A Partitioning Methodology for Helicopter Avionics System with a focus on Life Cycle Cost

by

Leon M. Silva

B.S. Electrical Engineering, University of New Haven, 1991

Submitted to the System Design and Management Program
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management
at the
Massachusetts Institute of Technology

January 2001

Signature of Author____
System Design and Management Program
January 2001

Certified by____
Dr. Eric Rebentisch
Research Associate, Center for Technology, Policy, and Industrial Development
Thesis Supervisor

Accepted by____
LFM/SDM Co-Director: Dr. Paul A. Lagace
Professor of Aeronautics & Astronautics and Engineering Systems

Accepted by____
LFM/SDM Co-Director: Dr. Stephen C. Graves
Abraham Siegel Professor of Management

A Partitioning Methodology for Helicopter Avionics System with a focus on Life Cycle Cost

by

Leon M. Silva

Submitted to the System Design and Management Program
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Engineering and Management

Abstract

Traditional system engineering methods rely on decomposition for establishing system partitions. As one of the key responsibilities of a system architect, the decisions made with respect to how an avionics system is partitioned play a significant role in the system's Life Cycle Cost (LCC). Despite this, most of the decomposition methods available focus on managing the complexity of the system with respect to the architect's ability to understand the system. In other words, the method is designed to keep the complexity of the system below the architect's inherent threshold limit of understanding. Given the impact that the partitioning process has on the system's LCC, it seems appropriate to derive a methodology that takes into consideration not only complexity but also other critical factors.

A framework was developed in order to quantify and compare the reliability and maintainability of five different helicopter avionics system architectures. The information was extracted from actual data collected during the maintenance of the five helicopter systems used by the US Government over a period of two years. Based on these five case studies, a set of partitioning criteria was developed that can be used in future programs in order to improve the LCC of the system.

The methodology provides a process whereby the analysis of maintenance data can assist the system architect in making better architecting decisions. The process identified some compelling issues with respect to data available in this area. There are significant limitations in the current maintenance infrastructure that need to be resolved to a greater extent before this method can be more generally applied. The process also identified some additional factors that have an impact on the LCC. The most compelling was that of legacy subsystems that offer significant problems with respect to their partitioning due to the fact that they are more entrenched in the current avionics market.

Thesis Supervisor: Dr. Eric Rebentisch
Title: Research Associate, Center for Technology, Policy, and Industrial Development

Table of Contents

List of Figures

Table of Contents

Acknowledgements

First I would like to thank my thesis advisor, Dr. Eric Rebentisch, for his support and patience. His level of attention and availability made this thesis a comprehensive learning experience. I would also like to thank all of the people back at Sikorsky who provided advice and support, especially Dr. Sandra Szabo, who encouraged me to initially enroll in the program and without whose support the experience might not have come to fruition. I'd like to thank my friends and colleagues who provided insight that contributed to this work. Special thanks to the architects that participated in the study who provided vital insight into the way that things are done and more importantly to how things should be done. Thanks to Phil Love who was the vital link to the maintenance data used for the study.

I'd like to also thank the many dedicated maintenance individuals from the US Navy who graciously answered questions with unending enthusiasm. Their daily challenges are certainly not fully appreciated by most. Thanks to the individuals from the US Coast Guard, Rockwell Collins, and Bendix that also provided their useful perspective.

Finally, I am most indebted to my wonderful wife Kelly for her love and endless support during this academic adventure. Without her support and patience, I could not have completed this work.

Acronyms

CDR	Critical Design Review
COTS	Commercial Off the Shelf
DP	Design Parameters
GPS	Global Positioning System
FR	Functional Requirement
HF	High Frequency
IPT	Integrated Product Team
KISS	Keep It Simple Stupid
LCC	Life Cycle Cost
NPV	Net Present Value
MA	Maintenance Action
MFD	Multi-Function Display
MH	Maintenance Hour(s)
MOTS	Military Off the Shelf
MTBCR	Mean Time Between Cannibalized Removals
MTBR	Mean Time Between Removals
MTBUR	Mean Time Between Unscheduled Removals
PC	Personal Computer
PDR	Preliminary Design Review
PV	Present Value
PVs	Process Variables
RNS	Radar Navigation System
SA	Sikorsky Aircraft
TRR	Test Readiness Review
UHF	Ultra-High Frequency
VHF	Very High Frequency

1 Introduction

1.1 Problem Statement

Historically, Sikorsky Aircraft has developed a majority of its airframes in direct response to a United States government military need. The structure of this development often led to programs with high technological and performance goals with less consideration for cost. In the recent decades, this has changed. Customers are now much more diverse and expect high performance at a lower cost. As a result, the number of new airframe developments has been significantly reduced with the industry converging on the concept of platforms. Thankfully, because of the performance-focused approaches of the past, Sikorsky Aircraft (SA) has benefited from a set of high quality and extremely robust designs that have made excellent platforms. Since few new airframes are being developed, even the modest growth in the helicopter market has resulted in the development of many new derivatives from the existing platforms.

A helicopter system is made up of many subsystems all of which play an important role in the ultimate performance of the vehicle. These include the airframe/structure, propulsion, avionics, flight controls, fuel, weapons, hydraulics, and hoisting amongst others. The technology changes in these subsystems occur at different rates. The subsystems that have a higher rate of change usually play a bigger role when it comes to new projects since the customers normally strive to capture the best performance available for the cost.

The relatively high rate of improvements and innovation in Avionics combined with the large role that avionics plays in mission performance makes it the biggest area of consideration and design when it comes to developing new derivative helicopters. There are typically two types of projects that can result from a derivative product. The first is a derivative that uses an existing platform but defines a unique set of avionics intended to support a particular mission. The second is an upgrade project that uses an existing aircraft with an existing configuration and incorporates changes to the avionics suite in order to achieve better mission performance. Airframes have very long operational lives usually exceeding 20 years and commonly reaching 30 years. It is very common given the high clock speed of avionics that several upgrades are required during that long life cycle. The life cycle costs include the total cost of developing and owning a system over its entire life. These costs include acquisition, utilization, and disposal.

The requirements for each of these systems are generally provided as some combination of functional level and commercial off the shelf (COTS). Deciding how and where to allocate the functionality is one of the bigger challenges of a system architect. Making these partitioning decisions to the satisfaction of the large group of stakeholders is complicated by many factors. The customer sometimes dictates an existing avionics subsystem that might be currently in wide use. These legacy subsystems are sometimes part of the existing logistics support system that will be used to support the new product. It may be because the customer has some previous experience with the system and is familiar and confident with its use. Customers very often have pre-conceived ideas of how the system will be configured based on their past experiences. In some cases, subsystem suppliers have already influenced the customer's position previous to the involvement of the avionics system architect. The unfortunate fact is that the architect rarely gets the opportunity to start with a clean sheet.

Traditional system engineering methods rely on decomposition for establishing system partitions. As one of the key responsibilities of a system architect, the decisions made with respect to how an avionics system is partitioned play a significant role in the system's Life Cycle Cost (LCC). Despite this, most of the decomposition methods available focus on managing the complexity of the system with respect to the architect's ability to understand the system. In other words, the method is designed to keep the complexity of the system below the architect's inherent threshold limit of understanding without particular consideration to LCC. Given the impact that the partitioning decisions have on the system's LCC, it seemed appropriate to derive a methodology that takes into consideration the maintenance costs attributed to the avionics subsystem. Therefore, the question is –

- Can a good understanding of maintenance costs drive different partitioning decisions?

1.2 Thesis Structure

This thesis proposed to answer the question expressed in the problem statement. The structure of the thesis is described below:

- Section 2 provides a review of current methods and techniques found in subject literature.
- Section 3 describes the methods developed in this work.
- Section 4 describes the interview of the avionics architects and the specific application of the method to maintenance data acquired for the five helicopter models studied.
- Section 5 describes the results and observations as it applies to the analyzed data.
- Section 6 provides conclusions made as a result of the development of this work and information on further study.

2 Literature Review

Architecture is defined as the science, art, or profession of designing and constructing the “structure of things” and architecting is defined as “the planning and building of structure”.[1] If we say that a system is a structure, system architecting is then the process of the planning and building of systems. Engineering is defined as the science concerned with putting scientific knowledge to practical uses.[1] At face value, there does not seem to be a big distinction between a systems architect and a systems engineer. This difference between engineers and architects has long been debated. Civil engineering offers some help in trying to make this distinction. Although civil engineers have not answered this question in the abstract, they have in practice. Engineering is a deductive process. In general, it deals almost completely with the measurable using analytic tools derived from mathematics and the hard sciences. Architecture deals largely with immeasurable quantities using non-quantifiable tools and guidelines based on practical experiences. That is to say that architecting is an inductive process. From a cost point of view, engineering is concerned with quantifying cost while architecture is concerned with quantifying value. Engineering targets technical optimization while architecture strives for customer satisfaction. Engineering is more of a science while architecture is more of an art.

In considering the differences between architecture and engineering, there is another important difference between the concerns of each. This is form versus function. The architect performs the most abstract, high-level function in product development. Function relates to the conduct of a system and to the operations and transformations that contribute to the performance of the system. Form relates to the structure, layout or arrangement of the physical or logical configuration of the system. Form also relates more closely to the aggregate sum of the parts of the system. Function can therefore be defined as the action for which a system exists while form more closely defines the particular ostentation or implementation of that function. In this context, the engineer tends to have more responsibility with respect to form while the

---

[1] Neufeldt, Victoria E., Editor in Chief, *Webster’s New World Dictionary – Third College Edition.* New York, NY: Webster’s New world Dictionaries, 1988.

architect's focus is on function. It is common and helpful to use the phrase "form follows function" in order to better relate the sequence.

2.1 System complexity

System complexity relates to having many interrelated, interconnected or interwoven elements and interfaces. Increasing complexity is at the heart of the most difficult problems facing today's system architects.[2] Therefore, complexity also drives the methods that are used by the architects. Although the number of subsystems can contribute to the complexity of the system, generally, the majority is contributed by the interactions between those subsystems. Because of this, a complex system requires a great deal of information to specify.

Complexity should be considered a system property that is absolute and quantifiable.[3] Possible measures of complexity can be the number of interactions organized by type. The sophistication of the interactions with respect to how much information is required to specify. The sensitivity or robustness of the interactions with respect to how much information is required to describe. The common premise relates complexity to the amount of information required for specifying or describing a given system in a comprehensive way.

Apparent complexity is the perception that something is complex.[3] It can be said that 'complicated' things have high apparent complexity. Complexity is also in the eye of the beholder. It is always with respect to some reference whether that be defined or interpreted based on the individual's experiences and point-of-view. Ideally, the complexity visible and managed by the architect should not be visible to the user of the system.

Functionality describes the grouping of utilities performed by a particular system. For example, the basic function of a watch is to indicate time of day. A more complex watch may also indicate the date, day of the week, have multiple alarms, and indicate time for more than one time zone. Time of day would be the functionality of the first

---

[2] Rechtin, Eberhardt. *The Art of Systems Architecting*. Boca Raton, FL: CRC Press, 1997.

[3] Crawley, E., *System Architecture Lecture Notes – Rev 2*, MIT, November 12, 1999

watch while time of day, date, the day of the week, multiple alarms, and time for more than one time zone would be the functionality of the second watch. Essential complexity is defined as the minimal level of complexity required in order to perform the fundamental utility desired from the system. It is that which is essential to deliver functionality before gratuitous complexity slips in.[4] Gratuitous complexity refers to the complexity associated with any part of the system that is not part of providing the essential functionality. Considering our more complex watch example, if there is a button press required in order to display some subset of the watches functions, then the button can be considered to contribute to the gratuitous complexity since it increases the total complexity of the watch without increasing its functionality.

What drives complexity? Mostly, it is driven by functionality. Functionality can also include a broader set of functions that the product will at some point be used for. Because of this, it is the basic functionality that drives essential complexity.[4] Complexity is not necessary good or bad; it should be considered a total cost that increases as you purchase more functionality, efficiency, flexibility, etc.

One of the main challenges of an architect is dealing with complexity. The human being's ability to simultaneously consider issues hovers around 7 ± 2 items.[5] Systems whose essential complexity exceeds the limit of humans to understand are perceived to be complicated. Systems below the limit of human understanding are considered simple. Integrated avionics systems are well beyond this threshold and tend to be considered complex. A couple of guidelines with respect to complexity have been used in order to make partitioning decisions. One is that the actual complexity should be kept close to essential complexity. This means that superfluous complexity should be kept to a minimum. This keeps total complexity of a system down. The other is that the perceived complexity should be kept below the limit of understanding from any single person's point of view. Keeping actual complexity levels near essential complexity facilitates the partitioning process because it reduces the number of functions that require a partitioning decision. A concept that results in low essential complexity is

---

[4] Crawley, E., *System Architecture Lecture Notes – Rev 2*, MIT, November 12, 1999

[5] Miller, G.A., *The Magic Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information*, An essay in The Psychology of Communication, Basic Books, 1967,

sometimes considered ‘elegant’. In order to manage the perceived complexity of these systems and maintain them within the range of human understanding, various methods of partitioning have been used in the past. These include decomposition, abstraction, and hierarchy techniques.

Decomposition refers to the separation of a system into smaller, more basic elements. This decomposition can be done with respect to function or form. Decomposition of function simply means taking a top-level function and breaking it up into lower level, more basic functions. Decomposition of form results in sub forms often called objects, modules, elements, or chunks. This technique for managing complexity takes on the concept of ‘divide and conquer’ reasoning that by dividing a big system into smaller systems results in the smaller systems being more manageable[6]. The big advantage of decomposition is that it reduces perceived complexity and that is what actually what makes the sub components of the system more manageable.

Abstraction is the process of separating the inherent qualities of something from the actual object or concept to which they belong. This allows functionality to have an intrinsic nature independent of form. This approach can be used to depict and mask more detailed behavior within subsystems permitting a simpler representation at the top level and can be used in both function and form.

Hierarchy is a technique in which groupings are ranked one above the other and is usually a byproduct of decomposition. This approach tends to conceal information that is more than one layer away from the reference point.

Using decomposition and creating abstractions and hierarchies can be used to maintain the actual complexity close to the essential complexity and keep perceived complexity below the understanding threshold of the individual. Some of the architecture in complex systems is not evident in the final product. This characteristic contributes to a system being user friendly.

Some indicators of good and bad architecture relate directly to a system’s standing with respect to complexity. See Table 2-1.

---

[6] Decomposition can actually increase the total complexity of the system above the essential complexity since it can increase the number of interactions within the sub components of the system. Increasing the number of interfaces inherently increases the complexity.

Characteristic	*'Good' Architecture*	*'Bad' Architecture*	*Avionics Systems*
Essential complexity	**Low**	**High**	**High**
Actual Complexity	**Slightly higher than essential**	**Significantly Higher than essential**	**High**
Perceived Complexity	**Within Limit**	**Above Limit**	**Above Limit**

Table 2-1 Complexity comparison; Good, Bad and Avionics Systems

2.2 Architecting approaches

There are four processes of Architecting described by Rechtin[7].

2.2.1 Normative method

The first is the *normative methodology*. This Architecting method is based on the predetermined definition of success or "what should be". This approach depends on the value judgments of experienced and successful architects. The guidelines tend to be prescriptive. Do this. Don't do that. This approach is based on the understanding of good, bad, right and wrong[8]. The rules are laden with the values of the creator[9]. These rules are obviously a function of the architect that propagates them. The better the architect, the better the results. There are some significant disadvantages to this method. Those that might follow this process may be deprived of developing innovative and more creative approaches because they are focused on the method based on someone else's personal experiences. This type of process can also be very difficult to

---

[7] Rechtin, Eberhardt. *Systems Architecting, Creating and Building Complex Systems*. Englewood Cliffs, NJ: Prentice Hall, 1991.

[8] Nadler, Gerald, System Methodology and Design, IEEE Transactions on Systems, Man and Cybernetics SMC-15, 6, 685-97. Dec 1985.

[9] Lang, Jon., *Creating Architectural Theory, The Role of the Behavioral Sciences in Environmental Design*. New York, NY, 1987.

follow. As the guidelines get more detailed, their application is less and less general and in many cases loses their use because they are too dependent on the context in which they were originally developed.

2.2.2 Rational method

The second is the *rational method*. The problems with the normative method resulted in the development of this method. This method depends on more structured approaches. Procedures play a bigger role in this approach. The tools used are more rigorous, methodical and mathematical. They can include probability theory, vector algebra, algorithms and problem solving methodologies.

The basic premise of the rational theory is that problem solving is inherently procedural. Following a relatively simple set of steps can lead to a solution.

- This procedural premise is based, to some extent, on drawing analogies between computers and human minds. Indeed, the rational theory was generated in large part by cognitive (knowledge), information, and behavioral scientists applying the scientific method to the study of how humans solve problems.[2]

Like computerized databases, human memory is thought to be organized in “chunks”. These smaller units are used to facilitate the search for a solution when the possible number of results is very large. This may be linked to the concept of partitioning that will be discussed later. Similar to a computer, the brain has both long and short-term memory and can carry out steps in sequential, logical fashion. Using these two facts, Simon proposed that the human brain’s functional complexity is not a function of its architectural complexity but that of the large amount of information related to experiences contained in the brain[10]. If this is true, then the process of architecting can be defined as a series of rational steps operating on an established database. The results may not be optimal but a useful conclusion should be achievable making the process of architecting mechanistic and not intuitive. These conclusions tend to be controversial to say the least. It appears that the brain actually has several different

---

[10] Rechtin, Eberhardt. *Systems Architecting, Creating and Building Complex Systems*. Englewood Cliffs, NJ: Prentice Hall, 1991.

ways of thinking, only one of them being classified as logical. It seems likely that the process of architecting uses them all.

Despite of its shortcomings, the rational method is an improvement over the normative method in that it has less dependence on specific personal experiences and the learning that comes from those.

2.2.2.1 General systems theory

One version of the rational method is the general systems theory (GST). When it was first developed in the late 1940's proponents such as the mathematician Norbert Weiner hoped that the "system approach" would solve many of the world's toughest systems problems. The premise is that systems can be studied in the abstract and in the absence of context. This is clearly a pure approach to systems engineering that is similar to pure mathematics where the processing of the numbers is done independent of their meaning. This would then facilitate the handling of complex systems. One would start by taking a particular system and mapping it into a general system. The general system could then be solved and then be remapped into a specific system. Compared to the analytical procedure of classical science with resolution into component elements and one-way or linear causality as basic category, the investigation of organized wholes of many variables requires new categories of interaction, transaction, organization, teleology, etc. These considerations lead to the postulate of a new scientific discipline that we call general systems theory. Its subject matter is the formulation of principles that are valid for "systems" in general, whatever the nature of the component elements and the relations or "forces" between them. General systems theory, therefore, is a general "science of wholeness"[11]. The meaning of the somewhat numinous expression, "The whole is more than the sum of its parts" is simply that essential characteristics are not explainable from the characteristics of the isolated parts.

There are many issues with general system problem solvers (GSPS). In order to make progress with these methods, generally some assumptions and constraints have

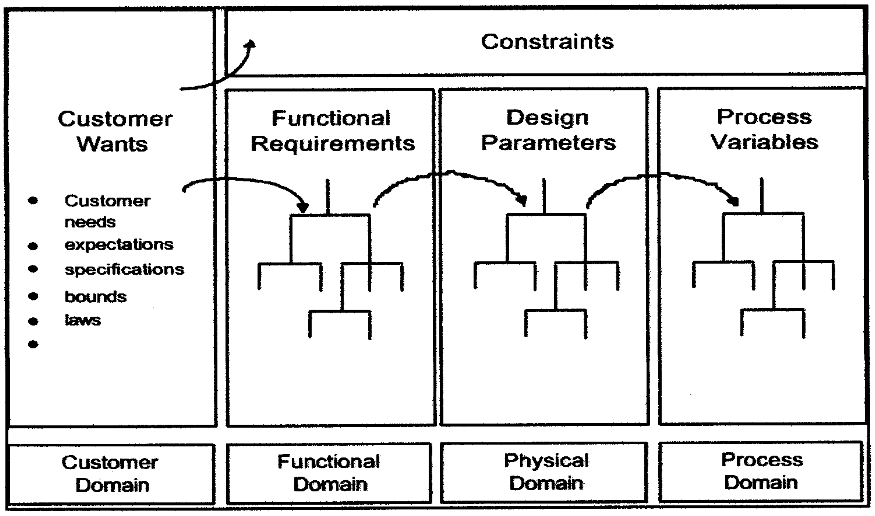
---

[11] International Society for the System Sciences (ISSS) web site, www.isss.org.

to be defined. These tend to be very context dependent and turn out to defeat the original objective of generality. Some structure is required in order to be able to operate within these GSPS methods and unfortunately it appears that the details of the structure are dependent on the context. Another obstacle is the difficulty in quantifying relationships within the particular system. It turns out that sometimes these are not quantifiable and commonly do not have the same "value" when compared to each other. That is to say that two relationships may not have the same impact on the overall system and therefore do not have the same importance.

2.2.2.2 Axiomatic design

Axiomatic Design defines design as the creation of synthesized solutions in the form of products, processes, or systems that satisfy perceived needs through mapping between functional requirements (FRs) and design parameters (DPs).[12] The goals of the design are represented as functional requirements (FRs). In order to satisfy the needs, which are defined in the customer domain, the FRs are defined in the functional domain. The Design Parameters (DPs) relate to how the functional requirements are satisfied. DPs are created in the physical domain to satisfy the FRs. See Figure 2-1.



---

[12] Suh, Nam P., The Principles of Design, Oxford Press, New York, 1990.

Figure 2-1 Four Domains of Axiomatic Design[12]

The customer's needs must be mapped to the functional domain where they are translated into a set of FRs. As a result of translating customer wants to FRs, constraints will start to appear. Constraints must be followed through the complete design process and can apply to FRs, DPs, or Process Variables (PVs). The FRs are then mapped to the physical domain and the DPs are mapped to the process domain in terms of PVs.

