Spring 2016

A meta-architecture analysis for a coevolved system-of-systems

George Anthony Muller IV

A META-ARCHITECTURE ANALYSIS FOR A COEVOLVED SYSTEM-OF-SYSTEMS

by

GEORGE ANTHONY MULLER IV

A THESIS

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

in

SYSTEMS ENGINEERING

2016

Approved by

Dr. Cihan Dagli, Advisor
Dr. Ivan Guardiola
Dr. Steven Corns

ABSTRACT

Modern engineered systems are becoming increasingly complex. This is driven in part by an increase in the use of systems-of-systems and network-centric concepts to improve system performance. The growth of systems-of-systems allows stakeholders to achieve improved performance, but also presents new challenges due to increased complexity. These challenges include managing the integration of asynchronously developed systems and assessing SoS performance in uncertain environments.

Many modern systems-of-systems must adapt to operating environment changes to maintain or improve performance. Coevolution is the result of the system and the environment adapting to changes in each other to obtain a performance advantage. The complexity that engineered systems-of-systems exhibit poses challenges to traditional systems engineering approaches. Systems engineers are presented with the problem of understanding how these systems can be designed or adapted given these challenges. Understanding how the environment influences system-of-systems performance allows systems engineers to target the right set of capabilities when adapting the system for improved performance.

This research explores coevolution in a counter-trafficking system-of-systems and develops an approach to demonstrate its impacts. The approach implements a trade study using swing weights to demonstrate the influence of coevolution on stakeholder value, develops a novel future architecture to address degraded capabilities, and demonstrates the impact of the environment on system performance using simulation. The results provide systems engineers with a way to assess the impacts of coevolution on the system-of-systems, identify those capabilities most affected, and explore alternative meta-architectures to improve system-of-systems performance in new environments.

ACKNOWLEDGMENTS

The work, guidance and support of many people made this research possible. I would to thank Dr. Cihan Dagli for his support and guidance while conducting this research, and for introducing me to the fascinating field of complex systems. It has been a privilege to learn from you. I am also grateful to Dr. Corns and Dr. Guardiola for their time, input and interest in this research. I have enjoyed working with you all.

I also wish to thank Captain Bill Peterson, USCG, Retired, for his willingness to provide an operator's perspective, and Dr. Robert Brigantic, USAF, Retired, for inspiring an interest in simulation. I am indebted to both of these gentlemen for their mentorship and for engaging me in interesting, meaningful work.

I would like to thank Dr. Dale Henderson, USA, Retired, for his willingness to teach, his insights into systems thinking and simulation, and for never doubting in my ability to accomplish this work.

Most importantly, this work would not have been possible without the constant support and sacrifices of my wife and boys. Thank you. I look forward to exploring new adventures together more than you know.

For the Creator of all things seen and unseen, who reveals their mysteries according to His time and purpose.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

1. INTRODUCTION

1.1. RESEARCH MOTIVATION

Modern systems continue to grow in complexity. This is driven in part by an increase in the use of systems-of-systems and network-centric concepts to improve system performance. This growth allows stakeholders to achieve improved performance at the cost of increased complexity. This complexity is a result of expanded mission sets, asynchronous development and integration of new and legacy systems, and changing operational environments. These challenges often result in systems unable to meet future demands, reducing their effectiveness from anticipated levels. As a result, systems-of-systems must adapt to these challenges to maintain or improve performance.

The concept of coevolution defines the behavior that results when systems and their environment each adapt to changes in the other. Coevolution is the behavior exhibited by a system and its environment adapting to changes in each other to obtain a performance advantage. While this phenomenon is recognized in many biological and ecological systems, its relevance and application to complex engineered systems has not been studied in great detail. The ability to characterize coevolution in engineered systems allows improved performance by focusing efforts on the right set of performance measures and system attributes early in the system lifecycle and when improving existing systems. The goal for understanding coevolution in engineered systems is to improve performance in uncertain future environments.

The aim of this research is to develop an approach for assessing a system-of-systems (SoS) that experiences coevolution. The counter-trafficking SoS is a system that reflects this behavior. This system is comprised of surveillance and interdiction systems, information sources, analytical practices, and decision makers. System components perform discrete functions which enable the detection, investigation, apprehension and prosecution of illicit network actors in order to disrupt their activities. Coevolution is specifically demonstrated between the detection and interdiction elements of the SoS meta-architecture and the smuggling vessels these sub-systems are designed to detect and interdict.

1.2. RESEARCH APPROACH

The research objectives are to demonstrate the impact of coevolution between a SoS and its operating environment. The following key research questions are explored in this work are:

- is coevolution present in current engineered systems-of-systems?
- if so, how can the system-of-systems meta-architecture be assessed to evaluate its performance?
- how should the system be targeted for improvement?
- what insights can modeling the system meta-architecture under coevolution provide?

The research approach to answer these questions evaluates the counter-trafficking SoS by extending a current trade study methodology, develops a conceptual meta-architecture representative of coevolutionary behavior, develops a set of models to assess this conceptual meta-architecture performance versus the existing meta-architecture in an adaptive environment, and evaluates the impact of uncertainty in the operational environment on the recommended SoS alternatives.

1.3. THESIS ORGANIZATION

This thesis is organized as follows: Chapter 1 has presented the research motivation and general approach. Chapter 2 provides a background in several relevant research areas, including systems engineering, complex systems, operations research methods and a history of drug-trafficking in the Americas. Chapter 3 reviews systems engineering and

SoS engineering concepts. A description of the counter-trafficking SoS is provided in Chapter 4. The research approach is described, thoroughly developed and applied to the counter-trafficking SoS in Chapter 5. Chapter 6 details an agent based model that expands the assessment to include variability in meta-architecture capabilities and uncertainty in the operational environment. Chapter 7 presents the results of this method and describes the insights resulting from this approach. The thesis concludes with a discussion of the applicability of the method and promising next steps in Chapter 8.

2. LITERATURE REVIEW

Existing research has explored the complex nature of systems-of-systems, characterized the relationships between constituent systems, and developed methods to estimate SoS performance. This body of research includes concepts from complex systems, SoS modeling and assessment, network centric systems, and multi-objective decision analysis.

Other research has focused on the effectiveness of counter-trafficking systems. This research includes systems engineering studies and analysis of drug trafficking organizations. Several operations research approaches have been used to support decision making in counter-trafficking. These approaches include search theory and the network interdiction problem.

This chapter reviews these related research areas.

2.1. SYSTEMS CONCEPTS

2.1.1. Systems-of-Systems. Jamshidi [1] defines systems-of-systems (SoS) as "large-scale integrated systems that are heterogeneous and independently operable on their own, but are networked together for a common goal." The engineering of SoS — SoS design, analysis, and development — is an emerging challenge due to the complex nature of these systems. These complexities result from defining, building and managing interfaces between systems that are asynchronously developed. New technologies that rely on rules governing behavior, such as autonomous systems, rather than fundamental control theory, such as plant or process control, contribute to this complexity.

The SoS concept is meaningful in that it challenges traditional views of complex problems. Biological, ecological and engineered systems have been identified as SoS, but the interest to systems engineers is on designing, assessing and managing engineered SoS. The Department of Defense (DoD) Systems Engineering Guide for Systems of Systems describes four SoS classes that each present unique challenges and opportunities: virtual, collaborative, acknowledged and directed [2]:

- Virtual systems-of-systems are not centrally managed, lack an acknowledged or stated purpose and rely on unmanaged interfaces for system operation.
- Collaborative systems-of-systems are primarily driven by volunteer effort, where standards are developed and maintained by a core set of stakeholders or agents. The Internet is an example of a Collaborative SoS.
- Acknowledged systems-of-systems rely on multiple systems that contribute to an overall purpose, but this purpose is not the single objective of any of the system within the SoS. Often, these systems have several objectives, are accountable to a wide variety of stakeholders, have different ways of measuring success, and obtain funding from disparate organizations. These factors combine to influence system development and sustainment approaches, which influence the overall performance of the Acknowledged SoS. The maritime counter-trafficking SoS, described in this thesis, is an example of an Acknowledged SoS.
- Directed systems-of-systems are developed and managed to provide a specific purpose. This SoS is centrally managed during development and operation in order to fulfill the stated purpose of the SoS. Like other SoS, this purpose may change over time, but support activities exist to enable the SoS to change in response to changing user needs and requirements, as well as the operational environment.

Jamshidi [3] discusses theoretical challenges for SoS. These challenges include developing robust SoS using biologically-inspired approaches, development of SoS standards, developing methods to design SoS architectures, designing SoS simulations, integrating constituent systems for the SoS, and characterizing emergence within complex SoS. Some of these challenges have been overcome for specific SoS efforts. However, the systems engineering community lacks general solutions and has not accepted standard methods or approaches for these SoS challenges.

The complexity of SoS interfaces, stakeholders, users and operating environments present challenges for traditional systems engineering analysis for SoS architectures. Agarwal et al. [4] propose a hierarchical architecture framework for Acknowledged SoS. This framework provides a means to describe a SoS meta-architecture, acquisition environment and constituent system interfaces using an agent based model (ABM). The model uses agent negotiation among the constituent systems to identify the best suited constituent systems in terms of capabilities and performance measures. The end result is a tool to aid SoS decision makers in negotiating and soliciting contributions from constituent system stakeholders.

Pape et al. [5] provide a method to compare SoS meta-architectures using fuzzy rules. The fuzzy rules are defined based upon SoS attributes such as performance, affordability and flexibility. The meta-architecture is represented as a chromosome in a genetic algorithm using a binary encoding scheme to define the presence of constituent systems and interfaces within the SoS. Chromosome fitness is then evaluated based upon SoS attributes, where constituent systems and interfaces present or absent in the SoS contribute to the overall SoS performance. The meta-architecture is optimized using a genetic algorithm to manipulate the presence of systems and system-level interfaces. Pape et al. [6] later applied a similar approach to intelligence, surveillance and reconaissance (ISR) and search-and-rescue problems, where both systems were categorized as acknowledged SoS.

Dagli et al. [7] developed a decision support approach for SoS managers. This approach uses the wave model of SoS development to model interactions between SoS managers and constituent system managers to negotiate the involvement of these constituent systems for certain SoS capabilities. This work developed a meta-architecture generation model, meta-architecture assessment model using key performance attributes, and cooperative, non-cooperative, semi-cooperative, and incentive-based negotiation models. The resulting integrated environment allows what-if analysis to explore collaboration between constituent systems and the resulting impact on SoS performance. By constructing a decision support tool, Dagli et al have addressed key challenges in SoS architecting, including addressing uncertainties from the variability and availability of constituent systems, exploring evolving needs of the SoS, accounting for socio-technical aspects of motivations of constituent system managers, and optimizing the architecture based on multiple objectives subject to budget and resource constraints.

Dagli et al. [8] describe an ABM that supports the acknowledged SoS manager in negotiating participation by the constituent systems. This tool uses agent behaviors for each of three participating agents classes (SoS acquisition environment, SoS agent, and constituent system agents). SoS meta-architectures are generated from negotiation rules, agent behaviors and a set of multi-objective optimization models. The result is a recommended SoS meta-architecture for an acquisition wave. This meta-architecture is optimized for the SoS environment while satisfying the constraints of the acquisition environment and constituent system preferences and behaviors.

Mour et al. [9] describe agent based modeling for SoS in the context of constituent system behaviors. Mour et al. [9] apply a discrete agent framework to the analysis of a littoral combat ship squadron. The purpose of this analysis is to assess the performance of the combat ship against different threats, since these ships can be reconfigured to perform a variety of missions. The authors identify unexpected results from the ABM of the combat ship, which provide insights into potential limitations or vulnerabilities of this SoS.

Garrett et al. [10] develop an assessment framework for the ballistic missile defense SoS. This framework focuses on the interfaces, interoperability and integration of constituent systems. The authors adopt a federated systems approach to constructing the SoS from the "bottom-up". Garrett et al. develop three approaches to assess the SoS. The first method uses graph theory to develop adjacency matrices that characterize SoS interfaces. Different matrices are developed for each mission within the fire control loop. The second approach is the development of interface readiness levels. Interface readiness levels are similar to technology readiness and enterprise readiness levels, but are focused on the maturity of constituent systems to deliver required information within critical time bounds. Finally, the authors propose agent-based modeling as a tool well suited for modeling the interfaces, interoperability and integration of SoS constituent systems.

Chepko and de Weck [11] implement a system architecture design optimization approach to reduce lock-in that arises from early design decisions. The authors developed a functional hierarchy for which system alternatives were developed. The optimization model addresses a hierarchical set of discrete and continuous variables, and compatibility constraints. The authors implemented the optimization model using a genetic algorithm, and identify future work opportunities including increasing the number of discrete variables and testing the genetic algorithm parameter space.

Ricci et al. [12] applies an options-based approach to maritime security SoS. The authors note that modern systems suffer from complex, highly dynamic environments that are inherently uncertain and contribute to reduced system performance. The work by Ricci et al. allows the identification of options early in the system development cycle to mitigate potential disruptions to system performance once the system is deployed. The authors apply the approach using a maritime security SoS and demonstrate that options can be identified early in the SoS architecture design process.

Alfaris [13] developed an approach called the Evolutionary Design Model and argues that such a framework would enable improved efficiency in the design of complex systems. Alfaris states that design is a complex, evolutionary process. This reflects design thinking that occurs over a period of time, as new information or environments alter the purpose or utility of certain system designs. This work uses logical modeling methods, including unified modeling language (UML), systems modeling language (SysML) and object-process methodology (OPM), in combination with mathematical modeling methods (synthesis, analysis, evaluation and optimization) to construct a framework for the evolutionary design model, and demonstrated the approach using an evolutionary design for a city.

2.1.2. Network Centric Systems. Network centric systems are systems that achieve a desired capability unachievable without connected communications [14]. These systems may be geographically separated but are connected by communication links. The concept of network centric systems has emerged with the growth in information access and sharing and the technologies that support these capabilities. Key performance concepts in network centric systems are information reach, quality, and timeliness. In general, maximizing these objectives results in improved performance for the system. Network centric warfare is the application of network centric systems to defense. The same concepts of network centric warfare apply beyond the defense space. Network-enabled capabilities support the "integration of sensors, decision-makers, weapon systems, and support capabilities to enable agility and thus permit commanders to better synchronize effects" [15].

The overarching goal of network centric systems is to enable improved capability, or system effectiveness, by improving the flow of information through the system to achieve information superiority. A successful network centric system takes advantage of these attributes to generate and use information superiority. This advantage is obtained through self-synchronization of actors and shared awareness of the operational environment. In network centric systems, not all actors need to have all information, but each actor requires the right information to use in their decision making process.

Cares et al. [16] describe fundamental considerations for networked, distributed systems and their application to network centric warfare. The authors highlight the benefits and drawbacks of different network architectures. Cares et al. build on network centric concepts to motivate system design from centralized, linear, non-networked systems to a decentralized, nonlinear, networked design paradigm. This early work was influential in motivating research into new ways of engineering SoS.

2.1.3. Complex Systems. SoS often have some characteristics of complex systems. According to Boccara [17], complex systems have three key characteristics:

- Independent agents that follow a set of rules which govern behaviors in response to the environment
- Emergent behavior of the system that results from interactions between individual agents
- No single control agent that governs the interactions between agents or prescribes the emergent behavior

Bohorquez et al. [18] and Spagat et al. [19] identify common relationships among complex conflict systems. Bohorquez et al. identifies trends in insurgent conflicts, which exhibit power law distributions for both number of casualties and number of attacks per day. They also find that the number of casualties is converging across insurgent conflicts. They use a simple model describing organizational dynamics and relate the size of attacks to group strength.

2.2. MULTI-OBJECTIVE DECISION ANALYSIS METHODS

System architecting problems are, except for the simplest systems, multi-objective decision problems. These problems involve multiple objectives that are evaluated against competing criteria or constraints. The objectives are typically the result of several desired system performance attributes for which no single system meets the desired performance. The criteria describe the set of feasible alternatives, given logical and functional dependencies across the set of alternatives [20].

