NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

QUANTIFYING CONSEQUENCES OF EXTERNALLY INDUCED FAILURES PROPAGATED THROUGH SYSTEMS DURING FUNCTIONAL SYSTEM DESIGN

by

Cameron A. Gunn

March 2022

Thesis Advisor:	Bryan M. O'Halloran
Second Reader:	Mark Stevens

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC, 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2022	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE QUANTIFYING CONSEQUENCES OF EXTERNALLY INDUCED FAILURES PROPAGATED THROUGH SYSTEMS DURING FUNCTIONAL SYSTEM DESIGN	**5. FUNDING NUMBERS**
6. AUTHOR(S) Cameron A. Gunn	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Assessment of failure propagation and potential within complex systems is a field open for continued exploration in the arena of systems engineering. Risk assessment and failure modeling processes such as PRA, FTA, and FMEA/FMECA are more widely understood and utilized in industry, yet are not designed to fully address and objectively quantify the impact on systems when exposed to intentionally malicious attacks, particularly in early design stages where changes to system architectures are best effected. Further, current methods do not identify and standardize attack modes that are likely to affect systems during their life cycle. This work first defines "attacks" and discusses their difference from "failures." The work then develops and discusses a hierarchical taxonomy of attack classes and mechanisms likely to affect a wide array of systems. Finally, it presents the Failure Path Length Method (FPLM) to quantify consequence on systems due to attacks on system functions by applying characteristics of those classified attacks to the functional architecture of a system. The author then implements the FPLM on a common EPS to verify applicability to realistic systems and objectively determine the consequence of an attack. The differences in consequence drive mitigating changes to the architecture of the EPS and validate the significant decision-making power provided to system designers by the proposed method during functional analysis and design.

14. SUBJECT TERMS failure, propagation, functional design, reliability, consequence, attack, taxonomy	**15. NUMBER OF PAGES** 105
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

QUANTIFYING CONSEQUENCES OF EXTERNALLY INDUCED FAILURES PROPAGATED THROUGH SYSTEMS DURING FUNCTIONAL SYSTEM DESIGN

Cameron A. Gunn
Lieutenant, United States Navy
BSE, Arizona State University, 2015

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2022

Approved by: Bryan M. O'Halloran
Advisor

Mark Stevens
Second Reader

Oleg A. Yakimenko
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Assessment of failure propagation and potential within complex systems is a field open for continued exploration in the arena of systems engineering. Risk assessment and failure modeling processes such as PRA, FTA, and FMEA/FMECA are more widely understood and utilized in industry, yet are not designed to fully address and objectively quantify the impact on systems when exposed to intentionally malicious attacks, particularly in early design stages where changes to system architectures are best effected. Further, current methods do not identify and standardize attack modes that are likely to affect systems during their life cycle. This work first defines "attacks" and discusses their difference from "failures." The work then develops and discusses a hierarchical taxonomy of attack classes and mechanisms likely to affect a wide array of systems. Finally, it presents the Failure Path Length Method (FPLM) to quantify consequence on systems due to attacks on system functions by applying characteristics of those classified attacks to the functional architecture of a system. The author then implements the FPLM on a common EPS to verify applicability to realistic systems and objectively determine the consequence of an attack. The differences in consequence drive mitigating changes to the architecture of the EPS and validate the significant decision-making power provided to system designers by the proposed method during functional analysis and design.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

C2	command and control
C5I	command, control, communications, computers, cyber, and intelligence
CBRN	chemical, biological, radiological, or nuclear
CFG	configuration flow graph
CPS	cyber-physical systems
DEW	directed energy weapons
DOD	Department of Defense
EFFBD	enhanced functional flow block diagrams
EM	electromagnetic
EMMI	energy, matter, material wealth, or information
EPS	electrical power system
EW	electronic warfare
FBD	functional block diagram
FFBD	functional flow block diagrams
FBED	functional basis for engineering design
FFPPM	functional failure propagation potential model
FFIP	functional-failure identification and propagation
FFL	function-failure-logic
FFRDM	functional failure rate design method

FMEA	failure mode and effects analysis
FMECA	failure mode, effects, and criticality analysis
FPL	failure path length
FPLM	failure path length method
FTA	fault tree analysis
HEL	high-energy laser
HPM	high-power microwave
ICAM	Integrated Computer-Aided Modeling
ICS	industrial control systems
IDEF0	Integrated Computer-Aided Modeling (ICAM) Definition for Functional Modeling
IE	initiating event
IR	infrared
MBSE	model-based systems engineering
MDT	maintenance downtime
MTBF	mean time between failures
MTBM	mean time between maintenance
MTTR	mean time to repair
NPRD-95	nonelectric parts reliability data - 1995
NPS	Naval Postgraduate School
PLC	programmable logic controllers
POE	projected operating environment

PRA	probabilistic risk assessment
PSYOPS	psychological operations
RAM	reliability, availability, maintainability
SCADA	supervisory control and data acquisition
SE	systems engineering
SME	subject matter experts
SoS	system of systems
SySML	Systems Modeling Language

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

This thesis explores the concept of malicious attacks as the cause for functional failures, and how the impacts or consequences of these individual attack mechanisms can be objectively quantified for use in improving system design. This work first develops a formal taxonomy to categorize the types of malicious attack mechanisms that can influence a variety of systems. The work also presents the Failure Path Length Method as a process to objectively quantify the negative impact a malicious attack may have on a system.

Reliability and failure propagation within systems are highly researched topics within systems engineering. System designers seek the ability to model the manner in which systems are impacted upon failure and categorize different failure modes and their frequencies with regard to the systems they impact. However, many existing failure analysis methods do not account for functional failures caused by malicious attacks generated *externally* on a system's boundary, nor are these potential attack occurrences categorized in a manner similar to system failure modes. Additionally, many previously developed failure propagation models or risk analysis tools are not fully applicable during the conceptual design phase of a system's life cycle, where changes to the system's functional or physical architecture may have more benign impacts on cost, schedule, or other developmental resources.

Adapting the concept of energy, matter, material wealth, and information being the main forms of exchanges occurring across functional boundaries within a system, the taxonomy developed in this thesis hierarchically categorizes attacks from the lowest level of fidelity to the highest using classes, types, and mechanisms to describe their occurrence within an operating environment. Additionally, the author found that introducing the concepts of attacking agents and behavior variables as amplifying descriptors of each attack mechanism adds value for the system designer by bringing more precision to describing the physical nature of the attack occurrence for modeling purposes.

The Failure Path Length Method utilizes elements of existing research on failure propagation and functional modeling while also adapting the developed taxonomy to drive determination of impact to system upon the occurrence of a malicious attack. By first developing a functional model of a system of interest, a designer may then initiate a variety of attacks from the presented taxonomy on a function or its interfaces within the system to discern unique failure propagation paths for each individual function. By comparing the number of failed functions that are created by simulating an attack on each individual function with the total number of functions within the system's functional architecture, the designer can then objectively quantify which function within the system presents the highest negative consequence provided it is attacked and fails.

When applying the Failure Path Length Method to the functional architecture of a realistic electrical power system, the method identified numerous functions within the original design that negatively impacted between 81% to 98% of the overall system, including generating a catastrophic failure by eliminating its ability to perform its primary function. Through consequence determination, the identification of these critical functions drove design improvements that, after implementation and re-examination, generated a maximum consequence value of 31% and minimized the critically vulnerable attack surface of the new electrical power system's architecture.

The taxonomy and Failure Path Length Method presented in this work hold substantial value to the fields of functional analysis, failure propagation, and system suitability and design. By creating a taxonomy structure for attack mechanisms, system designers are better able to standardize their descriptions of attacks and failures during system design by defining them with energy, matter, material wealth, or information designations. Through use of the Failure Path Length Method during functional analysis and analysis of alternatives, system designers are equipped with a new metric to objectively define the consequence of their functional design decisions and how those decisions may be actualized in the system's projected operating environment. The addition of the consequence metric within the systems engineering process provides an effective measure to correlate with analyses on reliability, availability, and maintainability in order to ensure improvement to suitability with each iteration of the method's application.

CHAPTER 1:
Introduction

This chapter provides the introduction to the thesis, including background, a statement of the problem, and the significance of the work conducted.

1.1 Problem Statement

Complex systems required to execute various functions to operate frequently experience failures in the performance of their required functions. These failure events impact the execution of other functions and ultimately the performance of the overarching system. While these failure events and their effects are often results of factors internal to a system, this thesis seeks to quantify the impacts on systems for which failures that occur are due to externally induced events of a malicious and intentional nature. These externally induced failures, or attacks, are defined by some attributes uncommon to internal failure modes, which drives the desire to develop an approach to account for the differences.

There exists a multitude of ways in which systems can be attacked, all inhibiting accomplishment of system function to varying degrees. When modeling system behavior as attacks are instigated, having an organized method of identifying potential attack types and their respective expected effects ultimately facilitates their implementation into failure propagation potential models. To assist in enhancing propagation models and drive system design improvements to mitigate impacts of different attacks, work in this thesis will also categorize attack modes.

Lastly, design changes implemented in later stages of a system's life cycle to accommodate for lack of consideration for safety, reliability, availability, maintainability (RAM), or survivability typically slows system development. These changes are costly with regard to time and financial resources once a system's components and subsystems are allocated to functions during design. For this reason, this work will focus on quantifying impacts of deliberate and malicious attacks conducted against systems during the early functional design stage, where knowledge of the system solution's physical architecture is low or non-existent.

1.2 Background

Analysis of system failure and the effects of those failures as they propagate throughout the system is a significant field of study within the systems engineering (SE) domain as designers consistently seek to improve system architectures in terms of RAM, and safety. There are various fault management, fault identification, risk management and assessment techniques and methods utilized in the SE world that are best fit for differing stages in the systems design process, however these methods do not sufficiently cover instances in which systems fail due to attacks. Specifically, many of these methods such as probabilistic risk assessment (PRA), failure mode, effects, and criticality analysis (FMECA), failure mode and effects analysis (FMEA), functional-failure identification and propagation (FFIP), and functional failure propagation potential model (FFPPM) are heavily probabilistic, which creates difficulty in implementation of most attack classes. These methods, their purposes, and shortcomings will be discussed further in Section 2.2.

1.3 Significance

The work within this thesis provides inherent value to common developers and users of highly complex systems. In the early 2010s, Stuxnet gained recognition as one of the most complex and highly capable cyber attacks to be executed and analyzed. The purpose of the cyber attack was to sabotage operations of power facilities in a likely adversarial nation by gaining control of industrial control systems (ICS) components and reprogramming programmable logic controllers (PLC) within the country's nuclear power grid architecture, causing them to operate well outside of normal operating limits [1]. In the case of Stuxnet, the cyber attack was likely based on political motives [2] as it targeted ICS in Iranian nuclear enrichment plants, in attempt to slow the advancement of the country's nuclear program and provide time for diplomacy and sanction imposition on the part of allied nations to take effect [3].

To effectively gain control of the correct ICS components and minimize functionality of the overall grid, the virus had to be installed onto secured systems at the target plant. The application of the work within this thesis would serve to help prevent an intrusion in this manner by analyzing the functional architecture of the ICS (point of entry/attack) within the nuclear power system and evaluating the impact of a failure or degradation in that system's nominal function. An objective value makes critical function determination immediate and simple. By understanding the interconnection of functions and their input/output flow exchanges within the nuclear plant, system designers may have been able to pinpoint functions that would require increased hardening or protection from specific types of attacks. Stuxnet being manually installed on target systems was representative of a sabotage attack committed by a human, but ultimately presented itself through the exertion of human energy flows on the target system, as discussed further in Section 3.3.3. The ICS on which it was installed is therefore a subsystem that performed a *critical* function and would have required further hardening.

The results of this work hold great significance to the SE community at large. By better learning to model and assess the effects of failures caused by intentional attacks on systems, designers are better able to develop and integrate effective mitigations and protections to enhance system RAM. In addition, the development of a preliminary taxonomy of attacks will provide a baseline for systems engineers to organize design mitigations based on attacking agents and mechanisms while simultaneously performing the aforementioned failure and risk management techniques.

This work will expand the existing knowledge base on early system design, specifically presenting the value-added benefit of providing additional knowledge of a function's consequence potential to better influence design decisions earlier in the life cycle. While failures in systems often appear to have the same effects when viewed through high-level analysis, understanding the implications of the underlying causes, or attack mechanisms, of these failures on the architecture of the system augments the overall conduct of the conceptual design phase. With the ability to quickly assess the consequence of an attack on a function during functional design as opposed to phases further along in a design project, opportunities are created to minimize costs in the form of financial resources and time necessary to better address design concerns and enhance system integrity.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Background and Related Work

This chapter provides further review of currently existing risk assessment and failure analysis methods, discusses the scope of some of these applicable methods, and where they lack with respect to the intended outcomes of this work. This chapter also discusses previous work defining “attacks” as they relate to their occurrences in different types of systems. Various types of attacks exist, with each having different potential effects on a given system, which has driven previous work in development of failure and attack taxonomies.

2.1 Failure and Attack Categorization

Attacks on systems are generated by threats within the operating environment of a system, and seek to cause damage or undesired effects to the target. Attacks are defined differently and to varying degrees, depending on the field of regard, such as when the targets are electro-mechanical systems, personnel organizations, biological systems, or cyber systems. In research to quantify risks and consequences observed in attacks on cyber-physical systems (CPS), O’Halloran, Papakonstantinou, and Van Bossuyt [4] define malicious attacks as actions that are “premeditated failures originated by humans” and are “initiated by an attacker’s intent to disrupt the system.” Systems and their functions are often targets of coordinated malicious attacks and while the failures resulting from these attacks are subjects of study, existing risk management methods and tools prove insufficient for various reasons.

Additionally, few works and efforts exist to provide comprehensive hierarchies and classifications to differentiate between attack types. O’Halloran, Stone, and Tumer [5] address the need for a process in which failure modes and mechanisms are organized and classified. Their methodology classifies failures using failure statements. A failure statement encompasses the five aspects of a failure: an “initial circumstance” that provides the environment or opportunity for failure to exist in the space, a “failure mechanism” that physically caused system failure, the “failure event” that specifies the component primarily affected by the mechanism, the “mode” that best depicts how the system is affected, and the “affected functionality” that dictates the change in functional system state [5]. This work supplements risk and fault assessment methods described more in depth in Section 2.2, however, the taxonomy developed lacks fidelity on classification for damage mechanisms, specifically as they relate to the intentional nature of malicious attacks on system functions. Because they are intentional, attacks on systems generated by external sources are generally less probabilistic, which requires expansion on the considerations included in defining the failure mechanism portion of the failure statement. Work in [6] and [7] present methods and descriptions to develop failure mode taxonomies for events relevant to mechanical and electrical systems, respectively, yet also stop short of discussing damage mechanisms when the cause is related to specific attacks.

In early works relating to malpractice on the Internet, work by Howard develops an initial taxonomy categorizing attacks within cyberspace. According to [8], categories classified in any satisfactory taxonomy should be mutually exclusive, exhaustive, unambiguous, repeatable, accepted, and useful to its specific field of regard. Howard's taxonomy, illustrated in Figure 2.1, follows a process-based approach that defines and connects groups of attackers to their varied objectives in an operational sequence consisting of tools used, accesses or "ways" of attack, and the end results of an attack on a cyber system.

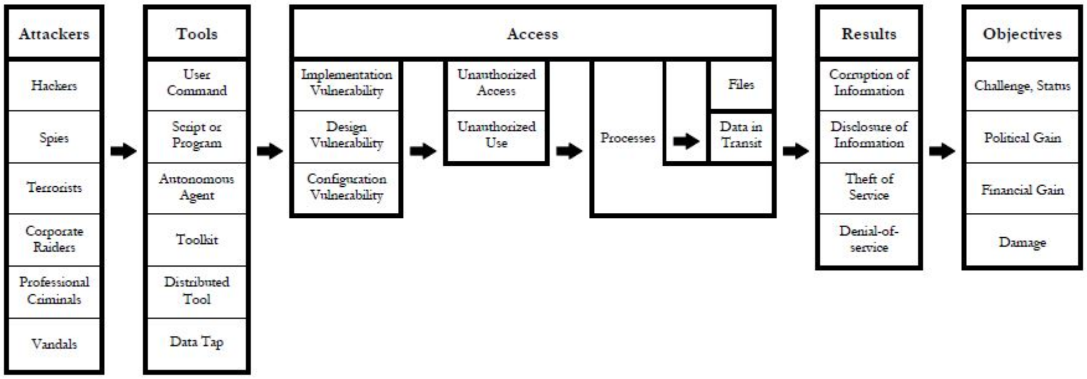
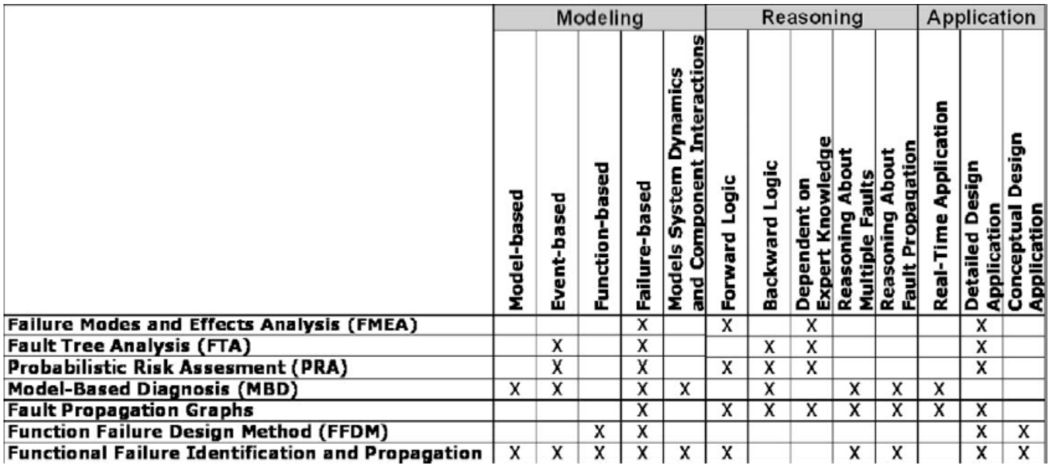


Figure 2.1. Computer and Network Attack Taxonomy in Operational Sequence. Source: [8].

While Howard's taxonomy is comprehensive in discussing the aspects of a cyber or network security attack, it fails to build an overarching view of attacks that could potentially impact a greater variety of systems. The author recognizes many systems exist outside the realm of computer network or cyber systems, which makes Howard's process-based approach valid but inhibits the work as an all-encompassing attack taxonomy.

