EMPIRICALLY CHARACTERIZING EVOLVABILITY AND CHANGEABILITY IN ENGINEERING SYSTEMS

by

Jay Clark Beesemyer Jr.

B.S. Astronautical Engineering
United States Air Force Academy, 2010

Submitted to the Department of Aeronautics and Astronautics
in Partial Fulfillment of the Requirements for the Degrees of

Master of Science in Aeronautics and Astronautics

at the

Massachusetts Institute of Technology

June 2012

Signature of Author..
Department of Aeronautics and Astronautics
May 23, 2012

Certified by..
Donna H. Rhodes
Principal Research Scientist and Senior Lecturer, Engineering Systems
Director, Systems Engineering Advancement Research Initiative
Thesis Supervisor

Accepted by...
Adam M. Ross
Research Scientist, Engineering Systems
Lead Research Scientist, Systems Engineering Advancement Research Initiative
Thesis Co-Advisor

Accepted by...
Daniel E. Hastings
Professor of Aeronautics and Astronautics and Engineering Systems
Thesis Reader

Accepted by...
Eytan H. Modiano
Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

EMPIRICALLY CHARACTERIZING EVOLVABILITY AND CHANGEABILITY IN ENGINEERING SYSTEMS

by

Jay Clark Beesemyer Jr.

Submitted to the Department of Aeronautics and Astronautics in Partial Fulfillment of the Requirements for the Degree of Master of Science in Aeronautics and Astronautics

Abstract

The beginning phases of system development and conceptual design require careful consideration, as these decisions will have significant influence on system lifetime performance and are often made with incomplete system knowledge. Decision makers may improve their capacity to discriminate between system concepts and design choices by measuring a system's "ilities" such as changeability, evolvability, and survivability. These ilities may enable systems to respond to perturbations in the design space, context space, and needs space in order to ensure system functionality and adequate performance over time. A system may be designed to change in response to perturbations, or remain statically robust/survivable to perturbations in order to avoid deficiencies or failures.

This research attempts to analyze the mechanisms that allow system changes to occur. More specifically, this research will further the characterization of system changeability and evolvability and ultimately provide a structured and meaningful way of classifying system characteristics often described as "ilities". *Value sustainment* is proposed as an ultimate goal of systems, providing value in spite of perturbations in design, context, or needs. The premise of value sustainment is investigated through four distinct research thrusts: 1) a basis for defining system changes and ilities; 2) a system change examples database with categorical cluster analysis case research; 3) epoch-shift, impact, response, outcome case research; and 4) expert interviews case research. Focusing on change-related ilities, this research proposes constructs for identifying and enabling vague, yet desirable, system properties. Evolvability is characterized as a subset of changeability and defined as *the ability of an architecture to be inherited and changed across generations [over time]*, with a set of ten proposed design principles including decentralization, redundancy, targeted modularity, scalability, integrability, reconfigurability, mimicry, leverage ancestry, disruptive architectural overhaul, and resourceful exaptation.

Thesis Supervisor: Donna H. Rhodes
Title: Principal Research Scientist and Senior Lecturer, Engineering Systems

Co-Advisor and Thesis Reader: Adam M. Ross
Title: Research Scientist, Engineering Systems

To my family, Mom, Dad, and Shaina, who have motivated me my entire life to better myself and the world through the pursuit of knowledge with an everlasting desire for new and amazing experiences.

Acknowledgements

This exciting chapter of my life began in August of 2010 when I arrived in Boston after 60 days of obligatory fun and adventure following graduation from the United States Air Force Academy. Reminiscent of the four exhilarating years at the Academy, my time spent in New England was filled with immeasurable joy as well as nerve-wracking stress; endless hours in a lab or office as well as accidental days off relaxing in beautiful Boston; and tears of happiness as well as tears of sorrow. The two years working towards this thesis have been unforgettable to say the least, and I have had such a tremendous amount of support from so many of my colleagues, family, and friends.

Members of MIT SEAri have supported me and helped me the entire way. I have received endless support, guidance, and re-worded sentences from Dr. Adam Ross. My meetings with him throughout these past couple of years have been some of the most stimulating and academically rewarding discussions I have ever had. This thesis also would not have been possible without the help of Dr. Donna Rhodes. She has provided invaluable feedback and advice in these few years. Dr. Ross and Dr. Rhodes have spent endless hours helping with my thesis and have allowed me to produce something I have never felt capable of before. Their mentorship, combined with the leadership from Professor Daniel Hastings, has changed me for the better, and I am so lucky to have had them with me throughout my MIT career.

Many other SEAri researchers have also helped me along the way. Dating back all the way to graduation from the Academy, I have been going through this with my lab mate, Dan Fulcoly. From being a great roommate, along with Ben Saunders, to exploring Boston or Cambridge to helping me on similar research fronts, Dan has been through it all with me. I hope him the best in his career and family, and look forward to serving with him in the "real Air Force." Nirav Shah has also given me so much of his time, advice, and skills over the past years, helping me with almost anything I could imagine. This thesis, particularly the information with categorical cluster analysis, would not be possible without his help. Other SEAri researchers have helped me with advice, comments, and great conversations throughout my time here at MIT. Starting in the early days, Matt Fitzgerald, Dr. Zoe Szajnfarber, Greg O'Neil, Kacy Gerst, Augustine Friedel, Julia Nickel, and Brian Mekdeci helped me get acquainted with MIT and teach me the ways of the graduate student. Towards the latter half of my time, I had the pleasure of working with Nico Ricci, Paul Grogan, Hank Roark, and Henrique Gaspar. They have all made my graduate studies at MIT more enjoyable and rewarding.

Finally, my family and friends back home in Reno, here in Boston, and around the world have continued to give me the support I need and love. They have all made my time here in New England fabulous—an amazing chapter in my life that I will remember forever.

Biographical Note

J. Clark Beesemyer holds a Bachelors of Science in Astronautical Engineering from the United States Air Force Academy, graduating class of 2010. This thesis, upon publication, marks the completion of his Masters of Science from the Massachusetts Institute of Technology (MIT) Department of Astronautics and Astronautics. This S.M. degree was earned through the Systems Engineering Advancement Research Initiative (SEAri), a MIT research lab that is affiliated with both the Engineering Systems Division (ESD) and the Department of Aeronautics and Astronautics.

Born and raised in Reno, Nevada in 1987, Clark Beesemyer attended Reno High School. His experiences there as student body president and other leadership roles motivated him to join a profession grounded in serving others, the military. Graduating a National Merit Scholar, Clark received nominations from Senator Harry Reid and Senator Jim Gibbons to attend the United States Air Force Academy. In June of 2006, he headed off to Colorado Springs to attend basic training on the "Hill."

While at the Air Force Academy, Clark matured his affinity for practical science and math and declared a major in Astronautical Engineering. Serving in many leadership roles, he involved himself in many different rewarding programs at the Academy. Serving as a glider Instructor Pilot, he spent many afternoons and weekends training fellow cadets how to fly and solo an aircraft. He held numerous leadership positions within his squadron and the cadet wing. In his senior year, he served as the Chief Engineer for FalconSAT-5, a small satellite program exploring the effects of space weather. FalconSAT-5 was a unique undergraduate program that allowed cadets to gain experience building an advanced 160kg satellite, containing a Hall Effect ion thruster and an associated sensor suite. Clark graduated USAFA in 2010 as a Distinguished Graduate and moved to Cambridge, MA to attend MIT for graduate school.

At MIT, Clark studied under the Systems Engineering Advancement Research Initiative within the Department of Aeronautics and Astronautics. As a graduate student and research assistant, Clark focused on systems engineering research dealing with systems designed for changing in response to dynamic environments.

At the time of this thesis publication, Clark serves as a 1st Lieutenant in the U.S. Air Force. He will be attending the Euro-NATO Joint Jet Pilot Training program at Sheppard Air Force Base in Wichita Falls, TX.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

1.1.1 Managing Change

"Adapt or perish, now as ever, is nature's inexorable imperative," - H.G. Wells (1866-1946)

If there is one thing we know in engineering systems, it is that things *change.* Technologies improve, political policies shift, economies ebb and flow, markets fluctuate, and therefore the expectations placed on our systems are forever dynamic. A system designed, tested, and integrated in one context, may be forced to operate in a completely different manner or dissimilar environment, possibly fulfilling missions never imagined during conceptual development. These lessons are timeless and lead to fitting quotations like the one above from H.G. Wells or the military proverb with the "generals always preparing to fight the last war."

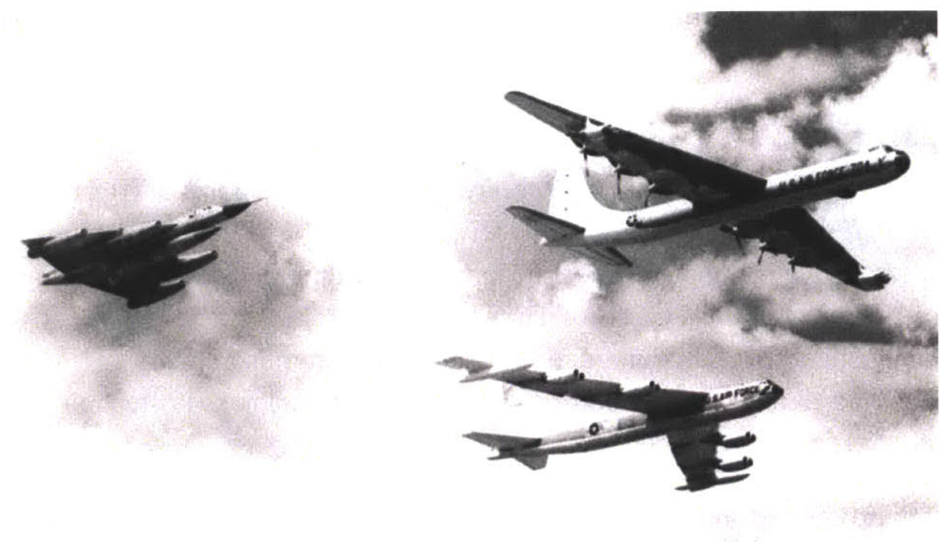


Figure 1-1. B-36, B-52 and B-58 from Carswell AFB, Texas in formation flight as B-36 is retired, 1958 (United States Air Force Historical Research Agency)

A great example of a system that has demonstrated the ability to manage this change is the Boeing B-52 Stratofortress or BUFF (Big Ugly Fat/Flying Fellow/*explicative*). With initial specifications issued on 23 November 1945, the first BUFF entered service in 1955 and has now served for 57 continuous years. According to current research, analysis shows that B-52 life spans may be extended out to 2045 (Saleh et al. 2003). The BUFF is long-range, heavy bomber supporting a diverse set of missions for the Air Force including air interdiction, offensive counter-air, or maritime operations and capable of dispensing a multitude of weapons including gravity bombs, cluster bombs, and guided missiles. From dropping thermonuclear weapons to ferrying manned and unmanned systems for altitude drops or orbital insertions, the B-52 has managed to provide value in many different contexts with diverse missions and stakeholders. Comparing the Convair B-58 Hustler to the B-52, as done in Saleh et al. (2003), the B-58 is filled with sophisticated equipment and capable of high performance supersonic flight. Together the B-58 and B-52 represented, in their time, the new-age Cold War Era bombers, as seen in the B-36 retirement flight to make room for the two new bombers at Carswell AFB, TX (Figure 1-1). However, even as the high-tech, first supersonic bomber to enter service, the B-58 remained in service for less than a decade. Both the B-52 and B-58 primary missions were replaced by intercontinental ballistic missiles in the Cold War, yet the B-58 lacked the changeability and evolvability of the B-52 that allowed it to remain in service for over five times as long and counting.

Derived from the stories of success like the B-52, the motivation of this thesis research is to further characterize and understand how systems manage this inevitable change and continue to deliver value to stakeholders while operating in environments completely different from those imagined during conceptual development. If we decide to heed H.G. Wells' advice in engineering systems, then we must better understand how to design systems to *adapt* rather than *perish*.

1.1.2 MIT SEAri

The Systems Engineering Advancement Research Initiative (SEAri) is a MIT research lab that is affiliated with both the Engineering Systems Division (ESD) and the Department of Aeronautics and Astronautics. Since ESD spans most departments within the School of Engineering, as well as the School of Science, the School of Humanities, Arts, and Social Sciences, and the Sloan School of Management, SEAri is uniquely positioned for interdisciplinary research in advancing systems engineering to meet contemporary challenges of complex socio-technical systems (Ross and Rhodes 2008a).

This thesis is a product of almost two years working as a resident research assistant within the SEAri group. Therefore, the discussion of motivation behind this particular thesis can be better understood within the context of the overall SEAri mission and motivations.

The early phases of system development and conceptual design require careful consideration as early decisions will have substantial influence on a new system, ultimately enabling or limiting success of the system over time. Figure 1-2 shows this classic relationship of costs, management leverage, and knowledge in complex system design. This research is aimed at manipulating these curves, extending managerial leverage and delaying committed costs further into development, while increasing knowledge gained earlier in design.

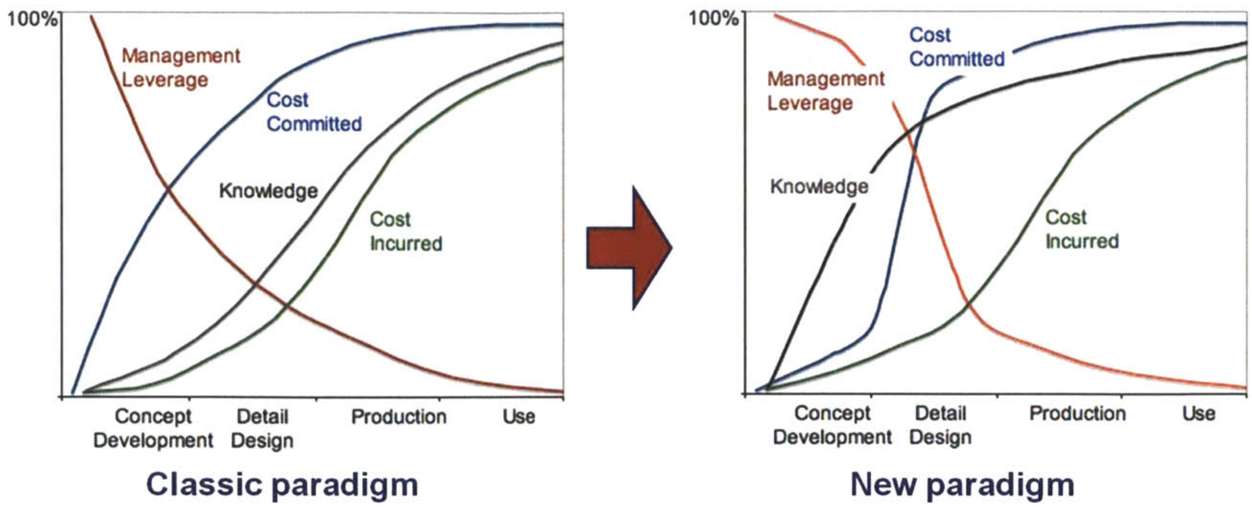


Figure 1-2. Desired Shift in Critical Front-End Complex System Design (Left side from Blanchard and Fabrycky (1998))

The International Council on Systems Engineering (INCOSE) defines systems engineering[1] as:

"...an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem."

MIT SEAri research focuses primarily on contributing to advanced systems engineering as opposed to traditional systems engineering as defined below:

> *Traditional Systems Engineering* is the process of selecting and synthesizing the application of the appropriate scientific and technical knowledge in order to translate system requirements into system design (Chase 1984).
>
> *Advanced Systems Engineering* is a branch of engineering that concentrates on design and application of the whole as distinct from the parts… looking at the problem in its entirety, taking into account all the facets and variables and relating the social to the technical aspects (Booton and Ramo 1984).

Systems engineering can be an effective discipline for improving efficiencies in design and leveraging decisions made early in development, but the underlying goal for systems engineering may be reduced simply to maximizing experienced, and therefore perceived, system success by

[1] Definition pulled from the INCOSE website at www.incose.org, accessed May 21, 2012

stakeholders. Part of the success of the system can be defined narrowly, in terms of minimizing costs, improving scheduling efficiencies, or meeting performance requirements, or it can be defined more broadly by maximizing the net benefit experienced by stakeholders through interactions with the system, while meeting or exceeding expectations (Ross and Rhodes 2008a).

If systems engineering aims at maximizing net benefits, value-centric thinking then is a crucial component of systems engineering. Understanding how people recognize value in the systems they use or care about is essential in realizing valuable systems (Hall 1962). The value one perceives from a system depends on the contexts and needs of that person. Since contexts, and therefore needs, are in constant flux, this presents a problem for systems engineers to be able to create a system that continues to deliver value despite this change. This idea of *value robustness* is a leading motivator for this research and will be discussed in more detail in Chapter 3.

Maintaining system performance in the presence of uncertainties in design and operating environments is both challenging and increasingly essential as system lifetimes grow longer. In response to perturbations brought on by these uncertainties, such as disturbances, context shifts, and shifting stakeholder needs, systems can continue to deliver value by being either robust or changeable. This research focuses more in the *change* spectrum of these strategies. Designing a *changeable* or *evolvable* system may reduce the long-term cost of system upgrades or replacements in the presence of these perturbations. A system that is designed to take advantage of effective design parameters from prior generations and carry them into future generations will be more responsive to changing environments. The effective design parameters are those design choices that lead to positive perceived value parameters, also referred to as "attributes" in prior generations of the system. Such 'evolvable' systems may help alleviate redesign costs and provide stakeholders with a more valuably robust system. In another approach, systems may be designed to maintain value-delivery by changing in response to these lifecycle perturbations while in operations.

Measuring a system's lifecycle properties, sometimes called "ilities", such as changeability, adaptability, flexibility, and survivability, gives stakeholders and decision makers an enhanced basis for differentiating between design alternatives. These lifecycle properties have been proposed as means to achieve system value sustainment in spite of changes in contexts or needs. Intentionally designing for these lifecycle properties is an active area of research, and no consensus has formed regarding how these and other "ilities" might trade off. In Bay et al. (2009), George Bernard Shaw is quoted, "England and America are two countries separated by a common language," showing contrast to an even bigger problem in systems engineering where we are separated by *separate* languages. Bay argues that to be an effective systems engineer, there needs to be a common language. The lack of common language in these often-vague properties (ilities) has motivated this research to better understand more than just *changeability* or *evolvability*, but ilities as a whole.

1.1.3 Research Questions

The systems we build today are operating with longer lifecycles in environments that are changing faster than in the past. If this research is to focus on system changes to enable value robustness, understanding how systems change from one state to another needs to be well characterized. This research attempts to analyze mechanisms that allow system changes to occur, and proposes a framework for allowing system designers to map vague, yet desirable, ilities to prescriptive system design principles. More specifically, this research will further the characterization of system changeability and evolvability and ultimately provide a structured and meaningful way of classifying system characteristics like "ilities". This research was guided by three key questions outlined below.

(1) What is "evolvability" in the context of engineering systems and systems of systems engineering (SoSE), and how does it relate to other ilities such as flexibility and adaptability?

(2) How can the properties and contexts of change mechanisms be used to classify a system's displayed ility types?

(3) How have ilities been implemented in historical systems/SoS, and how can we use this information to prescriptively include these desirable system properties in future designs?

This thesis will attempt to address these research questions in an effort to demonstrate why consideration of advanced ilities[2] within systems engineering is important in modern engineering. With systems that are increasing in cost, complexity, and life span, systems engineers will need more design tools and acuity to achieve the required performance.

1.2 Scope

The focus of this research was initially scoped to include only the "evolvability" of systems and systems of systems in systems engineering. However, a need for a broader scope research study in system changes and ilities became apparent as the research progressed. Kapurch (2007) talks about how NASA needs to develop a common language to effectively communicate across organizational lines, and the same is true when dealing with ilities across multiple domains within systems engineering. For this reason, this thesis, while focusing on changeability and evolvability, will also address relevant issues and insights in all ilities. A challenge in creating any useful guide is addressing the right amount of detail, researching for broad application, but not so general that no knowledge is added (Kapurch 2007). This thesis is intended to shed light on a newer, less defined evolvability ility, while also motivating a need for synergy among systems engineering research in the ever-growing area of ilities.

---

[2] For example flexibility, adaptability, changeability, and evolvability, as opposed to more traditional ilities such as reliability, maintainability, and availability.

1.3 Methodology

1.3.1 Research Approach

As this research began in an effort to define evolvability and further develop design principles that lead to evolvable systems, a research plan was developed to address the broad array of related literature and knowledge in a structured manner. Similar to the method used in Matthew Richards' dissertation on survivability (2009), this effort was to focus on both normative and descriptive means of characterizing evolvability design principles. This being a large task for a masters thesis, the normative and descriptive methods were separated into two thesis efforts. This thesis focuses on a descriptive approach to characterizing the nature of the problem, using historical cases and empirical investigations to identify key components of evolvability and design principles. The counterpart effort to this work is performed by MIT graduate student Dan Fulcoly and is expected to be published at the same time in 2012 as described in Beesemyer et al. (2011) and seen in Figure 1-3.

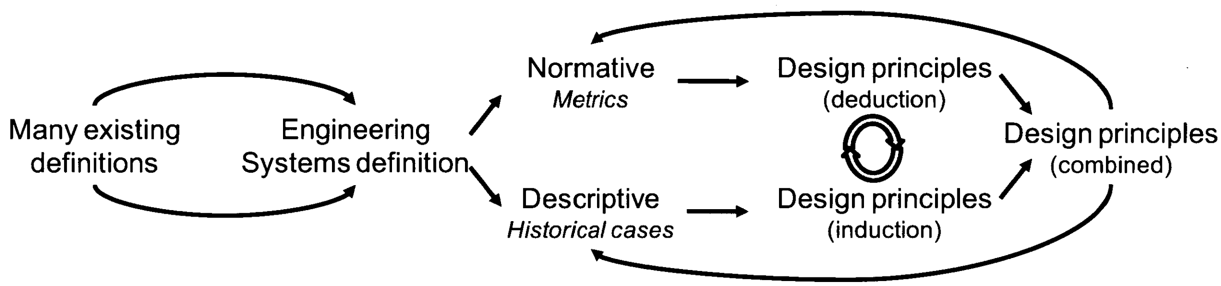


Figure 1-3. Descriptive and Normative Research Methodology

The normative approach focused on extracting relevant ideas from existing literature and looking for trends, which help form the basis of a theoretical model of design evolvability. The relevant ideas include definitions of evolvability, suggestions for when to design for evolvability, evolvability metrics, and principles for designing for evolvability. Several candidate metrics are explored with the intent of building on them to form a more comprehensive evolvability metric. Analyses of applications of this metric to selected case studies result in theory-based evolvability design principles.

In the descriptive approach, initial design principles of evolvability are derived using purported principles in literature and knowledge gathered from interviews with systems engineers. The validity of these design principles may be tested by inductively mapping the evolvable characteristics of existing systems to the set of preliminary design principles, similar to what was done in recent survivability design principle research (Richards 2008). Initial principles may be revised with the insight gained from these empirical test cases. If these test cases show patterns that cannot be mapped to the working design principle set or if certain principles are not found in practice, the working design principle set may be revised to reflect these insights.

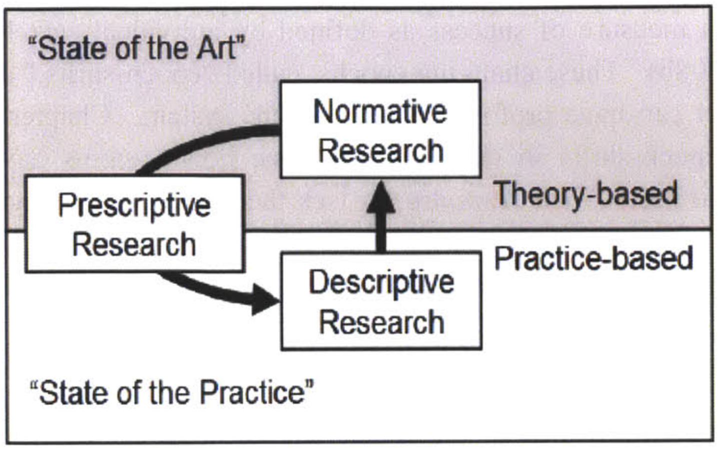


Figure 1-4. Underlying Structure of SEAri Research Program (Ross and Rhodes 2008a)

Moreover, this research is strategically placed in both theory-based and practice-based methods (Figure 1-4). While prescriptive methods seek to advance the state of the practice using the normative principles described, they must also be informed by practical limitations and constraints (Ross and Rhodes 2008a). This research in particular focuses on the "state of the practice" and informs the descriptive research approach with four distinct research thrusts: 1) a basis for defining system changes and ilities; 2) change database and categorical cluster analysis case research; 3) epoch-shift, impact, response, outcome case research; and finally 4) expert interviews case research.

1.3.2 Description of Descriptive Research Thrusts

The four research thrusts attack the research questions from multiple angles in an effort to draw together unique perspectives on this topic. Chapter 2 serves as a literature review of some of the most common system ilities considered in this research as well as applications to decision tools used in systems engineering to evaluate architectural concepts. In Chapter 3, more detail is given on value robustness and how changeability and evolvability support that objective. Chapter 3 is largely aimed at answering research question 1, in an effort to contextualize and characterize evolvability within the value robustness strategy of change. The following chapters, 4-7, detail research accomplished in the four research thrusts discussed earlier, with brief summaries below.

1.3.2.1 Epoch-Shift, Impact, Response, Outcome Case Research

In order to explore the impact early design decisions may have on lifecycle properties, called ilities, it is necessary take a broad perspective, looking at the environment in which a system operates. This environment, or operational context of the system, when combined with a set of stakeholder needs, is termed an "epoch" and characterizes a period of time during which the key exogenous factors that influence the ultimate success of a system are fixed. Since the goal of any system is to meet these needs in various contexts, delivering benefit at cost, or value, across changing epochs is a measure of success as defined by individual stakeholders of the system (Ross and Rhodes 2008b). These changing epochs, called "epoch-shifts," are outside the control of the system and yet can have profound effects on the system. Chapter 4 serves to dive into historical cases of epoch-shifts in an effort to frame how systems have been *impacted* by changing contexts and needs, their *response* (or lack thereof) to that change, and the *outcome* of their response. This chapter illustrates the need for this lifecycle value way of thinking, by looking into systems that have either weathered or succumbed to epoch-shifts.

1.3.2.2 Prescriptive Semantic Basis for Change-related Ilities

The concept of designing ilities into systems has grown in recent years as expressed by many strategic leaders in engineering and policy. As discussed earlier, there is a need for common understanding of these system properties if designers, engineers, and stakeholders are to ever have effective communication, especially in critical exercises like requirements-drafting. Tracing these desirable properties to verifiable system instantiations remains ambiguous at best (Ross, Beesemyer and Rhodes 2011). Chapter 5 details the need and development of a prescriptive semantic basis (sometimes abbreviated "basis") for identifying ilities in systems that change states. This section describes the semantic challenge in establishing a well-defined set of system ilities and proposes a basis for describing these characteristics without the need for agreeing on specific definitions. This section largely attempts to answer research question 2 in an effort to develop a framework for systems engineers to use in tracing desired ilities to verifiable system requirements and specifications.

1.3.2.3 Change Database and Categorical Cluster Analysis Case Research

In order to aid in research analysis of connections between change mechanisms and ilities, a database was created to hold the data for different change mechanisms in various systems experiencing a wide variety of changes. This database was created in an attempt to experiment with the semantic basis for identifying ilities in system changes, but matured to capture sufficient data about actual system changes in a structured manner that could be used for aggregate data mining as well. Chapter 6 outlines the creation and development of the database and the categorical clustering method used in analyzing the data. The purpose of this research was to begin to validate the semantic basis discussed in Chapter 5 while also shedding light upon and motivating further research into how populations of systems may have certain change strategies. Informed by Ross and Rhodes (2008b), these ilities and their conditions are being constantly refined and augmented to apply to any generic change for any system.

1.3.2.4 Expert Interviews Case Research

The final thrust of this research involved knowledge capture through communications with selected systems engineers, current and retired, who are known for their work in incorporating ilities into system design. To tie this research back to the "state of the practice," interviews were conducted with leading systems engineers to gain their insight into the problem of formulating verifiable ility requirements and methods of determining what ilities to include in design. Chapter 7 outlines the questionnaire and responses from systems engineers when asked about how they design systems and SoS so that they can change in order to continue delivering value in spite of perturbations. These insights serve to ground the other research described in this thesis to practical limitations, while gaining empirical knowledge in how "real-world" systems engineers have approached the same problems.

1.4 A Set of Empirical Studies for Ilities in SE

Overall, this research is motivated by the continuing push in systems engineering research for maximizing system value across system lifecycles. As systems become more complex and are forced to operate with longer lifecycles, we find ourselves as engineers being asked to do more with less. This creates a need to seek out new ways of thinking and new approaches to traditional systems engineering practices. The environments that systems operate in are often vastly different from the ones envisioned in conceptual design and are only changing faster and faster. Using descriptive and empirical approaches, this research intends to shed light on the effectiveness of making good ilities-based decisions early in conceptual development of systems. This research attempts to characterize less well-defined ilities like evolvability, while expounding on substantial existing research in ilities like changeability, flexibility, and survivability. Moreover, this thesis outlines research that makes the case for advanced practice for systems engineering and the much-needed synthesis of ideas, methods, and principles related to ilities in this fast-growing field.

2 Literature Overview

2.1 Basic Motivation

This chapter presents an overview of much of the relevant literature in systems engineering research that deals with ilities and decision tools. The review begins with research on a few selected ilities that are relevant to this thesis work. Following that is a discussion of methods in decision theory that systems engineers often use to aid in decision-making practices. This chapter lightly addresses a few decision methods and how those methods have been tied back to ilities. Lastly, there is a description of systems of systems and how they differ from traditional monolithic systems.

2.2 System Properties

2.2.1 Ilities in Literature

Non-traditional design criteria, such as flexibility, survivability or evolvability, are collectively referred to as the "ilities." The ilities are system properties that are increasingly recognized as qualities that lead to successful systems (McManus et al. 2007). With uncertain environments demanding "robust", "flexible", or "evolutionary" designs, there is room for growth in the methods to attain these desirable traits. In particular, the language involving systems ilities is plagued by poorly defined terminology (McManus and Hastings 2006).

The ilities show up often in literature, especially in work that deals with strategic level thinking. Strategic thinkers are focusing more on lifecycle properties, or ilities, in response to modern environmental pressures on systems engineering. As systems and systems of systems are operated in increasingly dynamic environments for longer life spans, systems engineers will require new design practices that promote value robustness in these environments. The practice of systems engineering must evolve to meet the requirements of increasingly dynamic operating environments (Rhodes and Hastings 2004). The idea of this "robustness" to future changes in technologies, contexts, and missions is a complex problem that lacks any comprehensive approach. As quoted in Ross, Rhodes, and Hastings (2008), Dr. Marvin Sambur, Assistant Secretary of the Air Force for Acquisition (at that time), defined "Robustness" as:

- Capable of adapting to changes in mission and requirements;
- Expandable/scalable, and designed to accommodate growth in capability;
- Able to reliably function given changes in threats and environment;
- Effectively/affordably sustainable over their lifecycle;
- Developed using products designed for use in various platforms/systems; and
- Easily modified to leverage new technologies.

This description of robustness includes other, separately defined ilities such as adaptability, scalability, reliability, affordability, sustainability, and modifiability. Accordingly , this can either be that robustness is a higher-level ility, encompassing many of the positive traits present in other defined ilities, or it can also be that the definition of robustness is unclear or ambiguous, or a mix of both.

An issue that must be addressed is that the ilities are mostly interpreted as positive characteristics a system may display, and therefore are commonly used to describe systems that may or may not have that quality. In Williams (2000), he argues that ilities should not be relegated to simple marketing schemes or hype, but researched and defined to produce a vocabulary for discussing performance potential amid dynamic environments. This relates to the McManus and Hastings (2006) work relating to systems providing value in spite of uncertainty. They introduce a framework that relates the problem of uncertainty to the desired outcome of ilities. In Figure 2-1, the proposed framework for handling uncertainties and their effects is broken into four categories: uncertainty, risk/opportunities, mitigations/exploitations, and outcomes.

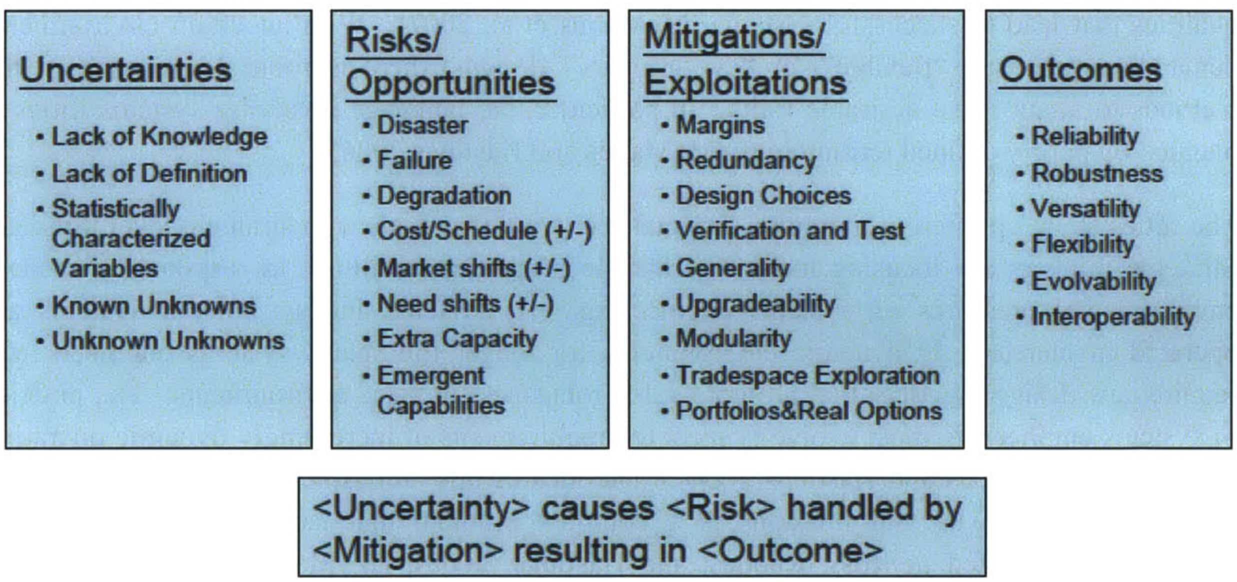


Figure 2-1. Framework for Handling Uncertainty (McManus and Hastings 2006)

This framework shows how there are many possible pathways in dealing with the numerous types of uncertainty (the various path permutations across the figure). Uncertainties in the world lead to consequences in the form of risk or opportunities. These consequences can be either mitigated or exploited through certain decisions or actions such as introducing margin, redundancy, verification and testing, modularity, open architecture, standardization, or decision analysis tools. Those decisions will relate to attributes of the system that characterize the interaction with uncertainty, like reliability, robustness, versatility, or evolvability (McManus and Hastings 2006). Often these decisions come with costs, which is where systems engineers often run into difficulties. System stakeholders often desire these outcomes, but may be unwilling to pay extra money for them, when they are unable to account for the full benefit of having these outcomes (Ross, Rhodes, and Hastings 2008).

If these ilities are desirable outcomes that will need to be verified and validated in future systems, there needs to be a discussion on what these words mean. Often words like "flexibility" or "agility" are used incongruently across the literature. For the purposes of this research, this thesis will use the definitions as detailed by SEAri, seen below in Table 2-1.

Table 2-1. MIT SEAri Definitions of Ilities

Value Robustness[1]	ability of a system to maintain value delivery in spite of changes in contexts or needs
Robustness[2,7]	ability of a system to maintain its level and set of specification parameters in the context of changing system external and internal forces
Changeability[2,7]	ability of a system to alter its form, and consequently possibly its function, or operations, at an acceptable level of resource expenditure
Flexibility[2,7]	ability of a system to be changed by a system-external change agent with intent
Adaptability[2,7]	ability of a system to be changed by a system-internal change agent with intent
Evolvability[3,7]	ability of a architecture to be inherited and changed across generations (over time)
Survivability[4,7]	ability of a system to minimize the impact of a finite duration disturbance on value delivery
Versatility[5,7]	ability of a system to satisfy diverse needs for the system without having to change form (measure of latent value)
Scalability[2,7]	ability of a system to change the current level of a system specification parameter
Modifiability[2,7]	ability of a system to change the current set of system specification parameters
Interoperability[5,7]	ability of a system to effectively interact with other systems
Reconfigurability[6,7]	ability of a system to change its configuration (component arrangement and links)
Agility[5,7]	ability of a system to change in a timely fashion
Extensibility[7]	ability of a system to accommodate new features after design

[1] From Ross and Rhodes (2008a)
[2] From Ross, Rhodes, and Hastings (2008)
[3] From Beesemyer, Fulcoly, Ross, and Rhodes (2011)
[4] From Richards (2009)
[5] From Ross, Beesemyer, and Rhodes (2011)
[6] From Ross and Rhodes (2011)
[7] From de Weck, Ross, and Rhodes (2012)

Having these ility definitions spelled out for this thesis is important, however for a few of the ilities there will be more discussion in this section on what the definitions mean and where they came from.

2.2.1.1 Value Robustness

Value robustness is the ability of a system to continue to deliver stakeholder value in the face of changing contexts and needs (Ross and Rhodes 2008a). In 2004, at the Air Force/Lean Aerospace Initiative Workshop on Systems Engineering for Robustness, leaders discussed the need for robust systems to be capable of adapting to change, reliably function when exposed to change, be easily modified to leverage emerging technology, scalable and adaptable (Rhodes, 2004). This need was further clarified and simplified in Ross (2006) as "to deliver value to stakeholders over time." There are many ways systems can achieve that goal—actively changing to meet new needs or passively remaining robust—but the important aspect to be noted is that value-robustness only really demands continued delivery of value.

Since value-robustness is an ility that applies at the overall system value level, it is common to see many other ilities thrown in with the same discussion. In the literature, we find discussions of robustness often paired with means to achieve it, such as adaptability or modifiability. For the purposes of this research, value-robustness is the ultimate goal of most systems engineers in the presence of epoch shifts. It is the motivating principle to changing the way we architect systems, and therefore encompasses many of the strategies in literature of achieving that aim, including passive and active means. While Ross (2006) described this ility as the "top" ility, or the over-arching goal in all system cases, this thesis holds value robustness as one of two means to achieve a more encompassing value aim, termed *value sustainment.* This research differentiates between value robustness and *value survivability* as means to achieve value sustainment in the presence of epoch shifts and disturbance respectively. More about value robustness and how it relates to tradespace exploration, value survivability, and value sustainment is discussed in more detail in Chapter 3.

2.2.1.2 Changeability

Changeability encompasses the active strategies of value-robustness. Systems can both attempt to passively weather changes in needs or context, or they can actively *change* in response to improve value delivery. These changes come in many flavors: flexible, adaptable, scalable, and modifiable (Ross, Rhodes, and Hastings 2008). Changeability then, is a high-level ility that contains the various ways a system can change states. In this way, changeability can be leveraged to maintain value delivery over a lifecycle, or value robustness (Ross 2006).

Fricke and Schulz (2005) characterize changeability as the "ability of a system to change easily," and can be broken down into four categories: robustness, agility, adaptability, and flexibility (Figure 2-4). Robustness is the ability of the system to continue delivering value in a changing environment. Agility is the ability to change rapidly. Flexibility and adaptability both deal with the ability of a system to change as differentiated by an external change agent (flexibility) or an internal change agent (adaptability).

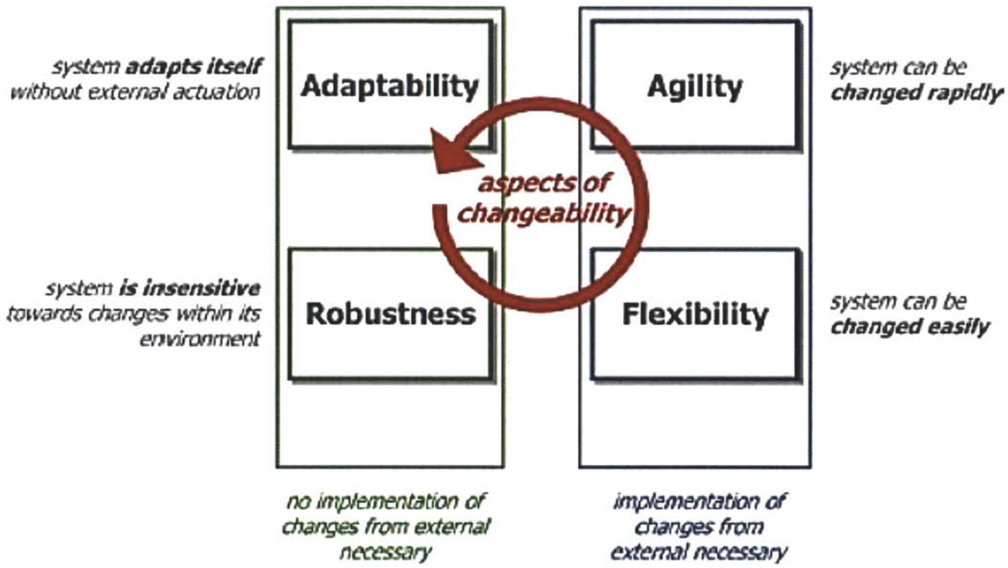


Figure 2-2. Changeability as Four Ilities (Fricke and Schulz 2005)

In the domain of manufacturing, ElMaraghy and Wiendahl (2009) define changeability as an umbrella concept that encompasses many change enablers at various levels of an industrial company throughout the lifecycle of the manufacturing system. Wiendahl goes further to describe types of changeability at different levels of manufacturing as seen in Figure 2-3.

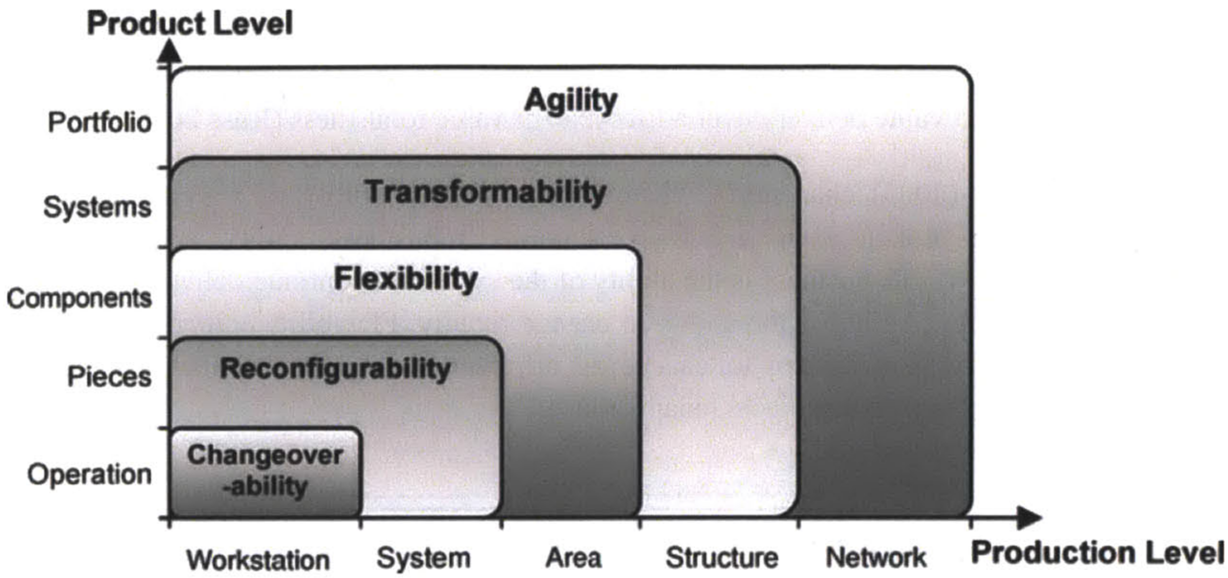


Figure 2-3. Types of Manufacturing Systems Changeability (Wiendahl and Heger 2003; Wiendahl 2005)

Algeddawy and ElMaraghy (2009) define changeability as classes, such as agility, flexibility, and reconfigurability introduced to manufacturing to increase adaptability to varying market conditions, competition, and rapid product changes and increases customization.

This research primarily deals with this strategy for achieving value robustness. There is a lot of literature dealing with robust systems performing under unexpected and expected adverse environments (McManus and Hastings 2006); however, this research scope will focus on systems that change. Robust systems, particularly in the space industry, are commonly designed to deal with known adverse environments like radiation, heat, or launch vibrations. Similarly, changeable systems can be designed for changes to expected shifts, or more difficultly, to unexpected shifts in needs or context.

2.2.1.3 Flexibility

Flexibility has been defined both broadly and acutely in the literature, but generally focuses on the ease by which systems are changed.

One of the more broad interpretations of flexibility comes from de Neufville and Scholtes (2011). In their book, flexibility is a strategy for managing uncertainty. This flexibility aims to take advantage and profit from possible upside opportunities while avoiding possible downside risks. They give a four-step process for developing design flexibility:

1. Recognize the major uncertainties in the system or environment.

2. Identify the specific parts of the system that provide the kind of flexibility best suited for dealing with these uncertainties.

3. Evaluate alternative flexible designs, and incorporate the best.

4. Plan for eventual implementation of flexibilities.

The flexibility in de Neufville and Scholtes (2011) uses similar thinking to that of financial options, where system may have the options, not obligations, to expand or reduce capacity as future uncertainties unfold. This is relevant to the staged deployments of communication satellites discussed in de Weck, de Neufville, and Chaize (2004).

In manufacturing systems, ElMaraghy (2006) defines Flexible Manufacturing Systems (FMS) as those designed for anticipated changes or variations built-in a priori. He goes on to say that the main types of flexibility are related to the two types of reconfigurability: hard (physical) and soft (logical). Other manufacturing definitions of flexibility are based on the notion of adaptability to uncertainties (Mandelbaum 1978; Slack 1987) or the capacity of a system to change and assume different positions or states in response to changing requirements with little penalty in time, effort, cost, or performance (Toni and Tonchia 1998).

Fricke and Schulz (2005) characterize flexibility as a "system's ability to be changed easily…to cope with changing environments."

Viscito and Ross (2009) define flexibility as a "dynamic property of a system that allows it to take advantage of emergent opportunity and to mitigate risk by enabling the system to respond to changing contexts in order to retain or increase usefulness to system stakeholders over time."

Ross (2008) writes that flexible systems are those that can be changed by an external change agent, or actor that causes the actual change.

McManus and Hastings (2006) describe flexibility as the ability of a system to be modified to do jobs not originally included in the requirements definition, and go on to say the modification can take place in the design, production, or operation of the system.

Butterfield et al. (2008) say "Flexibility is the ability of the architecture to handle changes, new features, or new knowledge late in the development cycle or after operational implementation is mature without significant consequences for costs, schedule and technical achievability."

A framework for defining flexibility in space systems is proposed in Nilchiani and Hastings (2007) and has six necessary elements: the uncertainty, time window of change, system boundary, response to change, the system aspect to which flexibility is applied, and access to the system.

Saleh et al. (2009) gives a multi-disciplinary literature review on the use of flexibility and its meanings in different domains. He proposes, after looking over much of the literature on flexibility, that it is "a property of a system that allows it to respond to changes in its initial objectives and requirements – both in terms of capabilities or attributes – occurring after the system has been fielded, in a timely and cost-effective way." This definition joins the idea of value to flexibility, saying that it is the ability to change in a "timely" and "cost-effective" manner, where other definitions, like Ross (2006), separated the value statement from the definition.

This research has focused more on the Fricke and Schulz and Ross definitions, where flexibility is ability of a system to be changed by an external change agent with intent, where value is not a part of the definition. Since value is often a subjective reference (e.g. "cheaper than X," or "faster than Y"), having definitions of ilities without value statements attached can be more general. This then, allows for explicit consideration of differences in perception between stakeholders. A change considered "valuably flexible" by one stakeholder may not be considered "valuably flexible" by another stakeholder.

2.2.1.4 Adaptability

Adaptability is very similar to flexibility but the change agent is internal as opposed to external or in other words, the change is self-motivated. Ross (2006) explains this subtlety in his thesis:

> *It is important to note that the only difference between flexibility and adaptability is the location of the change agent with respect to the system boundary: inside (adaptable) or outside (flexible). Of course, the system boundary could be redefined, changing a flexible change into an adaptable one, or vice versa. The fungible nature of the definition is often reflected in colloquial usage and sometimes results in confusion. If the system boundary and location of change agent are well defined, confusion will be minimized* (Ross 2006, p.108).

Incorporating a change agent inside of the system boundary results in a key additional consideration for system designers: the change agent must have the ability to sense, decide, and act to put the desired change in motion. For flexible-type changes, this change agent ability is outside of the system boundary and therefore not within the scope of system design in the same way it is for incorporating adaptable-type changes.

Fricke and Schulz (2005) define adaptability as "a system's ability to adapt itself towards changing environments… [they] deliver their intended functionality under varying operating conditions through changing themselves."

In this way, adaptability and flexibility are two ways of designing changeable systems under the umbrella of changeability.

2.2.1.5 Evolvability

Evolvability has significantly less literature in systems engineering than flexibility. Evolvability generally is associated with biological references that imply longer, generational changes. So starting with the biological interpretation, Colegrave and Collins (2008) define evolvability as the "ability of a population to both generate and use genetic variation to respond to natural selection" and Wagner (2008) says a system is evolvable if "mutations in it can produce heritable phenotypic variation."

In bringing the biological comparison into engineering systems, there is literature that tries to draw parallels between biology and systems engineering. Sussman (2007) says that robust and evolvable systems are those that "accommodate adaptive variation in some locales without changing the behavior of subsystems in other locales". He goes on to explain how good systems that are modular can be changed or adapted to new situations with only minor modification.

McManus and Hastings (2006) describe evolvability as the "ability of the system to serve as the basis of new systems (or at least generations of the current system) to meet new needs and/or attain new capability levels."

In systems of systems architecture literature, Butterfield et al. (2008) defines architecture evolvability as the ability of the architecture to handle future upgrades." This research goes on to say that evolutionary systems require "a process to plan, define, and prepare for program spirals" by identifying key technologies, processes, and attributes that will be required to support future capabilities. In this way, Butterfield et al. seems to suggest that evolvability is more about planning in future spirals/re-architecting, whereas flexibility is more about responding to random or unplanned changes, during development or operations. It is also worth mentioning that when Butterfield et al. talks about these "adaptations" he says that a system should be able to change in a cost-effective manner, thus separating the ability to change from the valuation of that change. Therefore you may be able to adapt, or evolve, but at a cost to the system.

Saleh et al. (2009) talks about evolving systems after they have been fielded, when the "system's technology base evolves on time scales considerably shorter the system's design lifetime". Flexibility is achieved when design considerations for evolution are implemented in the initial design.

This research focuses on the idea that evolvability refers to the generational changes made at the architectural level similar to Butterfield et al (2008).

2.2.2 Decision Analysis

Designers may understand what ilities or qualities they desire out of a system, but in order to find design concepts that actually achieve these goals is another problem entirely. There are many methods that designers can use to aid in this decision making process. This thesis focuses on a few of these methods to give context to where a lot of this ility research originated. McManus et al. (2007) talks about bringing the idea of ilities into previously developed decision-making tools like tradespace exploration (TSE). This task requires well-defined ilities and evaluation methodologies—like real options for flexibility—in order to be an effective tool for decision makers. While this research did not focus on producing metrics for ility evaluation, it does help to clarify meaning in the ilities, which is an important step in evaluating such non-traditional design criteria.

2.2.2.1 Utility theory

Utility theory is a framework that can be used by decision makers to help quantify the idea of value. In a world with limited resources, decisions must be made on the direction of system design. Decision makers are those who have significant influence on the funding or the requirements of the system. Utility theory is based on maximizing the value of a system with respect to a decision maker's or makers' objectives. Each decision maker may have multiple objectives that can be broken into a set of attributes. Attributes are metrics that measure how well an objective is met. Each attribute will have a definition, units, and range of accepted values, and requires careful consideration between both the designer and decision maker up front (Ross 2003).

If the ilities that this thesis discusses somehow provide extra value, a burden that systems engineering research has is to find ways to quantify that value. Utility theory can greatly aid the decision making process and has been used extensively in economic analyses. Von Neumann and Morgenstern (1947) formalized rules for utility theory in economics as a market demand metric. Ross (2003) unites utility theory from economics and psychology to capture the idea of "value" and applies it to space system design.

Moving on from single attribute utilities, Keeney and Raiffa (1993) introduce the multi-attribute utility function. This allows decision makers to address multiple attributes that may go into a decision, but it also presents a problem with comparing costs and benefits between attributes.

Prospect theory was introduced as a descriptive alternative to utility theory (Kahneman and Tversky 1979) merging knowledge from psychology and economics. Prospect theory decisions are based on potential values of losses or gains and are descriptive to how people generally act. Ross (2003) discusses the differences between these descriptive theories and the prescriptive, or normative, theories. Temporal aspects of value are discussed in Kahneman and Tversky (2000), arguing that "experienced" utility will be remembered and influence one's decisions on utility in the future.

Dyer et al. (1992) talks about the difference between multi-attribute utility theory (MAUT) and multi-criteria decision-making (MCDM), namely that MCDM uses a value function under certainty whereas MAUT uses a value function under uncertainty. Their research deals with assessing decisions in ambiguous environments where options can change over time. Dyer et al. (1992) briefly describe the two methods:

> *In its most basic form. MCDM assumes that a decision maker is to choose among a set of alternatives whose objective function values or attributes are known with certainty. Many problems in MCDM are formulated as multiple objective linear. integer, or nonlinear mathematical programming problems, and many of the procedures proposed for their solution are interactive.*
>
> *MAUT is sometimes subsumed under MCDM, but is usually treated separately when risks or uncertainties have a significant role in the definition and assessment of alternatives. It focuses on the structure of multicriteria or multiattribute alternatives, usually in the presence of risk or uncertainty, and on methodologies for assessing individuals' values and subjective probabilities. MAUT embraces both a large body of mathematical theory for utility models and a wide range of practical assessment techniques that pay attention to limited abilities of assessors. Information obtained from assessment usually feeds into the parent problem to rank alternatives, make a choice, or otherwise clarify a situation for the decision maker. Sensitivity analysis is often involved in the assessment and choice processes.* (Dyer at al. 1992, pp. 647)

2.2.3 Trade Studies

Conceptual design may often be plagued with many decisions or designs choices for a system to achieve mission requirements in different ways. If one is trying to optimize in cost and utility, choosing between these different concepts can be difficult and nearly impossible for the unaided human, without a method or tool. Trade studies act as a tool for decision makers to evaluate the utility of various concepts before proceeding with a specific design.

Trade studies apply some form of decision analysis (e.g. utility theory) to engineering conceptual design. This allows system designers to make cost-benefit tradeoffs early in concept selection (Keeney and Raiffa 1993).

In Space Mission Analysis and Design (SMAD) (Wertz and Larson 1991), system trades consist of "analyzing and selecting key parameters, called system drivers, which determine mission performance," where the key trades are those that determine system size, cost, and risk. SMAD goes on to describe a five-step process to frame a system trade:

1. Select trade parameter
2. Identify factors which affect the parameter or are affected by it
3. Assess impact of each factor
4. Document and summarize results
5. Select parameter value and possible range

This process is good at looking at particular concepts, but lacks effectiveness when trying to compare across vastly different concept groups (Ross 2003).

The NASA Systems Engineering Handbook (NASA 2007) discusses the need for trade studies to be open and inclusive, with individuals of different skills working together to effectively manage the subjectivity inherent in trade formulation. The trade study, in simple form, is depicted in Figure 2-4.

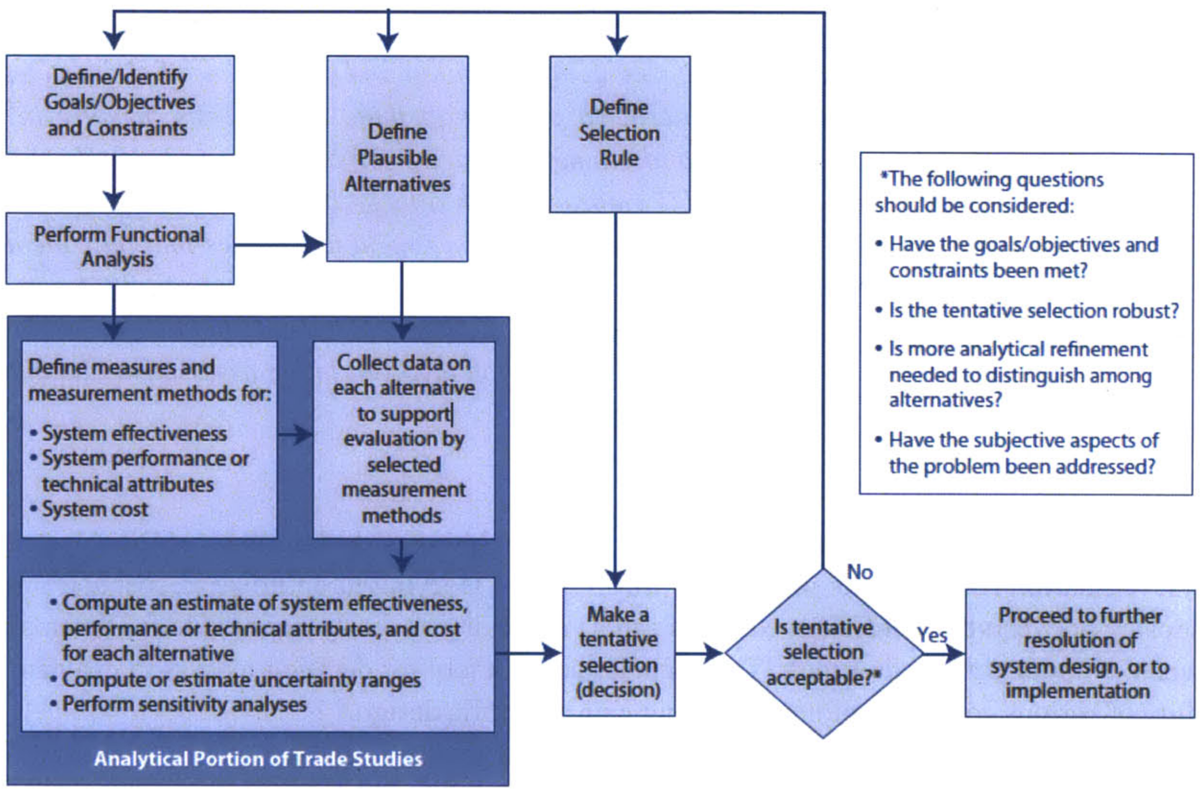


Figure 2-4. NASA Trade Study Process (NASA 2007)

The NASA method attempts to view concept-independent solutions, without concentrating on any point trades (Ross 2003).