2.2.1. Quality Function Deployment. QFD is a structured approach for solving complex systems problems. QFD provides a qualitative assessment and quantitative comparisons for how alternative solutions satisfy customer requirements. The QFD process begins with customer needs and captures customer value. Requirements are derived from the customer and stakeholders; the requirements define what the system must do and are then prioritized. A typical tool for documenting the QFD is the House of Quality. Using the requirements generated from elicitation, multidisciplinary teams identify alternative approaches to satisfying these requirements. The approaches are given a qualitative rating of how well the requirements are satisfied, and relationships among the solution alternatives are rated based upon dependence or conflict relationships. The results of the QFD process enable trade-space exploration by comparing the relative performance of each alternative, identifying requirements that are weakly or insufficiently satisfied and ranking each solution alternative. A rigorous QFD assessment further enables requirements traceability and subsystem design exploration through hierarchical matrices [21].

2.2.2. TOPSIS. The technique for ordered preference by similarity to the ideal solution (TOPSIS) is a quantitative multi-criteria decision making approach. TOPSIS allows the decision maker to compare alternatives to a positive ideal solution (the best conceivable alternative) and a negative ideal solution (the worst conceivable alternative). Decision attribute value is normalized and weighted according to decision maker preferences. Each alternative is evaluated based on its distance to the positive ideal (d_i^+) and negative ideal (d_i^-) solutions:

$$d_i^+ = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^+\right)^2}, \qquad i = 1, \ldots, m \tag{2.1}$$

$$d_i^- = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^-\right)^2}, \qquad i = 1, \ldots, m \tag{2.2}$$

where the value of each alternative against each attribute is v_{ij} and v_j^+ and v_j^+ correspond to the postive and negative ideal solution values for attribute j.

2.2.3. Fuzzy Inference System. Many complex systems engineering problems tend to be quite ambiguous, especially those with complex stakeholder environments. Ambiguity arises in system performance measures. Fuzzy logic is a method to handle such ambiguity that uses fuzzy sets. Fuzzy sets, unlike set theory, allow degrees of membership to membership functions. Singh and Dagli [22] use fuzzy inference systems to assess SoS performance using fuzzy attributes. Fuzzy rules are used for performance attributes, and these attributes are evaluated using a genetic algorithm that represents a system architecture.

2.2.4. Swing Weight Matrix. Value-focused thinking has been recognized for its applicability in multi-objective decision analysis and provides a way to assess multiple systems providing similar functionality [23, 24]. Parnell and Trainor [25] state that the goal of architecture development for precedented systems is to improve the system by enabling the system to better satisfy stakeholder values, ideally expressed as the multiple criteria that comprise formal MOEs. A comprehensive architecture analysis should consider stakeholder values, subsystem and component attributes and the operational environment which influences system performance. Parnell and Trainor [25] and Cilli and Parnell [26] investigated trades associated with ISR payloads, patrol craft range and endurance, and network centric concepts of information reach and quality by developing a trade study framework using swing weights to assess system performance. This framework follows the steps:

1. Define system objectives and measures in the fundamental objectives hierarchy
2. Develop value functions to map objectives and measures into a value (or utility) space
3. Develop the swing weight matrix by defining objective importance and measuring the range in value (swing) across all alternatives
4. Generate creative alternatives that satisfy system objectives and measures
5. Assess alternatives using deterministic analysis by scoring each alternative according to the swing weight matrix

6. Summarize results through graphs of objective measures versus performance (value)

7. Perform sensitivity analysis using probabilistic analysis and stochastic simulation

8. Communicate tradeoffs to stakeholders and decision makers

The swing weight matrix is used for complex decision problems that arise in trade studies. With large numbers of stakeholders, the complexity of the objectives also grows. The swing weight matrix allows alternative system architectures to be assessed in this environment, and also offers a method of communicating the results to decision makers. Swing weights are used to account for the variation in performance measures as well as the relative importance of each objective and measure to the decision. Table 2.1 presents the swing weight matrix. Objectives are given a weight corresponding that corresponds to the importance of the capability (columns) and measure variability (rows).

The swing weight matrix uses an additive value model to determine the value of each alternative. Suppose there are m different alternatives to select from, and n different attributes to compare. Then

$$v(x_j) = \sum_{i=1}^{n} w_i \, v_i(x_{ij}) \tag{2.3}$$

where $v(x_j)$ is alternative j's value in terms of all objectives, $i = 1, 2, \ldots, n$ is the index of the attribute measure, x_i is the performance value (or score) for attribute measure i and

Table 2.1. A general swing weight matrix structure. The swing weight matrix captures differences among both the decision criteria importance and measure variation.

		Capability Importance		
		Enabling	Critical	Defining
	High	F	C	A
Variability (Range)	Moderate	H	E	B
	Low	I	G	D

w_i is the weight given to attribute i. Note that $\sum_{i=1}^{n} w_i = 1$ for the additive value model. The additive value model allows quantitative assessment of the trade-offs among multiple competing criteria in terms of stakeholder value.

2.3. DRUG TRAFFICKING

Drug trafficking is one of the most prevalent, persistent and widely viewed forms of illicit trafficking and places a substantial burden on United States' social, economic and health institutions. The United States Office of National Drug Control Policy (ONDCP) estimates monetary costs of $120 billion in lost productivity, $11 billion in healthcare, and $61 billion in criminal justice costs [27]. The United States (US) has invested significant resources to develop systems that support counter-drug trafficking efforts and other interventions to thwart these impacts.

2.3.1. Chronology of Drug Trafficking. The drug trade has evolved over the last fifty years. Hyland [28] and Chindea [29] provide detailed descriptions of how these changes emerged in response to domestic and international policy. These changes include the trafficking routes and methods of moving drugs from South America into the US, and organizational structure changes in response to law enforcement and legal framework advances, and as competition among drug trafficking organizations (DTOs) grew.

Current counter-drug trafficking efforts involve a complex system that relies on muliple US and partner country organizations, intelligence, surveillance and reconaissance assets, and interdiction resources that span federal, state and local law enforcement agencies [30, 31]. Further, DTOs themselves are a complex system comprised of multiple organizations, each with unique goals and objectives, different functional roles and performance capabilities. These groups work together, largely through a network of semi-autonomous agents, to profit from the sale of illegal drugs in the US and Europe [32, 33, 34]. There is no uniform consensus regarding a well defined structure for DTO organizational structures over time. Rather, these groups tend to self-organize in response to the environment.

Trafficking objectives, competition for the drug trade, and law enforcement activity comprise this environment. Key trafficking objectives such as maximizing profit, minimizing the likelihood of interdiction and preventing the arrest and prosecution of group leadership serve as indirect measures of effectiveness for the trafficking system as a whole [35]. To meet these objectives, groups operate in a spectrum ranging from cooperation to competition with one another, employ tactics that lead to corruption or cooption of law enforcement and utilize ingenuity to gain an operational advantage in moving drugs from South America to the US. The tactics, techniques and procedures used to maximize system level measures of effectiveness (MOEs) are influenced by values held by the organization. The values differ among DTOs, as demonstrated by the use of violence in Mexico and Colombia [36].

Hyland [28] and Corcoran [37] identify four periods over the last century that typify the changing organizational structures and operational practices of DTOs. The following Sections summarize these periods and the changes that occurred [28, 37].

Drug trafficking in the Americas dates to the late 19th century. The origins of narcotics trafficking began with medical and scientific discoveries in the late 19th century. At the time, some narcotics (e.g. cocaine) were not regulated or criminalized as were marijuana and opium. Legitimate trade markets formed for cocaine, bolstering the economies of supplier countries such as Peru. At the same time, illegal narcotics were trafficked primarily by opportunistic individuals acting in self interest.

The criminalization of cocaine in the latter part of this period negatively impacted coca suppliers in South America, resulting in poor economic conditions in regions once booming with opportunity. However, demand for such products remained steady in the US and some European countries, providing an opportunity for illicit trafficking which developed throughout Latin America and Mexico. During this time, government officials in these regions realized the opportunity to profit from colluding with local traffickers, resulting in a culture of corruption.

The period after World War II gave rise to better structured organizations that focused on geographic specialization, which led to an initial surge in drug-trade related violence. The international community largely criminalized the use and trafficking of coca at this time, causing trafficking organizations to emerge in unstable regions in South and Latin America.

These organizations developed areas of specialization, including production, distribution and smuggling, that allowed them to take advantage of their location and indigenous capabilities. Cuba became one of the distribution centers as a metropolitan city and intermediate destination from South America to the US and Europe.

Toward the end of this period, the Cuban Revolution disrupted the drug trade, causing traffickers to move distribution operations to Central America and Mexico, which introduced trafficking to Mexican political and social structures. Well funded, the DTOs of this era capitalized on novel technologies, such as aircraft, to transport drugs into the US.

The demand for marijuana and cocaine surged during the 1960s to 1980s, creating lucrative opportunities for DTOs. As a result of US and Mexican efforts to curb marijuana production and distribution, Colombia emerged as a new haven for production and distribution hub for the drug. This was fueled by experience in historical trade of illicit goods and political instability in the region.

During this period, DTOs became much larger and well organized, typically exhibiting a hierarchical structure through the emergence of cartels. Production, trafficking and security operations were often bolstered by seasonal workers. At this time, Medellín, Colombia emerged as a trafficking hub and turned the distribution and smuggling activities into large logistics operations. The Caribbean emerged as a key transit route for drugs entering Miami and south Florida. Smugglers used cigarette boats (go-fast boats) and small aircraft to transit this route.

Toward the latter part of this era, cartels sought refuge in other Latin American countries as a result of Colombian and US pressure. These moves allowed the cartels to establish new relationships with other traffickers specializing in other commodities and drugs. These established DTOs had reliable routes through Central America and Mexico into the US.

Mexican DTOs emerged as the principle traffickers since the 1980s. The changing nature of the social and political environment contributed to the dissolution of cartels and their hierarchical structure. In response to law enforcement operations and judicial prosecutions, the organizational structures flattened, evolving into a network of specialized actors. This structure insulates key members from highly visible edge operations where law enforcement interdictions are likely to occur. In addition, these groups specialize in trafficking, coordinating production with other Latin American groups, and distribution in the US with US-based gangs. This structure allows the network to capitalize on localized control and areas of specialization while minimizing full network exposure to law enforcement detection and interdiction.

Drug related violence has risen sharply throughout Mexico since 2009. This violence stems from fighting among Mexican DTOs in efforts to control large, strategic geographic sections along the US–Mexico border. This allows DTOs to control the flow of illicit goods across the border. As the demand for illicit goods continues abroad, Mexican DTOs have identified new routes across the Atlantic to satisfy the demand.

Key transit corridors have emerged to allow DTOs to continue to bring drugs into regional distribution centers within the US. Mexican DTOs have adapted their commodities beyond drugs, and now move people (human smuggling) and other profitable contraband. As these illicit goods are brought north into the US, bulk cash and weapons are trafficked south into Mexico, allowing these organizations to continue their operations. Figure 2.1 depicts the estimated cocaine flows across each sector of the transit zone. The transit zone is comprised of approximately 6,000,000 square miles of open water. Asset capabilities to detect activities across this expanse are limited.

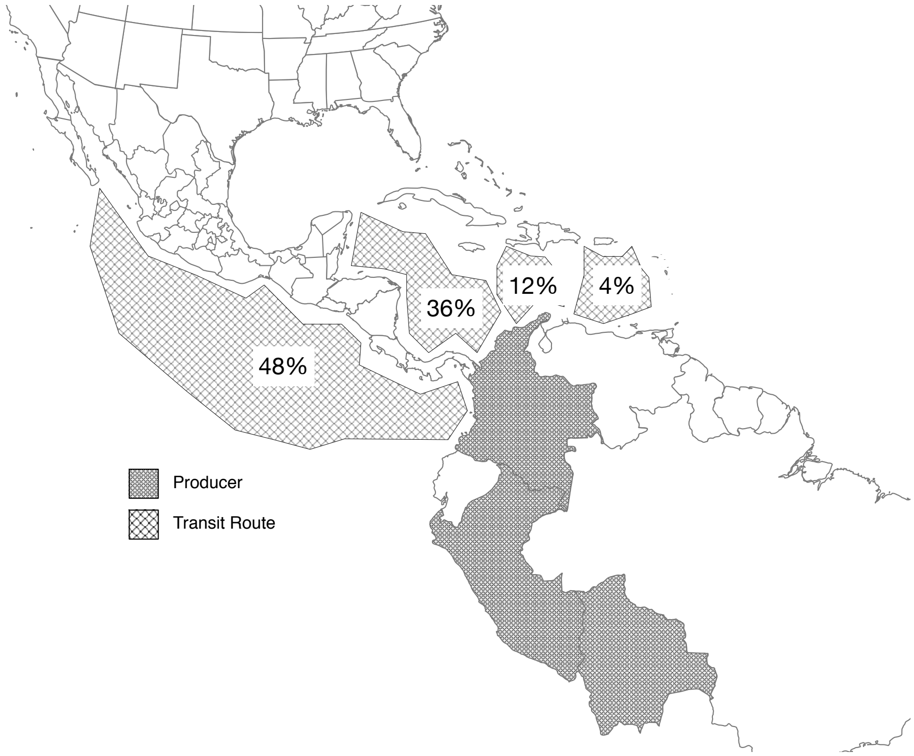


Figure 2.1. Trafficking routes across the Transit Zone. Adapted from United States Government Accountability Office [38].

2.3.2. Trafficking as a Complex System. The evolution of DTOs and counter-trafficking organizations are integrated throughout their history. Each system architecture adapted in response to changes implemented by the other. This is evident as counter-trafficking efforts closed gaps in one area, DTOs identified new opportunities elsewhere as they sought to profit from the demand of their illicit goods.

This artifact of coevolution is present in many complex systems, and remains a challenge to fully characterize and understand in even simpler systems [39, 40].

Thomas [41] provides a description of a recent counter-drug trafficking operation. Table 2.2 summarizes seizures that occurred during operation *Panama Express*. This is an example of coevolution between DTOs and the counter-trafficking SoS from the late 1990s to 2010. As a result of early interdiction successes, DTOs increasingly relied on sophisticated transportation methods to defeat counter-trafficking detection and interdic-

Table 2.2. Drug seizures and interdictions from Operation *Panama Express*. These seizures over ten years of the Organized Crime Drug Enforcement Task Force operation demonstrate changes in smuggling vessel use. Data are from [41].

Year	Fishing Vessels	Go-fasts	Semi-submersibles	Arrests	Cocaine (kg)
2000	3	7	0	46	23,960
2001	2	7	0	47	12,955
2002	5	11	0	75	35,446
2003	8	10	0	103	25,748
2004	10	13	0	131	58,997
2005	11	12	1	138	50,994
2006	11	17	1	127	45,907
2007	12	19	6	115	46,114
2008	5	9	6	90	32,834
2009	3	11	11	95	23,018

tion capabilities. The United States Coast Guard, the lead US law enforcement agency for maritime drug interdiction, focused substantial assets on the US Virgin Islands and Puerto Rico [38]. This shift in asset allocation occurred in response to perceived spillover violence in these US territories as DTOs used these locations as transfer points for entry into the US, but likely limited the maritime domain awareness for larger transit routes. These characteristics underscore the complexities involved in managing and operating the complex counter-drug trafficking SoS, as stakeholder values can overtake considerations of system performance as a whole.

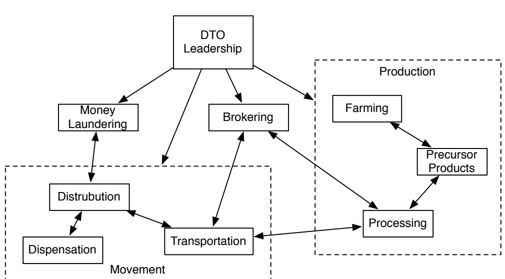
DTO physical architecture also changed during this time period. The limited speed and large profiles of commercial fishing vessels resulted in a growth in interdictions. Go-fast boats took the place of commercial fishing vessels since they could travel at much greater speed and out-maneuvre traditional US interdiction assets. As a result of this change in DTO physical architecture, the US responded by deploying helicopters with airborne use of force capable of interedicting go-fast boats.