In most cases, decomposition of the problem is required. Figure 2-2 shows hierarchies in the functional, physical, and process domain. Going back and forth between two domains as required in a sort of zigzag pattern develops the hierarchies. A design concept is generated after defining the FR of the top level. This mapping is demonstrated in Figure 2-2.

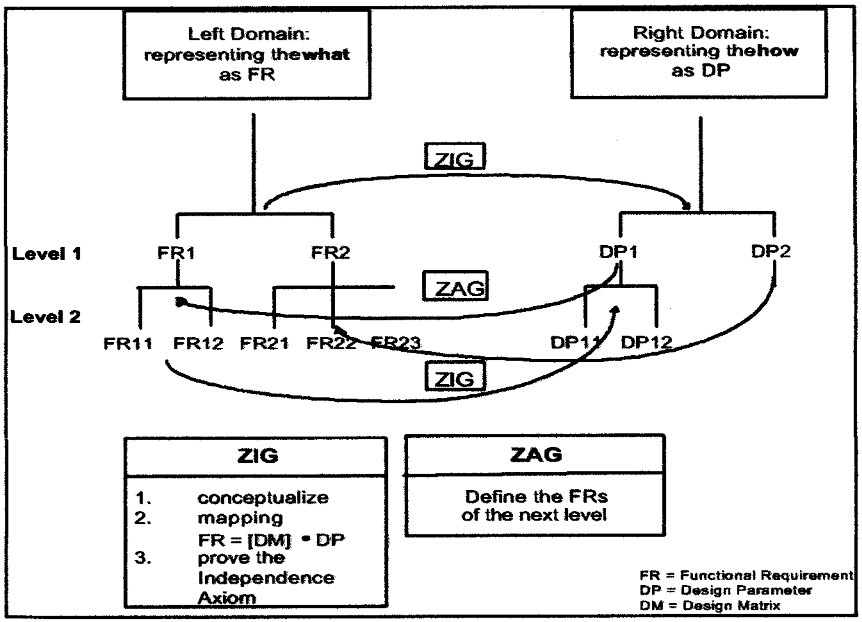


Figure 2-2 Developing an hierarchy by zigzagging between two domains[13]

The mapping requires that two axioms be followed[12]:

Axiom 1: The independence Axiom
Maintain the Independence of the FRs
Axiom 2: The Information Axiom

Minimize the Information Content of the Design

The FRs and DPs are described mathematically as a vector. The Design Matrix (DM) describes the relationship between FRs and DPs.

$$\{FRs\} = [DM]\{DPs\}$$

An element in the design matrix Dm_{ij} is defined by:

$$Dmij = \delta FR_i / \delta DP_j$$

This is a constant in linear design. DM must be a diagonal or triangular matrix in order to satisfy the independence axiom. Designs involving diagonal matrices are referred to as uncoupled designs. Designs involving triangular matrices are referred to as decoupled designs. Decoupled designs can satisfy the Independence Axiom if it DPs are implemented in a specific order.

The Information Axiom, 2nd Axiom, is defined in terms of probability of successfully achieving FRs and DPs. The information is defined as:

$$I = \sum_{i = 1, n} [\log_2 (1/p_i)]$$

Where p_i is the probability of DP_j satisfying FR_j. I is in units of bits. The total information content is the sum of the probabilities since there are n FRs. The information Axiom states that the design with the smallest I is the best design, since it requires the least amount of information to achieve the functional requirements of the design.[13]

This approach, although theoretically interesting, offers one significant disadvantage, the dependence on the 1st Axiom. The Independence Axiom is for the most part not particularly practical since the interdependence between the functions of the avionics systems being considered is very high.

---

[13] Reynal, Vincent A. and Cochran David S., *Understanding Lean Manufacturing According to Axiomatic Design Principles*. The Lean Aircraft Initiative Report Series, #RP96-07-28, November 1996.

2.2.3 Argumentative method

The *argumentative approach* refers to the scenario where an Architect is arguing with and learning from others involved in the process. These participants make contributions of new problems, possible new solutions, and different points of view while the architect weights these inputs at a higher level and tries to reach a satisfactory architectural result. It can be likened to brainstorming with a single decision-maker within the brainstorming session. This process is difficult to implement and control. It often leads to dispersion of the original goals and to the implementation of particularly dramatic ideas without enough consideration for the well thought out ones. Establishing a structured and repeatable argumentative method is nearly impossible.

2.2.4 Heuristic method

The *heuristic approach* has some of its origins in the pronouncement and rational approach. The central idea came from asking architects what they do when confronting a complex problem. The skilled architect likely would answer, "Just use common sense"[14]. A better way to describe "common sense" might be to call it contextual sense. This refers to knowledge of what is reasonable given the particular context being considered. Through their education and experiences over the years, practical architects build up a body of contextual sense. This process seems to take about 10 years[15]. Only then are they generally entrusted with complex, system level problems. Just as important as accumulating these experiences, architects learn how to use their body of common sense in order to effectively, although not always efficiently, solve complex problems.

A heuristic can be thought of as a statement of common, or contextual, sense that aids in concept development, problem solving, decision-making, or judgment. The difference between a pronouncement made by an individual and a heuristic is that the heuristic tends to apply to subjects with some general consensus. Heuristics are not

---

[14] Rechtin, Eberhardt. *Systems Architecting, Creating and Building Complex Systems*. Englewood Cliffs, NJ: Prentice Hall, 1991.

[15] Rechtin, Eberhardt. *Systems Architecting, Creating and Building Complex Systems*. Englewood Cliffs, NJ: Prentice Hall, 1991.

quite scientific laws because they generally are not qualitative. They are different from mathematical statements because they tend to be derived as generalizations from specific examples, not conclusions derived from general principles. An example of a heuristic night be:

> ***Success is defined by the beholder, not by the architect.***[15]

Heuristics can be derived from almost any source. Although they can be helpful under certain conditions, they have the disadvantage of not being quantifiable. Relating the information to others or substantiating a particular decision becomes more difficult in the absence of a quantifiable method. Heuristics should be thought of as guidelines and they are not likely to be successful without some additional supporting tools.

2.3 System partitioning

The driver for decomposition and partitioning is complexity. Decomposition can be performed on a function, concept, or form and can be done at any level. Decomposition can operate on goals, function, needs, and processes resulting in abstract objects. Decomposed things are never recomposed to any literal sense of the word.[16] Partitioning is defined as the decomposition of a set into a family of mutually exclusive sets but is done in either a physical or logical domain (form) and is concept specific. One distinction between partitioning and decomposition is that the results of partitioning are generally a function of the thing being partitioned. For example, if you partition a form, you get a product or a system. Partition timing and you get a timeline. Partition operator actions you get operations. Partition a process schedule you get a schedule. Partition a team you get an organization. The big difference with partitioning is that it results in actual physical or logical things and that partitioned things do integrate to form a whole. Neither decomposition nor partitioning is necessarily unique. Two different architects will generally develop different decompositions and partitioning.

---

[16] Crawley, E., *System Architecture Lecture Notes – Rev 2*, MIT, November 12, 1999.

Decomposition tends to be more generic while partitioning tends to be more specific to the form.

A useful guideline might be to partition for easy integration and for flexibility in the evolution of the product. These are key but their implementation can be very difficult. Decomposition and partitioning are key to managing complexity from the architects perspective. Although the challenge in a complex project can appear to be the integration of the subsystems, the source is usually related to bad partitioning decisions at the start of the project. "…60% of all failures are caused by connectors or wiring… partitioning should minimize the interaction across boundaries. For example, an excessive number of wires between boxes suggests poor minimization of coupling and poor adhesion within the boxes."[17]

Given the complexity and typical boundless characteristic of systems, the architect has a significant challenge in deciding how to partition the functionality contained in a complex system. Here are some reasons why the partitioning problem is difficult. Architects mostly evolve from engineering backgrounds. The traditional training for an engineer focuses on optimizing designs. This is in some sense a contrast to the architect's goals, which might be described as 'optimize' nothing but the complete system in the largest context that you can imagine. If a system has been optimized, it is usually with respect to some single or at most, a few characteristics. These 'optimized' designs are inherently difficult to change. Because products tend to evolve from their original defined functions, large complex systems often experience many specification changes through the development process. It is important for the system to have some flexibility not only at the point of the original conception but also during the subsequent period of development and use. A fine balance must be achieved with respect to flexibility. Too much flexibility can compromise performance while too much optimization on performance can compromise flexibility.

So how are partitioning decisions made now? Arbitrary slicing through a tightly integrated system is clearly a mistake. Unfortunately, decisions on partitioning sometimes get made for less then stellar reasons. A common approach is to base

---

[17] Lacy, J.A., *Systems Engineering Management,* New York, NY, McGraw-Hill, 1992.

partitioning on the enterprise's organizational structure. The system is partitioned following the division of the organization currently in place. The traditions and decisions of the past then influence the way the functions are distributed and how partitioning decisions are made in the future. This influence can be thought of as the Legacy Effect. In many cases, the enterprise and its groups have some legitimate validation for how they are focusing on the job. The way it was done in the past may not be the best way to do it in the present given the current set of goals. The subsystems that play a role in avionics systems and their integration also suffer from the legacy effect. Many of the functions are predetermined either by the customer or by the lack of options available.

2.3.1 Partitioning using heuristics

The industry still depends on a small set of experienced individuals who use mostly the normative or heuristic methods in order to fulfill their architecting responsibilities. Some interesting heuristics in the area of partitioning are listed below:

In partitioning, choose the elements so that they are as independent as possible, that is, elements with low external complexity and high internal complexity. (Alexander, 1964)[18]

In partitioning a system, choose a configuration in which local activity is high speed and global activity is slow change. (Courtois, 1985)[19]

In partitioning a system, choose a configuration with minimal communication between the systems. (For example, aerospace, communication network, and software systems.)[20]

---

[18] Rechtin, Eberhardt. *Systems Architecting, Creating and Building Complex Systems*. Englewood Cliffs, NJ: Prentice Hall, 1991.pp 41.

[19] Ibid, pp 41.

[20] Ibid, pp 41.

Don't partition by slicing through regions where high rates of information exchange are required. (For example, computers)[21]

The heuristics noted don't seem to be so much concerned with the performance of the system from a life cycle cost point of view but more from the point of making it more manageable for the architect.

An important point refers to keeping functionality isolated within an element such that only a few critical external events can disturb the inner workings. This isolation guideline can contribute to maintaining stability within a particular subsystem. A related heuristic is:

Design the elements to make their performance as insensitive to unknown or uncontrollable external influences as practical.[22]

This helps in the development process since it means that those subsystems can be developed separately. This approach is designed to achieve good system performance from a holistic point of view but will not result in optimal performance within a given parameter. It does not necessarily result in a system with minimum weight, minimum power consumption, or any optimization for any other characteristic.

2.3.2 Partitioning trees

There are some more formal methods supporting partitioning decisions. The most common form is using decomposition, sometimes in the form of partitioning trees. This process involves dividing the problem into smaller parts at each successive level of form or logic. This approach can be thought of as having vertical cuts through the solution space. This results in a multilevel tree-structure hierarchy. The process continues until the parts are small enough to perform detailed design. The last step is to integrate the results. Figure 2-3 depicts the tree approach to partitioning.

---

[21] Ibid, pp 41.
[22] Ibid, pp 42.

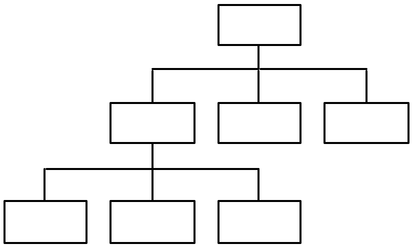


Figure 2-3 Structure of partitioning tree

This approach has some advantages. First, it is very general since you can break up any large problem into smaller parts. It allows the system to be made up of a subsystem design, developed and implemented by others. It leads to a clear definition of interfaces and responsibility for each so it offers some advantages to the organizational structure.

The tree process of partitioning also has some disadvantages. The main disadvantage is the lack of connectivity between the branches. Horizontal movement is usually more difficult. You can connect the elements at the same level on your branch going in the vertical direction but cannot connect the other branches without going up and over. One important weakness relates to flexibility. When there are significant and repetitive changes to the specification, the tendency of making local changes in the implementation when using a tree approach compromises the result. It can lead to changes that change the approach to a non-tree problem with lots of complicated branches. As the tree complexity increases, as with any other system, making changes becomes increasingly difficult. This sometimes results in highly modified large system definitions becoming so cumbersome that the architect may once again start from scratch.

2.3.3 Partitioning layers

A less common method of partitioning is layers. The method is to break the problem up into groups or layers no characteristic of which has any explicit connection to those in the layer above and below it. This approach can be thought of as having horizontal cuts through the solution space. Some examples would be assembler and the Internet. Software assemblers are generally structured so that the functions operate at a given level without significant interaction between levels.

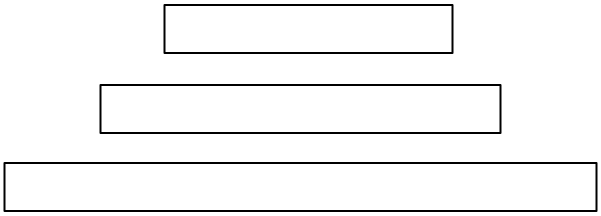


Figure 2-4 Structure of partitioning using layers

The advantage of this approach comes to light when making a specification change. If the system has been properly designed, changes can be made at the highest levels without impacting lower ones. This allows the architect to better focus on reducing the lifetime cost of the system without being limited by the changes of the day. Organizations that emphasize rapid creation of effective teams, or very long organizational life, tend to be layered.

The disadvantages with this method are related to the extra effort that is required in order to establish and maintain each layer. This has to be a carefully evaluated effect but it can often be worth it given the advantages of this approach. This approach is inherently more difficult. It requires more experience and foresight. Object oriented design is an example. If you consider a general concept where functionality is organized in layers that do not have intensive links, the aggregation of the objects results in a system that is easier to modify. The additional effort to initially set up the system that way is the trade-off.

2.3.4 Partitioning using tree and layer combinations

There have been many attempts at taking advantage of both the vertical implications of tree structures and of the horizontal implications of the layer method. Most of these are in the area of organization and they include integrated product teams (IPTs), process reengineering, flat organizations, matrix organizations, military organizations (rank and unit). These organizations tend to be structured more like matrices.

Rate of change is an important consideration for choosing which one to use. Low rates of change will generally make the tree-structured approach seem better. Very high rates of change will make the make the layer approach seem better. PC architecture is an example.[23] This can be also seen in organizations. Organizations that are involved in industries with large rate of change tend to establish small and therefore flexible organizations or combinations of small units.

Much of the abstract thinking about complexity, decomposition, and connectivity is common to both tree and layer methods. Team organization and Form connectivity are highly coupled and in many ways determine product success. A traditional view is that teams exist so consider fitting product to the model of the organization. Eppinger's fundamental point is to fit the team to the organization for best product results[24].

2.3.5 Partitioning using tree and layer combinations

Another method for organizing functions into groupings is referred to as the N x N Matrix.[25] This derivation from the popular Design Structure Matrix method can be used for establishing organizations, subsystems, components, processes, and so forth. For organizing functions with the goal of achieving good partitioning, one could start by establishing an N x N matrix filling the diagonals with N elementary functions. See **Figure 2-5** for a depiction of this approach.

The remaining elements are filled with the one way interfaces between the basic functions. The interfaces can be thought of as "flowing" clockwise with columns mapping to the input into the basic function contained in that column and the rows mapping to the outputs out of the basic function contained in that row.

---

[23] Baldwin, Carliss Y. & Clark, Kim B. *Design Rules, The Power of Modularity.* Cambridge, MA: The MIT Press, 2000.

[24] Eppinger, Steven D. and Ulrich, Karl T., *Product Design and Development*, New York, NY, McGraw-Hill, 1995.

[25] Adapted from Kockler, DSMC Systems Engineering Management Guide.

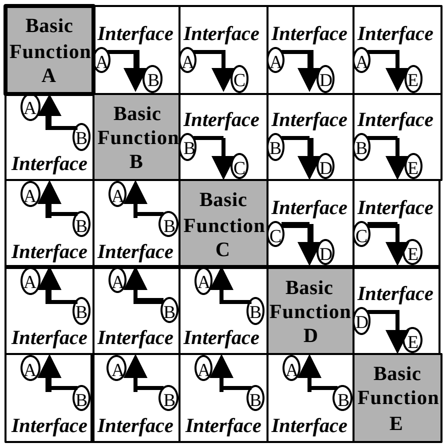


Figure 2-5 N x N Matrix Structure

"Squaring off" sections that have high interface density defines the functional groupings that are contained in one subsystem. An example of this is provided in Figure 2-6.

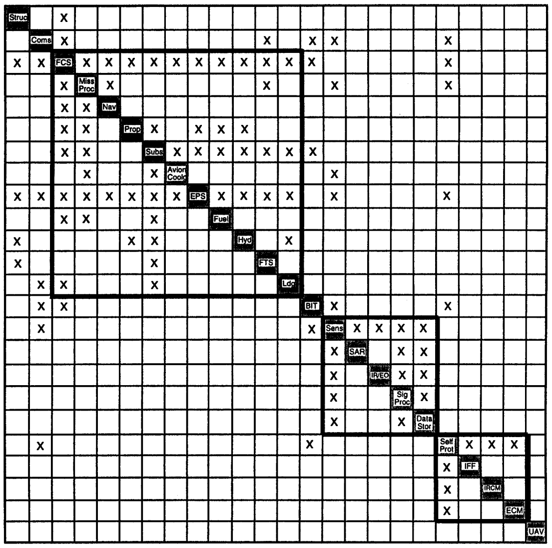


Figure 2-6 Partitioning defined by the groupings of related functions.

Three groupings are established given the matrix's functional interface density. Partitioning here refers to the assigning of the functions to subsystems. This method provides a good method for aggregating and grouping with respect to interdependencies of the functions in the system. The biggest limitation with this approach is that it only accounts for one parameter; functional interfaces.

2.4 Life Cycle Cost

Life cycle cost (LCC) is defined as the total cost of acquiring and owning a system over the entire life of the system including acquisition, operation, and disposal. Figure 2-7 Distribution of costs through life cycle shows a typical breakdown of the LCC indicating that the operations and support component is about 60% of the total costs.

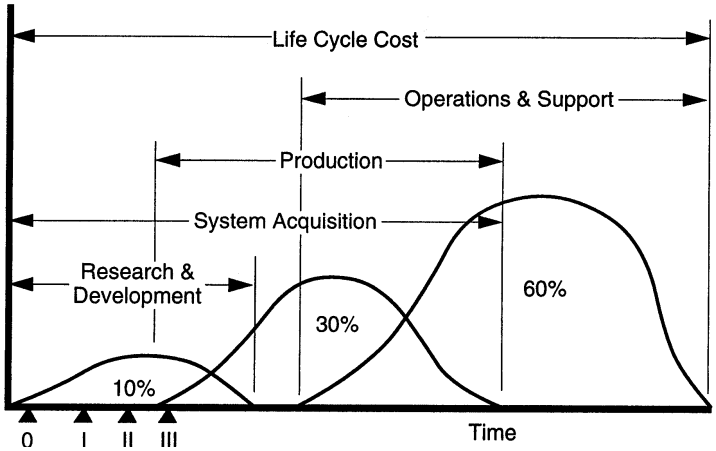


Figure 2-7 Distribution of costs through life cycle[26]

This large cost is generally made up of the following:

- Actual operations (flying hours, fuel, personnel, etc.)
- Labor costs associated with the maintenance of the aircraft including troubleshooting, removing and replacing of equipment
- Costs associated with the tracking of maintenance actions
- Purchase of spare components
- Maintenance of logistics system to manage spare components
- Training for maintenance personnel

Many of the current methods used for partitioning revolve around managing the system architect's view on complexity. This results in systems that could be developed and operated within the limitations of the existing context but seems to limit the possibilities of implementing a methodology that considers other factors. One area that

---

[26] Kochler, Defense Systems Management College, Systems Engineering Management Guide, 1989.

contributes significantly to the final life cycle cost is the maintenance of an aircraft once it is situated in an operational environment.

2.5 Objectives of a System Architect

The systems architect is concerned with the overall architecture, not just the system architecture. Although this work focuses on the aspect of partitioning, it is recognized that partitioning is only one of several important parts of effective system architecture. In an effort to be most concise, four basic points have summarized the responsibilities of the system architect[27].

The first is "Fit form to function". This implies the full understanding of "function" before a form can be determined. Functioning in an abstract world for some period of time before deciding on the form is critical.

The architect should aggregate function into a few groups. This speaks to the idea of forming functional modules in order to reduce complexity but also takes advantage of aspects that might be common between more than one function. Keeping in mind not to partition into too few groups or else compromise the flexibility of the system.

The architect should partition into autonomous elements. These elements need to contain the required functionality but not be too isolated from the functionality of the system. Eliminating too many of the interfaces will reduce the benefit of integration.

The challenge is to "fit", "balance", and "compromise". Saying that everything is important may sound ridiculous but to some extent, it is true. Relating and understanding the context enough to make good decisions with respect to partitioning will consider the reasonable compromises and achieve the system balance required for a successful system.

---

[27] Crawley, E., *System Architecture Lecture Notes – Rev 2*, MIT, November 12, 1999.

3 Methods

3.1 Interviews with Avionics System Architects

In order to identify the methods that are currently being used, a group of experienced avionics system architects was interviewed. The group of architects was made up of twelve individuals with an average experience in avionics system architecting of 14.5 years. Of the twelve, nine are employed at Sikorsky Aircraft and have all been involved to some extent with the architecture of the five aircraft types studied. The three remaining architects were from outside the organizations. One was from a large organization significantly larger than SA that is involved in the design and integration of advanced avionics systems. The other two are from smaller companies that have done subcontract work for SA in the past.

With an eye on identifying the actual processes practiced, the criteria for interview candidates was:

- Extensive experience in the specific area for this study, Avionics System Architecting
- Minimum Experience of 10 years in the field
- Experience of at least three projects involving significant architecting content
- Willingness to participate in the interviews
- Demonstration of interest in the results of the study

Each individual interviewed had the title of Sr. Avionics System Engineer or higher.