2.2 Existing Failure Analysis Tools

Chapter 1 identified the presence of tools and methodologies used to illustrate and analyze system failure, failure propagation potential, and general risk management within system design. Each method or approach offers complimentary or overlapping techniques to examine and assess the effects of failures on system operation. Figure 2.2 illustrates some of the existing fault assessment methods and their various levels of applicability as compared to each other.



	Modeling					Reasoning					Application		
	Model-based	Event-based	Function-based	Failure-based	Models System Dynamics and Component Interactions	Forward Logic	Backward Logic	Dependent on Expert Knowledge	Reasoning About Multiple Faults	Reasoning About Fault Propagation	Real-Time Application	Detailed Design Application	Conceptual Design Application
Failure Modes and Effects Analysis (FMEA)				X		X		X				X	
Fault Tree Analysis (FTA)		X		X			X	X				X	
Probabilistic Risk Assesment (PRA)		X		X		X	X	X				X	
Model-Based Diagnosis (MBD)	X	X		X	X		X		X	X	X		
Fault Propagation Graphs				X		X	X	X	X	X	X	X	
Function Failure Design Method (FFDM)			X	X								X	X
Functional Failure Identification and Propagation	X	X	X	X	X	X			X	X		X	X

Figure 2.2. Comparison of Fault Assessment Methods. Source: [9].

The methodologies mentioned in Figure 2.2 are core to failure and reliability analysis within the SE community, however they each offer varying levels of applicability to the fields of attack categorization, attack assessment, and consequence quantification. Additionally, some tools and methods mentioned are not best fit for providing decision-making leverage at earlier phases in the system life cycle.

2.2.1 Functional Design Applicability

Existing fault and risk assessment methods often require knowledge of the physical architecture of a system in question, limiting their applicability during the functional design phase of the life cycle. According to the Department of Defense (DOD)'s guide on FMECA procedures [10], FMECA seeks to determine and classify potential modes of critical and catastrophic failure within a system for the purpose of improving design. FMECA, and by extension FMEA, identifies potential modes of failure within complex systems and assigns priority to the mode based on impact of failure (severity) to system operation, and the resulting analysis dictates priority in corrective action within system design. While explicit knowledge of the system's physical architecture is not required for a functional analysis, the nature of FMECA and FMEA procedures significantly strengthens the effectiveness of the analyses with high levels of fidelity in physical system architecture because the impact of failure on the system is only subjectively deduced based on knowledge of what each component (or function) accomplishes [10]. In addition to being less effective during functional design phases, FMECA and FMEA assume singular damage mechanisms for each individual failure mode. When systems are deliberately attacked it is typically the intent of an attacker to induce maximum damage, which insinuates a high likelihood of multiple simultaneous damage processes or uncoupled propagation instances causing negative terminal effects [4]. Under this assumption, neither FMECA nor FMEA are sufficient for the objective quantification of the effects an attack may have on a system during functional design.

PRA is another method that focuses specifically on risk assessment for the primary purpose of enhancing system safety and reliability [11]. PRA identifies and uses an initiating event (IE) to develop sequence diagrams outlining paths to terminal effects on the system. Figure 2.3 is an example of a sequence diagram for a hydrazine leak within a spacecraft system. The leak is the IE that drives logic statements to numerous pivotal events until ultimately, one of many potential end terminal effects are reached.

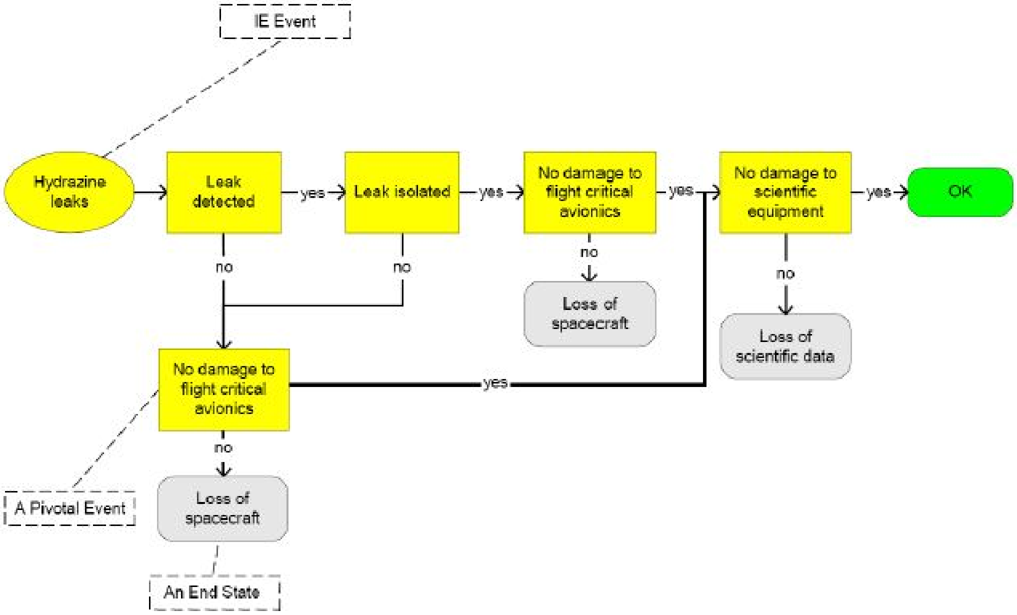


Figure 2.3. Event Sequence Diagram for a Hydrazine Leak in an Aircraft. Source: [12].

PRA, like FMECA and FMEA, is not solely based in functional system design. While functional capability of a system can be ascertained from end states annotated in a sequence diagram, IE focuses initially on an occurrence involving components and subsystems. Knowledge of a system's physical architecture is required in the top-down logic approach for PRA when tracing end states to identified IE [12]. This prevents PRA from being used as the singular methodology for assessment of functional failures or malicious attacks in early design. Additionally, PRA's effectiveness lies in its ability to enumerate outcomes of an IE, the likelihood of them occurring, and the risks associated with each outcome. While the outcome (and ultimately, the *qualitative* consequence) of each IE occurring can be assessed by designers upon determination, PRA does not preemptively identify the most critical system functions that may inherently have higher risks associated with attacking by computing the effect of the IE on a function.

2.2.2 Failure Propagation Potential and Measurement

When systems malfunction, failures in specific functions propagate throughout the system, sometimes in non-linear fashions, and can impact the ability of other functions to complete their tasks. Kurtoglu and Tumer [9] developed the FFIP framework as a means to assess functional failures in system design. FFIP "estimates potential faults and their propagation paths under critical event scenarios" [9]. Depicted in Figure 2.4, Kurtoglu and Tumer first built a functional block diagram (FBD) and an equivalent configuration flow graph (CFG) as a high level possible representation of the intended structure of a system. The FBD, or functional model, outlines the functional flow of the overall system, while the CFG allocates physical components to the functions involved and is more specific with the energy, matter, material wealth, or information (EMMI) exchanged between components [9]. FFIP uses these models concurrently to capture nominal system behavior and uses function-failure-logic (FFL) to abstract reason from dynamic system behavior for given failed functions [9].

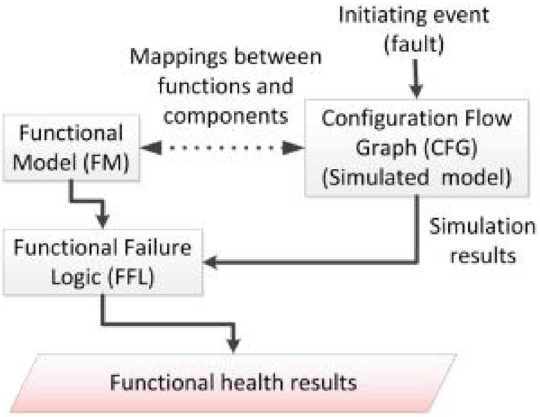


Figure 2.4. Process Followed Simulating Functional-Failure Identification and Propagation (FFIP) Methodology. Source: [13].

While FFIP addresses failure propagation in the modeled system, the mechanism *causing* failure, i.e. the behaviors and variables correlating to an externally induced attack, is not modeled or integrated any differently from a failure mode caused by normal system operation. A designer solely utilizing FFIP would not be able to numerically assess the implications of an attack on the system, nor would they be able to adequately recommend mitigations based on the *type* of attack inflicted on a function because it would appear in the same manner as a general failure. Ultimately, lack of a form of catalog of attack methods or IEs as well as a manner of representing them via model in FFIP is a key shortfall.

Further research conducted by O'Halloran, Papakonstantinou, Giammarco, and Van Bossuyt [14] produced an additional method to further FFIP objectives through the FFPPM process. Figure 2.5 outlines the steps to perform the FFPPM.

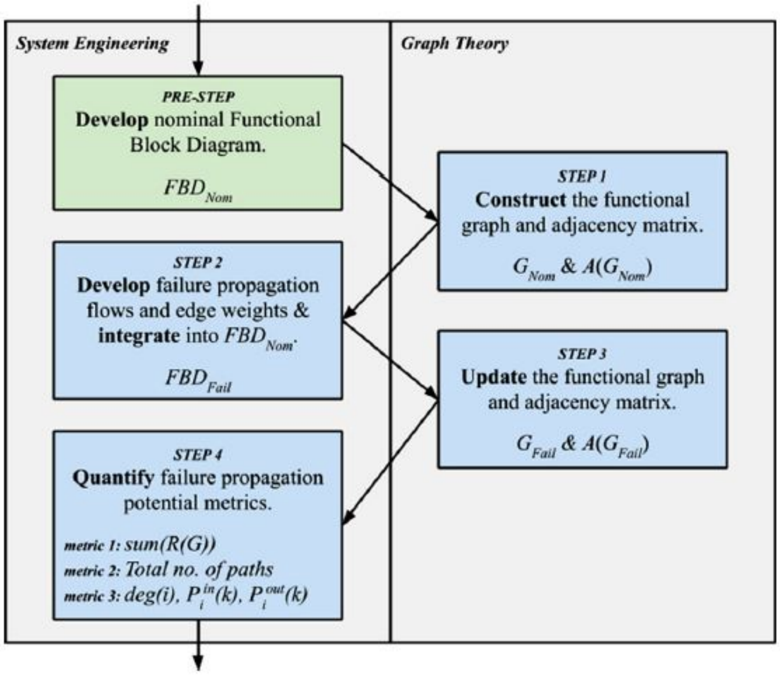

System Engineering
Graph Theory
PRE-STEP
Develop nominal Functional Block Diagram.
FBD_{Nom}
STEP 1
Construct the functional graph and adjacency matrix.
G_{Nom} & $A(G_{Nom})$
STEP 2
Develop failure propagation flows and edge weights & integrate into FBD_{Nom}.
FBD_{Fail}
STEP 3
Update the functional graph and adjacency matrix.
G_{Fail} & $A(G_{Fail})$
STEP 4
Quantify failure propagation potential metrics.
metric 1: $sum(R(G))$
metric 2: Total no. of paths
metric 3: $deg(i)$, $P_i^{in}(k)$, $P_i^{out}(k)$

Figure 2.5. Steps to Conduct Functional Failure Propagation Potential Model (FFPPM) Method. Source: [14].

FFPPM employs graph theory to generate matrices of functional system connectivity based on an abstracted FBD of a system in nominal operation. The flow paths, or "edges," connect functions, or "nodes," in the FBD. The edges are further defined by a myriad of behavior variable terms standardized by the use of the functional basis for engineering design (FBED), which is the result of efforts to develop an evolved functional basis to support enhanced functional modeling of system behavior [15]. In [14], the original concept of the FBED flow hierarchy from [15] is expanded to allocate behavior variable designators to flow definitions, adding the required fidelity to support its use in failure propagation modeling.

FFPPM proceeds to generate a FBD for a system under failure conditions. FFPPM quantifies the degree of connectedness for the system in order to illustrate the propagation of the failure through functions as well as the summation of function reachability that directly correlates to failure propagation potential within the system.

While the failure mode is identified for the purpose of the methodology, FFPPM does not address underlying causes of the failure when propagating through the system's functions to develop the FBD for the system in a failed state. Like many previously discussed methods, FFPPM is heavily probabilistic and usable only with the implementation of failure rates for functions or components. At early design phases, failure rates of functions can be inferred from historical knowledge bases for components commonly allocated to perform said functions since the physical system architecture is not ordinarily known. The functional failure rate design method (FFRDM) process presented in [16] supports this FFPPM requirement by linking nominal knowledge of functional failure rates to generally common failure modes for functions and components using the nonelectric parts reliability data - 1995 (NPRD-95) [17] and FMD-97 [18] as historical knowledge bases (and by extension, updated failure rate data in [19], [20], and [21]), however these historical failure rates only apply to *designed* reliability data previously collected and are no indication of the probability of a specific type of attack occurring as the desired failure mechanism [20]. Because attacks in a system's operating environment cannot be reliably predicted in similar manners to failure events in functions or components (based on failures per hours of operation), neither FFPPM nor FFRDM completely allow for attack consequence quantification.

2.2.3 Malicious Attack Integration

As CPS become more complex and integrated with accomplishment of everyday functions, vulnerabilities become abundant. This allows for attacks on these systems, where failures are not entirely due to internal factors within the system. In [4], subject matter experts (SME) conducted research to assess the propagation of failures due to malicious attacks specifically on CPS. In developing the methodology to assess a risk of malicious attacks in the life cycle of a CPS, the research evaluated the critical failures of CPS when exposed to malicious attacks, information which inherently adds value in being able to influence early design and decision making processes. However the method is primarily from an "attacker-centric" viewpoint [4]. The research develops a method to identify attack methods from human attackers, but does not explicitly discuss different attack classes, modes, and how those may impact consequence on a function. The limited application of FFPPM principles in [4] was also heavily driven by the physical architecture of the identified system, which detracts from its applicability in functional design.

2.3 Chapter Summary

This chapter introduced and discussed the concept of an attack with regard to system operability. Attacks create undesired damage in the form of failures within systems in more unpredictable, yet often highly catastrophic ways. Many of the existing fault analysis and risk management tools in industry and research developed within the SE community such as PRA, FMEA, FMECA, FFPPM, and FFIP offer ways to predict the likelihood and evaluate the risks of different failures occurring within systems, however none take the perspective of an intentional, well-coordinated attack from outside of a system's boundaries executed on one or multiple functions within it. By first generating a taxonomy discussing a broad range of attacks and developing a methodology to evaluate the consequence and impact to system operability resulting from these attacks within behavioral models, the author seeks to originate a process applicable to functional models utilizing principles from previous work on the FFIP and FFPPM methods. With the methodology introduced herein, attacks against system functions will be modeled, and benefits of quantifying resultant failures during early functional and conceptual design phases will be demonstrated through enhancements to different aspects of system suitability.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Attack Taxonomy

This chapter defines and further expands on the concepts of system attacks. This chapter categorizes levels of attacks hierarchically and classifies IE and externally generated mechanisms that induce system damage and failures. This chapter will first define an "attack" by explicitly outlining requirements and characteristics that ordinarily differ a malicious attack from a general failure event. Secondly, the author will discuss the generation of the attack taxonomy by first presenting the requirements for an effective attack taxonomy in general terms. This chapter will then describe the hierarchical classifications that will be applicable the final taxonomy, and finally, present a full attack taxonomy that works to identify a wide range of attacks that could impact various systems. The contents of the taxonomy will be vital in the implementation of the methodology to model failure propagation consequence due to attack methods, and will focuses on DOD-centric systems for analysis.

3.1 Attack Definition

From the perspective of defense and combat systems relevant to the DOD, systems of various complexity with a myriad of functions are relied upon in to achieve objectives vital to national defense and security. However, as is common in hostile or abnormal operating environments, systems become damaged, ineffective, and inevitably fail due to occurrences not directly related to the system's internal structure and overall architecture. This is the primary factor defining an "attack" with respect to the following methodology: the occurrence of an event intended to cause abnormal or sub-optimal performance in a system that is initiated by forces *external* to the system's infrastructure and general operation. To reiterate and clarify, an attack is specifically defined as an event that is:

1. Due to man-made actions and intentions, i.e. systems are not made to fail from the effects of weather, natural disasters, or other uncontrollable forces.
2. Externally induced, i.e. system failure is not a result of nominal system operation, but originates from some foreign force (or lack thereof) acting upon the system and its functional, physical, or behavioral interfaces.

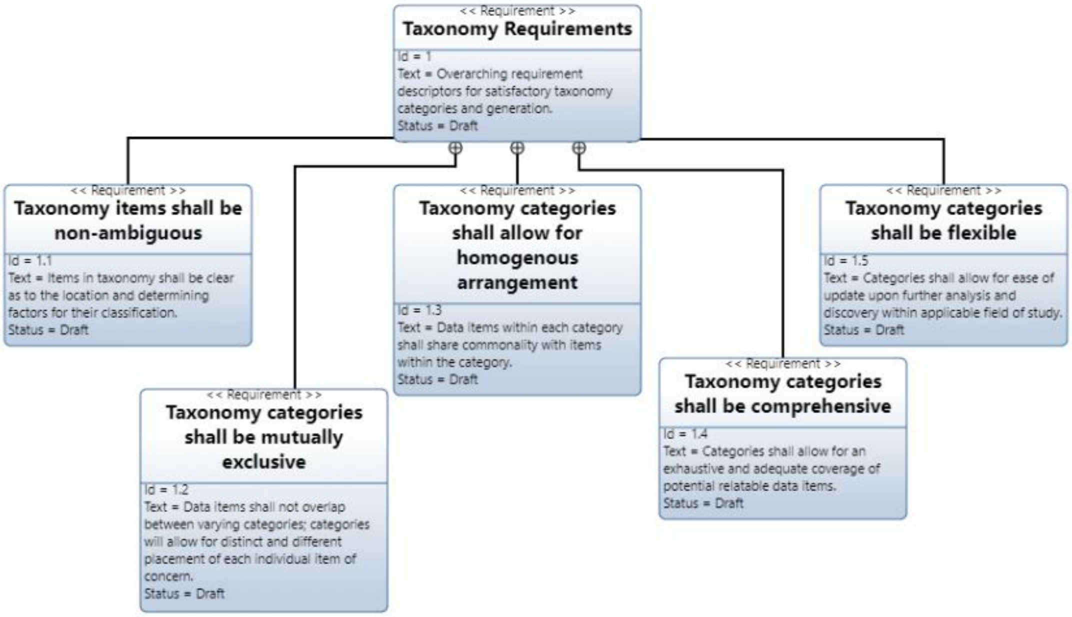
3. Made with the intention to degrade or completely render useless a system or any of its normally performed functions.

