calculating an average estimated probability and consequence for each potential HAZOP incident

Component Requirements

1.1.1.1.1 Component shall maximize MTBF for all 6 tiers.

1.1.1.1.2 Component shall maximize MTTF for all 6 tiers.

1.1.1.1.3 Component shall maximize MTTF for all 6 tiers.

1.1.1.2.1 Component shall maximize inherent availability for all 6 tiers.

1.1.1.2.2 Component shall maximize achieved availability for all 6 tiers.

1.1.1.2.3 Component shall maximize operational availability for all 6 tiers.

1.1.2.1.1 Component shall provide the ability for user to type in each node and problem.

1.1.2.1.2 Component shall provide instructions on how to complete each step of HAZOP in a simple way.

1.1.2.2.1 Component shall calculate average based on algorithm involving past five years.

1.1.2.2.2 Component shall be able to calculate average based on world statistics (embedded in database)

All the requirements are listed together in Table 3 below.

Customer Requirements		Detailed Requirements		
Mission Requirements	**Originating Requirements**	**Systems Requirements**	**Sub-systems**	**Components**
System of Systems shall minimize the number of deaths, injuries, and structural damage on the job (maximizing profits)	*System of Systems shall minimize the probability and consequence (a.k.a. risk) of the average failure incident*	System shall maximize reliability for all 6 tiers.	*Sub-system shall be able to maximize maintainability for all 6 tiers*	Component shall maximize MTBF for all 6 tiers.
				Component shall maximize MTTF for all 6 tiers.
				Component shall maximize MTTF for all 6 tiers.
			Sub-system shall be able to maximize availability for all 6 tiers	Component shall maximize inherent availability for all 6 tiers.
				Component shall maximize achieved availability for all 6 tiers.
				Component shall maximize operational availability for all 6 tiers.
		System shall be able to calculate failure rate based on provided information.	*Sub-system shall provide user with an interface where HAZOP information is still provided*	Component shall provide the ability for user to type in each node and problem
				Component shall provide instructions on how to complete each step of HAZOP in a simple way
			Sub-system shall have a method of calculating an average estimated probability and consequence for each potential HAZOP incident	Component shall calculate average based on algorithm involving past five years
				Component shall be able to calculate average based on world statistics (embedded in database

Table 3: Breakdown of Requirements

Input/Output Requirements

Once the overall requirements are defined, one can determine the "Input/Output Requirements", which define what will be placed into the system, followed by what will come out. The Input/Output requirements of this system are in Table 4 below.

Input	Output
Failure Possibilities (#) **Values for Failure Possibilities (Probability and Consequence)** **Solutions to Failures** **Cost of Solutions** **Effectiveness of Solutions**	Total Failure Rate (at each tier) Total Reliability (at each tier) Total Maintainability (at each tier) -MTTF (at each tier) -MTBF (at each tier) -SAD (at each tier) Total Availability (at each tier) -A_{inh} (at each tier) -A_a (at each tier) -A_o (at each tier) Total Cost of Failures ($) Total Cost of Potential Updates ($) Total Benefit of Potential Updates ($)

Table 4: Input/Output Requirements.

Visual Representation of System

Once the requirements are specified, one can start building the system. To do that, one must first create a blueprint of how the system will work. That means creating diagrams. To begin building this system, the diagrams of how the individual sub-systems and components would interaction, what sequence they would follow, how they would be "packaged" and the overview of the entire program, was shown. By creating an Interaction-Overview/Package diagram to define, in detail, the interactions of the sub-systems and components of the system, followed by a

packaging of the tiers, one can create a blueprint from which to build the system off of. Then, by doing the Sequence Diagram, one can see how the timing of the system would work, and go a step further in building the system by understanding what steps must be taken and what potential flaws in timing there may be.[36] Overall, creating a visual representation of the system is the next step towards making the system come to life, both by creating a blueprint to build the system, and by providing the audience a visual guide to understand how to use the system to their benefit.

Interaction-Overview/Package Diagram

To provide a blueprint for the building of the program, one must understand which sub-systems and parts are aligned the closest, and how they will interact, so that all the right connections and combinations are done and no steps are missed in completing the system. The The Interaction-Overview/Package diagram (Figure 13) combines the Interaction-Overview (Figure 11) and Package Diagrams (Figure 12) to create an overview of the entire system, "packages" each part of the system and its sub-systems, and shows how they all interact. Below are two examples

of a Package diagram and an Interaction-Overview diagram, and after that, an example of the combined Interaction-Overview/Package diagram for the optimization system.