The interview was designed to identify architecting decision processes with focus on the partitioning aspects. A copy of the structured interview questions is provided in Appendix A. The interviewing process began with a brief explanation of the area of study including a review of the research questions. Each interviewee was assured that specific details of their interview would be kept confidential to encourage openness. A specific time charge number allocation was made available and the interviews were scheduled around the interviewee's schedule in order to assure that the required amount of time was used without having to rush. The interview times ranged from 40 minutes to over two hours.

All but three of the interviews were done in person with the exceptions done via telephone due to distance constraints. Detailed notes were taken during each of the interviews and transcribed, usually the same evening, in order to capture the full essence of each of the interviewee's perspectives.

3.2 Maintenance data analysis methodology

The methodology developed here had as its primary objective the identification of partitioning rules that reduce life cycle costs of an aircraft with a particular focus on reducing the significant maintenance costs associated with helicopters. For this, maintenance and reliability data was gathered from five different helicopter products produced by Sikorsky and operated by the US Navy and Marine Corps. These consisted of 3 helicopter types in the medium-lift category (Seahawk variants) and 2 helicopter types in the heavy-lift category (H-53 variants). The data were acquired in raw format and processed as required in the development of the described methodology.

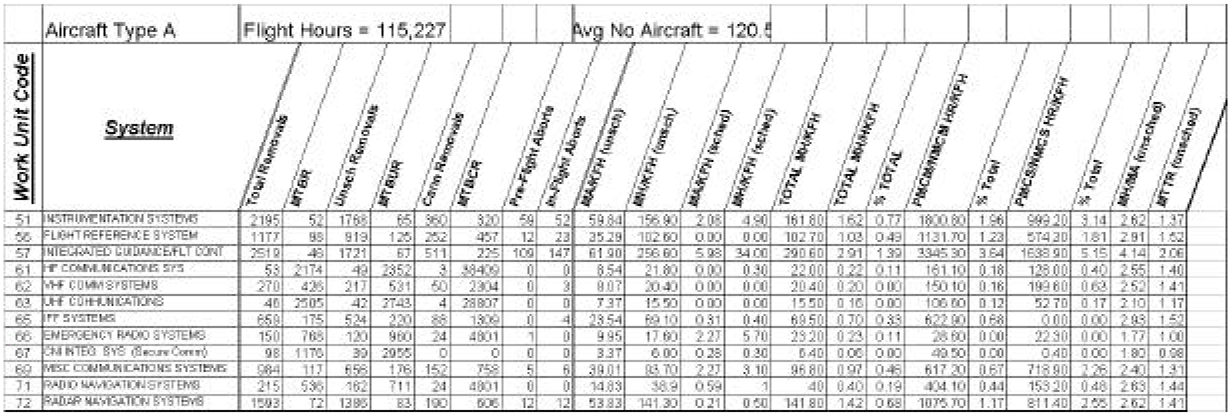
The hypothesis considered was that through the analysis of real data for well defined and understood avionics systems, a series of partitioning insights could be identified to be later applied in the design of new avionics systems. For this, the data gathered from the five aircraft were first compared in order to evaluate the feasibility of using this data set for a case study. The advantage of using five aircraft all operated by the two operators is that similar technique and an identical system was used for the collection of the maintenance and reliability data. The system used by all five aircraft utilized identical categories and nomenclature for the subsystems despite the fact that aircraft had different avionics architectures.

The full set of data tracked a total of 33 sub-system categories. Of these, twelve of the categories made up the avionics system and were studied in detail. The data covering the 12 systems were extracted from the full data set and scanned into a table in order to facilitate its analysis. There are several components of avionics system of the studied aircraft that fall into somewhat operationally sensitive areas. These include weapons controls and countermeasures. Although the existence of these types of devices are publishable, the maintenance and reliability data associated with these devices was considered to be sensitive from the authors perspective. For that reason, none of these subsystems or data associated with them were used in this work. These subsystems are naturally isolated from the rest of the system so excluding them from the study does not impact the other subsystem because they have no interfaces. The categories used in the study are listed in Table 3-1.

Work Unit Code	Sub-System
51	Instrumentation Systems
56	Flight Reference System
57	Integrated Guidance/Flight Controls
61	HF Communications System
62	VHF Communications System
63	UHF Communications System
65	IFF System
66	Emergency Radio System
67	CNI INTEG. System (Secure Comm)
69	MISC Communications System
71	Radio Navigation Systems
72	Radar Navigation Systems

Table 3-1 Avionics Subsystems Analyzed

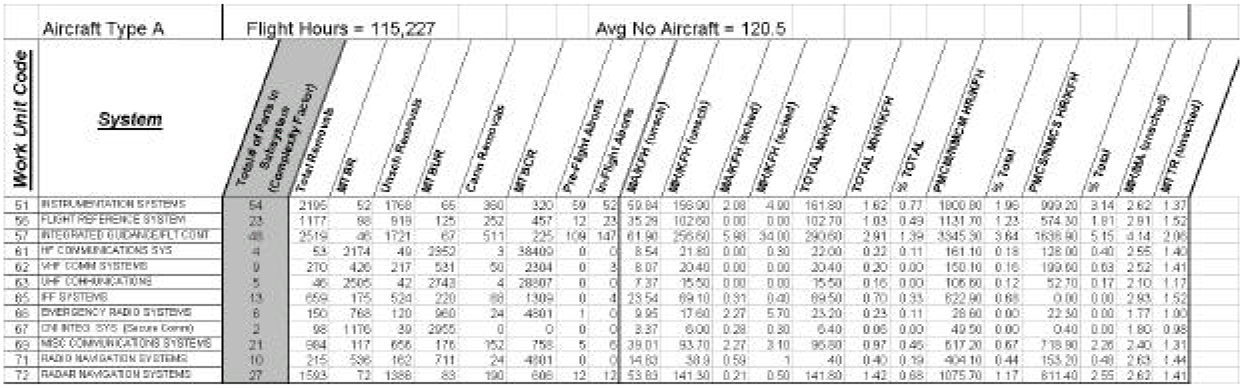
The Work Unit Code (WUC) is a unique numerical designator utilized by the maintainers to refer to each of the subsystems. As mentioned before, they are identical regardless of which of the five aircraft it references. The data were grouped by sub-system and consolidated into a summary sheet in order to facilitate the first round of review. A sample of this summary data sheet can be viewed in Table 3-2.



	Aircraft Type A	Flight Hours = 115,227							Avg No Aircraft = 120.5													
Work Unit Code	System	Total Removals	MTBR	Unsch Removals	MTBUR	Cann Removals	MTBCR	Pre-Flight Aborts	In-Flight Aborts	MA/KFH (unsch)	MMH/KFH (unsch)	MA/KFH (sched)	MMH/KFH (sched)	TOTAL MA/KFH	TOTAL MMH/KFH	% TOTAL	PMCM/NMCM HR/KFH	% Total	PMCS/NMCS HR/KFH	% Total	MMH/MA (unsched)	MTTR (unsched)
51	INSTRUMENTATION SYSTEMS	2195	52	1768	65	360	320	59	52	59.84	156.90	2.08	4.90	161.80	1.62	0.77	1800.80	1.96	999.20	3.14	2.62	1.37
56	FLIGHT REFERENCE SYSTEM	1177	98	919	125	252	457	12	23	35.29	102.60	0.00	0.00	102.70	1.03	0.49	1131.70	1.23	574.30	1.81	2.91	1.52
57	INTEGRATED GUIDANCE/FLT CONT	2519	46	1721	67	511	225	109	147	61.90	256.60	5.98	34.00	290.60	2.91	1.39	3345.30	3.64	1638.90	5.15	4.14	2.06
61	HF COMMUNICATIONS SYS	53	2174	49	2352	3	38409	0	0	8.54	21.80	0.00	0.30	22.00	0.22	0.11	161.10	0.18	128.00	0.40	2.55	1.40
62	VHF COMM SYSTEMS	270	426	217	531	60	2304	0	3	8.07	20.40	0.00	0.00	20.40	0.20	0.00	150.10	0.16	199.60	0.63	2.52	1.41
63	UHF COMMUNICATIONS	46	2505	42	2743	4	28807	0	0	7.37	15.50	0.00	0.00	15.50	0.16	0.00	106.60	0.12	52.70	0.17	2.10	1.17
65	IFF SYSTEMS	659	175	524	220	88	1309	0	4	23.54	69.10	0.31	0.40	69.50	0.70	0.33	622.90	0.68	0.00	0.00	2.93	1.52
66	EMERGENCY RADIO SYSTEMS	150	768	120	960	24	4801	1	0	9.95	17.60	2.27	5.70	23.20	0.23	0.11	28.60	0.00	22.30	0.00	1.77	1.00
67	CNI INTEG. SYS (Secure Comm)	98	1176	39	2955	0	0	0	0	3.37	6.00	0.28	0.30	6.40	0.06	0.00	49.50	0.00	0.40	0.00	1.80	0.98
69	MISC COMMUNICATIONS SYSTEMS	984	117	656	176	152	758	5	6	39.01	93.70	2.27	3.10	96.80	0.97	0.46	617.20	0.67	718.90	2.26	2.40	1.31
71	RADIO NAVIGATION SYSTEMS	215	536	162	711	24	4801	0	0	14.83	38.9	0.59	1	40	0.40	0.19	404.10	0.44	153.20	0.48	2.63	1.44
72	RADAR NAVIGATION SYSTEMS	1593	72	1386	83	190	606	12	12	53.83	141.30	0.21	0.50	141.80	1.42	0.68	1075.70	1.17	811.40	2.55	2.62	1.41

Table 3-2 Sample Avionics Subsystem Summarized Data for Aircraft type A

One thing that became clear from the initial review was that each subsystem had a varied level of complexity. This aspect had to be considered in order to make equitable comparisons between subsystems. A complexity factor for each subsystem was deemed required. Several methods were considered including making a nominal assignment to each subsystem given the author's first hand knowledge of each of the subsystems. Going back to the previous discussion, complexity is driven by interfaces. Complexity is also indicated by the tendency of certain components to fail periodically given that a majority of these failures are attributed to failures in physical interfaces either between subsystems or within subsystems between its modules. For each subsystem, the helicopter data tracks all components that make up the subsystem that have also failed at least once during the period covered. Using this total number of failed components captures both aspects of the component complexity, the inherent complexity contributed by the number of parts that have some inherent complexity and the fact that they have failed. This approach also eliminates the contributions made by parts that are involved in the subsystem but are particularly simple and don't fail. From the perspective of partitioning, the number of components in the subsystem provides an excellent indicator of complexity and was therefore used as the complexity factor. This factor was established for all of the avionics subsystems involved and added to the summarized table. The data for aircraft A is shown in Table 3-3.

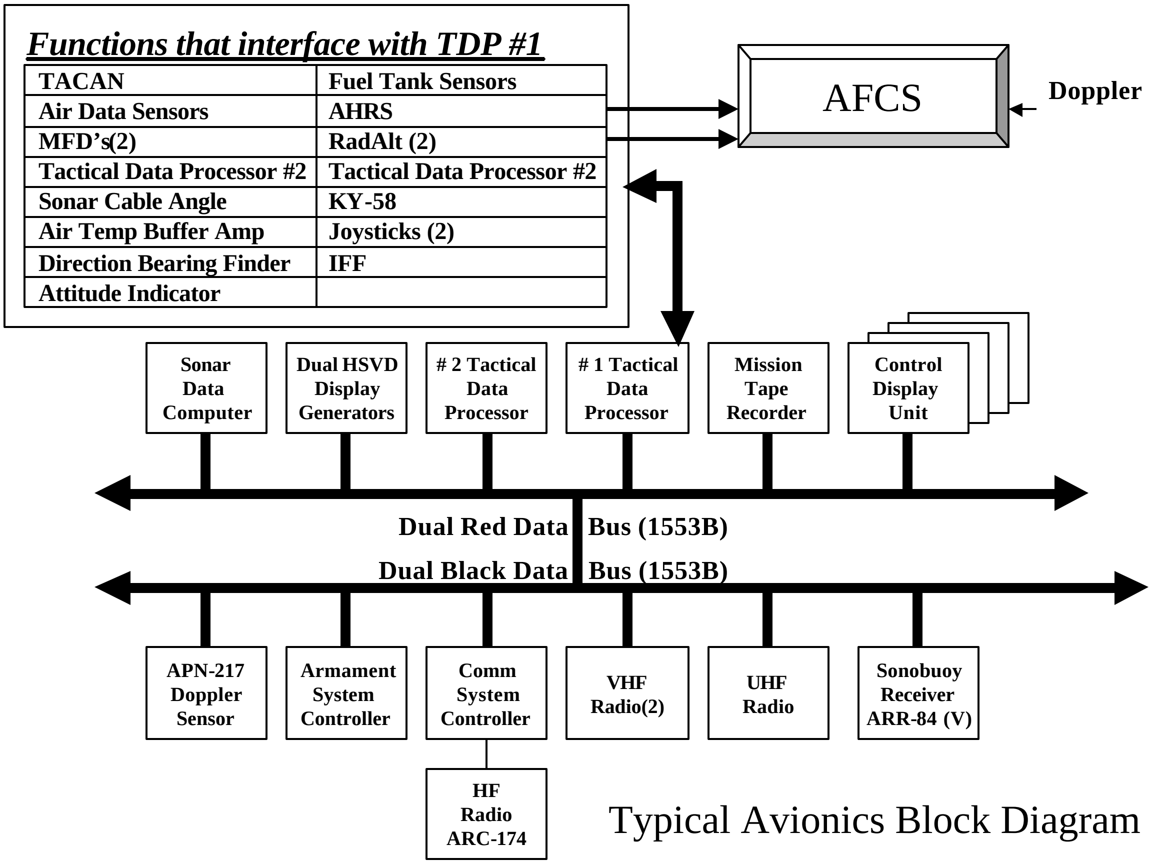


Aircraft Type A
Flight Hours = 115,227
Avg No Aircraft = 120.5

Work Unit Code	System	Totals of Parts in Subsystem (Complexity Factor)	Total Removals	MTBR	Unsch Removals	MTBUR	Cann Removals	MTBCR	Pre-Flight Aborts	In-Flight Aborts	MA/KFH (unsch)	MH/KFH (unsch)	MA/KFH (sched)	MH/KFH (sched)	TOTAL MH/KFH	TOTAL MH/KFH	% TOTAL	PMCS/NMCM HRS/KFH	% Total	PMCS/NMCS HRS/KFH	% Total	MH/MA (unsched)	MTTR (unsched)
51	INSTRUMENTATION SYSTEMS	54	2195	52	1768	65	360	320	59	52	59.84	198.90	2.08	4.90	161.80	1.62	0.77	1800.80	1.96	999.20	3.14	2.62	1.37
56	FLIGHT REFERENCE SYSTEM	23	1177	98	919	125	252	457	12	23	35.29	102.60	0.00	0.00	102.70	1.03	0.49	1131.70	1.23	574.30	1.81	2.91	1.52
57	INTEGRATED GUIDANCE/FLT CONT	48	2519	46	1721	67	511	225	109	147	61.90	256.60	5.98	34.00	290.60	2.91	1.39	3345.30	3.64	1636.90	5.15	4.14	2.06
61	HF COMMUNICATIONS SYS	4	53	2174	49	2352	3	38409	0	0	8.54	21.80	0.00	0.30	22.00	0.22	0.11	161.10	0.18	128.00	0.40	2.55	1.40
62	VHF COMM SYSTEMS	9	270	426	217	531	50	2304	0	3	8.07	20.40	0.00	0.00	20.40	0.20	0.00	150.10	0.16	199.60	0.63	2.52	1.41
63	UHF COMMUNICATIONS	5	46	2505	42	2743	4	28807	0	0	7.37	15.50	0.00	0.00	15.50	0.16	0.00	106.60	0.12	52.70	0.17	2.10	1.17
65	IFF SYSTEMS	13	659	175	524	220	88	1309	0	4	23.54	69.10	0.31	0.40	69.50	0.70	0.33	622.90	0.68	0.00	0.00	2.93	1.52
66	EMERGENCY RADIO SYSTEMS	6	150	768	120	960	24	4801	1	0	9.95	17.60	2.27	5.70	23.20	0.23	0.11	28.60	0.00	22.50	0.00	1.77	1.00
67	CNI INTEG. SYS (Secure Comm)	2	98	1176	39	2955	0	0	0	0	3.37	6.00	0.28	0.30	6.40	0.06	0.00	49.50	0.00	0.40	0.00	1.80	0.98
69	MISC COMMUNICATIONS SYSTEMS	21	984	117	656	176	152	758	5	6	39.01	93.70	2.27	3.10	96.80	0.97	0.46	617.20	0.67	718.90	2.26	2.40	1.31
71	RADIO NAVIGATION SYSTEMS	10	215	536	162	711	24	4801	0	0	14.60	38.9	0.59	1	40	0.40	0.19	404.10	0.44	153.20	0.48	2.63	1.44
72	RADAR NAVIGATION SYSTEMS	27	1593	72	1388	83	190	606	12	12	53.83	141.30	0.21	0.50	141.80	1.42	0.68	1075.70	1.17	611.40	2.55	2.62	1.41

Table 3-3 Aircraft A Data Summary including Complexity Factor

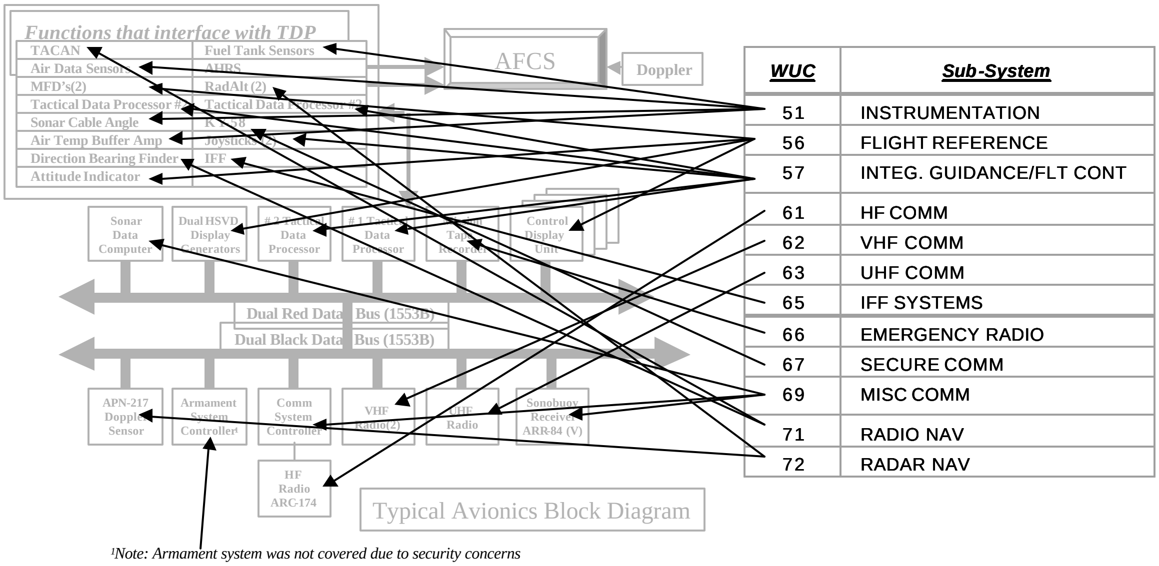
Many Navy and Marine Corps aircraft maintenance crews including all five helicopter systems considered in this work use the same system for the collection of maintenance data. The fact that this system does not make a consideration for the five varied helicopter systems indicates that it does not consider the architecture of the system. The tracking system simply uses the definitions established at some earlier point with no particular consideration for the architecture of the specific helicopter system being tracked. In order to address this issue, an avionics system block diagram was first acquired for each of the five aircraft. A typical system block diagram is shown in Figure 3-1. The subsystems for each of the five considered systems were then mapped to an avionics system block diagram in order to assure that no significant discrepancies existed between the assumed standard architecture imposed by the tracking system and the actual avionics architectures recognized by the original designers. A sample of this mapping is shown in Figure 3-2.



Functions that interface with TDP #1

TACAN	Fuel Tank Sensors
Air Data Sensors	AHRS
MFD’s(2)	RadAlt (2)
Tactical Data Processor #2	Tactical Data Processor #2
Sonar Cable Angle	KY-58
Air Temp Buffer Amp	Joysticks (2)
Direction Bearing Finder	IFF
Attitude Indicator	

Figure 3-1 Typical avionics system architecture block diagram



[1]Note: Armament system was not covered due to security concerns

Figure 3-2 Sample mapping between data tracking system and architecture

With the five aircraft analyzed, there were no significant discrepancies identified between the way that the data was collected when compared with the original architects avionics system view. Each subsystem in the block diagrams mapped to one of the categories in the data collection system and did so uniquely. In other words, it mapped to only one category and the groupings were consistent. For this reason, Figure 3-1 and Figure 3-2 are only included as an example for aircraft type A and for purposes of demonstration of the method followed. For the cases studied, this step had no impact. If there had been a discrepancy, the data would have to have been further arranged in order to align with the original architecture because that could have provided further areas of search for partitioning guidelines.

Each of the five data summary sheets was then normalized as required using the complexity factor previously established. The general idea was to calculate a value for each parameter that was specific to complexity and that could be used to make reasonable comparisons. All but two parameters required normalization. Normalization for the parameters "maintenance hours per maintenance action" and "mean time to removal" (MTTR) was not appropriate since they both refer to time per action. The first refers to the number of hours required to complete the given action. MTTR is similar but also considers the possibility of more than one person completing the action. MTTR is then always either the same or smaller than MH/MA. These parameters are described in Table 3-4. Neither of these parameters is particularly useful for the analysis in this work and are included here only for completeness

Parameter	**Definition**
Unscheduled, MH/MA	Unscheduled Maintenance Hours per maintenance action.
Unscheduled, MTTR	Unscheduled Mean Time To Repair reflects, on average, the time in hours that a maintenance action associated with this part took to complete given applied personnel

Table 3-4 - Parameters not requiring normalization

Of the parameters that required normalization, it was performed in one of two ways. Parameters whose value simply accounted for the number of times an event occurred within the period covered were divided by the complexity factor. As an example, Total Removals is a number that simply counts the number of times a component was removed and replaced in an aircraft. For the Instrumentation Systems category for aircraft type A, there were 2,195 removals during the 2-year period. The complexity factor for this category is 54. Therefore, for the Instrumentation System, 2,195/54 equals 40.6 normalized total removals. In the data studied, there were 12 parameters normalized by dividing by the complexity factor. These are described in Table 3-5.