Beginning around 2005, the DTO physical architecture changed again with the growing use of semi-submersible vessels. These vessels evade detection by many counter-trafficking surveillance assets. Semi-submersible vessels typically operate 80%-90% submerged. The US responded with new detection technologies to provide surveillance of these craft. Currently, DTOs use a mix of these trafficking vessels, and have recently deployed fully submersible vessels to carry drugs across the Transit Zone.

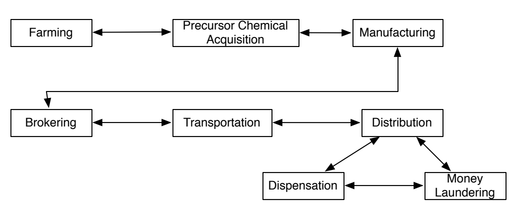
As a result of interdiction and prosecution successes of *Panama Express*, the structure of DTOs changed. Many cartel leaders were captured, and the DTOs changed into loosely connected logistics chains. These logistics chains, which include buffers to protect key members [28], emphasize the value that decentralized organizations can bring in addition to offering skill specialization drug trafficking supply chain.

Kenney [36] identifies different organizational structures observed in Latin American DTOs. These structures are broadly described as hierarchical (or wheel) and chain networks. Each network type offers unique benefits and limitations in managing trafficking activities, and insulating key group members from investigation or prosecution from law enforcement. Figure 2.2 presents the hierarchical network and chain structure observed by Kenney [36].

Physical architecture and organizational structure coevolution are clearly present between DTOs and the counter-trafficking SoS architecture over the last several decades. The ability of both to adapt to new tactics, techniques and technologies enables them to improve the respective system performance for a period of time. Well financed, DTOs



(a) Hierarchical, or Wheel, DTO Structure



(b) Chain, or Loosely Coupled, DTO Structure

Figure 2.2. DTO organizational structures. These structures are typical of Latin Amercian drug trafficking organizations, and have changed over time (adapted from Kenney [36]).

readily adapt using improvised methods to evade detection and avoid interdiction by law enforcement agencies.

2.3.3. Data Analysis. The United Nations Office on Drugs and Crime (UNODC) [42] hosts a research database which contains information on drug interdictions around the world provided by member states. The database contains records on the interdiction date, drug type, quantity seized, transportation method and, if known, source country, destination country and interdiction country. The data are available at http://data.unodc.org. A representative subset of this data is provided in Table 2.3.

Initial data analysis was performed by summarizing the number of interdictions and total cocaine seized across all sub-regions for each month in years 1998-2012. Figure 2.3 provides a sample result of this analysis. A subset of this data was used for the below modeling effort. This subset included total interdictions for the Caribbean subregion between the years 1998–2006.

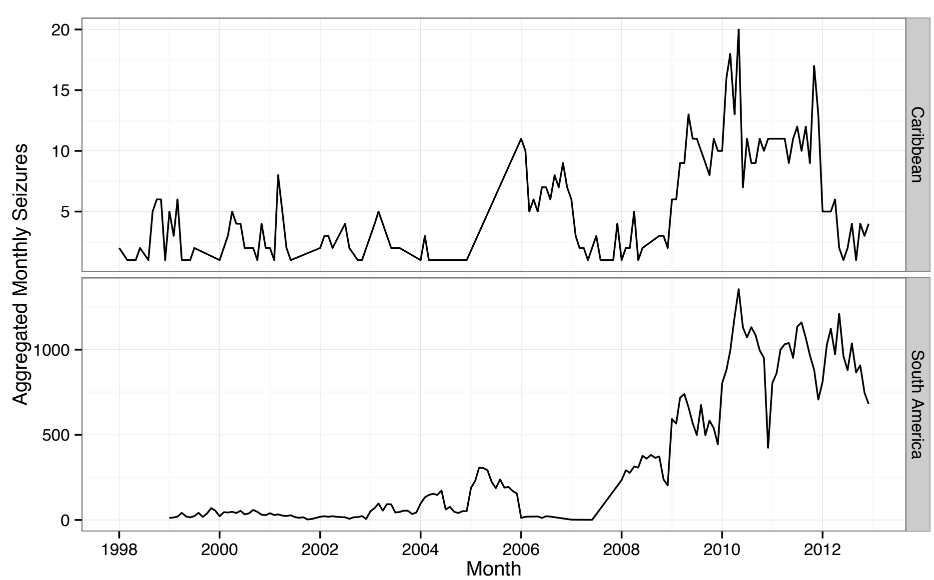


Figure 2.3. Monthly cocaine seizure summaries from the UNODC dataset. Only the Caribbean and South America subregions are shown. Note the significant difference in y scales to show the overall trends in each subregion.

Table 2.3. UNODC drug interdiction data subset. Additional information on date and departing/interdiction country, if known, are available in the original dataset.

SubRegion	Country	PlaceOfSeizure	DrugType	Qty (kg)	Installation	Transportation	Source	Destination
South America	Bolivia	La Paz	Cocaine Base	26.00				
South America	Bolivia	Cochabamba	Cocaine Base	6429.00				
South America	Bolivia	Cochabamba	Cocaine Base	6807.00				
South America	Bolivia	La Paz	Cocaine Base	2660.00				
South America	Bolivia	Chuquisaca	Cocaine Base	4.00				
South America	Bolivia	Cochabamba	Cocaine Base	6051.00				
South America	Argentina	Salta	Cocaine	135.66	Vehicle		Unknown	Unknown
South America	Argentina	Jujuy	Cocaine	50.62			Unknown	Unknown
South America	Argentina	Buenos Aires	Cocaine	0.53			Unknown	Unknown
South America	Bolivia	Cochabamba	Cocaine Base	0.76			Unknown	Unknown
South America	Bolivia	Santa Cruz	Cocaine Base	0.33	Airport	Commcl air	Unknown	Unknown
South America	Bolivia	Cochabamba	Cocaine Base	4538.00				
South America	Bolivia	Santa Cruz	Cocaine Base	6051.00				
South America	Bolivia	Cochabamba	Cocaine Base	1.27			Unknown	Unknown
South America	Argentina	Buenos Aires	Cocaine	3.31			Unknown	Unknown
South America	Colombia	Taraza	Cocaine Base	348.22			Unknown	Unknown
South America	Bolivia	Santa Cruz	Cocaine Base	1513.00				
South America	Bolivia	Santa Cruz	Cocaine Base	3782.00				
South America	Bolivia	Cochabamba	Cocaine Base	1.52			Unknown	Unknown
South America	Bolivia	Cochabamba	Cocaine Base	5.99		Commcl air	Unknown	Unknown
South America	Colombia	MEDELLIN	Cocaine Base	0.11	Residence	Pvt road	Unknown	Unknown
South America	Bolivia	Cochabamba	Cocaine Base	9077.00				
South America	Bolivia	Santa Cruz	Cocaine HCL	0.51			Unknown	Unknown

2.4. RELATED RESEARCH

Operations research methods and systems engineering studies have supported decision making for the trafficking problem. These include deterministic analysis methods, trade studies and simulation experiments. This section reviews these related studies.

2.4.1. Deterministic Analysis Methods. Game theory is suited to problems where one agent employs one of any number of strategies against a competing agent. Generalizations and extensions to games have been developed to include multiple players, cooperation, limited information, and continuous strategies among others. Two-person zero-sum games are a special class of games where exactly two agents whose payoffs are in direct opposition. These two players choose one strategy against the other player in order to maximize (or minimize) some payoff. These games have been extensively studied for network interdiction problems, where one player must choose some number of arcs to interdict in order to minimize the flow available to the adversary [43, 44, 45, 46]. In addition, Shieh et al. [47], Pita et al. [48] have recently implemented game theoretic models for decision support tools in airport and harbor security operations.

The network interdiction problem considers the allocation of limited surveillance assets to detect adversary activity. In this problem, described and formulated by Washburn [49], different types of detection assets m are deployed to monitor a region separated into n sectors (assuming the physical conditions across these sectors are equivalent, i.e. the detection rate for each asset type m is equal across all n sectors). Let d_{ij} be the detection rate for an asset of type $i \in m$ deployed to sector $j \in n$. Assuming that all assets monitor a sector independently, the total detection rate for sector j is $\sum_{i=1}^{m} d_{ij}x_{ij}$. The network interdiction problem can be summarized with the following:

- Index use

  $i \in m$: a detection asset type

  $j \in n$: homogeneous (operationally equivalent) sectors for asset deployment

- Data

  b_i : the total number of resources of asset type i available to assign

  d_{ij} : the detection rate of asset i operating in sector j

  y_j : the probability of detectable actions taking place in sector j

- Decision variables

  x_{ij} : the total number of assets of type i to deploy to sector j

  v : the total expected detection rate of adversary activity

- Formulation

  Washburn provides a linear programming formulation for the surveillance problem as follows [49]:

$$
\begin{aligned}
\text{maximize } & v \\
\text{s.t. } & \sum_{i=1}^{m} x_{ij} - v \geq 0, \qquad & j = 1, \ldots, n, \\
& \sum_{j=1}^{n} x_{ij} \quad \leq l_i, \qquad & i = 1, \ldots, m, \\
& x_{ij} \qquad \geq 0 \ \forall\, i, j.
\end{aligned}
$$

- Description

  To represent a limited number of deployable assets for each type m, a bounding constraint on x_{ij} is defined as $\sum_j x_{ij} \leq l_i$, where l_i is the maximum number of deployable assets of type $i \in m$. The payoff of this matrix game is the average detection rate against the activities across all sectors, represented as $A(\mathbf{xy}) = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} x_{ij} y_j$ where y_j represents the probability that a detectable activity occurs in sector j.

Pan [50] extended the network interdiction problem to consider adversary paths known probabilistically.

Search theory is a set of mathematical constructs that aim to provide estimates of the amount of effort required to detect a target in some search space [51]. Search theory is used in search and rescue (SAR), naval analyses, and astronomy [52, 53].

In the most general sense, two conditions must be satisfied for a successful search: the search must be performed in an area that includes the target, and searchers must be capable of detecting the search object, or target. There are a few key concepts in search theory that allow estimates of search effectiveness, probability of detection over time and other measures. In many cases, these solution methods require certain conditions to be satisfied, such as a stationary target or the absence of benign targets. Lateral range curves and sweep width are two key concepts in search theory that underpin its use for search problems. Lateral range curves correspond to the probability of a specified sensor detecting a specified target. The lateral range is the distance between the sensor and target at the point of closest approach. The lateral range curve is a probability distribution of detecting a target at different lateral ranges. The lateral range curve will vary by sensor-target combinations and environmental factors such as visibility.

"Cookie-cutter" detectors have a lateral range curve defined as follows:

$$p(x) = \begin{cases} 1.0 \text{ if } r \leq R, \\ 0.0 \text{ if } r > R. \end{cases} \tag{2.4}$$

where R is the lateral range of the detector and r is the lateral distance between the target and the detector. The M-beta search model is a generalization of the "cookie-cutter" detector, where $p(x)$ is a constant value in the interval $[0,1]$ across the entire lateral range of the detector [52].

The sweep width is a measure of search effectiveness for a particular sensor. Sweep width is defined as the area under the lateral range curve:

$$W = \int_{-\infty}^{\infty} p(x)\,\mathrm{d}x. \tag{2.5}$$

Search effort is another important concept in search theory. Search effort is the search area that is capable of being searched by a detector:

$$\begin{aligned} Z &= WL \\ &= WVT \end{aligned} \tag{2.6}$$

where Z is search effort, W is sweep width and L is the distance traveled by the sensor within the search zone, V is the speed of the sensor/detector and T is the amount of time spent in the search area. Coverage is another important factor in search theory. Coverage is defined as the ratio of search effort to search area:

$$C = \frac{Z}{A} \tag{2.7}$$

where C is coverage, Z is search effort, and A is the search zone

2.4.2. Systems Engineering Studies. Ruegger [54] explores the role of network centric systems to the mission of maritime domain awareness (MDA). He describes a network centric SoS for MDA by using a SoS engineering process, highlights alternative SoS capabilities and uses SysML and a discrete event network flow model to develop and evaluate the MDA SoS architecture. Ruegger uses a few performance measures to assess overall SoS performance, including the time to develop a common operating picture and the probability of common operating picture accuracy. The result is an assessment of alternative architectures focused on data exchange in a networked SoS. One of the key findings of this research is that in highly distributed networks (a key structure in network centric systems) communications should occur with reduced delay between any two nodes as long as there are no bottlenecks due to insufficient bandwidth.

Zorn [55] develops an architecture anlaysis of alternative system architectures for unmanned maritime systems (UMS). He explores the role of UMS to support United States

Coast Guard (USCG) missions, including maritime domain awareness, search and rescue, and counter-trafficking. Zorn develops capability needs for UMS, develops UMS alternative architectures, including unmanned underwater vessels (UUVs) for USCG acquisition, and performs a feasibility analysis for the implementation of these UMS architectures. The result is a recommendation for a path forward to developing and acquiring UMS capabilities for the USCG.

Hayes and Paulo [56] use discrete event simulation to assess a naval command and control (C2) system architecture. This method develops the functional and physical architectures for the system, then assesses different C2 structures in the form of a network, where functions are allocated to different node types within the network structure. Their preliminary results demonstrate the trade offs between a decentralized and distributed network structure and highlight the need for additional work in both C2 architecture analysis and inclusion of other system level performance measures in order to fully assess the differences between the system architecture alternatives.

Bong [57] applies a systems engineering approach to analyze interagency coordination and effectiveness in support of DoD combatant commands. This approach reviews the functional and physical architectures of successful systems that support interagency coordination. Bong applies the results to the development of a notional architecture for an interagency coordination center for the Joint Interagency Counter-Trafficking Center. Bong demonstrated that the systems engineering process could be successfully applied to organizations.

Abeto [58] applies the systems engineering process for interagency counter-trafficking and counter-terrorism efforts in DoD's European Command (EUCOM). The process includes operational concept development, stakeholder analysis, system objectives and requirements definition, and functional decomposition. The result is a functional architecture that focuses on information sharing across stakeholders and a method to assess system performance as a feedback mechanism to improve system performance. This work help policy makers focus on key system interfaces to improve the counter-trafficking and counter-terrorism efforts by EUCOM.

2.5. SUMMARY

The design and analysis of complex coevolutionary SoS remains an open challenge in systems engineering. A body of research has explored alternative methods that support assessing SoS. This includes static models of SoS, deterministic analysis methods for trade studies, and ABMs to simulate SoS. However, this work has not been integrated to understand the impact of coevolution in the SoS domain.

The counter-trafficking system is an example of a coevolutionary system. This system has adapted to changes in the operational environment over the last several decades. However, most analyses of this system have focused on deterministic methods to optimize resource allocation or used traditional systems engineering approaches to evaluate alternatives or improve aspects of the system. Some work has been done to explore the utility of new capabilities (e.g. UUVs) in the future. Other work has explored evolutionary technological changes in smuggling vessels. These works have not been integrated to explore the resulting performance degradation as a result of smuggling vessel changes or explored new architectures in response to these changes.

Existing research provides the basis for an approach to assess complex coevolutionary SoS. This includes trade studies for constituent systems using multi-objective decision analysis and agent-based modeling to support SoS meta-architecture evaluation.

Coevolution exists in many systems. This thesis demonstrates that it exists in the counter-trafficking SoS and adapts existing work to measure the impact on the SoS, create new alternative meta-architectures, and assess performance trades. This is accomplished by integrating multi-objective decision analysis, deterministic analysis and agent-based modeling. Multi-objective decision analysis provides an approach to evaluate SoS meta-architectures by assessing the contribution of constituent systems that provide a specified capability through the value of constituent systems to the overall SoS objectives and measures. This results in a set of constituent systems for an initial SoS meta-architecture. Coevolution impacts SoS performance through changes in the operating environment. Agent based modeling is used to simulate the SoS in representative operational environments that mimic coevolution between the SoS and its environment. The results of the simulation allow assessment of the SoS while accounting for evolutionary changes in the SoS architecture, operating environment, and smuggler behavior.