3.2 Taxonomy Generation

Classification tools such as object taxonomies often differ in structure based on the objects of inquiry, and therefore undergo varying processes in developing useful hierarchies. Despite this, taxonomies for any object of inquiry have common features and requirements that are considered during generation.

3.2.1 Requirements Decomposition

According to [22], "taxonomy" typically describes the manner in which living things are classified within the realm of biology, however the principles and general science of taxonomy can still apply to the classification of other topics, objects, and like concepts. Based on the general science of taxonomy, a good taxonomy "takes into account the importance of separating elements of a group into subgroups that are *mutually exclusive* and *unambiguous*, and taken together, include *all* possibilities" [23]. Abiding by information in [8], [23], and knowledge of classification techniques, the author uses Figure 3.1 to illustrate the derived high-level requirements for a satisfactory categorization tool:



<< Requirement >>
Taxonomy Requirements
Id = 1
Text = Overarching requirement descriptors for satisfactory taxonomy categories and generation.
Status = Draft
<< Requirement >>
Taxonomy items shall be non-ambiguous
Id = 1.1
Text = Items in taxonomy shall be clear as to the location and determining factors for their classification.
Status = Draft
<< Requirement >>
Taxonomy categories shall allow for homogenous arrangement
Id = 1.3
Text = Data items within each category shall share commonality with items within the category.
Status = Draft
<< Requirement >>
Taxonomy categories shall be flexible
Id = 1.5
Text = Categories shall allow for ease of update upon further analysis and discovery within applicable field of study.
Status = Draft
<< Requirement >>
Taxonomy categories shall be mutually exclusive
Id = 1.2
Text = Data items shall not overlap between varying categories; categories will allow for distinct and different placement of each individual item of concern.
Status = Draft
<< Requirement >>
Taxonomy categories shall be comprehensive
Id = 1.4
Text = Categories shall allow for an exhaustive and adequate coverage of potential relatable data items.
Status = Draft

Figure 3.1. High-Level Requirements of an Adequate Taxonomy and Categorization Tool.

The categories and child-level data entries at each level within the designed taxonomy presented in this work follows the descriptions for each decomposed sub-requirement as outlined in Figure 3.1 and further explained herein:

1. Non-ambiguous: each level of fidelity in the taxonomy is well-defined, preventing misconstruction of the presented information.
2. Mutually exclusive: each category or instance of data within the taxonomy will be assigned to one, and only one, higher class. Data instances do not overlap between classes or lower levels.
3. Commonality/Homogeneity: items within each category must have attributes or properties with which they share between each other.
4. Comprehensive: categorized data items at the lowest level should be exhaustive and cover the widest possible range of observable instances.
5. Flexibility: categories should allow for continuous update and improvement upon taxonomy structure should later analysis yield new applicable information.

3.2.2 Hierarchy Development

To assist in fulfilling requirements 1.1 and 1.2 of non-ambiguity and mutual exclusivity, the taxonomy is designed to employ a hierarchical format with varying levels of decomposition. As opposed to typologies, which organize data based on a space generated by nominally measuring each concept's adherence to multiple qualitative "dimensions" or axes, the hierarchical structure of a taxonomy will arrange attacks such that their similarities are "clustered" and seen within each individual category, yet each level of decomposition provides distinct variance and differentiation across categories [24].

First, classification levels must be clearly identified and delineated to enforce the vertical decomposition of differing attacks and their characteristics. This taxonomy offers three distinct levels of decomposition: "Class," "Type," and "Mechanism" as seen in Figure 3.2.

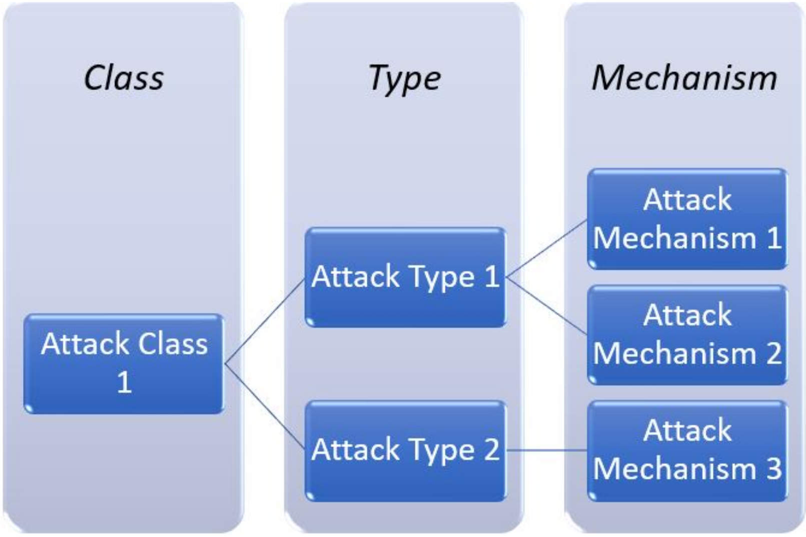


Figure 3.2. Example Hierarchy Delineating Order of Precedence and Progression from Attack "Class" to Attack "Mechanism."

Classes provide the first and highest level differentiation with regard to an attack's associated characteristics. As previously discussed, attacks can be categorized by their attacking agents that influence the function's nominal performance based on the characteristics of its flow type. Attacks cause system degradation by impacting EMMI exchange across individual functional boundaries, but the IEs that constitute attacks also have properties based on elements or instances of either energy, material/matter, or information. Classes first differentiate attacks based on these main properties of the IE and are therefore divided into three classes: Signal (Information), Energy, or Physical (Matter).

Types are the next level of differentiation between attacks, and they seek to describe lower attack modes or mechanisms based on common characteristics. To support taxonomy requirement of Commonality/Homogeneity as defined in Section 3.2.1, a type groups mechanisms by describing the common effect or method by which each child-level mechanism within said category will influence a system's functions or overall performance. For example, a "Destructive" type attack within the Energy class describes the occurrence of an attack by a form of energy that will ultimately result in a loss of a system's function through its complete destruction, whereas one of the mechanisms within the "Disruptive" type will ultimately result in a loss of system function only through inhibiting or prohibiting the flow of EMMI required for affected functions to perform their mission within the system, not necessarily the destruction of the function's allocated component.

Mechanisms are the specific event or process that created the intended effects of a malicious attack on a system. Similar to "damage processes" as defined in [25], mechanisms describe the externally induced IE or occurrence that causes the failed function within a system and outlines the interaction between the physical entity causing failure (or damage mechanism) and the terminal effect on the system function (i.e. failed function and loss of functionality) [25]. An example of an attack mechanism would be Kinetic; a missile (damage mechanism) *kinetically* impacting the superstructure of a surface vessel and destroying a surface search radar antenna creates a loss of "search," "detect," and "track" functions for the surface vessel system. When describing attacks, mechanisms are at a level of sufficient fidelity to directly describe an occurrence of the attack, such as a "*denial jamming* attack", "*biological* attack", or "*fabrication* attack."

3.2.3 Amplification of Attack Descriptions

Attacks on systems occur through a variety of means. Generally speaking, malicious attacks on systems have human origins, and work in [4] describes attacks from the perspective of the human initiator. To better serve the first purpose of this work in development of a hierarchical means of classifying attack types, the author seeks to build a system-centric perspective of attacks through the use of an “attacking agent.” The attacking agent is the way in which an attack is perceived by the system upon its occurrence. Very similar to the way Dr. Robert Ball [25] describes a damage mechanism as “the physical entity causing damage,” the author uses the term “attacking agent” as the primary descriptor of any physical force, quality, or occurrence that causes the degradation in system function or performance by affecting either the exchange of EMMI between system functions or components, or the function or component itself. Attacking agents are ultimately flow descriptors and use terminology from the expanded FBED in [14] to describe the nature of the degrading EMMI perceived by the attacked system. When utilizing the method presented in this work as seen in Section 4.1.3, the attacking agents in the following taxonomy serve as suggested and likely descriptions of their respective attack mechanisms as represented in the framework of the functional flow block diagrams (FFBD) construct used to model a system’s functional architecture.

Behavior variables are a concept introduced in [14] as an expansion of the FBED presented in [15]. Behavior variables serve the purpose of providing further fidelity with respect to system behavior and dynamics to the modeler within the framework of FFBDs [14]. By describing the physical nature of flow path edges within a FFBD, behavior variables assist in better understanding the nature of functional and system interfaces and allows for the process by which the author traces functional failures throughout a system’s architecture in Chapter 4.

The addition of both behavior variables and attacking agents to each attack mechanism within the attack taxonomy hierarchy as described in Section 3.2.2 amplifies the understanding of the physical nature of the EMMI being exchanged across functional boundaries within a system. Additionally, it provides modelers using the methodology presented within this work a clearer visual illustration of attack injections and flows within a FFBD, ultimately providing more information about an attack with which to develop clearer system solutions.

3.3 Taxonomy

Table A.1 in Appendix A provides a full breakdown of attack classes and types, and provides various examples of correlating attacking agents to their mechanisms. This section further discusses in detail the mechanisms and the classes to which they correlate.

3.3.1 Signal Class Attacks

Signal attacks are those in which system functions are negatively impacted and performance is degraded, however the functional or system failure caused due to the attacking agent may not be physically harmful in nature to the structural integrity of the component ultimately performing the function. Signal attacks primarily influence cyber and cyber-physical systems by altering *information* flow between functions to cause failure or degradation of a function's intended result. They are typically synonymous with the conduct of "cyber attacks" when discussing cyber-security and the integrity of cyber-physical or software-based systems and functions. The mechanisms within the Signal class are divided into Passive and Active Cyber types, further explained in each of the following sections.

Passive Cyber

In the realm of network security, cyber-physical systems and their functions are attacked at their interfaces where the flow of EMMI exchanged across functional boundaries is manipulated by unauthorized parties or sources. Discussed by Stallings [26], cyber attacks occur via four processes; Interruption, Interception, Modification, or Fabrication. The author of this work defines Passive Cyber attacks as those processes where EMMI is negatively influenced simply by the *presence* of a harmful attacking agent within the system's architecture, which occurs primarily in the Interruption or Interception mechanism methods described herein:

1. Interruption: signal information to be passed between multiple functions within the architecture of a system is interrupted in transit across interfaces, and functional failure is caused by the prevention of necessary signal information being output from one function being reached as in input to one or more follow-on functions [26].
2. Interception: signal information to be passed between multiple functions within the architecture of a system is passively gained and redirected by an unauthorized source

or attacker for gain or to initiate other actions elsewhere within the system to degrade or cause other functions to fail [26]. Instances where signal information is simply intercepted (e.g. stolen) and utilized for the gain of the attacker elsewhere outside the system boundary but the system's performance is not impacted is outside the scope of this work. The author qualifies an Interception attack as being one where other functions within the applicable system boundary are specifically degraded due to the instance of intercepted information.

Active Cyber

Active Cyber attacks are those processes which require not just the presence of a harmful attacking agent within the system's architecture, but that the attacking agent must perform its own *actions* on the exchanged EMMI to foster functional failure or degradation. The author describes Modification and Fabrication attack mechanisms as such:

1. Modification: signal information to be passed between multiple functions within the architecture of a system is output from a function, and intentionally *altered* in an interface while en route to follow on functions where that information would be used by follow-on functions as an input [26]. This modification of information leads to loss of integrity in cyber systems and is the primary cause of functional and/or system-level failure.
2. Fabrication: signal information to be passed between multiple functions within the architecture of a system is *falsely generated* and injected into functional nodes requiring input information, without the influence of proper prior functions within the system's organization [26]. The fabrication of false information at functional interfaces and injection as an input to subsequent functions in a system leads to loss of integrity in a cyber system and is the primary cause of functional and/or system-level failure.

Figure 3.3 visually depicts the four mechanisms within this class and how each of these attacks occur with respect to two circular nodes representing separate systems, subsystems, or functions. Known computer viruses, trojans, bugs, or malware utilize code and impact nominal functionality of computer systems in one or more of these four ways.

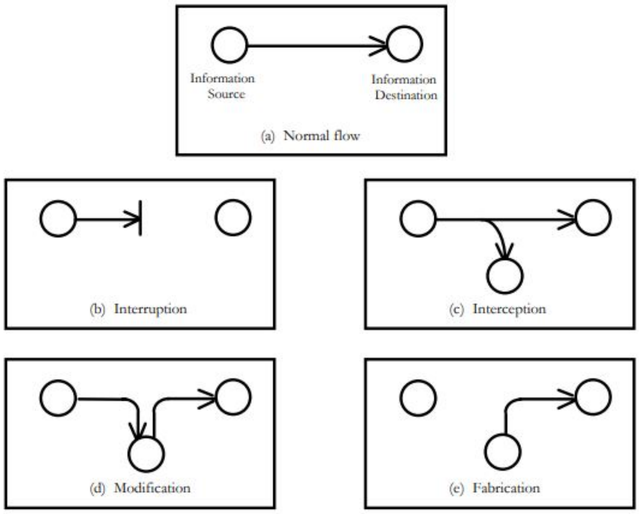


Figure 3.3. Illustration of Cyber Security Attack Methods. Source: [26].

Signal Taxonomy

Table 3.1 illustrates the portion of the final Attack Taxonomy hierarchy pertaining specifically to the Signal class, including the attacking agents and behavior variables correlating to the mechanisms within the class.

Table 3.1. Hierarchical Taxonomy of the Signal Class of Attacks. Signal Class Attacks Represent the "Information" Aspect of Energy, Matter, Material Wealth, or Information (EMMI) and Intentionally Influence Information Passed between Functional Nodes within Cyber Systems to Cause Functional Failure.

Attack Class	Attack Type	Attack Mechanism	Attacking Agent (Flow)	Behavior Variables
Signal	Passive	Interruption	-Signal •Status	•Time[Ti] •Location[L]
		Interception	-Signal •Status •Control	•Time[Ti] •Location[L] •Amplitude[Am]
	Active	Modification	-Signal •Status •Control	•Time[Ti] •Location[L] •Amplitude[Am]
		Fabrication	-Signal •Status •Control	•Time[Ti] •Location[L] •Amplitude[Am]

The primary attacking agents for the mechanisms within the Signal class are electrical signals injected into cyber or software based systems in the form of code that generates directional control signals passed between functions. Systems view Signal class attacks as **status** or **control** information signals creating functional failure by altering or inhibiting proper flow of information that ultimately impact the inputs and outputs of functions requiring that information. In command and control (C2) and supervisory control and data acquisition (SCADA) systems where large organizations of cyber-physical systems, their operations, or troubleshooting and recovery in the case of equipment casualties may be automatically or manually managed by human-machine interfaces, integrity and availability of information generated by functions are of utmost importance to make effective decisions and take corrective action based on the information presented. Altered or inhibited status and control information signals created by Interruption, Interception, Modification, or Fabrication attacks influence and change the behavior variables exchanged within operating systems such as time of information passage, location of necessary information occurring, or amplitude measurement information (i.e. voltage or current levels in different areas of a power plant) to facilitate failures or degradation at a system level.

3.3.2 Energy Class Attacks

Energy attacks are those that primarily utilize various forms of natural and physical energy to destroy or cause damage to a system or its individual functions. While energy exists in various forms such as gravitational, sonic, electrical, and chemical energy, this attack class focuses on attacks most likely to be used in a malicious manner against common cyber-physical and national defense systems, namely electrical and electromagnetic (EM) energy. Despite the Energy class attacks discussed in this section primarily utilizing EM energy to cause functional and component failure as the primary attacking agents, the types differ based on power density of the attacking agent and the target system functions. As such, EM energy attacks serve to create functional failure in manners ranging from simply disrupting nominal operation to physically destroying and eliminating functional capability altogether.

Disruptive Energy Attacks

Disruptive attacks disable system functions by disturbing the flow of EMMI exchanged between target functions. Disruptive attacks, similar to those listed in 3.3.1 for software-based information technology systems, are most effective at causing failures by altering EMMI at functional interfaces instead of causing physical damage directly to the attacked function and its corresponding component. The most common disruptive attacks utilizing EM energy result from different types of jamming, described as the primary attack mechanisms within the type:

1. Denial Jamming: attacks causing functional failure by introducing "sufficient noise (i.e., EM energy) into a sensor system such that desired signals cannot be reliably detected or analyzed" [27]. Denial jamming attack mechanisms *deny* systems' use of the EM spectrum to perform various functions and initiates failures that propagate through a system.
2. Deception jamming attacks cause functional failure by introducing "signals into a sensor system that the sensor system will mistake for the desired signals and

initiate incorrect actions" [27]. Functional failure created by a single instance of deception jamming propagate through a system as the results and outputs of the aforementioned "incorrect actions" are utilized as inputs for other functions within a system architecture.

Destructive Energy Attacks

Unlike Disruptive type attacks, Destructive attacks utilize EM energy to directly eliminate function capability, likely through structural failure of a function's correlating component. The high-energy laser (HEL) and high-power microwave (HPM) mechanisms within this type are versions of directed energy weapons (DEW) that seek to focus generated and radiated EM energy at targeted system functions to destroy overall target system functionality and capability.

1. HEL: attacks system functions via beams of optical-wavelength EM energy [28] at narrow beamwidths. A HEL typically utilizes EM energy generated by chemical elements as fuel sources or electrical energy inputs.
2. HPM: attacks and interferes with EMMI input into system functions via emission of microwave-wavelength EM energy [28] at wide angles and high power outputs.

While both mechanisms within the Destructive type of attacks utilize EM energy to attack system functions and generally are expressed through similar behavior variables and attacking agents, there are slight differences. The biggest difference in the two mechanisms may be the use case due to their applicability; HEL rely on narrow, focused beams of energy whereas HPM weaponry used in attacks are often area weapons by nature of microwave-wavelength energy. In a general sense, this equates to HEL being used for point attacks and HPM being used for wide area attacks, potentially covering more target components (and therefore, more target functions or systems) simultaneously.

Energy Taxonomy

Table 3.2 outlines the mechanisms within the Energy class, as well as their correlating attacking agents and behavior variables.

Table 3.2. Hierarchical Taxonomy of the Energy Class of Attacks. Energy Class Attacks Cause Functional Failure through Intentional Introduction of Forms of Energy into a System to Prohibit Optimal Performance of Functions or Alter the State of EMMI Exchanged between Them.