Tradespace exploration is the expansion and use of trade studies to evaluate larger sets of conceptual designs in an efficient manner (Ross et al. 2004). Using computer-based models and simulations, stakeholders can evaluate thousands of potential designs efficiently (McManus, Hastings and Warmkessel 2004). This approach will avoid local point solution trades and look at the broader relationships between potential concepts and stakeholder preferences. Tradespace exploration, then, is an effective tool to evaluate the costs, benefits, and risks of systems concepts early in the design process.

Ross and Hastings (2005) discuss the risk of settling into a solution or set of solutions too early in tradespace exploration. This premature focusing may introduce issues with finding the globally "best" solution and can degrade the quality of concept selection if not considered. The four classes of trade considerations are 1) Local point solutions, 2) Frontier subset solutions, 3) Frontier solution sets, and 4) full tradespace exploration and can be seen Figure 2-5.

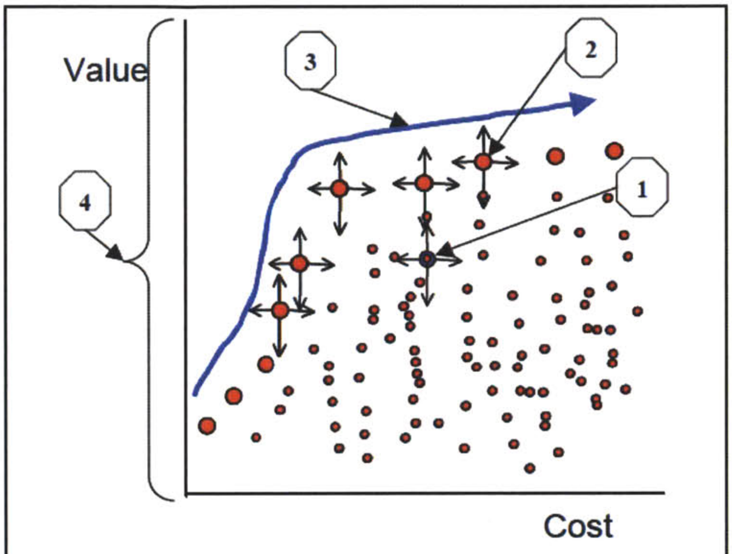


Figure 1. Types of trades: 1) local points, 2) frontier points, 3) frontier sets, 4) full tradespace exploration

Figure 2-5. Types of Trades in Tradespace Exploration (Ross and Hastings 2005)

Understanding all of the types of trades is important when exploring all of the concept options. The Pareto Set of designs is the set of designs that cannot increase utility without increasing costs, and are therefore efficient designs. One may find a solution using only Pareto methods but may miss out on knowledge gained by exploring designs that are not Pareto optimal. Jilla (2002) discusses how the Pareto designs form a "Pareto Front" within the tradespace and represent the most efficient potential concepts.

Multi-Attribute Tradespace Design (MATE) furthers a conceptual design methodology that unites decision analysis, tradespace exploration, and model-based design to provide a decision-making tool for stakeholders (Ross and Hastings et al. 2004). In Ross (2003), a 48-step description of the MATE process is given and can be summarized visually in Figure 2-6.

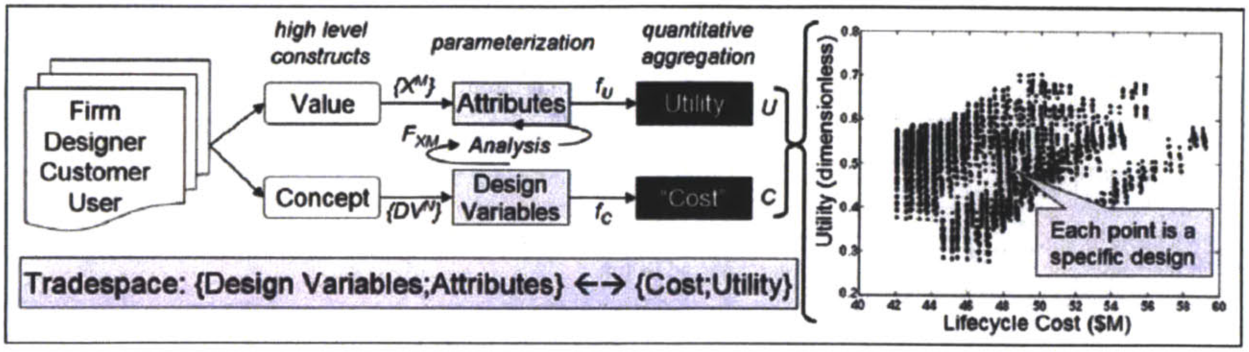


Figure 2-6. Multi-Attribute Tradespace Exploration (MATE) (Ross and Hastings et al. 2004)

The MATE process can be decomposed at a high level to three phases: 1) need identification, 2) design alternative enumeration, and 3) design alternative evaluation (Ross and Hastings et al. 2004). The process begins in eliciting the real mission needs and preferences and proceeds to selecting attributes and design variables under consideration then evaluates all of the concepts lifecycle costs and utilities.

Incorporating ilities into this tradespace exploration may allow for better discrimination between different conceptual designs (McManus et al., 2007). However, ilities are difficult to incorporate in a classic tradespace with two key challenges proposed by Richards (2009): 1) representation of temporal properties in a static construct and 2) axiomatic restrictions on the incorporation of the "ilities" in attribute sets (attributes need to be perceived as independent, yet the ilities are defined by attribute performance over time).

Ross et al. (2010) furthers this research to include dynamic MATE. This tradespace exploration analysis takes into account changes in the tradespace due to changes in needs or context to help evaluate design concepts in a dynamic manner. In Ross et al. (2010), high-level decision makers can frame their tradespace exploration with the following six questions:

1. Can we find good value designs?
2. What are the strengths and weaknesses of selected designs?
3. Are lower cost designs feasible? What compromises are needed to lower costs?
4. What about time and change?
5. What about uncertainty?
6. How can detailed design development be initiated in ways that maximize the chance of program success?

Metrics can be imposed on tradespaces in order to better understand designs that have desirable dynamic MATE aspects, like being on the Pareto front in many different epochs for example. Ross and Hastings (2006) develop the idea of tradespace networks (Figure 2-7). When decision makers specify transition rules or paths (dynamic change opportunities), a tradespace may become a tradespace network. Using these tradespace networks, one can track the changeability of a design by counting the outgoing arcs from a design. This metric from Ross (2006), known as Filtered Outdegree, is an example of how one may use a tradespace to evaluate the ility performance (in this case "changeability") of a design.

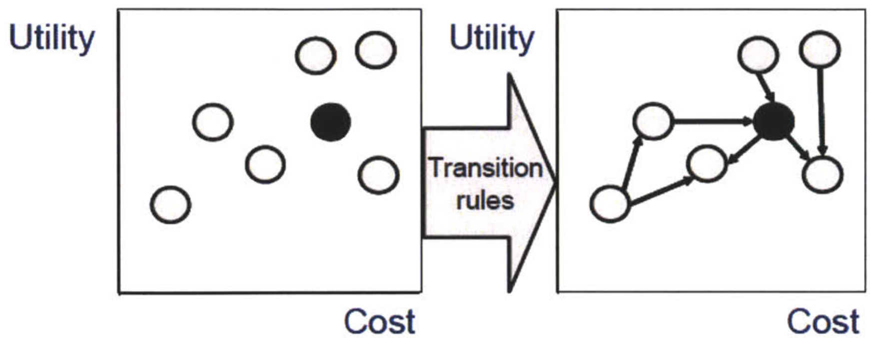


Figure 2-7. The Tradespace Network (Ross 2006)

More research in temporal aspects of tradespace exploration has been accomplished with Epoch-Era Analysis and is discussed in Chapters 3 and 4 of this thesis.

2.3 Systems of Systems

While systems of systems are increasing in recognition and importance, what constitutes a system of systems has been unclear (Maier 1998). This literature review of systems of systems begins with a simpler, yet important, definition of a system. A system, as defined by the International Council on Systems Engineering (INCOSE), is:

> "*A combination of interacting elements organized to achieve one or more stated purposes,*" (INCOSE 2006).

If there is a clear definition of a system, one would assume the definition of a system of systems would logically follow as a "combination of interacting systems organized to achieve a stated goal." However, this definition does not capture some of the needed components of more accepted definitions of a system of systems (SoS). There may be large, complex and distributed systems that seem like SoS, but do not meet all of the requirements of a SoS classification described in this section. For example, a computer could be considered a "system of systems" in that it is a system that includes other systems like a monitor, a hard drive, a processor, and so forth, but it is not a true SoS (Maier 1998). The concept of SoS was introduced to describe a particular class of system that is more than just simply applying the definition of "system" to a system. This class of system has unique characteristics that come about when one combines interacting systems in a way to achieve additional functionality.

2.3.1 Systems of Systems Overview

Maier (1998) proposes five key characteristics of systems of systems (SoS), which distinguish them from traditional systems:

1. *Operational Independence of the Elements*: If the system of systems is disassembled into its component systems, the component systems must be able to usefully operate independently. The system of systems is composed of systems that are independent and useful in their own right.

2. *Managerial Independence of the Elements*: The component systems not only can operate independently, they do operate independently. The component systems are separately acquired and integrated, and maintain a continuing operational existence independent of the system-of- systems.

3. *Evolutionary Development*: The system of systems does not appear fully formed. Its development and existence is evolutionary with functions and purposes added, removed, and modified with experience, over time.

4. *Emergent Behavior*: The system of systems performs functions and carries out purposes that do not reside in any component system. These behaviors are emergent properties of the entire system of systems and cannot be localized to any component system. The principal purposes of the systems of systems are fulfilled by these behaviors.

5. *Geographic Distribution*: The geographic extent of the component systems is large. Large is a nebulous and relative concept as communication capabilities increase, but at a minimum it means that the components can readily exchange only information and not substantial quantities of mass or energy (Maier 1998).

The Maier (1998) characteristics for systems of systems are the guiding characteristics for classification that this thesis will use. Mekdeci et al. (2011b) describes additional characteristics of SoS that distinguish them from normal "monolithic" systems (Jamshidi 2009) derived from other research literature: Abstruse Emergence, Distributed Authority (Boardman and Sauser 2006), Multi-Functionality (Eisner, Marciniak, and McMillan 1991), Increased Contextual Diversity, Decreased System Awareness, and Dubious Validation (Ellison and Woody 2007).

Systems of systems engineering (SoSE) is the practice of systems engineering applied to the special case of systems that are classified as SoS. This is a relatively new field in systems engineering (SE) and has several key differences from normal SE to SoSE. Eisner et al. (1991) summarizes seven of these differences in Table 2-2 below.

Table 2-2. Seven Differences in SE and SoSE (Eisner et al. 1991)

	SoS Engineering	Traditional SE
1	There are several independently acquired systems, each under a nominal systems engineering process.	Subsystems are acquired under centralized control.
2	Overall management control over the autonomously managed systems is viewed as mandatory.	The program manager has almost complete autonomy.
3	The time phasing between systems is arbitrary and not contractually related.	Subsystem timing is planned and controlled.
4	The system couplings can be considered neither totally dependent nor independent, but rather interdependent.	Subsystems are coupled and inter-operating.
5	The individual systems tend to be uni-functional and the systems of systems multi-functional.	The system is rather uni-functional.
6	The optimization of each system does not guarantee the optimization of the overall system of systems.	Trade-offs are formally carried out in an attempt to achieve optimal performance.
7	The combined operation of the systems constitutes and represents the satisfaction of an overall coherent mission.	The system largely satisfies a single mission.

Maier (1998) explains even further and clarifies three different classes of systems of systems in order to distinguish various aspects of management:

1. ***Directed:*** *Directed SoS are those in which the integrated system of systems is built and managed to fulfill specific purposes. It is centrally managed during long-term operation to continue to fulfill those purposes, and any new ones the system owners may wish to address. The component systems maintain an ability to operate independently, but their normal operational mode is subordinated to the centrally managed purpose. For example, an integrated air defense network is usually centrally managed to defend a region against enemy systems, although its component systems may operate independently.*

2. ***Collaborative:*** *Collaborative SoS are distinct from directed SoS in that the central management organization does not have coercive power to run the SoS. The component systems must, more or less, voluntarily collaborate to fulfill the agreed upon central purposes. The Internet is a collaborative SoS. The Internet Engineering Task Force works out standards, but has no power to enforce them. Agreements among the central players on service provision and rejection provide what enforcement mechanism there is to maintain standards. The Internet began*

as a directed SoS, controlled by the US Advanced Research Projects Agency, to share computer resources. Over time, it has evolved from central control through unplanned collaborative mechanisms.

3. ***Virtual:*** *Virtual SoS lack a central management authority. Indeed, they lack a centrally agreed upon purpose for the system of systems. Large-scale behavior emerges, and may be desirable, but the supersystem must rely upon relatively invisible mechanisms to maintain it.* (Maier 1998, pp.3-5)

Another class of systems of systems was identified by Dahmann and Baldwin (2008), which marries certain aspects of Maier's Collaborative and Directed SoS classes:

4. ***Acknowledged:*** *Acknowledged SoS have recognized objectives, a designated manager and resources for the SoS, however, the constituent systems retain their independent ownership, objectives, funding, and development and sustainment approaches. Changes in the systems are based on collaboration between the SoS and the system.* (Dahmann and Baldwin 2008, p. 2)

Acknowledged SoS have a central authority similar to directed SoS, however this authority is minimal as the constituents retain their own budgets, decision-making and objectives. The behavior between the constituents and the central authority is collaborative. The Army's future combat system (FCS) is an example of an acknowledged SoS—with a program office tasked to create a lighter, more integrated force, but many of the individual FCS components being acquired through independent acquisitions or evolved from legacy systems.

These SoS classes represent the types of SoS that can be found in the world and all have distinct managerial aspects. For this reason, there are many facets to SoSE that make finding general design heuristics or design principles difficult. What may work for one class of SoS, may not work for another class. Systems of Systems are a type of system with particular characteristics and these characteristics may lead to unique challenges.

2.3.2 Scope and Applicability

The scope of this thesis will focus primarily on traditional systems engineering (SE). However, a goal of this thesis is to apply the knowledge gained from traditional systems engineering research to systems of systems engineering. The classes of systems of systems that this thesis will focus on will be Directed and Acknowledged systems of systems. These SoS classes generally have more centralized control and subordinated infrastructure. In this way, the goals of the system of systems may parallel traditional systems more closely. The military systems under examination in this thesis most often fall into the Directed SoS type. In addition, the types of SoS that would benefit most from this area of research are those with some degree of control over the architecture of the SoS. Therefore, the classes of SoS that have more centralized management will be the focus.

Much of the system of systems perspective is addressed in the discussion section of this thesis in Chapter 8.

2.4 Chapter Summary

The topics that have been reviewed and presented in this chapter make up the majority of the necessary knowledge to understand the methods, concepts, and figures/tables in this thesis. When new concepts arise in the body of this thesis, which may have not been fully covered in the literature review chapter, there will be added clarification and outside source review in the respective chapter. This is in an attempt to provide the knowledge and research in the most clear and logical progression with respect to the research described in this thesis.

3 Value Sustainment

The purpose of any system is to provide some level of value to the stakeholders of that system. A watch may provide its user value by keeping the time or adding some aesthetic component to an outfit. A more complex system like a high performance jet fighter may provide value through exerting force on an enemy or serving as a deterrent threat. The interest of the stakeholder is that these systems *keep* providing value throughout their expected lifetime. The expectations placed on value delivery can be very different between systems—a watch may be expected to last a year whereas a jet fighter may be expected to fly for a half a century. Stakeholders also have some level of expectation on system performance when exposed to perturbations. Mekdeci (2012) defines a *perturbation* as any "unintended state change in a system's form, operations, or context which could jeopardize value delivery." This definition should also include any change in needs of the system as well. A watch owner may expect that the watch should function with average wear-and-tear, bumps, or shocks. Similarly, a fighter jet would be expected to handle perturbations like jamming or thunderstorms as well. If system stakeholders desire better performance when subjected to perturbations, they generally pay for extra capability. A watch owner might demand that the watch perform underwater, or the Air Force might demand that the fighter be undetectable by radar; these requirements will add cost to the systems. Perturbations often come in unpredictable frequencies at varying levels of intensity, and system designers attempt to make systems that can provide value in spite of these perturbations.

Ross (2006) describes the concept of design exploration as a mapping and movement in both a "real space" and a "perceived space." His real space describes the space of potential actual systems as it represents possible tangible implementations. The perceived space is where stakeholders perceive value as it is interpreted through the experiences of its stakeholders. An engineer can make changes in the real space to affect perceived value in the perceived space. This relationship, in the context of potential design alternative enumeration and evaluation can be seen below in Figure 3-1.

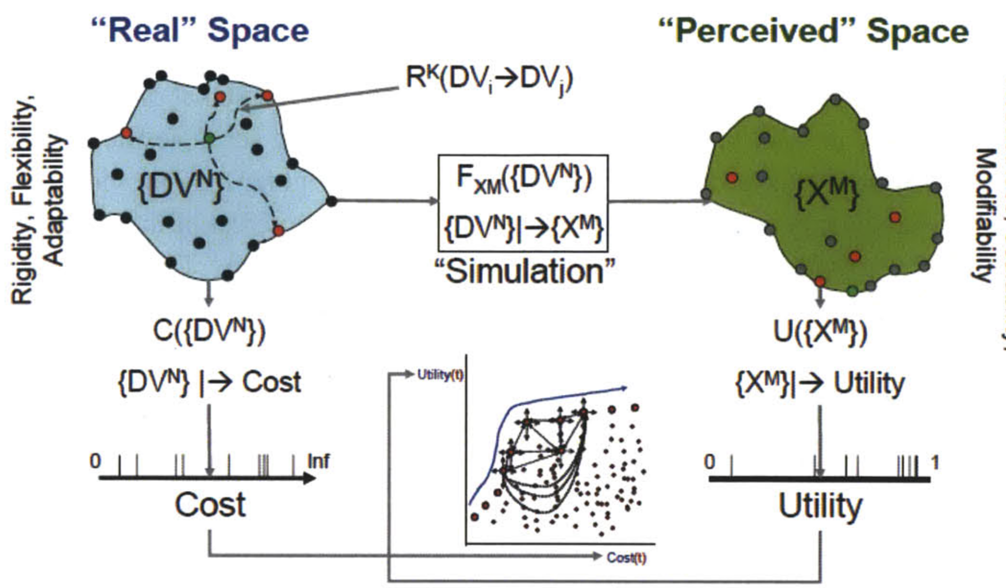


Figure 3-1. MATE Key Functions and Variable Mappings (Ross 2006)

Figure 3-1 outlines the key functions used in in Multi-Attribute Tradespace Exploration (MATE). The real space is comprised of the possible designs, which are combinations of all the design variables (DV). These design variables are related to costs [C(DV)], which together form the total cost of the system (C). The design variables also have some relationship with performance metrics [F(DV)]. Some of the performance metrics might be attributes of interest from decision makers (X) that relate to some perceived value or utility [U(X)]. The functions are not as important as the relationships drawn between the real space and the perceived space. Perturbations can take place in both spaces and have different impacts on the system depending on the type of perturbation.

This thesis carries a similar construct, but is somewhat different (Figure 3-2). Keeping the idea of "real" and "perceived" spaces, this construct expands on the various sub-spaces and functions within them. This real space describes not only the potential tangible systems but also the context and the performance metrics of the design as well. The real space then can be sub-divided into a function and three components: 1) the design space, 2) the context space, and 3) the performance space. The design space is all the possible combinations of the design variables (DV) where the context is the space of exogenous factors describing the environment within which the system may operate. The design space and context space may be enumerated and evaluated [X(DV)] to yield some performance (X) in the performance space. From a system architect perspective, this can be done through modeling and simulation, and is therefore described as an evaluation. In the real world however, this is how the system actually operates in the given context. Then there is the perceived space, or where value is perceived by system stakeholders. The perceived space can by sub-divided into a function and two components: needs space and the value space. Decision makers and other stakeholders have needs for the system. Decision makers (sometimes at the behest of other stakeholders) articulate their needs into a set of preferences on specified performance metrics [V(X)], that ultimately relate the performance space and needs space to some value in the value space.

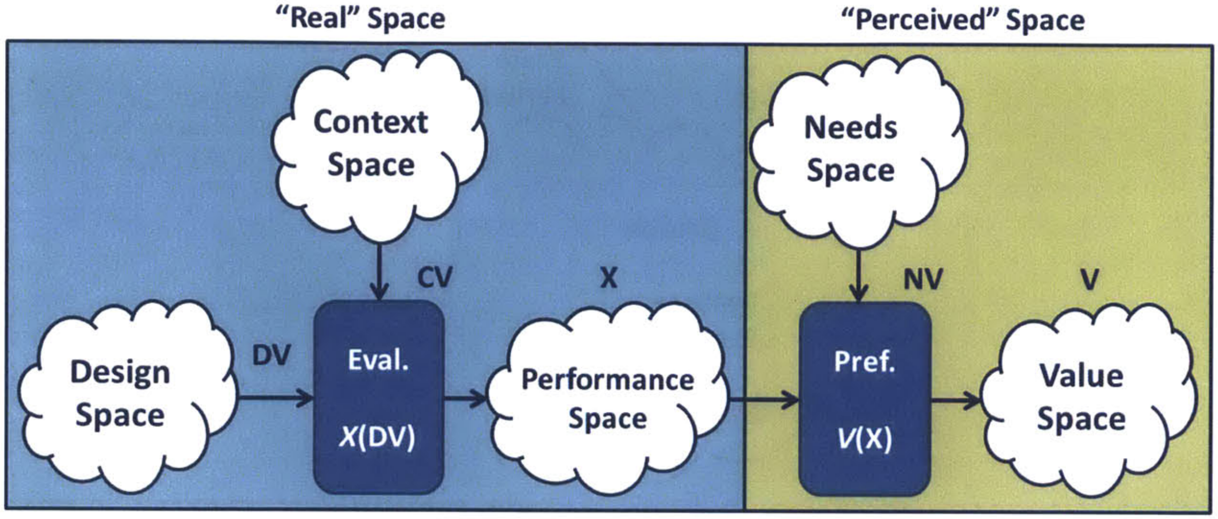


Figure 3-2. "Real" Space and "Perceived" Space

Figure 3-2 represents a useful construct going forward in this thesis. The "real" space and "perceived" space, as discussed above, represent the "world" in which a system is conceived, operated, and valued. The goal of design is to choose a solution (point) in the design space in order to achieve a successful point in the value space. However, factors in the context space and needs space may change, requiring strategies that will enable success in spite of these possible changes in the context and needs spaces.

3.1 Types of Perturbations

The idea of value delivery in the face of perturbations has been well thought out in literature and discussed earlier in this thesis as value robustness. Value robustness is the ability of a system to continue to deliver stakeholder value in the face of changing contexts and needs (Ross and Rhodes 2008a). To help characterize these shifts in contexts and needs, Ross and Rhodes (2008a) describe an approach called *Epoch-Era Analysis*. Derived from the analogous historical or geological use of the words, *epochs* are periods of time with relative consistency that make up *eras*. To be more specific for engineering systems, the full lifespan of a system is referred to as an *era*, which can be decomposed into *epochs*. An epoch is a period of time with fixed context and fixed value expectations (Ross and Rhodes 2008a). When there is a meaningful change in context or needs, this can be defined as an *epoch shift*. Thus, an epoch shift is a type of perturbation that a system may experience during its lifespan.

A parallel may be drawn to the real and perceived spaces discussed in Figure 3-1. Shifts in context will impact the "real" space, through which it may indirectly impact the "perceived" space, whereas shifts in needs will directly impact the "perceived" space. Perturbations can represent either a change in the space itself (e.g. the possible designs in the design space) or a forced transition between instances within the space (Figure 3-3).

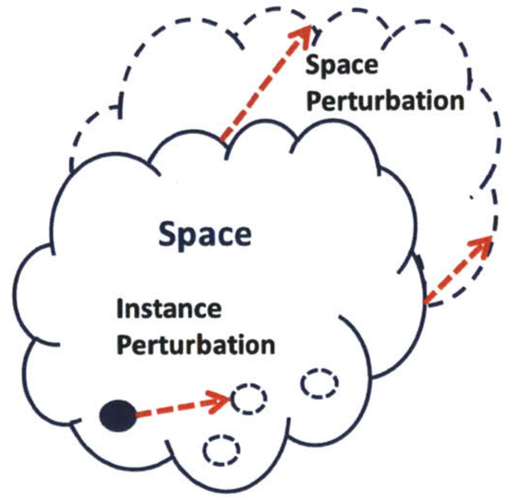


Figure 3-3. Types of Effects from Perturbations

In addition to epoch shifts, it is necessary to discuss another type of perturbation: a disturbance. Richards (2009) introduces the idea of *survivability*, or the ability of a system to minimize the impact of a finite-duration disturbance on value delivery. A disturbance then is some finite-duration perturbation that can affect a system. Mekdeci (2012) further clarifies:

> Since systems are expected to provide satisfactory value under ideal conditions, a *disturbance* can be defined as an unintended, finite duration, continuous state change of a system's form, operations, or context that could jeopardize value delivery (Mekdeci 2012, pp. 1-2).

This definition aligns disturbances with changes in context that may impact the real space. This thesis however will allow disturbances to affect not only the design or context, but also the needs. In this way, there are disturbances that affect both the real space and the perceived space.

Perturbations may then be categorized into two types: shifts and disturbances. The main difference being the nature of the change—whether it is some short-duration impact on the design, context, or needs, or a longer duration shift with respect to the decision-making timescale. Distinguishing between shifts and disturbances can be difficult, and depends largely on the timescale of decisions in the system of concern. Richards (2009) distinguished robustness from survivability in that robustness mitigates effects from "permanent" (or longer duration) changes, whereas survivability mitigates effects from (shorter) finite duration changes. While he was referring to distinguishing between robustness and survivability, the concept is parallel to the problem in distinguishing between disturbances and shifts.

The characteristic of a perturbation being finite, and therefore classified as a disturbance, can be ambiguous when examining responses to seasons or similar cyclical changes. While these changes are finite, they are also not of short duration and seem more like epoch shifts than disturbances. Seasons might *seem* like shifts to the average person due to the timescale on which decisions are made based on seasons. For example, even though the average person determines what to wear on a daily basis, depending on current weather forecasts, larger wardrobe decisions are made on a seasonal basis (snow pants and jackets are made available in the winter and sandals and t-shirts are made available for summer). Therefore, it makes more sense to classify seasonal changes (with respect to wardrobe decisions) as epochs, where random uncharacteristically hot or cold days would be classified as disturbances. However, if the system under concern is one where decisions are made on longer timescale (like decades), as in bridge maintenance, seasonal changes might be classified as disturbances. One can imagine a budgetary meeting to discuss the renovation of a city bridge where the timing of the decision may or may not take place in a season unfit for construction. In this case, the winter could be classified as a disturbance on bridge renovations until weather improves. The idea being that disturbances and epoch shifts are distinguished by the relative timescales of the change and decision-making. This distinction between shifts and disturbances will be clarified further in the following examples of the types of perturbations present in Figure 3-4.

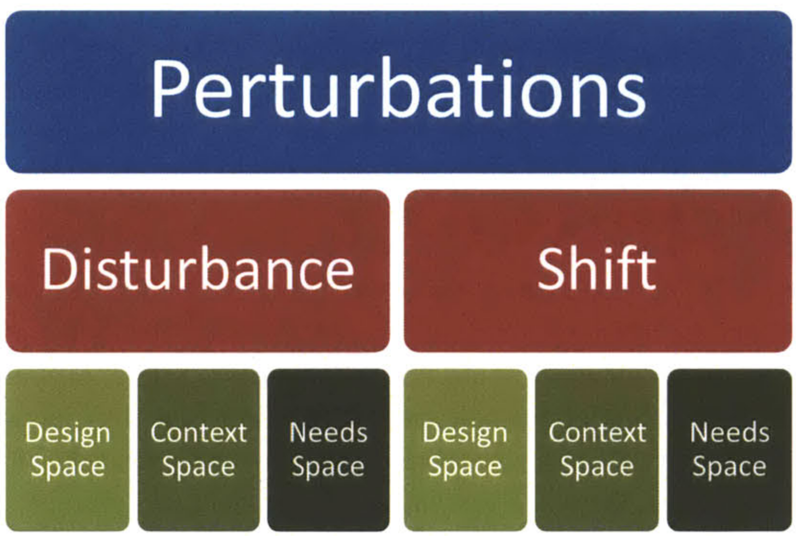


Figure 3-4. Types of Perturbations

Another important distinction among perturbations is the scope of the perturbation; perturbations can affect the "space" itself, by changing the *possible* points encapsulated by the space, or they can affect a change within the space, forcing one instance within the space to change states to another instance (Figure 3-3). For the remainder of this thesis, when referencing possible perturbations in the three spaces—design, context, or needs—it is assumed that these perturbations can have affects *to* the space and *within* the space (Figure 3-3).

3.1.1 Epoch Shifts

Epoch shifts, sometimes simply referred to as shifts, are defined as shifts in context and/or needs in order to gain insight into the impact of changing system exogenous factors (Ross and Rhodes 2008a). A shift in context will affect the real space whereas a shift in needs will affect the perceived space. For example, El Niño is period of warming of the surface of eastern Pacific Ocean that increases likelihood of extreme weather patterns (Rasmusson and Wallace 1983). This context shift can be classified as an epoch shift for passenger airlines, possibly introducing increased flight delays and cancelations due to weather issues. These increases in ocean temperatures often last for a year or more, which is a longer timescale than airlines make decisions on routes and daily operations and can therefore be considered a shift. This shift can affect how the airlines' design space translates to the performance space (and therefore the value space), possibly requiring the airlines to adjust their design parameters (such as flight schedules) to maintain acceptable value.

El Niño represents a context shift that affects how the design space translates to the performance space, but context shifts may also affect the design space directly as well. A change in FAA regulations, requiring flight at lower altitudes, may represent a shift in context (since these decisions last for the unforeseeable future) for an airline company, and directly constrain the design space (operations of flying lower). This may possibly affect the performance and value spaces in turn; flying at lower altitudes may improve safety of passengers by limiting exposure to solar radiation, but it may also decrease cruising speeds and engine lifetime while increasing costs. These shifts in context will often lead to responses in the design space. For example, the El Niño context shift may lead to a response of extra route planning or the FAA context shift may lead to a response of new wing or engine design.

Shifts in needs may also constitute epoch shifts and can directly affect the perceived space. For example, the value space of passenger airlines may be characterized by the system performance and various decision maker multi-attribute utility functions. These attributes, like passenger capacity or average flight delay for example, are the indicators that decision makers use in order to "perceive" system value. Assuming accurate representation of needs, if these attributes or their utility functions change, the perceived value of the system will change. For example, if an airline CEO is replaced, it is possible that the new CEO will have different expectations (needs) for the airline company, indicating an epoch shift that may or may not require a response in the design space (e.g. a redesign or CONOPS change).

The traditional epoch shifts just discussed involve only changes to contexts or needs. However, an epoch shift may also include an involuntary change to the design space or instance. This type of perturbation traditionally would have been classified as a disturbance, however this research argues that there can be shifts, in addition to disturbances, in the design space or instance within the design space. One can imagine a perturbation to the system that is not finite-(short) duration, but rather permanent (long) duration. For example, a passenger airliner may experience a bird-strike in flight that scratches the pilot's window, degrading the performance of the pilot. This could be seen as a disturbance originating in the context, but it may also lead to a shift in the design instance of the aircraft (i.e. the system itself is changed by the encounter). This in turn may affect the visibility performance of the aircraft and could possibly lead to an accident, affecting the value space[3].

Similarly, a shift in available technology could be described as a change in context, constituting an epoch shift. However, in the proposed framework, the change in technology can be viewed as a shift in the design space as well, through the technology's impact on potential design alternatives. With the new technology, the design space has widened, including the new possible designs. This is another example of a context shift directly affecting the design space (like the bird-strike example above).

For the purposes of this research then, the epoch shift definition is amended to include not only long-lasting perturbations in context or needs, but also long-lasting perturbations in the design. When deciding whether to classify a perturbation as an imposed change on context, needs, or design, one should consider that perturbations are constructs intended to separate out the factors that are within a designer's control and those exogenous factors that are uncertain and could impact the ultimate value delivered by the instantiated system design. The concept of "context" versus "design" space may appear a bit fuzzy, after all, the strict interpretation of context includes all factors outside of the system boundary. A good rule of thumb is that the design space describes the potential system designs, within which the actual instantiated system resides. An imposed change to the system would correspond to imposed movement within the design space. Likewise, a change to the possibilities of designs, both expansion (e.g. through new technology or concepts) and contraction (e.g. through constraints) is an imposed movement to the design space. Context perturbations then would be anything that changes the mapping of "design" space to "performance" space, where performance space includes not only traditional concepts of performance, but also cost factors and anything else that could be potentially chosen as decision criteria for value perception.

3.1.2 Disturbances

Disturbances, as described before, are finite-(short) duration changes of a system's design (i.e. form and operations), needs, or context that could affect value delivery. A disturbance in form

---

[3] The concept here described detailing shifts and disturbances in the design space serves only as an introduction to the construct. There may be additional research needed to characterize what constitutes a shift versus a disturbance (see for example, Mekdeci et al. 2012). The term "disturbance" is more commonly used to align with Richards' (2009) construct of survivability. However, more research is needed to more fully classify perturbations to the system that effect performance for the remaining life span of the system. The scope of this thesis does not cover details for perturbations in the design space, rather, it proposes the relationship within the real and perceived spaces and the imposed change in instances within or to spaces.

or operations affects the design space. Disturbances that affect the design or the context take place in the real space where disturbances that affect the needs take place in the perceived space. For example, an engine failure on an aircraft, occurring in the design space as an imposed state-change in system instance, can drastically change the flight characteristics and performance of the plane. This perturbation would result in an emergency landing and has a short enough duration that it does not constitute an epoch shift within the lifespan of the aircraft. This disturbance then would affect the performance of the aircraft and therefore the value of the system.

A context disturbance example could be a bad storm that would cause an aircraft to reroute. A bad storm usually passes through an area with a timescale approximately in hours, whereas plane routes are generally made on a day-to-day basis. Therefore, a bad storm would be a disturbance, taking place in the real space (just as the engine failure example). The new route might affect the plane's performance, using more gas and taking more time, which would then affect the value space.

The idea of needs disturbances was implicitly covered in Richards' survivability work. The short-term transition from an airline company with CEO A to CEO B may result in performance loss when interpreted through the old expectations. However, stakeholders may change their expectations because they realize CEO changeover is a period of transition. Therefore, this can be considered to be a disturbance in the needs space, which may make the lower performance of the organization acceptable (at least until expectations adjust back). This relates to the "permitted recovery time" and "emergency value threshold" idea from Richards (2009). In this case, shortly after a design space disturbance (the CEO changeover) there is a needs space disturbance (short-term lowered performance expectations). For Richards (2009), in the survivability definition is a short-term disturbance in value expectations (needs) immediately after a disturbance to the system.

3.1.3 Mapping Perturbations

A perturbation map can help to summarize all the ways a perturbation can affect a system—through the design space, context space, or need space (Figure 3-5). As discussed above, these perturbations include disturbances and epoch shifts and can be initiated at various nodes within the real and perceived spaces. It is important to note however that the performance space and the value space are not directly affected by these perturbations. Instead, the perturbations take place as inputs to the performance evaluation and value evaluation. In this way, performance depends on instances of design and context, and value depends on instances of performance and needs (preferences).

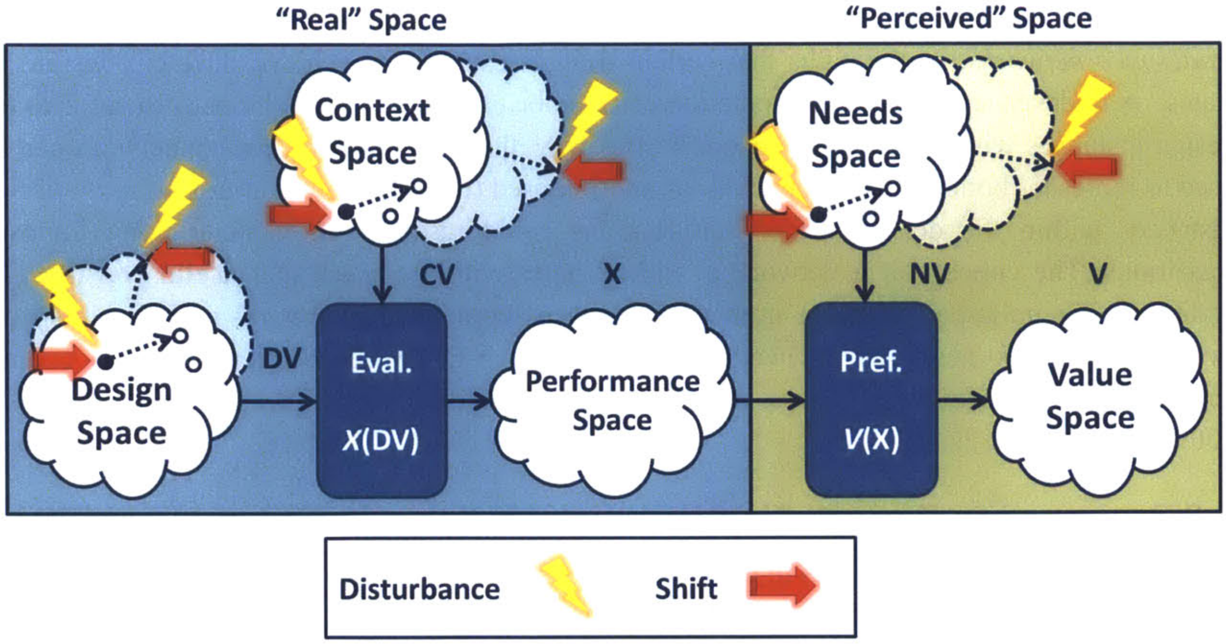


Figure 3-5. Mapping Perturbations to the "Real" Space and "Perceived" Space

In this perturbation framework, there are twelve ways perturbations can impact a system's value delivery. Table 3-1 summarizes the twelve possible permutations. The bolded permutations are highlighting the types that will be considered within this thesis. The table also orients the reader to where some previous research has classified or described some of these types.

Table 3-1. Mapping of Twelve Possible Perturbation Types

		Perturbation	
		Shift	Disturbance
Design	Instance	Specific Design	**Specific Design** (Richards 2009)
	Space	**Possible Designs** (mentioned in Ross et al. 2008)	Possible Designs
Context	Instance	**Specific Context** (Ross et al. 2008)	Specific Context
	Space	**Possible Contexts** (Ross et al. 2008)	Possible Contexts
Needs	Instance	**Specific Needs** (Ross et al. 2008)	**Specific Needs** (implicit in Richards 2009)
	Space	**Possible Needs** (Ross et al. 2008)	Possible Needs

As discussed in Chapter 2.2.3 on *trade studies*, Ross and Hastings (2006) develop the idea of *tradespace networks*, or the paths that certain designs can take to achieve different designs or states. A tradespace network representation can also be used to represent the perturbations to the design instances within the design space. In this way, the arcs in the tradespace network can be used to represent both intended (i.e. chosen) and imposed (i.e. forced) transitions between design instances within the design space. Perturbations could result in movement along imposed transitions. The concept of a network of linked states within a space is a useful construct; in addition to the tradespace network in the design space, there may be networks in the other spaces as well (i.e. context networks and needs networks). Epoch-Era Analysis can make use of context networks in the construction of eras, which are time-ordered sequences of epochs (Roberts et al. 2009; Fulcoly et al. 2012)[4].

3.2 Response to Perturbations

Systems can use various methods to counteract the effects of perturbations. The ultimate goal is to sustain value across disturbances or shifts. In order to distinguish between these two types of perturbations, this research uses the following concepts for maintaining value across perturbations: in terms of value, systems should seek to survive disturbances and be robust to epoch shifts.

3.2.1 Value Sustainment

The concept of *value sustainment* is introduced here as a new construct to integrate research in both disturbances and epoch shifts relating to survivability and value robustness respectively. *Value sustainment* is defined as the ability to maintain value delivery in spite of epoch shifts or disturbances and is represented in Figure 3-6 below.

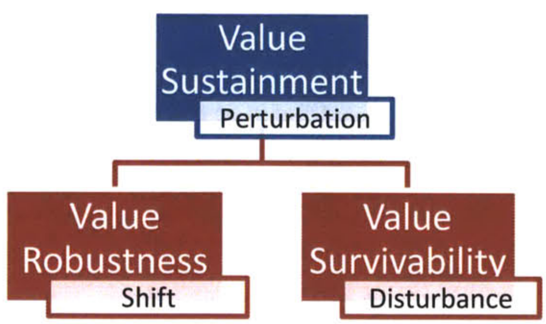


Figure 3-6. Value Sustainment

[4] Currently, MATE methods do not explicitly consider perturbations to "spaces" discussed in this section. Most simulations include changes only in the instances of designs, contexts, or needs, within their respective spaces. More research should be done to investigate the implications of space disturbances or shifts.

This small, but necessary, clarification reconciles the idea of value robustness (maintaining value delivery in response to shifts in contexts and needs) with survivability (minimizing the impact of a finite-duration disturbance on value delivery).

3.2.2 Value Robustness Research

Epoch-Era Analysis (EEA) is an approach for conceptualizing system lifespans. As systems progress through changing contexts and needs, their performance may exceed or dip below expected utility values. The idea of value robustness is to maintain value delivery despite these shifts (Ross and Rhodes 2008a). A system can change, remain robust, or be versatile to meet expectations in response to epoch shifts. Therefore, the three strategies that systems designers may use to achieve value robustness would be changeability, robustness, and versatility. Changeable systems can change to meet new needs or operate in new contexts. Robust systems maintain level and set specification parameters in new contexts while versatile systems satisfy diverse needs without needing to change.

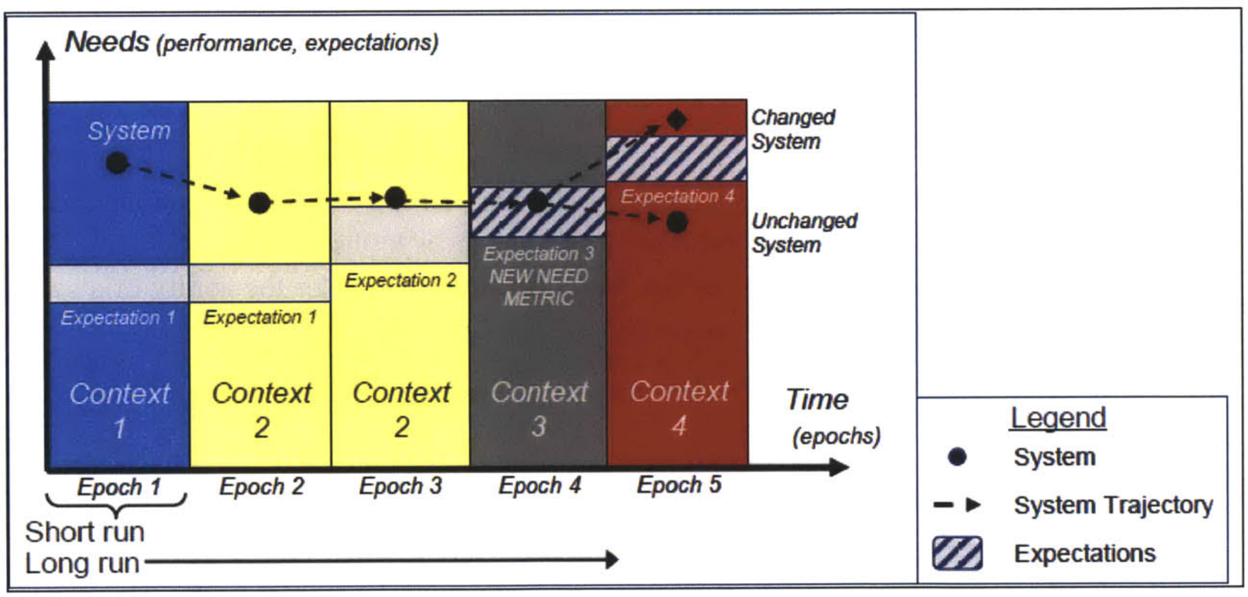


Figure 3-7. System Needs versus Expectations across Epochs of the System Era (Ross and Rhodes 2008a)

Figure 3-7 shows this temporal progression of a system as needs and contexts change. The vertical columns represent the epochs that are time-ordered to form an era. The different colors of these epochs represent changes in context. The horizontal bands capturing the minimum and desired expectation levels for that epoch represent expectations (needs). Notice that contexts and needs can change independently of one another. In Epoch 1, the system exceeds the needs of stakeholders. The epoch shifts to Epoch 2 and Epoch 3 represent changes in context and needs respectively. In both cases, the system still meets expectations and displays *value robustness*. The shift to Epoch 4 represents a new need of the system, which the system satisfies, displaying *versatility*. In Epoch 4, the system does not exceed all the needs, but does meet the minimum required level and remains successful. The final epoch shift to Epoch 5 represents the need for a changeable system. In this case, the system is not robust or versatile to new needs and context, and thus must be changed in order to remain successful (Ross and Rhodes 2008a). The depicted perturbations represent epoch shifts that the system must endure in order to remain successful. There may be however, disturbances that the system must survive in addition to dealing with the epoch shifts described above.

In support of evaluating value robustness, Epoch-Era Analysis can be used in conjunction with Multi-Attribute Tradespace Exploration (MATE) to evaluate designs in many discrete epochs. Stringing epochs together may inform lifecycle performance across system eras. Ross (2006) uses dynamic MATE across system eras to assess certain ilities (changeability, flexibility, adaptability, scalability, modifiability, and robustness). Transition rules specify the availability, cost and time of possible state transitions from designs to other designs. To assess the robustness of a system, Ross introduced the Pareto Trace metric, which represents the frequency with which a particular design appears in the Pareto Front of enumerated tradespaces. A system's changeability is assessed using Filtered Outdegree, which represents the number of potential transition paths available to a design (Ross and Hastings 2006, Richards 2009).

3.2.3 Value Survivability

While a system can achieve value sustainment in the long term through value robustness, the short-term value sustainment may be perturbed by disturbances to the system. Richards (2009) formalizes many of the constructs used in this thesis for survivability, or the ability of a system to minimize the impact of a finite duration disturbance on value delivery.

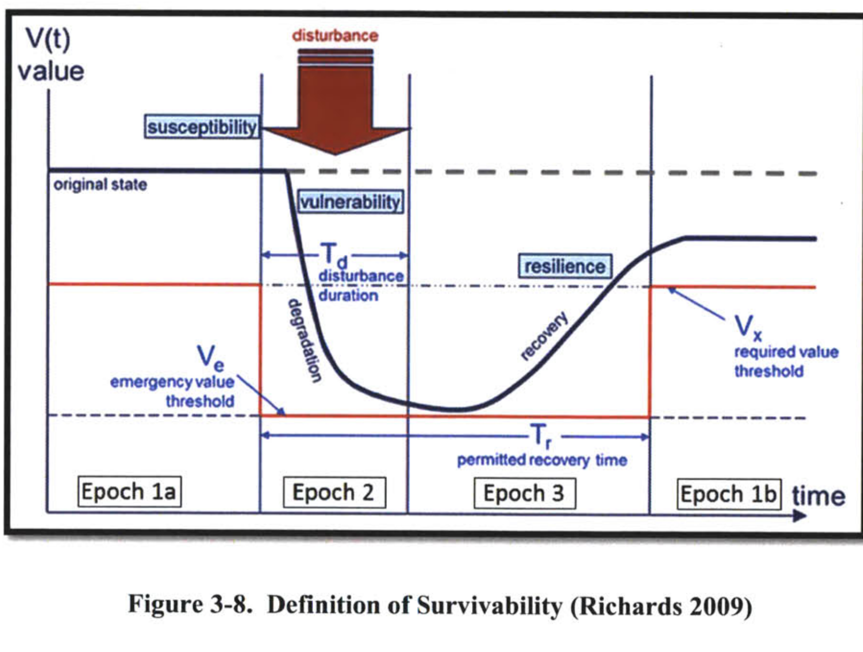


Figure 3-8. Definition of Survivability (Richards 2009)

Figure 3-9 shows three types of survivability; susceptibility (Type I), vulnerability (Type II), and resilience (Type III). Richards' survivability construct uses the idea of epochs (Ross and Rhodes

2008a) to define the periods throughout the disturbance. The idea being that after some disturbance, there is a return to the original epoch (i.e. the time period of the disturbance is short and the system environmental condition reverts to its prior state). In Epoch 1a, the system successfully delivers value in the baseline environmental conditions and stakeholder expectations. Then, in Epoch 2, the system experiences a finite duration disturbance that degrades performance (e.g. damage to the system, or harsh operation environment). There may be a lowered threshold value expectation for the system when exposed to a disturbance, and the system in this case meets that threshold. Epoch 3 is the permitted time period for recovery of the system in response to the degradation. Finally, the system returns to the original context with the system possibly permanently degraded or fully recovered. Type I survivability then represents the prevention of the possible disturbances. If a system is not susceptible to the disturbance, then it is unaffected by it. Type II survivability requires meeting some minimum accepted value expectation during and immediately following the disturbance, that is disturbance impact mitigation. The degree to which the impact of the disturbance reduces value delivery is the system's vulnerability to the disturbance. Finally, Type III survivability requires the system be able to recover to exceed the original value threshold within a permitted recovery time, and then the system is considered to be resilient to the disturbance (Richards 2009).

Richards (2009) came up with design principles Figure 3-9 that enable survivability that coincide with the different phases of a disturbance seen in Figure 3-9.

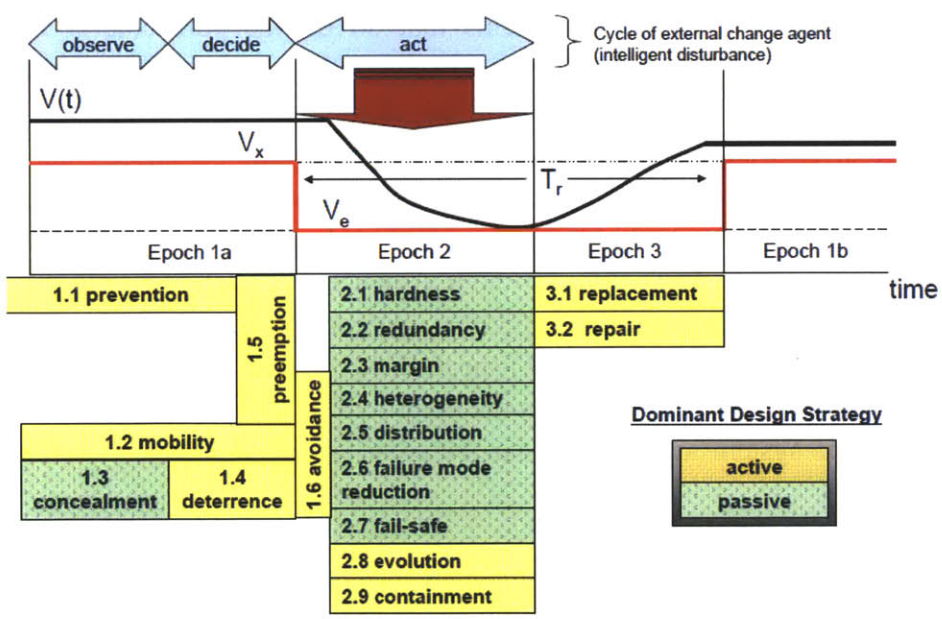


Figure 3-9. Mapping of Design Principles to Disturbance Lifecycle (Richards 2009)

The survivability design principles represent methods or strategies that can be used to enable survivability in a system. These principles are aimed at reducing susceptibility, reducing vulnerability, or enhancing resilience (Richards 2009).

In Richards' definition of survivability, value is explicitly identified as the key parameter that is maintained across the disturbance. One can imagine however, that a stakeholder might be interested in the maintenance of a different system parameter. Often these system parameters are attributes that relate to value. However, an ility definition may be made clearer if it avoids attaching a value statement to the meaning. In this way, survivability might be defined as the ability of a system to minimize the impact of a finite duration disturbance on a system parameter. If one cares about the survivability of paint color versus exterior form of a car (dents), those parameters may relate to value, but can be specified more clearly. To make a clear parallel between survivability and *value robustness*, this thesis will from now on refer to *value survivability* when discussing the survivability that aims at maintaining value-delivery across disturbances.

3.3 Change Over Time

While value sustainment may be achieved by intentionally changing, or not changing, the system in response to perturbations, this thesis focuses on systems that intentionally change. It is important however to see how these types of systems fit in with those that do not change, that is, those that remain statically robust and/or survivable.

3.3.1 Change vs. No-Change

One aspect that makes distinguishing between static systems—those systems that do not change—and dynamic systems—those systems that do change—difficult is whether one is talking about "change" relative to the system parameter or to the outcome parameter. Often when we think about robustness or survivability, the outcome parameter is the focus. In the case of value robustness, or value survivability, value is the outcome parameter that is targeted to remain above threshold values. The end user may not care how this robustness or survivability is achieved. The system can "change" in order to maintain a level of value, or it may "not change" to remain statically robust or survivable. To help visualize this relationship, Figure 3-10 below shows the two-by-two matrix of possible system parameter options and outcome parameter options. A system parameter will be some aspect of the system, form, function, or operations that relates to the design space, and is within the control of the designer. An outcome parameter can be either another system aspect or it can be some outcome attribute of the system, like performance or utility.

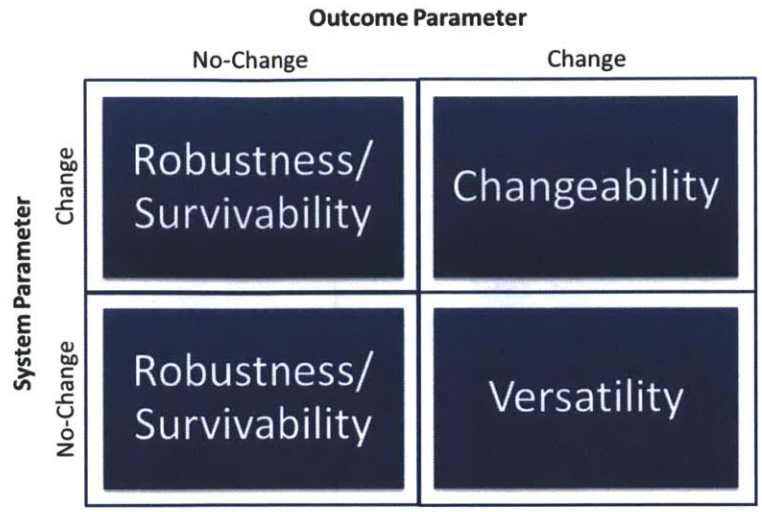


Figure 3-10. Typology of System Parameter vs. Outcome Parameter Change or No-Change to Achieved Desired Quality in Outcome Parameter

Figure 3-10 shows a matrix of possible combinations of changing system parameters or keeping them static in order to change or maintain an outcome parameter. The labels placed in each of the quadrants are with respect to the outcome parameter. The most explicit label is changeability in quadrant one. If a system changes any of its parameters in order to achieve a change in an outcome parameter, then that transition is changeable. Figure 3-11 shows an example of this with a camping utility knife that can change form and operations (system parameter) to achieve a change in function or even a change in value (outcome parameter). If a system does not have to change in order to achieve a change in outcome parameter, it is a versatile system (quadrant four). For example, Figure 3-11 shows a camping knife/fork/spoon combo tool that requires no change in form (system parameter) order to achieve multiple functions or meet multiple needs (outcome parameter). The left side of the matrix displays the survivable or robust designs that can either change or remain static in their system parameters to result in no-change of outcome parameters. The labels in both quadrants two and three are the same, "Robustness/ Survivability," since the only distinguisher is the type of perturbation under consideration. This dimension turns this relationship into a three-dimensional space and is discussed later, in Figure 3-12. If a system does not need to change in response to a perturbation in order to maintain an outcome parameter level, the system is robust or survivable. Figure 3-11 shows an example of this in quadrant three with a strong camping knife that can handle disturbances (like drops) or shifts (like use under water in SCUBA) and still function or provide value. Finally, there are the systems that change a system parameter in order to maintain an outcome parameter. In this case, Figure 3-11 shows a camping lantern that can change off and on or level of brightness to maintain visibility or light. The construct displayed in Figure 3-10 is helpful when trying to classify whether certain responses are static or dynamic, but identifying "change" or "no-change" is not sufficient, one needs to also specify the type of parameter under consideration (system or outcome).