3. SYSTEM CONCEPTS

A system is a collection of objects that interact to achieve or perform a capability that individual components cannot perform alone. The International Council on Systems Engineering (INCOSE) defines systems engineering as “an interdisciplinary approach and means to enable the realization of successful systems” [59]. The systems engineering process begins with a need for some capability that a collection of components cannot readily address. Several models have been developed to describe the systems engineering process. In general, the systems engineering process includes iterating through the activities of assessing, designing, building, and validating the system.

- the process of describing what the objects are to do
- defining the allowable ways it can be done
- identifying alternative ways of doing it
- defining the criteria that govern how well an alternative satisfies the objectives
- assessing the performance of the available alternatives according to these criteria
- recommending an approach
- documenting the design and development of the alternative
- testing system components

3.1. THE SYSTEMS ENGINEERING PROCESS

The systems engineering process serves as a risk mitigation against failures in challenging development efforts. For monolithic systems, the formal systems engineering process is a collection of activities that have developed over time, refined by experiences in increasingly complex and large-scale systems. Ideally, systems engineering maximizes the benefit to the customer through analysis of the system problem, evaluation of stakeholder needs, and the assessment, design, development, testing and implementation of the best solution. The boundaries between systems engineering activities is often fuzzy, and can require iterative refinements, especially for requirements development and analysis. Figure 3.1 is one model of the systems engineering process including formal deliverables.

Systems engineering is, in essence, a risk mitigation against failed development for complex engineered systems. The systems engineering process described by Blanchard and Fabrycky [21] is summarized in Sections 3.1.1–3.1.4.

3.1.1. Conceptual Design. The conceptual design phase is a critical step in the systems engineering process. This phase sets the design and implementation trajectory for all later steps. Errors or mistakes made early on in the systems engineering process heavily influence downstream activities. The conceptual design phase should strike a balance between concepts that are too narrow, missing key requirements and design alternatives, and concepts that are too broad, which can push design refinement to further phases, risking project schedule and budget. The following activities are performed during system conceptual design:

- Need Identification and Problem Definition: The systems engineering process begins with the identification of a needed capability, which can arise for both new capabilities (and hence a need for new systems), and for precedented systems, in which improved performance is required. It is important to document the need and reasons why the specific capability is necessary. This informs the system requirements, stakeholder analysis and conceptual design activities which will influence the overall system design and implementation.

- System Planning and Architecting: System planning activities develop formal documentation, including a program management plan, systems engineering management plan, development of technical requirements. Systems architecting activities include

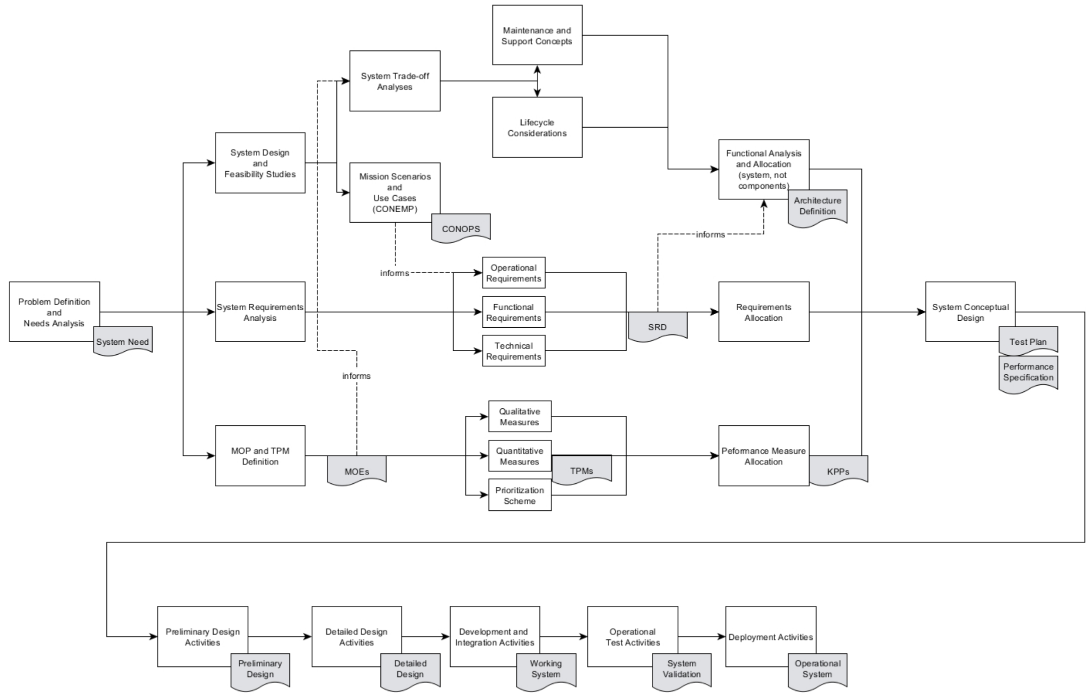


Figure 3.1. A model of the Systems Engineering process with deliverables. The Conceptual Design phase is emphasized, and formal deliverables are highlighted by shaded boxes.

development of the functional architecture, the physical architecture (mapped to the functional architecture), operational requirements definition, development of alternative concepts, and performing feasibility studies of these concepts.

- Conceptual Design and Feasability Analysis: The conceptual design phase identifies alternative system concepts that address the stated system need and evaluates these concepts against important stakeholder criteria. These criteria typically include system performance, effectiveness, sustainment and life-cycle cost considerations. The result is a recommended system concept that best meets the stated need.

  The analysis of system alternatives results in major decisions that are made and heavily influence the resulting work to bring the system into being. The considerations that go into these analyses are problem dependent. Weather considerations impact alternatives designed to operate in outdoor environments, obsolescence considerations impact technology development and deployment systems, and data communication standards impact network systems. Systems that encounter each of these scenarios should plan and account for all of them. These decisions can have immense impact on the performance, behavior, and ultimately, the utility of the system that is developed to address the need.

- Requirements Definition: The requirement definitions, specifically the operational requirements definition activity, identifies the missions the system is expected to perform, key performance parameters, deployment and distribution estimates, lifecycle considerations, utilization requirements, effectiveness factors, and environmental factors. For systems-of-systems, interoperability requirements are also addressed.

- Maintenance and Support Concepts: System operation and support are often the most costly activities for developed systems. The system maintenance and support concepts identify how the system will be maintained once it is developed and in operational use. This includes levels of maintenance, organizational responsibilities and maintenance support activities.

- Measures of Effectiveness: Qualitative measures that describe how well the system meets its intended purpose. Measures of effectiveness (MOEs) are provided by the acquirer or user of the system to describe the operational effectiveness of the solution [59].

- Measures of Performance: Quantitative measures that describe how well the system meets the required functionality; measures of performance (MOPs) characterize the physical and functional attributes relevant to system operation [59]

- Technical Performance Measures: Technical performance measures (TPMs) are quantitative measures that specify the standard, or threshold, against which a requirement should be met. TPMs typically result from operational requirements and describe the performance of system components to ensure that they meet system requirements [59].

- Functional Analysis: Functional analysis explores what the system is to do. There are a number of functional decomposition methods and tools, including integrated definition (IDEF) modeling and functional flow block diagrams.

- Trade-off Analyses: System level trades evaluate alternatives with regard to technical performance measures and system level objectives. The result is a recommendation for a set of perferred alternatives given objectives and constraints. At the system level, these are typically multi-objective decision problems.

- System Specification: The above conceptual design activities contribute overall guidance for how the system is brought to be. The system specification isa formal document describing how conceptual design elements are combined and integrated to specify the system.

- Conceptual Design Review: A formal review to evaluate the system specifiation and conceptual design. This review allows statkeholders to provide recommendation for correction before progress to preliminary design.

3.1.2. Preliminary System Design. The focus of this phase is to allocate requirements to subsystems and describe the interfaces between subsystems. The goal is to reduce the abstraction of conceptual design prior to detail system design. Steps of preliminary system design include:

- Preliminary Design Requirements: Preliminary requirements describe what the subsystems are to do.

- Preliminary Design Specifications: This includes specifications for Development, Product, Proces and Materials. These are formal documents detailing the technical requirements and design standards.

- Subsystem Functional Analysis and Allocation: Subsytem function and interfaces are assigned to subsystems. Models such as functional flow block diagrams are often used.

- Preliminary Design Criteria: These criteria describe how the "–ility" design considerates are to be addressed.

- Design Engineering Activities: These activities call for an increased role of design engineering disciplines within integrated teams.

- Trade-off Studies and Design Definition: This inlucdes evaluation of alternative subsystem configurations and establishing a system configuration at the component level.

- Preliminary Design Review: A formal review of the preliminary design with stkeholders. This allows recommendations or adjustments before moving to detailed system design.

3.1.3. Detail Design and System Development. This phase includes the detailed design of system components and interfaces. The following are elements of detailed design and system development:

- Detail Design Requirements: The lowest level requirements that the system must satisfy are allocated to components.

- System Integration: Ensures the collection of components operate together as a system, and that the system meets performance objectives and satisfies system-level requirements.

- System Development: This includes prototype development, iterations through design reviews, and includes evaluation and feedback reviews.

3.1.4. System Test and Validation. The purpose of system test and validation is to ensure that the system operates as designed and satisfies operational and performance requirements. This phase includes test and evaluation planning and reporting to validate that the system meets these requirements.

3.2. ARCHITECTURE ANALYSIS

Architecture analysis is among the first activities that solidify the abstract notions described by the system need and system concept. It enables trade-space exploration, accounts for multiple stakeholder objectives, and forms a systematic set of views of the system [60]. Systems engineering has a strong influence on the overall design and direction of the system during the conceptual design stages, where architecture analysis is performed (Figure 3.2).

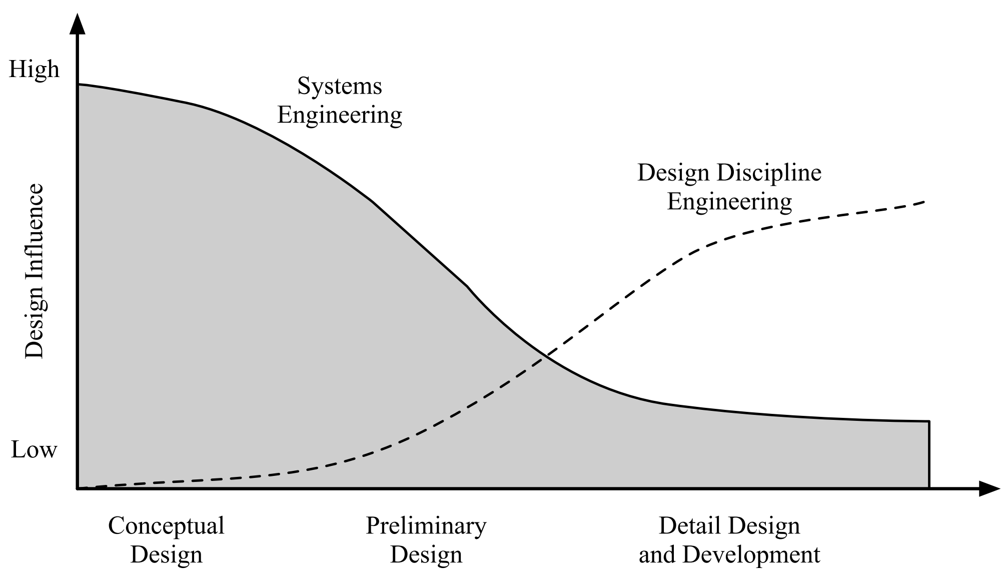


Figure 3.2. The influence of systems engineering and design disciplines in overall system design. Early systems engineering design phases have tremendous influence in the overall system design. System architecture development is a key task in developing a successful system concept. Adapted from Blanchard and Fabrycky [21].

3.2.1. DoD Architecture Framework. The DoD Architecture Framework (DoDAF) is an overarching model that supports decision making in DoD organizations [61]. The DoDAF provides a common definition for system architecture development. A standardized set of models, or *views*, enable information to be shared across DoD organizations to communicate effectively. The DoDAF view families include:

- All Viewpoint provides a summary for the scope, constraints, and assumptions used to develop the system architecture views
- Capability Viewpoint describes the capability taxonomy and visualizations of capability evolution
- Data and Information Viewpoint defines the business and operational information rules and requirements
- Operational Viewpoint describes the tasks and activities and resources required to carry out the operational aspects of the system
- Project Viewpoint provides information on the organizations, programs and projects involved in developing the required capabilities
- Services Viewpoint describe the resources that support the operational and support capabilities of the system
- Standards Viewpoint defines the rules for the configuration and relationships between elements of the architecture description to ensure operational and capability requirements are met
- Systems Viewpoint describes the systems and interconnections necessary to support operational activities and information exchange across system components

3.2.2. Role of Modeling and Simulation. Models support decision making and are constructed based on the specific decisions they are intended to support. Simulations execute models over time and space to understand the behaviors that result from interactions between system components.

Modeling and simulation are used throughout the systems decision process [20]. This includes problem definition (conceptual models), solution design (cost analysis), decision making (risk and trade analysis) and solution implementation (system control and logistics). Simulation is used within the systems decision process, such as discrete event simulation for assessing operational factors of the system, and physics-based simulations that model the physics of system comopnents, such as communications in electronics components or probability of detection for radar systems.

3.3. ENGINEERING COMPLEX SYSTEMS-OF-SYSTEMS

A system-of-systems is a system composed of multiple subsystems [59], called constituent systems. Systems-of-systems typically involve complex stakeholder environments as a result of some degree of autonomy of each sub-system owner. As a result, the system architect or system engineer is constrained not only by the existing performance, maturity or technological capabilities of constituent systems, but by organizational and policy challenges as well. In these environments, the SoS missions can become secondary or tertiary roles for constituent systems. However, the SoS mission requires the constituent systems to gain some performance advantage.

The traditional systems engineering process is well suited for monolithic systems. In these cases, the environment does not substantially change and diminish system performance after it is deployed into the operational environment. Systems-of-systems differ from monolithic systems in a few ways. Parnell et al. [20] identify several trends that create challenges in applying traditional systems engineering to complex SoS. These challenges are a result of increasing:

- Complexity due to engaging increasing numbers of engineering disciplines and growth in the types and number of interfaces
- Dynamics due to interactions with, and changes in, the environment
- Stakeholders that contribute input and have unique objectives for the system
- Security and Privacy Concerns as a result of the amount of information stored, accessed and available through interconnected systems

A number of characteristics help distinguish monolithic systems from SoS (Table 3.1). Sage and Cuppan [62] identify these as operational and managerial independence of the constituent systems, geographic distribution (networked systems), emergent behavior and evolutionary development. Sage and Cuppan identify that SoS exhibit aspects of complex adaptive systems, and may require federated systems engineering principles and evolutionary acquisition approaches to address the complexity inherent in SoS. However, systems-of-systems integrate constituent systems to enable new capabilities for a particular mission or purpose. This has, in many cases, changed the system development perspective from requirements based to capabilities based [63].

Bar-Yam [65] describes recent challenges in applying the traditional systems engineering process for complex SoS and provides a description of the differences between traditional SE and SE for modern complex systems. For example, the NASA systems engineering process does not account for drastically changing operating environments — the requirements do not change. Current SoS have asynchronous development cycles for each constituent system. A current challenge is integrating these constituents over time to realize the desired set of capabilities for the SoS. Systems developed throughout the Cold War provide examples of changing architectures. These systems were developed in anticipation of adversary response. The goal in such systems is to assess how best to respond to uncertain futures by focusing on the behavior that we are concerned with. Network centric

Table 3.1. Characteristics of monolithic systems and acknowledged SoS. Characteristics are adapted from [63] and [64].