Attack Class	Attack Type	Attack Mechanism	Attacking Agent (Flow)	Behavior Variables
Energy	Disruptive	Denial Jamming	-Energy •Electromagnetic -Signal •Status	•Intensity[I] •Location[L] •Time[Ti] •Amplitude[Am]
		Deception Jamming	-Energy •Electromagnetic -Signal •Status	•Intensity[I] •Location[L] •Time[Ti] •Amplitude[Am]
	Destructive	High-Energy Laser	-Energy •Electromagnetic •Thermal	•Chemical Elements [Ce] •Intensity[I] •Dimension[D] •Heat[H] •Particle Velocity[Pv] •Electromotive Force[Ef] •Current[C]
		High-Power Microwave	-Energy •Electromagnetic •Thermal	•Intensity[I] •Dimension[D] •Heat[H] •Particle Velocity[Pv] •Electromotive Force[Ef] •Current[C]

Attacks in the Energy class primarily adopt EM energy flow descriptions with respect to their correlating attacking agent model representations. However depending on the specific attack occurrence, other forms of energy (or class characterizations in the case of the Disruptive attacks) may be present. When conducting jamming attacks in some form of an electronic warfare (EW) environment, a target system's functions will perceive the attack mechanism as an electromagnetic signal, but that signal energy will likely carry misguiding information about the *status* of the attacking system that will be required for a successful attack (i.e., incorrect size, location, or even presence of the attacking system). The intensity and amplitude of the EM energy radiation, length of time it is applied, and the location or direction of emission are behavior variables important to the modeler as description of the EM energy used against a target function or system in the jamming attack occurrences.

In Destructive attacks, thermal energy created at the target component (function) as a result of the incident EM energy used also helps achieve the destructive effect on the target. Behavior variables for both mechanisms within the Destructive type are similar, however due to the use of chemicals being used as fuels in many HELs, chemical elements are included as a driving behavior variable to help describe the nature of potential HEL attacks when modeled.

3.3.3 Physical Attacks

Attack mechanisms in the Physical class inhibit system function by physically damaging or destroying components within the system's architecture. However unlike most Energy attacks, the attacking agents of Physical attacks also involve the presence of a physical material or mass coming into direct contact with system components in order to degrade or eliminate functionality. In many instances of the Physical class, target functions most influenced by these attacks are ultimately represented by human actors within the target system, e.g., ground warfare scenarios involving multiple troop organizations and supporting equipment systems. This section of the taxonomy describes Physical attacks from the perspective of weapon systems most likely to be used to initiate attacks within hostile scenarios such as the aforementioned example, which lends to attack mechanisms being divided into Conventional and Unconventional types.

Conventional Type Attacks

Mechanisms in the Conventional type follow the notion that failures of target functions are induced by the use of traditional and *conventional* offensive systems. Offensive conventional systems degrade target system functionality through destruction of the target function via Kinetic impact with some form of material (e.g., impact of a missile against the superstructure of a ship, or impact of bullets to target personnel) or Concussive damage from traditional bombs or explosives.

1. Concussion: attacks that degrade or eliminate target functionality or capability through blast, overpressure, or other forces resulting from the forced agitation of the target's surrounding environment.
2. Kinetic: attacks on functions or systems that eliminate functionality or capability via direct contact with a given material, where the transfer of kinetic energy from the impacting material to the impacted function creates failure.

Unconventional Type Attacks

Unconventional type attacks are those that also require the presence of, and contact with, a physical material or mass to cause damage to system functions. However, unlike Conventional attacks where damage is kinetic or concussive, Unconventional attacks involve the use of chemical, biological, radiological, or nuclear (CBRN) material that may cause additional significant repercussions to a system's functions. Unconventional attacks are often results of the use of specialized weapons or combat systems, and while use cases vary, targets of Unconventional attacks are frequently humans executing the desired functions within an adversary system operating in warfighting environments.

1. Chemical: attacks on systems and functions where capability is severely degraded or eliminated through the introduction of chemical toxins [29] and chemically reactive materials to the function's operating environment.
2. Biological: attacks of systems and functions where capability is severely degraded or eliminated primarily through the introduction of biological pathogens [30], organic materials, or otherwise biologically harmful material to the function's operating environment.

3. Radiological: attacks on systems where capability is severely degraded or functional failure is created primarily through the introduction of ionizing radiation [31] or nuclear fallout [32] to the function's operating environment.

Physical Taxonomy

Tables 3.3 and 3.4 illustrate the Conventional and Unconventional type attack mechanisms within the Physical class of attacks, respectively, along with their accompanying behavior variables and attacking agents.

Table 3.3. Hierarchical Taxonomy of Conventional Type, Physical/Material-Based Attacks. Physical Attacks Represent the "Material" Aspect of EMMI and Create Functional Failure through the Introduction of Some Form of Traditional or Explosive Material or Mass to a System's Functional and Physical Architecture.

Attack Class	**Attack Type**	**Attack Mechanism**	**Attacking Agent (Flow)**	**Behavior Variables**
Physical	Conventional	Concussion	-Material •Gas •Solid -Energy •Acoustic •Mechanical •Pneumatic •Thermal	•Pressure[P] •Force[F] •Heat[H] •Linear velocity[Lv]
		Kinetic	-Material •Solid •Liquid •Gas •Mixture -Energy •Mechanical •Human	•Volume[V] •Location[L] •Force[F] •Pressure[P] •Dimension[D] •Linear Velocity[Lv]

Table 3.4. Hierarchical Taxonomy of Unconventional Type, Physical/Material-Based Attacks. Physical Attacks Represent the "Material" Aspect of EMMI and Create Functional Failure through the Introduction of Some Form of Chemical, Biological, Radiological, or Nuclear (CBRN) Material or Mass to a System's Functional and Physical Architecture.

Attack Class	Attack Type	Attack Mechanism	Attacking Agent (Flow)	Behavior Variables
Physical	Unconventional	Chemical	-Material •Liquid •Gas -Energy •Chemical •Thermal	•Reaction rate[Rr] •Intensity[I] •Temperature[Te] •Heat rate[Hr] •Chemical elements[Ce]
		Biological	-Material •Liquid •Gas -Energy •Chemical •Biological	•Reaction rate[Rr] •Intensity[I] •Chemical elements[Ce]
		Radiological	-Material •Liquid •Gas •Mixture •Solid -Energy •Radioactive •Chemical	•Reaction rate[Rr] •Intensity[I] •Chemical elements[Ce]

Attacking agents within the Physical class are primarily represented by “Material” flows as the nature of a Physical attack requires the introduction of some form of liquid, solid, or gaseous material to cause functional failure. In addition, many Physical class attacks may also be accompanied by additional Energy-based flows as secondary based on the specific attack occurrence (e.g., the mechanical energy transferred from the solid material used in a Kinetic attack into the component performing the targeted function upon impact).

In Conventional type attacks, solid materials drive the inclusion of behavior variables such as volume, force, linear velocity, location, and pressure, as knowledge of each of these descriptors with respect to the object(s) used for attack creates a better understanding of the severity of an attack occurrence. In a similar manner, Unconventional type attacks are often best described by the specific type and amount of chemical element, biological hazard, or radiological material used in the attack occurrence, driving the inclusion of Chemical elements and Intensity as major behavior variables of concern when discussing these occurrences. CBRN attacks hold potential to induce failure in physical system structures depending on the material composition of the system in question, such as chemical reactions occurring between acidic agents and metallic structures or multiple gases within a system. This drives reaction rate as an inclusive behavior variable to some Unconventional type attacks as well.

3.4 Chapter Summary

This chapter presented a method for developing a hierarchical taxonomy for a modeler to help better understand the types and natures of attacks that are likely to impact various systems and their functions. Taxonomies are made to organize information in a non-ambiguous, comprehensive, and flexible manner to ensure the presented information is usable in its original presentation but may be amended further with new information. The taxonomy developed here divides mechanisms into common or homogenous groupings of classes and types based on their common behavior variables and attacking agent representations, yet with a level of mutual exclusivity in each mechanism’s description that allows a system designer to differentiate possible occurrences and IEs in different operating environments. The method presented in Chapter 4 illustrates how information from the taxonomy developed is applied to both injected attack occurrences as well as amplification of functional models required for attack and failure modeling within systems.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4: Failure Path Length Method

This chapter describes the failure path length method (FPLM), or the process followed to objectively quantify the consequence of a malicious attack on any given function within a system. The FPLM presented in this chapter adapts concepts discussed in [14] with respect to reachability and failure propagation tracing methods, however the consequence of an attack is assessed using a different metric that represents the total impact on a system. Figure 4.1 generally outlines the process followed to quantify system impact due to attacks on functions.

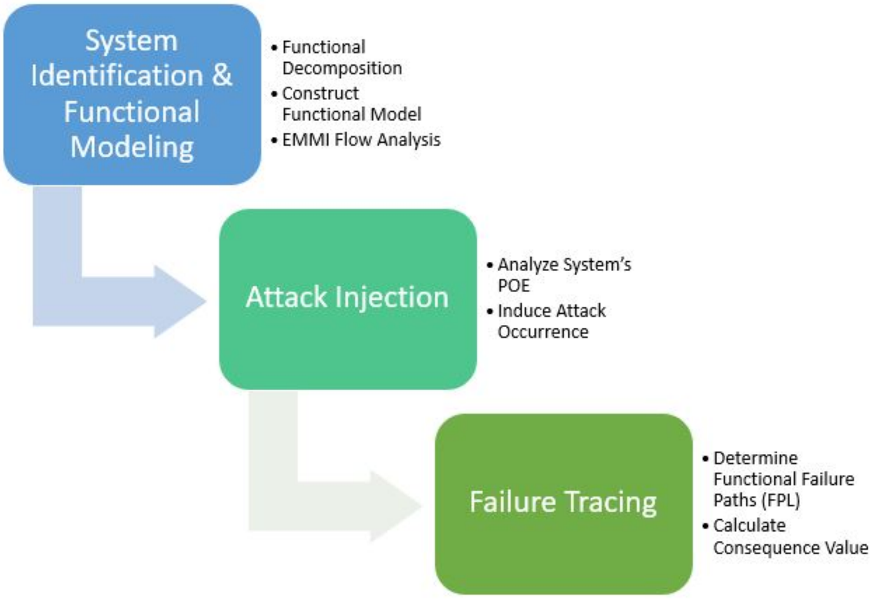

System Identification & Functional Modeling
• Functional Decomposition
• Construct Functional Model
• EMMI Flow Analysis
Attack Injection
• Analyze System's POE
• Induce Attack Occurrence
Failure Tracing
• Determine Functional Failure Paths (FPL)
• Calculate Consequence Value

Figure 4.1. The Failure Path Length Method (FPLM) Process Outline.

The FPLM begins first by identifying a system and developing a functional model with sufficient detail to describe total system functionality at the conceptual phase of design. Upon development of a functional model of a system, one generates a desired attack based on the system's projected operating environment (POE) on each individual function iteratively, and traces the failure through the model to determine each function's failure path length (FPL) value. A function's FPL is the number of functions within the system that ultimately fail as a result of an attack on, and complete loss of, the specified function, and is therefore a direct representation of the impact on total system functionality. Ultimately, the function's FPL is the primary input to objectively quantifying the consequence of an attack on the targeted function.

While the true consequence of a loss of functionality is also influenced by subjective qualities, to include the system's overall purpose (particularly if in a system of systems (SoS) architecture) and it's operating environment, many assumptions are made within this methodology and its application in the following chapter to provide an objective and numerical assessment example local to a simple system. Assumptions made in this method are as follows:

1. A function whose only output flow leads to the environment outside the boundary of the system is given an FPL of "one," under the assumption that effects would propagate outside the system in question and cannot be accurately assessed without knowledge of the outside systems and functions impacted. A function's FPL considers only the other impacted functions within the appropriate system boundary, and will end at the last function prior to crossing the system's functional boundary, therefore the FPL value of "one" accounts for loss of the targeted functional only.
2. Functions are assumed to have a binary set of operational states, i.e. functions operate either "nominally," or are "failed." This methodology assumes that an attacked function or a function along a failure path loses all capability, and the idea of a "degraded" or "partially functional" state does not occur.
3. The methodology assumes that all failures occurring due to an externally induced IE or attack on a function, to include those in forward, backward, and uncoupled failure propagation paths, are counted in a function's FPL.

4.1 Initial Functional Mapping

The FPLM first requires the development of a functional model of the system in question for analysis. This section will detail some of the various methods of modeling system function as well as how they pertain to the application of the FPLM.

4.1.1 Functional Modeling Techniques

Systems Modeling Language (SySML) is one of the primary languages used to statically or dynamically model system behavior, structure, parametrics, interactions, and requirements [33]. FFBD and enhanced functional flow block diagrams (EFFBD) are SySML techniques that assist with functional decomposition of a system and models the system's normal operating process [33]. Figure 4.2 is an example EFFBD developed to outline the process of powering on and charging a laptop device.

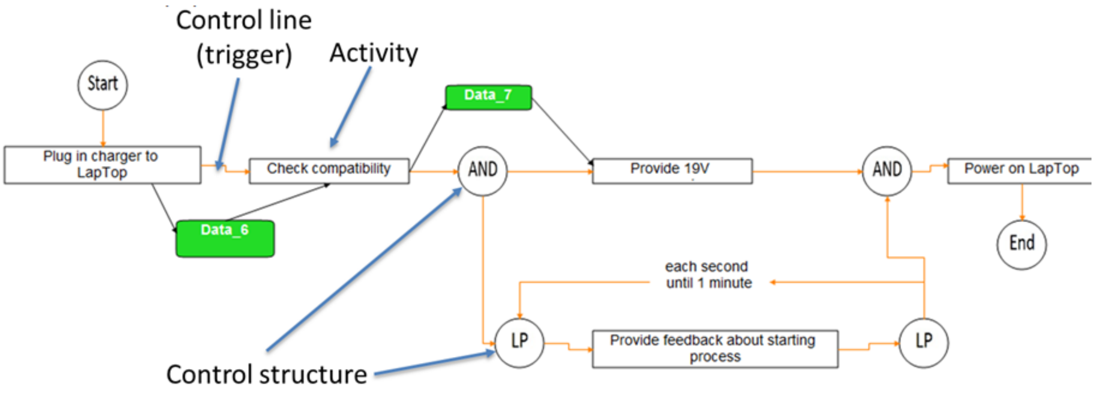


Figure 4.2. Example EFFBD of the Process of Powering on and Charging a Laptop. Source: [34].

A modeler employing the FPLM first constructs an EFFBD like in Figure 4.2 by determining all functions or "activities" required in the process to fulfill the requirement that the system must achieve, in this case, providing the appropriate power source to a laptop computer. When the functions are identified through functional decomposition processes, they are organized in chronological order as required by the identified process. EFFBD specifically includes further control logic and syntax to outline the order of functional operation, namely, the "Start,", "And", "LP" (for "Loop"), and "End" control structure nodes in Figure 4.2. Finally, they are connected with control lines that represent directional flow of control in the process and trigger the following connected function to occur. While this methodology does not emphasize the use of specific control structures utilized in EFFBDs, the requirement of chronological arrangement of functions within the system and denoting control lines to trigger sequential function operation are elements of the functional models used in this development.

Integrated Computer-Aided Modeling (ICAM) Definition for Functional Modeling (IDEF0) is a separate, non-SySML, method for static functional modeling that also has similarities to the process used in this methodology [33]. Similar to FFBD and EFFBD, IDEF0 does model functional activities in a system process and does support the functional decomposition process, however it differs in that there is also a primary focus on the flows of EMMI between each function. Figure 4.3 is the basis for an IDEF0 diagram.

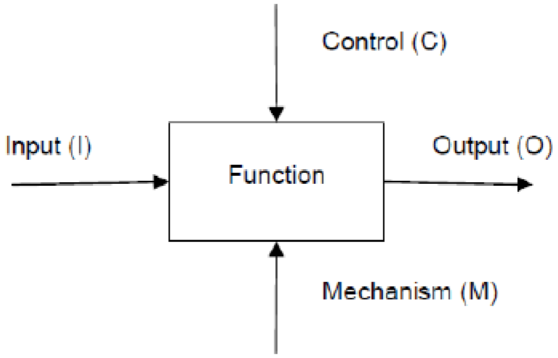


Figure 4.3. Basic Illustration of an IDEF0 Diagram with Elements and Semantics. Source: [35].

IDEF0 diagrams are first formed by determining all the sub-functions required to accomplish a parent function's objective, as well as the inputs into the function, outputs from the function, controls that guide the function in its activity, and the mechanisms that actually execute the function's activities. Each sub-function is connected via inputs and outputs (which in the case of the following process, are the EMMI flows), and may utilize syntax to create closed feedback loops between inputs, controls, or mechanisms for interrelated sub-functions [33]. IDEF0's static representation of a system's functions and emphasis on functional EMMI inputs and outputs specifically contribute to the functional mapping conducted in the following methodology.

4.1.2 Functional Model Development

Functional models, as described in [9] and [36], arrange functions and sub-functions within a system and graphically represent how each individual function imports EMMI, alters it via performed action, and exports it to other functions for the overall system to accomplish its mission. Within a functional model, functions (nodes) are connected by flows (edges) that are descriptive of the type of EMMI exchanged between the connected functions. The construction of the functional model in this section follows an approach incorporating elements of IDEF0 and FFBDs/EFFBDs. The first step in the FPLM is to identify the system, all requisite functions performed by the overall system, and determine the nature of the functions' connectedness.

Figure 4.4 illustrates a basic functional model of a fictitious liquid distribution system designed to generate, separate, and distribute both cold and hot water to other external systems operating in a shipboard environment.

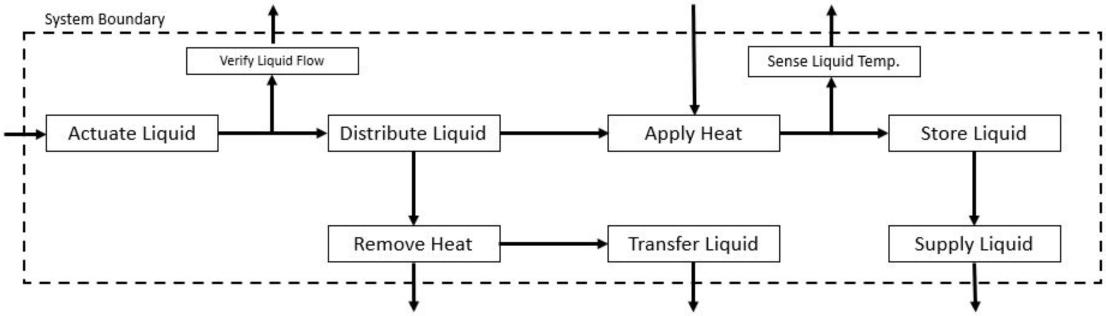


Figure 4.4. Functional Block Diagram of a Liquid Distribution System.