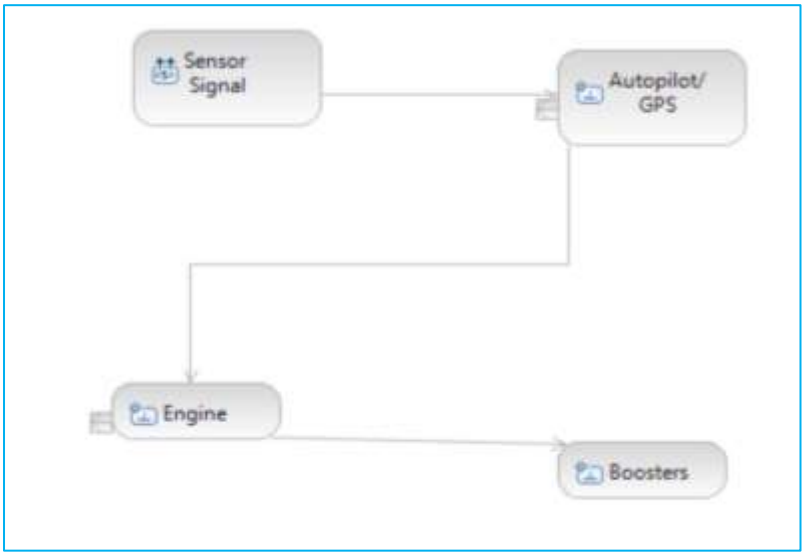

Figure 11: Example of Interaction-Overview Diagram for a Drone

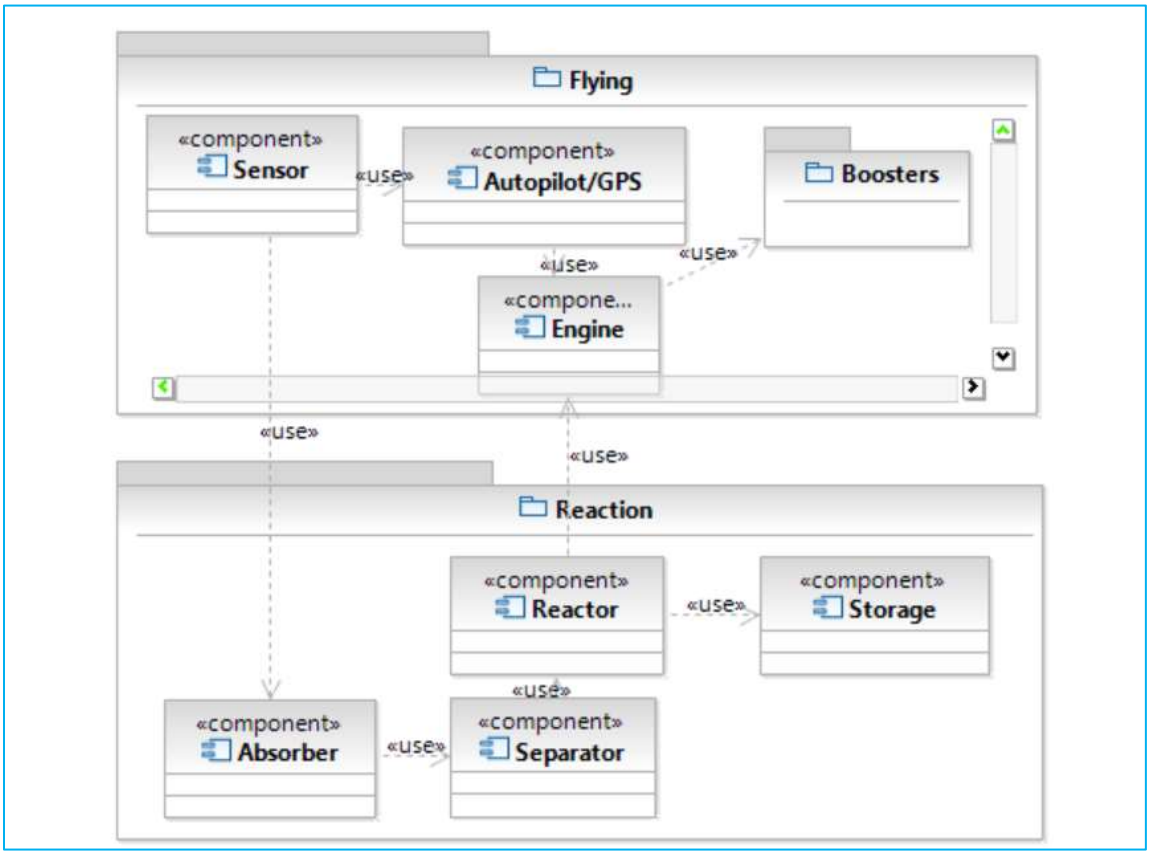

Figure 12: Example of Package Diagram for Same Drone

As Figure 13 shows, each of the 6 tiers provides its own "package" that the three R(t), M(t), and A(t) systems interact within, before sending the results up to the next tier. Starting from the inputs the user provides, the system sends the information to the R(t) component of the Tier 1 sub-system, resulting in a calculated value for the R(t) and $\lambda(t)$, that then is sent to the M(t) component to find the MTBF, MTTF, and SAD, which is then sent to the A(t) to find the A_{inh}, A_a, and A_o in a space-mapping style of optimization. Once this is done, the process is iterated using the new information, until the optimum for all three values are found, before sending it up a tier, repeating the process until the final results are found and the outputs are given to the user. This all goes back to the Bayesian Network, in that the RAM values in the lower tiers of the system affect the RAM values in the upper tiers. The current diagram shows the Bayesian Network of the entire system, showing exactly levels of the system, and which factors within each level, depend on each other.