Parameter	**Definition**
Unscheduled Removals	*Number of times part was removed from fully operational system in response to a maintenance action.*
Cannibalization Removals	*Number of times part was removed from grounded system in response to a maintenance action in a second aircraft.*[28]
Total Removals	*Unscheduled Removals + Cannibalized Removals*
Pre-Flight Aborts	*Number of times that a mission was aborted during pre-flight preparations due to maintenance required in this specific part*

---

[28] Cannibalization refers to the removal of a part from an aircraft that is already in a non-mission ready state because of a required maintenance action in order to fulfill a need in another aircraft. A large number of these occurrences can indicate a particularly unreliable part with a very high demand rate and/or inefficiencies in the logistical supply chain.

In-Flight Aborts	*Number of times that a mission was aborted during flight due to maintenance required in this specific part*
Unscheduled Maintenance, MA/KFH	*Number of unscheduled maintenance actions per 1,000 hours of flight*
Unscheduled Maintenance, MH/KFH	*Number of unscheduled maintenance hours per 1,000 hours of flight*
Scheduled Maintenance, MA/KFH	*Number of scheduled maintenance actions per 1,000 hours of flight*
Scheduled Maintenance, MH/KFH	*Number of scheduled maintenance hours per 1,000 hours of flight*
Total MH/KFH	*Total of scheduled and unscheduled maintenance hours per 1,000 hours of flight*
PMCM/NMCM HR/KFH	*Partial Mission Capable Maintenance or Non-Mission Capable Maintenance hours required per 1,000 hours of flight.*[29]
PMCS/NMCS HR/KFH	*Partial Mission Capable Status or Non-Mission Capable Status hours unavailable per 1,000 hours of flight.*

Table 3-5 – Cummulative parameters normalized by dividing by comlexity factor

Mean Time is generally an indication of average time between some occurrence. For example, Mean Time Between Removals (MTBR) is the average number of hours that a device functions before requiring removal. In the data, each component of the subsystems has an associated MTBR and each subsystem has an associated total MTBR. The MTBR of the subsystem is calculated by considering the MTBR of its components. A simple example will illustrate how this works.

Consider a simple subsystem that has 2 components with component 1 having an MTBR of 200 and component 2 having an MTBR of 300. Given that a removal of either of the 2 components will result in a removal allocated to the subsystem, the MTBR for the subsystem is the sum of the removal rates of its components. In this simple example the MTBR for the total subsystem is,

$$\frac{1\text{ removal}}{X\text{ hours}} = \frac{1\text{ removal}}{200\text{ hours}} \overset{\text{(or)}}{+} \frac{1\text{ removal}}{300\text{ hours}}$$

---

[29] This information is difficult to measure and allocate to a particular part given the fact that cannibalization happens mostly after an aircraft has been down. This number can be misleadingly high and for this reason, it is not used widely in this study.

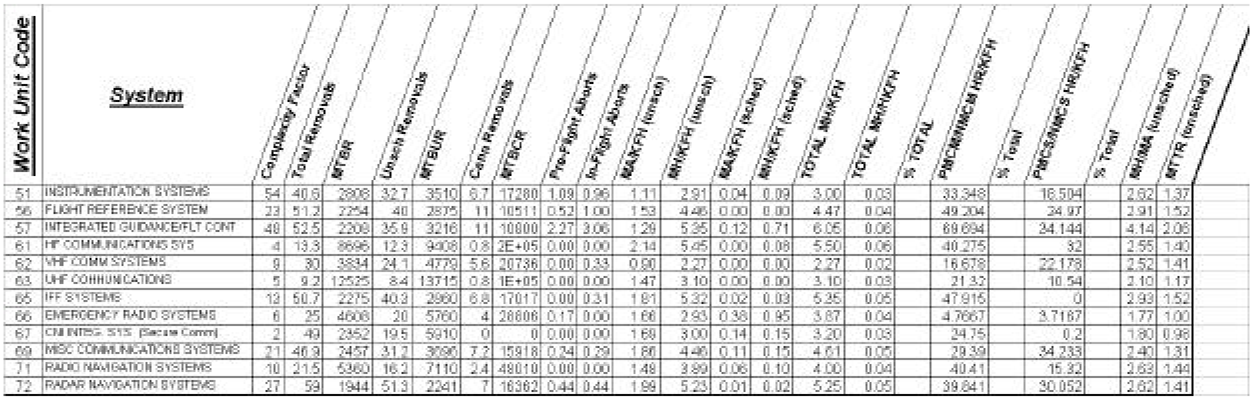
This resolves to X=120; one removal per 120 hours or an MTBR for the subsystem of 120 hours. The numbers extracted from the data were totals per subsystem. The more complex the subsystem (larger number of components), generally, the lower the expected result for numbers like MTBR. In order to normalize with a consideration for complexity, the subsystems MTBR was multiplied by the complexity factor. For the simple example discussed above, we get 2 x 120 = 240, which yields a sort of average mean time per component. This same approach is applicable for any number of components, or any other rate term, and was used for normalizing those used in this study described in Table 3-6.

Parameter	**Definition**
MTBUR	Mean Time between Unscheduled Removals - Average times between unscheduled removals. MTBUR = (Flight Hours Exposed) / (Unscheduled Removals)
MTBCR	Mean Time Between Cannibalized Removals - Average times between cannibalized removals. MTBCR = (Flight Hours Exposed) / (Cannibalized Removals)
MTBR	Mean Time Between Removals - Average times between removals. MTBR = (Flight Hours Exposed) / (Total Removals)
Unscheduled Maintenance, MA/KFH	Number of unscheduled maintenance actions per 1,000 hours of flight
Unscheduled Maintenance, MH/KFH	Number of unscheduled maintenance hours per 1,000 hours of flight
Scheduled Maintenance, MA/KFH	Number of scheduled maintenance actions per 1,000 hours of flight
Scheduled Maintenance, MH/KFH	Number of scheduled maintenance hours per 1,000 hours of flight
Total MH/KFH	Total of scheduled and unscheduled maintenance hours per 1,000 hours of flight
PMCM/NMCM HR/KFH[5]	Partial Mission Capable Maintenance or Non-Mission Capable Maintenance hours required per 1,000 hours of flight.[30]
PMCS/NMCS HR/KFH[5]	Partial Mission Capable Status or Non-Mission Capable Status hours unavailable per 1,000 hours of flight.[5]

Table 3-6 – Rate parameters normalized by multiplying by comlexity factor

---

[30] This information is difficult to measure and allocate to a particular part given the fact that cannibalization happens mostly after an aircraft has been down. This number can be misleadingly high.



Work Unit Code	System	Complexity Factor	Total Removals	MTBR	Unsch Removals	MTBUR	Cann Removals	MTBCR	Pre-Flight Aborts	In-Flight Aborts	MA/KFH (unsch)	MH/KFH (unsch)	MA/KFH (sched)	MH/KFH (sched)	TOTAL MH/KFH	TOTAL MH/KFH	% TOTAL	PMC/NMCM HR/KFH	% Total	PMCS/NMCS HR/KFH	% Total	MH/MA (unsched)	MTTR (unsched)
51	INSTRUMENTATION SYSTEMS	54	40.6	2808	32.7	3510	6.7	17280	1.09	0.96	1.11	2.91	0.04	0.09	3.00	0.03		33.348		16.504		2.62	1.37
56	FLIGHT REFERENCE SYSTEM	23	51.2	2254	40	2875	11	10511	0.52	1.00	1.53	4.46	0.00	0.00	4.47	0.04		49.204		24.97		2.91	1.52
57	INTEGRATED GUIDANCE/FLT CONT	48	52.5	2208	35.9	3216	11	10800	2.27	3.06	1.29	5.35	0.12	0.71	6.05	0.06		69.694		24.144		4.14	2.06
61	HF COMMUNICATIONS SYS	4	13.3	8696	12.3	9408	0.8	2E+05	0.00	0.00	2.14	5.45	0.00	0.08	5.50	0.06		40.275		32		2.55	1.40
62	VHF COMM SYSTEMS	9	30	3834	24.1	4779	5.6	20736	0.00	0.33	0.90	2.27	0.00	0.00	2.27	0.02		16.678		22.178		2.52	1.41
63	UHF COHHUNICATIONS	5	9.2	12525	8.4	13715	0.8	1E+05	0.00	0.00	1.47	3.10	0.00	0.00	3.10	0.03		21.32		10.54		2.10	1.17
65	IFF SYSTEMS	13	50.7	2275	40.3	2860	6.8	17017	0.00	0.31	1.81	5.32	0.02	0.03	5.35	0.05		47.915		0		2.93	1.52
66	EMERGENCY RADIO SYSTEMS	6	25	4608	20	5760	4	28806	0.17	0.00	1.66	2.93	0.38	0.95	3.87	0.04		4.7667		3.7167		1.77	1.00
67	CNI INTEG. SYS. (Secure Comm)	2	49	2352	19.5	5910	0	0	0.00	0.00	1.69	3.00	0.14	0.15	3.20	0.03		24.75		0.2		1.80	0.98
69	MISC COMMUNICATIONS SYSTEMS	21	46.9	2457	31.2	3690	7.2	15918	0.24	0.29	1.86	4.46	0.11	0.15	4.61	0.05		29.39		34.233		2.40	1.31
71	RADIO NAVIGATION SYSTEMS	10	21.5	5360	16.2	7110	2.4	48010	0.00	0.00	1.48	3.89	0.06	0.10	4.00	0.04		40.41		15.32		2.63	1.44
72	RADAR NAVIGATION SYSTEMS	27	59	1944	51.3	2241	7	16362	0.44	0.44	1.99	5.23	0.01	0.02	5.25	0.05		39.841		30.052		2.62	1.41

Table 3-7 Aircraft A Data Summary, Normalized

The normalization approach described was applied to each of the five summary data sheets for each of the aircraft. A sample of a normalized sheet can be seen in Table 3-7.

Once the data were normalized, the next step was to compare the data from the subsystems that make up the avionics system in order to identify the outlying subsystems that might offer an opportunity for improvement. The data were plotted in order to facilitate these comparisons. A sample is shown in Figure 3-3. Once the

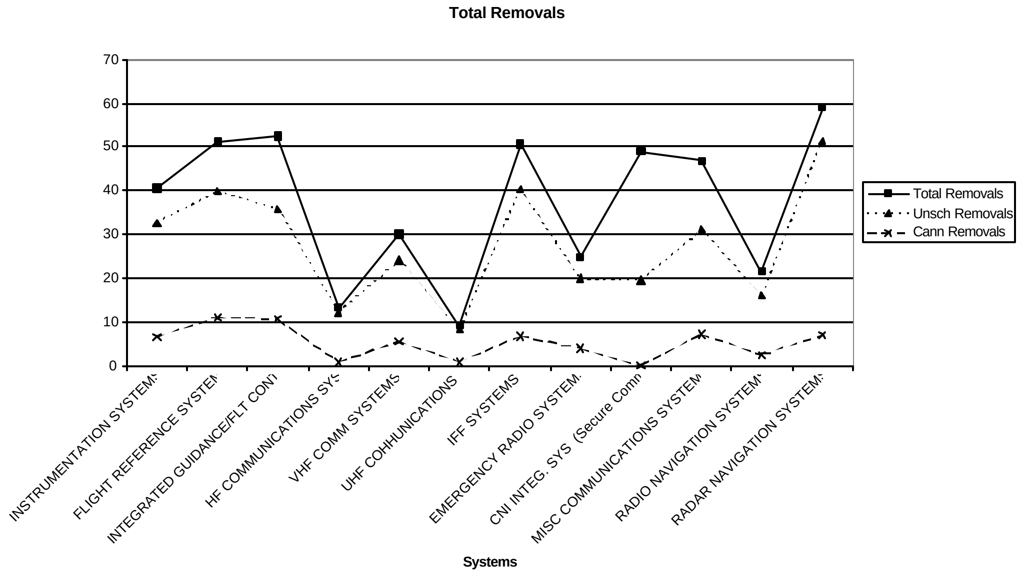


Figure 3-3 Sample plot of normalized removals used to identify outliers

outlying subsystems were identified, a comprehensive review of the details of the data was made in order to identify new opportunities for partitioning that might improve the overall maintenance performance of the system.

A different partitioning approach focusing on reducing LCC that performs identical function is conceptualized. The last step is to try and quantify the options being considered. Based on actual data, estimate contribution to the cost of maintenance by 'discrepancy'. These costs were calculated in terms of Net Present Value (NPV) in order to be reasonably valid for comparison. An estimate for the expected cost reduction with new conceptualized partitioning is made in order to make a comparison. If the cost comparison appears fruitful, a generalization is made in an attempt to establish a Partitioning Insight.

4 Data

In support of this work, two types of data were collected. First, a series of interviews were conducted with experienced avionics system architects. The main purpose of these interviews was to understand the current state of practice amongst these architects with respect to the architecting process and with specific interest in the way that partitioning decisions are being made. The second set of data consisted of maintenance data collected from five different helicopters operated by the US Navy and Marine Corps. These data were used to validate and develop the hypothesis pursued in this work.

4.1 Results of Interviews with Avionics System Architects

This section summarizes the results of the interviews given to the group of Avionics System Architects. The system architects interviewed did not generally have concise processes for system architecture or partitioning decision. Some still identify their work as being entirely experience based and as being more of an “art” than a science. There is significant skepticism amongst the group of architects from SA toward the success of implementing a standardized system engineering process that covers architecture.

From the set of questions used for the interview, only a subset of the questions offered insights that were applicable to the development of the hypothesis in this thesis. These are discussed below. Interview questions and answers that did not have particular supportability of the effort were omitted from discussion.

4.1.1 Important considerations in system architecting

The intent of the first question was two-fold. First, it was important to identify the critical aspects of architecting and determine if the partitioning was an aspect that architects consider when designing new systems. Second, the question was also intended to help orient the interviewees toward a fairly comprehensive context of system architecture. The question was phrased as follows:

Question 1. When architecting a new system, what are the most important considerations? Are any of these particularly reflected in the partitioning of the system? (Please include any and all aspects that are relevant to life cycle value; not necessarily only “engineering” issues.)

The interviewees listed particular items that were felt to be important. These were not provided in any particular order but noted as they came to mind. The large variety of responses are listed below:

- Functional requirements
- Development costs
- Cost per aircraft
- Technical risk
- Partitioning
- Availability of subsystems
- Size
- Weight
- Project development schedule
- Planning for logistic support
- Timeline
- Standards and emerging standards
- “function to cost”
- Buy-in of team
- Understand industry
- Manufacturing constraints
- Balance cost and function
- Flexibility
- Open architecture
- Maintainability
- Design to cost
- Reliability
- Commercial off the shelf (COTS)
- Cost target
- Certifiable
- Mission profile
- Failure modes
- Data flow
- Redundancy
- Equipment repair costs

Some of the interviewees were more concise noting that cost was the biggest consideration. Another architect said, “We never work with a clean sheet. Most times, there is baggage in the form of legacy and that is what defines the things that are important for that particular architecture.” The idea of balance between a series of parameters was predominant amongst most of the interviewees. All of the individuals interviewed noted that a very important consideration was the partitioning of the functions to components within the architecture. This helped to support the practical applicability of this thesis.

The most interesting aspect of these answers is that each of the architects had a different opinion with respect to what is important in architecture. This indicates very little commonality between the approaches of each architect. With varied approaches come varied results. Interestingly commonality is a stated goal of each organization from which architects were interviewed. But without a common understanding of the important goals to pursue, it is most likely that the results will miss the mark. The responses to the first question were the first indicator that, for the most part, each of the architects was working independently without a coordinated enterprise wide strategy.

4.1.2 How should partitioning be done

The intent of the second question was to identify each individual’s general philosophy towards partitioning without consideration or bias toward the level of formality associated with his or her processes. The wording for this question was:

Question 2. When architecting a new system, how should the system be partitioned?

Many of the architects felt that partitioning is a function of the specific project. Flexibility was mentioned often with many references to common architecture. A few of the individuals identified partitioning as one of the most important aspects of architecture. A few partitioning “rules” were given. Here are a few examples:

- Partition so that you minimize the number of nodes through which data travels.
- Flight critical functions should be isolated from mission subsystems.
- Specialized function should remain close to sensors.

- Handle mandated hardware as a given, focus on what is “up in the air”. “Military off the shelf (MOTS) can be put aside since these items generally will not change”. Remainder is evaluated with respect to system response and maintenance.
- If you get the chance with a clean sheet, opportunities for functionality to be located somewhere else can be more economical.

The general consensus was that certain functions needed to be separated from others. The required separation between flight critical functions and mission functions was cited several times as an example. To some extent, this type of answer was an indication that many different interpretations of the term “partitioning” existed. Even after the interviewer provided a definition, most of the architects interpreted partitioning in the context of some sort of existing, baseline architecture. The group is immersed in a culture with a clear assumption that legacy subsystems are a given and that the only real opportunities for “partitioning” are present in the integration effort of the various subsystems. This may be a shortsighted assumption that could miss some opportunities for improvement in LCC, which will be discussed later in section 4.2. Although many of the architects noted concerns with LCC, none showed concrete evidence that it directly influences their approach to partitioning or system architecture.

4.1.3 Structured approaches to System Architecture

Question 3 was designed to identify if the methods used for architecting discussed in question 2 were part of a structured approach. The question was:

Question 3. Do you use a particular structured process in defining system architectures? If so, please provide an example.

Of the twelve architects interviewed, only three answered yes to this question. There was no clear division down the company lines for this answer. In other words, architects from one organization answered yes and colleagues sitting across the aisle from them answered no. The architects with a “structure” approach have developed them for their own use. The processes referred to are neither standard nor institutionalized. They are arguably not structured since they are used by a single individual and are applied to few programs sometimes with years of separation.

Inevitably, lessons learned in one experience are not fully applied to the next project given the loose documentation associated with these processes.

From the group that indicated that they did not follow a structured approach, several stated that clearly a standard process would be a great improvement and that they felt one was needed desperately. A standard process was felt to offer the ability to keep historical data on the process that would facilitate future programs. One of the big problems mentioned by some of the architects is the inability to estimate efforts with any level of accuracy as a result of not having the “real” numbers from the previous experiences. Each time that a new project is planned, it is done with little quantified data from previous projects since the information is not readily available. There were other architects that said that a structured approach could never be implemented because the activity is too complex and typically involves so many variables. This position of system engineering being viewed as an “art” is clearly a cultural obstacle that is going to be difficult to overcome depending on how predominant that thought is through the whole organization. It is possible that the perspective of these individuals may change as they gain more experience with a structured approach. Nevertheless, it is critical that anyone who attempts to incorporate a process change be aware of this type of obstacle because it can easily jeopardize the success of the implementation. Steps to mitigate this issue can be taken although they will not be discussed here since it would be well outside of the scope of this work.

4.1.4 Differences and similarities in recent architectures

The next question was designed to identify trends in system architecture that might be on going over the last decade or so. The question phrased as:

> *Question 4. When comparing the architecture of the last three systems that you defined or worked on, what are the notable differences? Similarities?*

The answer to this question provided some interesting insight into how current avionics systems compare to past systems in the architect’s extensive experiences. The differences noted indicated changes in complexity, architectural philosophies, and customer sophistication. The interviewees believe that there have been changes in complexity although not necessarily an increase in complexity. The opinion was that functional complexity has increased but architectural complexity has actually decreased. The point is that although the systems tend to be much more capable, the increased use of standard communication standards and more standard processors has made integration easier and more flexible. This increase in flexibility also increases the importance of partitioning because particular functions might have several possible components in which they can be implemented. Deciding the best allocation becomes more valuable to the total value of the system since each decision will undoubtedly involve some tradeoffs between performance and costs.

The power and popularity of modern desktop computers has contributed to some changes in avionics system architectural philosophies. Many of the architects made comparisons to the PC architecture. Like in PCs, the term “plug-and-play” has become popular. This term refers to the ability to easily integrate new functions into an existing architecture without having to write special software. Having some very specific specifications for the software that controls the interfaces allows for this capability. Along with this, follows the concept of “open architecture” that refers to the existence of a system whose interfaces are clear and well understood, i.e. no proprietary interfaces. These two concepts, common in the PC world, are still under consideration for the avionics systems. Some of the architects feel that these concepts will also inevitably be common in the avionics systems. This ability to plug-and-play would be a great advantage to reducing the costs of establishing new systems once the interfaces are standardized. It would also have a profound impact on partitioning since partitioning will be directly impacted and controlled by the standards that control the interfaces. Skeptics of this concept point to the significantly smaller market size and longer system life that would make the initial high cost for developing the standard and flexible interfaces a big business challenge. There is some doubt that the savings resulting from having a plug-and-play capability would justify the expense of developing the standard interfaces.

Because of the high clockspeed[31] of many consumer technologies, aircraft customers are much more aware of the capabilities that are possible with modern avionics systems. Advances in products like PCs, cellular telephones, and hand-held Global Positioning System (GPS) have made the average helicopter system customer much more educated. This has been, for the most part, a good thing since it allows for better customer input with respect to the system's functionality. Unfortunately, it also complicates the work of the architect. The architect now needs to manage much higher expectations including sometimes having to explain why certain apparently common and inexpensive functions are either not possible or very expensive in their aircraft system. Expectations of faster communication standards that are popular in the PC environment, for example, will likely have a very short life before becoming obsolete and are not always compatible with systems that have life spans measured in decades. Pressure to use COTS does not assist with standardization and becomes another challenge to architecting and partitioning.