Within Figure 4.4, functional nodes are described in verb-noun format to emphasize a lack of fidelity in the physical architecture of the system. The edges illustrate functional connectivity and directionality, however they require further description to model the exchange across their individual interfaces and facilitate failure propagation path tracing.

4.1.3 Flow Delineation

The Functional Basis established in [15] and its expansion in [14] form flow descriptions for EMMI exchanged across functional and physical boundaries, to include behavior variables that correlate to different flow descriptors within the realms of energy, matter, or information. Applying the expanded functional basis to Figure 4.4 yields the more enhanced and complete view of the liquid distribution system seen in Figure 4.5 by specifying applicable FBED descriptors for each individual functional flow.

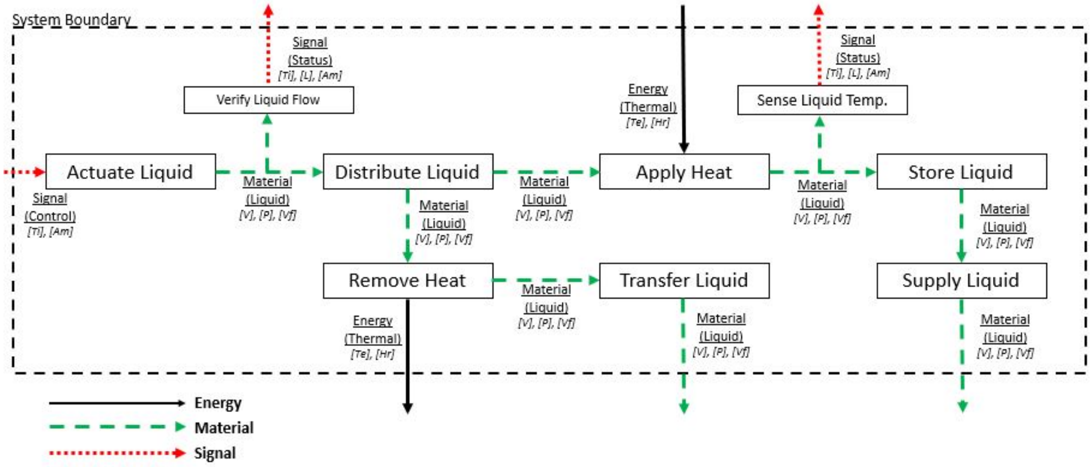


Figure 4.5. Liquid Distribution Piping System with FBED Designations.

Each edge connection in Figure 4.5 is defined by flow type and further decomposed with appropriate behavior variables. Behavior variables are a required feature in the FPLM to help the assessor verify propagation of a failure from an attack, as seen in the next step, and also help define the nature of the attack and its flows when induced in a model.

4.2 Attack Injection and Failure Tracing

As seen in Figure 4.1, Attack Induction and Failure Tracing are the follow on processes required for application of the FPLM once an adequate functional model is developed. This section addresses the differences and relationship between attacks and failures, discusses how attacks are injected into a functional model during application of the FPLM, and how resulting functional failures are identified and traced.

4.2.1 Attack - Failure Relationship

Attacks and failures are related, yet dissimilar in a few ways. First, while both failures and attacks can be described as events that may occur during nominal system operation, the author chooses to use failure to describe the *result* of an attack occurrence. Reiterating the assumptions made in this chapter, the concept of an attack shares a “cause and effect” relationship with the concept of failure, where attack occurrences cause failures in the functions or systems they affect. Second, failures generally occur in a probabilistic manner and are a function of time in operation. Equation 4.1 is the equation for system reliability, R(t) [37]:

$$R(t) = e^{-\lambda t} \tag{4.1}$$

where λ, or the failure rate for a system or function, is determined by Equation 4.2 [37]:

$$\lambda = \frac{1}{MTBF} \tag{4.2}$$

Traditionally, mean time between failures (MTBF) is measured in operating hours per system failure, which illustrates how failure is often quantitatively driven by time and determined using historical data of a system's operation. Unlike failures, when measured under ideal conditions where Equation 4.2 is most applicable, attacks are random and unpredictable; even in objectively hostile system operating environments, the exact time when a specific type of attack or IE will occur is typically unidentifiable by traditional means. However, using the attack taxonomy presented in Section 3.3 and the system developer's knowledge of the system's POE and potential threats, the FPLM is best utilized early in the system life cycle where *identification* of functional failure and its impacts are more significant than determining the time-based *likelihood* of failure. Here, attacks are induced in functional models based on attacking agents and corresponding behavior variables that attack is likely to influence as identified in Section 3.3.

Designers can induce attacks in various ways in functional models. In the case of Physical class attacks, an attack is likely to be induced on a specific function to initiate failure propagation within the system due to most Physical class attacks being most effective at targeting the components of a system that are accomplishing a function. However, in the case of Signal and Energy class attacks, they may be induced on the input or output flows of a function to initiate failure via alteration of the exchanged EMMI required for proper execution of the targeted function, or the function itself. Attack induction during modeling is overall based on the type and nature of the attack, and its desired result at the functional/system-level or in the realm of the capability normally provided by the system in its POE.

4.2.2 Attack Occurrence

This section continues the application of the attack taxonomy to the FPLM using the system described in Figure 4.5. In a shipboard environment where the ship system's POE may be an overall hostile theater of operations with various adversaries present, numerous opportunities for Physical class attacks exist. The most important of these would be Conventional type Kinetic attacks, where an adversary may fire ballistic weapons or missiles to impact the ship system and degrade functionality and capability. In Figure 4.6, a ship subsystem designer tests the vulnerability of the liquid distribution system to physical attacks on its major components by inducing a Kinetic attack first on the "Distribute Liquid" function, which is intended to cause immediate failure.

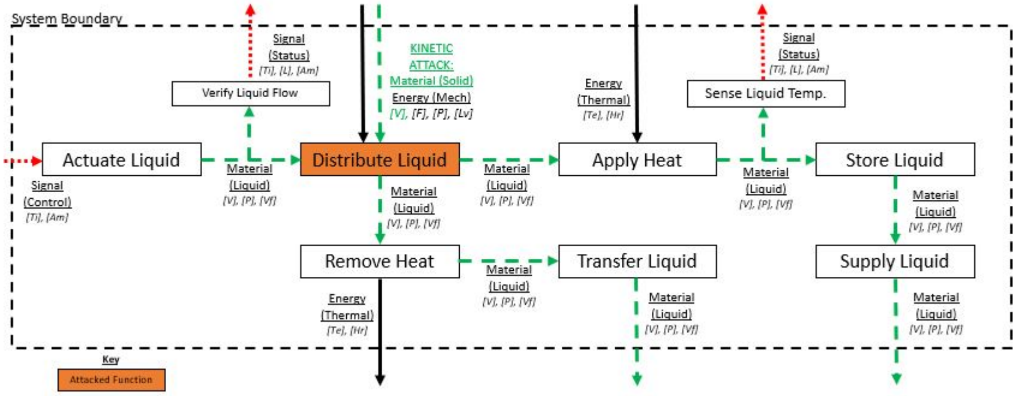

System Boundary
Signal (Status) [Ti], [L], [Am]
Verify Liquid Flow
KINETIC ATTACK: Material (Solid)
Energy (Mech)
[V], [F], [P], [Lv]
Energy (Thermal) [Te], [Hr]
Signal (Status) [Ti], [L], [Am]
Sense Liquid Temp.
Actuate Liquid
Signal (Control) [Ti], [Am]
Material (Liquid) [V], [P], [Vf]
Distribute Liquid
Material (Liquid) [V], [P], [Vf]
Material (Liquid) [V], [P], [Vf]
Apply Heat
Material (Liquid) [V], [P], [Vf]
Store Liquid
Material (Liquid) [V], [P], [Vf]
Remove Heat
Material (Liquid) [V], [P], [Vf]
Transfer Liquid
Supply Liquid
Energy (Thermal) [Te], [Hr]
Material (Liquid) [V], [P], [Vf]
Material (Liquid) [V], [P], [Vf]
Key
Attacked Function

Figure 4.6. Liquid Distribution Piping System with a Physical Attack Initiated at "Distribute Liquid" Function.

Based on Table 3.3, the attacking agent for this induced Kinetic attack is represented by a material flow and an additional energy flow representing the transferred mechanical energy at impact. The attack caused failure of the Distribute Liquid function due to the volume and linear velocity of the object at impact, and the force imparted on the component performing the function by the projectile or object used in the Kinetic attack. As failures follow behavior variables within flows that they affect [14], the failure path propagates to each sequential or connected function whose flows are impacted by a change in the "Distribute Liquid" function's output flow behavior variables. Failure paths end at the point in which an affected behavior variable is no longer produced as an output to a function, or in the case of the scope of this methodology, at the system boundary as appropriate.

4.3 Failure Path Length and Consequence

The final step of the FPLM estimates the objective impact and consequence to a system based on an attack and loss of a function through the determination of the FPL. A function's FPL is the total count of the number of functions within the system boundary that fail as a direct result of an induced attack on that individual function. Upon determining a function's FPL, the function's consequence metric is the ratio of impacted functions to total system functions, as seen in Equation 4.3.

$$Consequence = \frac{\text{FPL}}{TotalSystemFunctions} \tag{4.3}$$

In Figure 4.7, the flows in and out of the "Distribute Liquid" function indicate that the function is designed to influence the volume [V], pressure [P], and volumetric flow rate [Vf] of the liquid material (water) in the system. Because functional failures propagate through systems via impacted behavior variables, the failure of the attacked function ultimately continues and impacts a total of six more functions with directly related input and output behavior variables before the path ends at the system boundary interfaces after the "Transfer Liquid" and "Supply Liquid" functions. As mentioned in the assumptions of the FPLM, the failure propagation path discontinues at the system boundary of the system of concern; without knowledge of the functional architecture of outside connected systems or functions, the consequence of the attack and resulting failure cannot be assessed quantitatively.

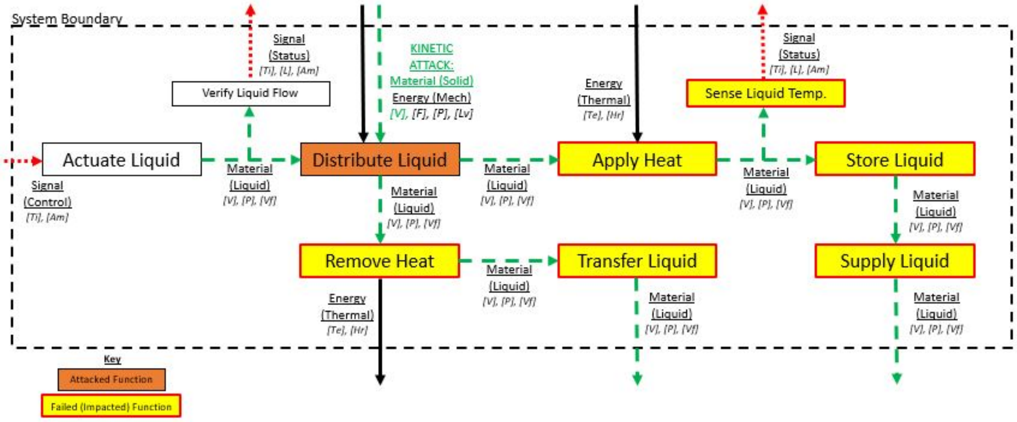


Figure 4.7. Liquid Distribution Piping System with Failure Propagation Path Illustrated from Attack Initiated at "Distribute Liquid" Function.

In this iteration, the function has a FPL of seven (including the failure of the attacked function), while there are nine total functions within the system boundary. Based on the propagation path initiated by a solid material attack on the "Distribute Liquid" function, there is an objective total consequence of 78%.

$$Consequence_{DistributeLiquid} = \frac{\text{FPL}_{DistributeLiquid}}{TotalSystemFunctions} = \frac{7}{9} = 0.78 \tag{4.4}$$

This method to produce the FPL should be conducted for each system function to provide a consequence value for each. Table 4.1 displays the consequence based on the Kinetic attack as induced at each of the functions listed. While different types of attacks may subjectively have increased or decreased impact or effect on different types of systems, components, and functions, this method treats each attack occurrence similarly due to the assumption that an attack targeting a function or it's EMMI inputs initiates a full failure on said function, and the functions along its unique propagation path fail in similar manner.

Table 4.1. Failure Path Length and Consequence Values for Liquid Distribution System Functions.

Function	Function Name	Failure Path Length	System Impact (Consequence)
1.0	Actuate Liquid	9	100%
2.0	Verify Liquid Flow	1	11%
3.0	Distribute Liquid	7	78%
4.0	Add Heat	6	67%
5.0	Sense Liquid Temp.	1	11%
6.0	Store Liquid	2	22%
7.0	Remove Heat	2	22%
8.0	Transfer Liquid	1	11%
9.0	Supply Liquid	1	11%

Through the process followed and based on information provided in Table 4.1, a direct attack on the "Actuate Liquid" function would have the highest negative impact on this system. This follows a logical process of thought seeing functional flow for the liquid distribution system originally illustrated in Figure 4.5; if the system's ability to initiate fluid flow is inhibited, all other functions within the system will be unable to conduct their respective actions due to lack of liquid material to act upon. This instance displays application of the FPLM on a simple system, following logic outlined by a generally forward propagated failure path. The following chapter will apply the FPLM to a more realistic system and show capability to manage more complex failure path possibilities.

4.4 Chapter Summary

A function's FPL is a simple method to initially determine the level of potential consequence to an attack initiated on that function. While total consequence has a subjective measure dependent on system design and purpose, the author utilizes the FPLM to determine and objective and quantitative consequence value via iterative FPL determination to assist in risk mitigation decisions. Design decisions can also result from this manner of consequence assessment, as seen in the following chapter.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Case Study

This chapter will show the application of the FPLM on the functional flow of a realistic electrical power system (EPS). The purpose of an EPS is to provide electricity to the systems, or "loads," requiring it to function. Because of the common purpose of an EPS in many real-world applications, this case study ensures and verifies applicability of the FPLM and its benefits to a versatile set of industries and systems.

In this chapter, the case study further engages in a "What-If" analysis by conducting the FPLM on different functional architectures of the EPS. The "What-If" analysis conducted seeks to emphasize the decision-making power provided to designers that consider functional consequence values with regard to driving improvements in RAM and overall system suitability.

5.1 Initial EPS Architecture

For reference, Figure 5.1 illustrates a potential physical architecture of a standard EPS system solution designed to store, condition, and transport electricity to equipment loads requiring power.

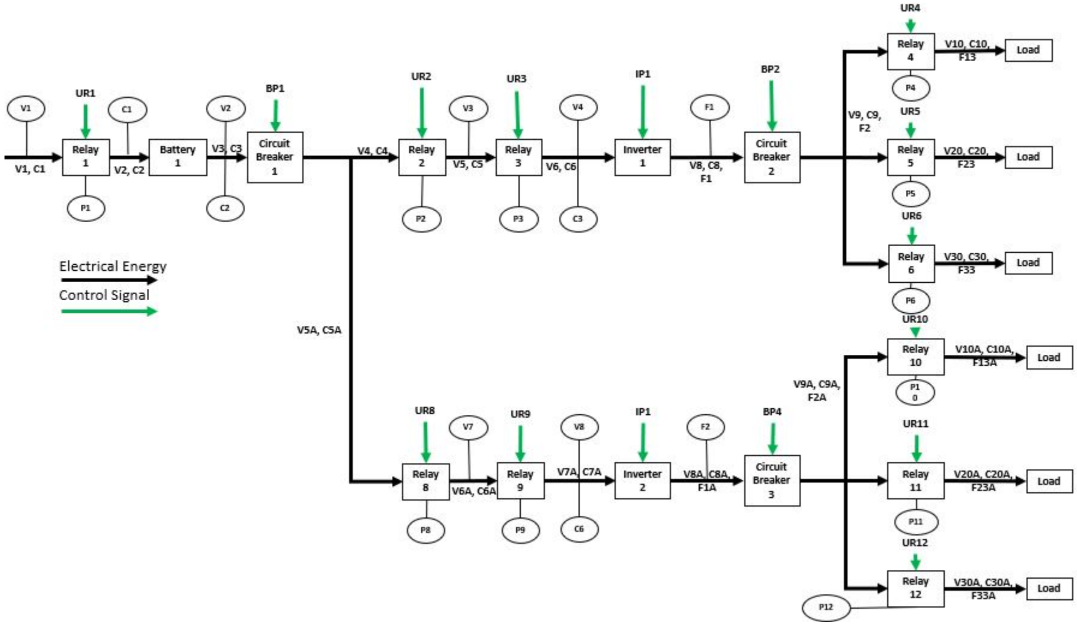


Figure 5.1. Physical Architecture Model of an EPS with One Power Source and Two Load Sets.

This EPS has a single source of electricity (as evidenced by only one Energy flow entering the system boundary from the top-left of the figure at Relay 1) ultimately providing power to six electrical loads arranged in two load sets on the right side of the figure. Along the line are breakers, relays, batteries, and inverters used to store, condition, and control the flow of electricity while in transit. Each breaker or relay involves the input of a control mechanism to initiate and control the flow and amount of energy through the component.

While Figure 5.1 seems to provide a well-intact solution to the need for an EPS, both the SE process and the FPLM require the development of a functional architecture to first ensure accomplishment of the main requirements of the system during early design phases that occur prior to allocation of physical components. This section begins the process of functional decomposition and proceeds to apply the FPLM to this original EPS architecture.

5.1.1 Functional Decomposition and Model Generation

Figure 5.2 is the decomposition of the system-level function required to be performed by an EPS, which is to “Provide Electrical Power” to all necessary equipment.