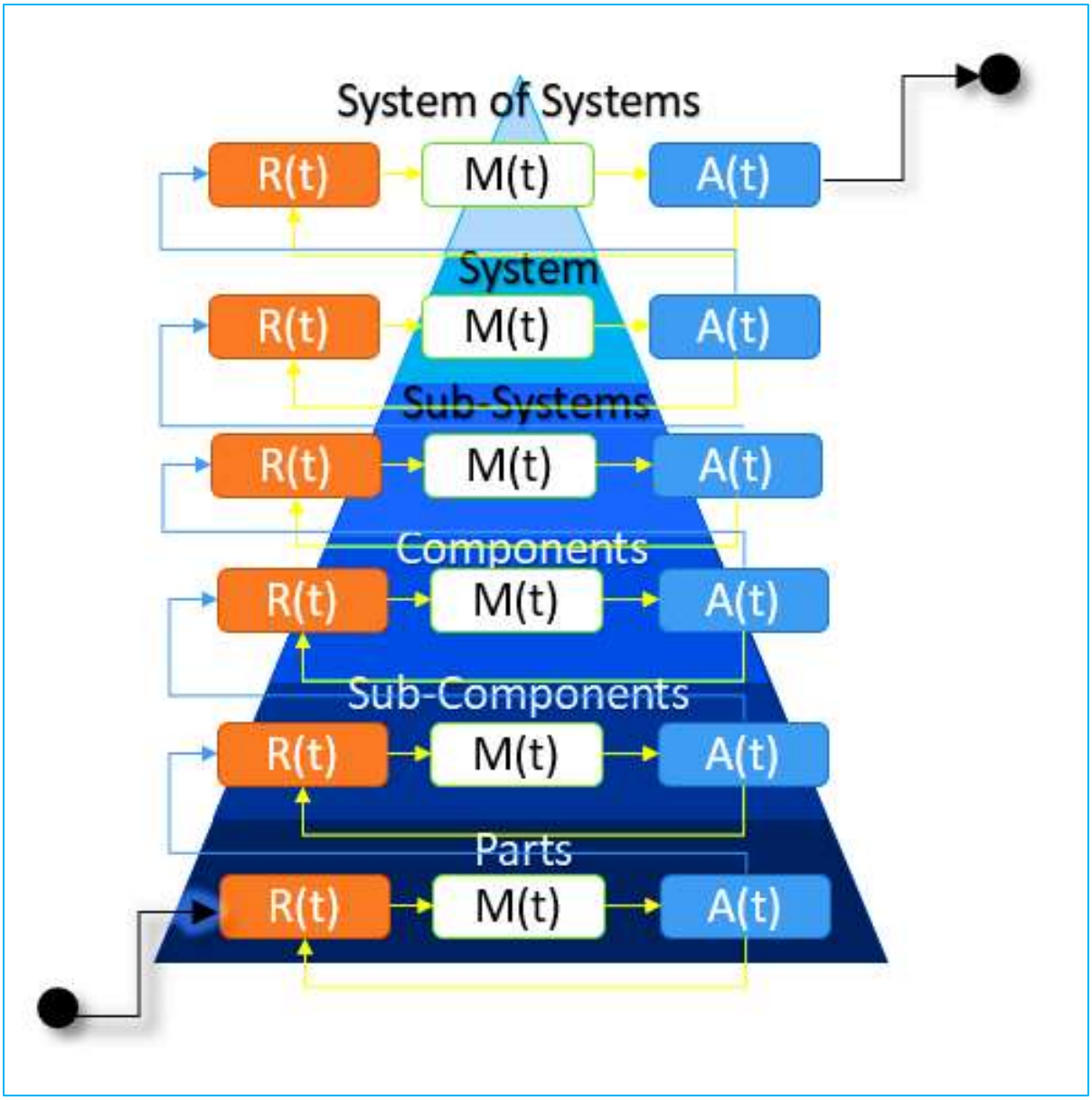

Figure 13: Interaction Overview/Package Diagram for Optimization System

Sequence Diagram

In addition to understanding how the system is built and interacts, the timing of the system is also important to understand, and for this reason, the Sequence Diagram is important. The sequence diagram (Figure 14) shows which parts may overlap, which parts work individually, and allows the user to see where repetitions occur, where time can be saved, and in this particular system, how Tiers 1-6 all have nearly identical processes. While much of the information is similar

to the diagram in Figure 13, this provides another blueprint to the user and a visual aid to the audience that shows that creating and R(t), M(t), and A(t) system at one tier could effectively streamline the entire process, and creating it at all the other tiers would require very little adjustments beyond determining exactly how to send information from one tier to the next. Overall, the Sequence Diagram is another visual representation, that here shows there may be a quick way to build this program by looking at how the process between each tier is virtually identical. Combined with the Interaction-Overview/Package Diagram, this provides a start for the blueprint for the overall system, with the next step being to develop more blueprints to build the entire system, including breaking down each part into its surrogate models. Again, similar to how a Bayesian Network shows which parts of a system affect the others, the sequence diagram shows the sequence that the Bayesian Network follows.