Environment	Monolithic System	Acknowledged SoS
Scope	Fixed or known	Varies with availability of constituent systems and operating environment
Stakeholders	Clearly defined	Differences between SoS and constituent systems with competing objectives
Organization	Hierarchical; centralized	Networked; decentralized or autonomous
Management	Formal roles; funded scope	Independent funding and development of constituent systems
Operational Focus	Conceived, designed and developed to meet operational objectives	Needed to satisfy operational objectives that may not align with constituent system objectives
Acquisition	Documented requirements and milestones; SE process	Independent development; coordination between novel, new and legacy systems
Evaluation	Usually performed bottom up; planned test phases	Challenges due to availability and technological maturity (synchronization) of constituent systems; potential for unintended consequences; continuous testing with changes in constituent systems
Boundaries	Includes system components	Includes systems that contribute to the SoS; changing operational environment
Interfaces	Component-component interfaces	Enable control, information and data flow; balance constituent system needs
Performance	Clear, unambiguous and measurable performance measures	Capabilities that rely on contribution of constituent systems
Behavior	System behavior is deterministic	Uncertainties introduced as a result of constituent system availability; systems working together for a particular capability
Evolution	Planned; version controlled	Largely uncoordinated; opportunities by reduced capability from changing missions or operating environment
System Development	Systems engineering process	*An unsolved engineering challenge*

systems and modern combat systems are other examples where traditional SE needs to be extended to address inherent complexity.

One of the challenges in SoS is evolutionary development. Planning for new capabilities is a challenge as a result of future uncertainty regarding the operating environment

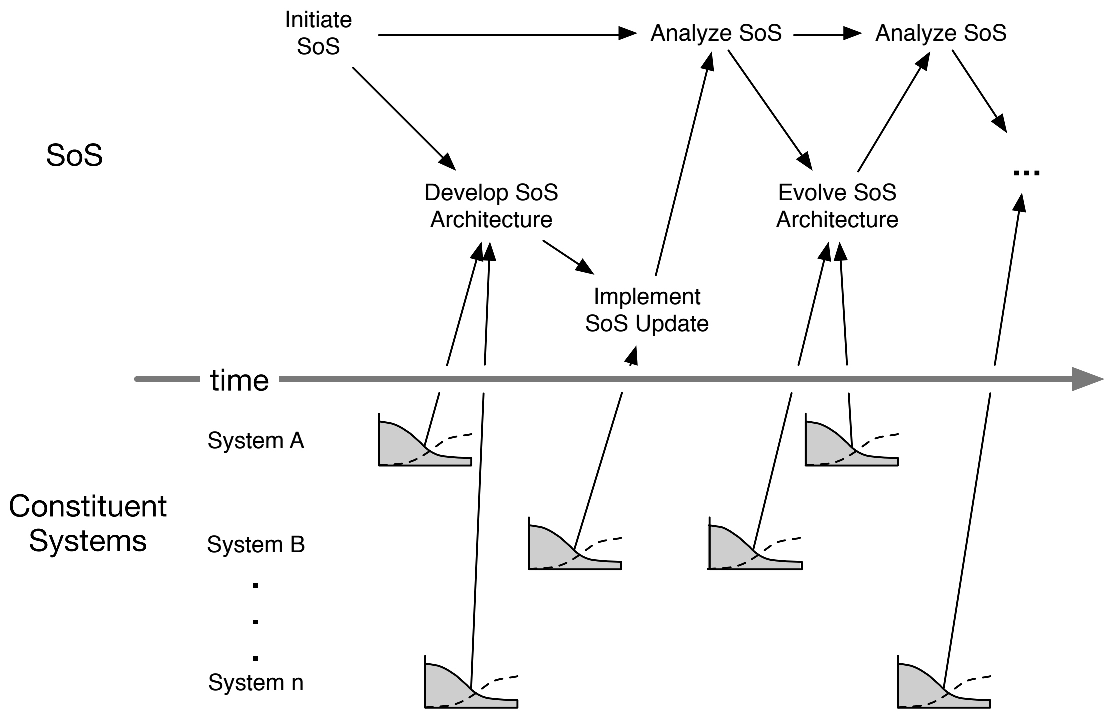


Figure 3.3. A model of asynchronous development in a SoS. The acknowledged SoS development concept begins with a needed capability and identified constituent systems. The constituent systems undergo independent development and evolution which lead to asynchronous SoS development and integration. This represents a key challenge that separates engineering SoS from traditional systems engineering for monolithic systems.

and availability of capabilities. Figure 3.3 illustrates the changes and adaptions of the SoS by incremental changes in constituent systems. The SoS is initiated with planned constituent systems. The constituent systems undergo development and evolution independently, which leads to asynchronous SoS development and integration. These challenges are examples of the difference between SoS analysis and traditional, monolithic systems engineering analysis. Fang and DeLaurentis [66] use an approximate dynamic planning approach to optimize the iterative SoS development process based on cost, schedule, performance and risk. This supports SoS architecture decision making by recommending alternatives at each iteration that SoS managers can consider for implementation.

Meilich [67] reviews some challenges in applying traditional systems engineering to the SoS environment. Meilich also identifies that, given current trends toward complex SoS, considerations such as flexibility, adaptability, and interoperability with systems or capabilities that were not envisioned early in the system development are becoming increasingly important. This contrasts with monolithic systems, for which optiming the system for a particular purpose tends to be an objective for the system engineer. Meilich notes that experimenting with the SoS as it evolves is one way to address the complex behavior that emerges from the SoS. This requires modeling and simulation tools to assess SoS adaptations in the environment in order to generate insights for planning future capabilities.

However, simulating complex SoS, including the constituent systems, interrelationships and environment interactions, is an open challenge Kewley et al. [68] develop federated simulations for a SoS. Federated simulations, including distributed interactive simulation (DIS) and high level architecture (HLA), allow interoperability between models that have developed at varying levels of detail using a defined interface. The federated simulations developed by Kewley et al. use federates for information exchange, environment representation, entity representation, model development and data collection applied to a swarm of semi-autonomous unmanned aircraft systems. The goal of this work is to develop improved rules governing the behavior of the SoS constituents. Baldwin et al. [69] suggest that agent based modeling is well suited for general SoS simulation. However, agent based models can be challenging to validate empirically. Baldwin et al. find that discrete event simulation supports validation of the agent based model for the SoS.

The test and evaluation process also separates traditional systems engineering from SoS engineering. Due to asynchronous development of the SoS constituent systems and coupling of the SoS with the environment, the testing process becomes convoluted. Testing of SoS constituents can take place with operational systems

There are several open challenges in engineering systems-of-systems. These challenges include evaluating the SoS alternatives at the SoS level, and generating feasible alternatives for the SoS:

1. Evaluating SoS alternatives at the SoS level requires that the interfaces between each SoS constituent system may be well defined in terms of the physical, functional and communication layers. However, understanding the interdependencies between the systems that comprise the SoS, especially with respect to changes in performance, is much less understood. In essence, these interdependencies can exhibit force multiplier effects, where systems that serve similar, but distinct, functions, work in tandem to generate an improved overall performing architecture. This is an example of non-linearity resulting from SoS capabilities and interfaces.

2. Generating feasible alternatives is an unsolved problem in engineering SoS. This is a combinatorial problem in the SoS case, where each combination of constituent systems becomes an SoS alternative.

In the architecture evaluation for a single system, trade studies compare several alternatives in order to choose a single best alternative in terms of overall system performance, expressed through MOEs, MOPs and TPMs (or collectively, a value model). For the SoS case, we may seek to employ multiple alternatives due to reliance on these systems for other missions. In the case of drug interdiction, USCG surveillance aircraft are used for other missions beyond surveillance of drug traffickers. Search and rescue missions are a greater priority when loss of life or property are at stake. Similarly, US Navy vessels serve to provide national level defense capabilities that are required in times of conflict. These platforms are equipped with a variety of other technologies and equipment to support the national defense mission.

A meta-architecture represents the constituent systems within the SoS. This representation defines the relationships between the constituent systems, oriented toward a desired SoS capability, and based on individual capabilities of the SoS constituents. The interfaces between constituent systems typically are in the form of communications interfaces. However, other interfaces, including physical interfaces, can also be important in situations where one system depends on the other. For example, when one system depends on another in order to be delivered to the operating environment, the payload of the delivery system becomes a consideration given the volume and weight of the delivered system [70, 71].

4. THE COUNTER-TRAFFICKING SOS

The counter-trafficking SoS exists to detect trafficking activities and disrupt these activities through supply reduction, smuggling interdiction and trafficker prosecution. This requires the coordination of several geographically dispersed systems that operate together. These systems support resource planning, information sharing, detection and interdiction operations, and coordination with law enforcement for trafficker prosecution.

The counter-trafficking system is an example of a system-of-systems. Table 4.1 describes the stakeholders involved in this SoS. Each stakeholder has a role in the counter-trafficking effort, but none exist solely to detect and interdict drug trafficking between South America and the US. Each stakeholder has slightly different values and performance measures to assess counter-trafficking performance, and each brings unique capabilities to the counter-trafficking mission.

The counter-trafficking SoS has many characteristics of an acknowledged SoS:

- several stakeholders with differing objectives at the SoS and constituent levels (Table 4.1),
- constituent systems that participate in the SoS and other, unrelated missions,
- constituent systems that contribute required SoS capabilities,
- a decentralized management structure, with organizational efforts combined to satisfy the overall counter-trafficking SoS goals and objectives,
- constituent systems developed asynchronosly under differing management and systems engineering structures,
- interfaces centered on information sharing across constituent systems and associated organizations,

Table 4.1. Stakeholders of the counter-trafficking SoS. Many stakeholders are users or operators of the system. Some functions are shared, but tailored for specific missions. US policy and law control the roles, responsibilities and jurisdictions of US organizations.

Agency	Role or Function	Goals and Objectives
Department of Defense (USSOUTHCOM)	Performs monitoring and detection of trafficking activities; provide support to law enforcement interdiction operations	Targeted number of interdictions [72]
JIATF-South	Performs interagency and international coordination for detection and monitoring; resource allocation; facilitates interdiction of illicit trafficking and other threats in support of national and partner nation security.	
United States Navy	Interdiction vessels, ISR asset support [38]	
Department of Homeland Security		
United States Coast Guard	Maritime surveillance and interdiction	Reduce the flow of cocaine [73]
Customs and Border Protection	Land and maritime border drug interdiction	Sustain desired readiness rate [73]
Immigrations and Customs Enforcement	Interdiction support	
Intelligence Community	Drug related intelligence [74]	
Department of State	Coordination with partner countries [74]	
Department of Justice	Prosecution [30]	
Drug Enforcement Agency	Drug related intelligence and interdiction [41]	
Federal Bureau of Investigation	Drug related intelligence and interdiction [41]	
Partner Countries	ISR and interdiction; asset and resource deployment [38]	
State and Local Law Enforcement	Drug interdiction; intelligence; prosecution [30]	

- opportunistic SoS development; architecture is incrementally developed based on the availability of tools and platforms

Based on these characteristics, the counter-trafficking SoS can be considered an acknowledged SoS.

A key counter-trafficking effort is the interdiction of cocaine and other illicit drugs in the transit zone. Cocaine typically moves from South America, through the western Pacific Ocean or Caribbean Sea, to Central America and the United States. The ONDCP estimates that 84% of illicit drugs were moved across the transit zone using noncommercial maritime means in fiscal year 2013 [38]. Drug trafficking organizations use different types of vessels to smuggle narcotics across the transit zone. Table 4.2 identifies a few categories of these vessels and their characteristics.

The United States Coast Guard is the lead US organization for executing maritime counter-trafficking efforts in the transit zone. The Joint Interagency Task Force - South (JIATF-South) coordinates the activities of several organizations to support these efforts. The USCG, Customs and Border Protection (CBP) and USN contribute vessels and aircraft. JIATF-South also receives resources and support from partner countries, including the United Kingdom and Canada. The US effort to disrupt the drug supply focuses elements of the counter-trafficking SoS close to the source of drug supply to maximize interdiction opportunities of higher-value cargo loads. Due to the Posse Comitatus Act, DoD is prohibited from engaging in civilian law enforcement. As a result, USCG personnel are assigned to certain USN vessels in order to perform law enforcement functions in drug interdiction areas [38, 73, 75].

Table 4.2. Maritime smuggling vessel properties. These DTO maritime smuggling vessels are encountered in the Transit Zone. These vessels have differing speeds, ranges, payloads and detection and evasion capabilities that are unique to each type of vessel. The values of these characteristics are estimated for each smuggling vessel type.

Name	Domain	Range (nm)	Payload (metric tons)	Speed (knots)	Cost
Fishing Vessel	Sea	3,000	5–10	12	$100,000
Go-fast Boat	Sea	400	0.5–3	70	$300,000
Semi-Submersible	Sea	2,000	3–10	10	$500,000
Fully Submersible	Sea	2,000	3–10	10	$900,000

4.1. SOS ARCHITECTURE

The counter-trafficking SoS uses existing DoD and DHS systems to provide the necessary capabilities to perform the counter-trafficking mission. The engineering challenge for this SoS is to improve how these systems operate in order to maximize the amount of smuggled drugs and other contraband being trafficked into the US. In order to do this, it is important to understand the required capabilities of the counter-trafficking SoS, the systems that can be used to provide those capabilities, and how those systems can be integrated and operated to maximize SoS effectiveness. The following sections describe the set of capabilities and functions of the SoS, the physical architecture in terms of constituent systems that satisfy teh capabilities, and an operational view of how the system works today. The definition of these systems begins with a functional decomposition of *what* these systems do, defining and assessing the relationships that exist between system functions.

4.1.1. Capabilities and Functional Architecture. In systems engineering, a functional architecture describes what the system is to do. For SoS, missions specify what the SoS is to do, and capabilities are derived from those missions. The mission of the counter-trafficking SoS is to disrupt the flow of illicit drugs, typically cocaine, into the US. The capabilities for the counter-trafficking SoS are provided in Figure 4.1 [76]. The remainder of this research omits the capabilities shaded in grey because they are considered supplementary to the primary focus of the counter-trafficking SoS system studied here.

- Detect Smuggler: Detecting the smuggler consists of the following capabilities:
  - Detect Smuggling Vessels: The counter-trafficking SoS must be able to detect smuggling vessels in order to allocate resources and understand the operating environment. Without detection, the counter-trafficking mission cannot be performed.
  - Identify Smuggling Vessels: Smuggling vessels typically operate in areas where commercial and some recreational traffic are also located. Smugglers use fish-

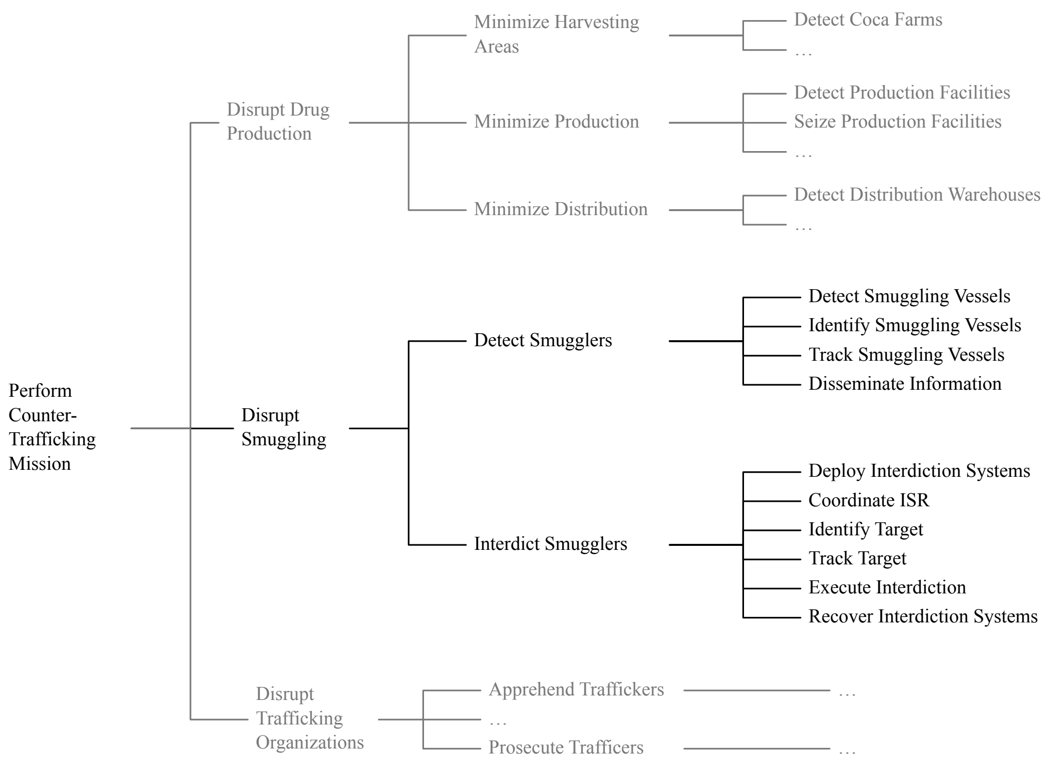


Figure 4.1. Counter-trafficking SoS capabilities. These capabilities include disruption of drug crop harvesting, smuggling, and drug trafficking organizations. Each of these capabilities is decomposed to identify the systems and platforms that provide those capabilities in order to perform the counter-trafficking mission. Capabilities shaded grey are not considered in the remainder of the SoS analysis.

ing vessels to blend in with the environment. Smugglers have also used sailboats and yachts to traffick drugs. The ability to identify smuggling vessels from benign traffick is necessary to reduce the number of false positive identifications, which reduces the availability of valuable assets.