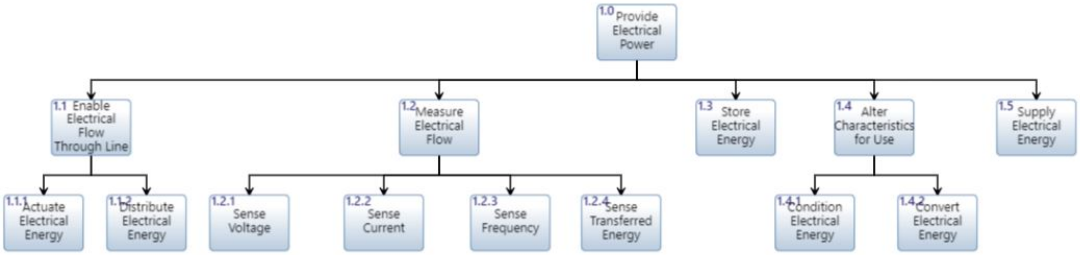

1.0 Provide Electrical Power
1.1 Enable Electrical Flow Through Line
1.2 Measure Electrical Flow
1.3 Store Electrical Energy
1.4 Alter Characteristics for Use
1.5 Supply Electrical Energy
1.1.1 Actuate Electrical Energy
1.1.2 Distribute Electrical Energy
1.2.1 Sense Voltage
1.2.2 Sense Current
1.2.3 Sense Frequency
1.2.4 Sense Transferred Energy
1.4.1 Condition Electrical Energy
1.4.2 Convert Electrical Energy

Figure 5.2. Functional Decomposition of an EPS in Early Design Phases.

The child-level functions that contribute to the accomplishment of the system-level requirement of providing electrical power are arranged to support the generation of a functional model of a system. Following the SE process, Figure 5.3 illustrates the functional model for a basic EPS that is designed to provide electricity to six pieces of equipment or components, correlating to a level of fidelity in functional architecture to that identified in Section 4.1.3.

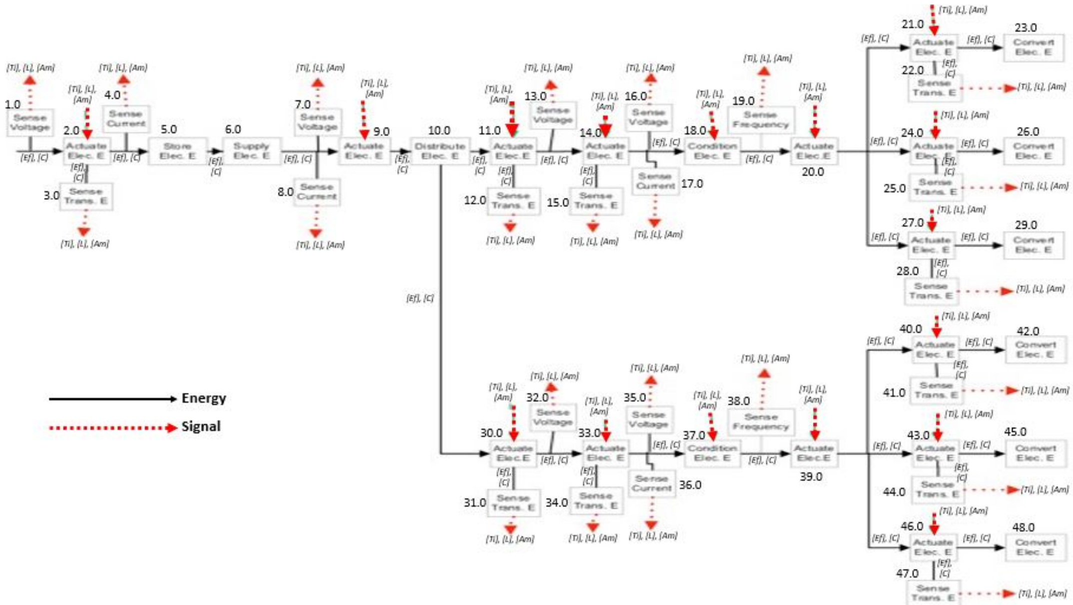


Figure 5.3. Functional Model of an EPS with One Power Source and Two Load Sets.

As seen in Figure 5.3, the primary EMMI flows in and out of most functions are Energy, further specified as electrical energy defined by electromotive force [Ef] and electrical current [C] behavior variables. Each “Sense” function reads current and voltage passively along functional interfaces and outputs a Signal flow carrying information defining the status of the electrical energy in that part of the line (e.g. the logged time [Ti] of signal reading, the location [L] of the reading within the system architecture, and the amplitude [Am] or strength of the input electricity characteristics) to facilitate measurement of electrical flow. The “Sense” functions output the information Signal flows to functions or operators outside of the EPS system boundary. In order to enable electrical flow through the line, “Distribute” and “Actuate” functions periodically placed in the line allow flow to and meter electricity levels through different parts of the system, with each “Actuate” function requiring an electrical Energy input flow, and a control-type Signal flow that directs the function to start or stop flow as well as change the amplitude strength of the current and voltage outputs to match input requirements of components further in the line.

5.1.2 Attack Considerations and Injection

As an EPS is simply a system of electrical components used to route, use, or store electrical energy, they are seen in a wide variety of industrial, residential, commercial, or military-operating environments. Therefore they can be exposed to many of the types of attacks discussed within Section 3.3. Commonly, attackers target electrical power grids as a means of reducing target capability via losses of power to critical equipment and infrastructure. For example, Signal class attacks focusing on systems that require significant information exchanges across functional interfaces for operation are a popular method of attack, particularly on functions that are highly automated and whose successful execution of the function is highly dependent on accuracy of information within the input signal flows to the function.

A risk not often considered during system design are incidents in which attacks are generated by insider threats; personnel intimately familiar with a system’s architecture and operation, such as maintenance personnel with continual authorized access to system components and functions, have opportunities to conduct malicious activity or sabotage. Based on Figure 5.3, fewer of the functions have functional inputs represented by signals that could be easily manipulated to cause failure, however each function ultimately requires allocation to a part or component that can be physically destroyed to cause failure. This leads the author to test a scenario where the illustrated EPS is utilized in an industrial environment where multiple maintenance personnel have authorized access to each function in the EPS, and a maintainer makes a choice to maliciously and physically destroy or degrade various components in the EPS. As a system designer, the significance of determining which function(s) would have the highest negative impact in the case of insider threat would be invaluable.

Figure 5.4 illustrates the EPS functional model in an instance in which the malicious attacker utilizes a solid material (tools for maintaining electrical components, shop parts, or other matter) to conduct a Kinetic attack and eliminates Function 2.0 capability within the EPS architecture.

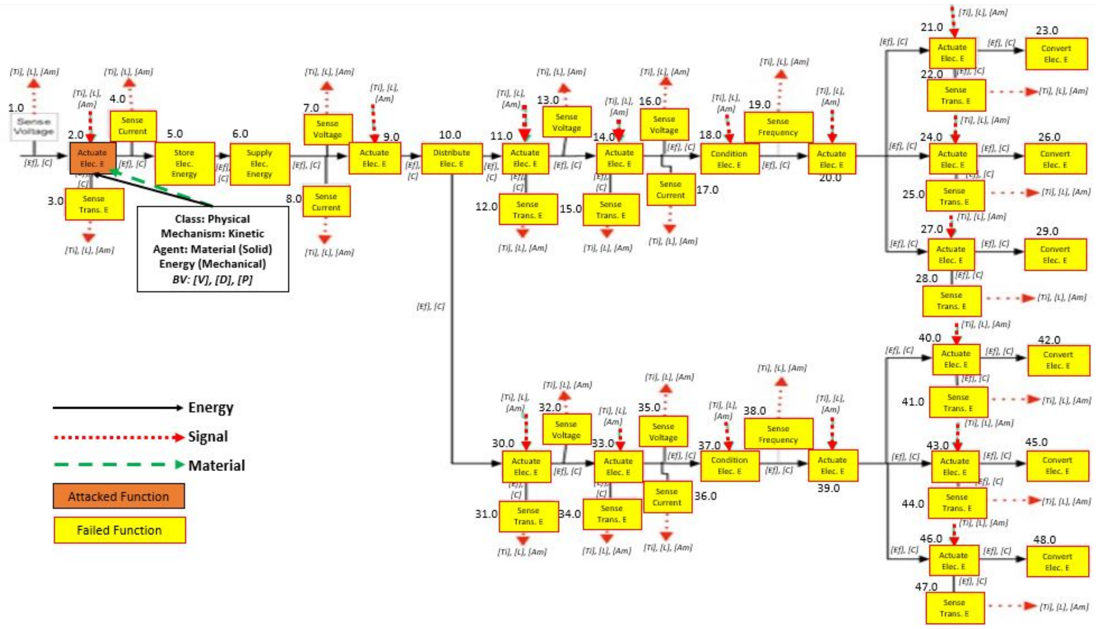


Figure 5.4. Functional Model of an EPS with a Kinetic Attack Conducted on Function 2.0, Actuate Electrical Energy.

An attack that causes failure of Function 2.0, "Actuate Electrical Energy," initiates a failure path that propagates to each follow on function requiring flow of electrical current and electromotive force (voltage). Because the execution of the FPLM requires the simulated attacking of each individual function iteratively and analysis of the ensuing failure propagation path to determine each function's consequence value, Figures 5.5 and 5.6 illustrate expected failure paths if the same attack were initiated on Functions 10.0 and 37.0, respectively, as further examples of each function's failure propagation path and FPL determination.

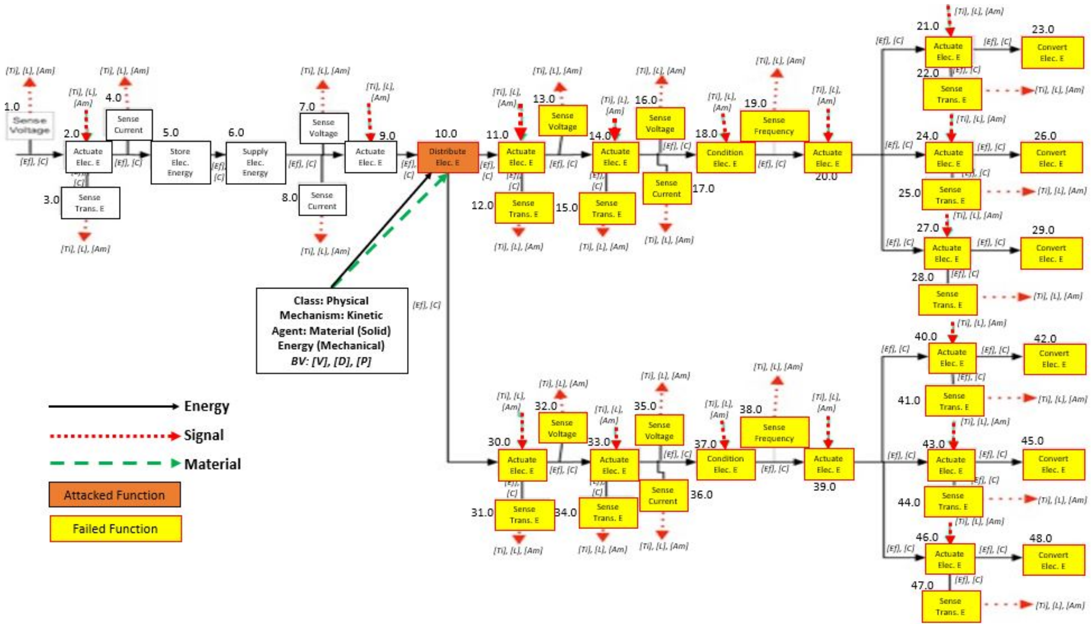


Figure 5.5. Functional Model of an EPS with Kinetic Attack Conducted on Function 10.0.

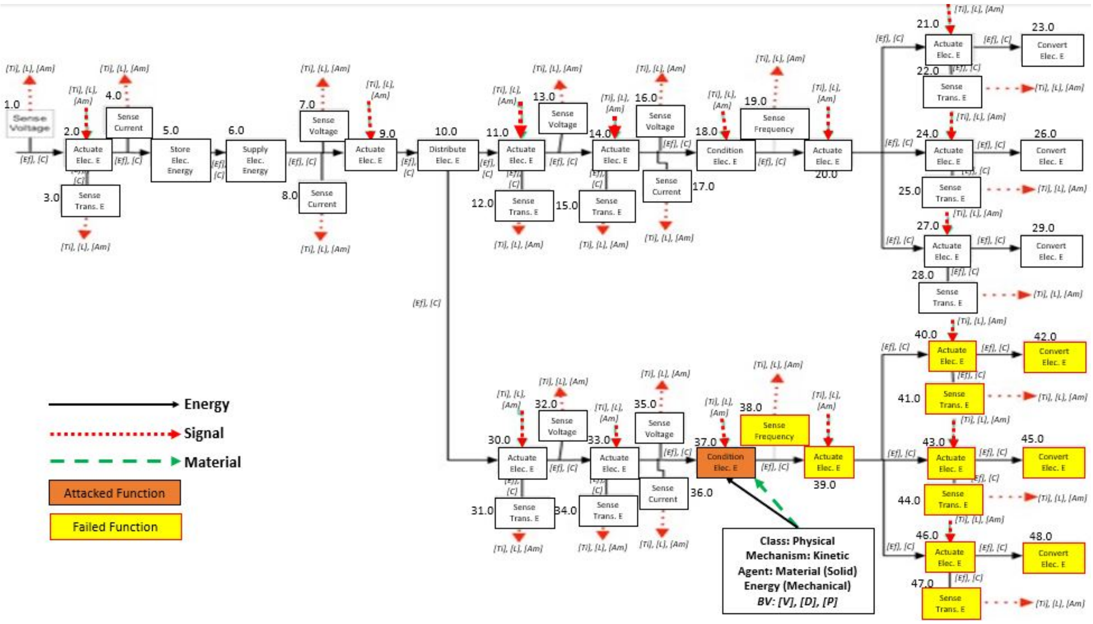


Figure 5.6. Functional Model of an EPS With Kinetic Attack Conducted on Function 37.0.

5.1.3 Failure Path Length Generation

Table 5.1 provides the FPL and resultant consequence value for the eleven functions with the highest objective impact provided an insider conducted a Kinetic attack on the EPS. The full table is listed in Table B.1 in Appendix B.

Table 5.1. Abbreviated Consequence Table for an EPS with One Power Source and Two Load Sets.

Function	Function Name	Failure Path Length	System Impact (Consequence)
2.0	Actuate Electrical Energy	47	98%
5.0	Store Electrical Energy	44	92%
6.0	Supply Electrical Energy	43	90%
9.0	Actuate Electrical Energy	40	83%
10.0	Distribute Electrical Energy	39	81%
11.0	Actuate Electrical Energy	19	40%
30.0	Actuate Electrical Energy	19	40%
14.0	Actuate Electrical Energy	16	33%
33.0	Actuate Electrical Energy	16	33%
18.0	Condition Electrical Energy	12	25%
37.0	Condition Electrical Energy	12	25%

In the initial EPS architecture illustrated in Figure 5.3, an attack causing failure of Function 2.0, “Actuate Electrical Energy,” causes the subsequent degradation and inhibits the passage of electrical energy to 46 follow-on functions. This FPL creates the highest consequence value in the entire system of 48 functions, impacting 98% of the entire EPS.

5.2 Alternative System Architecture and Consequence Variation

This section takes information provided in the first application of the FPLM in Section 5.1 and proceeds with a short analysis of alternatives driven by the previous consequence determination. A new EPS architecture is presented and the author applies the FPLM in a similar fashion to justify the value it adds in improving system design during functional analysis.

5.2.1 Alternative EPS Architecture

The purpose of this case study is to determine how FPL can drive changes in system design to ultimately lower consequence of attack. In Section 5.1.3, system designers recognized EPS architectures that involved one in-flow of electrical energy would present a small but highly impactful attack surface. FPL determination showed that a single insider threat action aimed at just one of any of five functions at the beginning of the line (functions 2.0, 5.0, 6.0, 9.0, and 10.0 in Figure 5.3) would have the potential to degrade the performance of 81-98% of the whole system.

There are numerous approaches to mitigating the specific risk posed by designing a system with only one input source. A common theme in the design of complex systems that generally strengthens overall reliability and availability is the addition of redundant input sources or parallel paths by which EMMI can follow that still allow the system to continually accomplish its system-level requirement despite the potential presence of functional failures. Figure 5.7 represents this post-analysis functional design improvement of the EPS architecture originally presented in Figure 5.3.

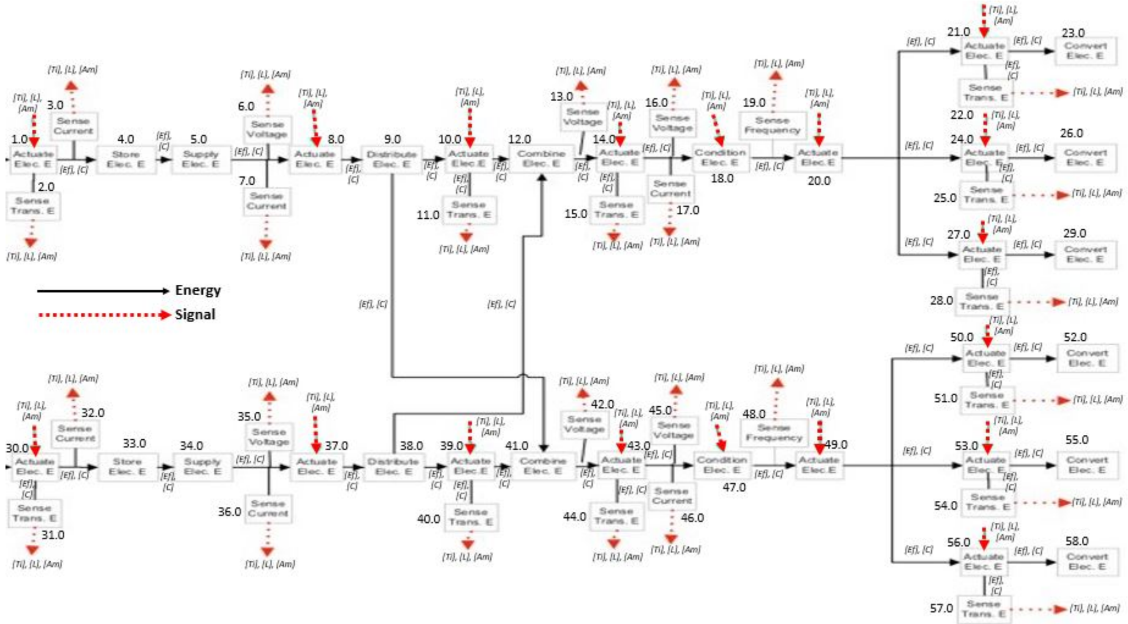


Figure 5.7. Functional Model of an EPS with Two Power Sources and Two Load Sets.