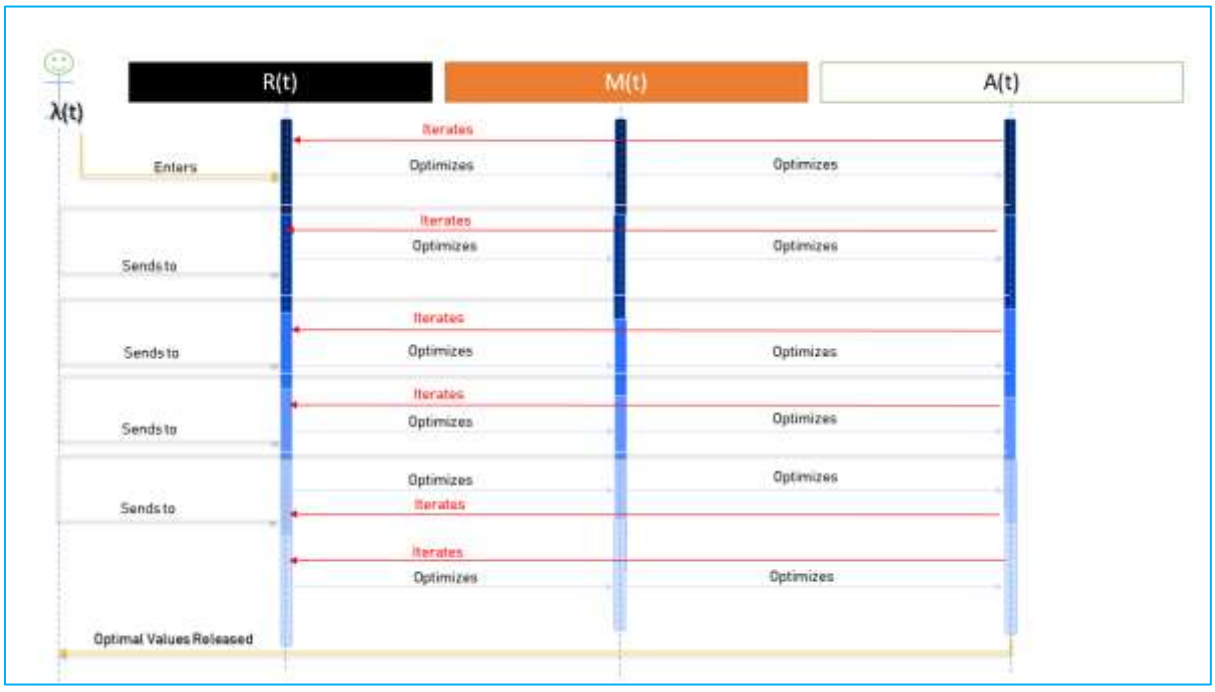

Figure 14: System Sequence Diagram for Optimization System

Other Diagrams

Other figures also allow visual representations to be made, helping to lay out the entire, overall blueprint of the system, what classes it has (to understand what aspects/components depend on which other aspects/components), what states it has and how the system goes from one state to the next, and what activities the system will be expected to perform. The class diagram (see Figure 15) helps explain the functions of the system and the hierarchy, giving the user an idea of how the overall structure of the system is built. The state-machine diagram (see Figure 16) mentions each state the system could be in, and how each state can move from one to the next. And finally, the activity diagram (see Figure 17) shows which activities the system will be expected to perform, giving the user an idea of all the different functions the system must be capable of and how it must be able to switch from performing one function to the next. Overall, these diagrams are a key step in building the final blueprint of the system, before process integration can begin, and the system can begin to be developed.