– Track Smuggling Vessels: In order to coordinate smuggler interdiction, targets must be tracked spatiotemporally. This requires that detection assets have visibility of targets over a period of time.

– Disseminate Information: The coordination of detection and interdiction assets is an important aspect. Information on targets allows targets to be prioritized and interdiction resources to be organized. Information dissemination is also an important feature in the broader counter-trafficking mission, since different organizations likely have access to different types of information, as well as the engagement and coordination of activities with partner countries.

- Interdict Smugglers: The interdiction of smugglers is also necessary to successfully disrupt drug trafficking. Interdiction capabilities include:
  - Deploy Interdiction Systems: Interdiction assets must be able to physically access the target vessel. This requires some deployment capability or platform that can place these assets within range of the target.
  - Coordinate ISR: Coordination of ISR enables the information disseminated from the detection systems to be used by the interdiction systems. The timeliness and availability of this information is critical for successful interdictions.
  - Identify Target: Smuggling vessels may be in proximity to other types of vessels. The capability to hand-off the target from the detection system to the interdiction system may be accomplished through visual means, or may require additional identification capabilities. The purpose of this capability to to ensure that the correct vessel is interdicted.
  - Track Target: Once the target is identified, it must be tracked in order for interdiction assets to physically interdict the vessel. This may be accomplished through coordination with detection systems that have detected, identified and tracked the vessel, or may require the interdiction system to track the target independently.
  - Execute Interdiction: Interdiction requires personnel to physicall board the target vessel. In some cases, an additional capability, such as airborne use of force, may be required. Boarding approaches may depending on the target vessel type.

   - Recover Interdiction Systems: After the interdiction is completed, interdiction personnel, smugglers and any drugs seized are recovered.

4.1.2. Physical Architecture. Disrupting maritime smuggling is a key focus of the counter-trafficking effort, and the focus of the remainder of this research. The capabilities of the counter-trafficking SoS require constituent systems for command and control, and detection and interdiction. For counter-trafficking, platforms are needed to deploy detection and interdiction assets. The constituent systems available to support these capabilities have varying performance in range, speed, cost and target detection and interdiction. These systems are coordinated to disrupt smuggling activities of the vessels described in Table 4.2. Table 4.3 summarizes the amount of time different types of counter-trafficking SoS constituent systems were used for the counter-trafficking mission in 2013 [38].

- Detection: Detection capabilities include surface search radar, synthetic aperture radar and visual detection. Intelligence can also contribute to detection, although the intelligence role is not explored for this analysis. These sensors require a platform to deploy into the operational environment. These platforms typically include aircraft and surface vessels.

- Identification: Target identification can occur using electro-optical/infrared sensors, or through visual means. Like detection sensors, the ability to identify targets re-

Table 4.3. Surveillance asset availability. Data are for fiscal year 2013, from United States Government Accountability Office [38].

Agency	Support Type	Total	Units
CBP	Maritime Patrol Aircraft	6134	Hours
USN	Maritime Patrol Aircraft	2100	Hours
USN	Maritime Patrol Vessels	429	Days
USCG	Cutter Boats (combined)	1500	Days
USCG	HC-130	3750	Hours

quires aircraft or surface vessel platforms to deploy the sensors in the operating environment.

- Tracking: Vessel tracks are based on target detection. These detection events are recorded and coordinate through a command and control center. Campos III [77] describes a method that develops target tracks to generate probability density maps of the target location over time. This supports the allocation and movement of interdiction assets to the vicinity of the target.

- Surveillance: Surveillance systems are platforms that support detection and tracking of smuggling vessels. These systems are typically aircraft that can operate for extended periods of time over large geographic areas. These platforms are also used for other missions. Representative surveillance systems are presented in Table 4.4.

- Interdiction: Systems to support interdiction must be capable of reaching the target location and conducting the interdiction operation. For some targets, such as evasive go-fast boats, interdiction can require an additional capability such as airborne use of force. In this case, the target is immobilized so that interdiction can occur. Other targets, such as submersibles, can evade current interdiction capabilities by staying submerged. Interdiction platforms are also used for other missions, such as search and rescue. Representative interdiction craft are presented in Table 4.5.

Table 4.4. Representative surveillance systems. The counter-trafficking SoS includes additional surveillance systems from other agencies and organizations.

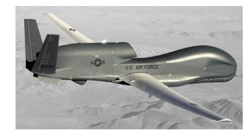
	HC-144A	C-27J	HC-130H	HC-130J	Scaneagle	RQ-4 Global Hawk
						
Organization	USCG	USCG	USCG	USCG	USCG	USCG
Class	Medium Range Fixed-Wing Aircraft	Medium Range Fixed-Wing Aircraft	Long Range Fixed-Wing Aircraft	Long Range Fixed-Wing Aircraft	UAS	UAS
Radar	Multi-mode surface search	Multi-mode surface search	Multi-mode surface search	Multi-mode surface search		
EO/IR	Y	Y	Y	Y		
Night Vision	Y	Y	Y	Y		
Encrypted Communications	Y	Y	Y	Y	Y	Y
Maximum Speed (knots)	246	317	380	380	80	112
Cruise Speed, est. (knots)	230	220	374	374	55	70
Range (nm)	2000	2675	2487	5000	809	12300
Endurance (hours)	11	12	14	14	24	32

Table 4.5. Representative interdiction systems. The USCG has primary responsibility for smuggling vessel interdiction in the Transit Zone, since DoD organizations are prohibited from performing law enforcement operations.

	MH-60T Jayhawk	MH-65D Dolphin	LRI-II	OTH-IV	RB-S II	RB-M
						
Class	Medium Range Recovery Rotary Wing	Medium Range Recovery Rotary Wing	Long Range Interceptor Boat	Long Range Boarding Vessel	Short Range Response Boat	Medium Range Response Boat
Interception Capability	Airborne use of force;	Airborne use of force;	Sea	Sea	Sea	Sea
Radar	Y	Y				
EO/IR	Y	Y				
Maximum Speed (knots)	180	175	38	40	45	43
Cruise Speed, est. (knots)	150	148	30	30	35	30
Range (nm)	700	290	225	200	175	250
Endurance (hours)	6.5	3			8	24
Launch Platform	NSC	NSC	NSC	NSC	NSC	Shore
Crew (personnel)	4				3	4

4.1.3. Operational View. The Operational View (OV) provides a high level context for how the system achieves the intended capability. Figure 4.2 depicts an OV-1 for the counter-trafficking SoS. Connections between systems represent available communication interfaces.

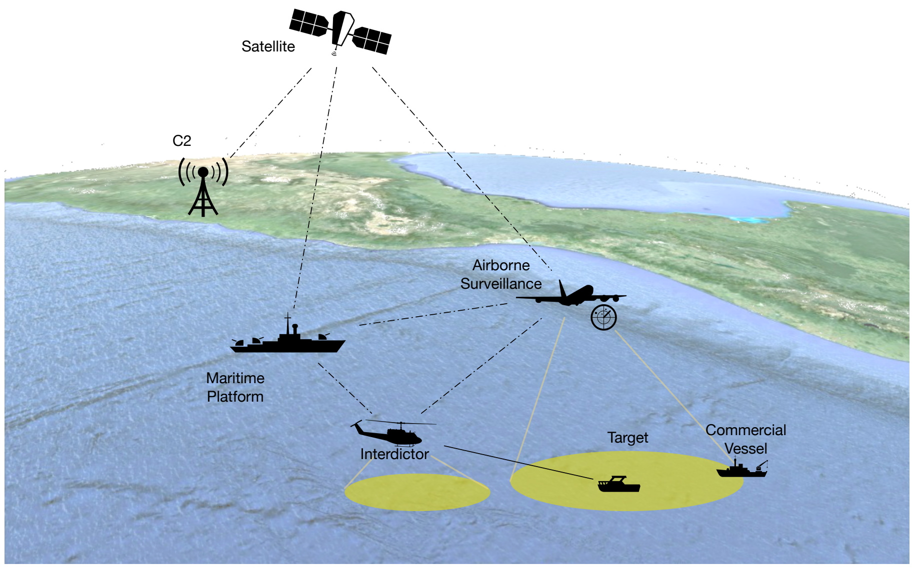


Figure 4.2. OV-1 of the initial SoS meta-architecture.

4.2. COEVOLUTION OF THE COUNTER-TRAFFICKING SOS

Drug trafficking organizations adapt trafficking modes and methods to circumvent detection and interdiction capabilities of the counter-trafficking SoS. Complexities such as the varied stakeholder environment, constituent systems integration, and competing mission priorities make analyzing the counter-trafficking SoS a challenge. A changing operating environment also leads to the complex nature of the counter-trafficking SoS.

Openly available data reporting specific types of trafficking vessels is limited. Table 2.2 provides a small dataset for a single operation that demonstrates one aspect of coevolutionary behavior on the part of DTOs. Between 2005 and 2009, the fraction of seized semi-submersibles increased from 0% to 44%. This significant increase reflects changing capabilities on the part of DTOs as well as the counter-trafficking SoS.

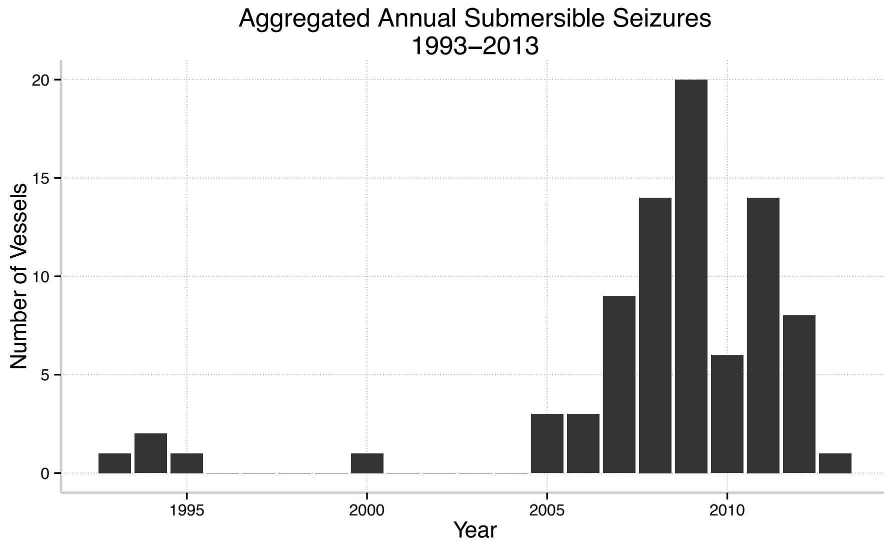
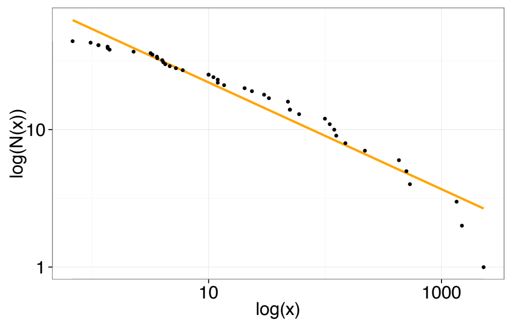


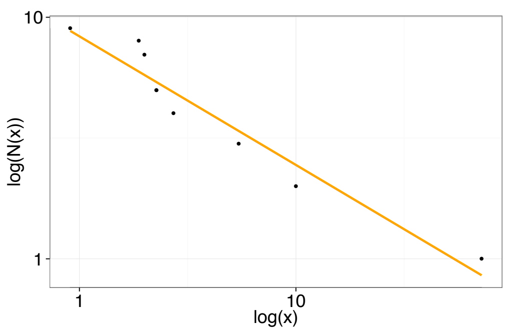
Figure 4.3. Submersible seizures between 1993–2013. The number of seized fully- and semisubmersibles has changed substantially since 1993. Between 2005–2009, the number of seizures grew significantly, but has declined since. This indicates that DTOs are becoming more proficient in the use of submersibles to evade detection and interdiction by the counter-trafficking SoS. Data are from Ramirez and Bunker [78].

Ramirez and Bunker [78] describe the evolution of DTO fully- and semi-submersibles over the last two decades. Figure 4.3 shows the number of seizures reported by Ramirez and Bunker. The number of submersible seizures has declined in recent years. Ramirez and Bunker indicate that newer technologies employed by DTOs to evade or outrun current detection and interdiction technologies are driving the decline, rather than reduced use of these types of smuggling vessels. This means that DTOs have evolved to new forms of trafficking vessels in response to the capabilities employed by the counter-trafficking SoS. This exemplifies the idea of coevolution between the counter-trafficking SoS and the environment.

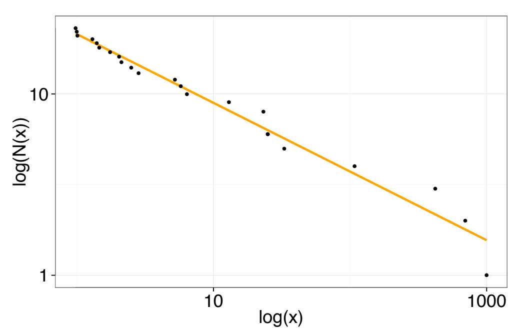
The counter-trafficking mission involves the implementation and operation of surveillance technologies, command and control nodes and interdiction assets. Similarly, drug trafficking organizations use transportation methods that have varying cargo capacities, range and speed of travel. The interactions between the counter-trafficking SoS and drug traffickers result in dynamics influencing the number of drug interdictions. These dynamics can be viewed as a population of interdictions that changes over time based on probabilities of detection and interdiction, the frequency of smuggling events, and the effectiveness of counter-trafficking operations. These dynamics exhibit complex system behavior. The seizure quantity of trafficked drugs approximates the power law distribution. Figure 4.4 displays the power law distributions for the quantity of cocaine seized for each of the three-year time periods between 1998–2012 (note the differences in x and y scales). The data for this figure come from the UNODC dataset. Within each period, x corresponds to the drug seizure quantity for each reported interdiction. $N(x)$ is the number of seizures that exceed a given quantity x. The logarithms of these values are plotted to represent the power law relationship between x and $N(x)$. Figure 4.4e corresponds to years 2010–2012, and indicates that something about the relationship between the counter-trafficking SoS and smuggler activities changed. This data no longer seem to follow the power law distribution. There are several more seizures of much greater quantities during this time period. There may be several factors that influence this relationship, but a change in the environment is likely an important variable.