4.1.5 Partitioning that resulted in problems

Question 6 was designed to extract some examples of situations where a particular partitioning approach has resulted in unforeseen problems. The hope was that it might identify other interesting aspects of partitioning that might be investigated with an eye on lowering LCC. The question was:

> *Question 6. Can you identify some specific instances where the current partition of functions in the architecture has resulted in challenges relating to the supply chain, assembly, maintenance, logistics, product support, or upgradeability?*

A few examples were selected from the set of answers. These were generalized in order to avoid references to specific equipment and projects.

- The processing on a weapons control system was partitioned without consideration for the effect of latency. Data samples at 10 Hz was assumed. When it was

---

[31] Fine, Charles H. *Clockspeed, Winning Industry Control in the Age of Temporary Advantage*. Reading, MA: Perseus Books, 1998.

realized that 250 Hz sampling was required, a very significant redesign of the architecture was required in order to address the requirement. This was a result of lack of analysis before the partitioning decision was made.

- A Multi-Functional Display (MFD) was selected that used a peculiar interface protocol. The chip set associated with this interface never became the dominant design. MFD had to be dropped because it was not supportable long term. This was a case of taking a technology risk and losing.
- A particularly complex radio required lots of processing power from the Control Display Unit (CDU) controlling it. In order to achieve a “more modular” approach, the architecture of the existing CDU was re-designed. The resulting efforts and overhead was a financial and processing challenge.

Discussing this last example, the actual problem was not that the CDU didn’t have enough computing power but that the added functionality lead the customer to believe that the present integrated software architecture of the CDU would be better if it were more modular. The argument was that a more modular design would facilitate longer-term software maintenance and upgrades. At the insistence of the customer, the software was re-designed into a more modular architecture. When the partitioning decision was originally made, the processor had plenty of throughput remaining so a further decision was made to incorporate a function that had been re-located in one of the other on-board processors. Once the software was redesigned and the customer increased the required processing by adding more functionality, the CDU was significantly loaded with respect to throughput. It turned out that with the additional overhead, the throughput margin specified in the contract was not achievable so the customer took responsibility and modified the specification in order to allow the design. The redesign required all of the functionality to be re-tested, probably tripling the original effort. At the conclusion of the program, the cost was much higher then expected so the customer adjusted by descoping the effort. The “descope” resulted in removing some of the additional functionality. Citing issues with complexity and additional costs in order to now remove some of the functionality, the design approach returned to an integrated architecture approach for the CDU software abandoning the modular approach. The conclusion was that no additional functionality was provided, lots of additional costs were incurred that resulted in the cancellation of the project, and no further advantage in maintainability of the software was attained.

The reasons for this are directly traced to serious mistakes in the partitioning of the system when the new project was conceived. First, the concept of modularity should have been better evaluated in order to confirm that it would indeed result in maintenance costs that would justify the additional costs of redesigning the software. The decision was made at the insistence of a few specific customer representatives whose software experience was limited. It was felt by all of the developers that the re-partitioning of the CDU software offered very little advantage in the maintainability of the software but their voices were not heard by the architect who should have spent more time to understand all of the implications. The second partitioning mistake was made when the decision was made to change software from another processor to the CDU. This was perceived as making the design “cleaner”, but, in fact, was completely transparent to the user. The change contributed additional costs through the implementation, integration, and testing of the newly located software. It offered no advantage. When the decision was made to incorporate the changes, no evaluation of the processing power required was made in order to assure that the CDU had enough throughput to handle the changes. The partitioning was decided before the implications were well understood. Understanding the implications is important in almost any aspect of architecture. When it comes to partitioning decisions, it becomes critical because it really defines the architecture to an extent after which change is generally very difficult.

4.1.6 Methods of review for system architecture

Understanding the role of architectural reviews was the focus of this question. The question was:

Question 8. What methods (either formal or informal) are used in order to review the design of a particular system architecture? Who is involved in these reviews? When are they held with respect to the project’s timeline?

Once again, the varied set of answers suggests that there are varied interpretations of the system architecture. At SA, architectural review occurs only sometimes and then, only to a limited extent. All of the architects were familiar with the MIL-STD-2167A events referred to as preliminary and critical design reviews. The PDR is a review that usually involves the customer and the engineering staff involved in a discussion over the preliminary considerations for the design of the proposed project. The CDR is much more detail and is intended to define to fair detail the design, as it will be implemented. The problem is that since this standard has been removed in order to reduce costs, there are varied interpretations of necessary content for these reviews. Since these reviews are more or less catered to the customer representatives in attendance, they often contain details in areas that might not include system architecture. In addition, it is common that representatives from the customer do not understand the details of a system architecture discussion so a detailed discussion might not be particularly fruitful. The few reviews that occur seem to happen very informally and are initiated and completely controlled by the judgement of the original architect who may not be impartial. The lack of standardization within the reviews reduces learning that could be made from one system development project to the next.

4.1.7 Advantages of partitioning into modules

This question was intended to support the understanding of advantages of partitioning from the architect's point of view. Since partitioning was noted by each as being important, the question was whether they had a basis for believing this. The question was:

Question 9. From your perspective, what are the advantages of partitioning a system into modules, groupings and interfaces with respect to life cycle value? Do you feel that it plays a role in product strategy or platform strategy? If so, how?

The reposes to this question revolved around four main points. Partitioning:

- Allows for improvements in overall system reliability and reduced costs in maintainability.

- Facilitates certification by allowing particular subsystems to be changed without requiring re-certification for the whole system.
- Allows the development process to be staggered so that progress can be made on a project without requiring full detailed definition of subsystems.
- Facilitates the ability of the architect to manage the complexity of the system.

None of the architects felt that there was a significant link between what they were doing with respect to partitioning and enterprise strategy. This was identified, as a result of the "too-busy-to-think" syndrome that the architects felt was prevalent. This "condition" is a cultural characteristic of all of the companies from which architects were interviewed. It refers to the persistent sense of urgency associated with every aspect of the job that makes the individuals feel as though they must always be performing some function that shows evidence of work. This environment developed through many years of management pressure is counterproductive to a good process in the opinion of this author. The lack of attention that many of the preliminary steps get is a major source of problems in the long term. As noted in the example is section 4.1.5, this fury to get started leaves some work incomplete especially in those first few weeks during which some of the most important decisions such as partitioning are made. The adage that "a project makes its biggest mistakes on the first day" comes to mind.

The suggestion that good partitioning results in lower LCC was mentioned by several of the architects with only anecdotal data being provided as proof. It was something that made sense to them, but not anything that they had ever pursued or quantified. This, of course, was interesting because it aligned well with the hypothesis brought forth in this thesis.

The notion that partitioning decisions can assist in certification process was very strong from the architects. Undoubtedly, reductions in development costs are evident as a result. What is important to note is that if the partitioning decisions are made with development process in mind then they are likely not the best decisions from a LCC perspective. As can be seen in Figure 2-7, the development costs of a typical project are in the order of 10% for development and research, while the associated costs for operation and support are more in the order of 60%. This suggests that more focus on the operation and support phases will likely produce more savings.

The suggestion that the ability for the architect to manage the complexity and to stagger development in order to have a more manageable development experience both tie directly to the traditional partitioning approach that has as a goal facilitating the architect job, not minimizing LCC. Because this was prominent opinion, it seems clear that the current approach discussed does not consider the LCC to any significant degree.

4.1.8 Incentives for partitioning into modules

This question intended to identify any present or developing efforts in the area of partitioning that might compel architects to take partitioning and LCC into wider consideration. The question was:

Question 11. Are there particular incentives for you to partition a system into modules?

Several of the architects referred to partitioning as a way to facilitate their jobs by allowing concurrent engineering to occur. The ability to show progress as you go through the development process was identified as an incentive. Other comments referred to reducing costs but always with development costs in mind. Yet, others talked about establishing long-term relationships by trying to maximize upgradeability and how it can stimulate more work in the future.

In this author's opinion, there are very few concrete incentives to motivate proper partitioning. The biggest reason is that the customers still award contracts on the basis of acquisition cost. The operational and support issues are handled separately and usually later in the process. There are some efforts at SA to establish full "turnkey" relationships with some customers. This would mean that SA would be responsible for all aspects of maintenance and support and not just development and production. Depending on how these customers decide to contract these arrangements, this may turn out to be the biggest incentives to date.

When asked what they considered their “system” when they talk about system architecture, there were some interesting answers. Most said the avionics system. One said “platform of performance”. With some explanation, he was referring to system as the full environment in which a helicopter may function. This includes the aircraft and its surroundings including all of the other things that it might have to interact with like other aircraft, targets, etc. Another said the “Universe”. This is indeed the right spirit and offers some encouragement in the development of better methods in this area.

4.1.9 Summary of interview results

A diverse list of important considerations for system architecting and partitioning suggested that the architects don’t generally share similar goals. This is partially related to project specific influences but is also due to a lack of structure and standardized system architecting processes within the enterprise. The lack of standardization reduces the amount of information transfer that could occur between projects. The low levels of formality appear to extend to the architectural review process. This is driven by a management view that does not perceive the significant value of the architecting process and exerts pressure for schedule rather than overall value. Since the structure of new projects have incentives with respect to acquisition cost, it appears that so do the architects. Despite their intent to consider LCC, it appears that a bigger focus on development and production costs still exists.

4.2 Analysis of Maintenance Data

Given the current state of architecting and partitioning of avionics system, the next step was to consider the maintenance data in pursuit of possible ways to increase the emphasis on life cycle costs from a partitioning standpoint.

Using the data from the five aircraft operated by the Navy and Marine Corps, the methodology was exercised in order to validate its application. All 24 metrics listed in Table 3-4, Table 3-5, and Table 3-6 were considered, but four were identified as being particularly interesting from the partitioning perspective. These were Unscheduled Removals, Mean Time Between Unscheduled Removals (MTBUR), Aborts, and Mean Time To Repair (MTTR). Unscheduled removals accounts for the number of removals performed for a particular component. High numbers for this metric may have some architectural implications. If the components were designed to accommodate lots of removals because of technology or manufacturing limitations, reconsidering the partitioning approach may result in improved life cycle costs.

MTBUR indicates the average time in hours between unplanned removals. This metric relates to the reliability of the component. A higher number indicates a more reliable component and may indicate a certain level of stability in design and performance. The functionality of a component with a high MTBUR number may be a good candidate for integration into another component. A lower number signifies a component that is less reliable, and therefore, more costly with respect to maintenance.

The abort metric counts the number of scrapped missions due to the malfunction of the particular component. There are two types of aborts. Pre-flight aborts are recorded if a mission is canceled after the aircraft has been released for flight but takeoff has not occurred. Generally it occurs during the pre-flight checklist. Besides indicating that the specific component has some specific failure rate, a failure of this type can also indicate that the particular component does not provide enough pre-release data to establish its full functionality. The maintenance crew may require supplementary training, or additional test equipment may be required in order to detect the fault before it impacts a mission. From an architecture standpoint, a high number of aborts can indicate a problem with the architecture with respect to lack of system diagnosis or even lack of redundancy. In-flight aborts indicate the number of missions scrubbed because of a particular failure in the system. This metric can be an indication of a component that, besides having some failure rate, can also have characteristics that make fault detection more difficult. This would be the case for functions that have some dynamic aspect for which flight-testing is required. This may identify a shortfall in diagnosis, a requirement for additional ground test equipment, or additional training for maintenance personnel.

The maintenance hours per maintenance action metric provides an indication of the time required for completion of a given maintenance action. A high number for this metric is an indication of a difficult or complicated maintenance action. This number also contributes directly to the availability of the system for mission performance. From an architecture standpoint, particularly high numbers for MTTR can be an indication of bad design or bad partitioning. Sometimes, improved partitioning decisions may alleviate the maintenance problem resulting in lower LCC.

4.2.1 Implications of reduced maintenance on life cycle costs

In order to estimate the LCC for various aspects of the analysis, a fully loaded rate that considered the above implications was desired. The cost of maintenance is a difficult thing to quantify. In a very large establishment like a defense branch, many different organizations are involved in the operational support of an aircraft. A direct maintenance cost of $100 per hour is a reasonable start when it comes to estimating costs. This includes the cost of the actual individual who is performing the maintenance when considering salary benefits, training, etc. Because of other considerations with respect to infrastructure and facilities, another burden factor ranging between 2 and 2.4 is often applied.[32] In the corporate world, a common burden factor used is 1. This brings the cost per hour number up to $300 to $340.

But direct maintenance costs are just the beginning. Reductions in maintenance costs have several other direct savings implications when it comes to LCC. Less maintenance reduces the life cycle costs by:

- Reducing the number of individuals required performing maintenance on the system. (Every 2,000 hours is approximately one man year).
- Reducing the number of spare components that must be purchased for the supporting logistics system.
- Reducing the size of logistics system (overhead associated with the maintenance and management of the logistics system).
- Reducing damage of other components associated with removal and replacement of other components.
- Reducing the tools required for support.

Given all these factors, a rate of $500 per maintenance hour was used when calculating costs for maintenance. This may seem surprisingly high but may be put

---

[32] Spitzer, Cory R., *Digital Avionics Systems*, New York, NY, McGraw Hill, 1993.

into perspective when compared to the infamous $50 fasteners and $600 hammers that were presented by investigative reports of the past. The focus of the various analyses that follow is on the methodology. The actual numbers used for the calculations although arguable, make up only a small part of the tool used for comparison.

4.2.2 Integrating functionality to eliminate high maintenance component

The reduction of high maintenance content components represents one objective when considering system partitioning. The data tracks three parameters associated with part removals. They are unscheduled removals, cannibalized removals, and the total of both. Cannibalization removals are removals from the aircraft after it has already been deemed not operational by a different problem. These are removals that are generally not made because the part being removed is non-operational but because it is needed by another aircraft. High levels of cannibalization removals are an indication of problems with the logistics system not being able to support the need for the given part. This can be due to lack of parts or it can be due to a component not functioning as reliably as expected. The most significant of these removal parameters from a partitioning standpoint are unscheduled removals. For this reason, more emphasis is given to the unscheduled removal numbers.

Unscheduled removals can be a partitioning indicator depending on the reason for the part removal. In our case, three of the five aircraft (aircraft A, B, D) had the most removals in the area Radar Navigation Subsystems (RNS). See Figure 4-1, Figure 4-2, and Figure 4-3.

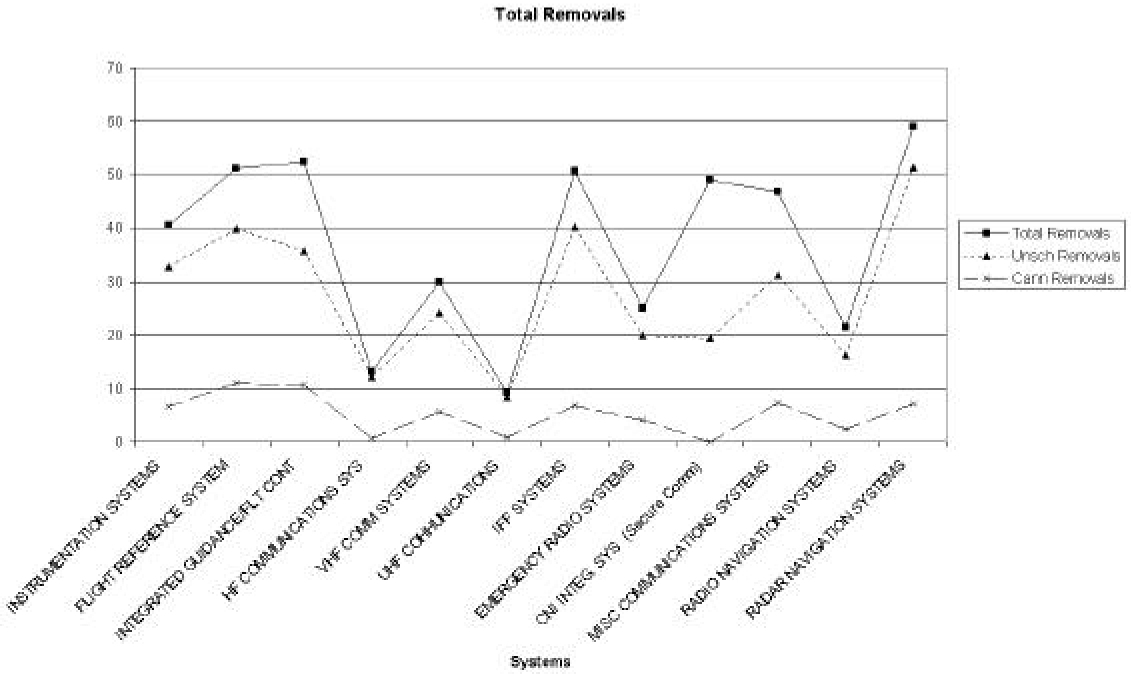


Figure 4-1 Aircraft Type A normalized removals

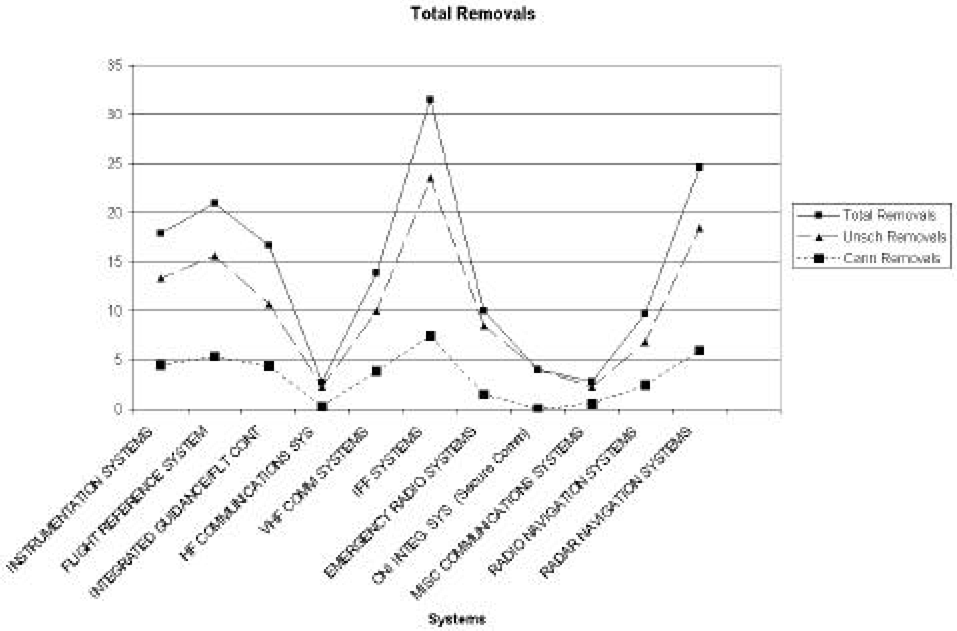


Figure 4-2 Aircraft Type B normalized removals

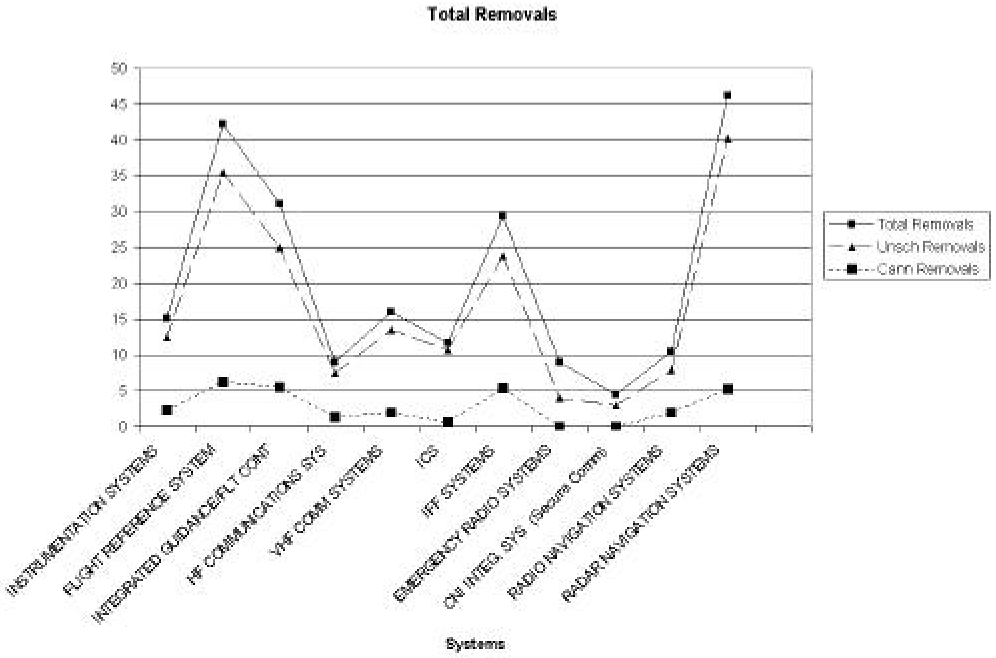


Figure 4-3 Aircraft Type D normalized removals

Looking at the specific data for the components that make up the RNS, there were two major culprits.

4.2.2.1 Radar Wave Guides

One of the components was the wave-guide that is used to transfer microwave signals from the radar receiver to the processor. This type of radar function is only in aircraft type A. These are mechanical tubes that are very susceptible to damage during the installation process. Furthermore, it is particularly challenging to determine if the radar system is working without performing significant ground testing. Oftentimes, the wave-guide is changed based on a non-functioning radar system only to be replaced several times before another component is identified as the problem. This “fly and try” approach results in not only increase and unnecessary maintenance work but also expenses associated with the additional and non-product flying. Further investigation of the history of these wave-guides indicated that at least some of the motivation for having them as distinct components stems from their sensitivity and common need to be replaced. Several things drove the sensitivity. First, due to limitations in the manufacturing process, it was previously impossible to construct a component that integrated the receiver, guides, and processor. According to two engineers in one of the world's biggest radar system manufacturers (Bendix/King), current state of the art would allow the development of a single unit. That would not only assure that the wave-guides were constructed in a consistent fashion eliminating required tuning but would also eliminate the need to replace the wave-guides at all. The reliability of this integrated component would be expected to offer an improvement over the reliability of the current receiver or antenna radome, which are the two components to which the wave-guides connect. Architects appear not to have noticed this opportunity probably because it does not appear to offer any significant improvement in performance.