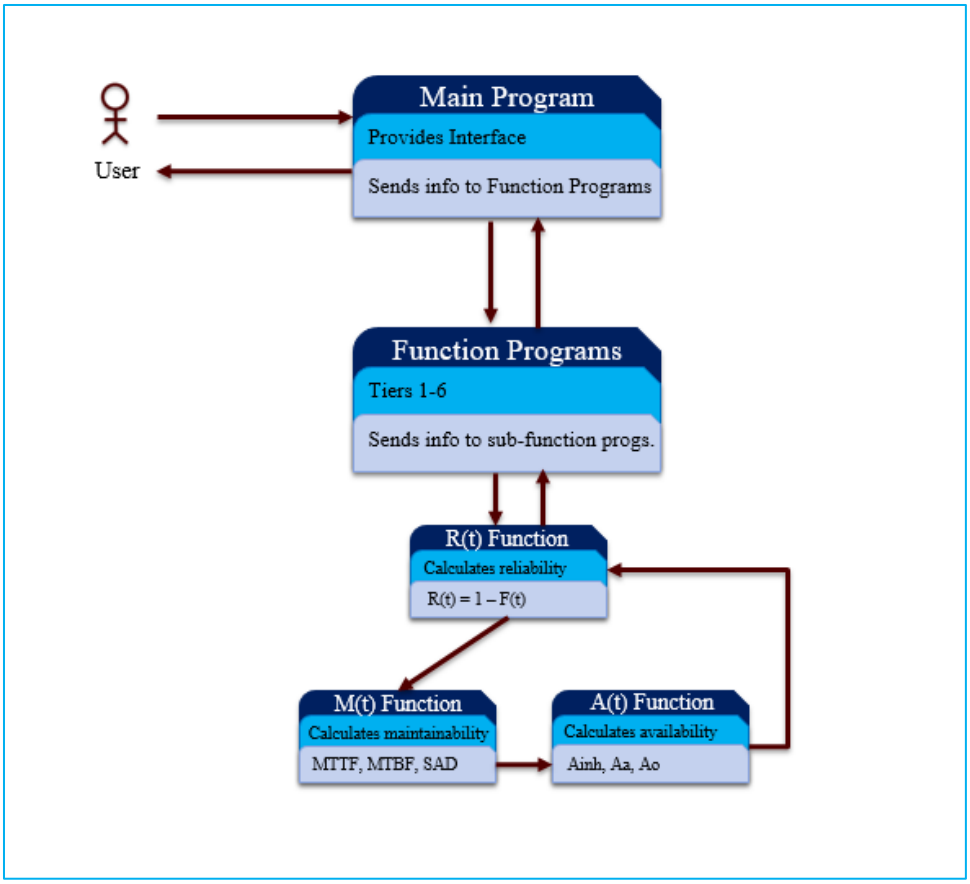

Figure 15: Class Diagram for Optimization System

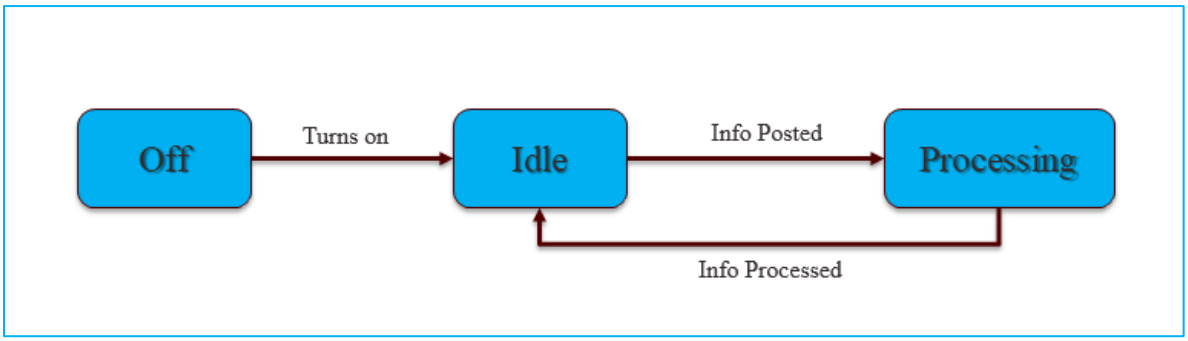

Figure 16: State-Machine Diagram for Optimization System

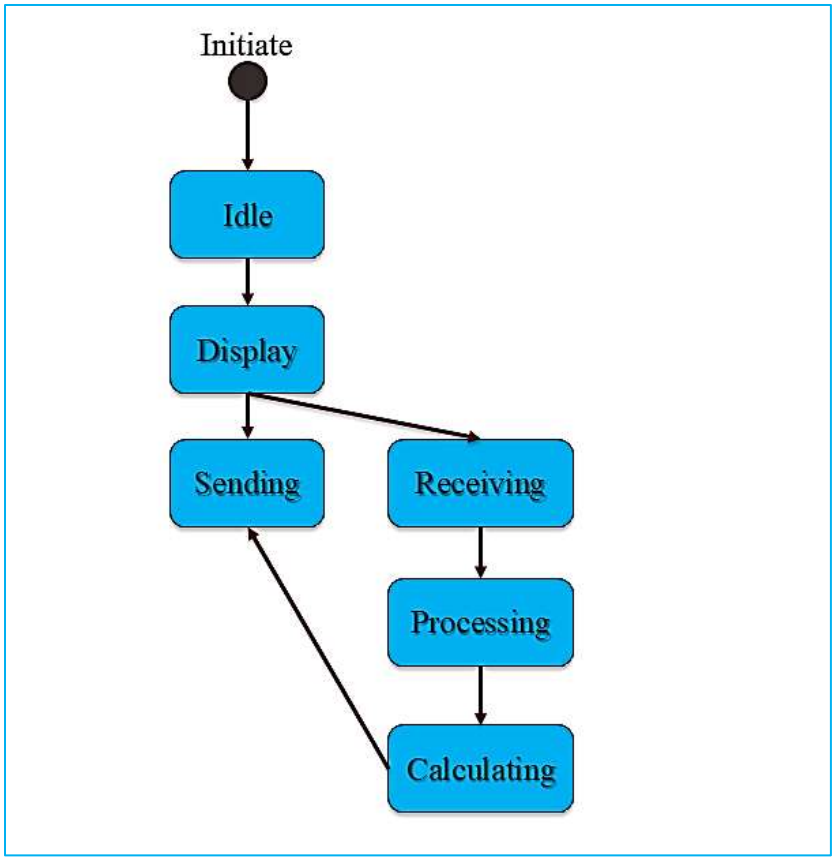

Figure 17: Activity Diagram for Optimization System

CONCLUSION

Overall

Overall, if the following program is made, it would allow companies to cost-effectively optimize RAM by combining current optimization methods, solving RAM optimization at the smallest level of a system hierarchy (the parts), and being able to cover the entire range of potential values for RAM and $\lambda(t)$. The program would be able to create an optimization value that will reduce the potential failures to its minimum, and be the least costly in terms of both direct and indirect costs to the customer, as well as show the customer where the best changes to be made are at the deepest levels of the system (down to the systems' parts) through a set-up of a Bayesian network, as opposed to previous models, which could not provide exactly which exact parts were leading to the exact drops in overall failures within the system. To prove that this is the best system, it will need to be tested against current optimization systems, both other RAM systems, as well as other surrogate-model and space-mapping systems outside of RAM to make sure that the program is transitioning smoothly to being used for RAM, as well as meeting its objective to reduce indirect costs to companies and boost not just their safety and failure incident records, but also improve the overall quality of their company.

Next Steps