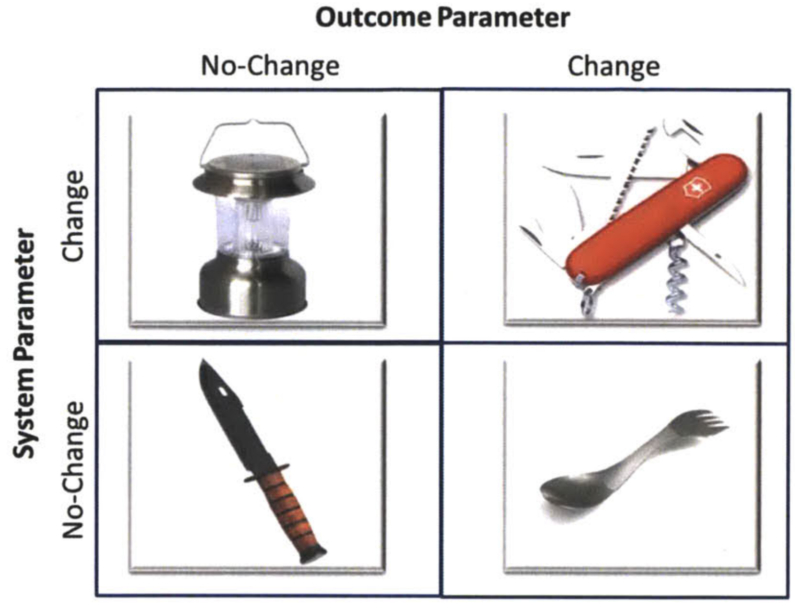


Figure 3-11. System Parameter Options vs. Outcome Parameter Options Examples

A three-dimensional representation of the categories can clarify the reason the four quadrants in Figure 3-10 are not mutually exclusive. There is another dimension (not shown in the figure) that characterizes the type of perturbation to distinguish between survivability and robustness. Figure 3-12 shows the third dimension and where the labels fall, depending on which axes are under consideration.

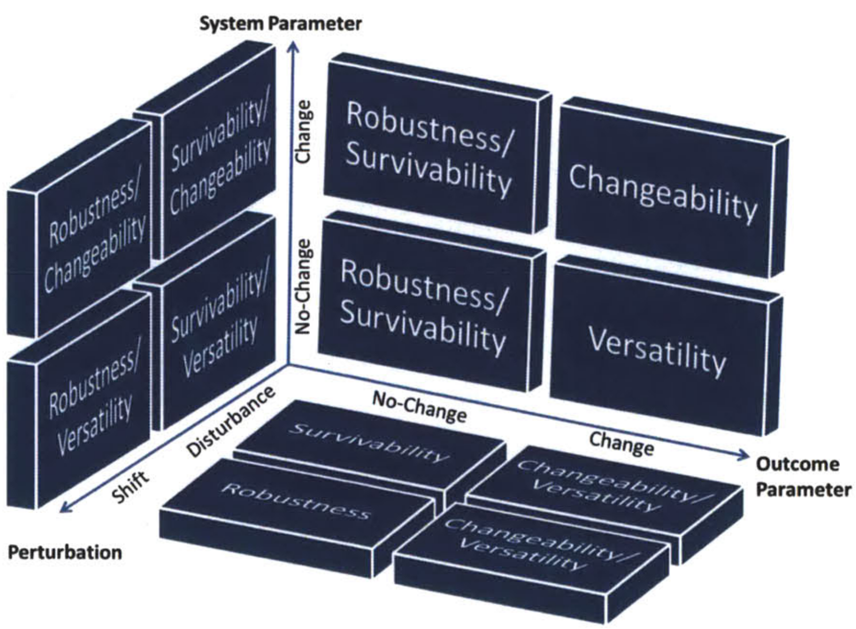


Figure 3-12. Three-Dimensional Projection of Perturbation and System/Outcome Parameter Changes

Figure 3-12 is a projection of the three-dimensional space onto the three different planes. To visualize how versatility and changeability interact with robustness and survivability, Figure 3-13 shows how the constructs may become exclusive depending on which perspective the user is taking. This illustrates the difficulty in having a clear discussion on what types of changes are taking place in a system as it attempts to maintain value across disturbances or epoch shifts. It is interesting to note, that if one is to evaluate system responses based on system parameter and perturbation type, that there can be no full distinction between any of the categories. This shows how we commonly need the outcome parameter as a reference with either perturbation type or system parameter to distinguish between robustness, survivability, versatility and changeability. Classifying these four ilities requires three dimensions (perturbation type, system parameter, and outcome parameter). If one were to consider more ilities in a common framework, we could imagine needing many dimensions. Chapter 5 discusses this in more detail.

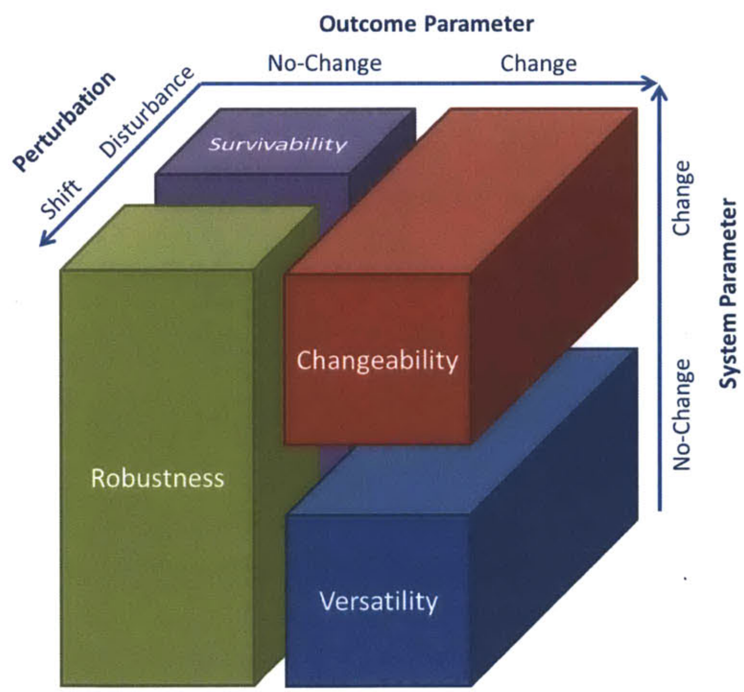


Figure 3-13. Three-Dimensional Perturbation-System Parameter-Outcome Parameter Space

3.3.2 Passive vs. Active Responses

Methods of responding to perturbations can be considered as either passive or active. Active responses are those where there is an agent that Observes, Orients, Decides and Acts, a cycle known as the Boyd cycle or OODA loop (Osinga 2006). Passive responses, on the other hand, have no agent that is intelligently making a decision—the system responds according to the laws of physics, with predicable cause-effect relationships. One might assume that active responses align with dynamic responses and passive responses align with static responses, however, this would be incorrect. There can actually be passive change and active no-change Figure 3-14. For example, explosive armor (Mania 1971) on artillery tanks reacts to incoming projectiles with controlled explosions to mitigate damage; this is an example of a passive response to change a system parameter (state of explosives) to maintain an outcome parameter (hull integrity). Another simple example of passive change is a mercury thermometer controlling a HVAC system, passively changing the state of a heater or air conditioner (system parameter) to maintain room temperature (outcome parameter). An example of a passive system change to change an outcome parameter could be triggering an automatic activation device (AAD) on a parachute to save a jumper's life. An AAD passively triggers when going through a certain altitude at a velocity deemed unsafe (the jumper may have passed out in free-fall and not opened the parachute) and automatically pulls the chute to save the life of the jumper. This is passive since the trigger mechanism is physically controlled by the change in pressure during free fall, and no active decision must be made to execute the change in system state.

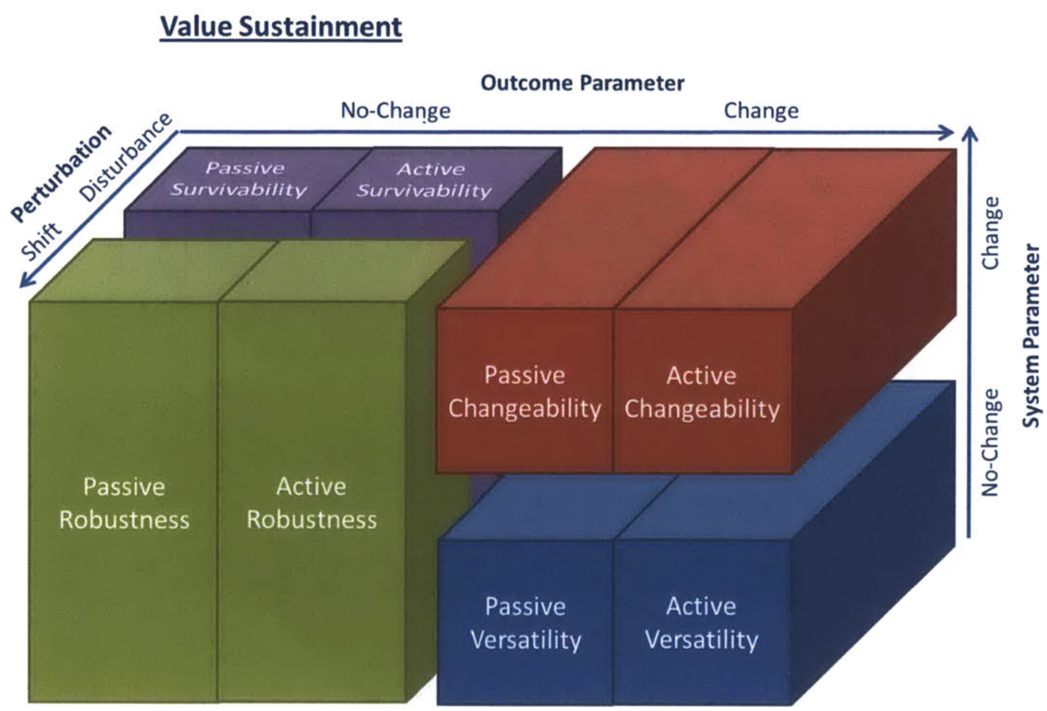


Figure 3-14. High-Level Value Sustainment Space

The important take-away from these representations of value sustainment is that one must specify the system parameter, or the response in terms of an aspect of the actual system, as well as the outcome parameter, the result of the response. Once the system parameter is distinguished from the outcome parameter, then one may trace the change/no-change in system parameter back to either a passive or an active response. Additionally the perturbation type may be used to distinguish between robustness and survivability. The four dimensional representation is shown in Figure 3-14.

This representation does not show that value robustness and value survivability are mutually exclusive from changeability and versatility. In fact, quite the opposite is shown: changeability and versatility can help enable value sustainment, or value survivability and value robustness. Figure 3-15 shows the relationship between the response types (green) used in the previous figures and the higher goals of value survivability and value robustness and finally value sustainment.

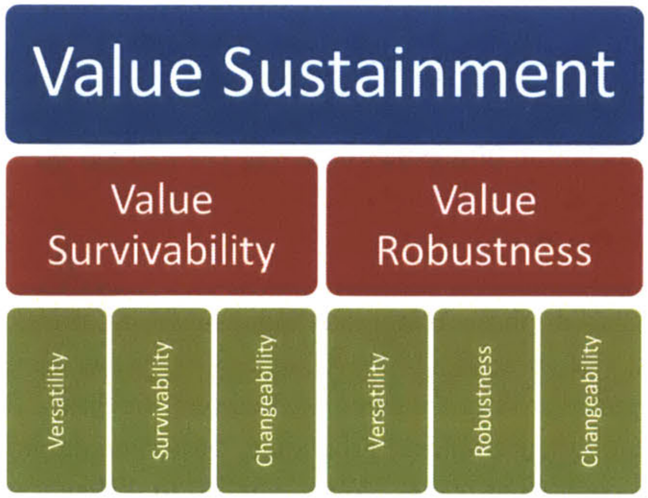


Figure 3-15. Value Sustainment Ility Breakdown

More research with ilities and possible interrelationships amongst ilities can be found in Chapter 5 of this thesis. As discussed earlier, this thesis tends to focus on the change-related ilities that involve dynamic responses to perturbations.

3.3.3 Changeability

This section details a little more of the intricacies of changeability than previously discussed and can be explained in even more detail in Ross, Rhodes and Hastings (2008) and Ross and Rhodes (2011). The motivation for changeability in a system is categorized into three major drivers according to Fricke and Schulz (2005) (Ross, Rhodes and Hastings 2008): 1) dynamic marketplace, 2) technological evolution, and 3) variety of environments. These drivers require that systems architectures address: 1) the ability to be changed easily and rapidly, and 2) insensitivity or adaptability towards changing environments (Schulz and Fricke 1999; Schulz et al. 2000). Changeability is often mentioned with other ilities in literature (flexibility, adaptability, scalability and modifiability), particularly with manufacturing processes literature (Giachetti et al 2003; Algeddawy and ElMaraghy 2009). Flexibility in product design is cited as an important characteristic for companies that design products in dynamic technological and market environments (King and Sivaloganathan 1998; Rajan et al. 2005).

This need for change in technological systems has been accompanied by proposed means to measure it. Empirical measures for flexibility (Chen and Yuan 1998), adaptability (Li et al. 2007), and robustness (Hwang and Park 2005) have been developed to help systems engineers design these systems with desired properties. deWeck, deNeufville, and Chaize (2004) use the notion of *real options* to valuate flexibility in space systems, using an example of Low Earth Orbit communication satellites with staged deployments. These definitions have been synthesized into a prescriptive six-element framework for space systems by Nilchiani (2005) and Nilchiani and Hastings (2007):

1. System boundary
2. System aspect
3. Time window of interest
4. Uncertainty profile within time window
5. Degree of Access
6. Value delivery response to change

One of the weaknesses with these many methods, however, arises from varying use and definitions of the terms. Rajan et al. (2005) as well as Schulz et al. (2007) have attempted at defining aspects of changeability and the associated design principles. The strengths in many of these definitions come from their empirical grounding. However, empirically derived definitions may rely too heavily on contextual biases. When applied to different contexts, or used in other applications, these definitions may fall short. An effective definition should be empirically grounded, but also free from contextual biases (Ross, Rhodes and Hastings 2008).

This research uses the following constructs to define changeability in systems engineering. Ross, (2006) defines a change made to a system as a transition from one state to an altered state over time. Every change can be characterized with three elements (Figure 3-16):

1. The change agent
2. The change mechanism
3. The change effect

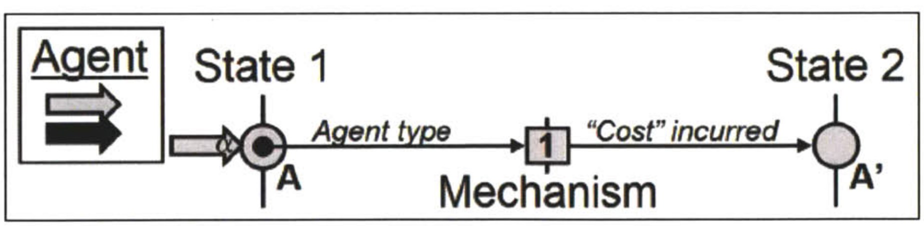


Figure 3-16. System Change Framework (Ross, Rhodes, and Hastings 2008)

The change in Figure 3-16 is a simplified case of a system change with only one particular change being captured. Ross, Rhodes and Hastings (2008) present this *agent-mechanism-effect* construct in a more complete representation in Figure 3-17. A change in this framework is represented by a *path* from State 1 to State 2. Changeability then is the ease in which a system can undergo various changes.

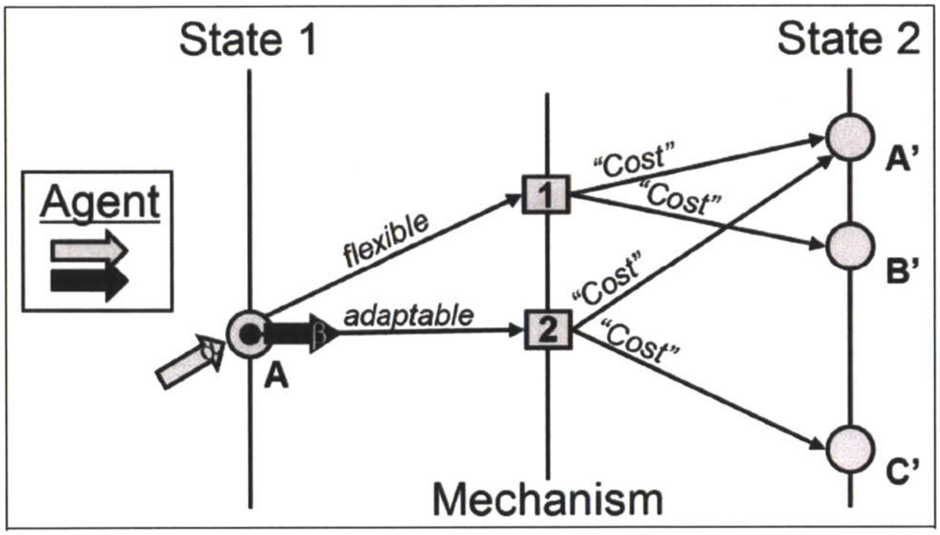


Figure 3-17. Expanded Change Framework (Ross, Rhodes, and Hastings 2008)

Figure 3-17 shows the difference between adaptable changes (those with internal change agents represented by a β) and flexible changes (those with external change agents represented by an α). The agents may use various mechanisms to achieve new states with varying costs for change. Table 3-2 shows the applicable variables each of the stages from Figure 3-17.

Table 3-2. Agents, Mechanism, Effects, and Paths of a System Change (Ross, Rhodes, and Hastings 2008)

Element	Description	As Illustrated in Figure 2
Change Agent	*The force instigator for the change to occur, for example humans, software, Mother Nature, etc.*	α, β
Change Mechanism	*The particular path the system must take in order to transition from its prior to its post state, including conditions, resources, and constraints*	1, 2
Change Effect	*The difference in states before and after a change has taken place.*	A'-A, B'-A, C'-A
Potential Paths	*The potential paths for the system to change from one state to another.*	α:A-1-A', α:A-1-B' β:A-2-A', β:A-2-C'

In order to incorporate a perturbation into this framework, Ross and Rhodes (2011) add a decision to execute a change after experiencing a perturbation, either changing in response or accepting disturbance (Figure 3-20).

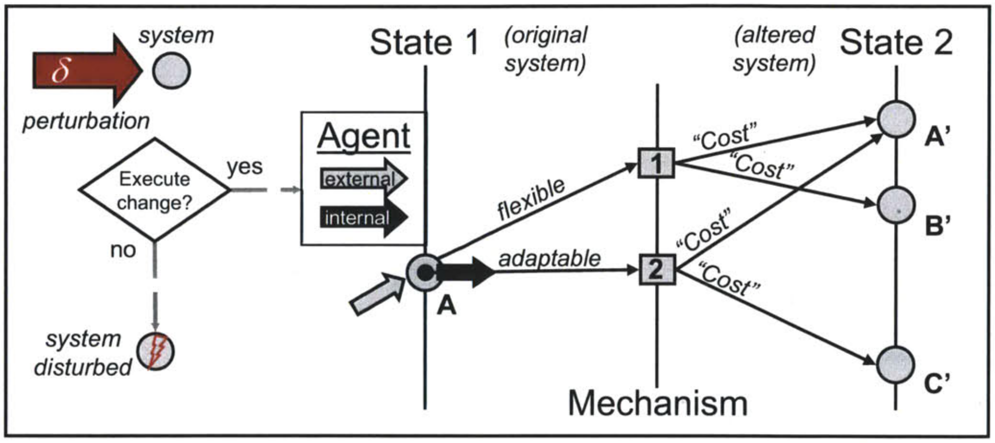


Figure 3-18. Change Pathway with Perturbation (Ross and Rhodes 2011)

The *change agent* as defined above is the force instigator for the change to occur. Change agents can include people, electronics, software, animals, or even Mother Nature. There need not be intent to change a system. Intelligent change agents may be differentiated on their degree or capability of using the Observe, Orient, Decide, Act loop previously discussed (Osinga 2006). The location of the change agent determines whether the change is adaptable or flexible. This distinction relies on a definition of the system boundary. One can imagine a plane and a pilot system where changes the pilot makes to the aircraft being adaptable if the pilot is regarded is interior to the system or flexible if the pilot is held outside the system boundary. Figure 3-19 summarizes the two classification types (Ross, Rhodes, and Hastings 2008).

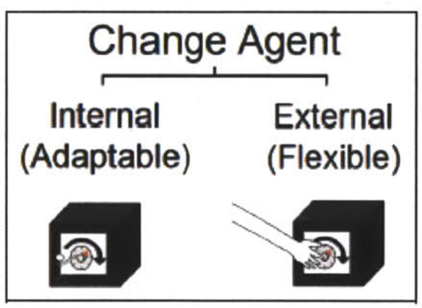


Figure 3-19. Change Agent Location (Ross 2006)

The c*hange effect* is the difference in the beginning and end states of the system. Ross (2006) classified the effects into three categories: *robustness, scalability,* or *modifiability.* These categories are based on parameters that can describe the system. *Robustness* is the ability to maintain constant parameters in spite of changes. *Scalability* is the ability to change the level of a parameter, and *modifiability* is the ability to change the set of parameters (Ross, Rhodes and Hastings 2008). For example, scaling an aircraft system parameter of fuel would be adding or emptying the fuel tank, whereas modifying the fuel parameter would be changing the type of fuel used. Figure 3-20 shows this relationship in the change effect graphically.

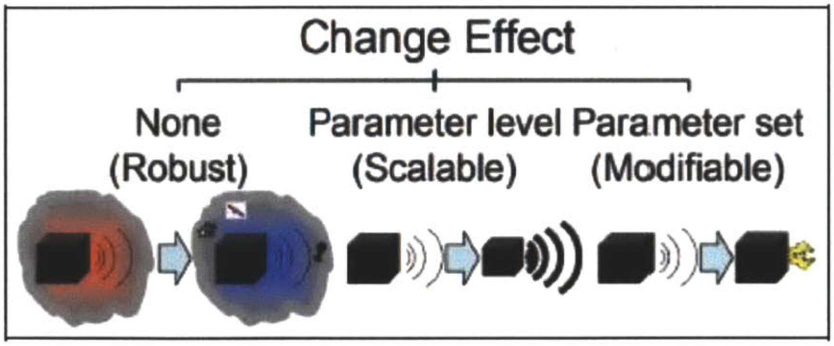


Figure 3-20. The Change Effect (Ross, Rhodes and Hastings 2008)

The *change mechanism* is the path the system must take to transition from State 1 to State 2 (Ross, Rhodes and Hastings 2008). This path includes the necessary components to bring about the change, (conditions, resources, constraints) and when paired with a decision make up an option.

Ross and Rhodes (2011) clarify this relationship further by describing the idea of path enablers. *Path enablers* are design decisions that give the system the option to execute a change mechanism, or reduce the cost of execution of a change mechanism, and empower the decision node in Figure 3-18. Path enablers can be implemented in systems using design principles to engineer effective systems with the ability to change states. These design principles, which can help to generate path enablers, are generally true prescriptive statements that can be used in a wide array of systems. Wasson (2006) defines principles as "A guiding thought based on empirical deduction of observed behavior or practices that proves to be true under most conditions over time." Additionally, the change mechanism execution can relate to specific change-related ilities that are determined by how well a system displays certain properties through the change. Therefore, a relationship can be drawn between design principles that system designers may use in the early stages of concept development to the ility significance of a change executed during a system's lifespan. Figure 3-21 shows this conceptual flow from design principles to the valuation of the ilities.

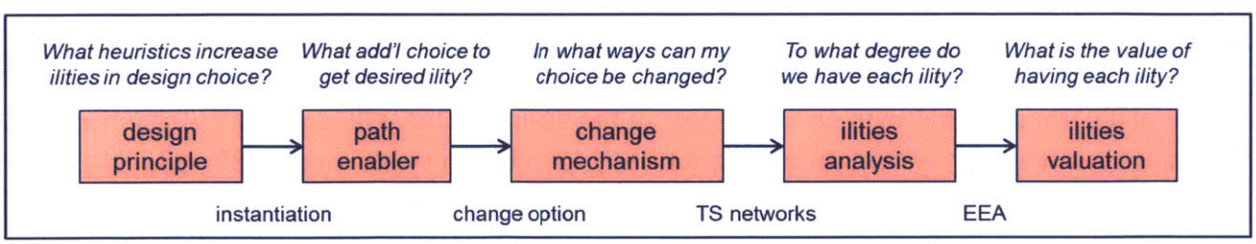


Figure 3-21. Conceptual flow between Design Principles and Ilities (Ross and Rhodes 2011)

In summary, a path enabler can be an instantiation of design principles used in system design, which enables the use of some change mechanism, giving the option to the system owner/user to change states of the system. The change option is similar to the idea of an option in real options analysis in that investing in certain path enablers gives owners the ability to execute an option *in* a system (Wang 2005; de Neufville and Scholtes 2011). The tradespace networks discussed in Chapter 2.2 of this thesis can represent the potential changes available for a system from change options. Tradespace networks can be used to characterize certain ilities that could be evaluated using techniques like Epoch-Era Analysis.

The various aspects of a change option, discussed in Ross and Rhodes (2011), are characterized in more depth using Figure 3-22, tracing from implementing the path enabler through execution or expiration of a change mechanism. The figure shows the applicable costs, temporal aspects, and anatomy of the change option.

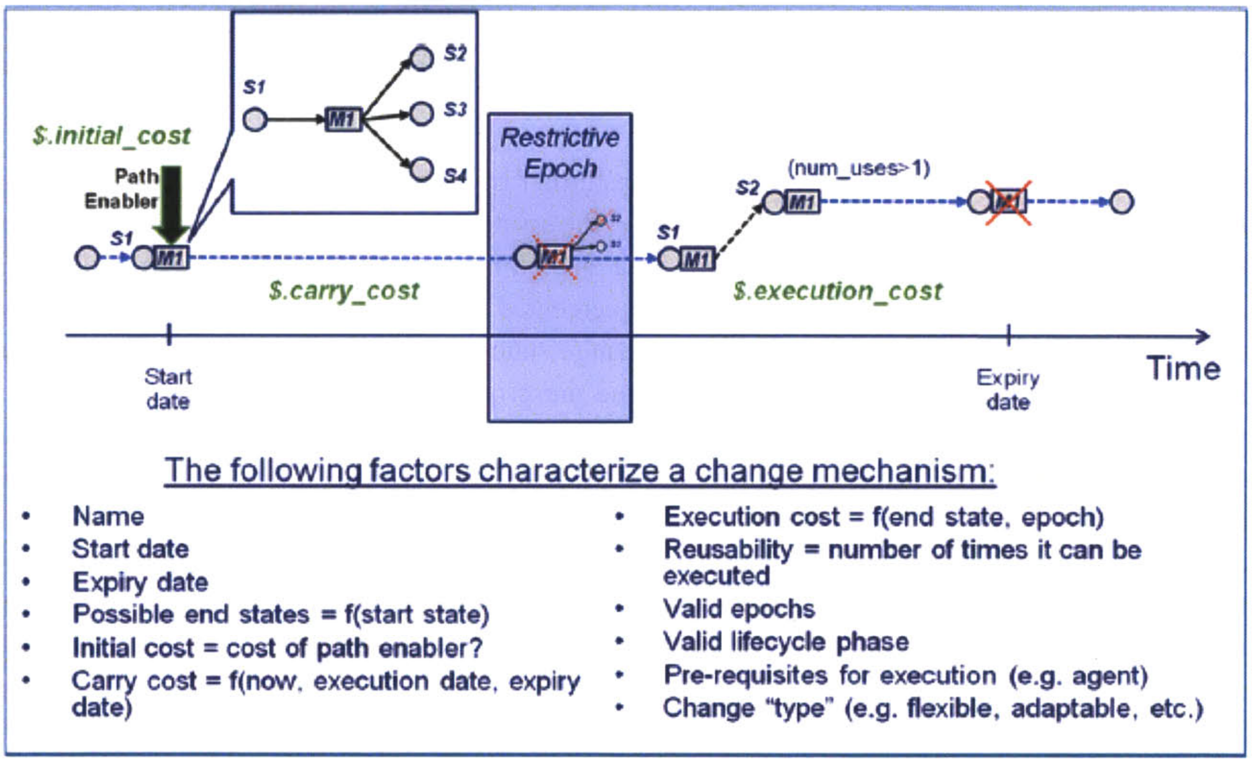


Figure 3-22. The Change Option (Ross and Rhodes 2011)

A path enabler may come with an initial cost in the design. This will enable a mechanism (M1) that can transition the system from State 1 (S1) to other possible states (S2, S3, S4). This mechanism may come with a carrying cost, maintaining the ability to initiate execution at a future point in time. There may be restrictions placed on when the system can execute the mechanism, a certain context or lifecycle phase as well as any pre-requisites. Once the mechanism is executed, there is a cost associated with the state change to State 2 (S2). This mechanism may be reusable or reversible or it may have a finite number of executions. Finally, the mechanism may expire, before or after execution. One can imagine the possibility of end states from either one powerful change mechanism or the use of multiple mechanisms in the same system. The varying number of change mechanisms and end states can range from specified to open-ended and countable to uncountable as seen in Figure 3-23. One can see that the degree of changeability of a system is dependent on the nature of the change mechanisms available to that system.

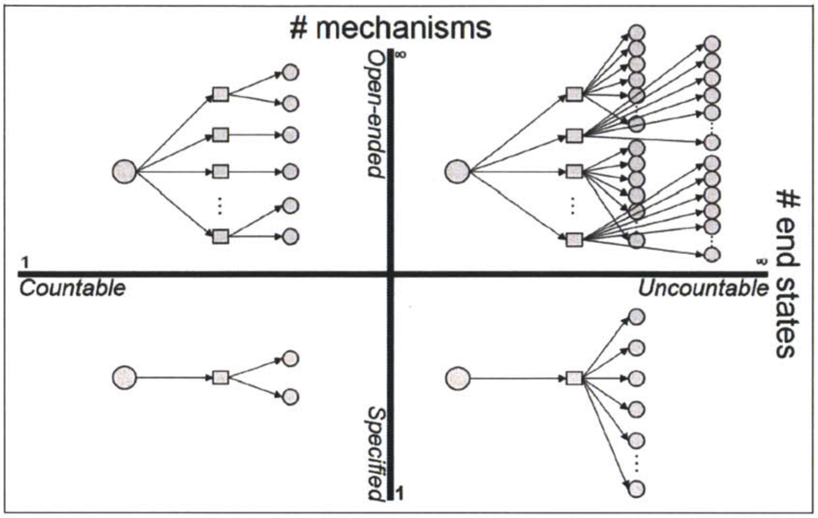


Figure 3-23. Counting Change Paths (Ross, Rhodes and Hastings 2008)

A convertible automobile may illustrate a simple example of this idea. A normal car may have a path enabler of a mechanically moving roof added to its design. This path enabler comes at an additional initial cost to the car owner for the added complexity in design. It enables a change mechanism of a button-activated automatic mechanism that changes the state of the car from closed-roof to open-roof. This mechanism, whether executed or not, may come with a carrying cost of increased fuel consumption or decreased storage volume. It may have execution restrictions that are context-based, like the car being in the "parked" position, or needs-based, like the operator desiring open-roof only in warm weather. The execution of the change mechanism will come at a cost like power (and therefore fuel) or time. This execution is reversible and can be executed repeatedly until the mechanism wears out or breaks. More examples of change mechanisms can be found in Table 3-3.

Table 3-3. Examples of Change Mechanisms

System	State Change	Mechanism	Agent	Ility
Convertible	Closed-roof to open-roof	Button-activated automatic mechanism	Internal driver	Adaptable
Military Aircraft	Adding external fuel tanks	Universal fittings for tanks/missiles/etc	External crew	Flexible
V-22 Osprey	Vertical to horizontal flight modes	Rotating mechanism on rotors	Internal pilot	Adaptable
Parachute	Packed to deployed	Pulling release pin to deploy pilot chute	External parachutist	Flexible
Parachute	Packed to deployed	Explosive activated to pull release pin	Internal emergency AAD pull device	Adaptable
BMW production plant	Change between 5, 6, or 7 series production	Similar joining sequences and load carrying points for assembly	Factory management	Flexible
HVAC system	Change temperature	Activating heater or air conditioner	Control thermometer	Adaptable
BMW production plant	Change in car order configuration	Build-your-own vehicle online service	External buyer	Flexible

The conceptual flow from design principles to ilities in Figure 3-21 captures how systems can be intentionally designed to change. However, in order to be a more thorough construct, that flow can be generalized to capture systems that are designed to not-change (Figure 3-24). In this sense, a system might use a design principle to generate a path inhibitor in a system. A path inhibitor, the "no-change" counterpart of a path enabler, limits the use of a change mechanism or enables the use of a resistance mechanism. *Path inhibitors* give the system the option to execute a resistance mechanism. This resistance mechanism then relates to certain ilities that use "no-change" strategies, like robustness or survivability, which can be desirable properties in a system.

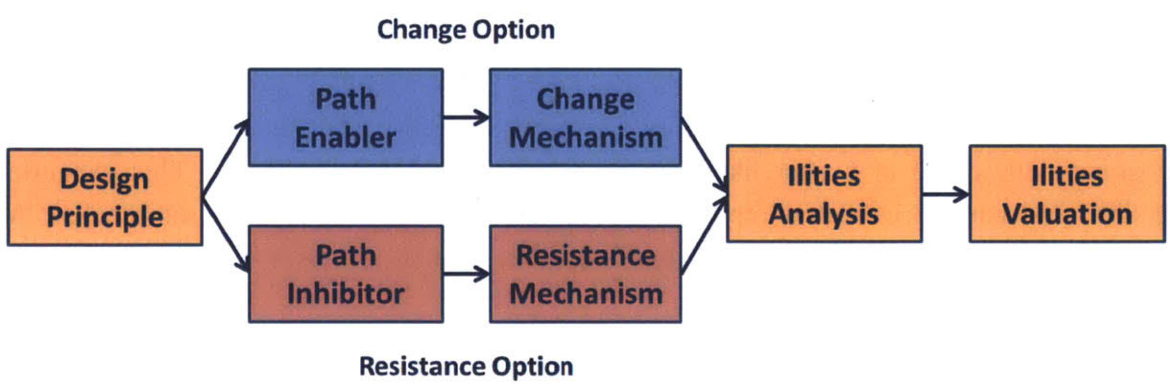


Figure 3-24. Relationship between Design Principles and Ilities (including both Change-related and Resistance-type)

Another important consideration in this framework is whether the change options or resistance options are executed voluntarily or involuntarily. Earlier, the relationship implied a voluntary action by the system owner, or designer. However, a similar construct may also be applied to involuntary events stemming from uncertainty, possibly system antagonists, stakeholders, or even Mother Nature (Figure 3-25). These involuntary events relate back to the discussion of perturbations that can apply to the design, context, or needs spaces. These perturbations can force the instance in any of these spaces to another instance or they can change the space directly.

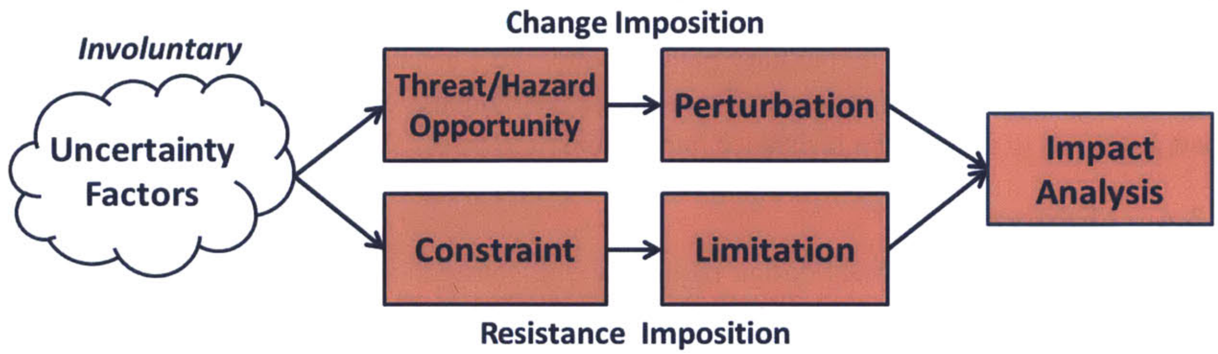


Figure 3-25. Involuntary Change and Resistance Impositions

Figure 3-25 shows the involuntary paths in a similar construct of the design principle-ility relationship. Figure 3-24 represents voluntary changes and includes the original wording of components since the general point of view taken was implied to be voluntary system changes. However, changes made to systems can be involuntary; that is, imposed, and these changes or resistances are outlined in Figure 3-25. Note that voluntary changes are not necessarily "good" and involuntary changes are not necessarily "bad," although this may often be the case. There are cases where an involuntary change may be beneficial to the system. For example, an automatic software update can be an involuntary change, pushed out from an external source, which improves the software of the system. This involuntary system change could be seen as an *opportunity*. On the other hand, negative instigators to involuntary change are seen as *threats* and *hazards*. Mekdeci et al. (2012) defines *threats* as external conditions that exist that may cause a perturbation and *hazards* as internal conditions that can cause a perturbation. Threats, hazards and opportunities are involuntary path enablers that may lead to a perturbation (here interpreted as involuntary change mechanisms) that can have negative or positive effects. Generally, these effects will be negative, as most system stakeholders want control over any changes made to the system, but there are cases of positive effects as previously discussed.

Path inhibitors can be voluntary or involuntary as well. A system might want to use path inhibitors and resistance mechanisms to minimize undesirable change from a perturbation. One can view resistance options as intentional countering of change impositions. For example, in reducing vulnerability to small arms fire, in order to achieve survivability of an Army vehicle, armor can be added as a path inhibitor to create a resistance mechanism of withstanding damage from incoming projectiles. Additionally, there can be an involuntary path inhibitor, or *constraint,* that leads to a limitation of a system as well. For example, a new regulation (constraint) may involuntary restrict the allowable states (limitation) available to a system (a homeowners association limiting the exterior paint colors that may be used in a particular neighborhood).

This construct shows how a system designer may use path inhibitors to reduce the impact of involuntary path enablers like hazards or threats. Conversely, a system designer may use path enablers to help deal with involuntary path inhibitors like constraints. More broadly, a system may develop resistance mechanisms to manage involuntary change impositions or develop change options to circumvent resistance impositions[5][6].

3.3.4 Evolvability

The early phases of conceptual design require careful consideration as early decisions will have substantial influence on the new system, ultimately enabling or limiting success of the system over time. In the face of changing contexts or needs, *epochs*, systems can be designed to change in response, or remain robust or versatile, in order to retain useful functionality to avoid suffering deficiencies and even failure. Designing an evolvable system may reduce the long-term cost of system upgrades or replacements in the presence of epoch shifts over its lifespan. Cases of 'clean-sheet' design in engineering systems may take more development time and lead to higher costs. Evolutionary design starts from an existing design, rather than a blank slate, and is an increasingly common trend; for instance, nearly 85% of GE's products are modifications of previous products (Holtta-Otto 2005). The US Air Force has declared that its acquisition of new systems will primarily involve evolutionary methods (Wolfowitz, 2002). In industries where redesign is the norm, evolvability clearly is a desirable trait.

Evolvability is a design characteristic that facilitates more manageable transitions between system generations via the modification of an inherited design. This thesis pursues descriptive (empirical-based) approaches to determine initial design principles for evolvability. Contrasting biological and technological evolutionary processes yields insight into possible design principles.

---

[5] This thesis initially holds the strategy discussed, designing change options for bypassing resistance impetuses and resistance options for inhibiting change impositions, as the standard relationship. However, it may be possible that change options can be used to combat both change and resistance impositions as well as resistance mechanisms being useful for combating change and resistance impositions. This thesis however will not dive too deeply into how this construct may be used, but will focus more on describing the relationship that may be present.

[6] This relationship of voluntary versus involuntary actions was discussed from the "positive" system perspective. That is, the system in question is operating and being exposed to perturbations and constraints from uncertainty factors. It is possible, however, to invert the relationships and examine the scenario from the perturbation or constraint perspective. If the aim is constraining or perturbing another system, similar techniques discussed in this section may be reversed to find "design principles" for achieving that aim.

3.3.4.1 Evolvability Defined

A meaningful definition of evolvability must be broad enough to apply to systems in multiple domains, yet free of unnecessary ambiguities. The proposed definition was initially informed by the biological perspective, the originating domain of evolution-related concepts. Additionally, applications from other fields such as systems and computational engineering were also considered. Beesemyer et al. (2011) proposed a definition of evolvability as:

The ability to change an inherited design across generations [over time].

The key aspects of the definition include some threshold amount of change in the system has occurred, and the new system is based upon or has 'inherited' part of its design from a prior 'generation.' This change between generations will occur through some mechanism of variation and selection. In Figure 3-26, the inheritance from prior generations of the system is shown as vertical connecting lines going into the design of the new system generation. It should be noted that the older generation system may continue to operate in parallel with newer generations of the system, and that inheritance may come from any prior generation.

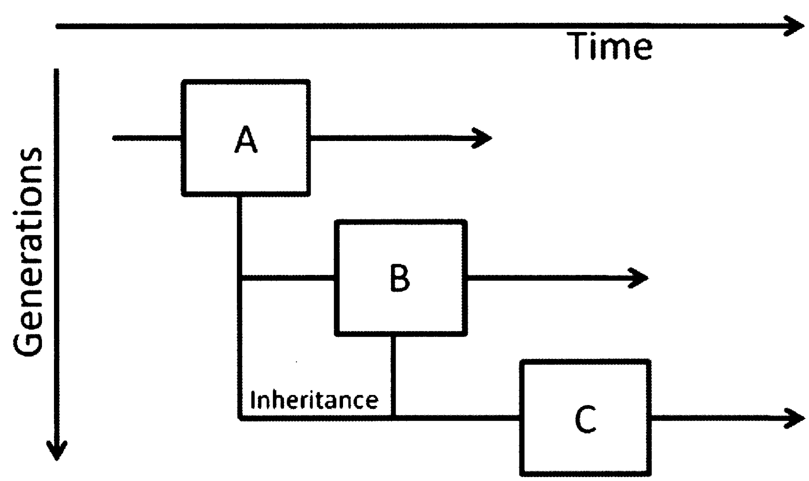


Figure 3-26. The Evolutions of Systems and Systems of Systems Over Time

This initial definition of evolvability included the main components of this idea: some level of inheritance and a generational change. Biological evolution results in a new generation from inherited, but altered, code of a prior generation. Evolvability in engineering systems then is a subset of changeability where the change is occurring not to the actual system, but at some higher generational level. An architecture-design-system construct can help clarify this level of system abstraction. Mekdeci et al. (2011) defines an architecture as consisting of two core elements: 1) what the system is composed of, known as the *operational elements* and *components* (i.e. the form), and 2) how the system operates, known as the *Concept of Operations*

(CONOPS) (i.e. the operations). A system architecture is comprised of one or more *system designs*. A system design is a particular chosen set of operational elements and CONOPS within a given system architecture. The instance of this design is the actual physical instantiated system. This relationship is best understood with an example. Table 3-4 summarizes three examples of these relationships with an iPhone, an aircraft carrier, and the International Space Station (ISS).

Table 3-4. Architecture-Design-System: Level of Abstraction Construct

Concept	 iPhone	 Aircraft Carrier	 Int. Space Station
Architecture	[iPhone 3G/ iPhone 4/...]	[Nimitz/Ford/ Forrestal/...]	The International Space Station
Design	[4/4S; Black/White; 16GB/32GB/64GB]	[USS Ronald Reagan/ USS Harry Truman/ USS George Bush/...]	
System	Clark's Black 32GB iPhone 4S		

In Table 3-4 the levels of abstraction construct of the system are explained for three different systems. These three cases differ in the way they relate to the levels of abstraction. The concept of an "iPhone" could be defined by an iPhone 4 architecture. This is a high-level form, function, and CONOPs mapping that includes all of the validated configurations to which an iPhone 4 can be manufactured and operated. The architecture includes only "validated" configurations since Apple only guarantees expected performance within that architecture. If the phone is altered or operated in a different manner than intended (under water or "hacked" for example), Apple does not guarantee normal system performance, and there is a departure from the architecture. The iPhone 4 designs are instances of the iPhone 4 architecture. Different designs could be the iPhone 4 or iPhone 4S, black or white, and have various levels of storage capacity (16G, 31G, or 64G). An iPhone system is an instantiation of the design, or literally a specific person's iPhone.

Similarly, an aircraft carrier can be separated into the levels of abstraction. However, as in this case, it is possible for this construct to degenerate into cases where an entity can fall into multiple levels of abstraction. The architecture of aircraft carrier could be the different classes of carriers, Nimitz-class or Ford-class. Within these classes there may be multiple ship designs. These ships are not completely the same even though they were built from the same architecture. Each ship has its unique designs at some level, and each has only one built, meaning they are all instances as well. In this case, the various designs of the Nimitz class carriers are instances of the architecture and are each instances of their unique design as well.

Further degeneration can take place when a system is completely singular, like the ISS. The ISS has an architecture, with a design, and only one built system. In this case, the ISS can be seen as all three levels of abstraction. Any change made to the system will be a change in the design and a change in the architecture as well.

Changeability can take place at any of these levels of abstraction. However, changes at different levels of abstraction may relate to other ilities as well. Mekdeci (2011) defines the set of designs within an architecture as the *pliable* set of validated designs. In this sense, pliable changes occur at the design level and could also be thought of as design-level changeability, or *pliability*. Evolvability then occurs at the architecture level, where changes are made to the architecture that may not be validated. Evolvability can then be thought of as architecture-level changeability.

This distinction requires an update to the previous definition of evolvability. Evolvability may now be defined as:

The ability of an architecture to be inherited and changed across generations [over time].

This captures the important aspects of evolution previously discussed as well as showing the proper level of system abstraction in which the change occurs (architecture). In biology, we might make the comparison for a sanity check (Table 3-5). Evolution by natural processes takes place at the genetic level of a population. A populations' collective DNA may be analogous to a system architecture in that DNA encodes the instructions from which the human system is created. The population represents the pliable set within that architecture. Each person's individual DNA is an instance of the architecture at the design-level where the person himself is the instance of the design. Therefore, the changes to the architecture change the possible designs and instances of design, or individuals. These changes are evolution and occur at the population genetic level, or architecture.

Table 3-5. Biological "Sanity Check"

Levels of Abstraction	Biology	Technology
Population-level construct	Human Genome	Architecture
Blueprint for the individual	Individual's DNA	Design (Instance of Architecture)
Specific "built" individual	Individual	System (Instance of Design)

3.3.4.2 Evolvability in Biology and Technology

As the definition for evolvability was initially informed by the biological perspective, exploring characteristics of biological evolution may be beneficial in developing initial design principles for engineering systems.

Darwinian evolution, as summarized by Maynard Smith (1989), requires populations of entities to have three properties: (1) the ability to multiply; (2) variation of characteristics within the population; and (3) some level of heritability with the variations. Natural selection is the emergent behavior of populations with such principles. Variations are passed to future generations through genetic code in an entity's DNA, or genotype, and their specific realizations, or phenotypes, are selected on by the fitness function of their environment (Ziman 2000). These variations are generally created through some means of recombination and mutation. If these variations turn out to give some sort of competitive advantage to an entity, it is more likely that the new genotype will be carried into future generations. Thus over time, favorable characteristics are passed on, and the species evolves. It is important to note that these variations are random, and generally not induced by their environment, and that the variation-selection mechanism acts on the population (Ziman 2000).

While there are currently much greater levels of complexity to the biological basis for evolution, the Darwinian view of variation and selection is remarkably still applicable to the natural evolution that has been going on for millennia. This basic framework of biological evolution seems to relate to the engineering systems' design process of concept creation and selection.

Over time, life has developed and refined alternative means for implementing variation and selection. Most notably, the emergence of sexual reproduction, a much more complex means to reproduce, has increased evolvability due to its effective capacity to create increased variation. Applying the concepts of evolution to engineering systems can be informed by and improved by these mechanisms of variation and selection that have emerged in nature. Kevin Kelly has done a large amount of work comparing the mechanisms of biological evolution to those of technological evolution (Kelly 2010). As biological evolution shapes how "living systems" change over time across generations, technological evolution shapes how "engineering systems" can change over time across generations. Kelly's study of technological evolution provides a descriptive basis for this comparison of natural "systems" evolution to engineering systems evolution.

3.3.4.3 Forces of Evolution in Biology and Technology

If the goal of designing for evolvability is to improve how a system can evolve through generations, understanding what forces drive evolution in biology and technology is an important first step. Figure 3-27 is a representation of Kelly's triad of forces that impact evolution in life and in technology. Each domain has three legs on which evolution stands. The

Historical/Contingent leg is the 'luck', or the happenstance of a species. Since variation is random, speciation comes from improbable triggers in the past that leads a species down a contingent path (Kelly 2010). The second leg of the triad is the Structural/Inevitable. This force drives the macro-level structures in biology and technology. Where the contingent force is historical, this force is ahistorical. Corresponding to the theory of convergent evolution, this force would be responsible for similar forms independently developed seen today. This force is based on the laws of physics in the world that yield general truths like air density affecting flight or fluid displacement affecting marine life and technology. In biology, an example of this might be the convergent evolution of the wing, where technology might show this principle in the form of hemispherical boats. While the causes of convergent evolution are debatable and controversial in biology, it is more important to see it as an emergent behavior of evolution and a useful tool in analyzing the evolvability of systems.

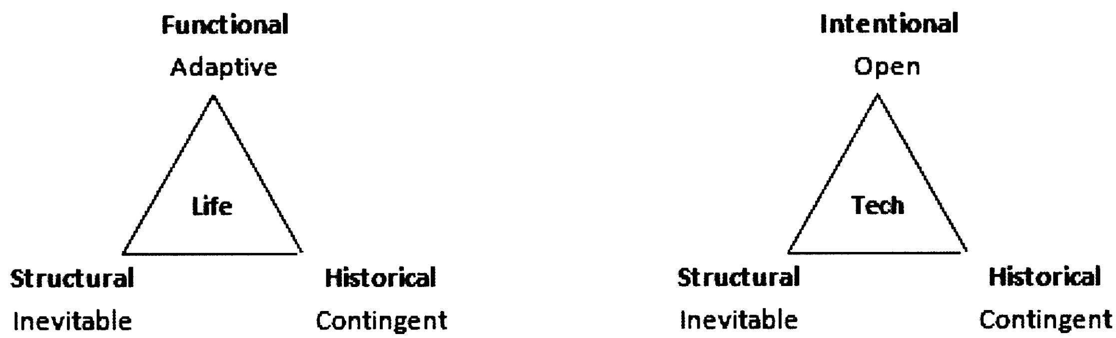


Figure 3-27. Triad of Forces Driving Evolution in Biology and Technology (Kelly 2010)

The Adaptive/ Functional leg is the fundamental, orthodox force discussed earlier. Those stronger species that are able to reproduce are naturally selected to continue. This force however, is where life and technology may differ. While biology relies on the mechanism of BVSR (Blind Variations Selectively Retained) (Ziman 2000), technology, and more specifically, a system, can be deliberately designed and implemented by the Intentional/Open force.

In the third force, possibilities are expanding in time. This intentional part of Kelly's triad is the optimization engine of technology. The intentional mechanism differs from life's natural selection engine in that it may be completely conscious, driven by free will and choice. This is why technology, once given the tool of the human mind, or an intelligent designer, has evolved so much faster than life on Earth (Kelly 2010).

Kelly's construct is a good starting point in comparing natural evolution to technological evolution. As Kelly intended to inform modern innovation by following evolutionary trends in the past, engineering systems may find value in early concept and design phases by effectively implementing principles derived from natural evolution. Designing an evolvable system requires the daunting task of predicting trends, or evolutionary pathways, and designing for future contexts and needs in order to maintain value to system stakeholders over time (Ross 2008).

3.3.4.4 Characteristics for Technological Evolution

In exploring different domains' ideas on the relationship between technological and biological evolution, common characteristics of evolutionary mechanisms emerged for technology that biology does not utilize as well, if it all. These observed aspects of technology may lead to initial design principles that could be applied to engineering systems in order to lower the required effort to move from one generation to the next.

Non-sequential Inheritance. Biological evolution requires sequential improvements passed down from parents to children (the next generation). Small changes that improve fitness are passed down to offspring and so on. DNA may be seen as a transcript of evolution over millions of years; however, each new population can only pull traits from their respective parents. Populations cannot directly pull from their distant relatives, many generations earlier. The evolution of technology, however, does not have this chronological constraint. Since most technology is recorded in patents and journals, for example, any new system can pull from any generation before itself, all the way back to the inception of the technology. A design principle for evolvability based off this advantage may then be *leverage ancestry*. Systems being upgraded, or changed with inheritance, between generations can and should do research into successful methods from all prior generations, not just the last.

Evolutionary Leaps. Living systems have slowly and incrementally evolved to improve fitness over time. Since more extreme variations generally lead to life that is not viable, biological evolution favors small changes between generations. In technology, however, there is no requirement of small, incremental steps in evolution. Transistors did not slowly evolve from vacuum tubes one small piece at a time. Technology is often characterized by fewer, but larger evolutionary steps between generations. A design principle for evolvability based off this technical advantage could be *disruptive architectural overhaul*, similar to the disruptive innovations discussed in Henderson and Clark (1990). Taking out large parts of a prior generation and completely revamping could be more beneficial to a system than trying to alter components slowly. This type of design principle is often seen in car manufacturing in that small changes are made to vehicles on a yearly basis, but every few years a larger, more complete overhaul is accomplished. This type of change can reinvigorate a "dying" design while still keeping aspects that were successful.

Conscious Variation and Selection. Technology has the benefit of having an 'intelligent designer.' While biology relied on natural selection to slowly find optimal solutions over millions of years, technology may produce systems that never could have naturally evolved with biological rules for reproduction. While technology may benefit from some level of random variations, the ability to channel or 'push' these variations into useful system concepts improves evolvability. Biology limits the inheritance of offspring not only to the previous generation, but also to the same species (the parent species) (Figure 3-28).

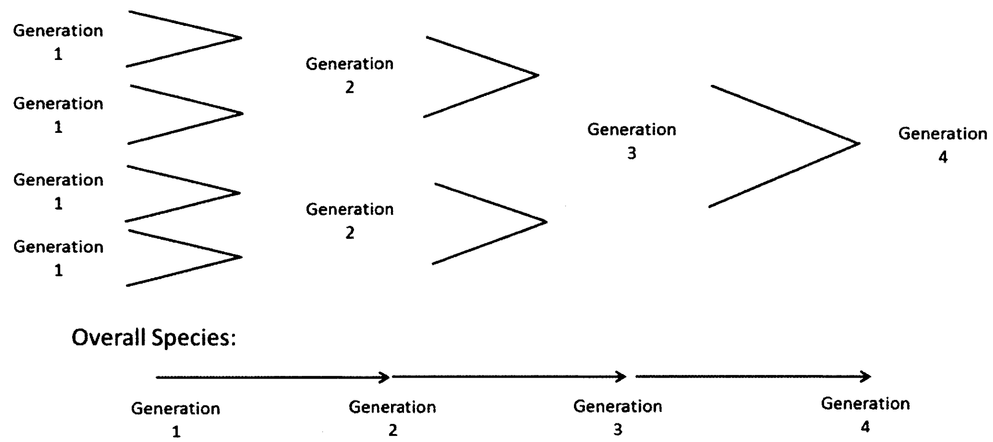


Figure 3-28. Inheritance Structure of Biological Evolution

Technology, however, can pull from any domain or 'species' in creating new 'life' or a new system. A principle or design from aerospace can be applied to automobiles or trains or vice versa. Any one system's success can be applied to benefit another system in the future. Figure 3-29 shows how systems may use this principle in their evolutionary path between generations.

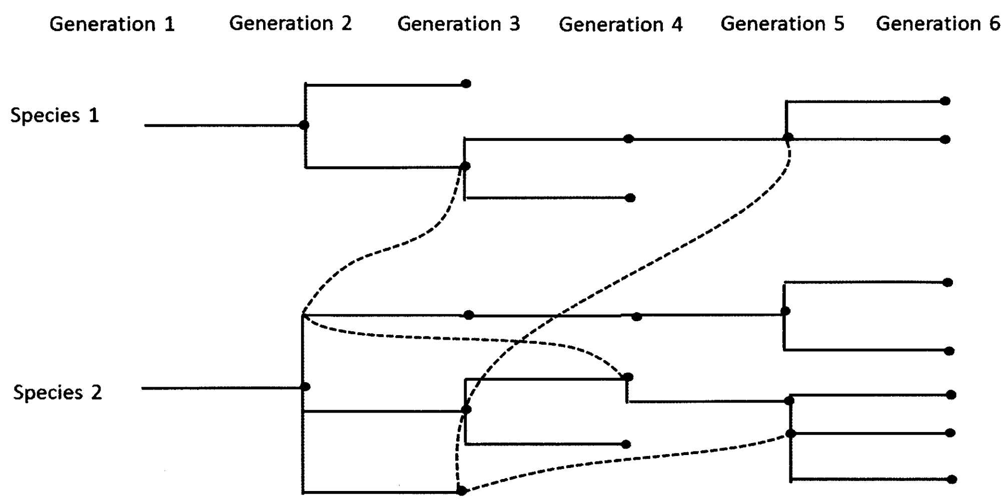


Figure 3-29. Inheritance Structure of Technological Evolution

Technology is not vertical, as in the family trees found in biological lineage. It can use lateral jumps across 'species' or generations to reuse or repurpose old ideas. Inheritance may occur from different systems or different generations at any time in development. Successful solutions may be shared across time and domains. A design principle for this advantage in technology could be *mimicry*. Re-architected systems should investigate successful solutions in other systems and other domains. This also aligns with the "modular innovation" pathways discussed in Henderson and Clark (1990).