In Figure 5.7, system designers determined the simple approach of redundancy in source power, as previously described, would be an effective risk management strategy with regard to minimizing the EPS' potential downtime. The redundancy in the design is represented by a second in-flow of electrical energy to the EPS functional architecture at Function 30.0's boundary, as well as two pairs of Distribute - Combine Electrical Energy functions (Function 9.0 connected to 41.0, and 12.0 connected to 38.0, respectively) ensuring cross-connected electrical flow in a parallel plant configuration.

5.2.2 Attack Injection and FPL Determination

System designers accomplish attack injection in the same manner as previous, iteratively initiating the insider threat action on each function in Figure 5.7. Figure 5.8 illustrates the failure propagation path occurring during the first iteration due to the Kinetic attack on Function 1.0, Actuate Electrical Energy.

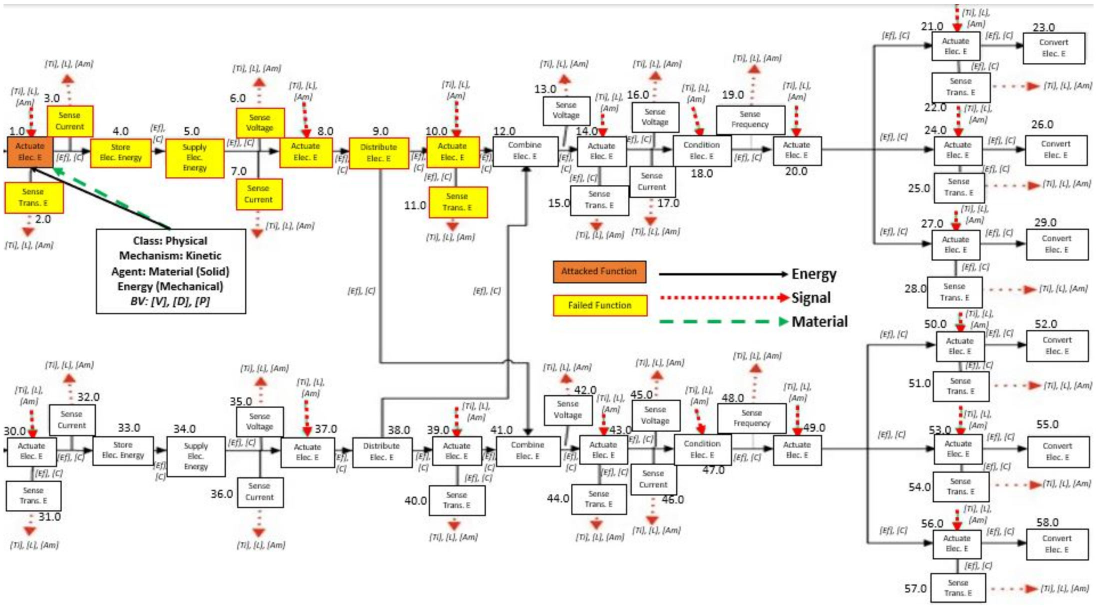


Figure 5.8. Functional Model of a Two-Power Source, Two-Load Set EPS with Kinetic Attack Conducted on an Actuate Energy Function.

The propagation path in Figure 5.8 should be compared to that of the attack exemplified in Figure 5.4 on the same function. Table 5.2 contains the FPL data for the eleven functions with the highest assessed consequence based on the iterative application of the Kinetic attack on each function in the improved EPS architecture, with the full listing of all functions and their respective FPLs in Table B.2 of Appendix B.

Table 5.2. Abbreviated Consequence Table for an EPS with Two Power Sources and Two Load Sets.

Function	Function Name	Failure Path Length	System Impact (Consequence)
12.0	Combine Electrical Energy	18	31%
41.0	Combine Electrical Energy	18	31%
14.0	Actuate Electrical Energy	16	28%
43.0	Actuate Electrical Energy	16	28%
Continued on next page			

Table 5.2 continued from previous page

Function	Function Name	Failure Path Length	System Impact (Consequence)
18.0	Condition Electrical Energy	12	21%
47.0	Condition Electrical Energy	12	21%
1.0	Actuate Electrical Energy	11	19%
30.0	Actuate Electrical Energy	11	19%
20.0	Actuate Electrical Energy	10	17%
49.0	Actuate Electrical Energy	10	17%
4.0	Store Electrical Energy	8	14%

5.3 Chapter Summary

This chapter presented a case study in which different architectures of a commonly used system, an EPS, were examined utilizing the FPLM to determine functions which would create the most substantial negative consequence should they be the target of an attack initiated by a source familiar with the system in question. Analysis of an initial functional model under attack provided ample information to drive design changes to the system and ultimately contributed to an analysis of alternatives process by determining consequence values for different potential system architecture solutions. Chapter 6 will further analyze the results of the case study in depth, and discuss the implications and advantages of the consequence metric overall.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 6:
Analysis and Discussion

This chapter serves as an analysis of the data compiled and presented during FPL determinations in Chapter 5. With the functional system architectures presented in Figures 5.3 and 5.7, as well as their consequence tables (Appendix Tables B.1 and B.2, respectively), the FPLM presents objective value during functional analysis and design. Through the analysis of the case study's results, this chapter also discusses the significance of the consequence metric and how interpretations of it can be used to drive improvements to system RAM at the functional design phase.

6.1 Consequence Reduction

A function's FPL can be interpreted as a numerical representation of its level of direct and indirect connectedness and impact potential within a system's architecture. Based on the assumption that executors of an attack have the intent of causing the highest rate of failure and capability degradation to their target, the consequence value associated with a specific function's FPL therefore provides a quantitative measure of the negative impact on a targeted system that function may have in the instance of an attack. The purpose of this functional analysis and follow-on improvements to system design is to significantly reduce any function's consequence value, for any reduction in consequence values within a system's functional architecture represents an improvement in the system's ability to perform all required subfunctions. Table 6.1 provides the consequence values of the eleven most highly impactful functions within each EPS architecture previously presented in Tables 5.1 and 5.2.

Table 6.1. Consequence Table Depicting the Positive Reduction Occurring to the Eleven Highest Consequence Values After Improvement to the EPS Functional Design and Configuration.

Rank	Consequence (Original Configuration)	Consequence (Parallel Configuration)	Reduction (Δ)
1	98%	31%	68%
2	92%	31%	66%
3	90%	28%	69%
4	83%	28%	66%
5	81%	21%	74%
6	40%	21%	48%
7	40%	19%	53%
8	33%	19%	42%
9	33%	17%	48%
10	25%	17%	32%
11	25%	14%	44%

The final column in Table 6.1 is the amount of reduction in consequence experienced due to system design improvement from the original EPS configuration to the parallel plant configuration as a percentage of the original configuration's highest values. The most consequential function in the improved architecture poses a negative impact to only 31% of the EPS as designed, which is a 68% reduction from the function with the most negative impact in the original architecture. Further discussed in Section 6.3, positive consequence reduction during functional design improvement processes represent enhancements to various aspects of system suitability, such as availability and maintainability.

6.2 Critical Attack Surface and Reduction

The original functional design for an EPS in Figure 5.3 allowed for a singular power source providing electrical power to six different electrical loads, each represented by a "Convert

Elec. Energy" function within the functional model. With the given attack injected at Functions 2.0, 5.0, 6.0, 9.0, or 10.0, the failure propagation paths for each of these functions eliminate the flow of electricity to *all six* Convert Elec. Energy functions, representing a full loss of system capability due to the inability of the EPS to accomplish it's system-level function identified in Figure 5.2. This illustrates how the FPL determination process ultimately identified the EPS' "critical functions," or those whose attack and resultant failure create the most significant likelihood of the system's inability to meet its system-level functional requirement. Equation 6.1 calculates the critical attack surface, or the ratio of critical functions to the total number of functions, in the original EPS architecture.

$$\begin{aligned} CriticalAttackSurface &= \frac{\#ofCriticalFunctions}{TotalSystemFunctions} \\ &= \frac{5}{48} \\ &= 10.4\% \end{aligned} \tag{6.1}$$

As seen in Table 5.1, these five critical functions also have the longest failure path lengths, leading to the highest consequence values. Consequence in this sense is defined simply as the number of functions negatively impacted by the attack on, and resulting failure of, a given function. Yet, these five functions additionally become critical functions for the EPS as their failure propagation paths also include all six electrical loads to which the original EPS is responsible for providing power, correlating to a loss of total system capability. In the case of this architecture, heightened consequence and critical attack surface size overlap, however they can be mutually exclusive.

In Table 5.2, the Combine Electrical Energy functions are identified as having the longest failure path lengths, correlating to the largest consequence upon attack and failure at 31%. Despite this, the Combine Electrical Energy function, individually, is not a critical function; Figure 6.1 illustrates the expected failure propagation path after attack on the Combine Electrical Energy function:

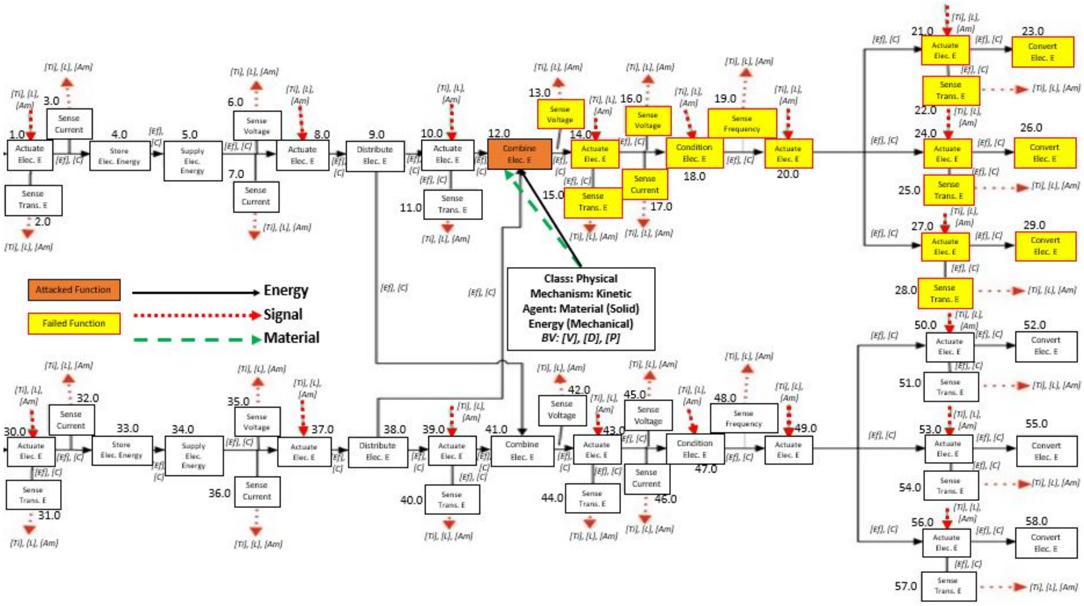


Figure 6.1. Failure Propagation Path Resulting from a Kinetic Attack on the "Combine Elec. Energy" Function within a Two-Power Source, Two-Load Set EPS.

Seen in Figure 6.1, the attack on the Combine Electrical Energy function causes the subsequent failure of 18 total functions, including only three of the six loads for which the EPS is responsible for providing energy. Despite being the function with the highest consequence value within the improved EPS architecture, the risk of complete failure of the EPS at providing electrical power to all six loads is mitigated due to the configuration change made during an analysis of alternative configurations originally influenced by FPL and consequence determination.

Simultaneously, the functional design improvement that occurred during the EPS' functional design phase resulting in the inclusion of cross connections with paired Distribute - Combine Electrical Energy functions ultimately created a functional architecture where no function, individually, is a critical function that causes a system-wide failure for the EPS as designed. Figure 6.2 shows the failure propagation path for an attack on the "Distribute Elec. Energy" function:

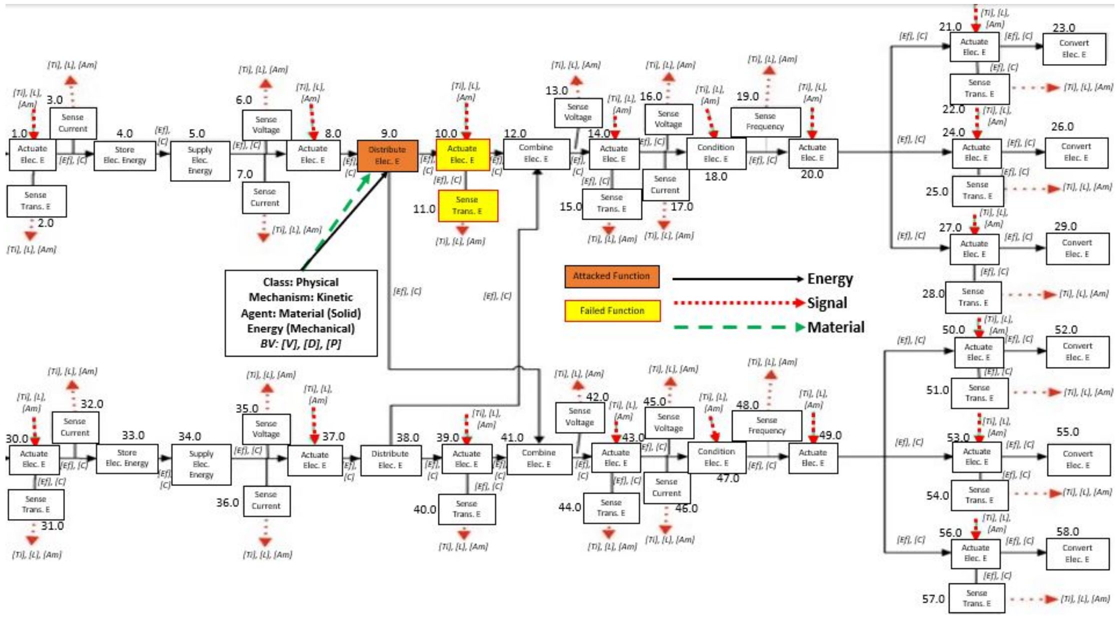


Figure 6.2. Failure Propagation Path Resulting from a Kinetic Attack on the "Distribute Elec. Energy" Function within a Two-Power Source, Two-Load Set EPS.

Seen in the figure, the failure path for the Distribute Electrical Energy function ends at each Combine Electrical Energy function. The removal of only one in-flow of electrical energy due to the attack on Function 9.0 does not prevent Function 41.0 from functioning, as electrical energy is still being received through the secondary line and power source represented by Functions 30.0 through 40.0. In a similar manner, Function 12.0 would still able to operate through the receipt of electrical energy from the parallel line's Distribute Electrical Energy function (Function 38.0), highlighting the effectiveness of the parallel plant configuration and how critical attack surface may be effectively minimized through a consequence-driven architecture improvement.

6.3 System Suitability Enhancement

The development and inclusion of a consequence metric during functional design is significant and provides added value as it can additionally serve as a metric to support understanding and enhancing other non-functional system suitability requirements.

Availability is one of the most prominent areas of system suitability supporting operational effectiveness throughout the life cycle, and is most easily related to the benefits of FPLM application. System designers ultimately seek to maximize a system's operational availability, defined as the "probability that a system or equipment, when used under stated conditions in an *actual* operational environment, will operate satisfactorily when called upon" [37]. Various maintainability metrics contribute to the nominal calculation of operational availability, as seen in Equation 6.2:

$$OperationalAvailability = A_0 = \frac{MTBM}{MTBM + MDT} \tag{6.2}$$

where mean time between maintenance (MTBM) considers times between both corrective and preventative maintenance actions, and maintenance downtime (MDT) includes all active maintenance time, logistics, and administrative delay times when parts, tools, and personnel are required for such action [37].

Based on Equation 6.2, any increase in MDT ultimately leads to decreased A_0. Functions identified as having high FPL and consequence values become functions that, when attacked, potentially create a higher MDT as a greater number of associated subsystems (sub-functions) in its failure path also fail or become damaged unexpectedly and more time becomes necessary for full restoration of capability. For example, many command, control, communications, computers, cyber, and intelligence (C5I) systems in use on DOD platforms have specific shutdown procedures required to be followed when equipment must be placed in a state of unavailability to perform preventative and corrective maintenance. Failure to perform these procedures often leads to catastrophic materiel casualties and requires additional downtime (increased MDT) for repair. This example becomes the exact case in the event of an unexpected attack on a function causing the untimely failure of various other associated functions and component subsystems. Determining consequence metrics and utilizing those results to drive design improvements during functional analysis with the sole purpose of reducing consequence serves as a value-added activity by augmenting operational availability through the reduction of the frequency of required corrective maintenance actions (increasing MTBM) and reducing MDT.

6.4 Chapter Summary

This chapter provided an in-depth analysis of how the application of the FPLM drove quantifiable improvement in a system's design at the functional stage of development. In the case of an EPS, FPL and consequence determination and analysis provided results that allowed system designers to alter and enhance the functional architecture of the system to allow for a reduced critical attack surface and system vulnerability. Additionally, the consequence metric provides the systems engineer with an additional measure to gauge potential instances where system RAM may suffer in the event of attack on certain functions. The results of the FPLM allows for a unique metric in which system design and prediction of impact to system is quantified and may be used as an additional key performance parameter during development of design requirements.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 7:
Conclusion

7.1 Overview

This thesis produces two distinct but related pieces of work to supplement existing knowledge on failure propagation within SE processes and functional analysis. First, in defining “attacks” and describing the concept of an attack as a separate concept from “failure,” this work produces an Attack Taxonomy that seeks to lay the foundation for categorizing different types of attacks that may impact a wide array of systems in their operating environments. The taxonomy presented first separates attacks at their lowest level of fidelity based on classes. The three attack classes, Signal, Energy, and Physical, each correlate to the type of EMMI they are best represented by. Signal class attacks occur through the manipulation of *information*, Energy class attacks occur through the injection (via radiation) or manipulation of *energy* exchanged between functions to render functions incapable, and Physical class attacks occur through the introduction of a *material* to a system’s functions to cause a loss of capability. The taxonomy further decomposes the classes into similar types and ultimately, mechanisms that represent the highest level of fidelity in describing the occurrence of an externally induced IE with malicious intent. Each mechanism within the taxonomy adapts flow descriptors and behavior variables from the functional basis to further describe the nature of the attack in question, and is included in the taxonomy.