The next steps to building this program are to build each individual part of the system on a proven software system that can accurately build surrogate-models. Likely using MATLAB to build the model, the $\lambda(t)$ would need to be calculated by hand, and inserted, allowing the function to run. In addition, bugs would need to be worked out for each function file of the program, and then bugs between communicating files would need to be worked out to allow the function to complete its job. Finally, once the program is tested on actual systems and plants against previous

models, finding the confidence interval of the models, and allowing researchers to know whether or not the system is meeting its goal of being a more precise, more accurate optimization model that helps find the correct optimization standard within an actual system more often than previous models in RAM have been able to. Overall, if all those steps are met, then by creating this program, and making it available to as much of the public as possible, not only can a company lower its own personal risk within its entire system of systems and reduce its own hidden costs, thus heavily increasing its revenue, but the industrial risk in all of human society may potentially be reduced to a far lower point than it is currently at, saving millions of lives and millions of dollars for the overall global economy, increasing the standard of living for everyone.

REFERENCES

1. Asfahl, C. Ray, and David W. Rieske. Industrial Safety and Health Management. Pearson, 2012.

2. United States Department of Labor. Occupational Safety and Health Administration, www.osha.gov/dcsp/success_stories/compliance_assistance/abbott/abbott_casestudies/slide16.html.