Exaptations. Exaptations are inadvertent inventions that give value in a repurposed role. These are rare in biology. For example, feathers were first used in biology as a means of thermal control, but in time, became very useful in another task, flight, adding another competitive advantage to participating species. Basically, this is changing the function of a feature to a new function not intended in the original design. In technology, exaptations are common and very useful (Kelly 2010). Inadvertent discoveries in one domain can translate to more utility in another domain. An example of this is the Guttenberg printing press, originally invented using technology from wine presses. Guttenberg combined a moveable type printing method from the Chinese with a wine press for pressing grapes to create the first printing press. *Resourceful exaptation* then may be a design principle for technological evolution.

3.3.4.5 Candidate Evolvability Design Principles

Modularity is a design principle that is beneficial for many more reasons than just evolvability. The concept can be traced back all the way back to evolutionary biology, where data shows that the number of traits a variation affects is inversely related to its likelihood of being selected (Hansen 2003). The concept applies in the same sense to systems engineering; if a system is designed into modules such that a proposed change to a system only affects one module, potentially much less redesign will be required than the same change being implemented on a highly integral system. Modularity is not always good however; Holtta-Otto (2005) points out that designing for modularity is accompanied by the potential for over-design and potentially inefficient performance. As mentioned earlier, being optimized to only one design point (efficient performance) does not allow for a system to thrive in uncertainty. Only a certain degree of modularity might be needed. For instance, if a system has components A, B, C, and D, but technology and requirements concerning A and B are constant and show no signs of changing, leaving A and B coupled will not hinder evolution.

In their paper on designing for changeability, Fricke and Schulz (2005) list several extending principles for enabling changeability. Despite being applied to changeability, many of these principles appear to also be applicable to evolvability. The first principle is *integrability*, which is characterized by compatibility and common interfaces. This goes hand-in-hand with modularity; modules are only as good as the interfaces through which they interact. *Scalability* is another one of the extending principles that applies to evolvability, which can apply to either a single parameter or the entire system. The range of a parameter's scalability is determined by the capacity of the rest of the system to accommodate the change. The third applicable design principle proposed by Fricke and Schulz (2005) is *decentralization.* This principle calls for distributing resources to appropriate locations, rather than having them located at a single place. Decentralization, like modularity, aims to minimize change propagation. The final applicable design principle mentioned in their paper is *redundancy.* Redundancy allows for more constant performance and functionality in the face of potential faults or failures. Anticipating where these failures might occur "facilitates an identification of objects/units likely to be affected by architectural evolution" (Fricke and Schulz 2005). Redundancy and modularity can be very powerful together, because it potentially allows a module to be taken away without the system losing a critical function.

Reconfigurability is a design principle explored extensively by Siddiqi and de Weck, who claim reconfigurability aids evolvability through "[enabling the system to change] easily over time by removing, substituting, and adding new elements and functions" (Siddiqi and de Weck 2008). Siddiqi suggests two design principles that lend themselves well to designing for evolvability: using self-similar modules and maximizing information reconfiguration. *Self-similarity* can enable radical change and utilize the same components to achieve a very different function. *Maximizing information reconfiguration* is based on the fact that changing an informational structure is almost always less costly than physically reconfiguring a system or redesigning physical components.

From the advantages of technology over biology, we can derive a few more candidate design principles for evolvability that are less architecture quality principles, but more re-architecting heuristics for evolvability. *Leverage ancestry* allows system architects to look back at all prior generations, not just the immediate preceding generation for design choices that may be successful in modern environments; what once worked in the past may work again in the future. *Mimicry* allows system designers to examine generations of other systems and other domains; what worked for submarines may work for aerospace. *Disruptive architectural overhaul* is a design principle that designers can use when re-architecting a system to intentionally disrupt their architecture, possibly enabling radical new capabilities within entire aspects of the design. Car manufacturers disruptively overhaul their models every three to five years—possibly other systems could profit from similar changes. *Resourceful exaptation* can be an effective way to capitalize on features designed for a function in one system and serve as a different function in the system being evolved. These principles were indirectly derived from Kevin Kelly's (2010) work on technological and biological evolution.

The candidate design principles for evolvability in engineering systems are summarized in Table 3-6 below. Listed are the implications for evolvability and the sources from which the candidate principles were derived.

Table 3-6. Candidate Design Principles for Evolvability

Design Principle	Implications for Evolvability
Targeted Modularity	Limits change propagation (Hansen 2003) (Holtta-Otto 2005)
Integrability	Compatibility and common interfaces (Fricke and Schulz 2005)
Scalability	Of a parameter or entire system to meet new needs (Fricke and Schulz 2005)
Decentralization	Distributed resources to limit effect of changes (Fricke and Schulz 2005)
Redundancy	Gives flexibility to designer to eliminate components (Fricke and Schulz 2005)
Reconfigurability	Self similar parts and maximizing information reconfiguration (Siddiqi and de Weck 2008)
Leverage Ancestry	Successful design choices from all prior generations (Kelly 2010)
Mimicry	Successful design choices from other systems/domains (Kelly 2010; Henderson and Clark 1990)
Disruptive Architectural Overhaul	Upgrading large aspects of architecture at a time (Kelly 2010; Henderson and Clark 1990)
Resourceful Exaptation	Repurposing successful design choices from other systems (Kelly 2010)

Evolvability may be an effective concept that describes the ability of an architecture to be modified across generations in the presence of changing contexts, or needs, allowing for the potential to deliver more value over the course of a family of systems' lifetime. The evolvability design principles described in this thesis, including modularity, integrability, scalability, decentralization, redundancy, re-configurability, (achieved through self-similarity and maximizing information reconfiguration), leverage ancestry, mimicry, disruptive architectural overhaul, and resourceful exaptation may be applied to lower costs or to increase utility when re-architecting systems.

3.3.5 Summary

This chapter outlines high-level strategies for sustaining value throughout system lifecycles. The concept of value sustainment has been framed with respect to perturbations that systems will inevitably encounter throughout their lifecycles. Systems can either dynamically change in response, or statically remain robust or survivable to shifts or disturbances in design, context, or needs. System designers and operators can control the design space and attempt to influence the context and needs spaces in order to affect the performance space, and therefore the value space.

This research focuses on systems that change or resist imposed change in response to perturbations in order to maintain value delivery. These change-related responses may be facilitated through design principles that lead to path enablers or path inhibitors in the system. The path enablers and inhibitors relate to change mechanisms or resistance mechanisms respectively. Together, these form change and resistance options that system stakeholders can use to exploit change opportunities or combat undesirable imposed changes. The change and resistance options that a system employs relate to its ilities, that is, desirable system properties that the system displays in order to respond to perturbations.

The concepts outlined in this chapter describe the different ways systems can respond to the dynamic environments in which they operate. In order to look towards the future, and learn to design systems that perform well in the dynamic environments described earlier, one may examine historical systems that have experienced perturbations. Empirically looking back on case studies can be an effective way to form prescriptive knowledge for future systems.

4 Epoch-Shift Case Analysis: A Space Systems Perspective

Operating systems in space, with its associated "high-ground" perspective, enables capabilities otherwise not available or possible using terrestrial systems. Space provides an opportunity for platforms from which systems can affect lives on a global scale through enhancements to navigation and timing, communications, and remote imagery. These enhancements affect not only government agencies, but also civilian consumers as well, with such systems as GPS, imaging, satellite radio, and TV. While enabling the use of such technologies, the development and operation of space systems comes with a high cost. Manufacturing precision, experimental technology, international cooperation, vehicle launch, on-orbit operations, regulation compliance, and many other factors generate high risk and high costs for most space systems. These systems tend to be very complex and expensive, often operating in unforgiving environments for large amounts of time. The impact of changes in these environments and the ability of a system to effectively respond to these changes could mean the difference between success and failure.

4.1 Overview

Using the concepts of value sustainment in the face of perturbations, such as shifts in context, investigating historical cases of systems may shed light on strategies that systems may use in the future by looking at successful and unsuccessful strategies of the past.

4.1.1 Epoch-Era Analysis

Traditional systems engineering and acquisitions approaches tend to focus on meeting technical requirements, as these are easier to verify and manage than some of the more non-technical uncertainties. However, failing to allocate enough time to proactively consider changes in requirements, or to account for non-technical uncertainties, may leave a system vulnerable in dynamic environments with changes occurring in both technical and nontechnical factors. This environment, or operational context of the system, and the set of stakeholder needs, can change over time. Briefly reiterating concepts from Chapter 3, an "*epoch*" is a period of time, defined by a fixed set of context and needs, which impact the ultimate success of a system (Ross and Rhodes 2008a). A long-lived system may face a large number of epochs over its lifetime. Since the goal of any system is to meet its needs in various contexts, delivering benefit at cost—or value—across changing epochs is a measure of success as perceived by individual stakeholders of the system.

By the very nature of traditional acquisition phase-based lifecycles, system designers are required to make key design choices early in the system lifecycle that will impact the ultimate operational performance of the system in alternative future contexts, which may or may not match the original designed-for, anticipated context. A value robust system is one that maintains value delivery in spite of shifts in the system, contexts, or needs (epochs). Discussed in more detail in Chapter 3.1, these shifts in system, needs, or contexts are referred to as "*epoch shifts,*" with a system's success depending on how well it responds to shifts across its lifecycle as seen in Figure 4-1.

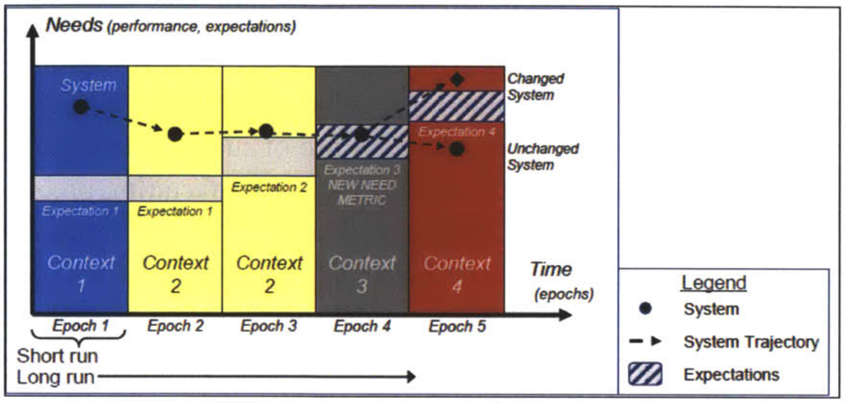


Figure 4-1. Epoch-Era Analysis (EEA) (Ross and Rhodes 2008a)

4.1.2 Epoch-Shift, Impact, Response, Outcome

Using a construct similar to the Epoch-Era Analysis representation presented above (Figure 4-1), system case examples can be discussed using the Epoch-Shift—Impact—Response—Outcome construct as seen below in Figure 4-2. The figure describes how a system may be operating at an acceptable level of performance in Epoch 1 and then experience an *epoch shift*. After experiencing this imposed shift in system, context or needs, the system may display some degradation in performance, known as the *impact*, possibly bringing performance below expectation levels. The system then, in order to recover to an acceptable performance level, may initiate a *response*, which then results in some *outcome* for the system.

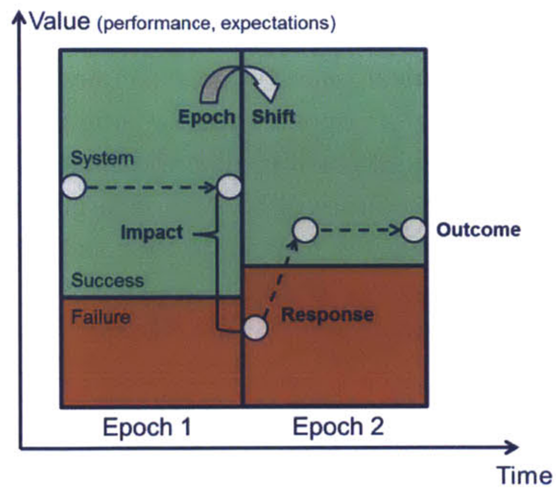


Figure 4-2. Epoch-Shift - Impact - Response - Outcome Construct

A simplified example of this construct can be visualized in Figure 4-3. One can imagine a space environment in which a satellite is operating at expected performance levels. Some shift in context occurs, possibly the discovery of incoming debris outside of collision risk tolerances. This increase in risk impacts the performance of the system if stakeholders deem that risk unacceptable. System decision makers can choose to respond to this shift, possibly by adjusting the orbit of the satellite by fire onboard thrusters. This change in orbital parameters may result in an outcome where the satellite is within acceptable risk levels, allowing the system to operate within stakeholder-defined expectations. If the satellite was unable to respond, it is possible the system could collide with the debris, resulting in a failure of the system. This is a simplified example just to illustrate the use of the construct.

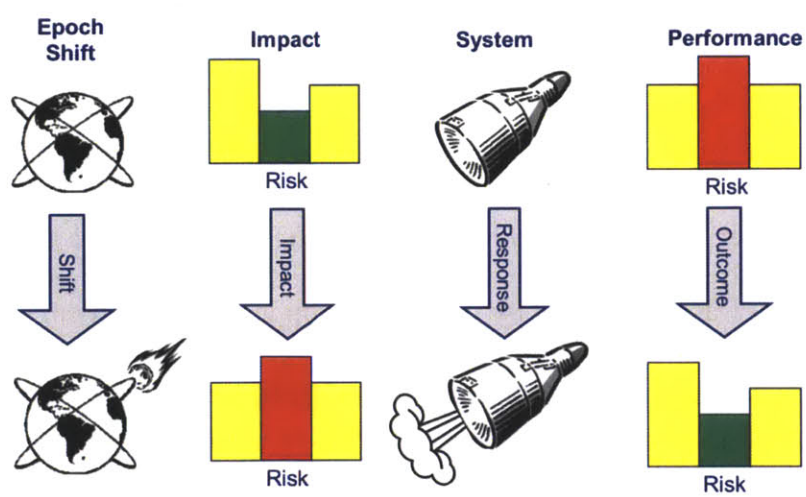


Figure 4-3. Simplified Epoch Shift--Impact--Response--Outcome Example

4.2 Case Studies

This section will present a set of historical space system examples of epoch shifts, the impacts of these shifts on the systems, the system's response, and the ultimate outcome of these responses on each system's success. The epoch shifts will be described in terms of exogenous uncertainty. Patterns of response across the case studies will be studied to give insight on possible impacts from intentionally designing in the ability to respond to such shifts during system operations or earlier. Lifecycle properties regarding the ability to respond, also known as *"ilities"*, allow systems to react to potential epoch shifts during system design, implementation, or operations. The number of cases in this thesis will help to consolidate historical examples of epoch shifts and seek to identify preliminary patterns of responses and impacts.

4.2.1 Iridium

The Iridium story anecdotally began in 1985 when a Motorola engineer's wife complained about an inability to reach clients while vacationing in the Caribbean (Finkelstein and Sanford 2000). In short time, Motorola announced this new concept for Iridium as "a *global communications system that will allow people to communicate by telephone anywhere on Earth – whether on land, at sea or in the air – via portable cellular radiotelephones operating as part of a satellite-based system*," (Fossa 1998). Numerous companies, including Motorola, Kyocera, Lockheed Martin, and Raytheon, helped to design and develop the Iridium system. The system attempted to make space communications viable by using Low-Earth-Orbiting (LEO) satellites (allowing for smaller handsets and no voice delay compared to Geostationary Earth-Orbiting (GEO) satellites).

Almost a decade after the concept of the system was envisioned, detailed design of the Iridium system took place. The system itself is comprised of 66 cross-linked satellites (plus 6 in-orbit spares) at a 778 km altitude in 6 polar planes with 11 satellites in each. Iridium represented one of the biggest technical achievements in space communications in the 1990's, delivering more than 1,000 patents (Finkelstein and Sanford 2000). The system includes inter-sat links, ground control facilities, terrestrial Earth gateways (to interconnect with telephone networks), and the actual mobile phones and pagers (Fossa 1998). By 1998, when the system became operational, the system had cost over $5 billion to build and maintain (Finkelstein and Sanford 2000).

Iridium as a company was represented by very strong top leadership and engineering teams, and was focused on solving many of the technical requirements of getting a LEO communication constellation to function properly (Finkelstein and Sanford 2000). In 1998, the company started a $180 million marketing campaign and even had Vice President Al Gore make the first Iridium phone call. The company commenced commercial operations with a $3,000 handset cost and $3-$8 per minute calls and the results were devastating. After two quarters, the company had only 10,000 subscribers, and only 20,000 subscribers almost a year into operations, far less than the projected 500,000 users. The company needed 52,000 customers to meet loan interest payments and was unable to make necessary payments, defaulting on $1.5 billion in loans. On August 13, 1999, Iridium filed for Chapter 11 bankruptcy and became one of the 20 largest bankruptcies in U.S. history (Finkelstein and Sanford 2000).

The company, planning to de-orbit the satellites, instead sold and restructured the company for just $25 million and became Iridium LLC. Iridium ended up selling for just about ½ a percent of the initial investment, and began new operations with $72 million Department of Defense (DoD) contract (de Neufville and Scholtes 2011).

Finkelstein and Sanford (2000) argue that the company failed due to three main reasons:

1. An escalating commitment, particularly among Motorola executives who pushed the project forward in spite of known potentially fatal market and technological problems.
2. Iridium CEO's reluctance to cut losses and abandon the project due personal and professional reasons.
3. The Iridium board structure that prevented it from performing its role of corporate governance.

A highlighted flaw in this report is the fact that Iridium executives knew about the risks in the market. A 1998 prospectus listed 25 full pages of risks (Finkelstein and Sanford 2000):

- a highly leveraged capital structure
- design limitations – including phone size
- service limitations – including severe degradation in cars, buildings, and urban areas
- high handset and service pricing
- the build-out of cellular networks
- a lack of control over partners' marketing efforts

Evidence shows that even though Motorola knew about these risks, no effort was effectively made to address or mitigate these risks. The belief that the innovative technological advances of the system would carry the company did not come to fruition. While the company saw lots of hype and excited initial investors, its launch was underwhelming, put best by CEO John Richardson:

> *"We're a classic MBA case study in how not to introduce a product. First, we created a marvelous technological achievement. Then we asked how to make money on it."* –
> Iridium Interim CEO John A. Richardson, August 1999 (Finkelstein and Sanford 2000)

The company had miscalculated the terrestrial cellular network development throughout Iridium's design and testing in the 1990s. The potential customers began to value small handsets, coverage indoors and in cars, and reasonable prices. The Iridium handset was costly, large and heavy, and it did not work well indoors or in cars.

Applying this case to the Epoch-Shift—Impact—Response—Outcome construct we can characterize these events in a structured manner. The *epoch-shift* the company undergoes is a gradual shift from the conception of the system in 1985 through its design in the 1990's. The system began in a context where there was a need for global communications from a portable handset with limited voice delays. This context did not include a developed terrestrial cellular infrastructure. Throughout the 1990's, the need for global communication did not change, in fact it increased. However, with increasing terrestrial cellular developments, consumers' needs shifted to lighter, affordable devices that could work everywhere, including indoors and in cars. This *impacted* Iridium by moving the system into an unaffordable, too heavy, non-functional (indoors and in cars) performance space from the perspective of consumers. Iridium *responded* to this impact by continuing with original plans, not changing much at all. Iridium launched the complete constellation and initiated service with $3,000 handsets that were large and heavy and cost about $3-8 per minute to operate. The *outcome* Iridium experienced was a lack of user subscriptions, defaulting on loan commitments, and the ultimate bankruptcy of the company and possible de-orbiting of the constellation. Either this characterization shows how Iridium miscalculated the *impact* of the *epoch-shift*, resulting in an improper *response*, or they may have understood the impact correctly, but simply did not respond in an effective manner. This Epoch-Shift—Impact—Response—Outcome characterization is summarized in a snapshot in Figure 4-4 below.

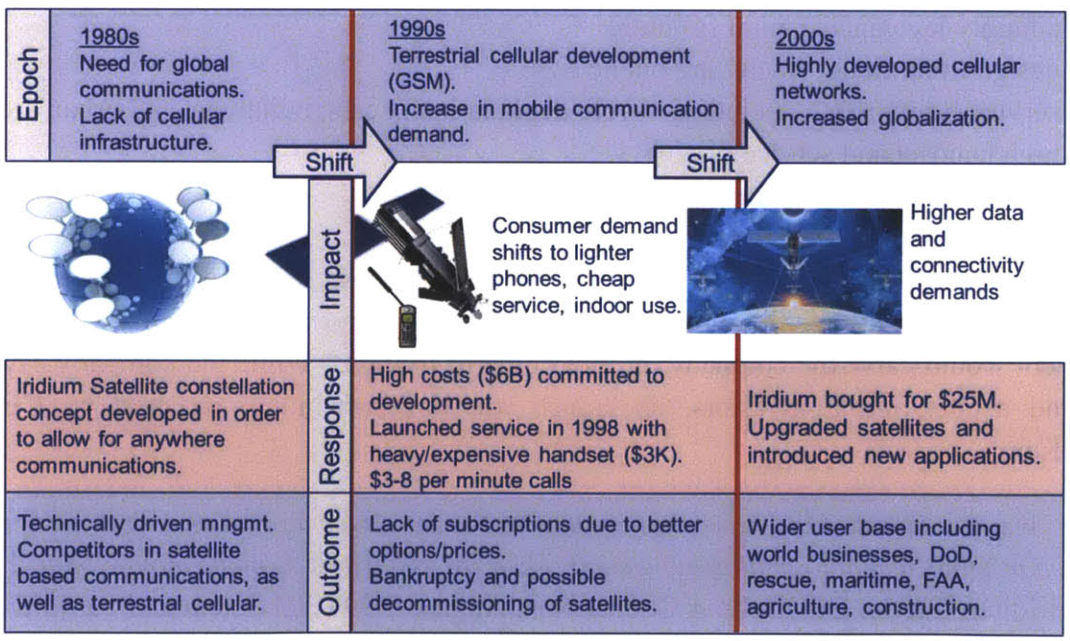


Figure 4-4. Iridium Epoch-Shift--Impact--Response--Outcome Snapshot

The aftermath of Iridium however did not involve the decommissioning of the constellation. The company underwent another epoch-shift and was able to respond in a way that allowed it to continue operations. The context of the system had shifted in the 2000's to highly developed cellular networks and an even larger amount of globalization and world communications in not just voice, but data as well. Following the events of September 11, 2001, increased emphasis on tracking trans-oceanic airplanes as well as in-flight data services allowed for new markets in which Iridium could compete. This impacted the system by increasing the data and connectivity demands in the market in areas terrestrial cellular systems could not reach. Iridium responded to these shifts first by selling and restructuring for $25 million, finding new customers, including the FAA and DoD, and upgrading satellites to introduce new functions. New functions include airplane guidance over oceans and poles, airplane black-box data collection and storage, remote location services (e.g. construction, oil rigs, foresting), military communications, and emergency response efforts (Ercetin et al. 2004). This resulted in a positive outcome for Iridium—a successful system operating in the green with increased commercial subscriptions (450,000 subscribers as of March 2011) and other contracts like the DoD. The company is preparing for its next response to the currently shifting context with a new constellation of IridiumNEXT satellites in 2015, with increased functionality, higher data speeds and new services in Enterprise data and voice, asset tracking, and other machine-to-machine applications. Time will tell if this response is too delayed or appropriate for the current context.

4.2.2 Globalstar

The story of Globalstar is similar to that of Iridium (de Weck et al. 2004). Funded primarily by Qualcomm Inc. and Loral Space & Communications, Globalstar was designed to provide global space-based communications including voice, data, fax, paging, and positioning (Puttalsri et al. 2006).

Globalstar implemented a 48 LEO satellite constellation (4 spare) at a 1414 km altitude in 8 polar planes with 6 satellites in each. Similarly to Iridium, Globalstar implements terrestrial gateways to link satellites to telephone networks. However, in order to simplify in-orbit satellites, Globalstar does not use on-board inter-satellite switching links (ISL), requiring more earth gateways to interconnect with telephone networks. Unlike Iridium, processing occurs on the ground, in an effort to make a more affordable system. Satellites act as simple signal repeaters, known as a "bent-pipe" architecture (Puttalsri et al. 2006).

Globalstar required $3.8 billion to build and maintain by 2000 when it became commercially operational. Service commenced with a $1,000 handset cost and $1-$3 per minute calls. Globalstar attempted to avoid Iridium's fate by appealing to broader target markets with a more aggressive marketing campaign while lowering costs and usage fees. However, like Iridium, Globalstar projected more potential users than what was realized. Globalstar filed for bankruptcy for $3.3 billion in 2002. Like Iridium, Globalstar was able to restructure and capitalize on niche markets to allow the system to survive, albeit with less users than expected. Both Iridium and Globalstar suffered an initial failure, ending in bankruptcy, however both systems also lived on in the end, with bankruptcy allowing the systems to be salvaged and continue to deliver value.

In this case, the epoch shifts were similar to those in the Iridium case. The responses were different however. Globalstar attempted to cut costs by altering the architecture, requiring more ground gateways, with fewer and less complex satellites in the constellation. Despite these efforts however, the company was not able to gain enough subscribers to cover the still high costs of system development. In the end, Globalstar suffered the same outcomes as Iridium (de Weck et al. 2004), an initial business failure, followed by a bankruptcy-permitted rebirth and restructuring. Below, in Figure 4-5, the Globalstar Epoch-Shift—Impact—Response—Outcome is summarized in a snapshot figure.

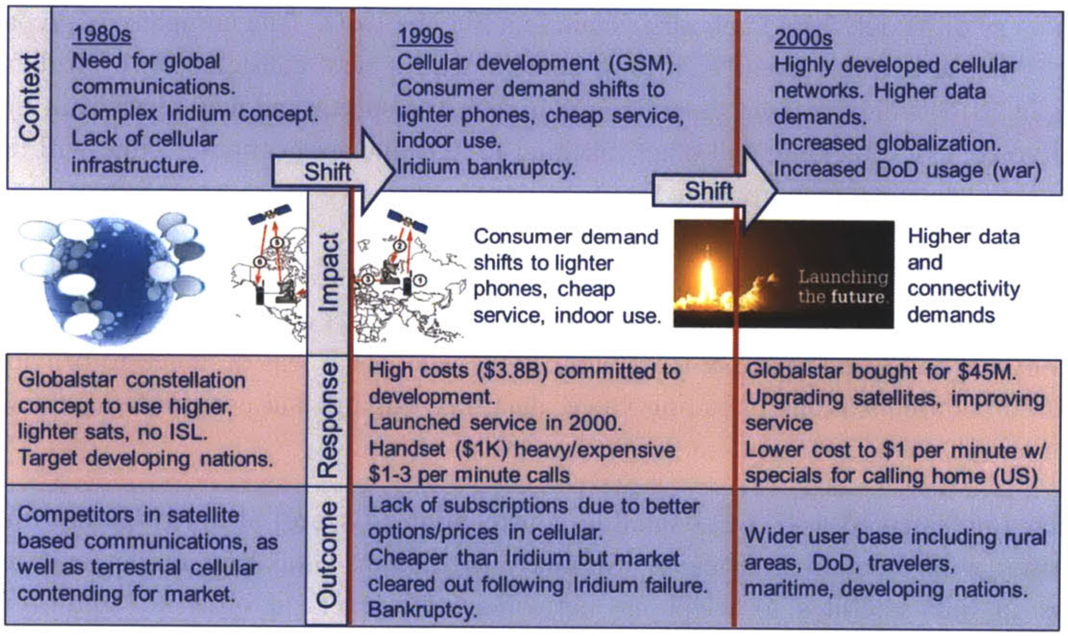


Figure 4-5. Globalstar Epoch-Shift—Impact—Response—Outcome Snapshot

4.2.3 Teledesic

One of the most ambitious satellite communication systems ever planned was Teledesic in the early 1990s, a $9 billion (in 1995 dollars) space-based internet provider (Wu 2010). It was proposed as the "Internet in the Sky," and aimed to provide the ability to send and receive information anywhere on the face of the planet. Numerous telecommunication companies displayed interest in such a system, including Craig McCraw of AT&T, Bill Gates of Microsoft, and Prince Alwaleed Bin Talal Bin Abdulaziz of Saudi Arabia. The company was able to raise $1 billion before the Iridium failure showed signs of declining market for LEO communication satellites (Rittenberg et al. 2009).

Teledesic originally planned for 840 LEO satellites at a 700 km altitude with 21 orbital planes containing 40 satellites per plane. It planned for complex inter-satellite switching links (ISL), as well as terrestrial gateway links, mobile links, and terminal links (Wu 2010).

In 1997, Teledesic scaled its large constellation down to 288 satellites at a higher 1400 km altitude with 12 orbital planes containing 24 satellites in each plane (Wu 2010). Later, Teledesic further scaled down to 30 satellites as market demand continued to decrease.

While Teledesic did manage to build and launch one test satellite in 1998, the company ended up releasing its frequencies and ceased work in 2003. While Teledesic did lose money, and ultimately did not become a viable system, unable to be robust to shifts in context, it was wisely shut-down, minimizing excessive downsides costs. This system had the benefit to see other systems fail, like Iridium and Globalstar, and used that information to respond to the shift in market needs. De Neufville and Scholtes (2011) might argue that the Teledesic response displays flexibility in that the system was able to minimize possible downsides as market uncertainty unfolded. Since the system would require a full constellation to provide internet for the first user, the system could not capitalize on any staged-deployment strategies as discussed in de Weck et al. (2004). Below, in Figure 4-6, the Teledesic Epoch-Shift—Impact—Response—Outcome is summarized in a snapshot figure.

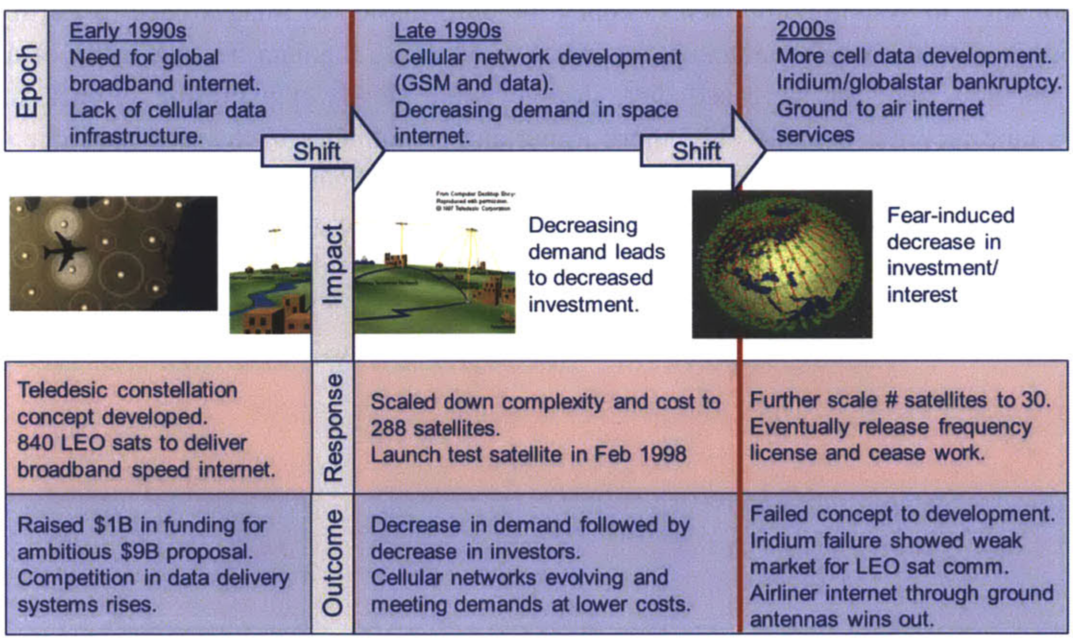


Figure 4-6. Teledesic Epoch-Shift—Impact—Response—Outcome Snapshot

4.2.4 Galileo

Galileo was a National Aeronautics and Space Administration (NASA) satellite built to explore Jupiter and its moons (Nilchiani 2005; Saleh et al. 2003). The satellite initially was intended for launch aboard a space shuttle in 1985, but due to launch delays and the 1986 Challenger disaster, the mission did not launch until 1989 aboard space shuttle Atlantis. This new launch meant that the satellite spent a lot of time waiting, and would have to wait even longer to reach Jupiter by taking a slower, less direct route to its final destination.

The Galileo satellite was a $1.4 billion space probe that relied heavily upon its ability to deploy its high gain antenna, its primary payload that would be used to relay data and images back to Earth. In 1991 however, during the 6 year transit to Jupiter, the antenna failed to deploy and was stuck in a closed and dysfunctional position, severely limiting capabilities of data communications. The reasons for this failure are not definitively known , but it is speculated that the long time spent on the shelf waiting for launch resulted in dried lubricants. It was feared that if the satellite could not deploy this antenna, the mission would be a failure. Efforts to open the antenna over the rest of the travel time to Jupiter failed, so extensive flight and ground software modifications were made in order to use a lower-powered antenna on the satellite as a substitution. Additionally, modifications were made to NASA's Deep Space Network to make up for the performance loss of Galileo's transmission capabilities. In the end, the satellite was able to achieve at least 70% of its original science mission objectives and some unplanned ones as well (Nilchiani 2005).

Later in its mission, after much time spent in a harsh radiation environment, the satellite suffered other anomalies in system components. Since the transmission of images back to Earth took so much longer with the weaker antenna, recording and storing the data became more of an issue. Due to the new need of increased data storage while slowly transmitting it back to Earth, Galileo's tape recorder became even more important. Anomalies in the tape recorder required NASA engineers to rework how the satellite captured data and what portions of the tape to safely use. Other issues due to radiation caused more anomalies in the spacecraft that required software and operational modifications. In the end, the Galileo spacecraft was able to remain robust to these context shifts and remained functional throughout its (extended) mission lifetime. The snapshot of these Epoch-Shift—Impact—Response—Outcomes can be seen below, in Figure 4-7.

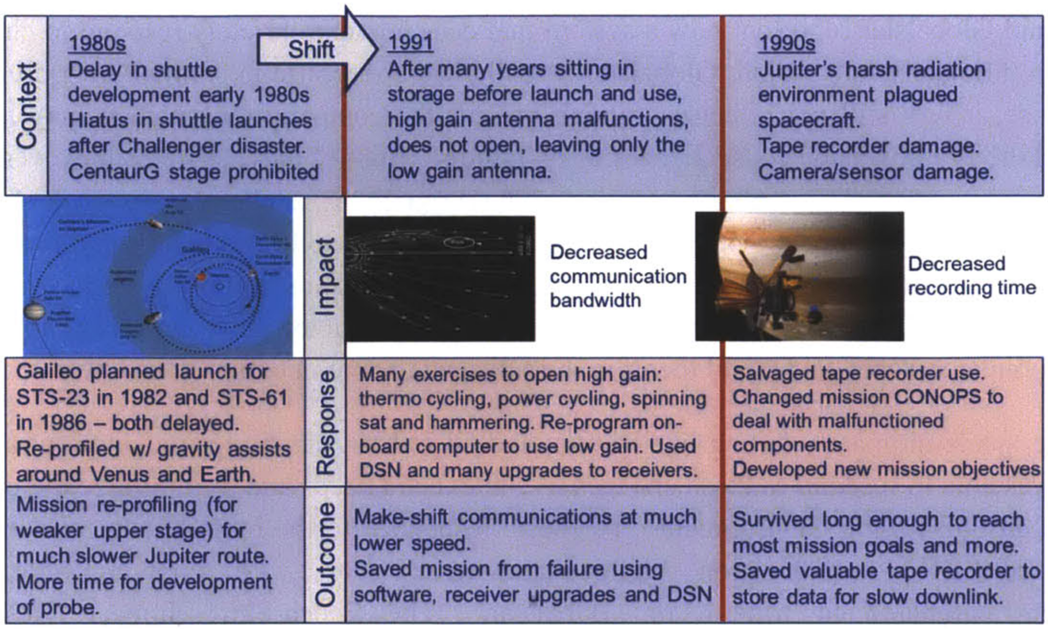


Figure 4-7. Galileo Epoch-Shift—Impact—Response—Outcome Snapshot

The Epoch-Shift—Impact—Response—Outcome for this system is one of repeated success. First, the system went through an epoch shift when it was forced to wait on a shelf for years until a new launch could be acquired. This may have resulted in degraded lubricants which ultimately ended up with the high gain antenna malfunctioning. This shift impacted the mission of the spacecraft, limiting the data that Galileo could transmit back to Earth and possibly rendering the satellite useless. The Galileo team responded to this shift by reworking the software and operations of flight and ground segments, as well as upgrading the Deep Space Network. The outcome was a successful primary mission for the satellite. Additionally, the system went through another epoch shift as the radiation environment plagued system components, all requiring custom modifications to software and operations in order to keep Galileo functional during its extended mission.

4.3 Insights from the Case Studies

These cases represent the kind of shifts in contexts that systems must be prepared to face throughout their lifecycle, including development as well as operations. While space systems have unique environments that require special strategies, all systems will require forethought in design to handle these perturbations successfully in their lifespans. Space systems demand this foresight due severe limitations in the ability to change the system in significant ways once the system is deployed. The construct of Epoch-Shift—Impact—Response—Outcome is useful in examining these cases to clarify how miscalculations in impact or inappropriate responses may result in failed or successful systems.

Iridium and Globalstar represent how a system that does not appropriately respond to changes in needs can ultimately fail. While Globalstar actually had a response in contrast to Iridium's lack of response, it was still not effective enough to save the company (although it may have had mitigated the full downside consequences of its failure). Both of these systems however live on, operating even today. The continued service of these satellite constellations however should not be considered a success of the original system, but rather the ability of a new company to take advantage of existing capital. These systems had large amounts of investment (both in technology and dollars), and through the mechanism of bankruptcy protection were able to shed their accumulated debts, and could therefore be salvaged to continue to provide value.

Teledesic, while ultimately failing like Iridium and Globalstar, can be seen as more of a success since it was able to respond to these changes in context and needs and to minimize the downside losses of an unfavorable environment. Teledesic responded to the burst bubble in demand for space-based LEO communication satellites and ceased work with relatively little upfront investment compared to Iridium or Globalstar. The Epoch-Shift—Impact—Response—Outcome construct therefore classifies Teledesic as more of a success.

Interestingly, when considering systems that were actually built and continue delivering value to this day, Globalstar and Iridium might be viewed as a success. In a perverse way, these systems succeeded in responding to their environment, using bankruptcy as a change option. While initial investors lost big money, from the system point of view, bankruptcy allowed the system to shed the downside losses that Teledesic avoided. This could possibly be an actual strategy for getting a system into operations, sacrificing investors' money to end up with an inexpensively acquired system. This strategy, however, comes with risks such as "spoiling the well," and leading to the decrease of investment in all space-based communications, which is what Teledesic suffered from after the bankruptcies of Iridium and Globalstar.

Finally, we get to Galileo. A system that was able to respond to every shift in design, context or needs and enable the satellite to continue to deliver value. As the satellite was forced into new, unpredicted contexts, the system was able to be changed to capitalize on these shifts to meet most initial science objectives and even to accomplish new objectives. The system's ability to reconfigure software after launch and change operations enabled value robustness for the program. Galileo was subjected to shifts in design (failure of high gain antenna deployment) as well as shifts in contexts (new launch and travel route) and needs (new mission objectives as spacecraft passed asteroids and comets) in an effective way that enabled value-delivery across the entire lifespan.

Table 4-1. Epoch-Shift—Impact—Response—Outcome Summaries

System	Shift	Impact	Response	Outcome
Iridium	Cellular development	Low subscription	None	Failure
Iridium	Increased communications	More data demand	Bankruptcy/ Target niche markets	Success
Globalstar	Cellular development	Low subscription	Cheaper architecture	Failure
Globalstar	Increased communications	More data demand	Bankruptcy/ Target niche markets	Success
Teledesic	Terrestrial data development	Decreased demand	Scaled down	Failure/ Success
Teledesic	Iridium/ Globalstar bankruptcies	Decreased investment	Cease work	Failure/ Success
Galileo	High gain antenna failure	Decreased bandwidth	Tech/ Ops re-work	Success
Galileo	Component damage	Decreased performance	Ops/ objective re-work	Success

Table 4-1 shows the summary of Epoch-Shift—Impact—Response—Outcomes for the four case studies. The summary shows how Iridium and Globalstar failed as initial systems, but ultimately succeeded in system deployment through the use of bankruptcy. Teledesic both succeeded and failed in both epoch shifts since the system was never developed, but large amounts of downside losses were avoided. Success or failure depends on the criteria being used (providing value as a system or providing profits or minimizing losses). Finally, Galileo shows an example of success in response to both epoch shifts.

When discussing the success of this program, often the lifecycle properties discussed earlier in this thesis are used. Nilchiani (2005) and Saleh et al. (2003) attribute much of Galileo's success to its "flexibility" in design. Similarly, de Neufville and Scholtes (2011) would label the Teledesic response and outcome as flexibility, scaling the architecture during system development with increased contextual knowledge. These cases show how the responses, or the change and resistance options employed by a system, can lead to very different outcomes in value sustainment. These non-traditional lifecycle properties, or ilities, may be useful concepts for designing systems that are capable of value-sustainment and are discussed in more detail in the next chapter.

5 Prescriptive Semantic Basis for Change-related Ilities

If non-traditional lifecycle properties, or ilities, are commonly referred to when discussing successful systems in history, or are characteristics that stakeholders demand for their systems, ilities themselves then must be well understood. This chapter outlines research aiming to better understand these desirable system properties and discusses current pitfalls and ambiguities surrounding them.

5.1 Motivation

The beginning phases of system development and conceptual design require careful consideration, as these decisions will have significant influence on system lifetime performance and are usually made with incomplete system knowledge. Decision makers may improve their capacity to discriminate between system concepts and design choices by measuring a system's "ilities" such as changeability, scalability, and survivability. These ilities may enable systems to respond to perturbations in the design space, context space, and needs space in order to ensure system functionality and adequate performance over time. A system may be designed to change in response, or remain statically robust/survivable to perturbations in order to avoid deficiencies or failures. This research attempts to analyze mechanisms that allow system changes to occur, and propose a framework for allowing system designers to map vague, yet desirable, ilities to prescriptive system design principles (Beesemyer et al. 2012).

While expressing desires for ilities seems uncomplicated, tracing these desires to verifiable system instantiations remains ambiguous at best. This chapter aims to outline the semantic challenge in assembling a coherent set of system properties and their definitions. This research introduces a prescriptive semantic basis for specifying ilities while avoiding the assertion of new definitions (Ross, Beesemyer, and Rhodes 2011)[7].

5.1.1 Ambiguity amongst Ilities

In addition to systems engineers, prominent political and technical leaders are increasingly using ilities not only as desirable system qualities, but also as *necessities* in a world with ever-increasing complexity, schedule and budget pressures, and need for finding sustainable solutions. When leaders use ilities like "flexibility" or "evolvability" as not only high level goals, but also system requirements, precise meaning becomes more important. Williams (2000) describes how companies are increasingly using ilities to market their performance amid ever-changing requirements, but warns that they need to be better understood, not just used as *buzzwords*. Ross,

[7] This chapter outlines research done within SEAri and is largely pulled from the Ross, Beesemyer, and Rhodes (2011) paper. If not referenced to some other author or literature, the words in this chapter come from or are inspired by Ross, Beesemyer and Rhodes (2011).

Beesemyer and Rhodes (2011) give a few example quotes pulled from the AIAA Daily Launch in recent months:

> *"Rep. John Mica called on the agency to "reform" and "become...a thinking, risk-based, **flexible** agency that analyzes risks, sets security standards and audits security performance."*
>
> *"Defense Secretary Panetta: "The US joint force will be smaller and it will be leaner. But it will be more **agile**, more **flexible**, ready to deploy quickly, innovative and technologically advanced."*
>
> ... *"the Defense Department and the Office of the Director of National Intelligence pledged to foster an industrial base that is **'robust**, competitive, **flexible**, healthy, and delivers **reliable** space capabilities on time and on budget.'"*
>
> *Quotes from AIAA Daily Launch, 20 Jul 2011 – 13 Feb 2012*

To help clarify meaning in these ilities, this research proposes a semantic approach to disambiguate possible ility structure and meaning. Semantics is the study of "meaning" and is a good starting point for clarifying this murky area of subjective ilities (Ross, Beesemyer and Rhodes 2011). We can derive meaning of a word from its "use" (in speech) and "prescription" (definitions). In a perfect world, meaning is clear and universal, with congruency between "use" and "prescription." In practice however, meaning changes over time, with the common meaning, or the way people "use" words, departing from the original prescriptive meaning. A relevant concept for this research is the *semantic field.* A semantic field is a "group of words with related meaning, for example, kinship terms or color terms," (Akmajian et al. 2001)

Ilities are often used in many different domains colloquially and as technical terms, gathering multiple meanings over time (Ross, Beesemyer, and Rhodes 2011). They can display polysemy and synonymy. Polysemy is "the property of [a term] having multiple meanings that are semantically related" (Akmajian et al. 2001, p. 585). For example, flexibility can mean "the ability to be changed by an external agent" (Ross et al. 2008) or "the ability to satisfy changing requirements after the system has been fielded," (Saleh et al. 2009). Saleh et al. (2009) also refers to the over 50 definitions of flexibility identified by Sethi and Sethi (1990) in a manufacturing context where they added that these definitions "are not always precise, and sometimes naïve" (p. 289). Synonymy on the other hand, is "the property of multiple terms having similar meaning," (Akmajian et al. 2001, p. 585). For example, flexibility and changeability may both be defined as "the ability of a system to change."

Ross, Beesemyer and Rhodes (2011) discuss how much of the literature on ilities tends to focus on a single ility at a time, rather than sets of ilities, leading to this polysemy and synonymy between ilities. For example, "flexibility" is covered in Saleh, Mark, and Jordan (2009), Nilchiani (2005), and de Neufville and Scholtes (2011). There has been some research in sets of ilities, however, in research from Fricke and Schulz (2005), de Weck et al. (2011) and Ross, Rhodes, and Hastings (2008).

5.2 Related Ility Research Overview

This section outlines some of the research accomplished in sets of ilities as opposed to any research focusing on one ility in particular (Ross, Beesemyer, and Rhodes 2011). In Cotton et al. (2009), more than 120 ilities were considered and filtered in an attempt to answer three questions: 1) "what are the overall objectives?, 2) What values are essential to ensuring effective [system] protection?, and 3) What values are essential to architectures?" Using iterative introspection with experience and literature, this study recognized that "no standard list of applicable "ilities" exist…" and "almost any attribute may be created by adding '-ility' to the end of the word…" (Cotton et al. 2009). Figure 5-1 shows the resulting sets of hierarchies in "architecture quality" and "system effectiveness."

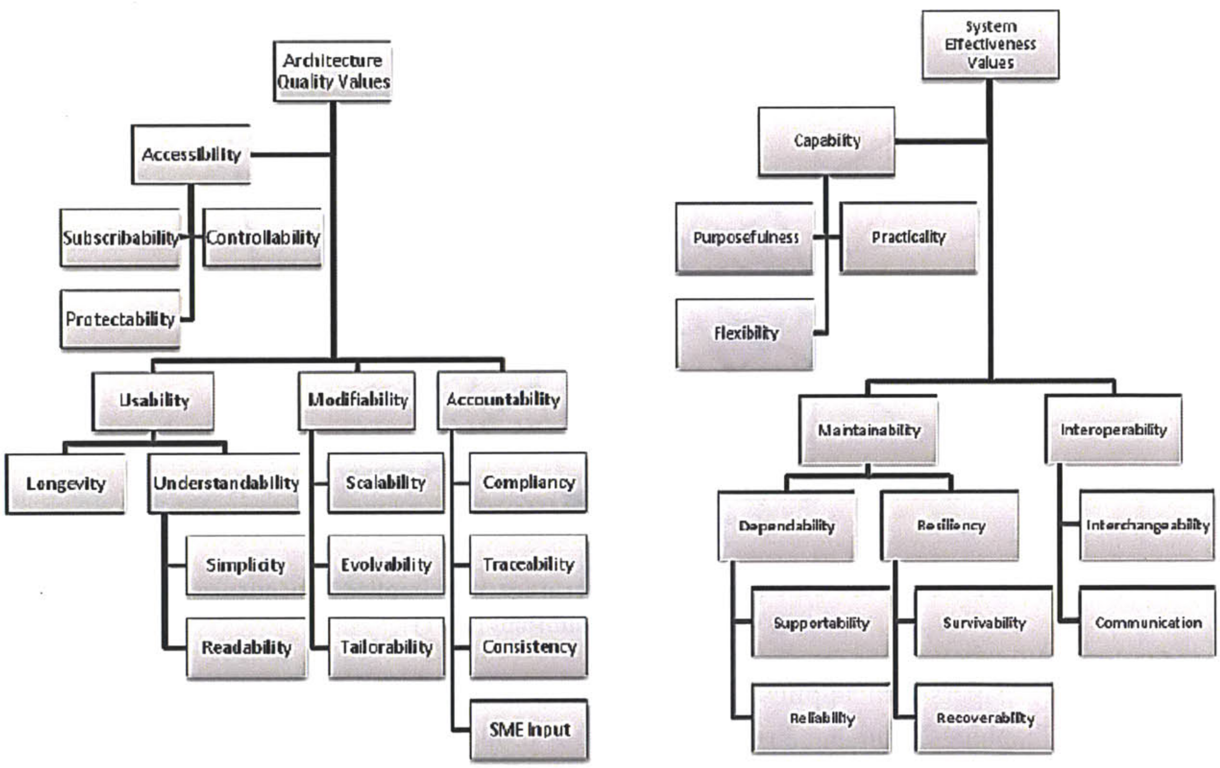


Figure 5-1. Ilities Hierarchies for "Architecture Quality" (left) and "System Effectiveness" (right) (Cotton et al. 2009)

A descriptive approach to understanding sets of ilities is described in de Weck, Magee and Roos (2011). In this research, de Weck et al. characterize ilities using their citation frequency in literature. In this way, ilities are compared to one another from their frequencies in literature and Google search "hits" over time. This shows a general increase in ility citations over time, especially recent rises in interoperability, sustainability, modularity, and testability. While this shows the increasing relevancy of ilities in literature, it does not get at meanings or relationships (Ross, Beesemyer, and Rhodes 2011).

An attempt to get more information in ility relationships, de Weck, Magee, and Roos (2011) examine the co-occurrence of the ilities, implying dependence amongst the terms. Figure 5-2 shows this co-occurrence by varying thickness of links between ilities. While this descriptive approach is very interesting in identifying common correlations between ilities, it falls short of describing any real relationships. These ilities are somehow related in the literature, but the specific nature of the relationships is unknown. The pairs of ilities could be complimentary, requisite, competing, or tradeoff in some other way. These results are a good motivator framing the links between these ilities, but are sensitive to the set of ilities examined or any polysemy/synonymy effects (Ross, Beesemyer and Rhodes 2011).

Figure 5-2. Ility Co-Occurrence in Literature, with Implied Dependence (de Weck, Magee, and Roos (2011)

One of the key papers that applies to this research is Fricke and Schulz (2005), previously discussed in Chapters 2 and 3. Fricke and Schulz (2005) use the idea of changeability as a higher order ility that includes adaptability, robustness, flexibility, and agility (Figure 5-3) as change-related ilities. This research distinguishes other ilities as "architecture principles" for achieving change-related ilities: simplicity, independence, modularity, integrability, autonomy, scalability, non-hierarchy, decentralization, and redundancy. The architecture-related ilities should be assessable with structural metrics (observable without having to operate the system). Separating ilities into architectural and change-related ilities shows up again later in this thesis during the development of semantic fields for ilities.

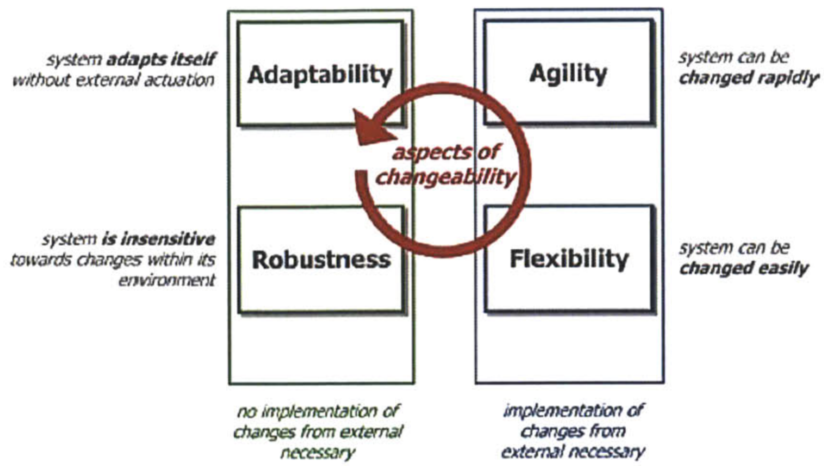


Figure 5-3. High-Order Changeability Including Four Ilities (Fricke and Schulz 2005)

Ross (2006) expounds on using changeability as a high-level ility. In this research, changeability is an umbrella ility, including five ilities along two relationship dimensions: flexibility and adaptability (change agent), and scalability, modifiability, and robustness (change effect) (Figure 5-4). Here, the concepts of change agent, change effect, and change mechanism are introduced in an effort to find the dimensions across which ilities differentiate (as in change agent). The differences between these ilities, per Ross, Rhodes and Hastings (2008), are described in more detail in Chapter 3 of this thesis.

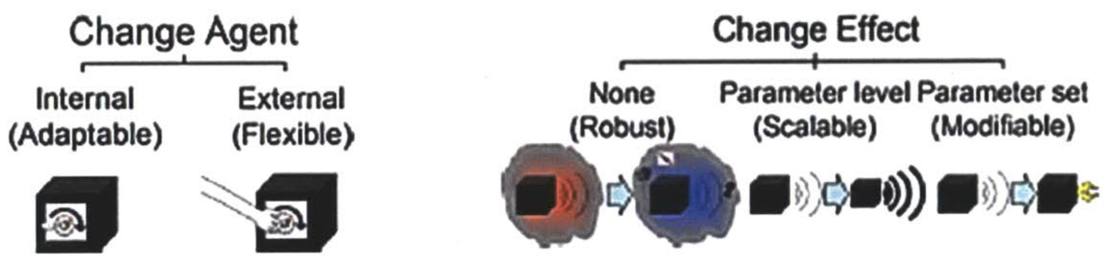


Figure 5-4. High-Level Changeability Including Five Ilities (Ross, Rhodes, and Hastings 2008)

Another important addition from the research of Ross, Rhodes and Hastings (2008) was the introduction of a template for specifying a verifiable changeability requirement. Figure 5-5 shows how a statement may be drafted as a verifiable requirement using dimensions of agent, effect, parameter, and resources. This motivates a similar style statement to be described with the semantic basis proposed later in this chapter.

The system shall be	______________	______________	in	______________	for less than	________.
	(change agent type) flexibly or adaptably	(change effects) scalable, modifiable		(system parameter) with range		(resources)

Figure 5-5. Template for Verifiable Changeability Requirement (Ross, Rhodes, and Hastings 2008)

Recently, more work has been accomplished in describing a means-end relationship amongst ilities (Ross, Beesemyer, and Rhodes 2011). Having an ility hierarchy could prove useful in designing systems for the specific desirable qualities that may be required. Constructing such a hierarchy, however, is a difficult task, as shown before in Cotton et al. (2009). An exploratory study in this endeavor is outlined in de Weck, Ross, and Rhodes (2012). In this study, four groups of graduate students from MIT independently derived ility hierarchies. In addition to means-ends relationships between the ilities, the groups independently proposed "levels" describing the "depth" of sets of ilities across their proposed hierarchies. Figure 5-6 shows the aggregated, independently derived hierarchies from the four groups. Solid lines represent three or four groups in agreement of the means-end relationship, and dashed lines represent two groups in agreement. The vertical placement of the ilities represents the "level" associated with that ility. The lack of consensus of the hierarchy and the emergent "level" aspect of the hierarchy suggests that there is more to the ilities relationships than just a simple means-end relationship.

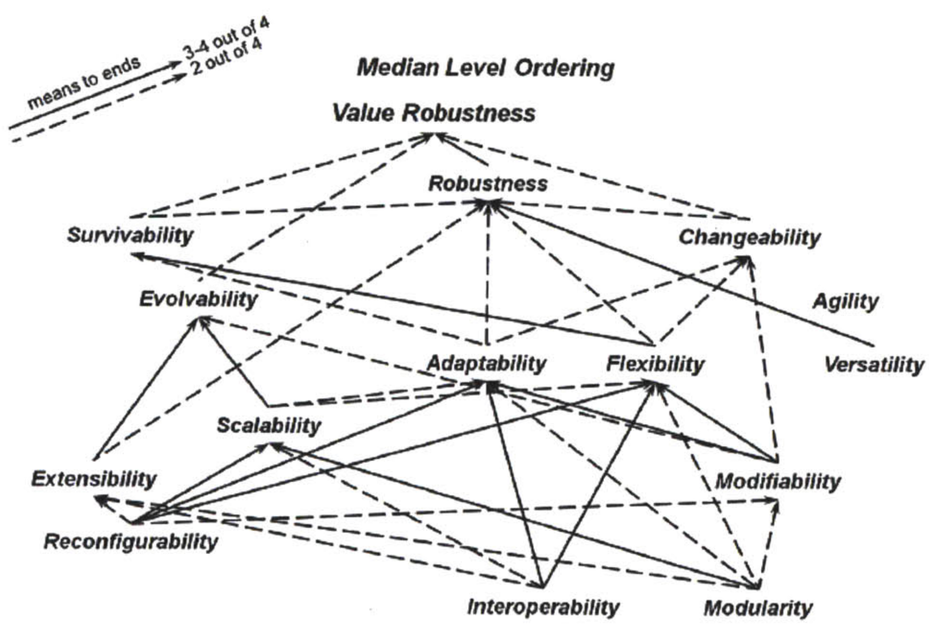


Figure 5-6. Aggregate Independently Derived Means-End Ility Hierarchies (de Weck, Ross, and Rhodes 2012)

Feedback from the four groups indicated the lower level ilities like interoperability and modularity exhibited some different "flavor" in contrast to the high-level ilities like value robustness (Ross, Beesemyer, and Rhodes 2011). The lower level ilities seemed to apply more to the architecture quality than to overall system performance. This result parallels Fricke and Schulz's separation of ilities into architectural and change-related ilities as well as the Cotton et al. separation of ilities into "architecture qualities" and "system effectiveness."