Let's use aircraft A as an example of how a change of this type might impact life cycle costs. Aircraft A registers a total of 27 components for the radar navigation system. Of these, 4 are wave-guides that account for 691 unscheduled removals. This represents 49.9% of the subsystem's removals allocated to 14.8% of the components. Even though the removal of these wave-guides is fairly simple, remember that they were known to require lots of maintenance when they were first conceived. This maintenance action still adds up to 29.1 maintenance hours per thousand hours of flight. This equates to 3,353 hours of maintenance just related to the wave-guides for the 2-year period covered by the data. If we project that savings over a conservative aircraft life of 20 years, we get a total of 33,530 maintenance hours that would be saved in this fleet of 120 aircraft if this maintenance could be eliminated.

An NPV calculation was done using the following assumptions:

- 1,677 hours of maintenance per year (33,530 Hours/20 years)
- 5.5% yearly rate of return. This was based on the current 30-year Treasury Bill yield with a consideration for 2-3% wage increase.
- Life cycle of 20 years
- A nominal rate of $500 per hour of maintenance that includes all direct and indirect costs associated with operational support as discussed in section 4.2.1.

Net present value was calculated using the following equation:

$$\sum_{n=0,\ 20} \frac{\text{Yearly Cost}}{(1+\text{Discount Rate})^{n}}$$

To calculate the NPV for the maintenance costs of the radar wave-guide, the 1,667 hours of maintenance per year is multiplied by $500 to establish “Yearly Cost” (1,667x500 = 833,500). This cost is incurred for each of the 20 years so each yearly cost is discounted by 5.5%. All of the discounted yearly costs are then summed to establish the net present value.

This results in an NPV of $10.1M for the 120-unit fleet of aircraft type A. This works out to about $84k NPV per aircraft. Depending on the development costs and on the market need for this system, an integrated component that includes all three functions may be desirable.

Architecture A						
	Removals	# of components	MH/KFH	20 Year MH	NPV for Fleet $M	NPV per Aircraft $k
For Radar Wave Guide:	691	4	29.1	33530	**$10.1**	$84.0
Removals for subsystem	1386	27	141.8	163392		
% of subsystem	49.86%	14.81%				
Flight Hours for 2 Yr period	115227					
Number of aircraft of type	120					

Table 4-1 Aircraft A Radar Wave-Guide results

4.2.2.2 Radar Altimeter Indicator

Another component that contributes to a large number of unscheduled removals for the radar navigation system is Radar Altimeter Indicator. This is a dedicated round gage, electro-mechanical device that displays the altitude above terrain to the pilot. It has been used in Sikorsky products for the past several decades without much consideration for alternative. The relatively low cost and the simplicity of their interface has kept this legacy system in place over the years with architects focusing on perceived “bigger fish”.

Speaking to some of maintenance personnel from both Sikorsky and the Navy, it appears that the most common failure in this component turns out to be problems with the mechanical parts of the gage mechanism. Some more advanced instrumentation developers have been proposing the integration of this signal into other displays in the aircraft with only a small, secondary back-up display dedicated to the radar altimeter. Their argument is generally made by emphasizing reduced acquisition costs and user friendliness of the integrated approach. It seems that the data also supports an argument for reduced maintenance costs.

For aircraft type A, the data are shown in Table 4-2. The analysis indicates that over 12,000 maintenance hours were dedicated to the maintenance of the radar altimeter indicator for aircraft type A. This results in an NPV for the fleet of $3.6M.

Architecture A						
	Removals	# of components	MH/KFH	20 Year MH	NPV for Fleet $M	NPV per Aircraft $k
For Radar Altimeter Indicator	189	2	10.5	12099	**$3.6**	$30.3
Removals for subsystem	1386	27	141.8	163392		
% of subsystem	13.64%	7.41%				
Flight Hours for 2 Yr period	115227					
Number of aircraft of type	120					

Table 4-2 Aircraft A Radar Altimeter Indicator results

Since the concept of integrating this function into another display is fairly simple to implement (likely less than $50k development, integration & test), it appears that this may be a good opportunity for life cycle cost reduction in a future model. There would be some other significant savings at acquisition time. Only one backup radar altitude indicator would be required for the system that integrated the information into another display. This display would most likely be a Multi-Function Display (MFD) used to display the flight instruments. The cost of the current indicators is about $12k for the two required per aircraft (1 pilot and 1 co-pilot). A new backup dedicated display costs approximately $5k so that the acquisition savings would be about $7k per aircraft. The change would also free up critical space on the instrument cluster that could be used for other things like larger displays for the flight instruments. The savings are of nearly $4.3M. The resulting savings are summarized in Table 4-3.

	Current Approach Two standalone RadAlt displays		Proposed Option Integrated RadAlt into MFD	
Non-recurring	$0	None required	$50k	Development, integration test
Acquisition Cost per Aircraft	$12k	2 @ $6k each	$5k	Backup display
Cost for fleet of 120	$1.4M	120 x $12k	$650k	(120 x $5k)+$50
NPV of RadAlt Maintenance over 20 yrs	$3.6M		$0	
Total of Acquisition and Maintenance	$5.0 M		$650k	***Savings of $4.3M!***

Table 4-3 Aircraft Type A Summary of Cost Comparison RadAlt approaches

Aircraft type C indicates per aircraft maintenance costs similar to aircraft type A. The data are shown in Table 4-4. This is an identical altitude indicator to that installed in aircraft type A so that the alternative implementation would result in the same acquisition savings of $7k per aircraft. A summary of the savings is shown in Table 4-5.

Architecture C						
	Removals	# of components	MH/KFH	20 Year MH	NPV for Fleet $M	NPV per Aircraft $k
For Radar Altimeter Indicator	16	2	12.7	3346	**$1.0**	$32.5
Removals for subsystem	50	6	74.5	19629		
% of subsystem	32.00%	33.33%				
Flight Hours for 2 Yr period	26347					
Number of aircraft of type	31					

Table 4-4 Aircraft Type C Radar Altimeter Indicator results

	Current Approach Two standalone RadAlt displays		Proposed Option Integrated RadAlt into MFD	
Non-recurring	$0	None required	$50k	Development, integration test
Acquisition Cost per Aircraft	$12k	2 @ $6k each	$5k	Backup display
Cost for fleet of 31	$372k	31 x $12k	$205k	(31 x $5k)+$50
NPV of RadAlt Maintenance over 20 yrs	$1M		$0	
Total of Acquisition and Maintenance	$1.4M		$205k	***Savings of $1.2M!***

Table 4-5 Aircraft Type C Summary of Cost Comparison RadAlt approaches

For aircraft type B, the data are shown in Table 4-6. The analysis indicates that about 20,700 maintenance hours were dedicated to the maintenance of the radar altimeter indicator for aircraft type B during an expected 20-year life cycle. This is a much more compelling case for future integration given the $103.8k maintenance cost per aircraft. This is quite high considering that the maintenance of this component is equal to over 8 times its acquisition price.

Architecture B						
	Removals	# of components	MH/KFH	20 Year MH	NPV for Fleet $M	NPV per Aircraft $k
For Radar Altimeter Indicator	60	2	43	20700	**$6.2**	$103.8
Removals for subsystem	129	7	111	53435		
% of subsystem	46.51%	28.57%				
Flight Hours for 2 Yr period	48140					
Number of aircraft of type	60					

Table 4-6 Aircraft B Radar Altimeter Indicator results

This is an identical indicator to that installed in aircraft type A and C so that the alternative implementation would result in the same acquisition savings of $7k per aircraft. A summary of the savings is shown in Table 4-7.

	Current Approach Two standalone RadAlt displays		Proposed Option Integrated RadAlt into MFD	
Non-recurring	$0	None required	$50k	Development, integration test
Acquisition Cost per Aircraft	$12k	2 @ $6k each	$5k	Backup display
Cost for fleet of 60	$720k	60 x $12k	$350k	(60x $5k)+$50
NPV of RadAlt Maintenance over 20 yrs	$6.2M		$0	
Total of Acquisition and Maintenance	$6.9M		$350k	***Savings of $6.6M!***

Table 4-7 Aircraft Type B Summary of Cost Comparison RadAlt approaches

For aircraft type D, the data are shown in Table 4-8. Aircraft D had a significantly lower cost associated with the maintenance of the radar altimeter indicator averaging to only about $10.5k per aircraft. Even this small potential savings is enough to justify the integration of this function into another modern display. Given the fleet size of 109 for type D aircraft, a summary of the savings is shown in Table 4-9. This amount would be enough to cover the costs of the implementation and validation of the integration bringing the additional benefits of improved user friendliness and reduced space requirements.

Architecture D						
	Removals	# of components	MH/KFH	20 Year MH	NPV for Fleet $M	NPV per Aircraft $k
For Radar Altimeter Indicator	81	1	7.2	3803	**$1.1**	$10.5
Removals for subsystem	201	5	119.5	63126		
% of subsystem	40.30%	20.00%				
Flight Hours for 2 Yr period	52825					
Number of aircraft of type	109					

Table 4-8 Aircraft D Radar Altimeter Indicator results

	Current Approach Two standalone RadAlt displays		Proposed Option Integrated RadAlt into MFD	
Non-recurring	$0	None required	$50k	Development, integration test
Acquisition Cost per Aircraft	$12k	2 @ $6k each	$5k	Backup display
Cost for fleet of 60	$1.3M	109x $12k	$595k	(109 $5k)+$50
NPV of RadAlt Maintenance over 20 yrs	$1.1M		$0	
Total of Acquisition and Maintenance	$2.4M		$595k	***Savings of $1.8M!***

Table 4-9 Aircraft Type D Summary of Cost Comparison RadAlt approaches

The data for aircraft type E does not indicate any significant issues with the radar altimeter, but since it shares the same component style, it was interesting to analyze it using the same approach. The data are shown in Table 4-10 and Table 4-11.

Architecture E						
	Removals	# of components	MH/KFH	20 Year MH	NPV for Fleet $M	NPV per Aircraft $k
For Radar Altimeter Indicator	30	2	2.6	591	**$0.2**	$5.4
Removals for subsystem	97	10	119.5	27167		
% of subsystem	30.93%	20.00%				
Flight Hours for 2 Yr period	22734					
Number of aircraft of type	33					

Table 4-10 Aircraft E Radar Altimeter Indicator results

	Current Approach Two standalone RadAlt displays		Proposed Option Integrated RadAlt into MFD	
Non-recurring	$0	None required	$50k	Development, integration test
Acquisition Cost per Aircraft	$12k	2 @ $6k each	$5k	Backup display
Cost for fleet of 33	$396M	33 x $12k	$215k	(33 $5k)+$50
NPV of RadAlt Maintenance over 20 yrs	$200k		$0	
Total of Acquisition and Maintenance	$596k		$215k	***Savings of $381k!***

Table 4-11 Aircraft Type E Summary of Cost Comparison RadAlt approaches

Since all five studied aircraft use the same component, it is interesting to quantify the potential savings if the customer incorporated the recommended change to all five different types of aircraft. In order to save on development costs, a coordinated effort would be the most cost effective. Table 4-12 shows the summary of the potential savings.

Aircraft type	Acquisition Savings	Maintenance Savings	Reduction in required maintenance personnel
A	$790k	$3.6M	0.30
B	$370k	$6.2M	0.52
C	$167k	$1.0M	0.08
D	$705k	$1.1M	0.10
E	$181k	$0.2M	0.02
Total:	**$2.2M**	**$12.1M**	**1 Person**[33]

Table 4-12 Potential savings for the whole Navy Fleet.

4.2.2.3 UHF & HF Communications

Consolidating relatively stable or mature subsystems represents another objective when it comes to partitioning system. Mean Time Between Removals (MTBR) is calculated by dividing exposure time by the number of total removals for the particular component. Mean Time Between Unscheduled Removals (MTBUR) and Mean Time Between Cannibalized Removals (MTBCR) are calculated by dividing exposure time by the number of unscheduled removals and cannibalization removals respectively. MTBCR accounts for cannibalization removals that have more implications with logistics then with partitioning. Since it also influences MTBR by contributing to total removals, the focus will be on MTBUR.

Upon review of each of the five aircraft types, it was interesting to note that the subsystems with the highest normalized MTBUR (higher MTBUR is better since it signifies that the component functions longer without requiring removal) were all part of the more general category of communication equipment. The data plot for aircraft type A is shown in Figure 4-4 indicating that the functions of UHF and HF communications have better than average MTBUR values. Better in this case because a larger number for MTBUR means that the subsystem is more reliable than the other subsystems.

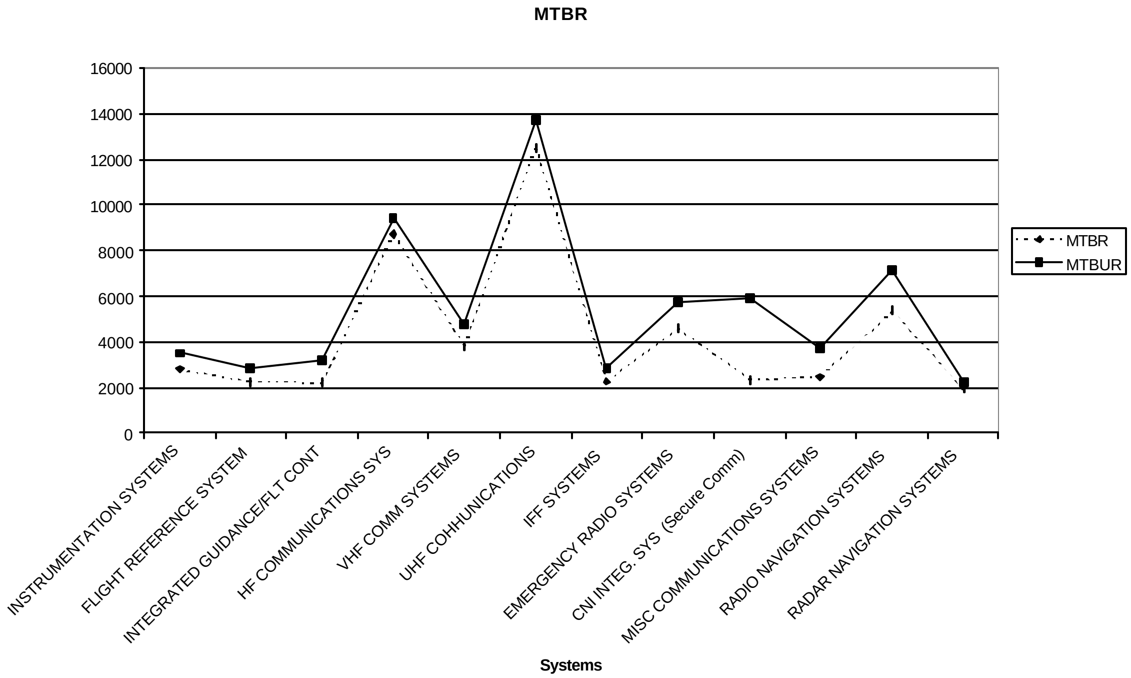


Figure 4-4 Aircraft A normalized MTBUR

Reviewing the maintenance details associated with each of the components that make up each of the two communication subsystems, HF and UHF, revealed that the MTBUR values for each of the components was fairly leveled. This indicates that the

---

[33] Although the savings would total to 1 person, it should be considered that this is spread across 5 fleets and a 20-year span. Taking into account recruiting, training, and associated costs associated with maintaining personnel, the impact is clearly significant.

subsystems are particularly stable and that the numbers registered for the subsystems are not particularly influenced by some more reliable components.

In a case such as this where a particular function appears more reliable than the others in the same system, the question of integration comes up once again. Given the higher reliability, it is possible that a component no longer requires the same modularity in order to allow the ability to remove and replace that particular function without impacting the other functions. If we consider the partitioning of the system in such a way so that these functions are included in the functionality of another component, can we say that it will reduce life cycle costs? As always, we have to consider that there might be other more compelling reasons for the current partitioning that must be factored into our decisions. The communication example in aircraft A effectively covers both cases. Table 4-13 shows the data for the UHF Communications from a MTBUR point of view. As shown, the UHF subsystem has a MTBUR value of 13,715 hours. This is 2.5 times better than the average for the rest of the system. The maintenance time associated with the UHF system, although low, still results in a LCC contribution of nearly $44.8kper aircraft.

Architecture A						
	MTBUR (norm)	# of components	MH/KFH	20 Year MH	NPV for Fleet $M	NPV per Aircraft $k
For UHF Communications	13715	5	15.5	17860	**$5.4**	$44.8
For Avionics System	5423	222	940	1083134		
subsystems % of system	252.90%	2.25%				
Flight Hours for 2 Yr period	115227					
Number of aircraft of type	120					

Table 4-13 Aircraft A UHF Communication results

If the UHF function is to be incorporated into another component, the question then is which component. One good candidate might be the VHF radio. UHF and VHF radios have similar enough power requirements that would allow the use of shared components like transmitter modules for example. As a matter of fact there are already aircraft radios that were developed very recently that incorporate VHF and UHF functionality in the same unit. The reason for doing this was precisely because both bands are particularly compatible with similar hardware. Since no significant differences in hardware are required, UHF and VHF can share the same hardware and perform both functions with little compromise. According to a Rockwell Collins engineer, the complexity of the VHF radio goes up by only about 20% when the UHF functionality is added to it. Using this information we can develop a scenario to compare the actual costs of having separate UHF and VHF subsystems versus one system that performs both functions. Table 4-14 shows the relevant data for this comparison.

Architecture A					
	MTBUR (norm)	MH/KFH	20 Year MH	NPV for Fleet $M	NPV per Aircraft $k
For UHF Communications	13715	15.5	17860	**$5.4**	$44.8
For VHF Communications	4779	20.4	23506	**$7.1**	$58.9
subsystems % of system	286.98%				
Flight Hours for 2 Yr period	115227				
Number of aircraft of type	120				

Table 4-14 Aircraft A UHF and VHF Communication results

First let's account for the costs associated with having the separate UHF and VHF functions in aircraft type A. The VHF radio in aircraft type A can be purchased for approximately $33k. The UHF radio in aircraft type A can be purchased for approximately $18k. The total acquisition and maintenance costs therefore are equal to the cost of both radios ($51k) plus the maintenance costs from Table 4-14, or about $154.7k (NPV) per aircraft if a 20-year life cycle is assumed.

For a system that uses a single unit with both UHF and VHF functionality, we can consider the implementation of a multi-band radio that actually provides some performance improvements. The current price of this type of multi-band radio is approximately $52k. If we simply compare the acquisition costs of each of the two options, $50k for separate radios versus $52k for the multi-band radio, it seems that the best choice is to select the two separate radios as currently implemented in aircraft type A. Of course, the point of this work is to identify methods of reducing LCC so that simply considering the original acquisition costs would be shortsighted. Given the respective increase in complexity, an assuming is made that the maintenance costs of the combined radio will be that of the VHF radio plus 20%. This would result in maintenance cost for the new radio of about $70.68k per aircraft assuming a life cycle of

20 years. The second choice results in a total cost of about $122.68k. That is a life cycle cost savings of about $32k per aircraft. With a fleet size of 120 like aircraft type A, the total savings would be $3.84M.

4.2.2.4 Secure Communications

Another example of a possible merge can be found when analyzing data for aircraft type B. This data can be seen in Figure 4-4 and Figure 4-5.

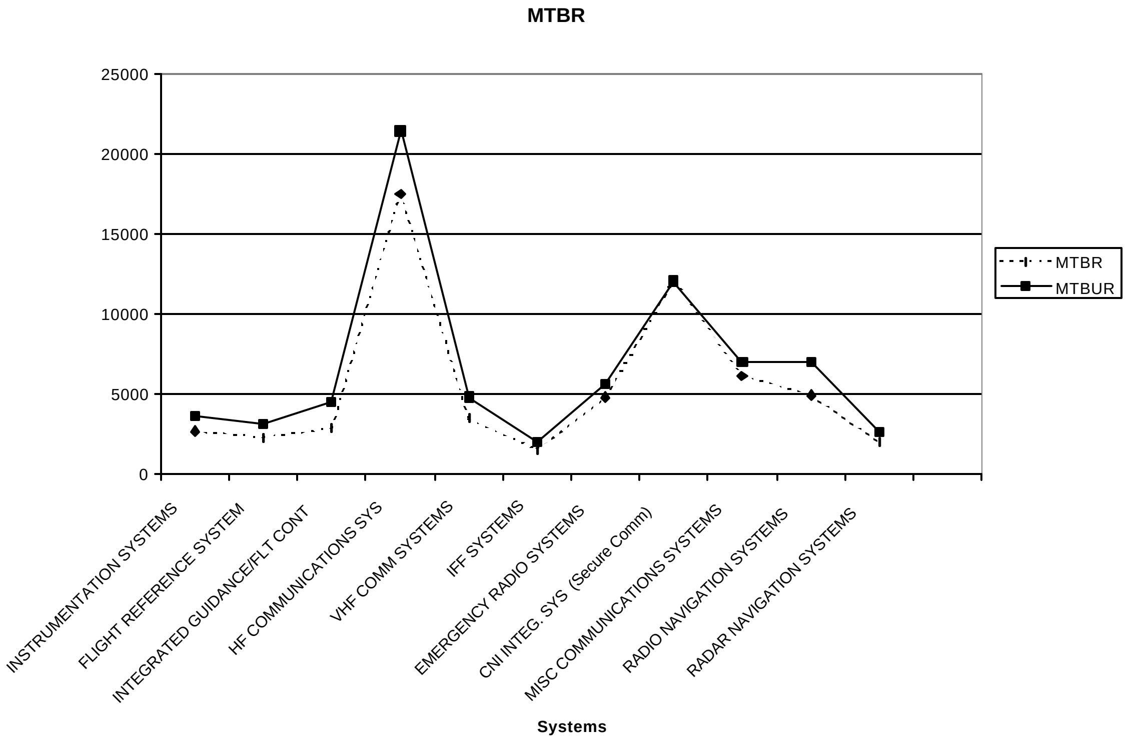


Figure 4-5 Aircraft B normalized MTBUR

The secure equipment being referred to in this case is a device called a KY-58. This device provides encryption of voice communication for the purpose of performing secure communications between two users of KY-58 equipment. The device is installed in-line with the analog signal prior to its transmittal. A reverse process is then used on the receiving end in order to re-generate an intelligible signal. The device is most often used in the VHF band.