3. Barstow, David, and Lowell Bergman. “A Family's Fortune, a Legacy of Blood and Tears.” The New York Times, The New York Times, 8 Jan. 2003, www.nytimes.com/2003/01/09/us/a-family-s-fortune-a-legacy-of-blood-and-tears.html.

4. Champion, John, Sheila Van Geffen, and Lynnette Borrousch. "Reducing Process Safety Events: An Approach Proven by Sustainable Results." Process Safety Progress 36.4 (2017): 326-37. Print.

5. “10 Essentials of McWane's Culture Change.” ISHN RSS, www.ishn.com/articles/89805-10-essentials-of-mcwanes-culture-change.

6. “Company-Histories.com.” McWane Corporation -- Company History, www.company-histories.com/McWane-Corporation-Company-History.html.

7. “Dangerous Business: Coverage by David Barstow and Lowell Bergman.” The New York Times, The New York Times, topics.nytimes.com/top/news/national/series/dangerousbusiness/index.html.

8. Charles E. Ebeling, An Introduction to Reliability and Maintainability Engineering, 2nd ed., Waveland Press, Inc., 2010

9. “Available.” Dictionary.com, Dictionary.com, www.dictionary.com/browse/availability.

10. “Reliability.” Merriam-Webster, Merriam-Webster, www.merriam-webster.com/dictionary/reliability.

11. “What Is Maintainability? Definition and Meaning.” BusinessDictionary.com, www.businessdictionary.com/definition/maintainability.html.

12. "Reliability." Social Research Methods – Knowledge Base – Reliability, socialresearchmethods.net/kb/reliable.php.

13. "Measurement Validity Types." Social Research Methods – Knowledge Base – Measurement Validity Types, socialresearchmethods.net/kb/measval.php.

14. DoD Guide for Achieving Reliability, Availability, and Maintainability, August 2005, accessed August 30, 2017.

15. "Maintainability Analysis." ReliaSoft, www.reliasoft.com/products/reliability-analysis/blocksim/maintainability-analysis.

16. "Reliability, Availability, and Maintainability." The MITRE Corporation, 26 Sept. 2017, www.mitre.org/publications/systems-engineering-guide/acquisition-systems-engineering/integrated-logistics-support/reliability-availability-and-maintainability.

17. Department of Defense, June 2009, Reliability, Availability, Maintainability, and Cost Rationale Report Manual, accessed August 30, 2017.

18. Modarres, M., M. Kaminskiy, V. Krivtsov, Reliability Engineering and Risk Analysis, 2nd ed, Taylor & Francis, 2010 (Modarres, RERA)

19. Jordaan, I., Decisions under Uncertainty, Probabilistic Analysis for Engineering Decisions, Cambridge University Press, 2005 (Jordaan, DUU)

20. Modarres, M., Risk Analysis in Engineering, Taylor&Francis, 2006 (Modarres, RAE)

21. Ayyub, B.M., Risk Analysis in Engineering and Economics, Chapman & Hall/CRC, 2003.

22. Making Acute Risk Decisions with Chemical Process Safety Applications, CCPS, AIChE, 1995 (AIChE, 1995)

23. Ang, A.H-S. and Tang, W.H., Probability Concepts in Engineering, 2nd ed, Wiley, 2007 (Ang, PCE)

24. Ang, A. H-S. and W.H. Tang, Probability Concepts in Engineering Planning and Design, Vol 2, Wiley, Decision, Risk, and Reliability, 1990 (Ang, PCEPD)

25. Hubbard, D.W., The Failure of Risk Management, Wiley, 2009

26. Hubbard, D.W., How to Measure Anything, 2nd ed, Wiley, 2010

27. Pasman, Hans J., Risk Analysis and Control for Industrial Processes-Gas, Oil and Chemicals: A System Perspective for Assessing and Avoiding Low-Probability, High Consequence Events, Elsevier, 2015

28. Barnard, R.W.A. (2008). "What is wrong with Reliability Engineering?". Lambda Consulting. Retrieved 5 October 2018.

29. Goel, Harish. "Integrating Reliability, Availability, and Maintainability (RAM) in Conceptual Process Design." AIChE.

30. Distefano, Salvatore, and Antonio Puliafito. "Reliability and Availability Analysis of Dependent–Dynamic Systems with DRBDs." Reliability Engineering & System Safety, Elsevier, 20 Feb. 2009, www.sciencedirect.com/science/article/pii/S0951832009000325

31. "Reliability & Validity." Social Research Methods – Knowledge Base – Reliability & Validity, socialresearchmethods.net/kb/relandval.php.