5.3 Formulation of Semantic Basis

Using the previous research as a motivation and starting point, more research was conducted in synthesizing different ways to look at ilities as sets. This thesis introduces the initial approach developed in this research for creating a prescriptive semantic basis for representing ilities within a particular semantic field. At this time, the semantic basis, made up of ten categories, is believed to span the *change-related* ility semantic field and excludes the *architecture-related* semantic field that includes lower-level ilities (de Weck, Ross, and Rhodes 2012) and "architecture principles" (Fricke and Schulz 2005) described above. Table 5-1 shows examples of ilities that fit into these change-related and architecture-related semantic fields (the list is by no means complete). As a result of this research, the change-related ilities are currently better defined than their architecture-related cousins[8].

Table 5-1. Change-Related and Architecture-Related Ility Examples

Change-Related Ilities	Architecture-Related Ilities
Adaptability	Accessability
Agility	Controllability
Changeability	Decentralization
Evolvability	Independence
Extensibility	Interoperability
Flexibility	Integrability
Modifiability	Modularity
Reconfigurability	Protectability
Scalability	Readability
Survivability	Redundancy
Versatility	Simplicity

[8] As the focus of this research, the change-related ilities are better defined, and can be assessed in terms of system changes or resistance to changes. . The architecture-related ilities should be observable in terms of the system form or operations. The architecture-related ilities become more confusing when one thinks about means-ends relationships. Ilities such as scalability or interoperability can be confusing because they refer to aspects that are observable during operations of a system, but may be enabled by form elements in the architecture. For example, modularity is a system property that is form-related. An area of future-research should include better defining the nature of architecture-related ilities and their relation to change-related ilities.

The methods discussed before have various "dimensions" in which ilities are differentiated. Ross, Rhodes, and Hastings (2008) described *change agent*, *change effect*, and *change mechanism* as three dimensions for differentiation. Implicit in that research was another dimension of *specification of the parameter* (more descriptive than just "parameter type", but also the specified parameter if known). Chapter 3 of this thesis describes various perturbations and how looking at different system or outcome parameters can map to different ilities.

Table 5-2. Ten Dimensions of Analysis in the Semantic Basis

Dimension	Name	Source
1	Perturbation	Chapter 3
2	Context	Chapter 3
3	Phase[1]	Ross and Rhodes 2011
4	Agent[2]	Ross, Rhodes, Hastings 2008
5	Effect[3]	Ross, Rhodes, Hastings 2008
6	Parameter[3]	Ross, Rhodes, Hastings 2008
7	Destination[1]	Ross and Rhodes 2011
8	Aspect	Chapter 3
9	Abstraction	Chapter 3
10	Value	Ross and Rhodes 2011

[1] Derived from *Change Mechanism*
[2] Derived from *Change Agent*
[3] Derived from *Change Effect*

Starting with these dimensions of change agent, change affect, and change mechanism (Ross, Rhodes, and Hastings 2008), more dimensions for differentiating ilities were added. The further decomposed change mechanism had change information that could also be used as dimensions of analysis, such as lifecycle phase, costs, and potential end states (Ross and Rhodes 2011). Putting these dimensions together with the others discussed in Chapter 3 of this thesis culminated in ten categories (shown in Table 5-2), which together form the semantic basis for specifying change-related ilities. Formal statements derived from this basis allow a system change to be verified in displaying certain characteristics that trace to desirable higher order system properties, or ilities. The distinct research across these 10 dimensions, or "categories," can be combined to create a more complete basis for describing the aspects of system changes that relate to system ilities.

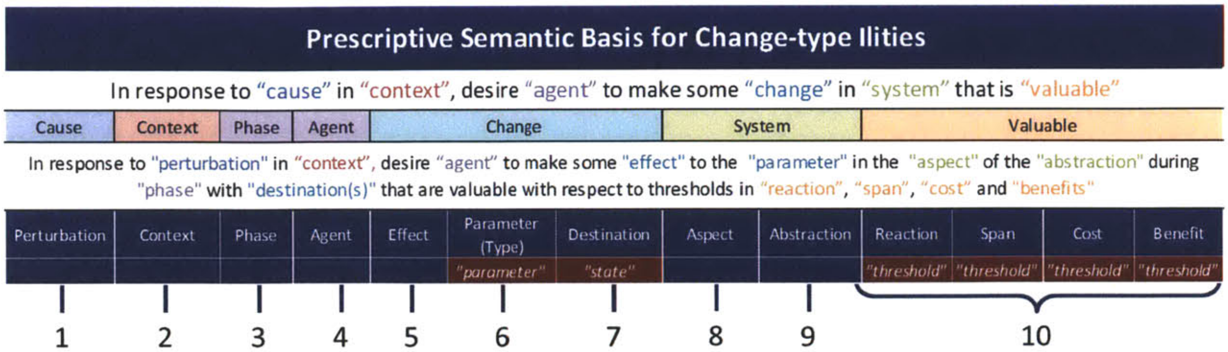


Figure 5-7. Change-related Prescriptive Sematic Basis Categories (Ross, Beesemyer, and Rhodes 2011)

The ten categories used in the basis include perturbation, context, phase, agent, effect, parameter type, destination, aspect, abstraction, and value (Figure 5-7). These ten categories have unique sets of choices from which a user chooses, with some having extra threshold or reference values that can be specified as well. When applied to a specific parameter, these categories formulate a change-related ility statement that may be adjusted depending on the level of detail or the categories of concern. The simplest form of the statement could be:

Desire some "change" in "system."

e.g. Desire hospital power source to switch from power grid to gas generator.

This is a simple statement that represents only the change and system information, signified as light blue and green in Figure 5-7. A more specific statement could be made however:

In response to "perturbation" in "context", desire some "parameter change" in "system" that is "valuable."

e.g. In response to a power outage in a severe winter storm, desire power source to be switched from grid to generator in the hospital to maintain operation of life-critical equipment lighting.

This statement captures more information and compared to the first statement, will yield a higher level of differentiation or clarification amongst ilities. As more dimensions are expressed, more detail about each change can express higher variation. For example, the first statement may only relate to system *changeability*, where the second statement may relate to *survivability* (since perturbation is now defined) as well. The most complete statement, using all ten categories and the four sub categories in the value section gives the most complete change requirement:

In response to "perturbation" in "context", desire "agent" to make some "effect" to the "parameter" in the "aspect" of the "abstraction" during

"phase" with "destination(s)" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits."

e.g. In response to a power outage in a severe winter storm, desire power control box to automatically switch the power source from grid to generator in the operations of the county hospital during daily ops with destination of full generator use that is valuable with respect to reacting within 1 minute of perturbation, change spanning less than 2 minutes, without losing any life-critical operations or equipment in order to maintain hospital care.

This statement captures all necessary change information and can correlate to numerous ilities that are present. For example, the added change agent information in this statement relates to an adaptable (internally driven) change.

Each of the categories in the basis has some unique responses that encapsulate constructs discussed in Chapter 3 of this thesis and fully capture available system decompositions[9]. Figure 5-8 shows the available responses for each of the ten categories below. Notice that three of the categories have clarification boxes that may be specified to relate to a threshold or reference value. The "parameter" clarification is required since these statements only make sense when framed around a specific parameter. If no system parameter is detailed, then the statement is unverifiable. Details about this difficulty are discussed later in the chapter.

Prescriptive Semantic Basis for Change-type Ilities

In response to "cause" in "context", desire "agent" to make some "change" in "system" that is "valuable"

Cause	Context	Phase	Agent	Change			System		Valuable			

In response to "perturbation" in "context", desire "agent" to make some "effect" to the "parameter" in the "aspect" of the "abstraction" during "phase" with "destination(s)" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits"

Perturbation	Context	Phase	Agent	Effect	Parameter	Destination	Aspect	Abstraction	Reaction	Span	Cost	Benefit
					"parameter"	"state"			"threshold"	"threshold"	"threshold"	"threshold"
disturbance	circumstantial	pre-ops	internal	increase	level	one	form	architecture	sooner	shorter	less	more
shift	general	ops	external	decrease	set	few	function	design	later	longer	more	less
none	any	inter-LC	either	not-same	any	many	operations	system	always	same	same	same
any		any	none	same		any	any	any	any	any	any	any
			any	any								

Figure 5-8. Change-Related Prescriptive Sematic Basis Category Choices (Ross, Beesemyer, and Rhodes 2011)

---

[9] In the development of this semantic basis, some other dimensions were considered. Two items from Chapter 3 that are not currently represented in the semantic basis are perturbations in "space" vs. "instance" and differentiation between system and outcome parameter. These dimensions were scoped out of this thesis, but represent areas for future research.

These ten categories can apply to the "real" and "perceived" spaces discussed earlier as well. Figure 5-9 shows the ten categories, including the four sub-categories within "value," as they fit into the two over-arching spaces and change/perturbation descriptions. There are four categories that apply to the "real" space: (1) abstraction, (2) aspect, (3) phase, and (4) context. There is one category that applies to the "perceived" space, value, with its four sub-categories: (1) benefit, (2) cost, (3) reaction, and (4) span. The perturbation category (perturbation type) and change categories (parameter, agent, effect, and destination) can apply to either the real space or perceived space.

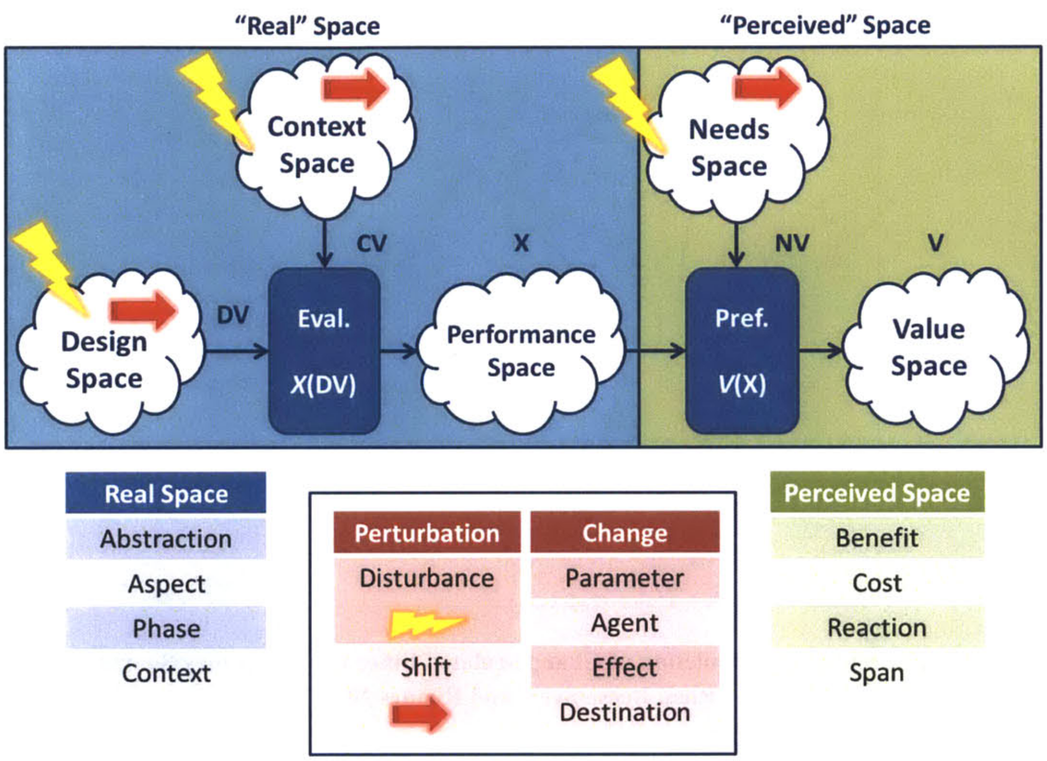


Figure 5-9. Perturbation Mapping in "Real" and "Perceived" Spaces with Ility Semantic Basis Categories

Within the ten categories, a change statement[10] can be assigned through unique categorical choices, or be left blank, depending on the level of specification when describing a type of change. However, if the semantic basis is used in a descriptive way, defining a change that has taken place, every category except value (since that category is based on the stakeholder preferences) should be specified. Depending on the combination of categorical values chosen, applicable ilities will emerge from the specified statement and parameter. In this way, the user

---

[10] The idea of an "imposition statement" was brought up during this research. If there is a change statement that applies to system changes, then there may be a resistance statement or imposed change statement that correlates. For the purposes of this research, change statements may apply to either voluntary changes or imposed changes. Resistance statement generation is an area for future research.

does not need to specify particular ilities *a priori*, thereby avoiding semantic ambiguity in the terms. If the semantic basis accurately and completely describes the underlying categories for change-related ilities, the user should be able to describe any change-related ility using only the basis. If not, that would suggest more dimensions are necessary.

The ten categories can be visualized using a radar plot, as seen in Figure 5-10, where different ilities trace out different paths around the axes. This plot only serves as a visual representation of the ility however. No information about the inter-relationships other than degree of similarity is given (each axis is either similar or dissimilar, not a continuous level of comparison).

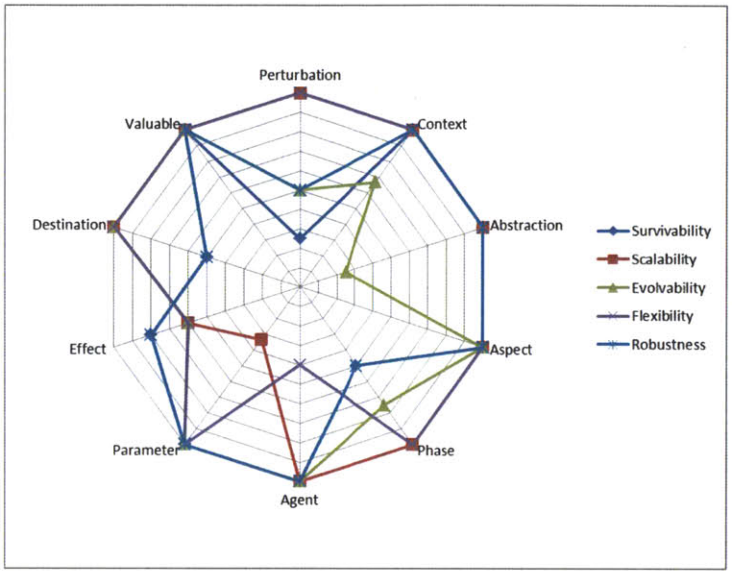


Figure 5-10. Radar Plot Depiction of Change-related Ilities in 10 Category Semantic Basis (Ross, Beesemyer, and Rhodes 2011)

Validation of the proposed basis is an iterative process of constantly refining the overall categories and choices within them over time. This chapter contains the iteration of the semantic basis, as of the publication date of this thesis. Implementing one's definitions of an ility may allow the basis to map certain different permutations of choices in the basis to respective ility terms. The basis must be able to consistently trace any particular change statements to applicable ilities. Usage of the basis results in a less ambiguous approach for specifying change-related ilities (Ross, Beesemyer, and Rhodes 2011).

5.4 Application

This section briefly describes how to apply the semantic basis to different system changes, for more in depth descriptions on the categories and their respective possible choices, refer to Appendix A. Application of the prescriptive semantic basis begins with specifying the change statement. This statement will become the basis for mapping a change to its representative ility or ilities according to its characterization. The change statement is assigned the categorical choices in each of the applicable categories (Figure 5-8). The first two components of the change statement deal with the cause and context of the change. This refers to the reason the change is occurring in response to what type of perturbation to the system and if the context is conditional upon certain epochs or general. The "system" is then defined by the abstraction being changed; is it the architecture changing, the design, or a specific system already in use? The aspect of the system (form, function or operations) as well as the lifecycle phase is described as well. For each change mechanism evaluated, there will be a defined change agent: either there is no change agent or whether the initiating force is internal or external to the defined system boundaries. No change agent refers to the passive types of system changes. The change itself is then defined by type and effect, whether the change is to a level or set of variables, and how much or many of that change is present. The number of potential end-states for each change mechanism is also captured. Since value-based analysis is useful to decision makers, change mechanisms may be even more deeply defined by individual preferences in the value they deliver. This value requires a baseline system or performance level for comparison, as value is usually characterized by aspects such as faster, shorter, cheaper, or other similar comparative ideas.

Based on the responses to the above categories of change mechanisms, various ilities may be observed as seen in a few examples in the bottom half of Figure 5-11. A change mechanism may be defined as flexible if an agent executes the change externally from the system, but that does not mean that change mechanism cannot also be labeled by reconfigurability, scalability, versatility, or other such ilities. Some ilities are dependent on more than just one specific change. Ilities like *versatility* require "strings" of change statements to capture the necessary information that relates to the particular ility. In the same way, other higher-level ilities can be described in more detail using strings of change statements. This gets back to the idea of changing or maintaining system parameters to change or maintain outcome parameters, described in Chapter 3. If there is a prerequisite placed on a system parameter for an ility based on another parameter, multiple statements will be needed. Versatility is the ability to satisfy diverse needs without changing form or operations. The form and operational prerequisites may be captured in two statements by keeping the form aspect constant and the operations aspect constant and then changing function in the next statement (Figure 5-11). Reconfigurability is another similar ility that requires the set of form components or operational components remain constant while either links or orders of those components are changed in a system during ops.

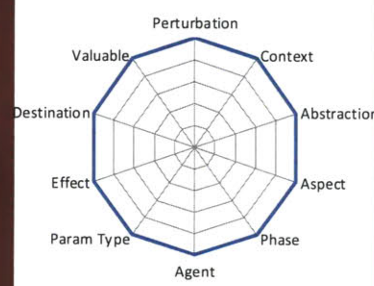
Prescriptive Semantic Basis for Change-type Ilities

In response to "cause" in "context", desire "agent" to make some "change" in "system" that is "valuable"

Cause	Context	Phase	Agent	Change	System	Valuable

In response to "perturbation" in "context", desire "agent" to make some "effect" to the "parameter" in the "aspect" of the "abstraction" during "phase" with "destination(s)" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits"

Perturbation	Context	Phase	Agent	Effect	Parameter (Type)	Destination	Aspect	Abstraction	Reaction	Span	Cost	Benefit
					"parameter"	"state"			"threshold"	"threshold"	"threshold"	"threshold"
disturbance	circumstantial	pre-ops	internal	increase	level	one	form	architecture	sooner	shorter	less	more
shift	general	ops	external	decrease	set	few	function	design	later	longer	more	less
none	any	inter-LC	either	not-same	any	many	operations	system	always	same	same	same
any		any	none	same		any	any	any	any	any	any	any
			any	any								



Perturbation	Context	Phase	Agent	Effect	Parameter (Type)	Destination	Aspect	Abstraction	Reaction	Span	Cost	Benefit	Ility Name
shift	any	any	any	same	"Value"	few	any	any	any	any	any	any	Value Robustness
disturbance	any	any	any	same	"Value"	few	any	any	any	any	any	any	Value Survivability
shift	any	ops	any	same	any	few	any	any	any	any	any	any	Robustness
shift	any	ops	none	same	level	few	form	system	any	any	any	any	Classical Passive Robustness
disturbance	any	ops	any	same	any	few	any	any	any	any	any	any	Survivability
any	any	any	either	not-same	any	any	any	any	any	any	any	any	Changeability
shift	general	inter-LC	any	not-same	any	any	any	architecture	any	any	any	any	Evolvability
any	any	any	internal	not-same	any	any	any	any	any	any	any	any	Adaptability
any	any	any	external	not-same	any	any	any	any	any	any	any	any	Flexibility
any	any	any	any	not-same	level	any	any	any	any	any	any	any	Scalability
any	any	any	any	not-same	set	any	any	any	any	any	any	any	Modifiability
any	any	ops	either	increase	set	any	any	any	any	any	any	any	Extensibility
any	any	any	any	not-same	any	any	any	any	any	shorter	any	any	Agility
any	any	any	any	not-same	any	any	any	any	sooner	any	any	any	Reactivity
any	any	ops	any	same	"Element set"	one	form	any	any	any	any	any	Form Reconfigurability
any	any	ops	any	not-same	"Link set"	any	form	any	any	any	any	any	
any	any	ops	any	same	"Element set"	one	operations	any	any	any	any	any	Operational Reconfigurability
any	any	ops	any	not-same	"Order set"	any	operations	any	any	any	any	any	
any	any	ops	any	same	any	one	form/ops	any	any	any	any	any	Functional Versatility
any	any	ops	any	not-same	set	few	function	any	any	any	any	any	
any	any	ops	any	same	any	one	form/ops	any	any	any	any	any	Operational Versatility
any	any	ops	any	not-same	set	few	operations	any	any	any	any	any	
any	any	ops	any	same	any	one	fnct/ops	any	any	any	any	any	Substitutability
any	any	ops	any	not-same	set	few	form	any	any	any	any	any	

Publish

Figure 5-11. Prescriptive Semantic Basis with Ility Definitions

5.5 Considerations for Use

The semantic basis has been continually updated and grown to include different areas of research involving systems that change. As the semantic basis has been developed, different insights, uses, and subtleties with the basis have been examined. Here is a brief discussion of some of these considerations.

5.5.1 Parameters

This semantic basis requires a clear description of the change mechanism under evaluation and the system parameter being specified. One can take the exact same change mechanism and redefine the system boundaries, or parameter that is altered, and come up with very different corresponding ilities. Even with very well defined ilities, altering a subjectively defined scope of analysis (e.g., system boundary) makes it possible to portray a system as more flexible or adaptable, or more modifiable or reconfigurable. For example, since flexibility or adaptability depend solely on whether the change agent is internal or external to the system, the system boundary is critical to differentiating between those ilities. This distinction within the basis arises whenever humans are in the loop as operators. Whether the operators are considered to be internal or external to the system establishes the classification of adaptable or flexible respectively. A pilot in a fighter jet makes many decisions, but there are also many decisions the jet makes for the pilot without any operator input. One may want to distinguish between automated and manual tasks within the aircraft and hold the pilot external to the system. Viewing the system from a broader, strategic perspective however may result in wanting to hold the individual actions of the aircraft as adaptable when contrasted with drones in an air space and hold the pilot internal the system. The important distinction to be made is that in order for a system to be adaptable, it must have the ability to perceive the need to change and choose to execute that change, which are the roles served by an "agent."

Similarly, there is sensitivity in the basis depending on how the parameter is described. Descriptive statements, depicting changes that have already occurred, tend to focus on implementation, or system parameters, whereas requirements statements (using the basis to create a verifiable requirement) tend to focus on outcome parameters. For example, when examining the changing wing sweep angle of the F-14 Tomcat fighter jet, the parameter for this change in the aircraft may be examined in different ways. The descriptive point of view could identify the changing parameter as the *wing sweep angle*, scaling from small to large. A requirement point of view however might look at the parameter of level of *lift* produced by the aircraft, scaling up and down. In the requirement point of view, there is no specification on how the lift increase is chosen, giving freedom to the design team as opposed to directing the design.

This shows how the parameter specification is necessary and important when drafting these ility statements. The parameter defines the focus of the sentence—where the change is actually taking place. The parameter can range from very specific ("diameter of screws") to more abstract or broad ("value"). An ility statement may string multiple statements together going from specific parameters to broad parameters, showing how higher-level ilities are achieved.

5.5.2 Levels of Abstraction

There are various levels or "abstractions" in which analysis of a system can be performed. Every system has a corresponding architecture, even if not explicitly defined, which is made up of the blueprint from which all designs of the system originate. The architecture may contain one or many different designs of the system. From the design level, a specific system may be constructed and implemented. Therefore, there are many instances of designs from a given architecture, and many instances of systems from a given design. As an example, consider the Apple iPhone. The iPhone has an overall architecture, with one being the iPhone 4 architecture. Within that architecture there are different instances of designs, for example there are 16 GB and 32 GB designs, and different cell-phone carrier designs, all considered variants of "iPhone 4." Within each of those designs there are multiple instances of systems, for example Bob's iPhone 4 32GB vs. Sally's iPhone 4 32GB.

These distinctions become relevant when analyzing the evolvability of a system. By definition, evolution of a system must take place between generations of a system at the architecture level (Beesemyer et al. 2011). Pliability is described as the ranges of parameters within a system architecture that yield viable system designs and can be viewed as the set of valid design instances within the architecture (Mekdeci et al. 2011). If a data storage device architecture were offered as giving 100 to 500 gigabytes of storage, a terabyte of storage would be considered outside the pliable range of that architecture. A designer could create a terabyte storage device, but it would require a change in architecture without which there may be no guarantee that the prior architecture would remain viable with a 1-terabyte drive. These breaks in architecture are where changes in generations occur and are exemplified by the iPhone 3GS vs. the iPhone 4, where a change in architecture enabled new design instances.

The semantic basis captures the difference in the level of analysis being executed in the abstraction of the change. Important to note is even if a specific system is scalable in design, it does not necessarily mean it is scalable in specific design instances. For example, designs of a rocket family may be scalable in deliverable mass to orbit, but a specific instance of the design being built may no longer be scalable. For example, the architecture of Atlas V rockets allows for scalable designs with 1 or 3 common booster cores (Alliance 2010). However, once a design is chosen (e.g. a 1 or 3 booster variant), it ceases to be scalable in the same way. A single common booster core rocket under construction cannot be scaled to a 3 common booster core rocket.

5.5.3 Ility Hierarchy

If there is one thing that strategic level thinkers would want, it might be a hierarchy of ilities. If it was very clear what ilities enabled other ilities, then system architects and designers might have an easier job at choosing the best designs. However, as shown in this effort earlier in this chapter, discovering an all-inclusive ility hierarchy is very difficult to accomplish due to the many dimensions across which ilities are related. Different semantic basis users may prioritize or care about different dimensions. These varying priorities may relate to different orders in which the basis dimensions are considered. In Chapter 3, where value sustainment is characterized with respect to perturbation type and change effect (change or no-change), a simplified hierarchy is created with value sustainability at the top, followed by value survivability and value robustness, followed by the subtypes, changeability, robustness, survivability and versatility. This hierarchy might change however if other categories of interest are examined earlier in the distinction, like internal changes vs. external changes. In this case, the branches may separate into flexible vs. adaptable changes at a higher level (Figure 5-12).

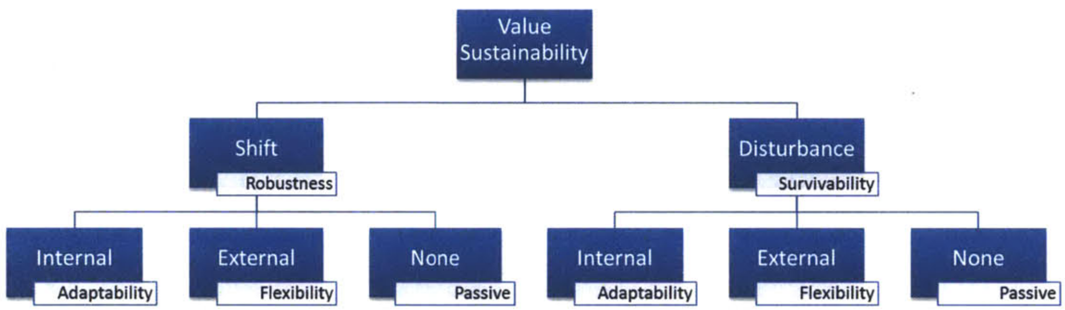


Figure 5-12. Hierarchy Example Based on "Agent"

If parameter type is a category examined at the higher priority level, then the branches may separate into scalable vs. modifiable changes (Figure 5-13).

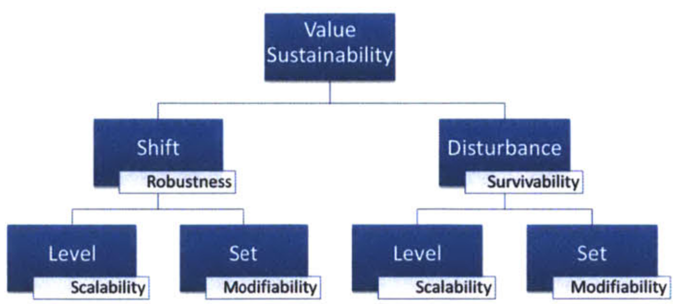


Figure 5-13. Hierarchy Example Based on "Parameter"

This may explain why different groups and different research suggest alternative hierarchies for ilities. Often these hierarchies may be similar, since the mental models for hierarchies can be very similar at high levels like value sustainment, but begin to diverge as more of the change information is clarified. It is hypothesized that this ten-dimensional space represents a "hyper-ility space" that can be "sliced" in different ways to represent different hierarchies. Depending on the order of the categories examined, or which slice of the ility space is taken, different ilities will appear to relate to one another in varying ways.

The design principles discussed in this thesis and other research areas (Richards 2009; Fricke and Schulz 2005; Ross 2006) are sometimes described as other ilities that would be thought of as "lower-level" or "architecture-type" ilities that can be used to enable the "higher level" ilities. This may align with our ility relationship hypothesis that the design principles for an ility may be the implementation of other ilities found in lower branches of a particular hierarchy. For example, design principles for survivability include mobility and evolution (Richards 2009) and this thesis has suggested design principles for evolvability like modularity, scalability, and reconfigurability.

Within the 10 main categories (13 columns), there are many different possible permutations of the basis. The number of "slices" depends on the order in which you examine the categories. Therefore, the number of possible slices equals the factorial of 10, or 3,628,800 different slice orderings, or hierarchies. If you take all of the possibilities from each category and multiply them together, that will be the number of possible permutations of the basis (4 x 3 x 4 x 5 x 5 x 3 x 4 x 4 x 4 x 4 x 4 x 4 x 4 = 58,982,400 permutations). That means that there are about 59 million different final "leaves" in the fully established hierarchy for every specified parameter[11]. This number is further multiplied when the addition of the reference and threshold values are considered or the stringing of statements. There can be two of the exact same hierarchies with different decision makers' threshold value levels, creating similar, yet different, hierarchies. In short, there are many possibilities for change statements. Figure 5-14 shows an example of a hierarchy following down one branch at each level (i.e. dimension) for illustration. Each branching point in the hierarchy represents a dimension of the basis, and correlates to the "levels" emergently found in de Weck, Ross, and Rhodes (2012) when graduate students attempted to create means-ends relationships between ilities. This hierarchy uses the order in which the sematic basis identifies dimensions, and excludes the "any" option or the detailed "value" categories. One can see how this hierarchy can grow quite large if drawn completely out, as this represents a branching that leads to just one leaf node.

[11] That being said, there are 52 unique branch labels, which are repeated throughout any given hierarchy, which correspond to the specific choices available across the 13 columns.

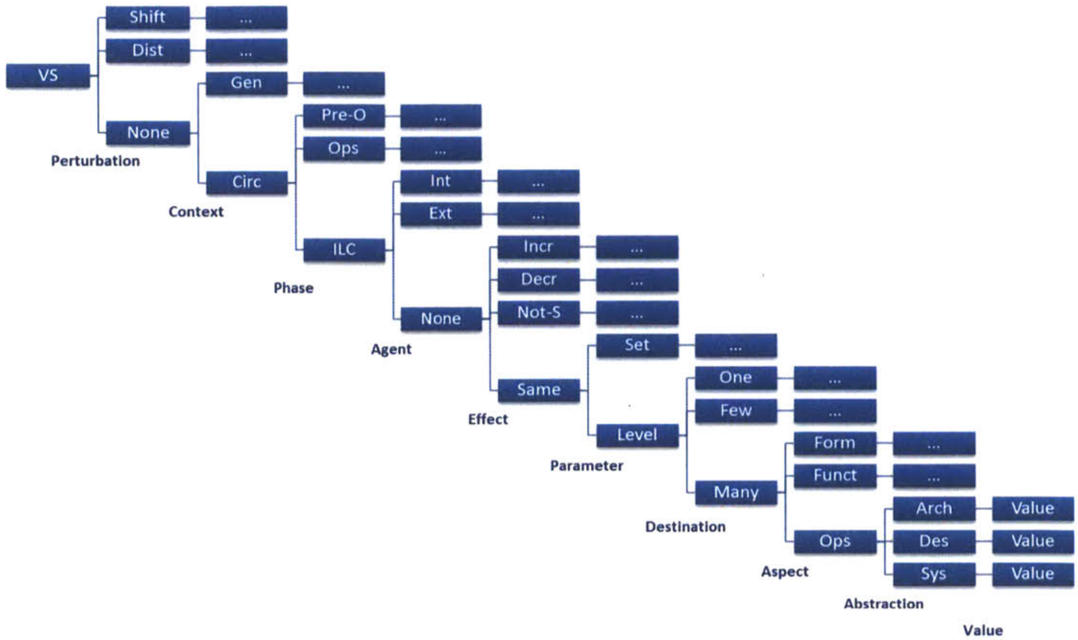


Figure 5-14. Example Hierarchy with 10 Dimensions (Abbreviated)

One does not have to specify down to the bottom leaf nodes in the fully established hierarchies, however, as "ilities" may exist in multiple branches and at multiple levels. Ilities that have more "any" responses to categories represent ilities that depend on fewer specification requirements. These ilities may correlate to the "higher-level" ilities that encompass the other, more specific ilities below them. For example, *value robustness* contains many "any" responses that somehow relate to system changes that maintain a parameter across a shift perturbation and may be considered a broader ility when compared to *functional versatility*. The ilities that contain more "any" responses may be broader, and can be identified by using the *any-count* metric, which ranges from 0-13 (13 being completely broad, and 0 being completely specified in each of the 10 dimensions, including the four value categories). Value robustness would have an *any-count* of 9, where functional versatility would have an *any-count* of 8. Higher *any-count* ilities could be considered broader of ilities.

5.5.4 Ility Discovery

One of the benefits the semantic basis for change-related ilities has afforded systems engineering research is the discovery of new types of ilities. With all the various permutations in the ility basis, there are millions of possible ilities. Many of these permutations may be trivial and not relate to any important ideas, but there may be types of changes that are not recognized as valuable properties until they can be "discovered" using the basis.

An example of such an ility is *substitutability*, which was discovered when examining the definition of versatility. The semantic basis-generated definition for substitutability is *the ability of a system to change the parameter set of form while maintaining original functions and* *operations*. This is similar to versatility in that versatility is changing function or operations while maintaining original form. Versatility is visualized in Figure 5-15 below.

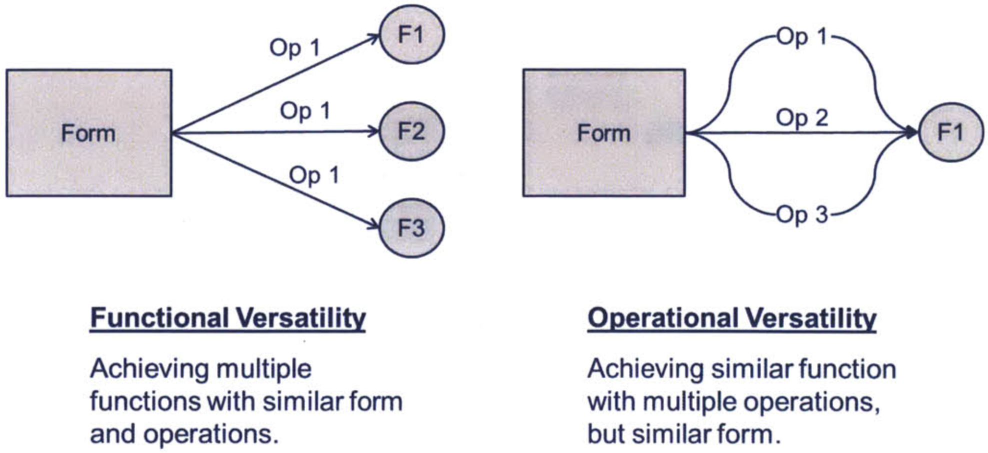


Figure 5-15. Functional and Operational Versatility

Versatility, in its original definition, only entailed meeting diverse needs without changing form. The semantic basis allowed for this definition to be clarified further, defining that either function or operations must change while the system maintains original form. In this way, there can be both operational and functional versatility (Figure 5-15). This clarification gave rise to the question of a third possibility of maintaining operations and functions, but changing form, giving rise to the concept of substitutability (Figure 5-16).

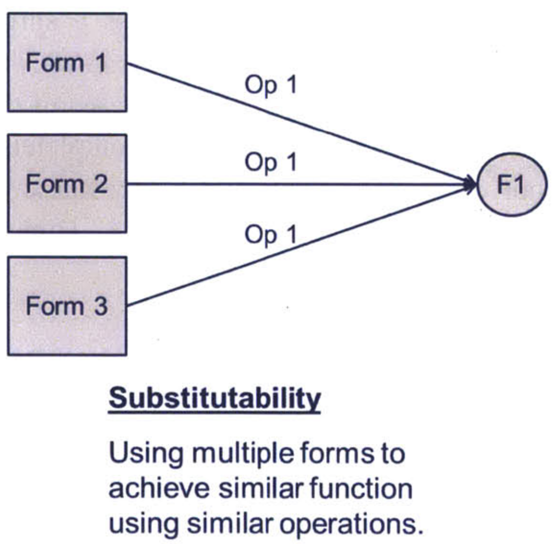


Figure 5-16. Substitutability Visualized

Figure 5-16 shows how a system might be able to use varying form in the same way to achieve the same function. A physical example of this could be the components in a computer. There are many types of hard drives or disk drives that can fit into the same laptop slot, which are used in exactly the same way, and for the exact same purpose, but have different forms. Substitutability then, can give the user the ability to substitute this component with other brands for any multitude of reasons.

Another clarification that emerged from developing the basis applies to *reconfigurability*. The original MIT SEAri definition for this ility was *the ability of a system to change its configuration (component arrangement and links)*. Using the semantic basis, the definition was updated to *the ability of an entity to change the link set of form/operations while maintaining the original element set of form/operations during ops*. This new definition clarifies some of the facets that were implied in the original definition, such as the lifecycle phase of "ops." Additionally, the basis highlighted the fact that there are actually two types of reconfigurability, operational and form. Generally, form is the obvious aspect of a system that is reconfigured, but software and operations can be reconfigured as well, changing the order of tasks to get different system performances.

Another example of clarification of a definition came when looking into the agility of a system. Agility in MIT SEAri terms means simply to change in a timely fashion. This is a subjective ility that instigated the need for threshold values when discussing any sort of value dimension of an ility. Often it may be the case that the unarticulated threshold level is the prior state of the system, but the semantic basis requests that this threshold be made explicit. There can also be confusion in what it means "to change in timely fashion." Change mechanisms have different reaction times (when the change is initiated) as well as different spans (how long the change takes). It is unclear whether the agile system is the one that reacts faster, or completes changes faster. One can imagine combinations of the two facets of time that result in overall change durations longer than the original system, yet could be classified as more agile or reactive due to better reaction time or span time. Therefore, reactivity was created to describe change reaction time, whereas agility describes change span time. Together they make up a responsiveness, as seen below in Figure 5-17.

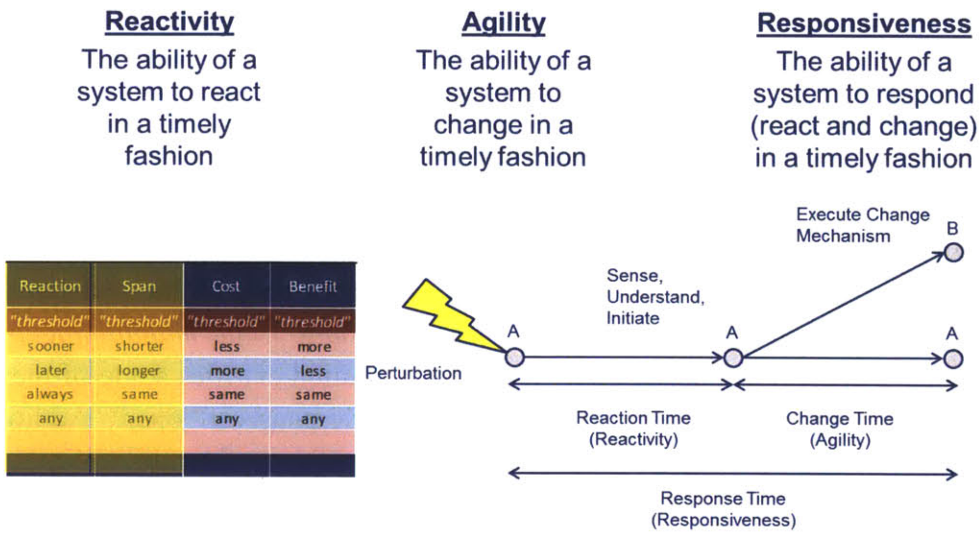


Figure 5-17. Reactivity, Agility, and Responsiveness Defined

5.5.5 Ility Definitions

The ility definitions that were implemented in this semantic basis, for demonstration purposes, were pulled from the definitions originally given in Table 2-1, the MIT SEAri ility definitions. As discussed earlier in the chapter, the semantic basis for identifying ilities was not developed to act as platform for declaring new ility definitions, but rather as a tool that system engineers may use as a common language in discussing desired system properties. However, in support of developing and implementing the semantic basis and providing examples for face validity, the MIT SEAri ility definitions were mapped using these ten dimensions.

This mapping required the ability to define any of the change-related ilities using only the 10 dimensions in the basis. If there was an ility that could not be completely represented in the basis, then that meant there was something missing from the current state of the basis. Iterating on coverage of the ilities using the basis brought the basis to where it stands in this thesis. Development of the basis also informed research in the different semantic fields, change-related, architecture-type, and other function-based ilities (drinkability) when certain ilities did not fit into the basis on the lacking of any change in the system.

With the MIT SEAri ilities implemented in the basis, as in Figure 5-11, the reverse exercise may be accomplished to see if definitions for ilities may be "reverse-engineered." That is, can we use the semantic basis for developing broad, yet clear and concise, definitions for the ilities? Using the basis definitions of the ilities, seen in Figure 5-11, the ility definitions were reverse engineered and can be seen in Table 5-3 below. The basis-generated ility definitions may be seen in blue, below the original definition. We can see that there are differences in many of the definitions. Often the basis-generated definition includes more than just a system, but the design and architecture entities as well.

Table 5-3. Reverse Engineered MIT SEAri Ility Definitions

Value Robustness	ability of a system to maintain value delivery in spite of changes in contexts or needs
	ability of an entity to maintain value delivery in spite of **shifts** in contexts, needs, or **design**
Robustness	ability of a system to maintain its level and set of specification parameters in the context of changing system external and internal forces
	ability of an entity to maintain a specified parameter during **ops** in spite of **shifts** in context, needs, or **design**
Changeability	ability of a system to alter its form, and consequently possibly its function, or operations, at an acceptable level of resource expenditure
	ability of an entity to alter form, function, or ops
Flexibility	ability of a system to be changed by a system-external change agent with intent
	ability of an entity to be changed by a system-external change **agent**
Adaptability	ability of a system to be changed by a system-internal change agent with intent
	ability of an entity to be change by a system-external change **agent**
Evolvability	ability of a architecture to be inherited and changed across generations (over time)
	ability of an architecture to be **changed** between generations in response to general shifts in context or needs
Survivability	ability of a system to minimize the impact of a finite duration disturbance on value delivery
	ability of an entity to maintain a **specified parameter** during **ops** in spite of disturbances in **context, needs, or design**
Versatility	ability of a system to satisfy diverse needs for the system without having to change form (measure of latent value)
	ability of an entity to change its **set of functions** or **operations** while maintaining original form during **ops**
Scalability	ability of a system to change the current level of a system specification parameter
	ability of an entity to change the level of a parameter
Modifiability	ability of a system to change the current set of system specification parameters
	ability of an entity to change the set of a parameter
Reconfigurability	ability of a system to change its configuration (component arrangement and links)
	ability of an entity to change the **link set of form/operations while maintaining the original element set of form/operations during ops**
Agility	ability of a system to change in a timely fashion
	ability of an entity to change in a shorter time **span** with respect to a **threshold value**
Extensibility	ability of a system to accommodate new features after design
	ability of an entity to increase **a parameter set during ops** by internal or external change agents

This comparative definition exercise is valuable in clarifying and integrating research performed in different contexts during different times by highlighting differences in definitions, like aspects that may have been implied or left out. The original definitions were largely developed in isolation, one ility at a time, like survivability in Richards (2009). The semantic basis-generated definitions take the experience and knowledge gained by creating a semantic basis that brings together many different dimensions that distinguish between ilities, and therefore can add clarity to definitions where ambiguity existed before. For example, the original definition of value robustness entailed maintaining value delivery in a system in spite of changes in contexts or needs. The new definition clarifies further, not changes in contexts and needs, but more specifically, shifts in contexts, needs, or designs. In addition, value robustness is not just a property available to systems, but to all entities—architectures, designs, and instances.

Similarly, survivability is an ility that in the past implied value as a parameter being maintained across a disturbance. In the new definition, value is eliminated, and a generic parameter is used, allowing for survivability to apply to more than just value, but any specified parameter, like form, or lift for example. The new definitions do not imply value as a parameter. The word "ability," however, relates to the relative ease in achieving the ility. A displayed ility is amplified then by improving efficiency/effectiveness/value of the ility. For example if two changes were considered flexible and lead to the exact same state change in a system, but one change was cheaper or more efficient, then that would be considered *more* flexible.

The semantic basis provides a foundation for describing many ilities using a common language, and is the start for being able to explicitly determine whether and how a system can display many ilities and the trade-offs that might exist amongst these ilities.

6 Historical System Change Database

In support of the research investigating connections between change mechanisms and ilities, a study was conducted to look at historical examples of system changes. A database was created to hold the data for different change mechanisms in various systems experiencing a wide variety of changes. This empirical database was originally created in an attempt to experiment with the semantic basis for identifying ilities displayed in system changes, but matured to capture sufficient data about actual system changes in a structured manner that could be used for aggregate data-mining as well. This chapter outlines the creation and development of the database and the categorical clustering method used in analyzing the data. The purpose of this research thrust was two-fold: (1) to serve as a preliminary means to validate the semantic basis discussed in Chapter 5, exposing weaknesses, holes, and ambiguities to help create a more complete construct, and (2) to shed light upon and further motivate research into how populations of systems may tend to display particular change strategies. Informed by Ross and Rhodes (2008b), the ilities definitions implemented in this database and their conditions were constantly refined and augmented for any generic change in a system.

6.1 Capturing Historical System Changes

6.1.1 Overview and Data Collection Method

During the development of the semantic basis, in order to determine ilities based on various change dimensions as discussed in Chapter 5, examples of systems that exhibit a wide range of changes were gathered in an effort to validate the completeness of the basis. The semantic basis needed to be able to effectively and completely represent many different types of system changes. If there were any examples of system changes that could not be completely characterized by the basis, then it served as an indication that more dimensions needed to be added to the basis. This activity was an opportunity to look into many different system changes and pull out ility labels based on populations of systems. If captured in a useful manner, these system changes could serve as an exploratory research effort into examining ilities co-occurring with system descriptors like lifecycle length, domain, or even production-type. A database was created to capture these changes along with system descriptors in order facilitate analysis for co-occurrence.

The database began as an exploratory endeavor, with system change examples collected through informal interviews with graduate students, researchers, and faculty at MIT. The system changes to be included in the database were chosen based on a guiding question of "what systems do you know that change states[12]?" Elicited system suggestions were not selected for any particular

[12] One aspect that is tricky when dealing with system state changes is determining the threshold of what constitutes a change. This database includes a wide variety of change levels. On one end of the spectrum, changing a parameter

predetermined ilities, but rather the idea of a system that changed some aspect of form, function, or operations. That is, no system was chosen because it represented an "adaptable" or "survivable" system, for example. The systems were collected on the premise that they displayed some sort of change as characterized by the change-related ilities discusses in Chapter 5. The elicited recommendations represented a diverse set of systems. In examining the set of system changes after they were chosen, it became apparent that the change examples spanned numerous domains and ranged from very expensive to very inexpensive, long-lasting to short lifecycle, and mass-produced to unique systems. With such a diverse sample set, it seemed reasonable that the dataset could be analyzed in an exploratory manner to uncover preliminary implications for the population of systems that change, and which have enough information publicly available to capture necessary change characteristics[13].

6.1.2 Development of the Database

Using FileMaker Pro®, the initial fields that the database needed to capture were defined by the semantic basis. Since this basis was being developed and refined in parallel to the database effort, the fields in the database evolved over time. The first task was to find out what information other than the specific change information could be helpful in the analysis. First were the justification fields that capture the system change example in terms of the ten dimensions of the basis, seen below in Table 6-1.

---

from on to off may constitute a change while the other end of the spectrum would classify changes only when the system has been completely changed or altered from original intended use.

[13] The population analysis using the change database, as discussed, was a product of using change data already collected. Therefore, this research represents more of an exploratory investigation into the analysis of population-level change characteristics. Future research should control for any sampling biases present in these collected system change examples. One could draft a formal questionnaire to gather system changes in a structured manner from more participants, and collect a larger set of change examples to help minimize any sample biases.

Table 6-1. Ten Dimensions for Characterizing a System Change

Dimension	Name	Source
1	Perturbation	Chapter 3
2	Context	Chapter 3
3	Phase[1]	Ross and Rhodes 2011
4	Agent[2]	Ross, Rhodes, Hastings 2008
5	Effect[3]	Ross, Rhodes, Hastings 2008
6	Parameter[3]	Ross, Rhodes, Hastings 2008
7	Destination[1]	Ross and Rhodes 2011
8	Aspect	Chapter 3
9	Abstraction	Chapter 3
10	Value	Ross and Rhodes 2011

[1] Derived from *Change Mechanism*
[2] Derived from *Change Agent*
[3] Derived from *Change Effect*

In addition to the change information, actual system characteristics were also captured. For each system in the database, information on that system was gathered, including the domain, population size, potential and actual varieties, average lifecycle, production type, and a brief system overview. The name and manufacturer of the system as well as a brief overview of the system was captured. The system domain was selected from one of the following eight categories: (1) Building/Structure; (2) Commercial Vehicle; (3) Consumer Product; (4) Consumer Vehicle; (5) Military Aircraft; (6) Military Vehicle; (7) Space Systems; (8) System of Systems (SoS). Production type was either "mass-produced," "few," or "unique," to capture the differences in manufacturing the system. Potential and Actual Variety were either "one," "few," or "many," to capture the possible and actual variety within the system population. The population size was also either "one," "few," or "many," capturing the size of the population where one is one, few is some reasonably countable number (i.e. 2-100), and many is some larger number (i.e. >100). Average lifecycle was either "less than five years," "between five and fifteen years," "or more than fifteen years" to capture general lifecycle length of the system.

The change information was capture with the addition of a few more fields including, a change description, parameter, path enablers, relative change costs, preliminary assumed ilities and actual mapped ilities. The change description uses the format of the change statement from Chapter 5 and includes any other relevant information and sources as well. The parameter clarifies the exact parameter under consideration. The path enablers are any supposed path enablers that helped to enable the change mechanism based on the system information. The relative change cost was the incurred change cost relative to the system cost and could be "negligible," "much less" (two orders of magnitude), "less" (one order of magnitude), "same" (same as the system itself), or "more" (more than the system cost). Preliminary ilities are the ilities that might have been assumed to be present in the system before inputting the change information into the database (subjective field input by the user), and the mapped ilities are the ilities that the database matches to the MIT SEAri definitions. These fields of the database are summarized in Table 6-2 below.

Table 6-2. Fields used in the Change Database

	Name	Options
System Info	System Name	[PowerTech Diesel Engine;...etc.]
	System Manufacturer	[John Deere;...etc.]
	System Domain	[Commercial Vehicle; Space System;... etc.]
	Production	[Mass Produced; Few; Unique]
	Potential Variety	[One; Few; Many]
	Actual Variety	[One; Few; Many]
	Population	[One; Few; Many]
	Average Lifecycle	[<5 yrs; 5-15 yrs; >15 yrs]
	System Overview	[text] – brief overview
Change Info	Change Description	[text] – using change statement
	10 Basis Dimensions	See Chapter 5
	10 Dimension Explanations	[text]
	Parameter	[text] – specified parameter
	Path Enablers	[text] – available path enabler info
	Relative Change Cost	[Negligible; Much Less; Less; Same; More]
	Preliminary Ilities	[text] – assumed ility
	Mapped Ilities	[text] – actual mapped ility

Together, these categories were used to capture as much information about each system change as possible. A given system could possibly have one or multiple changes associated with it. A unique record is the set of completed fields for a given change example for a given system.

When entered into the fields of the database, each change looked like the example shown in Figure 6-1 below[14][15]. Figure 6-1 shows an example record with a change made to the PowerTech PSX diesel engine of a John Deere tractor. This particular change is in response to tightening EPA restrictions on particulate matter and Nitrogen Oxides (NOx).

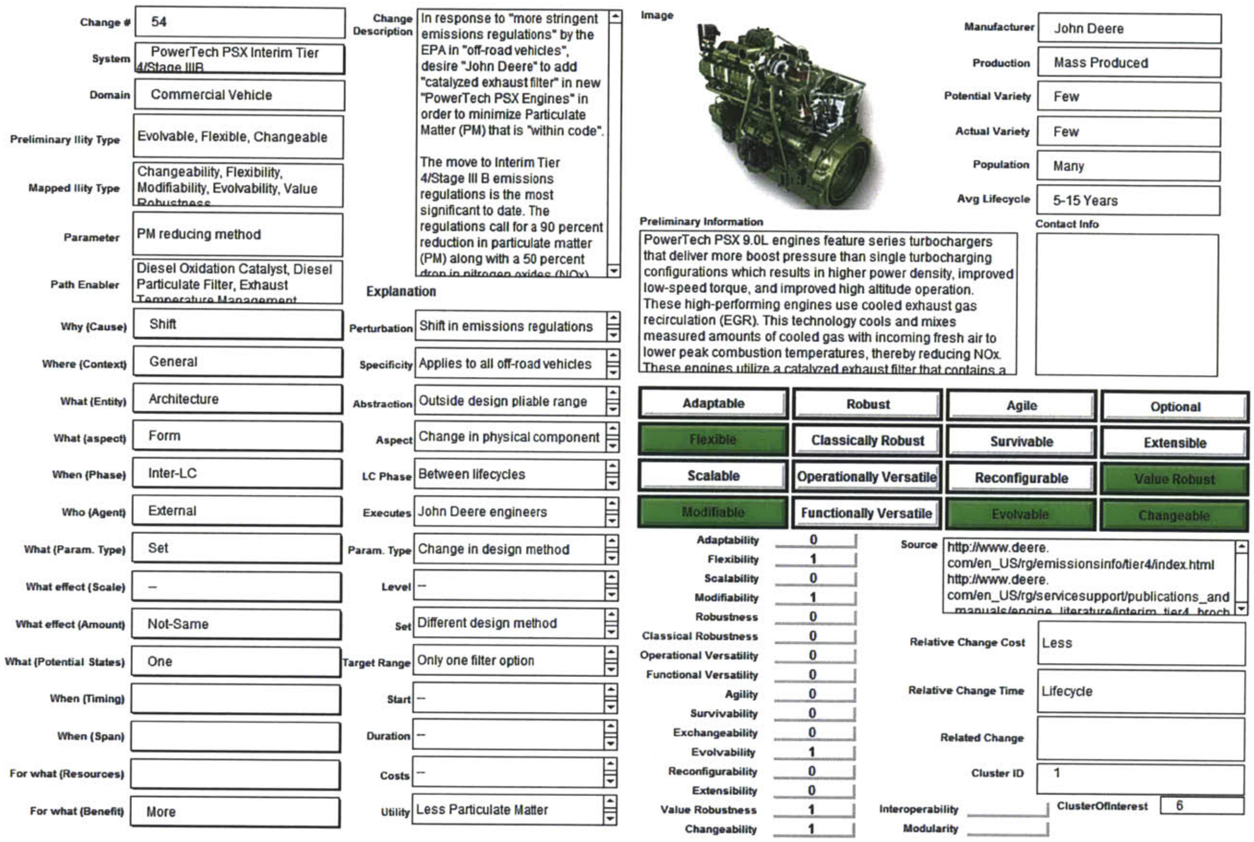


Figure 6-1. Example Record in the Change Database

Running through this example of a record in the database, the change statement may be used to help clarify what is changing in the system. This record's change statement is:

> *In response to "more stringent emissions regulations" by the EPA in "all off-road vehicles", desire "John Deere" to add "catalyzed exhaust filter" as a new Particulate Matter (PM) reduction method in the "PowerTech PSX Engine"*

---

[14] The example shown in Figure 6-1 represents an example and was chosen for no specific reason other than to display a record from the database.

[15] Due to the nature of the database, changing field names and field entry options became more difficult and time consuming when changes were made to the semantic basis. As a result, some of the fields in the Figure 6-1 are labeled with respect to prior versions of the semantic basis. However, the data collected is still consistent with the categories explained in Chapter 5 and 6 of this thesis.

"architecture" between "lifecycles" in order to minimize PM that is "within code."