Architecture B					
	MTBUR (norm)	MH/KFH	20 Year MH	NPV for Fleet $M	NPV per Aircraft $k
For Secure Comm Equip	12035	6.4	3081	**$0.9**	$15.4
For VHF Communications	4808	65.2	31387	**$9.4**	$157.3
subsystems % of system	250.31%				
Flight Hours for 2 Yr period	48140				
Number of aircraft of type	60				

Table 4-15 Aircraft B Secure Communication Equipment results

As can be seen in the data, the KY unit has a normalized MTBUR value about 2.5 times larger than the VHF communications. After speaking with maintenance personnel, it seems that the biggest problems with the KY units stem from the connectors that are used. Because of the standard design that has been imposed, the connectors commonly suffer damage from installation that later result in malfunction. This is commonly due to bending of the pins within the connector.

Since this device offers a high level of reliability, it suggests that it may be a good candidate for becoming a module to one of the radios. One potential consideration is the incorporation of this function into our UHF/VHF multi-band radio. Let us look at some projected costs in order to make a better evaluation. There is already a version of the previously mentioned multi-band radio that includes embedded KY. The version of this radio with the embedded KY is currently priced at about $56k. Let us assume that adding the KY function to the radio increases complexity by another 20%. Again, we consider two possible paths.

The first is to estimate the costs associated with the current implementation where the KY function is its own dedicated unit. The cost of a KY-58 for purchase is about $9k. As seen in Table 4-15, the cost per aircraft for maintenance projected for a 20-year life cycle is approximately $15.4k. This leads to a total cost of $24.4k per aircraft.

The second path is to consider the possibility of embedding this function into the UHF/VHF radio. The increase in the cost of the radio is about $3k. The increase in the maintenance of the full function radio can be estimated to be 20% of the expected maintenance of the VHF/UHF multi-band radio or .2 x (1.2)($19.5k) = $4.7k. This means that the total added maintenance cost and of having the embedded version of the KY function is $7.7k. When compared to the costs of the first scenario of $24.4k, this option offers potential savings of $16.7kper aircraft. In a fleet size of 60 like that of aircraft type B, this would be a total saving of over $1M.

It can be seen in Figure 4-6 that the secure communication function was also identified in aircraft type D as having a higher then average MTBUR.

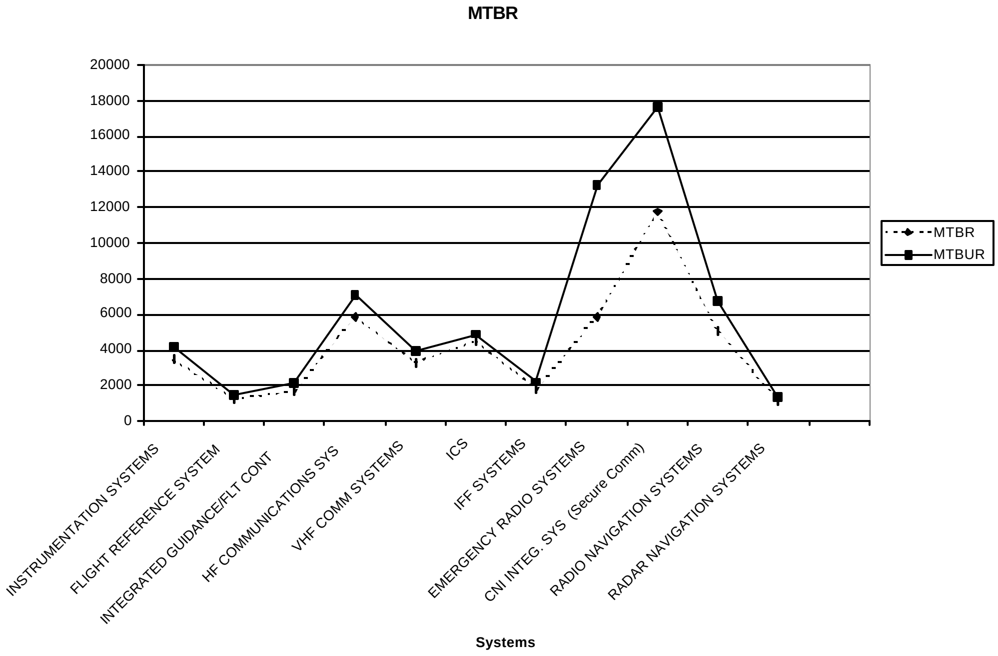


Figure 4-6 Aircraft D normalized MTBUR

The data for aircraft type D with respect to the secure communication equipment is similar to that of aircraft type A and is shown in Table 4-16. This case is very similar to the analysis done for the KY function in aircraft type B. The cost for keeping the KY function separate is actually higher. The maintenance cost of the KY function is projected to be about $29.6k per aircraft. It is not completely clear as to why the costs are almost double for this model compared to aircraft type B. It may be that aircraft type D simply performs more secure communications and that increases the load on its equipment use and this also makes faults more evident. Nevertheless, the total cost of having the KY function in a dedicated box for this aircraft is the cost of the unit, $9k plus the maintenance. That is about $38.6k per aircraft.

Architecture D					
	MTBUR (norm)	MH/KFH	20 Year MH	NPV for Fleet $M	NPV per Aircraft $k
For Secure Comm Equip	17624	20.3	10734	**$3.2**	$29.6
For VHF Communications	3933	166.8	88196	**$26.5**	$243.3
subsystems % of system	448.11%				
Flight Hours for 2 Yr period	52875				
Number of aircraft of type	109				

Table 4-16 Aircraft D Secure Communication Equipment results

As mentioned before, the version of the radio with the embedded KY function increases the cost by about $3k. The increase in the maintenance of the full function radio can be estimated to be 20% of the expected maintenance of the VHF/UHF multi-band radio or .2 x (1.2)($243.3) = $58.4k. When compared to the costs of the first scenario of $38.6k, this option offers potential savings of $5.1k per aircraft. In a fleet size of 109 like that of aircraft type D, this would be a total saving of nearly $560k.

4.2.2.5 HF Communications

An analysis of the High Frequency communication subsystem shows a similar data set. Similar data on the HF communications system is also found for aircraft types B, C, and E. The apparent results of this data might lead one to reach the same conclusion as in the case of the UHF communication function. Pursuing an approach of including the HF functionality would be a bigger challenge and likely one that would bare favorable results.

Architecture A					
	MTBUR (norm)	MH/KFH	20 Year MH	NPV for Fleet $M	NPV per Aircraft $k
For UHF Communications	13715	15.5	17860	**$0.7**	$5.6
For VHF Communications	4779	20.4	23506		
subsystems % of system	286.98%				
Flight Hours for 2 Yr period	115227				
Number of aircraft of type	120				

Table 4-17 Aircraft A HF and VHF Communication results

HF communications requires a significantly different power level than the UHF and VHF bands. The requirement to have a much more powerful transmitter would likely not provide a significant advantage to merging the HF radio function with the others since it would significantly compromise the performance of the other bands and, therefore, the practical performance of the whole unit. This is a good example of a case when it makes sense to maintain the current partitioning given other compelling issues outside of reliability and maintainability. In this case, the indication is that the HF function should remain as it currently exists; a separate and dedicated unit.

4.2.3 Reorganized functional partitioning within subsystem

In some cases, it may be possible to reconsider a different partitioning in order to address a particular maintenance problem that arises. This type of change may even involve only architectural innovation. Architectural innovation describes a situation when existing modules are simply rearranged in an innovative way. Regardless of the specific application, the first step is still to identify candidates. Looking at Figure 4-7 Aircraft Type E normalized removals identifies the ICS subsystem in aircraft type E has having higher then normal removals.

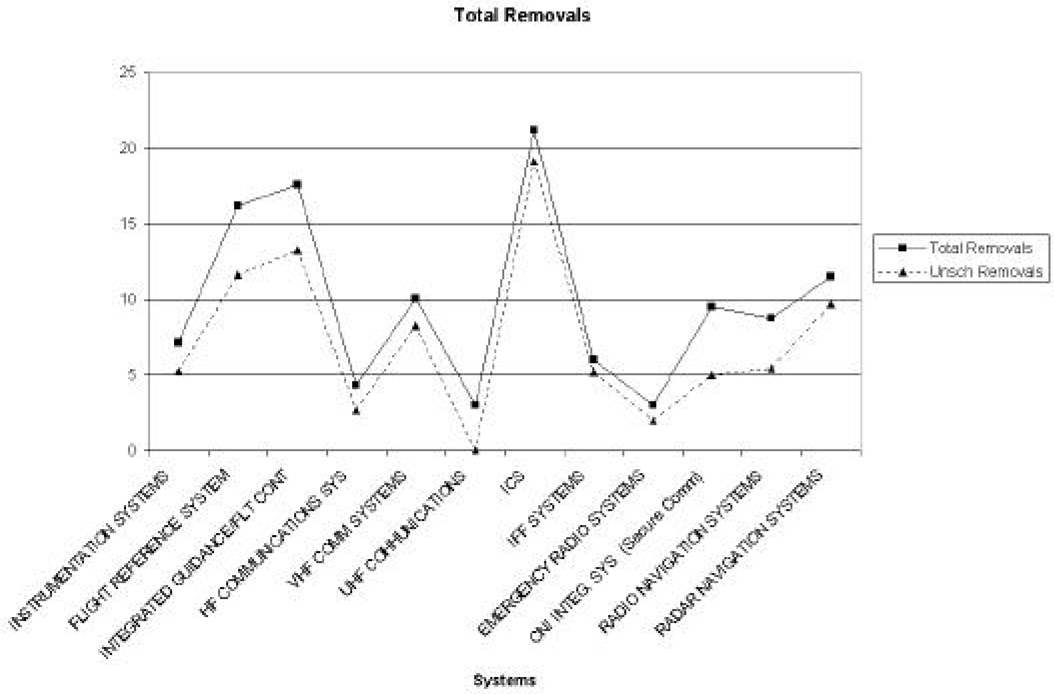


Figure 4-7 Aircraft Type E normalized removals

4.2.3.1 Keycord Assembly Cable

The keycord assembly cable is part of the Inter-Communication System (ICS). This simple device consists of a switch integrated into the cable that connects the crewmember's helmet to the ICS. A majority of the failures occur with the cords installed in the cabin area (not the two used by pilot and copilot). The switch is depressed whenever one of the crewmembers has the need to activate his/her microphone in order to communicate within the aircraft or in order to transmit on one of the radios as selected in the ICS control panel.

In aircraft type E, nearly 42% of the ICS subsystem's removals are associated with the maintenance of keycord assembly cable. This data are shown in Table 4-18. There are two types of failures associated with this cord. One is failure of the switches since they are used extensively under all conditions of flight. The other is wear and tear associated with the constant bending and scraping that the cables experience. The maintenance associated with repairing this cable projects to nearly 20,000 hours at an approximate cost of about $60k per aircraft.

This is a component of the architecture that is not usually considered and is simply installed as a given. One concept that can be considered is the elimination of the cable by the incorporation of the more modern cordless system. Systems of this type already exist but are seldom considered because of their significantly higher cost when compared to the current wired system. The typical aircraft has 5 stations each at a cost of approximately $6k each. A wireless system is projected to cost approximately twice as much. This results in a cost of about $30k for the wireless system over the current wired version. The current helmets would also require replacement at about a cost of $10k each. Savings of $180k per aircraft would justify the additional expenditure and improve the effectiveness and mobility of the crew.

Another possibility for this is the relocation of the keying switches to a less vulnerable area. This could include the relocating the switches to be a integral part of the helmet.

Architecture E						
	Removals	# of components	MH/KFH	20 Year MH	NPV for Fleet $M	NPV per Aircraft $k
For Keycord Assembly Cable	184	4	87.2	19824	**$6.0**	$180.7
Removals for subsystem	440	23	412.5	93778		
% of subsystem	41.82%	17.39%				
Flight Hours for 2 Yr period	22734					
Number of aircraft of type	33					

Table 4-18 Aircraft E Keycord Assembly Cable

4.2.4 Hybrid subsystems with high maintenance requirements

A hybrid system, where one class of components is responsible for an extraordinary degree of maintenance actions, offers another insight into strategies for partitioning. Whenever a mission is aborted because of a fault in the system, the maintenance personnel account for that occurrence. The decision to abort is usually made by the pilot based on knowledge of the aircraft's state of functionality. A call to abort is usually made when the probability of mission success is perceived as not being high. There are two types of aborts tracked. Pre-flight aborts are aborts that occur anytime during the preparation period of the aircraft prior to takeoff. In-flight aborts are a result of a fault detected during the performance of a mission that prevents its successful completion.

The total aborts recorded the avionics system of the five aircraft types studied are shown in Table 4-19. From this table, it appears that aircraft type A has an exorbitant amount of aborts. Since this could be a function of size of each fleet, Table 4-20 shows the same data normalized by the number of aircraft in each fleet.

Aircraft Type	Pre-flight Aborts	In-flight Aborts
A	198	247
B	26	52
C	10	8
D	25	30
E	15	16

Table 4-19 Summaries of Total Aborts per Aircraft Type

Even this normalized data seemed to indicate a large number of aborts for aircraft type A. In an attempt to explain the anomaly, several individuals who are part of these maintenance teams were contacted. Through conversations on the matter, it became a bit clearer that the criterion for counting an abort was impacted by an important difference between the various fleets. The nature of the mission of aircraft type A tended to be more dependent on specific equipment for its success. This resulted in more missions being aborted by aircrews of aircraft type A than there would have been for the other aircraft types under similar circumstances, i.e. their criteria for aborts was different for aircraft type A. This was true for both pre-flight and in-flight aborts. It was difficult to establish if this difference alone accounted for the noted discrepancy but no other difference was identified.

Aircraft Type	# of Aircraft In Fleet	Pre-flight Aborts per aircraft	In-flight Aborts per aircraft
A	120	1.65	2.06
B	60	.43	.87
C	31	.32	.26
D	109	.23	.28
E	33	.46	.48

Table 4-20 Normalized Summaries of Total Aborts per Aircraft Type

Regardless of this notable data difference, the process was continued in order to establish the bigger drivers of aborts within the particular avionics configuration.

4.2.4.1 Integrated Guidance

Analysis of the abort data for aircraft type A identified that the components associated with the integrated guidance function to be the biggest offender. The normalized integrated guidance subsystem contributed 48% of the total avionics pre-flight aborts and of the total avionics in-flight aborts. See Figure 4-8 Aircraft Type A normalized aborts.

The other four aircraft models studied also had the most normalized aborts in this area. There are several possible explanations for this. There are more aborts in this area because:

- Integrated guidance is the most challenging function to evaluate for proper functionality. It makes sense that this function might need the dynamic aspects of flight in order to recognize if a function is working. This must not be true though because if that were the case, pre-flight aborts would be much lower than in-flight aborts, and that is not the case.
- Components of the subsystem have particularly low reliability and fail often.

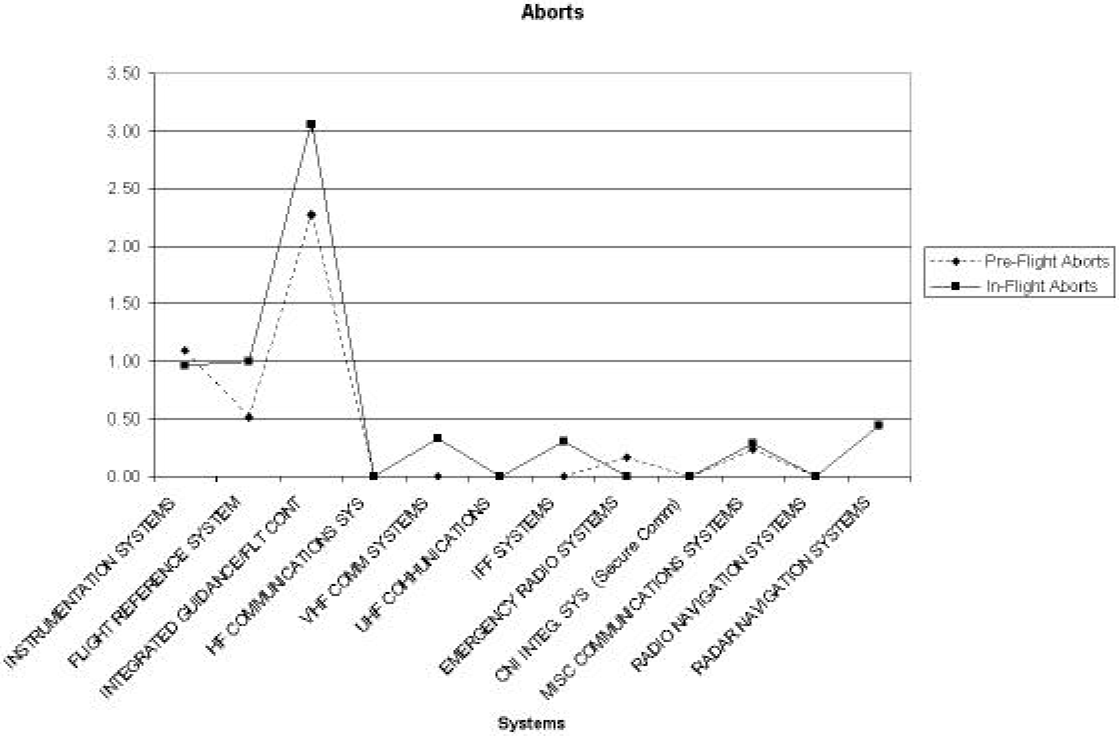


Figure 4-8 Aircraft Type A normalized aborts

Looking at the data in more detail identifies the stabilator system as the biggest contributor of aborts. The stabilator is a control surface located horizontally on the tail cone of certain aircraft used to provide control for the tail section during certain phases of flight. It contains a redundant control system due to its criticality with respect to low speed controllability. An in-flight failure can have catastrophic results. In the case of aircraft type A, components for the stabilator system were involved in 56% of the pre-flight aborts and 60% of in-flight aborts associated with the Integrated Guidance system. This system is made up of both electronic and mechanical components. The inherent complexity in this type of hybrid system results in very large numbers for maintenance. See table Table 4-21.

Architecture A						
	MTBUR (norm)	# of components	MH/KFH	20 Year MH	NPV for Fleet $M	NPV per Aircraft $k
For UHF Communications	13715	5	162	186668	**$56.1**	$467.8
For Avionics System	5423	222	940	1083134		
Subsystems % of system	252.90%	2.25%				

Flight Hours for 2 Yr period	115227					
Number of aircraft of type	120					

Table 4-21 Normalized Summaries of Total Aborts per Aircraft Type

Aircraft type A was originally designed with a fixed low horizontal stabilizer. Handling challenges resulted during flight transitions between hover and forward flight when the main rotor wash would impinge on the stabilator causing the aircraft to pitch up. Other related performance problems were present. The issue was resolved the addition of a stabilator. Every effort was made to avoid having to implement the stabilator but the aircraft required it in order to fulfill the required handling qualities.

There are many helicopter configurations in existence that have a fixed stabilizer including aircraft types D & E discussed in this study. Using a stabilizer, although much simpler, does result in some performance compromises. Although it is not the intent of this author to say whether a bigger effort in order to avoid the stabilator would have been justified, it is interesting to consider the value in maintenance savings that might have been possible had that been the case. For aircraft type A, the documented maintenance during the fleets 20 year life would add up to $56M.

Other options may be possible to the current stabilator configuration. Modifying the existing stabilator actuator mechanism with a small, dedicated hydraulic system may be a way to reduce the high maintenance costs of this function. A very detailed study on this specific subject would obviously have to be done in order to determine the feasibility of this idea. Considering improved levels of Built in Test (BIT) in order to more precisely identify faults with systems of this type could reduce costs. This type of on-board diagnostics improvement could reduce the maintenance time by eliminating the need to needlessly remove components that are fully functional solely as a part of the diagnostics process. In the case of hybrid systems such as the stabilator, a good understanding of the true root cause of the problems could be particularly helpful. If, for example, the electronic components of the system are intrinsically more reliable, partitioning them away from the mechanical components and improving them with BIT may result in a reduction in removals. The ability to identify faults as being associated with the mechanical components rather than the electronic ones would avoid needless removal of electronic components and focus attention on the true faults of the system.

From the results of this study, it seems that the value of having better on-board diagnostics could be higher than normally expected. This may possibly justifying the expenditure of additional efforts in order to improve these functions since the savings could be realized during the maintenance phases of the product.

4.2.5 Network changes based on unreliable network component

The Identification Friend or Foe (IFF) subsystem provides another interesting case that reiterates that LCC is not the only factor to be considered in partitioning decisions. The IFF was one of the components with high levels of relative removals on aircraft types B and C. The IFF provides a very critical function. It allows aircraft to identify each other as either being allies or not (friend or foe). It does this by responding with a specific coded message to interrogations from another aircraft and by performing its own interrogations.

Although projected costs may be compelling, this case serves as a good example of an item that has other contributing external affects. Because the IFF is used as part of a network, interoperability is a very important factor. In order to function well within the network, any aircraft participating in a particular mission or region must have IFF subsystems that are compatible. This network effect then must be considered before making any decisions that are unique to aircraft type B. If any change is to be made, it has to apply to the full set of components that are involved in the full set of participants in the network.

Note that Figure 4-2 Aircraft Type B normalized removals and Figure 4-9 Aircraft Type C normalized removals indicate that the IFF subsystem has higher than normal removals. This leads us into looking at the details of this functionality and its components.