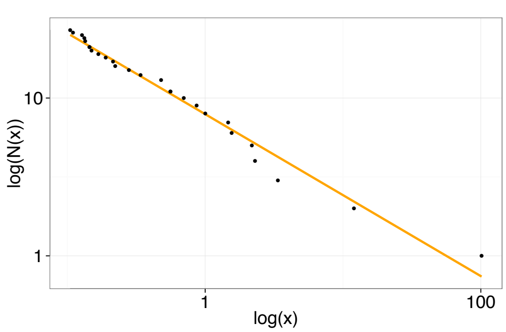
(a) 1998–2000



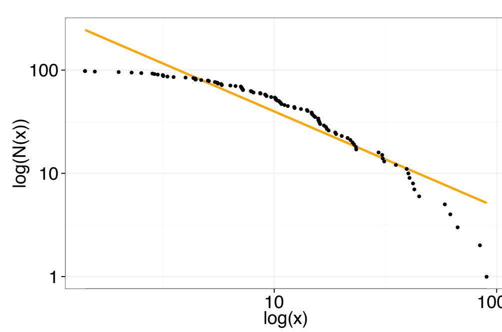
(b) 2001–2003



(c) 2004–2006



(d) 2007–2009



(e) 2010–2012

Figure 4.4. Power law distributions observed in cocaine seizures from the UNODC drug interdiction data. Here, x corresponds to the seizure quantity, and $N(x)$ is the number of interdictions that exceed x. Data from the most recent time period, 2010–2012, indicate that the SoS and the environment have changed to influence the frequency and size of interdictions.

5. ASSESSING COEVOLUTIONARY SOS META-ARCHITECTURES

Systems-of-systems do not operate in isolation. The complexities that arise result from interdependencies between the SoS and its environment. Different modeling approaches are used to answer different kinds of questions about the performance of the system. One challenge in systems engineering is enabling model interoperability so that information from one model can be used by another. DIS and HLA are formal modeling structures that have been used successfully to answer questions using modeling and simulation across multiple scales. These tools assist the exploration of future operating scenarios and estimation of candidate system performance against varied adversary capabilities.

The concept of coevolution defines the behavior that results when systems and their environment each adapt to changes in the other. The system and the environment can influence the architecture of each other. This means that environmental variables, which the system and its stakeholders may have no direct influence over, can affect the performance and future architecture of the SoS. This idea of coevolution and its impact on the SoS is illustrated in Figure 5.1, a conceptual model of the relationships between a system, the system architecture and the environment.

Two challenges for understanding how coevolution affects modern complex systems are:

- Understanding the Impact of Changes in the Operating Environment: since SoS do not operate in isolation, changes in the environment affect the performance of constituent systems, resulting in SoS level performance changes. These changes can increase or decrease the SoS performance. New constituent system alternatives or new capabilities must be integrated in the SoS to address performance gaps.

- Evaluating SoS meta-architecture alternatives: the interfaces between each SoS constituent system may be well defined in terms of physical and communication in-

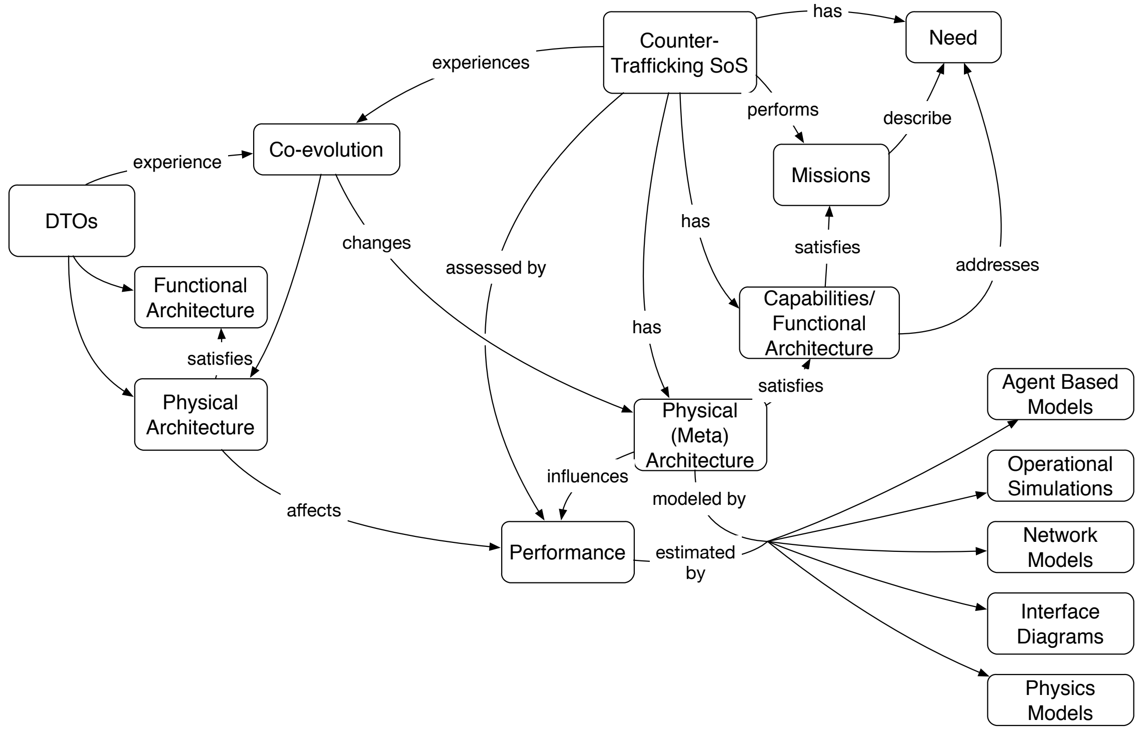


Figure 5.1. Conceptual model of the coevolutionary counter-trafficking SoS. Interactions between the SoS and the environment (DTOs) result in adaptations to the capabilities necessary for the SoS to gain a performance advantage. There are several methods that can be used to model different aspects of these complex interactions.

terfaces. However, understanding the interdependencies between the systems that comprise the SoS, especially in regards to changes in performance, is much less understood. In essence, these interdependencies can exhibit force multiplier effects, where systems that provide similar capabilities work in tandem to generate an improved overall performing architecture.

This research demonstrates a method to assess the the impacts of a changing operating environment on constituent system performance. The impacts are used to generate conceptual architectures that use new constituent systems to address capability gaps. Using this new meta-architecture, an agent based model is developed to explore the relationship between SoS meta-architecture and SoS performance under three scenarios.

In the architecture evaluation for a single system, trade studies compare several alternatives in order to choose a single best alternative in terms of overall system performance expressed through the objectives hierarchy. For an acknowledged SoS, multiple alternatives are required due to other mission priorities of the constituent systems. For the counter-trafficking SoS, USCG surveillance aircraft also perform search and rescue missions, which are likley a greater priority when loss of life or property are at stake. Similarly, USN vessels provide national level defense capabilities which impacts their availability. Both of these systems are equipped with technologies and equipment necessary for other missions.

The SoS architecture analysis considers differences among constituent systems to optimize the SoS meta-architecture. This meta-archtiecture must include constituent systems that perform well against the full set of smuggling vessels. The SoS must adapt to smuggling vessel changes, including new vessels and alternative routes, in order to maintain a performance advantage. This research demonstrates an approach for assessing a complex, acknowledged SoS that experiences this behavior. The following sections describe this approach and apply it to the counter-trafficking SoS.

1. Define SoS objectives and measures in the fundamental objectives hierarchy

2. Apply the trade study framework using swing weights to constituent systems. More than one constituent system that provides the same capability may be selected in the SoS meta-architecture.

3. Demonstrate the impact of coevolution by updating the SoS objective values as a result of changes in the environment. Re-evaluate SoS performance as a result of these changes.

4. Generate creative alternatives to address capability gaps of the existing SoS meta-architecture in the new environment

5. Develop and execute an agent based model to simulate the SoS in the operating environment

6. Perform experiments to compare performance of the original, impacted and adapted SoS architecture in varied scenarios

7. Construct statistical models to develop a representative model of the architecture and performance in uncertain future environments

8. Communicate tradeoffs to stakeholders and decision makers through depictions of trades associated with alternative SoS meta-architectures to inform research, development and acquisition decisions

5.1. OBJECTIVES AND MEASURES

System objectives and measures (values) are typically elicited from decision makers during formal systems engineering activities: stakeholder analysis, requirements definition and requirements analysis. In general, systems engineering trade studies balance multiple competing objectives. These objectives often involve affordability, performance, schedule (for acquisitions) and adaptability considerations.

For this research, the SoS-level objectives are derived from the stakeholder analysis and capabilities assessment in Chapter 4. The objectives and measures for the counter-trafficking SoS are presented in the Fundamental Objectives Hierarchy depicted in Figure 5.2. The description of each objective and measure follows.

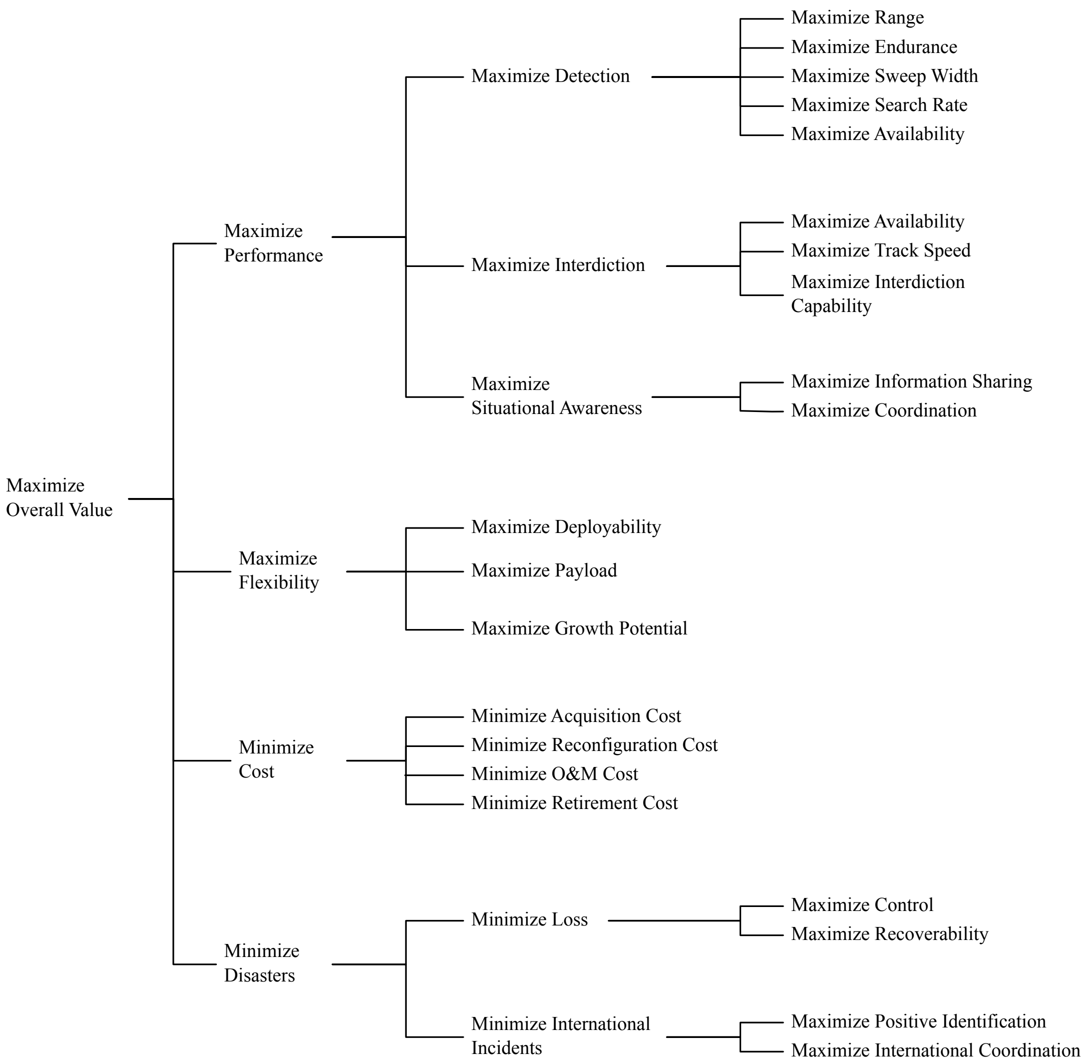


Figure 5.2. SoS fundamental objectives hierarchy. The fundamental objectives are those attributes most important to stakeholders about how the system performs its essential functions.

- Maximize Detection: This objective maximizes the ability of the alternative to detect the range of targets present in the environment. The measures for this objective are:
  - Maximize Range: the range that a surveillance craft can navigate.
  - Maximize Endurance: the amount of time that a surveillance craft can operate before returning to station.
  - Maximize Sweep Width: a measure of search effectiveness.

  - Maximize Search Effort: the amount of search area covered during a single dedicated search.
  - Maximize Availability: availability of the detection asset for use in the counter-trafficking mission.

- Maximize Interdiction: This objective maximizes the ability of the alternative to interdict the range of targets present in the environment. The measures for the Maximize Interdiction objective are:

  - Maximize Availability: the availability of the interdiction asset for the counter-trafficking mission.
  - Maximize Track Speed: the speed at which the interdiction asset can navigate to the target.
  - Maximize Interdiction Capability: the ability of the asset to successfully interdict the target. Some alternatives may not be equipped to physically interdict certain types of targets.

- Maximize Situational Awareness: This objective maximizes the ability of the information sharing alternative to inform connected systems about the environment. The measures for this objective are:

  - Maximize Information Sharing: information dissemination for relevant information, provided in a timely manner.
  - Maximize Coordination: coordinate resources to respond to available information.

- Maximize Flexibility: This objective maximizes the ability of the alternative to operate in uncertain operating environments, which improves the ability of these systems to adapt to future changes. The measure for the Maximize Flexibility objective are:

- Maximize Deployability: deployability corresponds to the ability of the surveillance or interdiction alternative to be deployed within the search area.
- Maximize Payload: the capacity of the alternative to support supplementary or complementary constituent systems. Larger payloads offer additional capacity for future systems or capabilities.
- Maximize Growth Potential: ability of the system to continue to support this mission while improving relevant capabilities. Technologies near the end of expected useful life generally will not have the growth potential that newer technologies do.

• Minimize Cost: Minimize the costs associated with the system. Acquisition cost is typically included in a trade study of competing alternatives. Because the systems employed in this trade study are used for other missions (e.g. USCG search and rescue, USN patrols), only operations costs are considered. The systems employed for the counter-trafficking mission, whether detection or interdiction, also perform other maritime security missions since they are part of the counter-trafficking acknolwedged SoS. Acquisition cost is not included because the constituent systems are used for other missions (e.g. USCG search and rescue, USN patrols). The measures for this objective are:

  - Minimize Reconfiguration Cost: the cost required to reconfigure the alternative with technologies that support new, or improve existing, capabilities.
  - Minimize O&M Costs: the cost to operate and maintain the system. These figures are not widely available, and are estimated for a six month planning horizon and 500 operating hours.

• Minimize Loss: Losses can result from crashed or abandoned craft, and provide opportunities for sensitive technologies to fall into the wrong hands. Avoiding these losses, and recovering lost capabilities, are important for this SoS. The measures for the Minimize Loss objective are:

  - Maximize Control: direct control of a surveillance or interdiction asset (e.g. piloted craft) offers more control than indirectly controlled craft (e.g. autonomous systems).
  - Maximize Recoverability: the ability to recover the capability if lost. In general, smaller assets with readily available technologies are easier to replace than custom or tailored systems.

- Minimize International Incidents: Disrupting routine traffic or violations of national maritime or air boundaries can cause incidents which adversely impact the overall counter-trafficking mission. The measures for this objective include:
  - Maximize Positive Identification: disruptions to legitimate commercial and recreational traffic reduce the tolerance for poorly performing counter-trafficking alternatives.
  - Maximize International Cooperation: in order to support efforts in certain areas within the Transit Zone, coordination and cooperation with international partners is required.

5.2. TRADE STUDY

Authoritative data to support precise values for each of the performance attributes of the counter-trafficking SoS are not all openly available. This research uses estimates for certain objectives and measures, and are noted in the detailed description of the objectives and measures above.

Ground truth for the mix of smuggling vessels used by DTOs are also not openly available. The initial meta-architecture analysis uses the mix of smuggling vessels from

2004 in Table 2.2. In 2004, 43% of seized smuggling vessels were fishing boats and 57% were go-fast boats. This is important since the counter-trafficking SoS detection and interdiction capabilities depend on the types of smuggling vessels used by DTOs.

5.2.1. Value Functions. The purpose of creating value functions is to transform measure space to value space, where stakeholder value becomes the decision criteria against which alternatives are assessed. Each system objective must have a corresonding value function in order to be used in this analysis. The value functions developed in this research are approximations since stakeholders were not involved in this research.