Second, the work presents the FPLM as a means of developing a quantitative representation of the impact of an attack on a system’s functions. The FPLM first builds a functional model of the system in question, utilizing the functional basis to annotate and emphasize EMMI exchange across functional boundaries. With an understanding of the proposed system’s potential operating environment(s), system designers inject attacks from the attack taxonomy into the functional architecture of the system in an iterative manner to determine each function’s FPL. The FPL is the count of functions that ultimately fail due to the attack on, and resultant failure of, the initially targeted function in each iteration. Utilizing the FPL, the system designer calculates consequence, which is the ratio of the number of failed functions to the total number of functions within the system's functional architecture. This consequence metric can be a standalone value to help system designers understand the negative impact any given function within an architecture can have on the full system, or can be used to relate to other well known SE metrics such as reliability, availability, and maintainability. Ultimately, determining consequence in a quantitative manner provides system designers with objective knowledge to make design enhancements at the functional design phase to decrease a function's negative impact on the system to more acceptable levels, enhancing system suitability and effectiveness.

7.2 Future Work

Limitations exist in the use of the attack taxonomy presented in this work in the form of adherence to the requirement of a taxonomy being "comprehensive." While the taxonomy seeks to address attack mechanisms that can influence various system types and operating environments, this thesis presented and discussed many mechanisms framed around military operations, systems, and environments. This is evident in the discussion of Energy class attacks, where aspects of electronic warfare are prevalent in the discussions of DEW and jamming techniques employed on naval platforms and in doctrine, dominating the development of the mechanisms within the class. Further research on, and addition of, attack mechanisms readily employing more aspects of the functional basis in [15] and its expansion in [14] may help better achieve the "comprehensive" taxonomy requirement.

Limitations of the FPLM lie in the application of attacks utilized from the taxonomy. The FPLM's assumptions discussed in Chapter 4 and expanded upon in Section 4.3 assume an attack results in absolute failure of the targeted function and the follow-on functions in the targeted function's unique failure path. This means the FPLM disregards the possibility of functions remaining operational but in a degraded state, regardless of the attack mechanism to which it was exposed. Binary operational states such as "nominal" and "failed" for each function that are seen in this work allow for a more conservative, worst-case scenario calculation of FPL and consequence. However, certain mechanisms may have a different impact on specific functions, i.e., an attack mechanism within the Physical class may be more effective or cause a different level of damage to a function or system than a cyber-based Signal class attack depending on the system itself. The addition of steps to create an intermediate scale of operability level for each impacted function utilizing specific changes occurring to the behavior variables of a targeted function's inputs and outputs may prove a satisfactory improvement to the FPLM via enhanced accuracy of consequence determination.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A:
Malicious Attack Table

A.1 Attack Taxonomy

Table A.1. Hierarchical Taxonomy of Attacks Affecting Common Systems and System Functions.

Attack Class	**Attack Type**	**Attack Mechanism**	**Attacking Agent (Flow)**	**Behavior Variables**
Signal	Passive	Interruption	-Signal •Status	•Time[Ti] •Location[L]
		Interception	-Signal •Status •Control	•Time[Ti] •Location[L] •Amplitude[Am]
	Active	Modification	-Signal •Status •Control	•Time[Ti] •Location[L] •Amplitude[Am]
		Fabrication	-Signal •Status •Control	•Time[Ti] •Location[L] •Amplitude[Am]

Continued on next page

Table A.1 continued from previous page

Attack Class	Attack Type	Attack Mechanism	Attacking Agent (Flow)	Behavior Variables
Energy	Disruptive	Denial Jamming	-Energy •Electromagnetic -Signal •Status	•Intensity[I] •Location[L] •Time[Ti] •Amplitude[Am]
		Deception Jamming	-Energy •Electromagnetic -Signal •Status	•Intensity[I] •Location[L] •Time[Ti] •Amplitude[Am]
	Destructive	High-Energy Laser	-Energy •Electromagnetic •Thermal	•Chemical Elements [Ce] •Intensity[I] •Dimension[D] •Heat[H] •Particle Velocity[Pv] •Electromotive Force[Ef] •Current[C]
		High-Power Microwave	-Energy •Electromagnetic •Thermal	•Chemical Elements [Ce] •Intensity[I] •Dimension[D] •Heat[H] •Particle Velocity[Pv] •Electromotive Force[Ef] •Current[C]

Continued on next page

Table A.1 continued from previous page

Attack Class	Attack Type	Attack Mechanism	Attacking Agent (Flow)	Behavior Variables
Physical	Conventional	Concussion	-Material •Gas •Solid -Energy •Acoustic •Mechanical •Pneumatic •Thermal	•Pressure[P] •Force[F] •Heat[H] •Linear velocity[Lv]
		Kinetic	-Material •Solid •Liquid •Gas •Mixture -Energy •Mechanical •Human	•Volume[V] •Location[L] •Force[F] •Pressure[P] •Dimension[D] •Linear Velocity[Lv]
Continued on next page				

Table A.1 continued from previous page

Attack Class	Attack Type	Attack Mechanism	Attacking Agent (Flow)	Behavior Variables
Physical	Unconventional	Chemical	-Material •Liquid •Gas -Energy •Chemical •Thermal	•Reaction rate[Rr] •Intensity[I] •Temperature[Te] •Heat rate[Hr] •Chemical elements[Ce]
		Biological	-Material •Liquid •Gas -Energy •Chemical •Biological	•Reaction rate[Rr] •Intensity[I] •Chemical elements[Ce]
		Radiological	-Material •Liquid •Gas •Mixture •Solid -Energy •Radioactive •Chemical	•Reaction rate[Rr] •Intensity[I] •Chemical elements[Ce]

APPENDIX B:
Consequence Tables

B.1 One Source - Two Load Set EPS Architecture Consequence Table

Table B.1. Attack Consequence Table for an EPS with One Power Source and Two Load Sets.

Function	Function Name	Failure Path Length	Consequence
1.0	Sense Voltage	1	2%
2.0	Actuate Electrical Energy	47	98%
3.0	Sense Transferred Energy	1	2%
4.0	Sense Current	1	2%
5.0	Store Electrical Energy	44	92%
6.0	Supply Electrical Energy	43	90%
7.0	Sense Voltage	1	2%
8.0	Sense Current	1	2%
9.0	Actuate Electrical Energy	40	83%
10.0	Distribute Electrical Energy	39	81%
11.0	Actuate Electrical Energy	19	40%
12.0	Sense Transferred Energy	1	2%
13.0	Sense Voltage	1	2%
14.0	Actuate Electrical Energy	16	33%
15.0	Sense Transferred Energy	1	2%
16.0	Sense Voltage	1	2%
17.0	Sense Current	1	2%
18.0	Condition Electrical Energy	12	25%
19.0	Sense Frequency	1	2%
Continued on next page			

Table B.1 – continued from previous page

Function	Function Name	Failure Path Length	System Impact
20.0	Actuate Electrical Energy	10	21%
21.0	Actuate Electrical Energy	3	6%
22.0	Sense Transferred Energy	1	2%
23.0	Convert Electrical Energy	1	2%
24.0	Actuate Electrical Energy	3	6%
25.0	Sense Transferred Energy	1	2%
26.0	Convert Electrical Energy	1	2%
27.0	Actuate Electrical Energy	3	6%
28.0	Sense Transferred Energy	1	2%
29.0	Convert Electrical Energy	1	2%
30.0	Actuate Electrical Energy	19	40%
31.0	Sense Transferred Energy	1	2%
32.0	Sense Voltage	1	2%
33.0	Actuate Electrical Energy	16	33%
34.0	Sense Transferred Energy	1	2%
35.0	Sense Voltage	1	2%
36.0	Sense Current	1	2%
37.0	Condition Electrical Energy	12	25%
38.0	Sense Frequency	1	2%
39.0	Actuate Electrical Energy	10	21%
40.0	Actuate Electrical Energy	3	6%
41.0	Sense Transferred Energy	1	2%
42.0	Convert Electrical Energy	1	2%
43.0	Actuate Electrical Energy	3	6%
44.0	Sense Transferred Energy	1	2%
45.0	Convert Electrical Energy	1	2%
46.0	Actuate Electrical Energy	3	6%
47.0	Sense Transferred Energy	1	2%
48.0	Convert Electrical Energy	1	2%

B.2 Two Source - Two Load Set EPS Architecture Consequence Table

Table B.2. Attack Consequence Table for an EPS with Two Power Sources and Two Load Sets.

Function	Function Name	Failure Path Length	System Impact
1.0	Actuate Electrical Energy	11	19%
2.0	Sense Transferred Energy	1	2%
3.0	Sense Current	1	2%
4.0	Store Electrical Energy	8	14%
5.0	Supply Electrical Energy	7	12%
6.0	Sense Voltage	1	2%
7.0	Sense Current	1	2%
8.0	Actuate Electrical Energy	4	7%
9.0	Distribute Electrical Energy	3	5%
10.0	Actuate Electrical Energy	2	3%
11.0	Sense Transferred Energy	1	2%
12.0	Combine Electrical Energy	18	31%
13.0	Sense Voltage	1	2%
14.0	Actuate Electrical Energy	16	28%
15.0	Sense Transferred Energy	1	2%
16.0	Sense Voltage	1	2%
17.0	Sense Current	1	2%
18.0	Condition Electrical Energy	12	21%
19.0	Sense Frequency	1	2%
20.0	Actuate Electrical Energy	10	17%
21.0	Actuate Electrical Energy	3	5%
22.0	Sense Transferred Energy	1	2%
23.0	Convert Electrical Energy	1	2%
24.0	Actuate Electrical Energy	3	5%

Continued on next page

Table B.2 – continued from previous page

Function	Function Name	Failure Path Length	Consequence
25.0	Sense Transferred Energy	1	2%
26.0	Convert Electrical Energy	1	2%
27.0	Actuate Electrical Energy	3	5%
28.0	Sense Transferred Energy	1	2%
29.0	Convert Electrical Energy	1	2%
30.0	Actuate Electrical Energy	11	19%
31.0	Sense Transferred Energy	1	2%
32.0	Sense Current	1	2%
33.0	Store Electrical Energy	8	14%
34.0	Supply Electrical Energy	7	12%
35.0	Sense Voltage	1	2%
36.0	Sense Current	1	2%
37.0	Actuate Electrical Energy	4	7%
38.0	Distribute Electrical Energy	3	5%
39.0	Actuate Electrical Energy	2	3%
40.0	Sense Transferred Energy	1	2%
41.0	Combine Electrical Energy	18	31%
42.0	Sense Voltage	1	2%
43.0	Actuate Electrical Energy	16	28%
44.0	Sense Transferred Energy	1	2%
45.0	Sense Voltage	1	2%
46.0	Sense Current	1	2%
47.0	Condition Electrical Energy	12	21%
48.0	Sense Frequency	1	2%
49.0	Actuate Electrical Energy	10	17%
50.0	Actuate Electrical Energy	3	5%
51.0	Sense Transferred Energy	1	2%
52.0	Convert Electrical Energy	1	2%
53.0	Actuate Electrical Energy	3	5%

Continued on next page

Table B.2 – continued from previous page

Function	Function Name	Failure Path Length	Consequence
54.0	Sense Transferred Energy	1	2%
55.0	Convert Electrical Energy	1	2%
56.0	Actuate Electrical Energy	3	5%
57.0	Sense Transferred Energy	1	2%
58.0	Convert Electrical Energy	1	2%

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] Symantec Security Response, *W32.Stuxnet Dossier*, Version 1.4, 2011. [Online]. Available: https://docs.broadcom.com/doc/security-response-w32-stuxnet-dossier-11-en

[2] "Lessons from Stuxnet," *Computer*, vol. 44, no. 04, pp. 91–93, apr 2011.

[3] H. M. K. Zetter, "Revealed: How a secret Dutch mole aided the U.S.-Israeli Stuxnet cyberattack on Iran," Sept. 02, 2019. [Online]. Available: https://news.yahoo.com/revealed-how-a-secret-dutch-mole-aided-the-us-israeli-stuxnet-cyber-attack-on-iran-160026018.html

[4] B. M. O'Halloran, N. Papakonstantinou, and D. L. Van Bossuyt, "Assessing the consequence of cyber and physical malicious attacks in complex, cyber-physical systems during early system design," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*. IEEE, 2018, pp. 733–740.

[5] B. M. O'Halloran, R. B. Stone, and I. Y. Tumer, "A failure modes and mechanisms naming taxonomy," in *2012 Proceedings Annual Reliability and Maintainability Symposium*. IEEE, 2012, pp. 1–6.

[6] I. Y. Tumer, R. B. Stone, and D. G. Bell, "Requirements for a failure mode taxonomy for use in conceptual design," in *DS 31: Proceedings of ICED 03, the 14th International Conference on Engineering Design, Stockholm*, 2003.

[7] S. J. Uder, R. B. Stone, and I. Y. Tumer, "Failure analysis in subsystem design for space missions," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2004, vol. 46962, pp. 201–217.

[8] J. Howard, "An analysis of security incidents on the internet," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, USA, 1997.

[9] T. Kurtoglu and I. Y. Tumer, "A graph-based fault identification and propagation framework for functional design of complex systems," *Journal of mechanical design*, vol. 130, no. 5, 2008.

[10] *Procedures for Performing A Failure Mode, Effects and Criticality Analysis*, MIL-STD-1629A, Department of Defense, Washington, DC, 1980, pp. 1–5.

[11] *Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners*, 2nd ed., National Aeronautics and Space Administration, Washington, D.C., USA, 2011, pp. 3–1–3–4.

[12] "NASA Probabilistic Risk Assessment (PRA)," class notes for Risk Analysis and Management for Engineering Systems, Dept. of Sys. Eng., Naval Postgraduate School, Monterey, CA.

[13] N. Papakonstantinou, S. Proper, B. O'Halloran, and I. Y. Tumer, "Simulation based machine learning for fault detection in complex systems using the functional failure identification and propagation framework," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2014, vol. 46292, p. V01BT02A022.

[14] B. M. O'Halloran, N. Papakonstantinou, K. Giammarco, and D. L. Van Bossuyt, "A graph theory approach to predicting functional failure propagation during conceptual systems design," *Systems Engineering*, vol. 24, no. 2, pp. 100–121, 2021.

[15] J. Hirtz, R. B. Stone, D. A. McAdams, S. Szykman, and K. L. Wood, "A functional basis for engineering design: Reconciling and evolving previous efforts," *Research in Engineering Design*, vol. 13, no. 2, pp. 65–82, 2002.

[16] B. M. O'Halloran, R. B. Stone, and I. Y. Tumer, "Link between function-flow failure rates and failure modes for early design stage reliability analysis," in *ASME International Mechanical Engineering Congress and Exposition*, 2011, vol. 54952, pp. 457–467.

[17] Nonelectronic Parts Reliability Data, 1995. Reliability Analysis Information Center - Quanterion Solutions Incorporated. [Online]. Available: https://www.quanterion.com/. Accessed Apr. 22, 2021.

[18] R. A. Center, *Failure Mode/Mechanical Distributions, 1997*, 1st ed. New York, NY, USA: Rome, NY, 1997.

[19] Nonelectronic Parts Reliability Data, 2016. Reliability Analysis Information Center - Quanterion Solutions Incorporated. [Online]. Available: https://www.quanterion.com/. Accessed Apr. 22, 2021.

[20] R. I. A. Center, *Failure Mode/Mechanical Distributions, 2013*, 1st ed. Utica, NY, USA: Quanterion Solutions Incorporated, 2013.

[21] R. I. A. Center, *Failure Mode/Mechanical Distributions, 2016*, 1st ed. Utica, NY, USA: Quanterion Solutions Incorporated, 2016.

[22] H. C. J. Godfray, “Challenges for taxonomy,” *Nature*, vol. 417, no. 6884, pp. 17–19, 2002.

[23] *Biology Dictionary*. "Taxonomy," Apr. 28, 2017. [Online]. Available: https://biologydictionary.net/taxonomy/

[24] K. D. Bailey, *Typologies and taxonomies: An introduction to classification techniques*. SAGE, 1994, vol. 102.

[25] R. E. Ball, *Fundamentals of Aircraft Combat Survivability Analysis and Design*, 2nd ed. Reston, VA, USA: American Institute of Aeronautics and Astronautics, 2003.

[26] W. Stallings, *Network and Internetwork Security Principles and Practice*. Englewood Cliffs, NJ, USA: Prentice Hall, 1995.

[27] R. C. Harney, *Combat Systems Fundamentals: Command and Control Elements*, Monterey, CA, USA, 2013.

[28] R. C. Harney, *Combat Systems Fundamentals: Conventional Weapons*, Monterey, CA, USA, 2013.

[29] N. Academies and the U.S. Department of Homeland Defense, “Chemical attack: Warfare agents, industrial chemicals, and toxins,” Department of Homeland Defense, 2004, accessed on January 23, 2021. Available: https://www.dhs.gov/sites/default/files/publications/prep_chemical_fact_sheet.pdf

[30] N. Academies and the U.S. Department of Homeland Defense, “Biological attack: Human pathogens, biotoxins, and agricultural threats,” Department of Homeland Defense, 2004, accessed on January 23, 2021. Available: https://www.dhs.gov/sites/default/files/publications/prep_biological_fact_sheet.pdf

[31] N. Academies and the U.S. Department of Homeland Defense, “Radiological attack: Dirty bombs and other devices,” Department of Homeland Defense, 2004, accessed on January 23, 2021. Available: https://www.dhs.gov/sites/default/files/publications/prep_radiological_fact_sheet.pdf

[32] N. Academies and the U.S. Department of Homeland Defense, “Nuclear attack,” Department of Homeland Defense, 2004, accessed on January 23, 2021. Available: https://www.dhs.gov/sites/default/files/publications/prep_nuclear_fact_sheet.pdf

[33] D. Buede, *The Engineering Design of Systems - Models and Methods*, 2nd ed. Hoboken, NJ, USA: John Wiley Sons, 2009.

[34] A. Guarnaccia, “effbd,” Systems Engineering Advanced 6.2.x, Jan. 20, 2020 [Online]. Available: https://support.k-inside.com/display/SEA62/1.1.4+eFFBD

[35] A. Greenwood, P. Pawlewski, and G. Bocewicz, “A conceptual design tool to facilitate simulation model development: Object flow diagram,” 12 2013.

[36] C. Mutha, D. Jensen, I. Tumer, and C. Smidts, “An integrated multidomain functional failure and propagation analysis approach for safe system design,” *Artificial Intelligence for Engineering Design Analysis and Manufacturing*, vol. 27, 11 2013.

[37] B. S. Blanchard and W. J. Fabrycky, *Systems Engineering and Analysis*, 5th ed. Upper Saddle River, NJ, USA: Prentice Hall, 2014.

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California