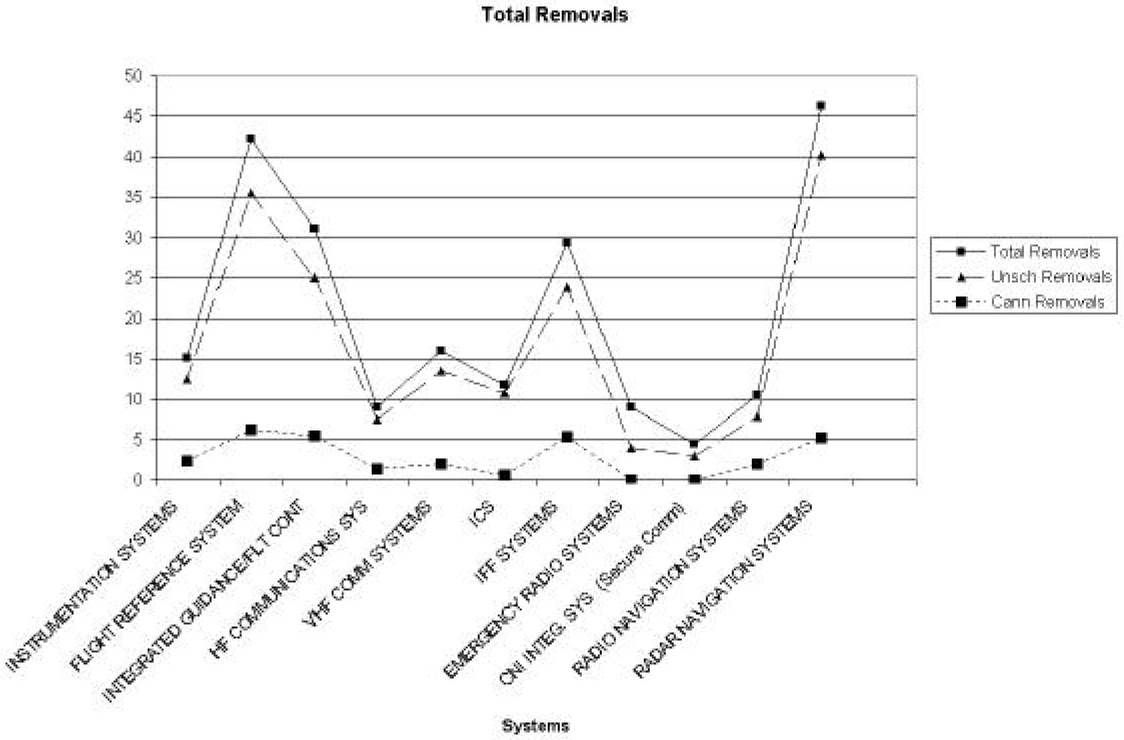


Figure 4-9 Aircraft Type C normalized removals

4.2.5.1 Identification Friend or Foe

The data for this case are shown in Table 4-22 and Table 4-23. The data indicate a significant cost associated with the maintenance of the IFF in aircraft type B.

Architecture B						
	Removals	# of components	MH/KFH	20 Year MH	NPV for Fleet $M	NPV per Aircraft $k
For Identify Friend or Foe	21	1	22	10591	**$3.2**	$53.1
Removals for subsystem	47	2	41.8	20123		
% of subsystem	44.68%	50.00%				
Flight Hours for 2 Yr period	48140					
Number of aircraft of type	60					

Table 4-22 Aircraft B Identification Friend or Foe results

Architecture C						
	Removals	# of components	MH/KFH	20 Year MH	NPV for Fleet $M	NPV per Aircraft $k
For Identify Friend or Foe	14	1	18.5	4874	**$1.5**	$47.3
Removals for subsystem	35	3	41.8	11013		
% of subsystem	40.00%	33.33%				
Flight Hours for 2 Yr period	26347					
Number of aircraft of type	31					

Table 4-23 Aircraft C Identification Friend or Foe results

The data for aircraft types B and C indicates a high level of maintenance for the IFF. This might motivate consideration for a change. In a case like this where there are serious network implications, there can be one of two types of changes. The first is a change localized to the unit without impacting the RF interface. This can be an improvement in reliability such as a change in manufacturing. A change that impacts the RF interface can be only made if the whole network system is taken into consideration. The processing performed by the IFF could conceivably be performed within the capabilities of a more general processor and then the RF be handled by a receiver transmitter module. This could be an alternate to the current partitioning.

The current partitioning in this case seems to make a lot of sense and can be used to reinforce a good partitioning guideline. In a case where a particular function is both critical to the mission in that it performs a vital function for the system and that it also is utilized as a component to a bigger network, then it makes sense to make it common across all other platforms. The advantage is that it is common to all aircraft and that it can be easily interchanged between different models. This encourages the architects to keep the same basic functionality because it makes it almost off limits when it comes to partitioning decisions.

Network effects aside, quantifying the maintenance costs associated with the fleet of five aircraft types analyzed is useful. The results are shown in Table 4-24.

Aircraft Type	Maintenance Hours for 20-Year Life Cycle	NPV, $M
A	19934	$6.0
B	10591	$3.2
C	4874	$1.5
D	7984	$2.4
E	4387	$1.3
Totals	**47770**	**$14.4**

Table 4-24 Summary of Maintenance costs for the IFF system

Considering the total cost of $14.4M for the maintenance of the IFF system, a decision would have to be made as to whether that is compelling enough when compared to the cost of making a significant reliability improvement.

5 Discussion

The study involved a review of current architecting and partitioning methods. This was performed through a series of interviews given to a group of experienced avionics system architects. Then, using a maintenance data from five aircraft types, an analysis was performed in order to apply the developed methodology and validate the hypothesis that improvements in partitioning decisions within architecting can result in a significant reduction in life cycle cost.

5.1 Observations

Since most of the avionics system architects interviewed were from SA, these observations tend to reflect on aspects within that organization. A diverse list of important considerations for system architecting and partitioning suggested that the architects do not generally share similar goals. This is partially related to project specific influences but is also due to a lack of structured and standardized system architecting processes within the enterprise. The lack of standardization reduces the amount of information transfer that could occur between projects. The low levels of formality appear to extend to the architectural review process. The structure is not in place to properly support these functions. Because the information is not readily available, the implications that high maintenance have on life cycle cost, although noted, are not fully appreciated when making high level architecting and partitioning decisions. Since the structure of new projects generally have incentives with respect to acquisition cost, it appears that so do the architects. Despite their intent to consider LCC, it appears that a bigger focus on development and production costs still exists. This is perfectly understandable given that these are the areas that are the most tangible activities from the architect’s point of view. These areas are also those for which the best data is generally available.

5.1.1 System engineering process

The architects all recognized that the current methods for architectural design and partitioning could be significantly improved. Lack of structure and processes was cited as a shortcoming with the current way that things are done. This problem is currently being addressed by the organization's system-engineering group. New procedures are being developed to support this need, which are starting to be used. Although it is likely that it will take some time before they become fully useful, having a baseline process to follow will stimulate necessary improvements. An open approach to the task of developing these new processes will make the resulting product much more useful. First, it will allow for input from the individuals who need to use the process on a daily basis. It will also facilitate the transition into the organization by defusing some of the skepticism that is inherent with this type of change. The new system engineering processes should include a description of the methodology developed in this thesis as another tool in the system engineers toolbox available for implementation. This new tool can assist the architect in the development of new or upgraded avionics system with a particular focus on the system's lifecycle costs. This new methodology not only assists in the engineering responsibility of the architect but also in the management of its development, again, with particular emphasis on life cycle costs. The more structured process to system engineering and architecting will make it simpler to collect important data both for the current project and for the future. Metrics will not only make the existing project more manageable by providing a clearer picture of the state of the program but it will also facilitate future projects by allowing better estimates and management approaches to be implemented. The key point is that the current plan for the development of standardized processes should continue. These processes are critical to the improvement and long term effectiveness of the enterprise and should be designed so that they can be updated as new aspects of system design are developed and quantified.

5.1.2 Legacy subsystems

As indicated by the interviewees, new projects seldom offer the opportunity to start the design from a clean slate. This forces the architects to deal with legacy[34]

---

[34] Legacy subsystems are components that have been around for some time and have an established position within a customer's logistic system.

subsystems. There are fundamentally two approaches that can be taken as an architect when a legacy subsystem is imposed on a new design. A study can be performed in order to understand the implications of keeping the legacy subsystems being encouraged. This must consider life cycle costs associated with keeping the legacy subsystem in order to make an equitable decision as to whether an alternative subsystem should be pursued. The other approach is to take the legacy subsystem as a given part of the architecture and concentrate on the architecture of the remaining undefined subsystems. Keeping legacy systems in the architecture is often less expensive with respect to acquisition, development, and testing. Because current incentives to the enterprise are more related to the acquisition cost than they are to the life cycle cost, most system architects take the legacy as a given and focus the less defined aspects of architecture. This may start to change with a newer trend towards "turn-key" services. This concept has the enterprise involved in the total operational activities of an aircraft system. In some cases, operational costs are even guaranteed to the customer so that if they are surpassed, the enterprise must cover the additional cost. This type of arrangement will likely help to drive the design approaches towards a bigger consideration of LCC. Legacy systems, although part of the reality of system architecting, need to be considered an area of study when developing a new architecture. Simply accepting legacy subsystems as part of a new architecture runs the significant risk of bypassing great opportunities with respect to LCC savings.

5.1.3 Development of system architects

Developing quality individuals for the task of system architecture may be the first challenge to the organization. At SA, the avionics integration organization is approximately 26 years old. Because of the relatively low clockspeed of this industry, the total number of integrated projects that have been developed by the team is limited. In the span of 26 years, the total number of integrated projects that could be used in order to provide experience appears to be less then 20. The typical experience level required before an avionics integration engineer can achieve the level of 'architect' is usually about 10 years.[35] This has to be supported by an appropriate level of formal education and a conscious effort to update the individual's understanding during the same period. Providing the diverse set of experiences required is a challenge that the enterprise has because an engineer typically spends 3-5 years on a single project. Even if an individual has a career of 20 years in the field, he/she can only participate in 4-6 projects during their career. Of course, 20 years is nowadays considered a long time to dedicate to a single organization given the current job market and the options available to individuals with the proper educational background. So even keeping an individual within the organization in order to develop a quality architect is a big challenge.

Because of the challenges noted above with respect to the development of system architects, a new approach needs to be considered. Not only should an organization such as SA have structured and repeatable processes that can guide the system architects, it should also have a well-designed process for developing the high-level system architects that it needs to succeed in the future. This includes identifying the individuals who have the interest, basic education and inherent personnel skills required, and adding the additional formal system related education and experiences to achieve a well-rounded person. The specific characteristics will likely differ from one organization to another but the existence of a defined process is the important aspect. A structured approach to system architect development will likely reduce the current timeline of 10 years for their development and encourage younger candidates to stay and pursue longer, more effective and satisfying careers.

5.2 Maintenance data analysis observations

During the process of acquiring and analyzing the maintenance data in support of this thesis, some interesting issues were identified. These issues are discussed below.

---

[35] Rechtin, Eberhardt. *Systems Architecting, Creating and Building Complex Systems*. Englewood Cliffs, NJ: Prentice Hall, 1991.

5.2.1 Existing maintenance data collection process

The original goal for this work involved acquiring the maintenance data for as many helicopter models as possible. The hope was that there might be an organization that had full access to these data. After searching for this type of data, it became clear the data were not available from a centralized location. Although some data were available, it was only a subset of what was originally expected. It turned out that the data availability was completely dependent on the particular aircraft operator and the collection methods that they were using. Some of the operators had stronger relationships with SA in the area of reliability than others. The US Navy was a strong example and one reason why the five systems used were selected.

From the data used for the analysis, other things were noted. All five systems used the same maintenance data collection system. This was for the most part an advantage since it made the comparison of different models easier to make. The fact that the same database appeared to make no differentiation between the five aircraft models in how it organized the data was a little unsettling considering that each of the aircraft had some differences in architecture. It seems that there is some latitude with respect to how the data is entered into the system. No significant issues were identified with respect to this categorization were found. A better-defined system would help to eliminate the likelihood of a problem occurring. There were some discrepancies when it came to definitions of tasks. A few of the task entries were clearly duplicates resulting in a task being tracked under more than one category. These issues were identified and corrected as part of the analysis process.

The other impression from discussions with maintenance personnel suggested that although some significant effort is made to collect the data, the customer performs only limited formal analysis. This suggests that the data are not being utilized to their full potential in the management of the system or in the design of future systems. The data set is used to identify particularly maintenance intensive components in order to encourage quality improvements from the manufacturers. There was also reference to the use of the data when it comes to the acquisition of new systems. If a component has been particularly troublesome, the customer may request that its function be performed by a different component.

5.2.2 A hypothetical system that would support architecting and partitioning

SA recognizes that not having direct access to the maintenance data for all of its products is a clear shortcoming. The enterprise is working on establishing a data collection system that addresses this. The new system would be usable by any customer regardless of the specific aircraft involved. This will address two important problems. It will make the system standard amongst all the operators and, more importantly, make the data available to SA.

There are a few suggestions with respect to this that will likely make for a more useful system in support of the system architecture and partitioning processes. Like many other situations, the incentives tend to define behavior. If the new system does not offer a compelling reason to change from their current ways, the customers will not embrace it. A few things can be done:

- The new data collection system could be provided at no cost to the customer.
- It should reduce the overall maintenance effort, or at the very least, not directly increase the overhead associated with the maintenance effort.
- Each of the systems being currently used by each of the customers should be studied so that the new system addresses all of the functions currently used by the customer. The new data collection system cannot be perceived as providing less performance than the old system that it replaces.
- Interfaces that currently exist between the data collection systems and other processes need to stay identical. This is important so that the new system does not directly drive changes (and costs) in other upstream systems that received their input from the data collection system.
- A simple interface and a direct connection to SA should be implemented. Possibly a web-enabled system would be good.
- A set of standardized analysis functions should be included.

All of these steps are very important and complementary to each other. Implementing only some is not recommended. The first four points do not provide any significant advantage but are necessary in order to make the system acceptable to the customers. The last point is the actual incentive to use the system since it will provide significant benefit to the customer. A good incentive will compel customers to participate.

In order to take complete advantage of such a data collection system, a link back to the system engineering and partitioning process is required. Not only should the system engineering processes discussed earlier include the consideration of LCC with respect to architecture and partitioning but the data collection system should also provide the required data for analysis. This goes beyond simply making the data collection system available to the system architect. The data acquisition system should provide the input data to an automated system analysis tool that focuses on identifying potential opportunities for alternate partitioning with a focus on LCC.

6 Conclusion

6.1 Results with respect to proposed hypothesis

The question proposed in this thesis was:

- Can a good understanding of maintenance costs drive different partitioning decisions?

Based on the results of the study, the answer is yes. The new methodology demonstrated an improved result with respect to life cycle costs when applied to the avionics systems studied in this work. The study identified several insights that were supported by specific examples within the maintenance data from the five aircraft. Each of these insights is summarized below.

6.1.1 Partitioning Insight # 1:

Integrating the functionality of a particularly unreliable component into an existing component that is already incorporated into the architecture may result in LCC savings. In certain cases, it may be a situation where an old technological or manufacturing limitation has been overcome allowing for an improved partitioning decision. A good general area to consider is displays. The case of the radar altimeter discussed in section 4.2.2.2 is an example.

6.1.2 Partitioning Insight # 2:

Integrating the functionality of a particularly reliable component into an existing component that is already incorporated into the architecture may result in LCC savings. In this type of partitioning, you are taking advantage of the fact that some component has a high reliability that suggests a particularly stable function. Looking for cases where common hardware or software can be shared among more than one function may also be helpful. The UHF radio function incorporation into the VHF radio discussed in 4.2.2.3 is an example. It is also important to recognize when other more compelling considerations play a role in the decision that prevent this type of integration. The HF radio case described in the same section is an example.

6.1.3 Partitioning Insight # 3:

Reorganize the functional partitioning within a particular subsystem with high maintenance costs to address a condition that might reduce the occurrence of damage. The keycord assembly discussed in section 4.2.3.1 is an example.

6.1.4 Partitioning Insight # 4:

Considering improved on-board diagnostics in order to more precisely identify faults with hybrid systems. This type of on-board diagnostics improvement could reduce the maintenance time by eliminating the need to needlessly remove components that are fully functional solely as a part of the diagnostics process. If a particular functionality of the hybrid system appears to have particularly high reliability, partitioning those high reliability portions will allow for better focus on the reliability culprit. The stabilator subsystem discussed in section 4.2.4.1 is an example.

6.1.5 Partitioning Insight # 5:

Components sometimes play a role that is extends outside of the aircraft in which it is installed. Circumstances where network effects are involved offer a bigger challenge but also generally offer higher potential for savings. Improving reliability of these components can have very large improvements to life cycle costs. The IFF system discussed in section 4.2.5.1 is an example.

6.2 Further Study

The process also identified some additional factors that have an impact on the LCC. The most compelling was that of legacy subsystems that contribute significant problems with respect to their partitioning due to the fact that they are more entrenched in the current avionics market and in some logistics systems. Legacy could likely be included in the approach given some further modifications to the methodology. Although legacy will always play a role, a trend toward open architecture is likely to reduce the impact of legacy in future systems.

6.3 Implications to learning and training

Current System Design and Management curriculum emphasizes the spirit of taking a holistic approach to systems architecture. Because of the difficulty in properly communicating this philosophy, it may be useful to consider adding a specific exercise to the program. The curriculum should include one project that requires the comprehensive analysis of an existing project that has enough operational data to support a comprehensive analysis. Soliciting real data from sponsoring companies would be a good approach since they would likely be very interested in the results. This would best demonstrate the various factors that go into making good decisions from the life cycle cost point of view. The project would fit well into the current subject matter in either the System Architecture class or the System Engineering class.

With organizations such as SA, inclusion of maintenance considerations within the system engineering processes is key. In order for the processes to be most effective, a training program must be developed that emphasizes the benefits of following a standardized process. Along with the initial training, there must also be support provided particularly in the beginning of the transition so that implementation issues can be addressed as they arise. An environment that encourages support will be the most successful since many of the changes associated with a more structured approach tend to be in some contradiction to the cultural environment that exists.

Bibliography

Baldwin, Carliss Y. & Clark, Kim B. *Design Rules, The Power of Modularity*. Cambridge, MA: The MIT Press, 2000.

Crawley, E., *System Architecture Lecture Notes – Rev 2*, MIT, November 12, 1999.

Eppinger, Steven D. and Karl T. Ulrich, *Product Design and Development*, New York, NY, McGraw-Hill, 1995.

Fine, Charles H. *Clockspeed, Winning Industry Control in the age of Temporary Advantage*. Reading, MA: Perseus Books, 1998.

International Society for the System Sciences (ISSS) web site, www.isss.org.

Kockler, DSMC Systems Engineering Management Guide.

Lacy, J.A., *Systems Engineering Management,* New York, NY, McGraw-Hill, 1992.

Lang, Jon., *Creating Architectural Theory, The Role of the Behavioral Sciences in Environmental Design*. New York, NY, 1987.

Miller, G.A., *The Magic Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information*, *An essay in* The Psychology of Commination, Basic Books, 1967.

Nadler, Gerald, System Methodology and Design, IEEE Transactions on Systems, Man and Cybernetics SMC-15, 6, 685-97. Dec. 1985.

Neufeldt, Victoria E., Editor in Chief, *Webster's New World Dictionary – Third College Edition*. New York, NY: Webster's New world Dictionaries, 1988.

Rechtin, Eberhardt. *The Art of Systems Architecting*. Boca Raton, FL: CRC Press, 1997.

Rechtin, Eberhardt. *Systems Architecting, Creating and Building Complex Systems*. Englewood Cliffs, NJ: Prentice Hall, 1991.

Reynal, Vincent A. and Cochran David S., *Understanding Lean Manufacturing According to Axiomatic Design Principles.* The Lean Aircraft Initiative Report Series, #RP96-07-28, November 1996.

Spitzer, Cory R., *Digital Avionics Systems*, New York, NY, McGraw Hill, 1993.

Suh, Nam P., The Principles of Design, Oxford Press, New York, 1990.

Appendix A - Interview Questions

Criteria for determining partitioning of System Architecture for complex helicopter systems with respect to life cycle value.

Researcher: Leon Silva
Advisor: Dr. Eric Rebentisch

Research Questions:

- What methods are currently used in order to design system architecture for life cycle value?
- How can the architecture of a complex system be partitioned into modules for life cycle value?
- What are the key criteria to be used in establishing the modules that together make the system?
- What are the key metrics that can be used to quantify the effectiveness of that partitioning?

Objective:
To develop a methodology that can be utilized to systematically partition the architecture of an aircraft system with considerations for the full life cycle of the aircraft.

Interview Questions:
Please comment on your program experiences. (Some question may not apply)

1. When architecting a new system, what are the most important considerations? Are any of these particularly reflected in the partitioning of the system? *(Please include any and all aspects that are relevant to life cycle value; not necessarily only "engineering" issues.)*

2. When architecting a new system, how should the system be partitioned?

3. Do you use a particular structured process in defining system architectures? If so, please provide an example.

4. When comparing the architecture of the last three systems that you defined or worked on, what are the notable differences? Similarities?

5. Do other stakeholders play a role (directly or indirectly) in making partitioning decisions? If so, describe that role and give specific examples of its impact.

6. Can you identify some specific instances where the current partition of functions in the architecture has resulted in challenges relating to the supply chain, assembly, maintenance, logistics, product support, or upgradeability?

7. When reviewing an existing architecture, what are the characteristics that you look for in a "good" design? What questions might you ask about it? Can you give some examples of what you consider well-architected systems?

8. What methods (either formal or informal) are used in order to review the design of a particular system architecture? Who is involved in these reviews? When are they held with respect to the projects timeline?

9. From your perspective, what are the advantages of partitioning a system into modules, groupings and interfaces with respect to life cycle value? Do you feel that it plays a role in product strategy or platform strategy? If so, how?

10. What are the disadvantages of partitioning a system into modules?

11. Are there particular incentives for you to partition a system into modules?

12. Any other thoughts related to the discussed topics?

Also,

- Interviewee's Definition of System
- Interviewee's Experience Level
- Interviewee's Title