This statement relates to the responses for the 10 dimensions shown in Figure 6-1. Not all value columns may be known since this information is subjective to specified thresholds. The information that was less subjective was captured and given justifications in the fields adjacent to the 10 dimensions discussed earlier. The database-implemented definition comparisons were derived from the MIT SEAri ility definitions outlined in Table 5-3 and Figure 5-11 in Chapter 5. The matched ilities would light up green if the field values for the 10 basis categories matched the ility definitions in the basis. In this way, ility labels are assigned to the change automatically, and based upon the input of the data. The preliminary ilities were collected to help highlight possible issues when dealing with ilities: incommensurate definitions for ilities (mismatch between a user definition and the definition in the database), or incomplete knowledge (the database auto-considers more ilities than the user may be aware of). In this example, the user correctly hypothesized that *evolvability*, *flexibility* and *changeability* would be ilities present in this change, but overlooked other ilities like *modifiability* and *value robustness*. It is important to reiterate that these ilities are the ilities that relate to this specific change. Instead of labeling a system as having a general ility, the system may have change and resistance options with ility labels; the more options a system has with a given label (e.g. evolvability), then the more it may be considered to display that ility. Additionally, it is possible for a single change to display numerous ilities.

The example shows how information from a change made to a system can be collected and input into the database, and how the semantic basis can be used to assign ility labels to that captured system change. These assigned ilities do not relate to the system per se, but rather to the *change* in the system. This means that the John Deere engine described above may or may not be an evolvable or flexible system; the data is insufficient to show that one way or the other. Instead, this data only shows one example change made to the engine and classifies what kind of change that was. Using to the concept of tradespace networks discussed in Chapters 2 and 3 of this thesis, the captured change is only one "arc" in the network, bringing the system from one state to another. One could conceivably apply these ility definitions to all the possible state changes and begin to say whether the system displays a level of *flexibility* or *evolvability*. This however only applies when looking at all the *possible* changes a system can initiate, as in Epoch-Era Analysis and tradespace networks (Ross and Rhodes 2008a). The historical change database does not look at all *possible* changes, but rather at changes that have actually occurred, and therefore cannot attribute ilities to systems as a whole in this manner.

6.1.3 System Characteristics in the Database

This section serves to describe the system characteristics in the database to show the makeup of the set of system changes that have been captured. The database contains 100 system change examples across 49 distinct systems and systems of systems (SoS). Each of the 49 systems sits in one of eight domains as seen in Table 6-3.

Table 6-3. Database Domain Categories and System Changes Breakdown

	Domain	Systems	Changes
1	Military Aircraft	9	15
2	Military Vehicle	3	9
3	Space System	3	10
4	Consumer Product	18	31
5	Consumer Vehicle	3	9
6	Building/Structure	4	5
7	Commercial Vehicle	4	9
8	System of Systems	5	12
	Total	49	100

The breakdown of the system changes can be visualized in Figure 6-2 below. Consumer products represent the largest group of changes (31%) with the rest of the domains representing about 10-15% of the total changes captured in the database.

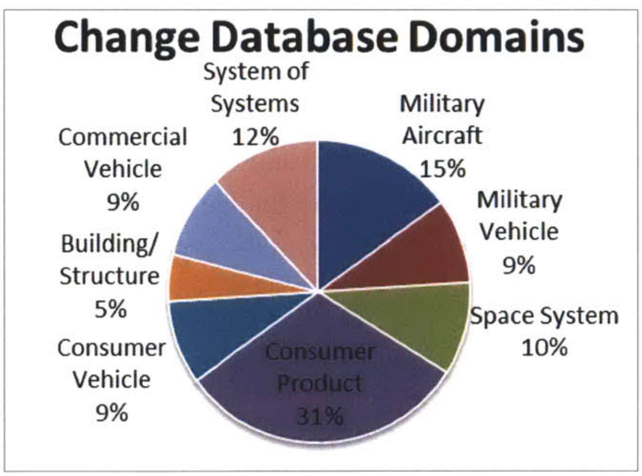


Figure 6-2. Database Changes by Domain

Additionally the changes in the database are broken-down by lifecycle length and manufacturing production type in Figure 6-3 below. The majority of the system changes were mass-produced (68%). The system changes were fairly evenly split in lifecycle length with a slight majority in systems that last longer than 15 years on average.

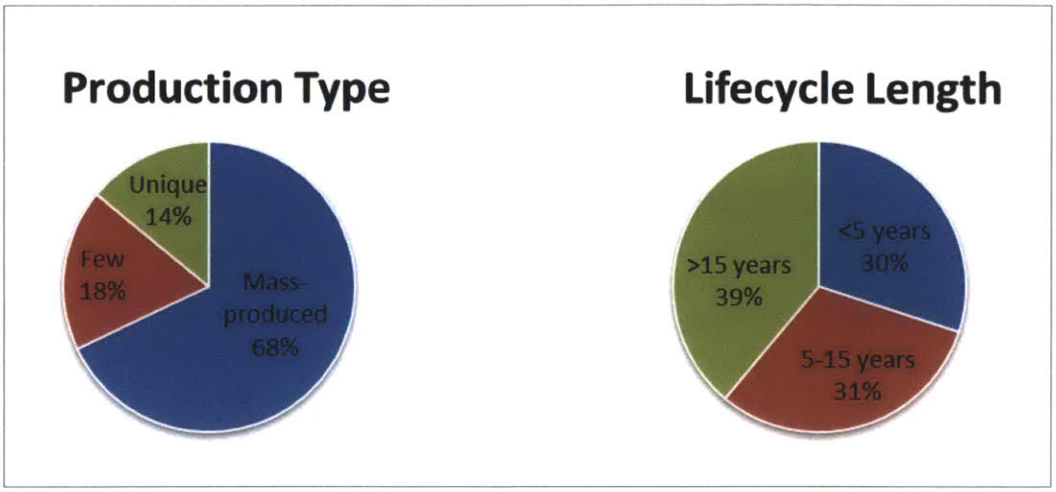


Figure 6-3. Database Changes by Production Type and Lifecycle Length

To get an idea of the break-down of different system changes and the various responses to the fields in Table 6-2, Figure 6-4 and Figure 6-5 show the database field responses and the MIT SEAri-based ility representations of the 100 systems changes. A "1" signifies that a given ility label has been assigned to a change example, and a "0" signifies that that particular ility label is not assigned. The little number above the response represents the number of system changes out of 100 that apply to the field response in the figure.

A report from the database can be found in Appendix B. This report is filtered by four main ilities (changeability, survivability, evolvability, and robustness) with each system and change statement given. Additionally, other ility labels for the particular changes are shown as well. This table can be used for getting ideas of certain types of system change examples and associated ility labels.

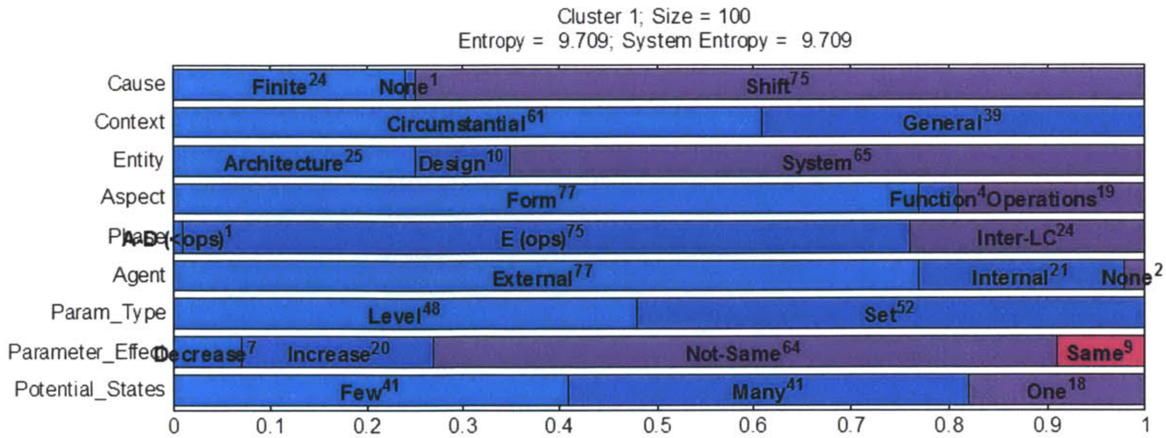


Figure 6-4. Database Field Responses

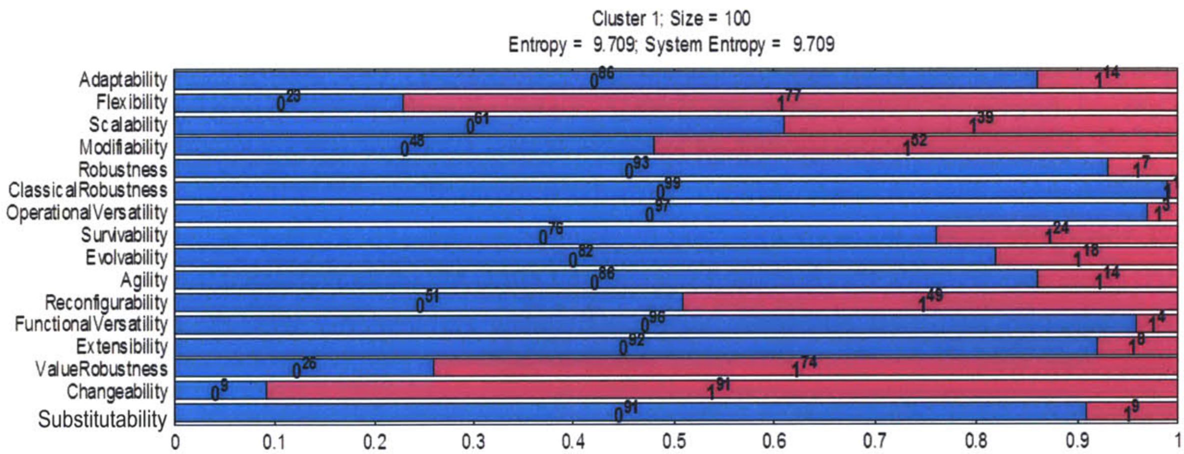


Figure 6-5. Database MIT SEAri-based Ility Representations[16]

As one can see in the data shown above, the database is heavily weighted to "changeability" as these are systems that are related to change-related ilities, and were selected for through the data elicitation by asking for system change examples. The no-change system changes were also captured since resistance to change still relates to the change-related ilities as discussed in Chapter 3. These relate to the "resistance options" described earlier, which are used to inhibit any unwanted change mechanisms from being executed (i.e. imposed) on the system.

[16] The database generated figures use the term "exchangeability" in place of "substitutability," however for the purposes of this chapter, discussion and labels of exchangeability correspond to substitutability as discussed in Chapter 5.

6.2 Categorical Cluster Analysis

Looking for trends within a database full of 100 changes of almost 50 different systems with numerous categories of information in each can be difficult using simple spreadsheet methods or pivot tables. There are populations that we may want to examine within the database based on characteristics of lifecycle length, domain, and productions type,. in order to gain insight into how various populations differ. Trying to find similarities or differences between these groups, however, would be limited if done by inspection, which can be difficult when looking into multiple categories of data for many different systems. Therefore, a numerical clustering method was applied to the database in order to group similar changes together, in order to overcome biases inherent in more subjective human inspection techniques.

6.2.1 Overview

The choice of a particular method for cluster analysis was limited due to the categorical nature of the database. Clustering databases that are completely numerical has been well documented, with many different methods and models that can be used, like the *k-means* or *hierarchical* clustering models (Gan et al. 2007). Clustering is an effective method to partition observations into groups, or clusters, which are more similar to each other than observations of another group. However, clustering with non-numerical categorical data, as in the various categorical responses in the change database, is more difficult than the traditional numerical methods (Barbará et al. 2002). In order to cluster with this type of categorical data, there have been a few different models proposed, and this thesis chose to use the COOLCAT method (Barbará et al. 2002). This method was chosen for its ability to cluster efficiently and scalably with the addition of new records.

6.2.2 Clustering Method: COOLCAT

COOLCAT is an entropy-base algorithm for categorical clustering from Daniel Barbará, Julia Couto Yi Li, (2002). Where many of the other clustering methods rely on some sort of distance metric to measure separation between records, COOLCAT uses the notion of entropy. Entropy, simply put, is an expression of *disorder*. Using entropy as a model for clustering, the objective of the method is to reduce the "disorder" between clusters or minimize the entropy. The name COOLCAT comes from this idea of "cooling" the categorical clusters and is not an acronym.

Entropy is the measure of disorder in a cluster and can be calculated using the Equation 1 below, where X is a random variable, $S(X)$ is the set of values X can take, and $p(x)$ is the probability of a response within the given data (Barbará et al. 2002).

Equation 1

$$E(X) = -\sum_{x \in S(X)} p(x) log(p(x))$$

Applied to multiple records in a dataset, D, of N records, we desire to create k number of clusters, $C_1, \dots C_k$, by minimizing the entropy, $E(X)$. In order to do this, the idea of expected entropy is needed to measure the entropy of the entire set of clusters. The expected entropy is the sum-weighted mean of the entropies of each cluster shown in Equation 2 where $E(P(C_1)), \dots,$ $E(P(C_k))$, represent the cluster entropies, $P(C_j)$ are the points assigned to the cluster C_j, and $\breve{C} = \{C_1 \dots, C_k\}$ (Barbará et al. 2002).

Equation 2

$$\bar{E}(\breve{C}) = \sum_{k} \frac{|P(C_k)|}{|D|} (E(P(C_k)))$$

Equation 2 represents the objective function that this algorithm aims to minimize. The COOLCAT algorithm initializes by finding the pairwise entropy for each record with respect to every other record. It initializes the first two clusters by maximizing the pairwise entropy, or in other words, it picks the two records that are the most dissimilar in the dataset. These two records then become the basis for the first two clusters of k clusters. To finish the initialization of remaining clusters up to k, the algorithm picks the first, or basis, record for each of the remaining number of specified clusters. Proceeding incrementally, COOLCAT maximizes the minimum pairwise entropies of the remaining records. This basically looks at each record's most similar record remaining, and then picks the one that is the most different to start the next cluster. The number of identical fields determines similarity between records. Once all clusters have been initialized with one record, the algorithm places the rest of the records in an appropriate cluster.

To do this, the algorithm implements an incremental step method, seen in Figure 6-6. Going through each of the remaining records, a record is placed in each cluster one at a time and the overall system-wide expected entropy (the weighted average expected entropy of all clusters) is measured. After being placed in each cluster, the algorithm chooses the cluster that yielded the minimum system-wide expected entropy, thus minimizing dissimilarity. After each record has been placed in its final cluster, the algorithm is complete with k clusters.

1.Given an initial set of clusters $\check{C} = C_1, \cdots, C_k$

2.Bring points to memory from disk and for each point p do

3. For $i = 1, \ldots, k$

4. Place p in C_i and compute $\bar{E}(\check{C}^i)$

where $\check{C}^i$ denotes the clustering obtained by placing p in cluster C_i

5. Let $j = argmin_i(\bar{E}(\check{C}^i))$

7. Place p in C_j

8. Until all points have been placed in some cluster

Figure 6-6. Incremental Step Algorithm (Barbará et al. 2002)

6.2.3 Results

With the database filled with the 100 system changes across 49 different systems, comparisons can be made across different sub-groups of the population. The method behind these comparisons was to look into the heuristic-based comparisons of different populations within the change database and compare those findings with any insights gleaned from the COOLCAT clustering method.

One of the ways this research compared results of different sub-groups was to compare the proportional level of MIT SEAri-based ilities between sub-groups. Since these groups are comprised of different numbers of systems, care was taken in comparing one ility proportion to another (Cluster X having 18 of 24 records that display flexibility vs. Cluster Y having 28 of 30). Using MATLAB®, a necessary chi-squared test can be used to determine the statistical difference of two proportions between sub-groups. In this way, each proportional difference may be given a p-value that represents the statistical likelihood that that result could be randomly achieved. Therefore, with a lower p-value, the data suggests that the differences between proportions is not just random, but carries with it some reasonable statistical significance. Below are some of the findings from looking into different sub-groups within the population of system changes.

6.2.3.1 Heuristic Based Results

The heuristic-based comparisons come from comparing different sub-groups of the dataset population to one-another based on various characteristics of the systems, such as lifecycle length or domain. The database keeps track of system characteristics like lifecycle length, production type, system population size, relative change costs, and system domain. These characteristics are used to compare different sub-groups within the database population. For example, military vs. civilian systems, relatively expensive changes vs. relative cheap changes, long lifecycles vs. short lifecycles, and mass-produced systems vs. more unique systems. Some of these comparisons are detailed in this section.

Short vs. Long Lifecycle Lengths

Below in Figure 6-7 is the comparison of short-lifecycles (<5 yrs.) vs. long-lifecycles (>15 yrs.) in the dataset. These sub-groups do not add up to 100 system changes since the mid-lifecycle (5-15 yrs.) group were left out of the analysis.

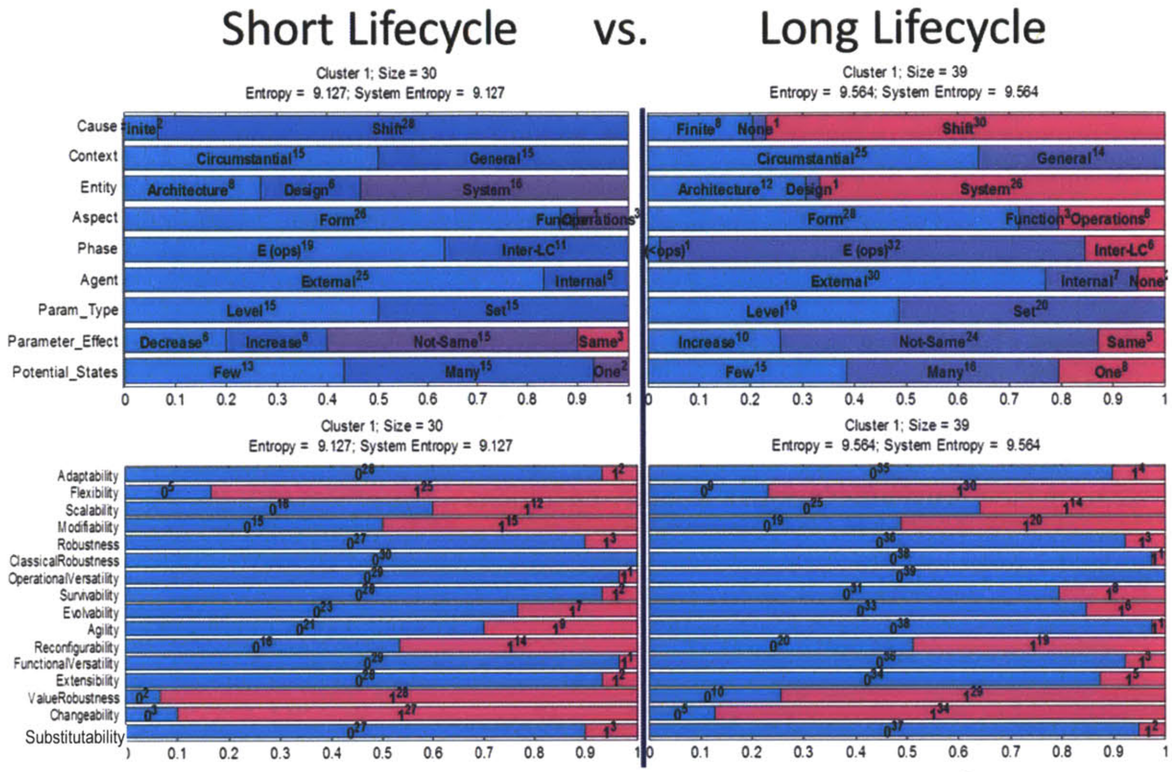


Figure 6-7. Short vs. Long Lifecycle Sub-Group Comparison

In these two sub-groups of the database, short lifecycle system changes were more agile ($p=0.001$), where long lifecycle system changes were more survivable ($p=0.10$). Neither sub-group had an evolvable characteristic significantly different from one another.

When we look to lifecycle length with respect to evolvable type changes in systems, it might be expected that the more evolvable systems would be those with shorter lifespans due to the more frequent changes. However, this data showed little statistical difference between the evolvable type changes between those sub-groups ($p=0.40$). When we look into biological literature on lifecycle length and evolvability characteristics (Stearns 1977), Life History theory may explain these results. Some evolutionary biologists believe that there are two main strategies for evolution, which depend on the environment of the species and largely trade-off number of offspring to length of reproduction. One strategy is "r-selection" where species have high growth rates, many offspring, short lifespans, high mortality rates, and minimal parental investment. This strategy thrives in rapidly changing environments, where traits that are successful one day, may soon become unsuccessful, and therefore are usually less complex or biologically expensive (requiring lots of resources or time to develop). On the other side, "K-selection" species mature more slowly with longer lifespans, have lower mortality rates with fewer offspring, and require more parental investment. These systems thrive in environments that are more stable that can be optimized for survival (Stearns 1977). Both of these strategies are present in the world and can be successful, which is consistent with the data shown above. However, the short lifecycle sub-group relates to the r-selection strategy not only in lifespan, but also in agility. The short-lifecycle systems are made to change generations *faster*, similar to r-selected species reproducing more frequently. Similarly, the K-selection strategy relates to the longer lifespan systems in that there is a higher level of survivability. When a system is required to live longer, more survivable aspects are desired, to deal with disturbances to the species/systems. The results from the database are consistent with these biological characteristics.

Consumer vs. Non-Consumer Systems

In Figure 6-8 below are the population statistics of the consumer products vs. the non-consumer products sub-groups. These are systems that were designed for consumer use or sale as opposed to systems made for manufacturing or military operations.

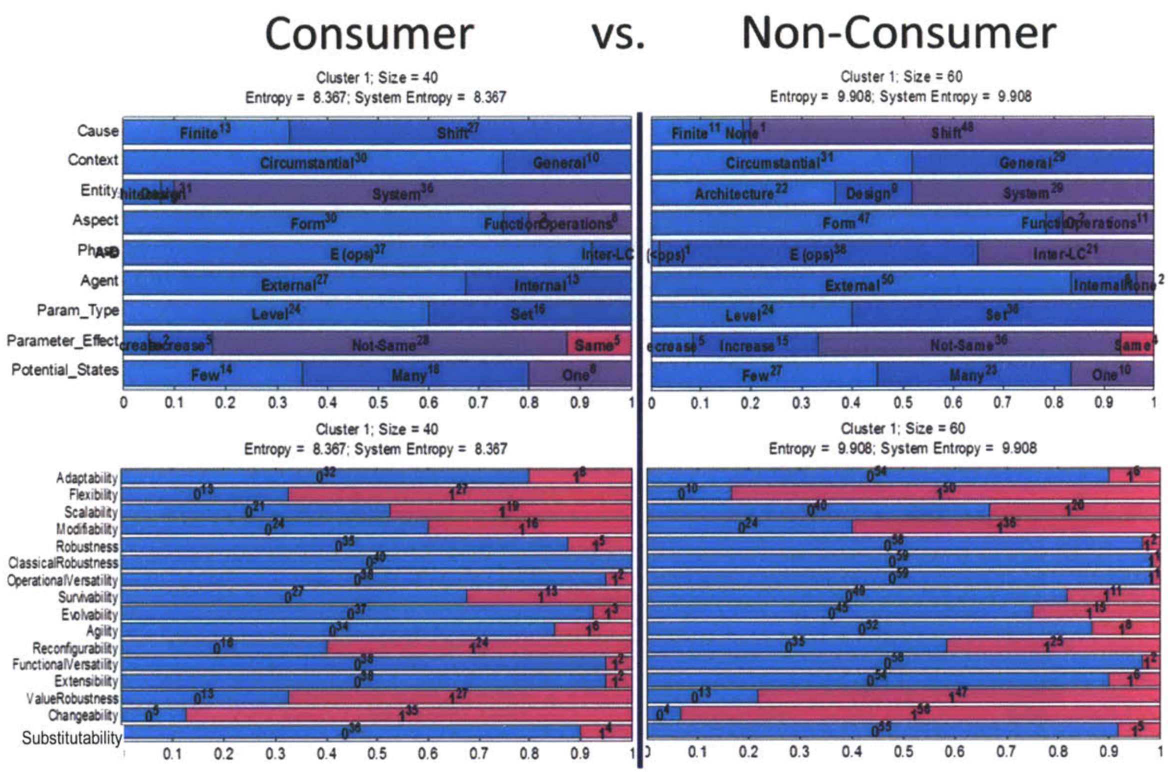


Figure 6-8. Consumer vs. Non-Consumer Sub-Group Comparison

Consumer product changes were slightly more scalable (p=0.15), more survivable (p=0.10), robust (p=0.08), and reconfigurable (p=0.07). Non-consumer product changes were more flexible (p=0.06), more modifiable (p=0.05), and more evolvable (p=0.03).

Consumers often demand systems that are robust and survivable to many different disturbances or shifts in needs, as well as systems that can be reconfigurable to those needs as well. Therefore, these characteristics in consumer products are consistent with data above. The scalable type changes in consumer products also follows to the types of advertising and platforming used in consumer products. Commonly, consumer products like iPods, cell phones, or computers are sold or upgraded on the basis of scaling a certain parameter (spend more to scale up speed, capacity, or size for example).

Space Systems vs. Terrestrial Systems

In Figure 6-9 below, you can see the population statistics of the space systems vs. terrestrial system sub-groups.

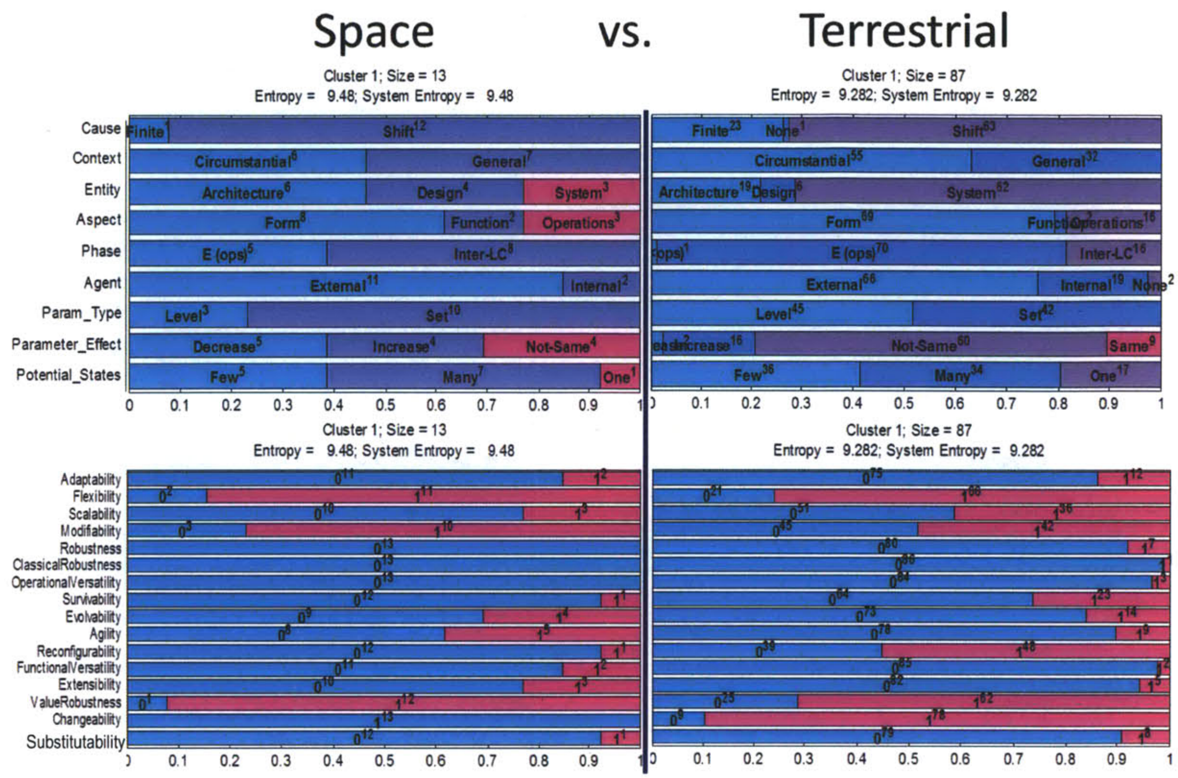


Figure 6-9. Space vs. Terrestrial Sub-Group Comparison

Space system changes were more modifiable (p=0.05), evolvable (p=0.19), agile (p=0.01), extensible (p=0.03) and valuably robust (p=0.10). Terrestrial system changes were more survivable (p=0.14) and reconfigurable (p=0.001) than space systems.

These results were somewhat perplexing. Some of the trends do not entirely make sense at first examination. For instance, one would tend to believe that space systems are not so much extensible and should be more survivable. This artifact could be due to the small sample size of space systems present in the database (just 13%).

The evolvable changes in space systems seem more reasonable however. These systems commonly are very generational in that new systems build off heritage systems, inheriting much of the prior work or research. The scalability of space systems also seems reasonable due to space systems using common architectures and scaling to different needs [the Evolved Expendable Launch Vehicle (EELV) scaling to different launch mass requirements (Saxer et al. 2002)].

High Population vs. Low Population

In Figure 6-10 below is the comparison of high population system changes vs. low population changes, where low population systems are those with some reasonably countable number (up to a hundred) and high population systems are those with more than one hundred. These results are very similar to the comparison of mass-produced vs. unique system changes as well.

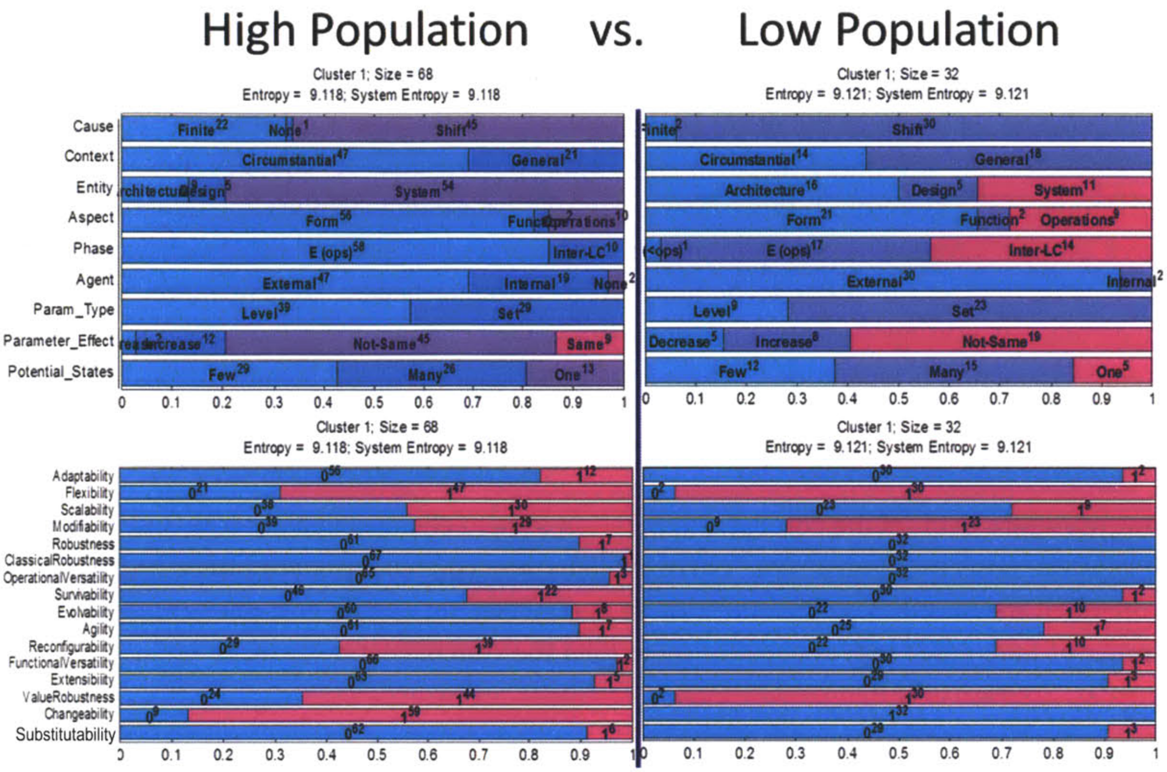


Figure 6-10. High Population vs. Low Population Sub-Group Comparison

High population system changes were more adaptable (p=0.12), scalable (p=0.13), robust (p=0.04), survivable (p=0.004), and reconfigurable (p=0.01), where the low population system changes are more flexible (p=0.006), modifiable (p=0.006), evolvable (p=0.000002), and agile (p=0.12).

It is interesting that the high-population systems and mass-produced systems (populations that are very similar but not quite the same) had more changes that were considered adaptable, meaning that many of these changes were internally executed by the system. In many cases, these changes can be associated with autonomy. This research may suggest that these systems meant for use by large numbers of users have been designed to be more autonomous in nature and adjust themselves for the user. This is initially counter-intuitive in that one would generally think the more unique systems or those of low-population would have more engineering thought put into doing things adaptably (sensing a need to change and executing that change internally). However, these mass-produced and high-population systems have bigger markets. This may relate to markets where more money can be spent on engineering "smart" systems that have the technology to adapt. Looking closer into these adaptable systems, we find that they are often luxury systems that come with higher costs, whether that be headlights on a luxury vehicle that adapt to changing conditions without input from any external change agent, or electronic countermeasures on fighter planes that automatically determine threat scenarios and adaptively adjust to appropriate responses.

The research also shows that the changes made in mass produced systems are more at the "level" type parameters, or changes the relate to the scalability of a system, where more unique systems have changes that are more in the "set" type parameters, or changes that relate to the modification of the system. In this case we can see that systems that are manufactured to appeal to large groups may have more parameters with adjustable levels (how much storage do you want in your device?) in order to be adjusted to different needs. Where more unique system changes are not scaling a parameter level, but rather changing the parameter itself, possibly changing in function or operations, which is what the data shows in the more unique systems.

Low Cost vs. Higher Cost Changes

Below, in Figure 6-11, is the comparison between system changes that are low cost (proportional to the system cost) and higher cost changes. The low cost changes are either negligible or are two orders of magnitude less than the system cost, where the higher cost changes range from just one order of magnitude less to equal or more than the system cost.

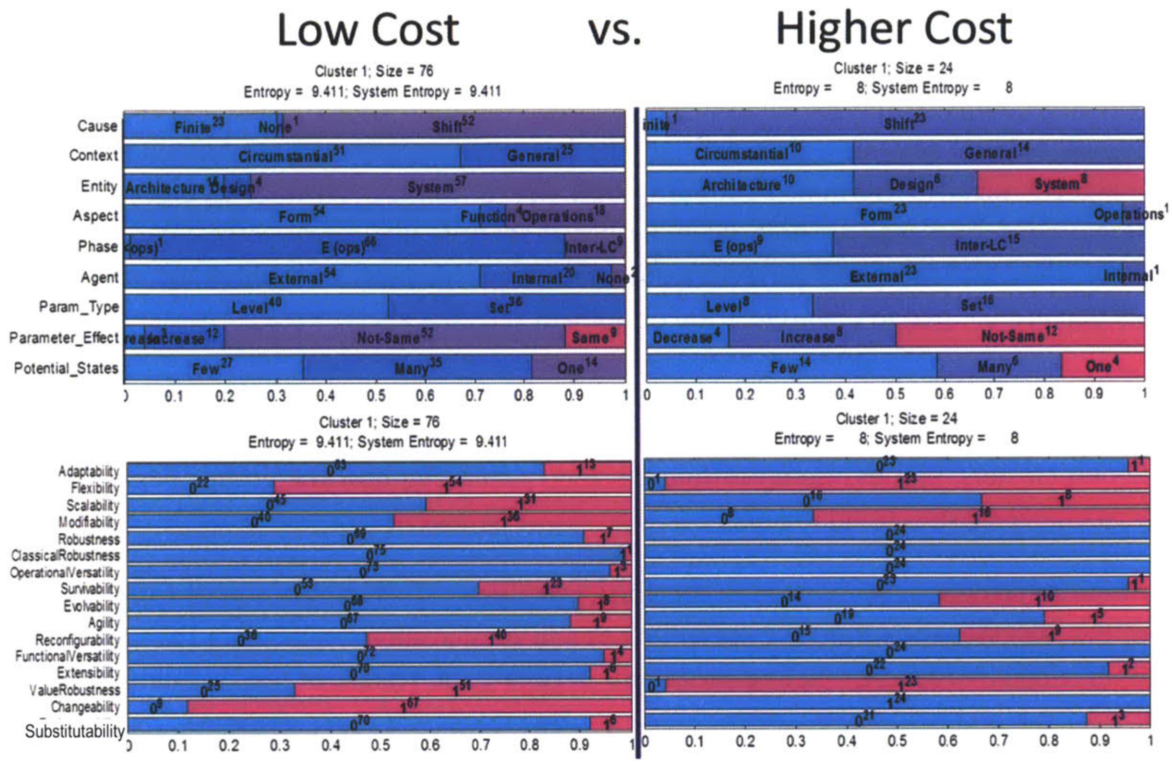


Figure 6-11. Low Cost vs. Higher Cost Changes Comparison

Low cost changes were more adaptable (p=0.11), survivable (p=0.01), and reconfigurable (p=0.19), whereas the higher cost changes were more flexible (p=0.12), modifiable (p=0.10), and evolvable (p=0.0005).

These results are interesting when comparing flexible changes to adaptable changes. Generally, systems that are designed to be adaptable require extra upfront costs that enable the system to sense the need to change and then execute that change internally. While these systems require extra upfront costs, the actual change costs associated with these changes are generally lower due to the extra foresight applied during design. This data is consistent with those ideas.

Reconfigurable changes may relate to cheaper change costs since the components are staying the same, just the relationship of the components are changing. Since no new components are being added, this makes sense that the change is generally cheaper.

Additionally, the evolvable type changes are shown to be on the more expensive side. When systems are being re-architected, these changes are generally very expensive, as a new system is being formed, which is consistent with this data.

6.2.3.2 *COOLCAT Results*

These past comparisons were made on a heuristic basis in order to gain insight in trends in different sub-groups within the dataset population. The following clusters are the result of the COOLCAT method and show data that may be less obvious, grouped not by heuristic tags, but by the algorithmically determined similarity of a change to other types of changes. Comparing categorical clustering analysis with heuristic based analysis may show some interesting results, that those heuristics are good ways to parse various populations, or if there are other types of clusters that are not tied to manufacturing or domain specific labels.

The inputs to COOLCAT are the change records in the database, which fields in those changes to evaluate and the number of clusters to form. This application of COOLCAT begins with the simple two clusters, and moves up to four clusters, evaluating how the groups begin to separate and what characteristics trend together.

Not all of the categories were used in the cluster analysis. This application of COOLCAT uses nine of the ten categories from the semantic basis to cluster upon—all except the value category due to its subjective nature. Below, in Figure 6-12, are the results of the two clusters from the COOLCAT method.

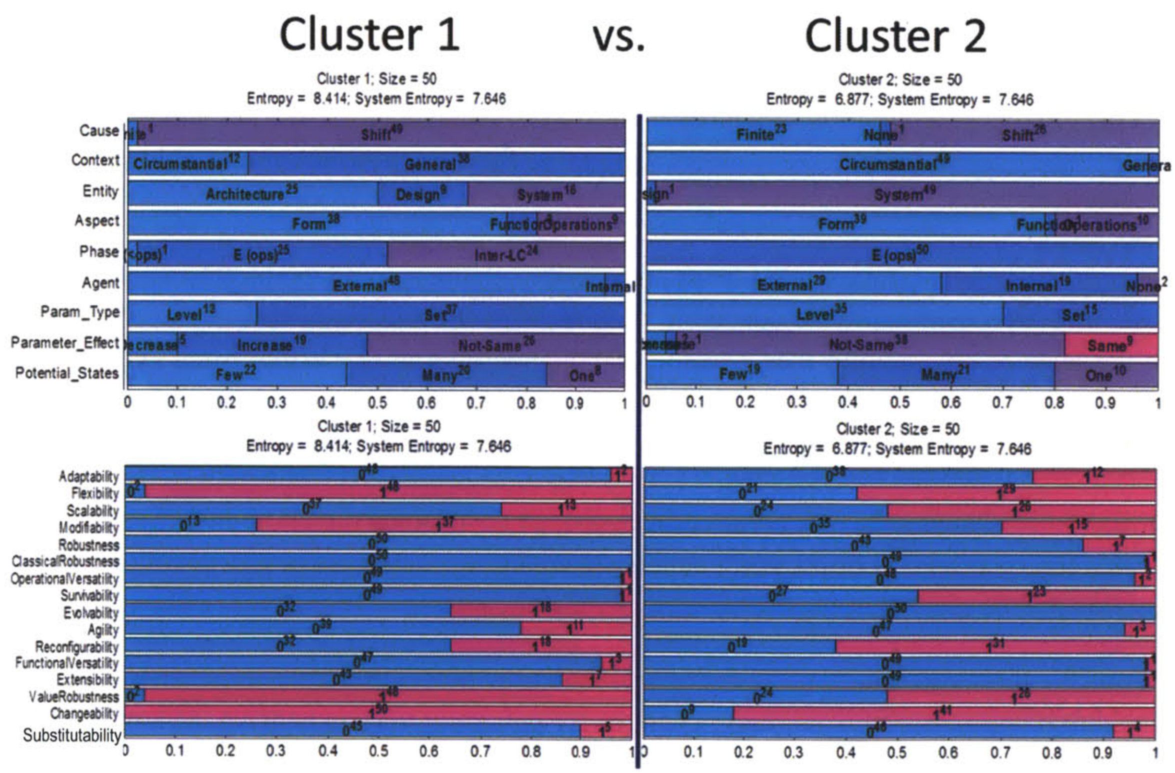


Figure 6-12. COOLCAT Cluster Results (2 Clusters)

The clusters can be broken-down by domain as seen in Table 6-4 below.

Table 6-4. COOLCAT Cluster Domains (2 Clusters)

	Cluster 1	Cluster 2
Military Aircraft	8	7
Military Vehicle	7	2
Consumer Product	10	21
Consumer Vehicle	0	9
Commercial Vehicle	2	7
Building/ Structure	2	3
Space System	9	1
System of Systems	12	0
Total	50	50

The COOLCAT clustering tool split the 100 system change examples into two clusters of equal size, completely coincidentally. Cluster 2 has a lower expected entropy (entropy of 6.88 compared to 8.41) and is therefore a tighter cluster than Cluster 1. Cluster 1 tends to be more flexible (p=0.000006), modifiable (p=0.00001), evolvable (p=0.000003), agile (p=0.02), extensible (p=0.03) and changeable (p=0.002). Cluster 2 tends to be more adaptable (p=0.004), scalable (p=0.16), robust (p=0.003), survivable (p=0.0000003), and reconfigurable (p=0.009). These p-values are much lower than the basic heuristic based cluster since COOLCAT has found clusters that are very similar to one another in change type. This is an expected result, and shows that the COOLCAT method is working as an effective clustering tool.

The differences in changes between COOLCAT clusters correlate very closely with the differences in the mass produced vs. unique comparison or the high population vs. low population comparison. However, while Cluster 2 is almost completely filled with high population type systems, Cluster 1 has a more even spread of "one," "few," and "many" in population size (13, 15, 22 respectively). This shows that there were some mass produced systems that differ from other mass produced systems in way that they are more similar to the unique systems. These mass produced systems that changed sides generally have change costs a little higher, are more flexible in form, and are more in response to shifts with a few at the inter-lifecycle phase. Since relative cost of change is not something that COOLCAT even looks at when clustering (COOLCAT only clusters on dimensions from the semantic basis, relative change cost was added just in the database), this reinforces the idea that inter-lifecycle changes and flexible changes may have higher costs when executed, as opposed to adaptable changes.

The other interesting quality in these clusters is that some of the domains are completely exclusive to one cluster. SoS are exclusive to Cluster 1 and consumer vehicles are exclusive to Cluster 2.

To get more variation in the clusters, we can have COOLCAT form four clusters as well. Figure 6-13 and Figure 6-14 below show the results of the four clusters from the COOLCAT algorithm.

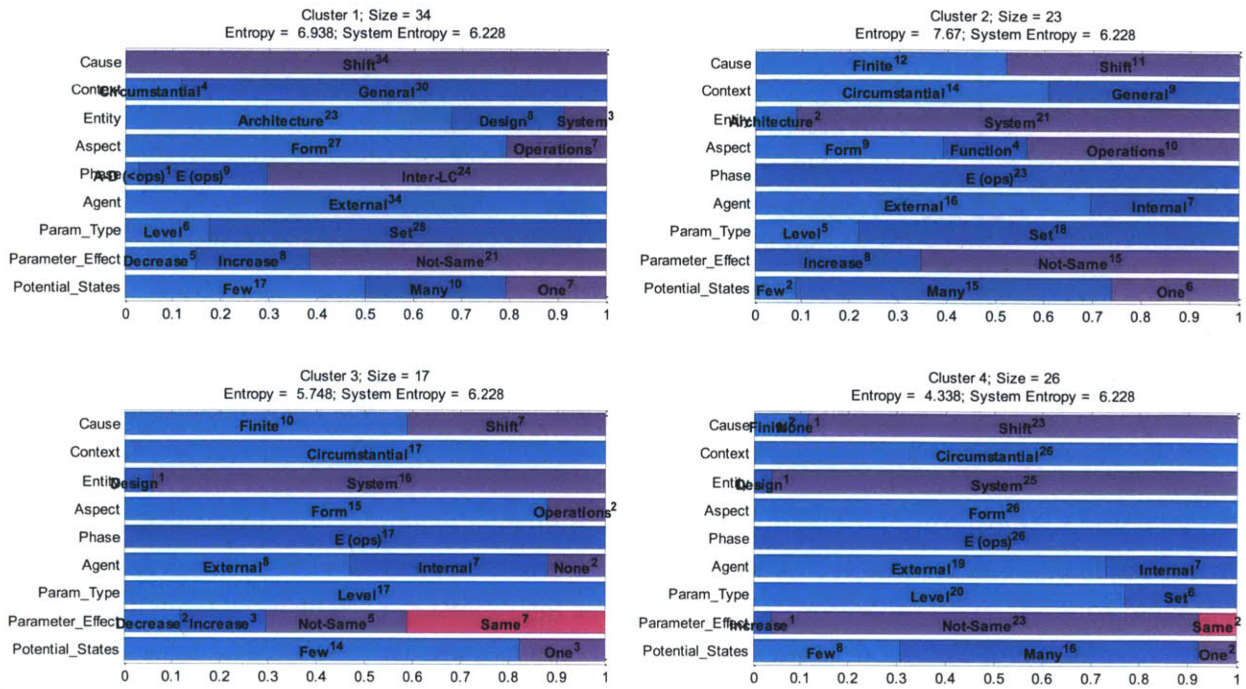


Figure 6-13. COOLCAT Categorical Cluster Results (4 Clusters)

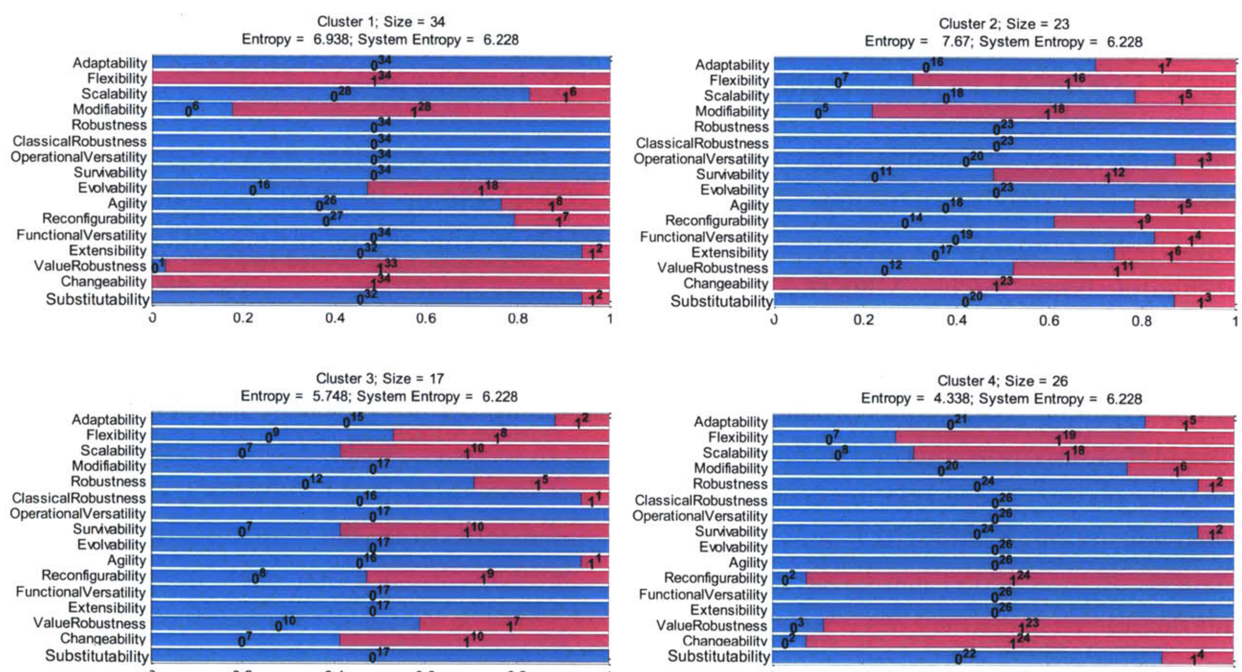


Figure 6-14. COOLCAT Ility Cluster Results (4 Clusters)

The clusters can be broken-down by domain as seen in Table 6-5.

Table 6-5. COOLCAT Cluster Domains (4 Clusters)

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Military Aircraft	4	0	4	7
Military Vehicle	6	1	0	2
Consumer Product	5	7	9	10
Consumer Vehicle	0	5	1	3
Commercial Vehicle	2	3	1	3
Building/ Structure	0	2	2	1
Space System	7	3	0	0
System of Systems	10	2	0	0
Total	34	23	17	26

Cluster 1 is more flexible and evolvable. Clusters 1 and 2 are more modifiable, where Clusters 3 and 4 are more scalable. Clusters 2 and 3 are more survivable. Cluster 4 is more reconfigurable. Clusters 1 and 4 are more value robust. Cluster 3 is more robust, where Clusters 1, 2, and 4 are more changeable. Cluster 2 is more extensible. These ility labels are summarized below, in Table 6-6.

Table 6-6. COOLCAT Cluster Ility Labels (4 Clusters)

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Size	34	23	17	26
Flexible	XXX			
Evolvable	XXX			
Modifiable	XXX	XXX		
Scalable			XXX	XXX
Survivable		XXX	XXX	
Reconfigurable				XXX
Value Robust	XXX			XXX
Changeable	XXX	XXX		XXX
Robust			XXX	
Extensible		XXX		

This data is consistent in that military and space system changes tended to be more evolvable and flexible. These changes in Cluster 1 tended to be more expensive, all in response to a shift in context, changing a set type parameter, and externally motivated. This group is not distinguished by lifecycle length, population, and production type in any significant way, but it does have slightly fewer mass produced and lower population systems. The cluster also contains the majority of the SoS changes.

Cluster 2 is the least tight group. It has the majority of the adaptable system changes, which tend to be the mid-lifecycle length range.

The tightest and smallest group is Cluster 3, which is comprised mostly of consumer products and military aircraft. These systems are robust and focus on level type parameters during operations. They are mostly mass-produced systems that responding to specific circumstances as opposed to general contexts. The types of changes with these systems tended to be more survivable, robust, and scalable, similar to the high population and consumer-based heuristic comparisons.

The four clusters generally represent the three levels of lifecycle length evenly. This suggests that lifecycle length may not be the strongest indicator of system change similarities. The bigger differentiators were the system changes that take place between lifecycles to higher levels of abstraction, and production type or population level.

6.3 Insights

In order to determine if the clusters from COOLCAT were significantly more similar than just random clusters, Monte Carlo simulations were run in two different ways with expected system entropy as a measure of cluster significance. The first method took existing data and randomly placed each system change into a cluster. The expected system entropy was then calculated for each of these random clusterings in 1000 trials. Figure 6-15 shows that the COOLCAT generated cluster are tighter clusters (lower expected system entropy) than the Monte Carlo simulation randomly generated clusters (higher expected system entropy). The difference becomes larger as more clusters are created, because COOLCAT can find more similar and tighter clusters with more clusters to make, which results in improved performance.

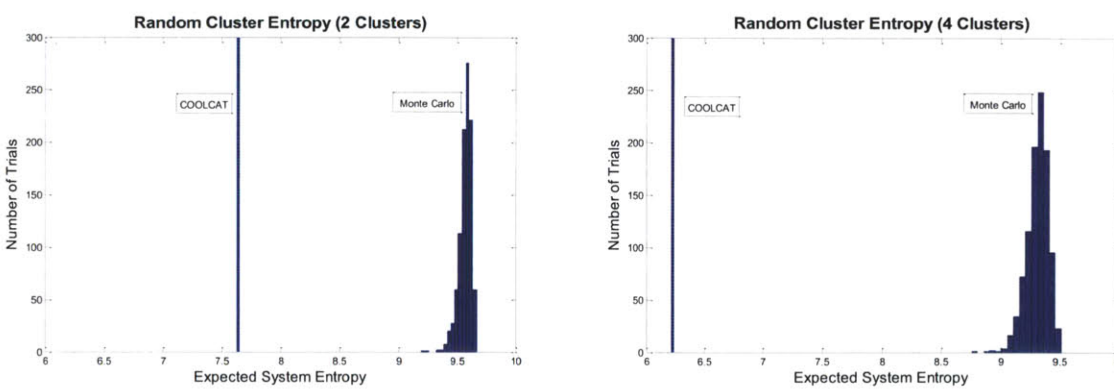


Figure 6-15. Expected System Entropy of Random Clusters vs. COOLCAT Clusters

While this shows that the COOLCAT algorithm is working properly in finding tight clusters within the population, it does not say anything about the quality of data inside those clusters. Therefore, another Monte Carlo significance test was run, this time using "fake data" that is clustered using the COOLCAT algorithm and comparing that to the "real data" clustered by the

COOLCAT algorithm. Since we know COOLCAT works by finding clusters, this test will show if there is any meaningful difference in the empirically collected data, versus randomly generated sample data. The first step in doing this was to generate a set of fake system changes that represented the frequency proportions present in the real data (of each of the responses within each category) so that the generated data represented the real data at the population level. After the generated samples were created, the COOLCAT algorithm could be implemented on them, clustering 1000 trials of the generated samples.

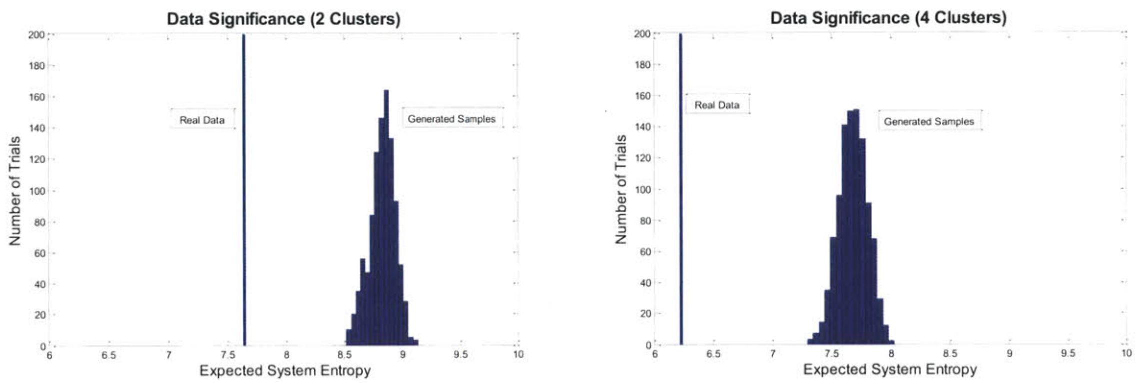


Figure 6-16. Entropy of Clusters from Real Data vs. Entropy of Clusters from Generated Samples

Figure 6-16 shows the real data had on average less entropy (i.e. tighter clustering) than the generated data, even though both sets of clusters were made using the COOLCAT algorithm. These results suggest that the empirical data collected implies certain trends in the real world that become present when looking at sets of system changes as opposed to randomly generated system changes.

The relationships discussed above in the heuristic-based comparisons and COOLCAT generated comparisons suggest that there are sub-groups within the dataset that do not align perfectly with any heuristic-based approaches. In the 2-cluster COOLCAT application, the clusters were largely separated similarly to the high population heuristic comparison. This may suggest that system population may be a leading indicator of different change strategies that could be used, however, with such a small sample set, these results are mostly exploratory.

This exercise provided interesting insight into different sub-groups of a population of system changes. Future work with more in depth and controlled studies could possibly find more complex and statistically significant relationships between change strategies and system domains, population sizes, or other system characteristics that could be useful when designing new systems. While this area of research was largely exploratory into looking at population-level changes, one of the biggest benefits of the change database was the exercise of creating it. This collection of system changes was gathered in conjunction with the semantic basis from Chapter 5, and helped to address pitfalls in the basis when applying it to real world system changes. Some of these benefits can be seen in the need for specifying parameters, or thresholds within the basis to make the change statement more clear, or clarifying what the "destination states" category applies to (the parameter destination, or the system destination). These insights were possible due to the diverse set of system changes that the change database attempted to classify within the semantic basis. Overall, the change database was a useful tool in examining sets of system changes and shows promise for future research in population-level analyses.

7 Expert Interviews

The final thrust of this research is to garner targeted, more in depth insights from systems engineers, both current and retired, who are known for their work in incorporating ilities into system design. To tie this research back to the "state of the practice," interviews were conducted with these individuals to gain their insight into the problem of formulating verifiable ility requirements and methods of determining what ilities to include in design. This chapter describes the questionnaire and responses from systems engineers when asked about how they design systems and SoS so that they can change in order to continue delivering value in spite of perturbations.

7.1 Motivation

This research primarily deals with the evolvability and changeability of systems and systems of systems (SoS). Maintaining system performance in the presence of uncertainties in design and operating environments is both challenging and increasingly essential as system lifetimes grow longer. In response to perturbations brought on by these uncertainties, such as disturbances, context shifts, and shifting stakeholder needs, systems may be able to continue to deliver value by being either robust or changeable. This research aims to find insights into how systems and SoS might be designed so that they can change in order to continue delivering value in spite of perturbations.