32. Buede, Dennis M., and William D. Miller. The Engineering Design of Systems: Models and Methods. Wiley, 2016.

33. CJCSI 3170.01I, January 23, 2015, Joint Capabilities Integration and Development System, accessed August 30, 2017.

34. Quanterion Solutions Incorporated, 2015, System Reliability Toolkit-V: New Approaches and Practical Applications, accessed August 30, 2017.

35. Kossiakoff, Alexander. Systems Engineering Principles and Practice. John Wiley & Sons, 2011.

36. Hitchins, Derek K. Systems Engineering: a 21st Century Systems Methodology. John Wiley, 2007.

37. Christensen, R., W. Johnson, A. Branscum, and T.E. Hanson, Bayesian Ideas and Data Analysis, An Introduction for Scientists and Statisticians, Taylor & Francis, 2011

38. Neapolitan, R.E., Learning Bayesian Networks, Pearson Prentice Hall, 2004

39. Mitchell, Toby J, and Max D Morris. “Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction.” Taylor & Francis, Statistic Sinca, amstat.tandfonline.com/doi/abs/10.1080/00401706.1993.10485320.

40. Uğurlu, Özkan, et al. “Risk Analysis Of The Fire And Explosion Accidents In Oil Tankers.” ResearchGate, 5th International Conference on Maritime Transport.

41. “Event Tree Analysis.” Reliability Education, ITEM Software, www.reliabilityeducation.com/intro_et.html.

42. Queensland Government Chief Information Office. “ICT Risk Matrix.” Queensland Government Chief Information Office, Queensland Government Chief Information Office, 21 Mar. 2018, www.qgcio.qld.gov.au/information-on/ict-risk-management/ict-risk-matrix.

43. Lee, Yong Hoon, et al. “A Multiobjective Adaptive Surrogate Modeling-Based Optimization (MO-ASMO) Framework Using Efficient Sampling Strategies.” Volume 2B: 43rd Design Automation Conference, 2017, doi:10.1115/detc2017-67541.

44. Iuliano, Emiliano. “Adaptive Sampling Strategies for Surrogate-Based Aerodynamic Optimization.” Springer Tracts in Mechanical Engineering Application of Surrogate-Based Global Optimization to Aerodynamic Design, 2016, pp. 25–46., doi:10.1007/978-3-319-21506-8_2.

45. Gorissen, Dirk, et al. A Surrogate Modeling and Adaptive Sampling Toolbox for Computer Based Design. Journal of Machine Learning Research

46. Xu, Qian, et al. “Knowledge-Based Surrogate Modeling in Engineering Design Optimization.” Surrogate-Based Modeling and Optimization, 2013, pp. 313–336., doi:10.1007/978-1-4614-7551-4_13.

47. Koziel, Slawomir, and Leifur Leifsson. “Introduction to Surrogate Modeling and Surrogate-Based Optimization.” Simulation-Driven Design by Knowledge-Based Response Correction Techniques, 2016, pp. 31–61., doi:10.1007/978-3-319-30115-0_4.

48. Hu, Zhen, and Zissimos Mourelatos. “Efficient Global Surrogate Modeling Based on Multi-Layer Sampling.” SAE Technical Paper Series, 2018, doi:10.4271/2018-01-0616.

49. Keane, Andy. “Surrogate-Based Optimization.” Encyclopedia of Aerospace Engineering, 2010, doi:10.1002/9780470686652.eae496

50. Crevecoeur, G., et al. “EEG Source Analysis Using Space Mapping Techniques.” Journal of Computational and Applied Mathematics, vol. 215, no. 2, 2008, pp. 339–347., doi:10.1016/j.cam.2006.03.058.

51. Crevecoeur, G., et al. “Reconstruction of Local Magnetic Properties of Steel Sheets by Needle Probe Methods Using Space Mapping Techniques.” Journal of Applied Physics, vol. 99, no. 8, 2006, doi:10.1063/1.2176321.

52. Leifsson, Leifur, et al. “Multidisciplinary Design Optimization of Low-Airframe-Noise Transport Aircraft.” 44th AIAA Aerospace Sciences Meeting and Exhibit, 2006, doi:10.2514/6.2006-230.

53. Koziel, Slawomir, and Leifur Leifsson. “Multi-Level CFD-Based Airfoil Shape Optimization with Automated Low-Fidelity Model Selection.” Procedia Computer Science, vol. 18, 2013, pp. 889–898., doi:10.1016/j.procs.2013.05.254.

54. Cressie, N. A. C. *The Origins of Kriging*. Iowa State University. Department of Statistics. Statistical Laboratory, 1989.