In order to gain insight from industry and government practitioners, interviews were conducted with systems engineers in various domains and with various levels of experience. The average length of these interviews was about an hour, and focused on how the interviewee (i.e., the system engineer, system designer or lead engineer) had experience with systems engineering and ilities. The insights from these interviews serve to ground the other research in this thesis with practical limitations, while gaining empirical knowledge in how "real-world" systems engineering approaches similar problems.

7.2 Method for Interviews

A formal interview process was developed in an attempt to structure the interviews as much as possible for data collection. However, since these interviews were largely dependent on the experiences that the interviewees had when dealing with their particular system, the interviews tended to be semi-structured and varied in length and content. In the interview, examples of decisions were gathered, which were made during the design and development of systems in order to create, enable, or enhance certain lifecycle properties, such as flexibility or evolvability. Traditional "ilities" such as reliability, maintainability, and availability (RMA) are still important; however, the focus of the interview was to better understand how practitioners pursue less well-understood non-traditional temporal lifecycle properties.

To conduct these interviews, a questionnaire was developed and can be found in full detail in Appendix C. This questionnaire was approved through the Massachusetts Institute of Technology Committee On the Use of Humans as Experimental Subjects in January 2012. This approval stipulated that, unless given permission from the interviewee, all names and data will be kept confidential. Therefore, the implications and examples from these interviews are discussed in a manner that conceals necessary identities.

Interviewees were selected for their experience or expertise in their respective domains in systems engineering or system design. Contacts were collected through feeder interviews with researchers from various system engineering and engineering departments at MIT. Potential participants identified through this endeavor were contacted and subsequently interviewed. Suggestions for new systems engineers to interview were gathered from these interviewees, resulting in an expanding list of potential participants for possible interviews. In this way, practitioners from a wide range of domains and systems were captured.

7.3 Questionnaire

As a part of this research, interviews were conducted with government and industry practitioners who have experience designing or redesigning systems with flexibility, adaptability, or evolvability (or possibly other ilities). The aim of these interviews was to find insights into how systems engineers intentionally pursue temporal lifecycle properties in systems and SoS. The interviewees were given a questionnaire to guide the interview that usually lasted about an hour, but ranged from short as 30 minutes to as long as 2 hours.

The complete introduction and questionnaire can be found in detail in Appendix C. The questionnaire with practitioners focused on the experiences and specific design or operations decisions they have made in order to inform prescriptive design flows for including ilities in design.

7.4 Results

This exploratory interview research to gather information empirically from experienced system engineers culminated in 25 interviews that ranged from 30 minutes to almost two hours. The average interview lasted slightly less than an hour.

The 25 systems engineers represented a wide range of domains and systems including aerospace engineering, IT security and development, electrical engineering, mechanical engineering, consumer and commercial auto-manufacturing, space and missiles, defense, communications, command and control, materials, naval engineering, and systems of systems. Experience of these practitioners varied from as high as 50 years to as low as 1 year, averaging 23 years (Figure 7-1).

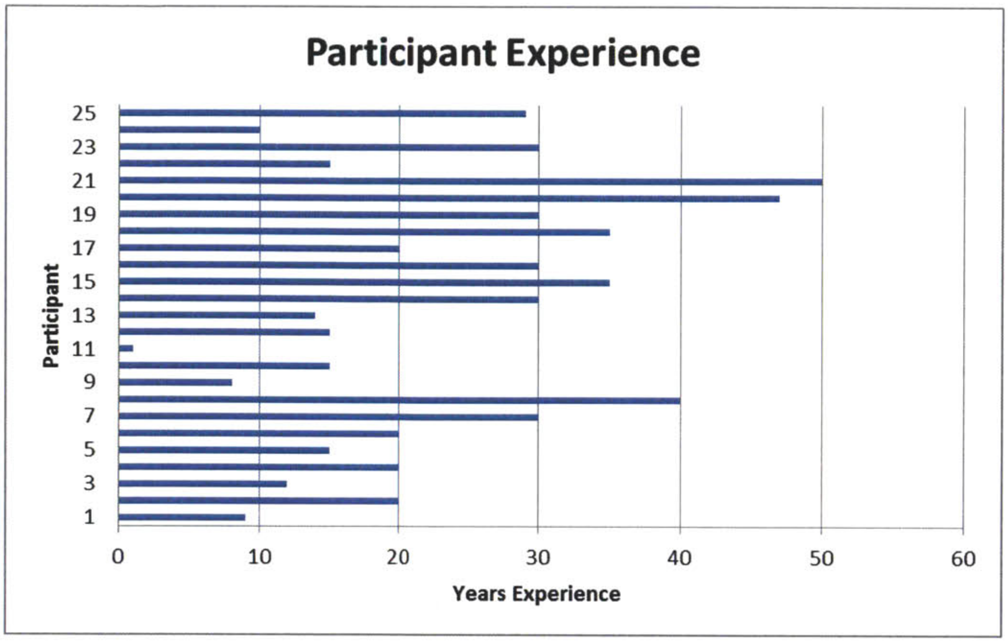


Figure 7-1. Participant Experience

Many of the practitioners have high-level degrees in engineering, over a third with PhD's and over two thirds with masters (Figure 7-2). The academic experience, coupled with many years in practice, represents the level and quality of experience among the 25 interviewed engineers. Many of these interviewees have served as chief engineers, program managers, or lead systems engineers in multi-million and billion dollar programs.

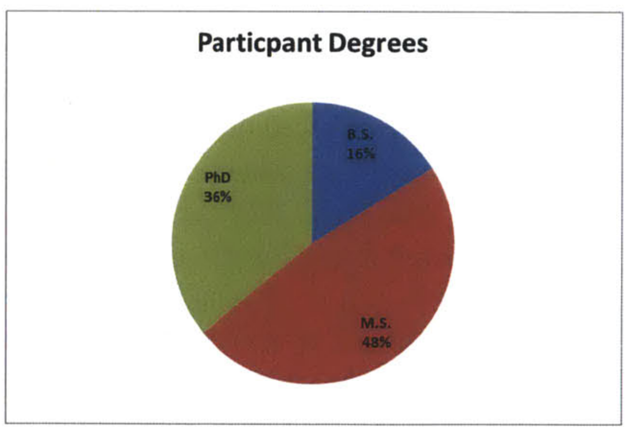


Figure 7-2. Participant Highest Academic Degrees

Participants were asked about their experiences in implementing the desired system properties[17], described earlier as "ilities," into their systems, and what decisions were made in this endeavor.

[17] This particular interview attempted to elicit ility discussions focusing on evolvability, adaptability, and flexibility, however, participants were allowed to talk about any ility they have worked towards in the past.

The interviews tried to capture the desired ility of the system in question (e.g. "the ship needed to be flexible to new components"), why and where that ility requirement came from (e.g. "high ranking government officials felt the new style of warfare required new ships to be flexible in this manner"), and how those qualities were implemented into the system. In addition to this information, the interviews attempted to capture the cost-benefit considerations in these decisions. Participants were asked if the implementation cost extra money, and whether or not any added value was actually perceived by stakeholders.

The second part of these interviews involved a change in scope, moving away from a specific system, towards more general insights in designing for ilities in systems. Participants were asked whether the proposed relationships (Chapter 3.3.3) between design principles, path enablers and change mechanisms were consistent with their experiences in "real life." They were also asked how the ilities relate to one another, if certain ilities enable others, or trade off in any way.

Largely, the interviewees rejected the existence of one universal hierarchy of system ilities, consistent with the discussion in Chapter 5 of this thesis. One of the common threads with many of the responses to the existence of ility hierarchies, or relationships, had to do with the *priorities* of the system stakeholders. Many of the participants mentioned the critical or priority ility that was desired would be considered the high-level ility. Different strategies to achieve that ility or system quality could possibly be other ilities or possibly design principles. This idea that the critical or priority ility represented the top ility depended on the mission requirements and what decision makers demanded of the system. This suggests that hierarchies may need to be redrawn or derived based on varying needs or requirements.

Another common thread in these responses is that these ilities did not drive costs, but rather cost constrained the ilities. In most cases, participants commented on how stakeholders had to scope how many ilities or which ilities are required based on development costs. Since many of these ilities are desirable qualities, most stakeholders would choose to implement many of them in design if there were no impact on system cost. However, securing funding in early stages of development can be difficult, and therefore the justification for spending extra on ilities of a system is often considered subordinate to the primary functionality of a system. One interviewee clarified that the government "will not choose to spend money on an ility like flexibility if that money is needed to enable functionality in the system or to meet a specific requirement." This relates to a common problem discussed in the systems engineering world that justification for spending extra up front to save money in the future is a challenging problem to overcome.

The best case scenario for systems engineers to justify the addition of ilities in design is when the implementation can save money up front and possibly save money in the future. There were examples of this type of ility implementation taking place in some of the interviews of this study. Participants argued that ilities need the backing of a sound business decisional analysis, like real options, to help justify any changes or new costs to system stakeholders or funders.

Another common idea that arose in different interviews with practitioners was the idea of implementing "hooks" or input points in early design in order to take advantage of them later in the system life. The concept of "hooks" was described in numerous ways as "options," "flexibility points," "hooks," or "leverage points." The idea behind them is that a designer can make it easier for some change to be made in the future by pre-engineering how that change will be implemented in the existing system. If hooks could not be explicitly defined, things like extra capacity or margin will help future engineers trying to implement a system change. More on "hooks" will be discussed in the next section of this chapter.

7.5 Preliminary Insights from the Interviews

Since every interview conducted was distinct in the topics discussed, the interview data was not in a format that could be readily compared statistically, such as would be the case in responses to a multiple choice survey. These semi-structured interviews were exploratory, trying to dig deeper into how systems engineers implement these somewhat ambiguous system qualities, or ilities, in design. Therefore, many of the responses were narratives of personal beliefs toward ilities that have been formed through the education and experience of the interviewees. This section contains some of the insights that have been taken from these talks that are preliminary and exploratory, rather than statistically-based. These insights can serve to motivate future research and preliminarily represent some of the concerns that professionals have with regard to systems engineering and ilities.

One of the aspects of designing for ilities that was brought up during a few of the interviews had to do with designing for changeable systems. The idea behind these changeable systems is the ability to alter some part of the system's form, function, or operations to improve performance when subjected to new needs or new contexts. These changes are made easier when the system being changed has had design "hooks," or what this thesis refers to as "path enablers," implemented in the original design. The idea of using hooks or path enablers to use at a later date coincides with the path enabler-change mechanism construct of the change option in Chapter 3 of this thesis. Change options are aspects of system designs that give stakeholders or users the "option" of executing some change in the system at a future date. Certain design requirements, however, can limit the addition of these path enablers. For example, high performance aircraft, that may put an emphasis on stealthy characteristics, are constrained in changes that can be made to the form of the aircraft. Stealth requires a carefully designed form that takes into account radar and thermal signatures. This form is sensitive to changes or modifications, and may not be simply changed without extensive re-engineering. The form constraints that the stealth requirement levies on the system can limit potential path enablers, like hard points for connections on the wings, or the addition of new sensors or antennas, ultimately impacting the changeability of the system.

The stealth requirement is an example of a system characteristic that, in conjunction with other extreme system requirements, drives high performance aircraft to be heavily optimized designs.

These optimized designs often require integrated systems to achieve the required performance levels. Having such integrated systems can limit the changeability available in a system, since any changes made to one aspect of the system necessitate changes in other aspects of design. Modularity is a design principle that can be used to combat the potential for change propagation that integrated systems suffer from (Giffin et al. 2007). Achieving ilities that relate to changing the system in response to changes in context or needs can be more difficult in designs that require high levels of optimization.

Another aspect of newer design techniques includes the high level of accuracy present in modern modeling methods. With advancements in computer-aided drafting (CAD), engineers are able to better model and optimize system components. While this seems like an improvement in the world of systems engineering, it may come with negative externalities in that margin in components or excess capacity in design is decreased with more efficient processes, when not intentionally introduced as path enablers. This saves money up front, often seen as a good thing from the system stakeholder perspective, but may cause problems when changes are made to the system in the future. For example, system changes may be needed in some future contexts, where some of the margin or excess capacity that normally would have existed due to imprecision in design, and would have enabled these changes, may no longer be there.

Overall, there are many other aspects that systems engineers have to deal with when thinking about implementing ilities in design. Social, political, fiscal, and other such pressures have large impacts on system design, and are difficult to capture in analyses. For example, certain exogenous factors may implemented in modeling in Epoch-Era Analysis (Ross and Rhodes 2008a) like exogenous political issues implemented in Multi-Attribute Tradespace Exploration (MATE), such as the transportation case study in Nickel (2010). Similarly, Weigel (2002) discusses the implementation of policy in the conceptual design phase. These aspects were mentioned often in interviews with practitioners as obstacles that they have to deal with on most of the systems they worked on. More research into how these non-traditional aspects of design or the negative externalities of "improvements" in design can be a useful area of research for systems engineering.

8 Discussions

This chapter addresses applicability of the research, discusses contributions of this research, addresses considerations for SoS, and discusses current and future ility research.

8.1 Applicability of research

This research is intended for applicability in engineering systems, for any system within any domain. While the research was grounded in traditional systems, considerations were made for systems of systems (SoS) and will be discussed later in this chapter. Throughout this research, the systems under consideration ranged from large technical systems, often military or government related, to smaller commercial and consumer systems or even buildings or structural systems.

The concepts of "real" and "perceived" spaces discussed in Chapter 3 are applicable to any system, and are not exclusive to a particular domain. A system as simple as a power screwdriver or as complicated as the space shuttle can be represented within those spaces. The design space is the traditional space where engineers, designers, and users can affect a system, impacting how it interacts with the context space and therefore its performance. That performance, combined with a set of stakeholder needs, relates to the perceived value of the system, regardless of the complexity of that system (the screwdriver or the space shuttle).

This research focuses on systems that change or resist change. However, throughout the research, context is given with regard to other research that could be accomplished without focusing only on change-related ilities. For example, architecture-related ilities are discussed in Chapter 5 as possible system properties that may have their own semantic basis. For this reason, the research mostly applies to the change-related ilities discussed in the development of the semantic basis.

The implications of the research are intended to apply to systems engineering as a whole, broad enough for wide application, but specific enough to be useful.

8.2 Discussion on Contributions

What follows is a discussion, organized by one of the five primary research contributions of this thesis: (1) value sustainment, (2) the epoch-shift—impact—response—outcome construct, (3) the prescriptive semantic basis for change-related ilities, (4) the historical system change database, and (5) interviews with systems engineering experts. The following sections outline discussions based on those contributions.

8.2.1 Value Sustainment

Chapter 3 briefly discusses the interactions between all the spaces (design, value, context, needs). The scope of this research did not include the complex interactions between each space, but some of these interactions are discussed. For example, instead of only changing the design space to affect the performance and therefore value space, system stakeholders can also attempt to affect the context and needs space. Political pressures or lobbyists may attempt to influence aspects in the context, such as laws or regulations. Advertising campaigns may attempt to influence the needs of consumers by convincing them they need a system to better their lives. Chapter 3 also briefly discusses how the context space and needs space have effects on the design space. For example, an innovation in the context space or a shift in a company wanting to become more environmentally friendly may relate to a shift in the design space. The innovation in the context space could open a completely new set of possible points, making the design space larger. The environmental push in a company may constrain the design space with fewer viable design points that satisfy the new environmental needs.

Chapter 3 also clarifies and characterizes evolvability, showing how it relates to changeability in an effort to enable value sustainment. The relationship between these and other ilities is connected to possible design principles. Design principles can be used to implement path enablers or inhibitors into the system, allowing for change or resistance mechanisms that relate to ilities. These change and resistance options are the basis for continued work into how systems can respond to dynamic environments.

8.2.2 Epoch-Shift—Impact—Response—Outcome

Chapter 4 explores specific cases that have gone through epoch shifts as perturbations, either successfully or unsuccessfully, when faced with the dynamic environments introduced in Chapter 3. The construct of the Epoch-Shift—Impact—Response—Outcome was proposed as a way to analyze the responses of four different cases. Iridium was a system that failed to initially respond to the changing needs of the world, and consequently went bankrupt, losing billions of dollars. Globalstar, a similar telecommunications satellite constellation system responded by lowering prices as much as possible and making a slightly less expensive system, but ultimately could not reach the appropriate price-point. In a world with growing terrestrial cellular, Globalstar went bankrupt as well. Both of these systems however were purchased and restructured through the benefits of chapter 11 bankruptcy and still live on today, producing value for their stakeholders. Success of these systems is doubled-sided then, as they were developed, operated and continue to operate, but they used methods that could be considered perverse to achieve ultimate success. Another satellite constellation system, Teledesic, properly adjusted to the changing needs in the world and scaled down in size and complexity, ultimately ceasing work all together, preventing billions in losses for what would have been a very expensive system. This system was more successful in responding to changing environment by minimizing downside losses, but was, in a sense, less successful than Iridium and Globalstar in that the system was never fielded. The comparison between these systems sheds interesting light on how systems can be designed and ultimately fielded via non-traditional methods. The final case looked into the Galileo deep space exploration satellite that experienced some very harsh mission perturbations, yet was able to overcome and produce value for the stakeholders, in some cases more than was expected. When trying to understand what qualities make a system successful, often the concept of ilities is referenced. The ilities, if they are desirable system properties that can lead to successful systems, should be better understood and be less ambiguous in order to enhance the ability to intentionally design for these ilities.

8.2.3 Prescriptive Semantic Basis for Change-related Ilities

This research proposed a ten dimensional prescriptive semantic basis in which to compare many different changes that systems can implement, and how those changes relate to the system properties, the "ilities." In the process of developing the basis, characteristics of ilities were identified, including the recognition that ilities fall in different semantic fields. Table 5-1 gives examples of two types (i.e. semantic fields) of ilities, which are "architecture-related ilities" like modularity, and "change-related ilities" like flexibility. The development of a semantic basis for architecture-type ilities is a potential useful area for future research. Additionally, a third type of ility that describes "new" abilities, like usability, may form another semantic field. Ilities of this latter type are not addressed in this research, but they could provide a fruitful path for future research, helping to classify unusual ilities like "drinkability[18]."

The semantic basis uses 10 dimensions of a change in order to assign ility labels in a consistent manner. Given properly encoded ility definitions in the basis, a given change statement formulated in the basis can automatically have ility labels assigned to it, based on the encoded definitions. In this way, comparisons between ilities may be made without concerns regarding consistent meanings in "use" terms, such as flexibility vs. changeability. In fact, ilities can be traded-off in three ways using the semantic basis: (1) definitions of ilities, (2) user-generated change statements, and (3) instances of ility implementations.

It is possible to define all change-related ilities using the ten dimensions of the semantic basis. Each ility definition is a composition of specific responses in each of the categories, with "any" signifying that the ility does not depend on that category. In this way, different definitions can be compared easily by looking at what categories are different. If a user knows which categories are important for that particular system, the user (perhaps a systems engineer) can focus on ilities that apply to those aspects. For example, if a user especially desires an automated or adaptable response from a system, the agent category needs to be labeled "internal."

The user-generated change statements can be used to trade-off ilities as well. The ten dimensions in the semantic basis are useful for characterizing change statements and therefore can be effective ways to compare and contrast requirements. In comparing differences between

---

[18] Drinkability was identified as a purported desirable beer quality in Bud Light™ advertisements seen on the Boston MBTA red line trains in March 2011.

choices within categories or threshold/parameter/destination reference values between requirements, the semantic basis makes differences easier to detect.

Finally, trade-offs between ilities can be accomplished through analysis of how systems could implement a given ility. For example, one could evaluate a system with ility A, with ility B, with ility A and B, and without ility A or B, and discover how ility A and B trade-off. The nature of this trade-off of this ility then is dependent on the baseline system under consideration. The semantic basis allows for clarity in how ilities are labeled in systems.

Discussion of how ilities may oppose each other lead to interesting insights back to path enablers from Chapter 3. Determining ilities that oppose, enhance, or are independent is largely based on how the ility is accomplished within a particular system. Using simply the definition of an ility, it is difficult to find ilities that naturally oppose or enhance one another. As discussed in Chapter 5, ilities may act as means to achieve other ilities, and this explains why many design principles proposed in this and other works are ilities themselves. This means-end relationship however changes depending on the order in which dimensions in the semantic basis are considered. When examining the path enablers that enable changes relating certain ilities, areas of opposition or enhancement may be found. If a path enabler can only be used once (consumable), or will end the ability to use another path enabler, then opposition of the path enablers may relate to opposition of ilities. A system may have limited resources for change, and have to choose between path enablers to execute in change mechanisms.

Overall, this semantic basis provides a means for consistently framing discussions of ilities in the future. The ambiguities present in the literature now make drafting requirements for ilities ambiguous, difficult to verify and difficult to validate.

An interesting aspect of the semantic basis research, and other related ility research discussed in Chapter 5, is the emergence of different "levels" of ilities. To know which ilities directly contribute to other ilities could be useful in system design, which drives the desire to create a hierarchy of ilities. Emergent in de Weck, Ross, and Rhodes (2012), where groups of graduate students made means-ends relationships of ilities, were "levels" that described the "depth" of ilities. These levels correlate to the dimensions in the semantic basis, and can be used to construct different ility hierarchies based on different orders of dimensions taken into consideration. The groups in de Weck, Ross and Rhodes (2012) independently came up with 4-6 levels of ilities, whereas the semantic basis research was able to identify 10. There seemed to be a natural order in that the "top" end ilities tended to be similar (e.g. value robustness and value survivability) and the "bottom" ilities tended to be similar (e.g. modularity and reconfigurability). There was little agreement on the "middle ilities," however. These ends of the spectrum tend to be correlated with change-related ilities (top) or architecture-related ilities (bottom) and show how some of the in-between ilities are more ambiguous in regard to which group they belong. An architecture-related ility should be completely based on the architecture, measureable or assessed without needing to operate the system. This may get confusing however, when "means" ilities can be confused with "ends" ilities. For example, scalability, according to Fricke and Schulz (2005), is an architecture-related ility, but according to this research is a change-related ility. The Fricke and Schulz viewpoint is about an architecture that is able to be scaled easily, and is therefore scalable. To achieve this however, the architecture may employ self-similar parts or modularity in design. This research argues that scalability, in this case, would be considered not an architecture-related ility, but rather the change-related ility, that is enabled by certain architecture-related ilities, such as modularity.

8.2.4 Historical System Change Database

The historical system change database was collected and used in this research to give insight into how the semantic basis could be applied to many different systems. The development of this database has directly aided the development of the semantic basis, spiraling between modifications to the database and improvements to the semantic basis. The database provides a set of change statements that vary in degree of complexity and domain for the semantic basis. Insights from the database using categorical cluster analysis (COOLCAT) were discussed in Chapter 6.

Comparing the COOLCAT clustering results to the heuristic-based results, research found that COOLCAT was able to find clusters of changes that were much more similar. This, of course, is what COOLCAT is intended to do, and the results validated that it performed as desired. This means that no matter which heuristics-based system characteristic was used to group a set of changes (e.g., lifecycle length, or domain) in the database, the COOLCAT algorithm found groups that were more similar. This may suggest that population-level research of large sets of changes could yield insights in the future that would never have been gained through looking at smaller sets of cases, providing strong motivation for a large scale follow-on research project.

Appendix B contains an example output report from the database. The table found there serves as a means for systems engineers to identify example system changes, arranged by specific ilities. One may easily search through the table and find different system changes and the associated ility labels. This table could be expanded, with additions of generic path enablers in future work, in order to allow system designers to have readily available design path enablers that they could use in their system.

The results from this exploratory research thrust can serve to motivate follow-on research in the semantic basis, and could possibly be taken even further, collecting many additional system changes in more domains and comparing them with additional cluster analysis.

8.2.5 Expert Interviews

Interviews with practicing systems engineers served to ground the theory-based research in this thesis to "state-of-the-practice." The experiences from these participants were captured with respect to some of the ideas proposed in this research, such as how ilities are implemented in systems and how ilities relate to one another. This research thrust was very interesting from a personal standpoint; it motivated efforts in this thesis as practicing systems engineers saw promise in some of the ideas put forth in better understanding how systems can be designed for change. More conclusions from the interviews discussed in Chapter 7 were desired, but due to time constraints of this research, the interviews remain more exploratory and motivational than in-depth empirical investigations of state-of-the-practice.

8.3 Design Principles

This research brought together descriptive and normative approaches for finding design principles for evolvability. Figure 8-1 below describes and summarizes the proposed design principles that relate to evolvability of engineering systems. Wasson (2006) defines a [design] principle as "a guiding [design] thought based on empirical deduction of observed behavior or practices that proves to be true under most conditions over time." Design principles were largely discussed as a means to achieve change and resistance options in systems through implementation of path enablers. While most of this research was descriptive, looking into historical system examples or evolvability and changeability literature, the normative research in Kelly (2010) yielded the last four design principles. By reading Kelly's work on technological evolution, design principles for re-architecting systems in an evolvable way are proposed.

Design Principle	Implications for Evolvability
Targeted Modularity	Limits change propagation (Hansen 2003) (Holtta-Otto 2005)
Integrability	Compatibility and common interfaces (Fricke and Schulz 2005)
Scalability	Of a parameter or entire system to meet new needs (Fricke and Schulz 2005)
Decentralization	Distributed resources to limit effect of changes (Fricke and Schulz 2005)
Redundancy	Gives flexibility to designer to eliminate components (Fricke and Schulz 2005)
Reconfigurability	Self similar parts and maximizing information reconfiguration (Siddiqi and de Weck 2008)
Leverage Ancestry	Successful design choices from all prior generations (Kelly 2010)
Mimicry	Successful design choices from other systems/domains (Kelly 2010; Henderson and Clark 1990)
Disruptive Architectural Overhaul	Upgrading large aspects of architecture at a time (Kelly 2010; Henderson and Clark 1990)
Resourceful Exaptation	Repurposing successful design choices from other systems (Kelly 2010)

Figure 8-1. Summary of Candidate Evolvability Design Principles

8.4 Systems of Systems

While the research in this thesis was gathered largely with traditional systems in mind, applicability of this research is desired for systems of systems (SoS) as well. As discussed in Chapter 2, SoS have special characteristics or properties that may relate to unique challenges for systems engineers.

SoS propose unique challenges in this change-related research because the degree of change is difficult to determine. SoS have constituent systems that are, to varying degrees, independent managerially. There may be constant changes being made to these systems at the constituent level. Determining whether these constituent-driven changes qualify as changes in the SoS may depend on the level of abstraction of the SoS under consideration. For example, if looking at the broad architecture of the SoS, concerned with how different constituents relate and interact with one another and what forms of communication and commerce are allowed between constituents, then small changes within constituents may be insignificant. However, if the change in a constituent has ripple effects that can be seen in other constituents, then this is probably more of a significant change.

Similarly, determining generations of SoS can be very difficult. The construct of evolvability in Chapter 3 should be able to be abstracted to apply to SoS as well. However, one of the leading premises of evolvability is the inter-lifecycle change. Since SoS are systems that, once formed, are constantly changing at some level, determining this idea of a generation is difficult. Perhaps generations of SoS are determined by when the SoS participates in a re-architecting exercise. The concept of re-architecting, however, mostly applies to directed or acknowledged SoS since re-architecting procedures are often driven by some sort of managerial component of the SoS. In certain directed SoS, when re-architecting is demanded on a 5 year basis for example, SoS generations are more clear, but for collaborative or virtual SoS, it may be harder to determine generations.

SoS may have characteristics that directly enhance or inhibit certain ilities as well. For example, SoS have characteristics that affect evolvability. SoS generally have distributed systems that are managerially independent, which relates to the evolvability design principle of decentralization. SoS are commonly forced to develop common interfaces for dealing with other constituent systems, which relates to the evolvability design principle of integrability. SoS are also, by design, somewhat modular in that the constituents are independent. If the constituents are independent and geographically separated, then they represent the evolvability design principle of targeted modularity. In this way, since they by their very nature implement some of the evolvability design principles, SoS may be naturally more evolvable than the average monolithic system.

8.5 Areas for Future Research

The research in this thesis represents a snapshot of the current continuous research in systems engineering for ilities. While this thesis focused on changeability and evolvability, general ility constructs were created to contextualize and compare ilities. As the set of ilities under consideration grows, research can hone in on broad ility constructs that may be used to describe this ever-growing aspect of modern systems. Some of the footnotes throughout this thesis refer to possible areas of future research, and are also described in this section.

The concept of perturbations discussed in this thesis detailing shifts and disturbances in the design space serves only as an introduction to the construct. There may be additional research needed to characterize what constitutes a shift versus a disturbance (see for example, Mekdeci et al. 2012). The term "disturbance" is more commonly used to align with Richards' (2009) construct of survivability. However, more research is needed to more fully classify perturbations to the system that effect performance for the remaining life span of the system. The scope of this thesis does not cover details for perturbations in the design space, rather, it proposes the relationship within the real and perceived spaces and the imposed change in instances within or to spaces.

In Chapter 3, there is discussion on the two types of effects that perturbations can have on instances within the space or to the space itself. Currently, Multi-Attribute Tradespace Exploration (MATE) and Epoch-Era Analysis (EEA) methods do not explicitly consider perturbations to "spaces" discussed in this section. Most simulations include changes only in the instances of designs, contexts, or needs, within their respective spaces. More research should be done to investigate the implications of space disturbances or shifts.

Additionally, in the development of the semantic basis in Chapter 5, some additional dimensions were considered. Two items from Chapter 3 that are not currently represented in the semantic basis are perturbations in "space" vs. perturbations in "instance," and differentiation between system parameter and outcome parameter. These dimensions were scoped out of this thesis, but represent areas for continued research.

The idea of an "imposition statement" was discussed in Chapter 3. If there is a change statement that applies to system changes, then there may be also a resistance statement or imposed change statement that correlates. For the purposes of this research, change statements may apply to imposed changes or voluntary changes. The development of "resistance statements" is an area for future research and could leverage the semantic basis in Chapter 5.

When looking at the relationship from design principles to ilities in Chapter 3, this thesis initially holds the strategy discussed, designing change options for bypassing resistance impositions and resistance options for inhibiting change impositions, as the standard relationship. However, it may be possible that change options can be used to combat both change and resistance impositions as well as resistance mechanisms being useful for combating change and resistance impositions. More research should investigate the possibility of the latter relationships.

In the semantic basis, change-related ilities have been distinguished from architecture-related ilities. As a result of this research, and the semantic basis in particular, the change-related ilities tend to better defined. The architecture-related ilities should be assessable in terms of observing the form and operations of the system (that is, the "architecture"). The architecture-related ilities become more confusing when one thinks about ends–means relationships. Ilities like scalability or interoperability can be confusing because they refer to aspects that unfold during operations of a system, but may be enabled by aspects of the architecture (e.g. modularity is an architectural property that can be measured, which may enable the change-related scalability). An area of future-research could include clarifying the differences between architecture and change-related ilities, as well as a possible new semantic basis for architecture-related ilities.

With regard to the change database and categorical cluster analysis, the population analysis was a product of using change data already collected. Therefore, this research represents more of an exploratory investigation into the analysis of population-level change characteristics. Future research should control for any sampling biases present in these collected system change examples. One could draft a formal questionnaire to gather system changes in a structured manner from more participants, and collect a larger set of change examples to help minimize any sample biases. Ideally, a crowd-sourcing method could be used to collect hundreds of system changes that could represent many different systems and changes. More detailed research could look into how different types of systems or different domains implement certain ilities. A product of such research could be a database of historical "ideas" or design principles that a system designer could reference when trying to find solutions to achieve certain ilities in a new system. This would enable the concept of exaptation and cross-domain inheritance from the evolvability section in this thesis.

9 Conclusions

"Adapt or perish, now as ever, is nature's inexorable imperative," - H.G. Wells (1866-1946)

Concluding this thesis with how it began, the quote from H.G. Wells reminds us that systems do not operate in fixed contexts or under fixed needs. Systems operate in dynamic environments and must be changed or built properly robust to handle this dynamic environment. In the ever-changing world of technology, policy, and people, systems must be designed to change with or withstand the times, or suffer the consequences.

Looking back at the B-52 vs. the B-58, the systems represent the struggle that all systems engineers face when developing systems. Competing aspects of design in performance and cost can drive systems into designs that are appropriate to change and live-on through many contexts producing value, like the B-52, or they can drive systems into optimized points that break at the slightest variation in context or needs, like the B-58. Is one system better than the other?—Value is in the eye of the beholder. Both systems were successful in their own right. The B-52 is a timeless design, and could possibly by a century aircraft. The B-58 represented huge advancements in technology and brought bombers into the supersonic realm of flight. An outcome of this research is that system stakeholders need to realize which type of system they are creating as well as which type of system they want, a B-52 or a B-58. If optimal very high performance is required, then maybe an optimal design like the B-58 is necessary. If longevity is required, then maybe a changeable design like the B-52 is necessary. Ideally, both goals can be achieved: longevity and performance. This research represents an attempt, like most system engineering research, in reducing the perceived tension between longevity and performance, achieving systems that can withstand the test of time, and perform to high-expected levels throughout.

9.1 Contributions of Research

Each of the main chapters in this thesis outlines one of the five primary research contributions of this thesis: (1) value sustainment, (2) the epoch-shift—impact—response—outcome construct, (3) the prescriptive semantic basis for change-related ilities, (4) the historical system change database, and (5) interviews with systems engineering experts. What follows is a summary of the main contributions of this research.

9.1.1 Value Sustainment

This research brings together different areas of research in robustness, survivability, changeability and evolvability to form the construct of *value sustainment*, the ability to maintain value delivery in spite of epoch shifts or disturbances. This high-level construct motivates much of the other research in this thesis and systems engineering as a whole. Systems can use different strategies to respond to dynamic environments—systems can be designed to be robust to these changes or they can be designed to be changed to meet new needs. Perturbations are characterized into two types, shifts and disturbances, differentiated by the relative timescale of the system they affect and how long they act on the system. While this thesis focuses on change-related strategies in systems, Chapter 3 contextualizes both strategies with each other, using the four dimensional, high-level value sustainment space in Figure 9-3.

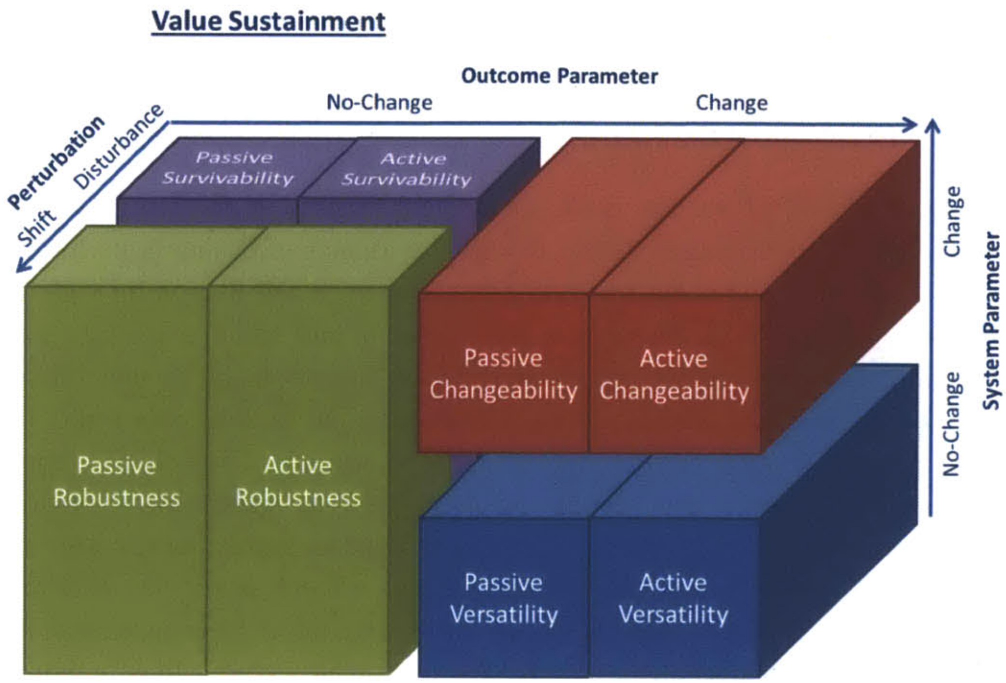


Figure 9-1. High-Level Value Sustainment Space

The way value from a system is perceived is clarified using the "real" and "perceived" spaces in Figure 9-2. This construct is particularly useful when understanding the nodes in the system that can have effects that propagate through, all the way to perceived value. This allows strategies to be formed to deal with these effects. The design space and context space interact to form the performance space. The needs space and performance space interact to form the value space. Therefore, system engineers can manipulate the design space, in response to changes in the context and needs spaces to improve value in the value space. Additionally, the design, context, and needs spaces may have changes imposed on them from perturbations, and those perturbations can affect the space in two ways, changing the entire space, or changing between points within the space.

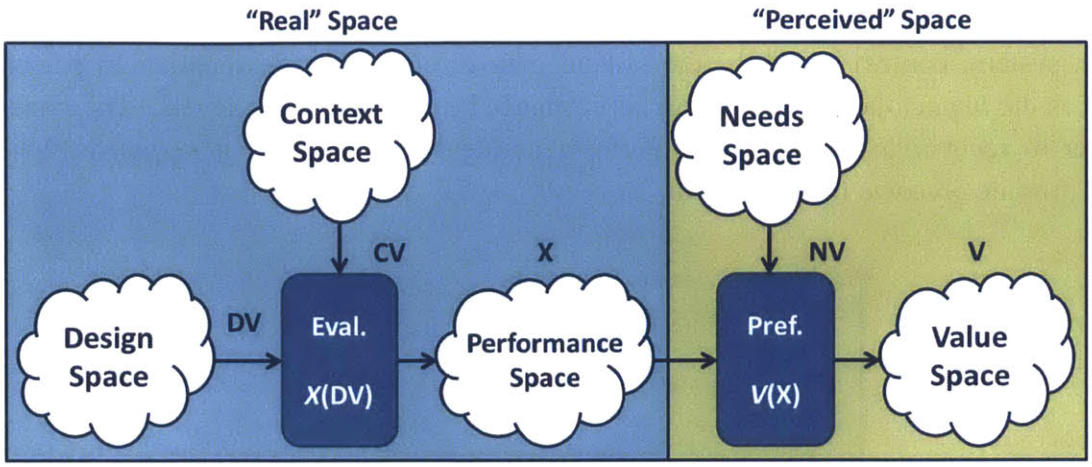


Figure 9-2. "Real" Space and "Perceived" Space

This research also clarifies and characterizes evolvability, and how it relates to changeability in an effort to enable value sustainment. Evolvability is a subset of changeability and is defined as *the ability of an architecture to be inherited and changed across generations [over time]*. The relationship between these and other ilities is connected to possible design principles.

Design principles can be used to implement path enablers or inhibitors in the system, allowing for change or resistance mechanisms that result in ilities. These change and resistance options are the basis for continued work into how systems can respond to dynamic environments. This flow from design principles to ilities is shown below in Figure 9-3.

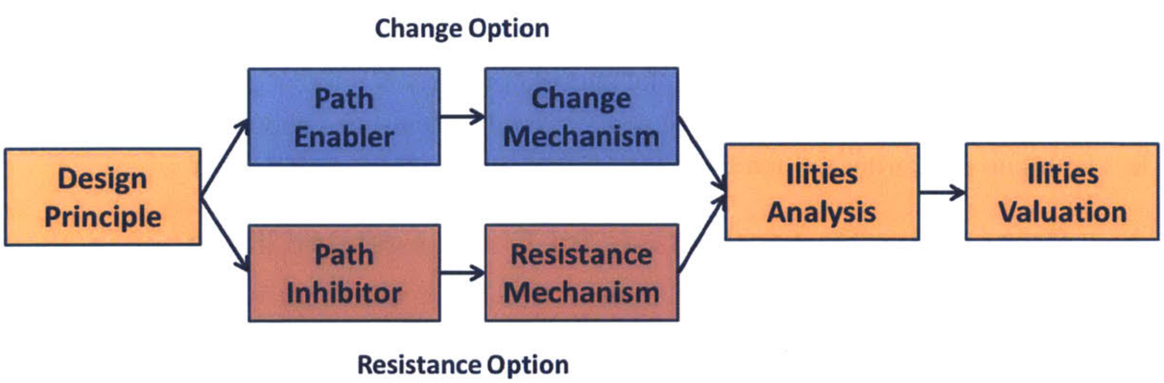


Figure 9-3. Relationship of Change and Resistance Options to Ilities

9.1.2 Epoch-Shift—Impact—Response—Outcome

The construct of the Epoch-Shift—Impact—Response—Outcome is proposed as a way to analyze system responses to dynamic environments. Using a construct similar to the Epoch-Era Analysis (Ross and Rhodes 2008a), system case examples are discussed using the construct as seen in Figure 9-4. The figure describes how a system may be operating at an acceptable level of performance in Epoch 1 and then experience an *epoch shift*. After experiencing this imposed shift in system, context or needs, the system may display some degradation in performance, known as the *impact*, possibly bringing performance below expectation levels. The system then, in order to recover to an acceptable performance level, may initiate a *response*, which then results in some *outcome* for the system.

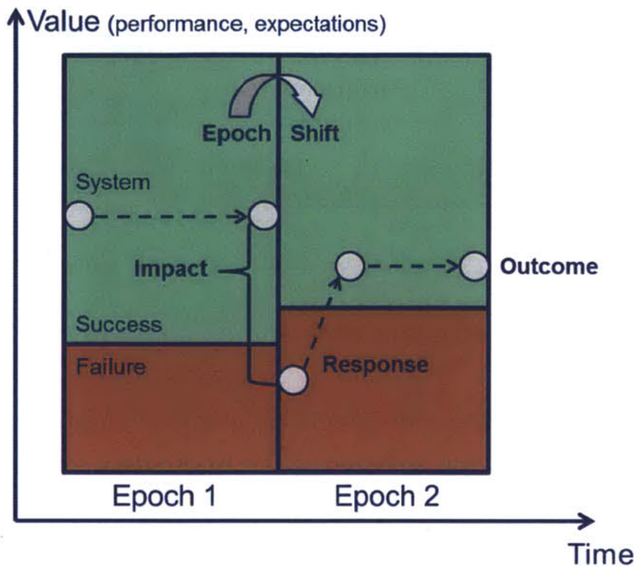


Figure 9-4. Epoch-Shift - Impact - Response - Outcome Construct

The comparison between systems in Chapter 4 sheds light on how systems can use non-traditional methods to be designed and ultimately fielded. When trying to understand what qualities make a system successful, often the concept of ilities is referenced. The ilities, then, are desirable system properties that often lead to successful systems, and should be better understood and less ambiguous in order to increase the ability to intentionally design them.

9.1.3 Prescriptive Semantic Basis for Change-related Ilities

This research proposes a ten dimensional basis in which to compare many different changes that systems can implement, and how those changes relate to desirable system properties, the "ilities." Figure 9-5 shows the semantic basis that can describe any change-related system ility based on descriptive change information, or aid in requirements drafting of desired ilities for future systems.

Prescriptive Semantic Basis for Change-type Ilities

In response to "cause" in "context", desire "agent" to make some "change" in "system" that is "valuable"

Cause	Context	Phase	Agent	Change	System	Valuable

In response to "perturbation" in "context", desire "agent" to make some "effect" to the "parameter" in the "aspect" of the "abstraction" during "phase" with "destination(s)" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits"

Perturbation	Context	Phase	Agent	Effect	Parameter	Destination	Aspect	Abstraction	Reaction	Span	Cost	Benefit
					"parameter"	"state"			"threshold"	"threshold"	"threshold"	"threshold"
disturbance	circumstantial	pre-ops	internal	increase	level	one	form	architecture	sooner	shorter	less	more
shift	general	ops	external	decrease	set	few	function	design	later	longer	more	less
none	any	inter-LC	either	not-same	any	many	operations	system	always	same	same	same
any		any	none	same		any	any	any	any	any	any	any
			any	any								

Figure 9-5. Prescriptive Sematic Basis for Change-related Ilities (Ross, Beesemyer, and Rhodes 2011)

The semantic basis provides a means for consistently framing discussions of ilities. The imprecision present in the current literature involving ilities make drafting requirements for ilities ambiguous, difficult to verify and difficult to validate. Examples are provided that show how the prescriptive semantic basis can be used to craft unambiguous and verifiable requirement statements for desired change-type ilities.

9.1.4 Historical System Change Database

In support of the research investigating connections between change mechanisms and ilities, a study was conducted to look at historical examples of system changes. The historical system change database was collected and used in this research to give insight into how the semantic basis could be applied to many different systems. A categorical cluster analysis algorithm (COOLCAT), as well as heuristic-based comparisons, was used to find trends in populations of system changes. The purpose of this research thrust was two-fold: (1) to serve as a preliminary means to validate the semantic basis discussed in Chapter 5, exposing weaknesses, holes, and ambiguities to help create a more complete construct, and (2) to shed light upon and further motivate research into how populations of systems may tend to display particular change strategies. The database provides a set of change statements that vary in degree of complexity and domain for the semantic basis. Insights from the database using categorical cluster analysis are discussed in Chapter 6.

9.1.5 Expert Interviews

Interviews with practicing systems engineers served to ground the theory-based research in this thesis to "state-of-the-practice." The experiences from these participants were captured with respect to some of the ideas proposed in this research, such as how ilities are implemented in systems and how ilities relate to one another.

Interview participants spoke about many of the difficulties in dealing with ilities that this research identified, confirming the challenges of ambiguity in language and difficulty in drafting requirements based on ilities. Additionally, ideas similar to the research-proposed constructs in this thesis were discussed, such as "hooks," which participants kept referring to as useful aspects in design that can be exploited later in the system lifespan. These hooks relate directly to the concept of "path enablers" discussed in Chapter 3. The implantation of path enablers or path inhibitors, as confirmed under the guise of "hooks," can be a useful way to achieve ilities in designs, and creates systems that are capable of responding well to dynamic environments.

9.2 Guiding questions

The systems we build today are operating with longer lifecycles in environments that are changing faster than in the past. This research focused on system changes to enable value sustainment, and aimed to understand how systems change from one state to another. This research attempted to analyze mechanisms that allow system changes to occur, and proposed a framework for allowing system designers to map vague, yet desirable, ilities to prescriptive system design principles. More specifically, this research characterized system changeability and evolvability, and ultimately provided a structured and meaningful way of classifying system characteristics, or "ilities". This research was guided by three key questions outlined below.

(1) What is "evolvability" in the context of engineering systems and systems of systems engineering (SoSE), and how does it relate to other ilities such as flexibility and adaptability?

(2) How can the properties and contexts of change mechanisms be used to classify a system's displayed ility types?

(3) How have ilities been implemented in historical systems/SoS, and how can we use this information to prescriptively include these desirable system properties in future designs?

The answers to these questions, developed during this research, now follow.

9.3 General conclusions

Using the research questions as guides, here are the general conclusions and contributions of this thesis.

9.3.1 What is "evolvability" in the context of engineering systems and systems of systems engineering (SoSE), and how does it relate to other ilities such as flexibility and adaptability?

Evolvability is a design characteristic that facilitates more manageable transitions between system generations via the modification of an inherited design and can be defined by *the ability of an architecture to be inherited and changed across generations [over time]*.

Evolvability is a subset of changeability that takes place at an architectural level, between generations, with inheritance from a prior system or system of systems. Flexibility and adaptability are also subsets of changeability. In a sense, evolvability is a sibling to these ilities, differentiated by the dimensions of analysis under consideration. Flexibility and adaptability are subsets of changeability based on the specification of the agent instigating the change, internal or external. Similarly, modifiability and scalability are subsets of changeability based on the type of parameter specified, whether it is a level or set type parameter. Evolvability then depends upon the abstraction undergoing the change, the architecture. Pliability is related to changeability and evolvability as change that takes place at the design level (Mekdeci et al. 2011). In this way, these ilities are related by making some change to the system (changeability), but differ in how these changes are implemented. The semantic basis has enabled this research to examine how evolvability relates to other ilities.

Systems of systems may inherently embody evolvable systems in that characteristics that typically define an SoS map to design principles for evolvability like decentralization, integrability, and targeted modularity. These principles make SoS naturally evolvable systems based on how we distinguish systems of systems from monolithic systems.

9.3.2 How can the properties and contexts of change mechanisms be used to classify a system's displayed ility types?

The discussion of change characteristics in Chapter 3 and the development of the prescriptive semantic basis in Chapter 5 combine to answer this research question. System changes can be described according to ten different dimensions that implement research from different areas to fully classify each change in a way that ility labels can be applied to the change. Chapter 5 discusses the ambiguity existing in the ility terms in use, and uses the semantic basis to create a formal structured way to address ilities in systems and therefore enable tradeoffs.

The contribution of the semantic basis will help system designers and stakeholders to have meaningful and effective conversations about what is required in the system, and how those requirements can be verified and validated after implementation.

9.3.3 How have ilities been implemented in historical systems/SoS, and how can we use this information to prescriptively include these desirable system properties in future designs?

After developing the prescriptive semantic basis for describing change-related ilities, this basis was put to use by collecting examples of changes in historical systems and systems of systems. The epoch-shift—impact—response—outcome construct used in Chapter 4 was used to evaluate four different historical space systems that have responded to dynamic environments in different ways leading to different outcomes. These examples show different strategies for achieving mission goals in different systems and shows how not properly responding to changing contexts or needs can have drastic effects on the system.

The change database of one hundred historical system changes was developed and used in cluster analysis to find trends in system changes based on the semantic basis categories. This exploratory research contributed to the development of the semantic basis as well, and shed light into how sub-groups of populations of systems may implement ilities differently depending on the characteristics of that sub-group. For example, systems can be differentiated based on population sizes from high population to low population. High population system changes were found to be more in the level type parameters, relating to scalability, whereas low population system changes were found to be more in the set type parameters, relating to modifiability. This suggests that the mass-produced systems may enable more mass differentiation through scalability rather than through modifiability.

To prescriptively include these desirable system qualities, Chapter 3 outlines the design principle to ility relationship. By using design principles, system designers may implement path enablers to allow change mechanisms in a system. These path enabler-change mechanism pairs give system stakeholders change options that can be used in a system. These change options can be classified with ility labels (using the semantic basis) and when implemented will give stakeholders desired ilities as seen in the relationship in Figure 9-3. Additionally, the idea of change options was extended to include resistance options, which similarly implement path inhibitors and resistance mechanisms. The change option and resistance option construct can be abstracted to involuntary change as well. Factors of uncertainty may lead to involuntary change or resistance impositions as shown below in Figure 9-6.

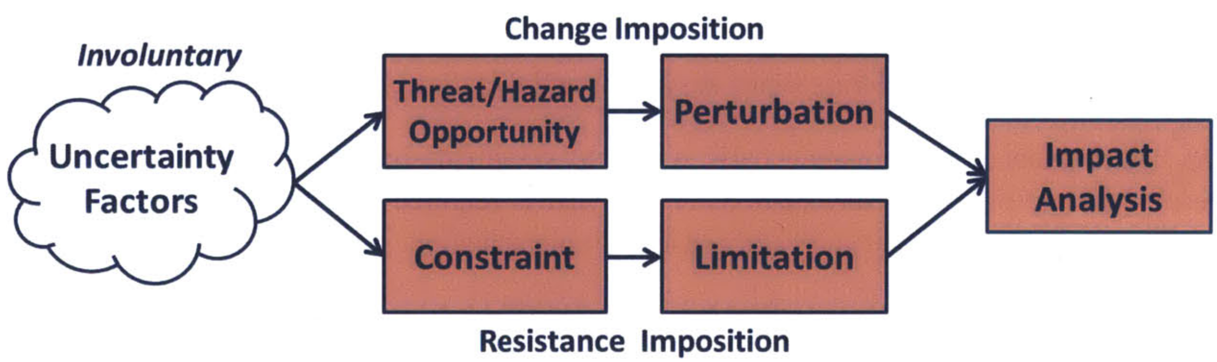


Figure 9-6. Involuntary Change and Resistance Impositions

Additionally, design principles for evolvability were proposed in Chapter 3 of this thesis. These are examples of design principles that system designers could use to generate path enablers or path inhibitors in a system. Table 9-1 summarized these proposed candidate evolvability design principles, similar to the design principles found in Figure 3-9 with Richards (2009) work on survivability design principles.

Table 9-1. Summary of Proposed Candidate Evolvability Design Principles

Design Principle	Implications for Evolvability
Targeted Modularity	Limits change propagation (Hansen 2003) (Holtta-Otto 2005)
Integrability	Compatibility and common interfaces (Fricke and Schulz 2005)
Scalability	Of a parameter or entire system to meet new needs (Fricke and Schulz 2005)
Decentralization	Distributed resources to limit effect of changes (Fricke and Schulz 2005)
Redundancy	Gives flexibility to designer to eliminate components (Fricke and Schulz 2005)
Reconfigurability	Self similar parts and maximizing information reconfiguration (Siddiqi and de Weck 2008)
Leverage Ancestry	Successful design choices from all prior generations (Kelly 2010)
Mimicry	Successful design choices from other systems/domains (Kelly 2010; Henderson and Clark 1990)
Disruptive Architectural Overhaul	Upgrading large aspects of architecture at a time (Kelly 2010; Henderson and Clark 1990)
Resourceful Exaptation	Repurposing successful design choices from other systems (Kelly 2010)

This thesis unites research in different areas to provide a more comprehensive construct for evaluating and understanding systems that change in order to maintain value delivery. The construct of value sustainment represents a high-level goal for systems engineering and motivates much of the research in this thesis. Systems and systems engineers can use the strategies and constructs proposed in this research to empower themselves with the ability to effectively respond to the inevitable and uncertain dynamic environments in which they must operate.

10 Appendix A: Ility Taxonomy Readme

The ility taxonomy worksheet begins with the user generating a change statement. This statement will become the basis for mapping a change to the type of ility or ilities it is characterized by. The structure of the change statement is generalized below:

In response to "perturbation" in “context”, desire “agent” to make some "effect" to the "system parameter" in the "aspect" of the "abstraction" during "phase" with "destination(s)" that are valuable with respect to thresholds in “reaction”, “span”, “cost” and "benefits".

The user fills in the blanks, signified by the “ “ marks in the general form of the statement. It is important that this statement is specific to a chosen parameter.

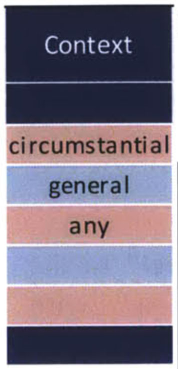
Working Through the Chart

For the purposes of working through the semantic ility basis, the following description will give examples that may apply to a passenger aircraft. These tables may be used to clarify the questions to be used when examining different ility statements.

1. Perturbation

This section refers to the cause or perturbation to the system. In a sense, it answers “why?” - or what the system is responding to.

Disturbance	The perturbation is some finite (short) duration of time, with a defined start and end time (e.g. a spike in temperature or a gust of wind) and relatively short in timescale compared to the system timescale
Shift	The perturbation is imposed with no end in the foreseeable future. The perturbation is constant and makes up part of the new environment in which the system must operate (e.g. a new federal regulation or environment of operation)
None	There is no perturbation. This response is for the system changes that do not occur in response to some perturbation
Any	A broader perturbation that may be either finite or a shift in environment. This response captures both of the types of perturbations)



2. **Context**

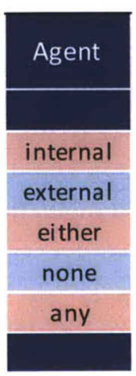
This section refers to the context on which the system responds to the perturbation

Circumstantial	The desired response is applicable only in certain conditions or a specific epoch (e.g. mitigating radio interference in critical phases of flight, such as take-off and landing)
General	The desired response is applicable to multiple conditions or many epochs (e.g. increase energy awareness desiring airliners to be more efficient)
Any	The context doesn't matter and could be either circumstantial or general

3. **Phase**

This section refers to the lifecycle phase in which the change occurs or "when?"

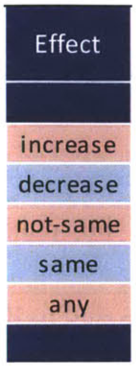
Pre-Ops	The change occurs in the design process before the system makes it to operational stages (e.g. a change in the concept or any change during initial design, testing, and integration of an aircraft)
Ops	The change occurs in the operational stages of the system (e.g. the redesign to the Airbus 380 engines)
Inter-LC	The change occurs between lifecycles (e.g. taking the design of an aircraft and altering it to create a new aircraft)
Any	The change could occur in any phase of the system lifecycle



4. **Agent**

This section refers to the change agent responsible for the change or "who?"

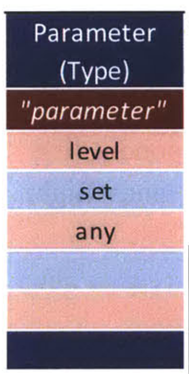
Internal	The change agent is internal to defined system boundaries (e.g. the autopilot recognizing a need to change course)
External	The change agent is external to defined system boundaries (e.g. an aircraft engineer upgrading a component such as landing gear)
Either	The change agent is either internal or external, the location of the change agent does not matter
None	There is no change agent involved, a passive response (e.g. the aircraft being robust to random gusts of wind in flight)
Any	The change agent could be external or internal to the system or none



5. **Effect**

This section refers to the effect or directionality of the change to the *parameter*

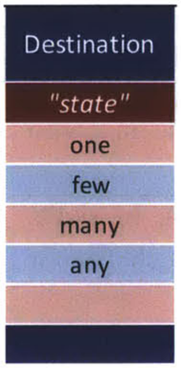
Increase	The change is an increase in parameter (e.g. an increase in cargo capacity or types of cargo that may be carried)
Decrease	The change is a decrease in parameter (e.g. a decrease in rivets needed or a decrease in amenities offered like meals, snacks, or movies)
Not-Same	The change in parameter just has to be different (e.g. a change in the exterior of the aircraft)
Same	There is no change in parameter (e.g. maintaining a level or set parameter, like lift, across a perturbation for robustness or survivability)
Any	The change in parameter doesn't matter and could be any of the above



6. **Parameter type**

This section refers to the type of parameter and includes a reference box for the specific parameter in question. This parameter is essential for the change statement to be clear and can be an attribute or a system parameter (lift, value, level of thrust, set of crew)

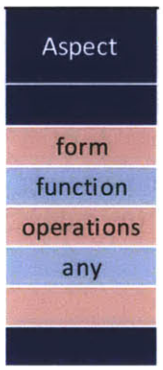
Level	The change is in the scale of the parameter (e.g. an increase in fuel capacity to the aircraft)
Set	The change is in the amount of parameters (e.g. changing the different types of runways the aircraft can land on)
Any	The change could be to either parameter change type



7. **Destination**

This section refers to the target range of potential states of the parameter as a result of the change and includes a reference box if a specific range or state is necessary (increase the level of thrust to "X pounds of force"

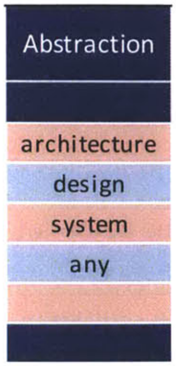
One	There is only one desired end state of the parameter due to the change (e.g. the landing gear being extended for landing)
Few	There are a few (countable number) different desired states due to the change (e.g. altering the in-flight media for varying length in travel times)
Many	There are many (uncountable) different end states due to the change (e.g. thrust level attainable through the lever (continuous))
Any	There is no specific number of target end states



8. **Aspect**

This section refers to the aspect of the abstraction being changed

Form	The change is in the physical components of the system (e.g. a change in the engine or wing design of an aircraft)
Function	The change is in the purpose of a system or system component (e.g. Using the in-flight media system to deliver safety and flight information)
Operations	The change is in how the system is used or implemented (e.g. a new set of flight rules to maximize fuel efficiency)
Any	The change could be to any of these aspects of the system



9. Abstraction

This section refers to the level of abstraction of the system being affected or what entity of the system

Architecture	The change is in the architecture of the system, where architecture is the higher level of mapping form to function of the system, affecting multiple possible designs (e.g. a change in the architecture of the Boeing 747)
Design	The change is in the design or the intended realization of a system (e.g. the design for the 747-8 intercontinental or other versions of the 747)
Realized System	The change is in the realized system. The particular system has been built, and the change is applied to this realized system (e.g. a specific aircraft with tail number ##)
Any	The change could be to any of these levels of abstraction of the system

** Another helpful way to think about levels of abstraction is the Apple iPod. The architecture would be the iPod in general, the design would be the more specific model such as the 16GB or 32GB iPod, and the system would be one specific instance of that design. This construct is degenerative for unique systems where the architecture, design, and instance may be represented by the same entity, like the International Space Station (ISS). More information on this may be found in Chapter 3 in the evolvability section.

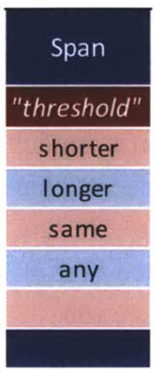
<u>The following columns refer to the value of the change described</u>

Note: the "valuable" section does not necessarily have to be specified, and multiple specifications of these columns could be deemed equally "valuable" (e.g. less benefit may be okay if it happens sooner, or slower may be okay if it is cheaper or has more benefit). These statements are in comparison to some "baseline" expectation on change "cost," often implicit to status quo change mechanisms (e.g. current product development schedules and budgets) or to a specified threshold amount.

10. Reaction

This section refers to the change start time or when the change initiates and has a threshold value that may be used as a reference (a specific target value, or reference to current value of the unchanged system)

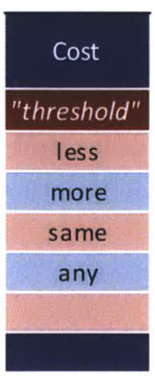
Sooner	The change initiates earlier than baseline/threshold (e.g. the autopilot recognizing a need to change course earlier than a previous model)
Later	The change initiates later than baseline/threshold (e.g. the autopilot recognizing a need to change course later than a previous model)
Always	The change or lack of change is always present and therefore does not initiate (e.g. the aircraft being robust to random gusts of wind in flight)
Any	It doesn't matter when the change initiates



11. Span

This section refers to the time it takes to change with respect to a baseline or comparable system or threshold value

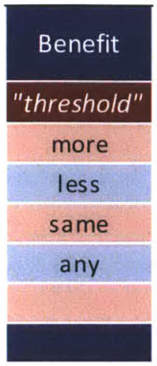
Shorter	The change duration is shorter than the baseline/threshold (e.g. the autopilot changes course in less time than a previous model)
Longer	The change duration is longer than the baseline/threshold (e.g. the landing gear takes longer to switch positions than a previous model)
Same	There is no change in duration with respect to the baseline/threshold
Any	The change could be any duration



12. Cost

This section refers to the resources required for the change with respect to a baseline or comparable system or threshold value

Less	The change costs less than it would in the baseline/threshold (e.g. the change in course or altitude uses less fuel than the previous model)
More	The change costs more than it would in the baseline/threshold (e.g. re-painting a bigger plane will cost more than a smaller plane)
Same	There change costs the same as it would in the baseline/threshold
Any	The change agent could be external or internal to the system



13. Benefit

This section refers to the utility as a result of the change with respect to the baseline system or threshold value

More	The change results in an more utility than the baseline/threshold (e.g. better brakes allowing shorter stopping times)
Less	The change results in an less utility than the baseline/threshold (e.g. a wing design the creates less lift than the baseline)
Same	The change results in the same utility as the baseline/threshold
Any	The change agent could be external or internal to the system

11 Appendix B: Database Output Examples

This appendix uses output data from the database to give examples of different ility types. The table may be used in a way to quickly find examples of certain ility labels across a wide variety of systems. One can scroll down the ility columns (Changeability, Evolvability, Survivability, or Robustness) and find different system change examples displaying that ility (based on the ility definitions used in this thesis). Also included, is the change statement, which is more of a change description as these are descriptive changes that have taken place. Also, there is a column of mapped ility labels that captures all of the ilities (in addition to and including those explicitly captured in the four ility label columns).

Change ID #	Changeability	Evolvability	Survivability	Robustness	System	Change Statement	Mapped Ility Labels
1	X				HMMWV	In response to improved IEDs and ambush tactics in urban warfare, desire AM General to design some increase in protection/armor in the HMMWV for quick valuable long term deployment.	Changeability, Flexibility, Scalability, Value Robustness,
2	X				B-52	In response to increased demand in alternative fuel sources in a context of climbing dependence on foreign fossil fuels, desire the Air Force to change fuel mixtures used in B-52s engines (synthetic fuels).	Changeability, Flexibility, Reconfigurability, Modifiability, Extensibility,
3	X		X		F-16	In response to need for longer range, desire ground crew to add external tanks for fuel to aircraft at hard points to increase fuel storage for increased range.	Changeability, Flexibility, Survivability, Scalability, Reconfigurability,
4	X				F-18 (Swiss)	In response to higher load cycles in a new operating environment (country), desire Swiss engineers to make some improvement to strength by replacing aluminum ribs with titanium ribs, that is always available and valuable	Changeability, Flexibility, Scalability, Reconfigurability, Value Robustness,
5	X	X			S-92	In response to changing market needs in the civilian helicopter sector and FAA regulations, desire Sikorsky to evolve the S-70 into a new helicopter suitable for military and civilian purposes. Developed from the S-70 or Black Hawk family, the S-92 was planned to utilize as many components and subsystems from the highly reliable Black Hawk. The S-92 ended up with a redesigned a new dynamic component system, rotor, and gearbox.	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness,

Change ID #	Changeability	Evolvability	Survivability	Robustness	System	Change Statement	Mapped Ility Labels
6	X				HMMWV	In response to "improved IEDs and ambush tactics" in "urban warfare", desire "Soldiers in the field" to "make a change to armor" in the "HMMWV" for immediate valuable deployment. This change refers to "Hill Billy" armor additions made by soldiers in the field to the Humvees to help protect against small arms fire.	Changeability, Flexibility, Scalability, Reconfigurability, Value Robustness,
7	X				HMMWV	In response to "increased weight of the system" in "the new Humvees that require more armor for protection", desire "AM General" to "design stronger chassis and better suspension" in the "HMMWV" for immediate valuable deployment.	Changeability, Flexibility, Scalability, Value Robustness,
8			X		A-10	In response to direct hits from armor-piercing and high explosive projectiles up to 23 mm to the cockpit during close air support, desire the A-10 to withstand attack and protect pilot, always.	Survivability,
9			X		A-10	In response to direct hits small arms fire to fuel tanks during close air support, desire the A-10 to seal leaks in fuel system to minimize fires, explosions, and fuel supply depletion.	Survivability,
10	X				F-14	In response to changing aerodynamic environments, desire central air data computer to adjust wing sweep angle to improve lift to drag ratio and change aerodynamic characteristics of the aircraft.	Changeability, Adaptability, Scalability, Reconfigurability, Value Robustness,
11	X	X			Evolved Expendable Launch Vehicle (EELV)	In response to DoD desire to improve space lift capabilities in a context of high space access costs desire Boeing to design the Delta IV to be more affordable and standardized. The cryogenic second stage is an evolutionary design incorporating the Redundant Inertial Flight Control Assembly (RIFCA) from Delta II and the Pratt & Whitney RL10B-2 engine. The Delta IV Medium & Medium-Plus (4,2) vehicles use the same 4-meter diameter second stage, while the Delta IV Medium-Plus (5,2), Medium-Plus (5,4) and Heavy vehicles use the same RL10B-2 engine, but have larger 5-meter diameter fuel tanks and stretched oxidizer tanks.	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness, Agility,
12	X	X			M1126 Stryker	In response to improved improvised explosive devices in urban warfare, desire GD to implement V-shape hull in new Stryker design.	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness,

Change ID #	Changeability	Evolvability	Survivability	Robustness	System	Change Statement	Mapped Ility Labels
13	X				iPhone 4	In response to user preference change in mobile phone operations desire user to change the layout of the applications/buttons of the iPhone to improve usability. The buttons on the iPhone may be easily reconfigured into groups or new pages to increase the users usability and functionality during operations of the phone.	Changeability, Flexibility, Reconfigurability, Optionability, Modifiability, Value Robustness, Agility,
14	X				F-14	In response to smaller storage areas on aircraft carriers, desire user to adjust wing sweep angle to decrease storage area requirement.	Changeability, Flexibility, Scalability, Reconfigurability,
15	X	X			Evolved Expendable Launch Vehicle (EELV)	In response to DoD desire to improve space lift capabilities in a context of high space access costs desire Boeing to design the Delta IV to be more affordable and standardized. Commonality between all of the systems is central to the Delta IV. Each Medium & Medium-Plus vehicle uses a single common booster core (CBC), while the Heavy uses three CBCs. The Pratt and Boeing Rocketdyne-built RS-68, a liquid hydrogen/liquid oxygen engine that produces 663,000 lbs of liftoff thrust, powers the first stage. This engine is mounted to the CBC first-stage structure and was designed for ease of manufacture by significantly reducing part count and thereby increasing reliability.	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness, Agility,
16	X	X			Evolved Expendable Launch Vehicle (EELV)	In response to DoD desire to improve space lift capabilities in a context of high space access costs desire Lockheed Martin to design the Atlas V to be more affordable and standardized. To increase Atlas V performance and reliability, the Common Core Booster (CCB) is 100 % common across all Atlas V types, with over 5,200 parts and 300 suppliers eliminated compared to Atlas IIAS (35% reduction in parts).	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness, Agility,
17	X				Evolved Expendable Launch Vehicle (EELV)	In response to DoD desire to improve space lift capabilities in a context of high space access costs desire Lockheed Martin to design the Atlas V to be more affordable and standardized. To scale Atlas V performance, the various configurations of the Common Core Booster (CCB) (1 or 3), the Aerojet strap-on solid rocket boosters (1-5), and the Pratt and Whitney RL10A-4-2 of the Centaur upper stage (1-2) may be changed to accommodate different payload mass to orbit requirements.	Changeability, Flexibility, Modifiability, Value Robustness, Agility,

Change ID #	Changeability	Evolvability	Survivability	Robustness	System	Change Statement	Mapped Ility Labels
18	X				Evolved Expendable Launch Vehicle (EELV)	In response to DoD desire to improve space lift capabilities in a context of high space access costs desire Boeing to design the Delta IV to be more affordable and standardized. To scale Delta IV performance, the various configurations of the Common Booster Core(CBC) (1 or 3) and the Alliant Techsystems strap-on solid rocket boosters (2 or 4) may be changed to accommodate different payload mass to orbit requirements.	Changeability, Flexibility, Modifiability, Value Robustness, Agility,
19				X	F-14	In response to changing aerodynamic environments, desire central air data computer to adjust wing sweep angle to improve lift to drag ratio and change aerodynamic characteristics of the aircraft.	Robustness, Value Robustness,
20	X	X			iPhone 4	In response to user higher demand in signal strength while not sacrificing size or weight in mobile phones, desire Apple to improve antenna for iPhone next generation. The iPhone 4 implemented a new antenna design using the frame of the phone as an external antenna. While plagued with problems due to antenna shorting, the design flaw could be fixed with a rubber stopper.	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness,
21	X				iPhone 4	In response to user preference change in mobile phone capabilities desire user to change/add an application to the iPhone to improve functionality. The App store allows users to choose from thousands of apps that add functionality and the ability to personalize the iPhone to be used in many different situations	Changeability, Flexibility, Functional Versatility, Modifiability, Extensibility, Value Robustness, Agility,
22	X				iPhone 4	In response to low light conditions desire iPhone to turn on flash to improve picture quality. The iPhone 4 uses an LED flash with its camera to add light in low-light conditions to improve picture quality automatically.	Changeability, Adaptability, Scalability, Reconfigurability, Value Robustness,
23	X	X			M1126 Stryker	In response to DoD requirement to operate from C-130s, desire GD to design a light armored vehicle that can operate from a C-130 by size and weight requirements.	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness,
24	X				F-35	In response to desire to land/take-off in a shorter than normal distance for fixed wing jet aircraft, desire F-35 to change configuration and thrust vertically.	Changeability, Flexibility, Reconfigurability, Modifiability, Value Robustness,

Change ID #	Changeability	Evolvability	Survivability	Robustness	System	Change Statement	Mapped Ility Labels
25	X				F-35	In response to desire to land/take-off in a shorter than normal distance for fixed wing jet aircraft, desire F-35 to change configuration and thrust vertically.	Changeability, Flexibility, Reconfigurability, Modifiability, Value Robustness,
26	X				B-52	In response to shifting threats in a an increasing lifetime for an old bomber, desire engineers to upgrade the B-52 with enhanced electronic countermeasures to thwart enemy jamming or missile threats.	Changeability, Flexibility, Reconfigurability, Modifiability, Extensibility, Value Robustness,
27	X				Swiss Army Knife	In response to user preference change, desire the user to change configuration of knife to increase usability.	Changeability, Flexibility, Reconfigurability, Optionability, Modifiability, Value Robustness,
28	X				Swiss Army Knife	In response to user preference change, desire the user to change use of knife component to increase usability.	Changeability, Flexibility, Functional Versatility, Modifiability, Value Robustness,
29	X				V-22	In response to desire/ requirement to land vertically desire pilot to change orientation of propeller on aircraft to redirect thrust.	Changeability, Adaptability, Scalability, Reconfigurability, Value Robustness,
30				X	Climate Manager	In response to change in water levels desire system to automatically engage irrigation systems to keep moisture within a certain range.	Robustness, Value Robustness,
31				X	Climate Manager	In response to change in temperature and humidity pushes desire system to automatically engage louvers, fans, vents, and shades to control temperature within a certain range.	Robustness, Value Robustness,
32				X	Climate Manager	In response to change in Co2 levels desire system to automatically engage Co2 valves or burner systems to keep levels within a certain range.	Robustness, Value Robustness,
33				X	Solar Tracking System	In response to changing solar locations, desire system to orient solar cell arrays towards sun to maximize energy output. Improves output by up to 40%	Robustness, Value Robustness,
34				X	A-10	In response to engine failure during close air support, desire the A-10 to continue to fly and land safely.	Robustness, Classical Robustness, Value Robustness,
35	X				HMMWV	In response to new mission requirements for HMMWV, desire "AM General" to implement new weapon capabilities or compatibility in HMMWV variants for immediate deployment.	Changeability, Flexibility, Reconfigurability, Optionability, Modifiability, Value Robustness,

Change ID #	Changeability	Evolvability	Survivability	Robustness	System	Change Statement	Mapped Ility Labels
36	X		X		8R Series Tractor	In response to changing loads and speeds during turns or other farming functions, desire tractor to automatically downshift or change transmission setting to allow user more free time for other tasks.	Changeability, Adaptability, Survivability, Scalability,
37	X				8R Series Tractor	In response to shift in available farming techniques in day-to-day farming desire tractor to automatically update GPS guiding software to increase automated functionality. System updates could include new turning capability or more efficient row-crop paths or new paths that minimize nitrogen or crop damage.	Changeability, Adaptability, Operational Versatility, Modifiability, Extensibility, Value Robustness,
38	X		X		8R Series Tractor	In response to change in task during daily farming activities desire user to change tractor attachment to make use of new function for more efficient farming.	Changeability, Flexibility, Survivability, Reconfigurability, Optionability, Modifiability,
39	X		X		8R Series Tractor	In response to task shift to lawn mowing desire operator to attach mowing accessory to tractor to allow for new function. The mowing accessory can easily be attached by simply driving over it, automatically engaging and ready to use.	Changeability, Flexibility, Survivability, Reconfigurability, Modifiability, Agility,
40	X		X		Total Gym	In response to change in desired exercise during a workout, desire user to change method of operation to change to many different exercises and focus a new muscle group.	Changeability, Flexibility, Survivability, Operational Versatility, Modifiability, Agility,
41	X		X		Total Gym	In response to change in desired difficulty level during a workout, desire user to adjust exercise load by increasing inclination level for more beneficial workouts.	Changeability, Flexibility, Survivability, Scalability, Reconfigurability,
42	X		X		Total Gym	In response to change in desired storage size in the home, desire the user to minimize system volume for storage easily and fast. System folds up and disconnects using pin-pulls and hinges.	Changeability, Flexibility, Survivability, Scalability, Reconfigurability,
43	X		X		Folding Bike	In response to change in desired storage level and security desire user to fold bike into size suitable for carrying into workplace for compact storage.	Changeability, Flexibility, Survivability, Scalability, Reconfigurability,
44	X		X		BMW ActiveHybrid 7	In response to parking the vehicle on the street desire vehicle to automatically fold in side-view mirrors for safety and minimized damage risk from traffic.	Changeability, Adaptability, Survivability, Reconfigurability, Modifiability,

Change ID #	Changeability	Evolvability	Survivability	Robustness	System	Change Statement	Mapped Ility Labels
45	X		X		BMW ActiveHybrid 7	In response to user desire to parallel park desire system to automatically find acceptable parking spot size and automatically park. http://www.youtube.com/watch?v=9L_LHgVhxVw	Changeability, Flexibility, Survivability, Modifiability,
46	X		X		BMW ActiveHybrid 7	In response to coasting down a gradient desire system to turn electric motors into generators and recharge battery, capturing energy.	Changeability, Adaptability, Survivability, Modifiability,
47	X		X		BMW ActiveHybrid 7	In response to stop-and-go traffic while driving, desire car to automatically shutoff engine and run on electric motors to save fuel.	Changeability, Adaptability, Survivability, Modifiability,
48	X		X		BMW ActiveHybrid 7	In response to rainy weather while driving desire car to automatically adjust windshield wipers to improve vision.	Changeability, Adaptability, Survivability, Modifiability,
49	X		X		BMW ActiveHybrid 7	In response to change in road elevation, turning corners, and inclination change desire car to automatically adjust level of headlamps to improve vision in night time driving conditions.	Changeability, Adaptability, Survivability, Scalability, Reconfigurability,
50	X		X		Drawbridge	In response to danger or threat around the fortress desire user to raise drawbridge by pulling the lift chains back, creating a security gap in front of the entrance.	Changeability, Flexibility, Survivability, Reconfigurability, Modifiability,
51	X		X		Bascule Bridge	In response to ships needing to pass under bridge where there is not enough clearance, desire bridge operator to raise bridge leaf or leaves to increase clearance.	Changeability, Flexibility, Survivability, Reconfigurability, Modifiability,
52	X		X		Bascule Bridge	In response to ships needing to pass under bridge where there is not enough clearance, desire bridge operator to raise bridge leaf or leaves to increase clearance.	Changeability, Flexibility, Survivability, Scalability, Reconfigurability,

Change ID #	Changeability	Evolvability	Survivability	Robustness	System	Change Statement	Mapped Ility Labels
53	X	X			PowerTech PSX Interim Tier 4/Stage IIIB	In response to "more stringent emissions regulations" by the EPA in "off-road vehicles", desire "John Deere" to implement "cooled exhaust gas recirculation (EGR)" in new "PowerTech PSX Engines" in order to minimize Nitrogen Oxides (NOx) that is "within code". The move to Interim Tier 4/Stage III B emissions regulations is the most significant to date. The regulations call for a 90 percent reduction in particulate matter (PM) along with a 50 percent drop in nitrogen oxides (NOx). Final Tier 4/Stage IV emissions regulations, which will be fully implemented by 2015, will take PM and NOx emissions to near-zero levels.	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness,
54	X	X			PowerTech PSX Interim Tier 4/Stage IIIB	In response to "more stringent emissions regulations" by the EPA in "all off-road vehicles", desire "John Deere" to add "catalyzed exhaust filter" as a new Particulate Matter (PM) reduction method in the "PowerTech PSX Engine" "architecture" between "lifecycles" in order to minimize (PM) that is "within code." The move to Interim Tier 4/Stage III B emissions regulations is the most significant to date. The regulations call for a 90 percent reduction in particulate matter (PM) along with a 50 percent drop in nitrogen oxides (NOx). Final Tier 4/Stage IV emissions regulations, which will be fully implemented by 2015, will take PM and NOx emissions to near-zero levels.	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness,
55	X				iPhone 4	In response to user preference change in mobile phone capabilities desire user to change/add an application to the iPhone to improve functionality. The App store allows users to choose from thousands of apps that add functionality and the ability to personalize the iPhone to be used in many different situations	Changeability, Flexibility, Operational Versatility, Modifiability, Extensibility, Value Robustness, Agility,
56	X				Tufts Dental Center	In response to need for more real estate, desire Tufts design team to add 5 floors vertically to existing building in order to increase space without adding an entire new building.	Changeability, Flexibility, Scalability, Reconfigurability, Value Robustness,
57	X				Blue Cross Blue Shield Tower	In response to need of more real estate, desire HCSC to add floors to existing tower to increase area while keeping same location without expanding outwards.	Changeability, Flexibility, Scalability, Reconfigurability, Value Robustness,
58	X		X		Lexus SC	In response to change in user preference desire car to fold hard top automatically into trunk and convert to topless car for more comfort/experience.	Changeability, Flexibility, Survivability, Scalability, Reconfigurability, Agility,
59	X		X		Personal Parachute	In response to need to slow down during skydiving desire system to open parachute during free fall for life sustainment	Changeability, Flexibility, Survivability, Scalability, Reconfigurability,

Change ID #	Changeability	Evolvability	Survivability	Robustness	System	Change Statement	Mapped Ility Labels
60	X		X		AAD	In response to passing through a specified altitude at a velocity deemed unsafe during free-fall, desire AED to automatically pull chute to slow user's velocity and save life.	Changeability, Adaptability, Survivability, Scalability, Reconfigurability,
61				X	HVAC	In response to change in temperature desire HVAC system to automatically add or take away heat to maintain desirable indoor temperature range for increased comfort.	Robustness, Value Robustness,
62	X				Transition Lens	In response to change in UV environment desire eyeglasses lens to automatically adjust darkness to compensate for lighting conditions and improve vision and eye safety.	Changeability, Adaptability, Scalability, Reconfigurability, Value Robustness,
63	X				Umbrella	In response to change in user preferences in sunlight exposure desire user to open/close umbrella to protect from light or store in smaller volume.	Changeability, Flexibility, Scalability, Reconfigurability, Value Robustness,
64	X				Umbrella	In response to change in weather conditions desire user to open/close umbrella to keep user dry from rain.	Changeability, Flexibility, Scalability, Reconfigurability, Value Robustness,
65	X				Drop-Leaf Table	In response to change in user need for table area, desire user to lift hinged drop-leaf to add/subtract area to table surface for storage or use.	Changeability, Flexibility, Scalability, Reconfigurability, Value Robustness,
66	X				AraMiS	In response to change in mission desire system design to change components of modular Aramis satellite design to account for different requirement (power, comm, ADCS, etc).	Changeability, Flexibility, Modifiability, Value Robustness,
67	X				AraMiS	In response to change in mission requirement desire system design to add or subtract components of modular Aramis satellite design to increase or decrease overall size of spacecraft needed for new mission requirements.	Changeability, Flexibility, Scalability, Value Robustness,
68	X				International Space Station	In response to deteriorating orbit altitude desire space shuttle to use extra fuel when docked to boost space station altitude to increase life-span of orbit.	Changeability, Flexibility, Scalability, Value Robustness,
69	X				International Space Station	In response to newly developed modules desire new module to be added to the ISS to increase capabilities of the station.	Changeability, Flexibility, Reconfigurability, Optionability, Modifiability, Extensibility, Value Robustness,
70	X		X		International Space Station	In response to possible collision prediction desire ISS to fire boosters move space station altitude/orbit to avoid possible collision and increase safety.	Changeability, Flexibility, Survivability, Scalability,
71	X	X			iPod	In response to demand in different file types for music and photos, desire apple to make iPhone compatible with new types of music files in new design.	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness,

Change ID #	Changeability	Evolvability	Survivability	Robustness	System	Change Statement	Mapped Ility Labels
72	X	X			iPod	In response to user demand shift in capacity design higher storage devices between lifecycles to increase disk/memory space holding more music and photos	Changeability, Flexibility, Scalability, Evolvability, Value Robustness,
73	X				iPod	In response to user demand shift in number of songs desire user to add songs to library to increase available music.	Changeability, Flexibility, Scalability, Reconfigurability, Value Robustness,
74	X				Adjustable Shower Head	In response to change in user preference for amount of water-flow desire user to easily adjust flow of water by lever for increased comfort.	Changeability, Flexibility, Scalability, Reconfigurability, Value Robustness,
75	X				Freight Train	In response to shift in power requirement for load of train desire train company add or subtract locomotives to add or subtract power as needed.	Changeability, Flexibility, Scalability, Reconfigurability, Value Robustness,
76	X				Freight Train	In response shift in load volume desire train company to add or subtract train cars to increase or decrease train volume as needed.	Changeability, Flexibility, Scalability, Reconfigurability, Value Robustness,
77	X				Virginia Class Submarine	In response to change in user depth requirement desire submarine to fill or empty ballast tanks to change depth to desired level.	Changeability, Flexibility, Scalability, Reconfigurability, Value Robustness,
78	X		X		Virginia Class Submarine	In response to change in user security requirement due to approaching enemy desire submarine to operate at various depths to avoid detection.	Changeability, Flexibility, Survivability, Scalability,
79	X	X			Virginia Class Submarine	In response to shift in congress' desire to save money per unit ordered while increasing functionality over time, desire block II designs to be built in 4 sections from the 10 sections in block I to reduce cost 300 million between lifecycles	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness, Agility,
80	X				Malibu Ski Boat	In response to user preference shift in wake size desire user to change ballast level to increase displacement and wake size.	Changeability, Flexibility, Scalability, Reconfigurability, Value Robustness,
81	X				Malibu Ski Boat	In response to user preference shift in wake size desire user to change trim level to increase displacement and wake size.	Changeability, Flexibility, Scalability, Reconfigurability, Value Robustness,
82	X				Futon	In response to user preference change desire user to change configuration to bed or couch to meet new needs.	Changeability, Flexibility, Scalability, Reconfigurability, Value Robustness,
83	X				Futon	In response to user preference change desire user to change mattress/couch cover for new look or quality replacement.	Changeability, Flexibility, Reconfigurability, Optionability, Modifiability, Value Robustness,

Change ID #	Changeability	Evolvability	Survivability	Robustness	System	Change Statement	Mapped Ility Labels
84	X				Track Shoes	In response to shift in running requirements in different running conditions desire user to switch out spikes for applicable length and material for better performance.	Changeability, Flexibility, Reconfigurability, Optionability, Modifiability, Value Robustness,
85	X				Multi-Color Pen	In response to change in user preference for ink color during writing, desire user to select new color without changing pens.	Changeability, Flexibility, Reconfigurability, Modifiability, Value Robustness,
86	X				Golf Driver	In response to change in user preference desire user to adjust system properties (loft, face angle, weight) by adjusting components.	Changeability, Flexibility, Scalability, Reconfigurability, Value Robustness, Agility,
87	X				Proteus	In response to new mission desire user to exchange modules for new purposes such as search and rescue, research, storage, or maintenance.	Changeability, Flexibility, Reconfigurability, Optionability, Modifiability, Value Robustness,
88	X	X			Air and Space Operations Center	In response to increased complexity and workload, leading to poor performance, desire USAF to change the AOC to be treated as a weapon system to increase performance. By being treated as a weapon system, the AOC had to be designed and standardized in operations and set-up.	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness,
89	X				Air and Space Operations Center	In response to increase in ISR sensors desire a change to the AOC to integrate new sensors (global hawks, predators, JSTARS, U-2, RC-135) to build sensor array or add new functions/features.	Changeability, Flexibility, Scalability, Reconfigurability, Value Robustness,
90	X				Air and Space Operations Center	In response to increase in ISR sensors desire a change to the AOC to integrate new display for real-time data on integrated displays to improve situational awareness.	Changeability, Flexibility, Reconfigurability, Optionability, Modifiability, Value Robustness,
91	X	X			KC-135 Simulator	In response to growing costs and needing to save money during flight avionics software changes, desire KC-135 manufacturer to put hooks into avionics software to be used as simulator avionics in order to cut costs of developing a mimic system.	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness, Agility,

Change ID #	Changeability	Evolvability	Survivability	Robustness	System	Change Statement	Mapped Ility Labels
92	X				Integrated Deepwater System	In response to oversight from Congress and GAO while the Deepwater program was suffering time and cost overruns, USCG commandant desires change in Acquisition Architecture (Consolidating G-A, G-D, CG-85, CG-66, R&D, and HCA under leadership of Asst Commandant for Acquisition (CG-9) to improve efficiency and align organizational structure with DHS and CG mission support org. (CIAO-1) Upon assuming the Office of Commandant of the Coast Guard in May 2006, Admiral Allen ordered a top-down review of the Coast Guard's acquisition structure and processes. Admiral Allen recognized that the Coast Guard performed procurement and acquisitions for basic services to major systems in a less than synergistic manner. He asserted, "Although often successful, the processes were not optimally aligned to ensure standardization, or to control cost or schedule (Blueprint, 2008)." In addition to this order, Admiral Allen issued Commandant Intent Action Orders (CIAO), which had a significant impact on the Coast Guard's acquisitions community. In particular, two of the ten CIAO's have a direct impact on the acquisitions structure and processes. The CIAO's and the top-down review of the Coast Guard's acquisitions system were direct results of past assessment reports conducted by the Coast Guard, U.S. Department of Homeland Security and the U.S. government Accountability Office (GAO).	Changeability, Flexibility, Modifiability, Value Robustness,
93	X				Integrated Deepwater System	In response to oversight from Congress and GAO while the Deepwater program was suffering time and cost overruns, USCG commandant desires transformation in Logistics Integration Office structure to improve efficiency and standardize procedures and centralize supply chain. (CIAO-4)	Changeability, Flexibility, Modifiability, Value Robustness,
94	X	X			Integrated Deepwater System	In response to aging National Distress and Response System (NDRS) and need for better SAR communications, desire IDS to upgrade NDRS to Rescue 21 technology using satellite and upgraded VHF/UHF to improve clarity, direction finding, coverage and recording. Rescue 21 is an advanced maritime computing, command, control, and communications (C4) system designed to manage communications for the United States Coast Guard	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness,
95	X	X			Global Earth Observation System of Systems (GEOSS)	In response to growing SoS needs, desire new technologies added to GEOSS to conform to data and information sharing standards in order to interact with other SoS constituent systems.	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness,
96	X				Global Earth Observation System of Systems (GEOSS)	In response to growing agricultural security needs in GEOSS building time, desire GEOSS to add function of global agricultural monitoring system to promote food security and predict market trends. http://www.earthobservations.org/documents/pressreleases/pr_1111_geo_glam.pdf	Changeability, Adaptability, Functional Versatility, Modifiability, Extensibility, Value Robustness,

Change ID #	Changeability	Evolvability	Survivability	Robustness	System	Change Statement	Mapped Ility Labels
97	X				Global Earth Observation System of Systems (GEOSS)	In response to growing need in deforestation monitoring in context of growing GEOSS SoS in an increasingly 'green' focus, desire GEOSS to implement forest carbon monitoring system to measure deforestation and associated carbon emissions with more accuracy to aid research and policy makers.	Changeability, Adaptability, Functional Versatility, Modifiability, Extensibility, Value Robustness,
98	X	X			NYC Yellow Cab SoS	In response to new technologies in hybrid engines and desire for cars to be more efficient to save fuel, bring down costs, decrease emissions, desire Yellow cabs to transition to hybrid vehicles to decrease resources	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness,
99	X	X			Future Combat System (FCS)	In response to new political environment and military needs, desire US Army to restructure FCS program in to new Army Brigade Combat Team Modernization (ABCTM) program and cancel contracts with current lead systems integrators to save money. Inherited some FCS systems while canceling others.	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness,
100	X				Future Combat System (FCS)	In response to changing needs of the Army in urban warfare and aging manned vehicle fleets, Secretary Gates desired manned ground vehicle (MGV) program of FCS to be canceled to reevaluate needs of manned vehicles including mine-resistant, ambush protected vehicles (MRAPS) to save money and create more useful vehicle platform.	Changeability, Flexibility, Modifiability, Value Robustness,

12 Appendix C: Expert Interview Questionnaire

Empirical Investigation for Mapping Design Principles to Ilities

Clark Beesemyer

S.M. Candidate, Systems Engineering Advancement Research Initiative (SEAri)

Introduction

Hello, my name is Clark Beesemyer and I am a masters student at MIT in the Aeronautics and Astronautics department. I work with the MIT Systems Engineering Advancement Research Initiative (SEAri). I am currently conducting interviews to identify empirically derived design principles for enhancing non-traditional system properties that engineers could use in the high leverage early stages of design. Because of your expertise and experience in systems engineering, I believe that you can help in this endeavor. I want to thank you for your assistance and intend to keep this interview to no more than 40 minutes.

In this interview, I am gathering examples of decisions that were made during the design and development of systems in order to create, enable, or enhance certain lifecycle properties, such as flexibility or evolvability. Traditional "ilities" such as reliability, maintainability, and availability (RMA) are still important; however, the focus of this interview is to better understand how practitioners pursue less well understood non-traditional temporal lifecycle properties.

My thesis primarily deals with the evolvability, adaptability, and flexibility of systems and systems of systems (SoS). Maintaining system performance in the presence of uncertainties in design and operating environments is both challenging and increasingly essential as system lifetimes grow longer. In response to perturbations brought on by these uncertainties, such as disturbances, context shifts, and shifting stakeholder needs, systems may be able to continue to deliver value by being either robust or changeable. In this research, I am trying to find insights into how systems and SoS might be designed so that they can change in order to continue delivering value in spite of perturbations.

In this interview, when I use the word "ilities", I am referring to the non-traditional system properties, such as flexibility, adaptability, and evolvability.

First, may I gather some background information from you?

Background/Demographics

1) How many years of experience do you have in systems engineering or designing systems?

2) In what domain(s) do you have expertise? (For example, aerospace, automotive.)

3) What is your current employer and position?

Primary

Goal: To gain useful insight in how to achieve desired system properties in the design process by talking to industry experts who have real world experiences doing just that.

4) Have you participated in the design or redesign of a system in which you intentionally considered certain a non-traditional ility, such as flexibility, adaptability, or evolvability? What was the ility of interest?

5) In what domain would you classify this system (e.g., aerospace, automotive.)?

6) What is the targeted user base (e.g., government, industry, consumer, etc.)?

7) What is the approximate development cost of the system (e.g., hundreds, thousands, millions.)? The unit cost of the system?

8) What was the motivation(s) for the design/redesign and the particular ility mentioned earlier?

9) Was that ility explicitly desired from system stakeholders, or did designers pursue the ility on their own?

10) Specifically, what design decisions did you make in order to incorporate the desired ility into the system design?

11) Did you consider any other supporting ilities to achieve your desired ility (for example, using modularity as an enabling ility)?

12) Were there actual or perceived "extra" costs associated with incorporating the ility? If so, how were these justified to system stakeholders/funders/customers?

13) Was the system perceived as successful in meeting its requirements, including the desired ility? Were there any deficiencies or compromises that needed to be made in order to achieve that ility?

14) Were there any good or bad unintended consequences (e.g. system characteristics or system behaviors)?

15) Did system stakeholders perceive or realize the benefits from designing for this ility?

Secondary

Goal: To gain useful insight in SEAri's prescriptive design flows for including ilities, by talking to industry experts and comparing our models with their mental models.

16) We are developing a structured approach for guiding the inclusion of desired ilities into system designs. This approach begins with design principles and ends with desired ility characteristics. Design principles initially suggest design choices that allow systems the option of executing classes of changes, which we call "change mechanisms". It is the ability to execute these change mechanisms that give systems the desired system ilities. Does this approach seem descriptive of how you achieved your desired outcome in your system design? If not, how does your approach differ?

17) Do you believe this design flow for including ilities (using design principles to motivate ilities) would be effective in your organization and could be implemented by design engineers? If not, how can it be improved?

18) In your experience, do you believe a hierarchy of system ilities exists? That is, are certain system ilities serving as means to achieve the ends of other system ilities? For example, modularity as a means for achieving reconfigurability. Did you experience any similar relationships in your system design?

19) Is there any other information you think would be helpful in developing a process for designing systems with ilities, or considering tradeoffs between the ilities, which may not have been covered in other parts of this interview?

20) May we contact you for a follow-up interview?

13 References

Akmajian, A., Demers, R.A., Farmer, A.K., and Harnish, R.M., *Linguistics*. MIT Press: Cambridge, MA. 2001.

Algeddawy, T.N. and ElMaraghy, H.A., "Changeability Effect on Manufacturing Systems Design," Changeable and Reconfigurable Manufacturing Systems, Springer London, 2009.

Alliance, United Launch. *Atlas V Launch Services User's Guide*; 2010.

Bay, M., et al., "The Art and Science of Systems Engineering," NASA, January 2009. Retrieved April 3 from http://www.nasa.gov/pdf/311199main_Art_and_Sci_of_SE_SHORT_1_20_09.pdf.

Barbará D, Li Y, Couto J., "COOLCAT: an entropy-based algorithm for categorical clustering." In: Proceedings of the eleventh international conference on Information and knowledge management. ACM; 2002:582–589. Available at: http://portal.acm.org/citation.cfm?id=584888.

Beesemyer, J.C., Fulcoly, D.O., Ross, A.M., and Rhodes, D.H., "Developing Methods to Design for Evolvability: Research Approach and Preliminary Design Principles," 9th Conference on Systems Engineering Research, Los Angeles, CA, April 2011.

Beesemyer, J.C., Ross, A.M., and Rhodes, D.H., "Empirical Examples of System Changes to Frame Links between Design Decisions and Ilities," 10th Conference on Systems Engineering Research, St. Louis, MO, March 2012.

Blanchard, B.S. and Fabrycky, W.J., Systems Engineering and Analysis. Prentice Hall, 1998.

Boardman, J., and Sauser, B., System of Systems-the meaning of. Presented at *IEEE / SMC International Conference on System of Systems Engineering*, Los Angeles, CA. 2006.

Booton, R., and Ramo, S., "The Development of Systems Engineering," *IEEE Aerospace and Electronic Systems,* vol.AES-20, no.4, pp.306-310, July 1984.

Butterfield, M.L., Pearlman, J.S., and Vickroy, S.C., "A System-of-Systems Engineering GEOSS: Architectural Approach," *Systems Journal, IEEE* , vol.2, no.3, pp.321-332, September 2008.

Chase, W.P., Management of System Engineering. R.E. Krieger. Malabar, FL. 1984.

Chen, W., and Yuan, C., "A Probabilistic Design Model for Achieving Flexibility in Design." ASME Journal of Mechanical Design, 120, no. 9. 1998.

Colegrave, N., and Collins, S., "Experimental Evolution: experimental evolution and evolvability," *Heredity,* 100, 464–470, 2008.

Cotton, L.D., Haase, G.A., Havlicek, J.D., and Thal, A.E., "Value-Driven Enterprise Architecture Score (VDEA-Score): A Means of DoDAF Architecture Evaluation," *7th Conference on Systems Engineering Research*, Loughborough, UK, April 2009.

Dahmann, J. S., Baldwin, K. J., Understanding the current state of us defense systems of systems and the implications for systems engineering. In *IEEE Systems Conference*. 2008.

de Neufville, R. and Scholtes, S., *Flexibility in Engineering Design*, MIT Press: Cambridge MA. 2011.

de Weck O.L., de Neufville, R., and Chaize, M. "Staged Deployment of Ommunications Satellite Constellations in Low Earth Orbit." *Journal of Aerospace Computing, Information, and Communication*. 2004.

de Weck, O.L., Roos, D., and Magee, C., "Life-cycle Properties of Engineering Systems," in *Engineering Systems*, MIT Press: Cambridge, MA. 2011.

de Weck, O.L., Ross, A.M., and Rhodes, D.H., "Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities)", *3rd International Engineering Systems Symposium*, CESUN 2012, TU Delft, 18-20 June 2012.

Dyer, J.S., Fishburn, P.C., et al., "Multiple Criteria Decision Making, Multiattribute Utility Theory: The Next Ten Years." Management Science 38(5): 645-655, 1992.

Eisner, H., Marciniak, J., and McMillan, R., Computer-aided system of systems (s2) engineering. *Systems, Man, and Cybernetics, 1991. 'Decision Aiding for Complex Systems, Conference Proceedings., 1991 IEEE International Conference on*, 1:pp. 531–537. 1991.

Ercetin, O., Ball, M.O., and Tassiulas, L., Next generation satellite systems for aeronautical communications. Int. Journal of Satellite Communications and Networks. Vol.22: 157–179. 2004.

ElMaraghy, H.A. and Wiendahl, H.P., "Changeability – An Introduction," Changeable and Reconfigurable Manufacturing Systems, Springer London, 2009.

Ellison, R. and Woody, C. "Survivability Challenges for Systems of Systems," *News at SEI*, http://www.sei.cmu.edu/library/abstracts/news-at-sei/securitymatters200706.cfm (Accessed May 18, 2012) 2007.

Finkelstein, S. and Sanford, S.H., "Learning from Corporate Mistakes: The Rise and Fall of Iridium." Organizational Dynamics, 29 (2):138-148. 2000.

Fossa, C.E., et.al., An overview of the Iridium low earth orbit (LEO) satellite system. *Proceedings of IEEE 1998 National Aerospace and Electronics Conference,* (A99-17228 03-01): 152-159, 1998.

Fricke, E. and Schulz A.P., "Design for Changeability (DfC): Principles to Enable Changes in Systems Throughout Their Entire Lifecycle." Systems Engineering - 8(4): 342-359, 2005.

Fulcoly, D.O., Ross, A.M., and Rhodes, D.H., "Evaluating System Change Options and Timing using the Epoch Syncopation Framework," 10th Conference on Systems Engineering Research, St. Louis, MO, March 2012.

Gan, G., Ma, C., and Wu, J., Data Clustering: Theory, Algorithms, and Applications, ASA-SIAM Series on Statistics and Applied Probability, SIAM, Philadelphia, ASA, Alexandria, VA, 2007.

Giachetti, R.E., Martinez, L.D., Saenz, O.A. and Chen, C.S., "Analysis of the structural measures of flexibility using a measurement theoretical framework." International Journal of Production Economics 86, 2003.

Giffin, M., de Weck, O., Bounova, G., Keller, R., Eckert, C., Clarkson, J., "Change Propagation Analysis in Complex Technical Systems," ASME Design Engineering Technical Conferences, September 4-7, 2007, Las Vegas, Nevada, USA, DETC2007-34652. 2007.

Hall, A., A Methodology for Systems Engineering, NJ; Van Nostrand, 1962.

Hansen, TF., "Is modularity necessary for evolvability? Remarks on the relationship between pleiotropy and evolvability." *Bio Systems*. 69(2-3): 83-94, 2003.

Henderson, R.M., and Clark, K.B., "Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms," *Administrative Science Quarterly* Vol. 35, No. 1, Special Issue: Technology, Organizations, and Innovation. March, pp. 9-30, 1990.

Holtta-Otto, K., *Modular product platform design*. Espoo: Helsinki University of Technology. 2005.

Hwang, K.H. and Park, G.J., "Development of a Robust Design Process Using a New Robustness Index." 2005 ASME International Design Engineering Technical Conference, Long Beach, CA, September 2005.

INCOSE, Systems engineering handbook v.3. Tech. Rep. INCOSE-TP-2003-002-03, INCOSE. 2006.

Jamshidi, M., *Systems of systems engineering: principles and applications*, Boca Raton, FL: CRC Press, 2009.

Kahneman, D. and Tversky, A., "Prospect Theory: An Analysis of Decisions under Risk." Econometrics 47: 263-29, 1979.

Kahneman, D. and Tversky, A., Eds., "Choices, Values and Frames," New York, Cambridge University Press, 2000.

Kapurch, S.J., "A Common Language for Systems Engineering: NPR 7123.1," NASA Ask, Issue 26, Spring 2007.

Keeney, R.L. and Howard R., "Decisions with Multiple Objectives—Preferences and Value Tradeoffs," Cambridge, UK, Cambridge University Press, 1993.

Kelly, K., What Technology Wants. New York: Viking, pp. 406, 2010.

King, A.M. and Sivaloganathan, S., "Flexible Design: a Strategy for Design Reuse." Proceedings of Engineering Design Conference '98 on Design Reuse, London, UK, June 1998.

Li, Y., Xue, D., and Gu, P., "Design for Product Adaptability." 2007 ASME International Design Engineering Technical Conference, Las Vegas, NV, September 2007.

Maier, M.W., "Architecting principles for systems-of-systems." *Systems Engineering*, 1(4): 267–284. 1998.

Mandelbaum, M., *Flexibility in decision making: an exploration and unification.* Ph.D. Thesis, Dept. of Industrial Engineering, University of Toronto, Canada, 1978.

Mania, J.R., "Explosive Armor." Patent 3,592,148. 1 July 1971.

McManus, H. and Hastings, D.E., "A Framework for Understanding Uncertainty and Its Mitigation and Exploitation in Complex Systems," *IEEE Engineering Management Review*, Vol. 34, No. 3, pp. 81-94, Third Quarter 2006.

McManus, H., Hastings, D. and Warmkessel, J., "New Methods for Rapid Architecture Selection and Conceptual Design." Journal of Spacecraft and Rockets, 41(1): 10-19, 2004.

McManus, H.M., Richards, M.G., Ross, A.M., and Hastings, D.E., "A Framework for Incorporating "ilities" in Tradespace Studies," AIAA Space 2007, Long Beach, CA, September 2007.

Mekdeci, B., Ross, A.M., Rhodes, D.H., and Hastings, D., "System Architecture Pliability and Trading Operations in Tradespace Exploration." presented at the IEEE International Systems Conference, Montreal, PQ, 2011a.

Mekdeci, B., Ross, A.M., Rhodes, D.H., and Hastings, D.E., "Examining Survivability of Systems of Systems," INCOSE International Symposium 2011, Denver, CO, June 2011b.

Mekdeci, B., Ross, A.M., Rhodes, D.H., and Hastings, D.E., "A Taxonomy of Perturbations: Determining the Ways that Systems Lose Value," 6th Annual IEEE Systems Conference, Vancouver, Canada, March 2012.

NASA., NASA Systems Engineering Handbook. Ed. Cecilia Haskins. NASA, 2007.

Nickel, J., *Using Multi-Attribute Tradespace Exploration for the Architecting and Design of Transportation Systems*, Master of Science Thesis, Engineering Systems Division, MIT, February 2010.

Nilchiani, R., "*Measuring the Value of Space Systems Flexibility: A Comprehensive Six-element Framework*," Ph.D. in Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 2005.

Nilchiani, R. and Hastings, D.E., "Measuring the Value of Flexibility in Space Systems: A Six-Element Framework," Systems Engineering, 10: 26–44, 2007.

Osinga, F., Science, Strategy and War: The Strategic Theory of John Boyd. London, UK, Routledge, 2006.

Puttalsri, K., Malaithong, A., Schroepfer, B. "Will Globalstar meet the same fate as Iridium?" Colorado University. 31 May 2006.

Ramo, Simon., The Development of Systems Engineering. IEEE Aerospace and Electronic Systems. Vol. 20, Issue 4, pp 306-310. July 1984.

Rajan, P.K.P., van Wie, M., Campbell, M.I., Wood, K.L. and Otto, K.N., "An Empirical Foundation for Product Flexibility." Design Studies 26, no. 4, 405-438. 2005.

Rasmusson, E.M. and Wallace, J.M., "Meteorological Aspects of the El Niño/Southern Oscillation." *Science* . vol. 222, no. 4629: 1195-1202. 1983.

Rhodes, D.H., Report on the Air Force/Lean Aerospace Initiative Workshop on Systems Engineering for Robustness, Massachusetts Institute of Technology, July 2004.

Rhodes, D.H., and Hastings, D., "The Case for Evolving Systems Engineering as a Field within Engineering Systems," MIT Engineering Systems Symposium, March 2004.

Richards, M.G., *Multi-Attribute Tradespace Exploration for Survivability*, Doctor of Philosophy Dissertation, Engineering Systems Division. Cambridge, MA: MIT, 2009.

Rittenberg, L., and Tregarthen, T., Principles of Microeconomics. Flat World Knowledge. Ch. 13. 2009.

Roberts, C.J., Richards, M.G., Ross, A.M., Rhodes, D.H., and Hastings, D.E., "Scenario Planning in Dynamic Multi-Attribute Tradespace Exploration," 3rd Annual IEEE Systems Conference, Vancouver, Canada, March 2009.

Ross, A.M., *Multi-Attribute Tradespace Exploration with Concurrent Design as a Value-centric Framework for Space System Architecture and Design*. Dual-SM thesis in Aeronautics and Astronautics; Technology & Policy Program. Massachusetts Institute of Technology. Cambridge, MA, 2003.

Ross, A.M., M*anaging Unarticulated Value: Changeability in Multi-Attribute Tradespace Exploration*, Doctor of Philosophy Dissertation, Engineering Systems Division, Massachusetts Institute of Technology, Cambridge, MA, June 2006.

Ross, A.M. and Rhodes, D.H., "Anatomy of a Change Mechanism," SEAri Working Paper WP-2011-1-1, 2011, http://seari.mit.edu/papers.php [cited 9 Apr 2012].

Ross, A.M., Beesemyer, J.C., and Rhodes, D.H., "A Prescriptive Semantic Basis for System Lifecycle Properties," SEAri Working Paper WP-2011-2-1, 2011, http://seari.mit.edu/papers.php [cited 9 Apr 2012].

Ross, A.M. and Hastings, D.E., "The Tradespace Exploration Paradigm," INCOSE International Symposium 2005, Rochester, NY, July 2005.

Ross, A.M. and Hastings, D.E., "Assessing Changeability in Aerospace Systems Architecting and Design Using Dynamic Multi-Attribute Tradespace Exploration," AIAA Space 2006, San Jose, CA, September 2006.

Ross, A.M., Hastings, D.E., Warmkessel, J.M., and Diller, N.P., "Multi-Attribute Tradespace Exploration as a Front-End for Effective Space System Design," AIAA Journal of Spacecraft and Rockets, Jan/Feb 2004.

Ross, A.M., and Rhodes, D.H., "Architecting Systems for Value Robustness: Research Motivations and Progress," 2nd Annual IEEE Systems Conference, Montreal, Canada, April 2008a.

Ross, A.M., and Rhodes, D.H., "Using Natural Value-Centric Time Scales for Conceptualizing System Timelines through Epoch-Era Analysis." *INCOSE International Symposium 2008*, Utrecht, the Netherlands, June 2008b.

Ross, A.M., and Rhodes, D.H., "Anatomy of a Change Mechanism," SEAri Working Paper WP-2011-1-1, 2011, http://seari.mit.edu/papers.php [cited 15 Apr 2012].

Ross, A.M., Rhodes, D.H., and Hastings, D.E., "Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining Lifecycle Value," *Systems Engineering*, Vol. 11, No. 3, pp. 246-262, Fall 2008.

Ross, A.M., McManus, H.L., Rhodes, D.H., and Hastings, D.E., "Revisiting the Tradespace Exploration Paradigm: Structuring the Exploration Process," AIAA Space 2010, Anaheim, CA, September 2010.

Saleh, J.H., Hastings, D.E., and Newman, D.J., "Flexibility in System Design and Implications for Aerospace Systems," *Acta Astronautica*, Vol. 53, 2003.

Saleh, J.H., Mark, G.T., and Jordan, N.C., "Flexibility: a multi-disciplinary literature review and a research agenda for designing flexible engineering systems," *Journal of Engineering Design*, Vol. 20, No. 3. pp. 307-323. 2009.

Saxer, R.K., Knauf, J.M., Drake, L.R., Portanova, P.L., "Evolved Expendable Launch Vehicle System. (Cover Story)." *Program Manager* 31.2, 2002.

Schulz, A.P. and Fricke, E., "Incorporating Flexibility, Agility, Robustness, and Adaptability within the Design of Integrated Systems--Key to Success?" IEEE/AIAA 18th Digital Avionics Systems Conference, St. Louis, MO, 1999.

Schulz, A.P., Fricke, E. and Igenbergs, E., "Enabling Changes in Systems throughout the Entire Life-Cycle—Key to Success?" 2000 INCOSE International Symposium, Minneapolis, MN, July 2000.

Sethi, A.K. and Sethi, S.P., Flexibility in manufacturing: a survey. *International Journal of Flexible Manufacturing Systems*, 2 (4), 289–328, 1990.

Slack, N., "The Flexibility of Manufacturing Systems," International Journal of Operations and Production Management, Volume 7, Issue 4:35–45, 1987.

Smith, J.M., Evolutionary Genetics. New York, Oxford University Press, 1989.

Siddiqi, A. and de Weck, O., "Modeling Methods and Conceptual Design principles for Reconfigurable Systems." *Journal of Mechanical Design*. 130(10). 2008.

Stearns, S.C., "The Evolution of Life History Traits: A Critique of the Theory and a Review of the Data," *Annual Review of Ecology and Systematics*, Vol. 8, pp. 145-171, 1977.

Suh, NP., Axiomatic Design: The Principles of Design. Oxford University Press. 1990.

Toni, D.E. and Tonchia, S., "Manufacturing Flexibility: A Literature Review," International Journal of Production Research, Volume 36, Issue 6:587–1617, 1998.

Viscito, L. and Ross. A. M., Combining pareto trace and filtered outdegree as a metric for identifying valuably flexible designs. In 7th Conference on Systems Engineering Research, Loughborough, UK, 2009.

von Neumann, J. and Morgenstern, 0., "Theory of Games and Economic Behavior," Princeton, NJ, Princeton University Press, 1947.

Wagner, A., "Robustness and evolvability: a paradox resolved," Proceedings of the Royal Society Biological Sciences, 275, 91-100, 2008.

Wang, T., *Real Options "in" Projects and Systems Design - Identification of Options and Solution for Path Dependency*. Phd Thesis, Massachusetts Institute of Technology, Cambridge, MA. 2005.

Wasson, C.S., "*Systems Analysis, Design, and Development: concepts, principles, and practices.*" Hoboken, NJ (US): John Wiley & Sons. 2006.

Weigel, A.L., *Bringing Policy into Space Systems Conceptual Design: Quantitative and Qualitative Methods*, Doctor of Philosophy Dissertation, Technology, Management and Policy Program, MIT, June 2002.

Wertz, J.R., and Larson, W.J., Space mission analysis and design / edited by James R. Wertz and Wiley J. Larson. Kluwer Academic, Dordrecht; Boston: 1991.

Wiendahl, H.P., Heger, C.L., *"Justifying changeability: a methodical approach to achieving cost effectiveness,"* Proceedings of the CIRP 2nd International Conference on Reconfigurable Manufacturing, Michigan, August 2003.

Wiendahl, H.P., *"Some remarks on changeability, reconfigurability and flexibility of manufacturing systems,"* Paradigms of Manufacturing—A Panel Discussion, 3rd Conference on Reconfigurable Manufacturing, Ann Arbor, Michigan, May 2005.

Williams, J., "Correctly assessing the -ilities" requires more than marketing hype," *IT Professional*, vol.2, no.6, pp.65-67, Nov.-Dec. 2000.

Wolfowitz, P., Memorandum: "Defense Acquisition," Office of the Deputy Secretary of Defense, 30 October 2002.

Wu, W.W., "Internet Satellite Challenges," *2010 Second International Conference on Evolving Internet (INTERNET)*, pp.31-35, 20-25 September 2010.

Ziman, J. ed., Technological Innovation as an Evolutionary Process. Cambridge University Press; 2000:398. 2000.