Surveillance alternatives satisfy the surveillance capability for the counter-trafficking SoS. The value functions for surveillance constituent system alternatives are shown in Figure A.1. These functions use linear or sigmoid functions to transform the rating for each measure to stakeholder value. All value function ranges are [0,1]. Operating costs are estimated for the purposes of this analysis. The measure ratings for each surveillance alternative is provided in Table 5.1.

Table 5.1. Measure ratings for surveillance alternatives in initial environment.

Measures	C-27J	HC-130J	P-3	MQ-9	Scaneagle	RQ-4
Range (nm)	2675	5000	2380	675	809	12300
Endurance (hours)	12	14	16	24	24	34
Availability (percent)	33	25	15	20	95	65
Deployability (index)	3	3	2	4	10	6
Payload (index)	8	8	9	4	1	3
Growth Potential (index)	4	4	3	6	6	5
Reconfiguration Cost (index)	7	8	7	8	6	4
O&M Cost (FY14 $/hr)	10,000	12,000	8,000	4,500	1,000	3,500
Control (index)	10	10	10	4	5	4
Recoverability (index)	3	2	3	5	9	6
Identification (percent)	(See	Table A.1)				

Interdiction alternatives satisfy the interdiction capability for the counter-trafficking SoS. The interdiction alternative value functions are shown in Figure A.2. These functions use linear or sigmoid functions to transform the rating for each measure to stakeholder value. All value function ranges are [0,1].

The set of alternatives and corresponding rating for each measure is included in Table 5.2. The interdiction system alternative performance differs between the types of smuggling vessels. The interdiction capability, b_{ij}, is a relative score of the ability of the interdiction system i to interdicte smuggling vessel type j. For all combined smuggling vessels in the environment,

$$c_i = \sum_{i,j} p_j b_{ij} \tag{5.1}$$

where p_j is the fraction of total trafficking events using smuggling vessel type j, c_{ij} is the capability of the initial environment, and c'_{ij} is the capability in the new environment. The Interdiction Parameters calculations table is presented in Appendix A.

5.2.2. Calculating the Alternative Value. Trade studies map performance ratings from measure space to value space by accounting for the degree of importance that the measure has on the overall system and the swing, or range of value, across the available alternatives. The `alt-swing` IPython Notebook was developed and used to perform the computations for this trade study. Appendix B describes the `alt-swing` IPython Notebook and provides the Python code.

The value of alternative i for measure j is evaluated based on the value functions described in Section 5.2.1,

$$v_{ij} = f_j(m_{ij}) \tag{5.2}$$

where f_j is the value function for measure j and m_{ij} is the score of alternative i for measure j.

Table 5.2. Measure ratings for interdiction alternatives in the initial environment.

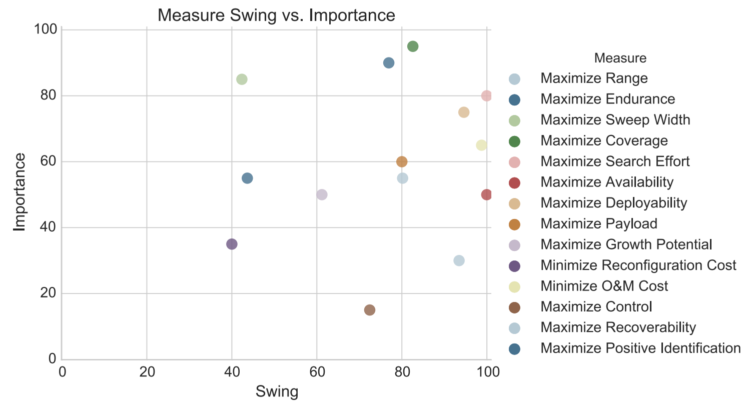
Measures	MH-60T	MH-65D	SH-60	LRI-II	OTH-IV	RB-S
Availability (percent)	50	50	30	75	80	85
Track Speed (knots)	170	160	146	38	40	45
Range (nm)	300	150	450	225	200	175
Interdiction Capability (percent)		(See	Table A.3)			
Deployability (index)	2	4	2	6	7	8
Growth Potential (index)	4	5	3	6	5	6
Reconfiguration Cost (index)	9	8	9	5	4	3
O&M Cost (FY14 $/hr	3000	2000	2500	1200	1200	1000
Control (index)	9	9	9	10	10	10
Recoverability (index)	2	3	1	7	8	9
Positive Identification (percent)	95	95	92	90	88	85

Measure swings are calculated as the range in value from the available alternatives. The minimum and maximum values are determined by the available alternatives in this trade study. The swing for each measure, s_j, is defined as:

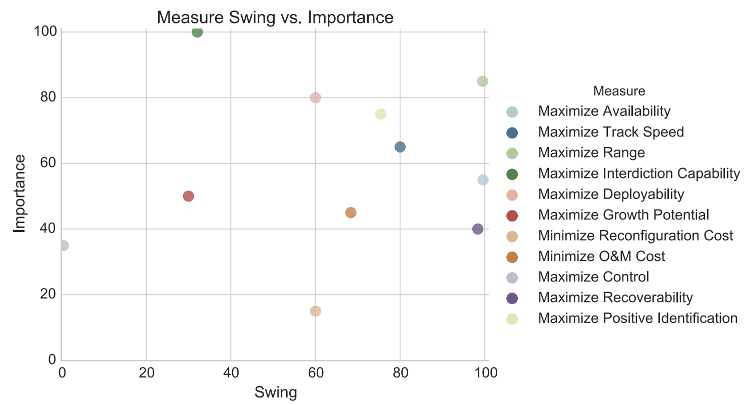
$$s_j = \max_i(v_{ij}) - \min_i(v_{ij}) \tag{5.3}$$

This trade study uses an additive value model to determine the total alternative value. This model uses both the swing weight (β) and importance weight (α), such that $\alpha + \beta = 1$, in calculating the total alternative value. Figure 5.3 provides scatterplots for the surveillance and interdiction importance and swings. The unnormalized weight for each measure is calculated as

$$w'_j = \alpha o_j + \beta s_j \tag{5.4}$$



(a) Surveillance measures.



(b) Interdiction measures.

Figure 5.3. Performance measure importance vs. swing for the initial environment.

where o_j is the measure importance. These weights are then normalized for the final measure weight, w_j, where

$$w_j = \frac{w'_j}{\sum_j w'_j}. \tag{5.5}$$

Then, the total stakeholder value for alternative x_i is

$$V(x_i) = \sum_{j=1}^{n} w_j \, v_{ij}. \tag{5.6}$$

The resulting values for each alternative provide a quantitative comparison among the alternatives considered.

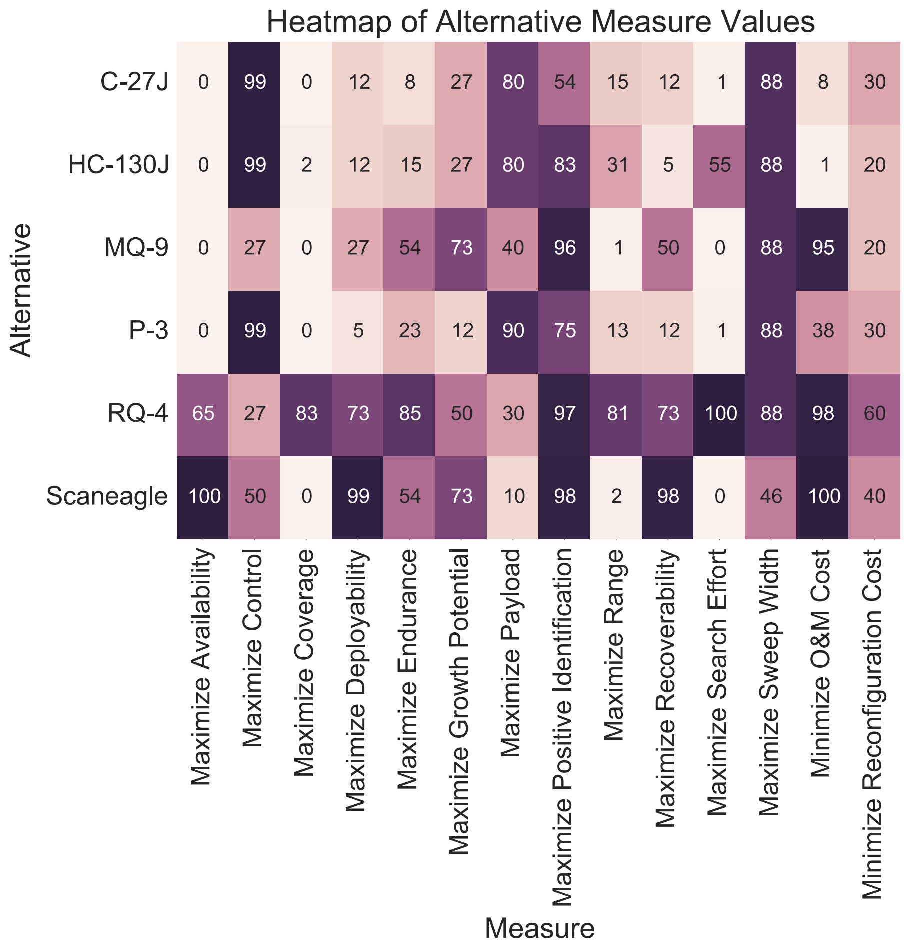


Figure 5.4. Surveillance alternatives consequences scorecard (initial environment). The consequences scorecard, or heatmap, depicts the value score for each alternative against each measure. For surveillance systems, the RQ-4 tends to outperform other alternatives in most measures. All systems tend to perform well for the Maximize Positive Identification and Maximize Sweep Width measures.

5.2.3. Results Comparison. The RQ-4 dominates surveillance alternatives in all measures except search effort and payload. The C-27J is dominated by most alternatives except for sweep width. For the interdiction alternatives, the MH-60T Jayhawk outperforms the other assets, but the set of alternatives tend to perform well in general. Figures 5.4 and 5.5 provide heatmaps of each alternative's score against each measure. This display allows straightforward comparisons between performance measures and across al-

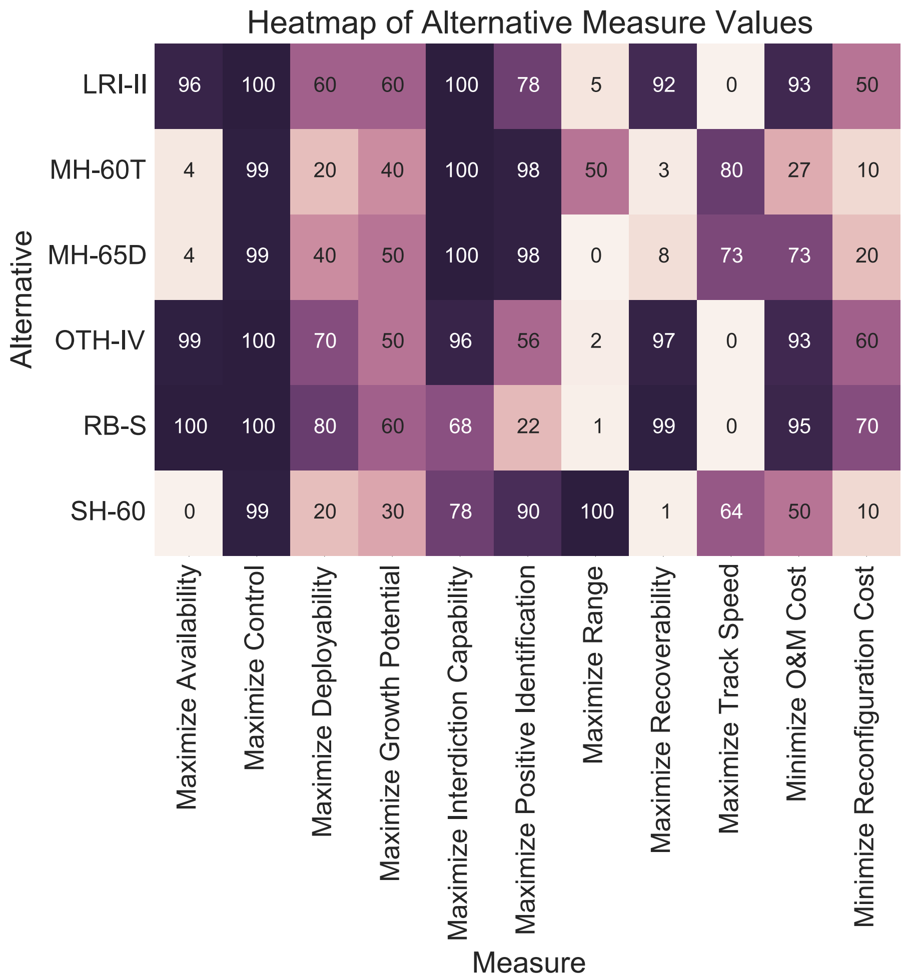


Figure 5.5. Interdiction alternatives consequences scorecard (initial environment). All of the interdiction systems tend to provide about the same value. The contribution for each alternative differs, and the LRI-II tends to perform the best among the set of alternatives. Most of the alternatives perform very well for control and positive identification. The alternatives vary the greatest in range, recoverability and track speed.

ternatives. Table 5.3 summarizes the resulting stakeholder value for each surveillance and interdiction alternative.

Multiple alternatives can participate as constituent systems within the counter-trafficking SoS. The resulting values indicate how beneficial each alternative is in terms of stakeholder value. Those constituents with higher stakeholder value can be targeted for

Table 5.3. Surveillance and Interdiction alternative values. The results of the trade study provide a value for each surveillance (left) and interdiction (right) alternative in the initial environment.

Surveillance Alternative	Value	Interdiction Alternative	Value
C-27J	27.2	LRI-II	82.7
HC-130J	31.1	MH-60T	59.9
MQ-9	32.2	MH-65D	63.9
P-3	32.5	OTH-IV	81.7
RQ-4	51.6	RB-S	78.5
Scaneagle	43.0	SH-60	61.2

more frequent or regular use within the SoS, subject to availability constraints for other missions.

5.3. SOS EVALUATION IN A NEW ENVIRONMENT

The initial SoS architecture may be used for a period of time. After DTOs understand the capabilities available to the counter-trafficking SoS, they develop new forms of smuggling vessels to mitigate performance advantages of the counter-trafficking SoS. The emergence of submersible vessels affects the detection and interdiction performance of the counter-trafficking SoS. SoS stakeholders need to know how the SoS is affected in order to make acquisition decisions for new systems or to engage stakeholders that have needed capabilities that exist for other missions. Understanding how the SoS is affected allows the right set of capabilities to be targeted for development or inclusion in the SoS. This behavior exemplifies the idea of coevolution between the SoS and the environment.

5.3.1. Changes in the Operating Environment. Because information on the smuggling vessel mix used by DTOs is not openly available, the mix of DTO smuggling vessels from 2009 in Table 2.2 is used to demonstrate the change in the operating environment. In 2009, 12% of smuggling vessels were fishing boats, 44% were go-fast boats and 44% were submersibles, an increase from 0% in 2004. For this analysis, submersibles are evenly split between semi-submersibles and fully-submersibles. Each type of smuggling vessel has different ratings for the counter-trafficking detection and interdiction capabilities.

Evaluating SoS performance in terms of stakeholder value enables the comparison of the new environment with the existing counter-trafficking SoS meta-architecture. Other considerations such as anticipated time for the DTO to overcome an alternative's detection or interdiction capability can also influence the alternative performance. In essence the coevolutionary behavior exhibited by these dueling architectures allows one architecture to hold an operational advantage for a limited amount of time.

Understanding how coevolution affects the SoS is important in order to identify candidate solutions or improvements that address the changes in the environment. By evaluating the performance changes of each objective and measure within the objectives hierarchy allows the key capabilities impacted by environmental changes. In this case, the environmental changes consists of new DTO vessels that are challenging to detect and interdict. Identifying key capabilities that improve the SoS performance focuses the solution space on those capabilities that directly improve SoS performance.

5.3.2. Impact on SoS Value. The DTO smuggling vessel change reflects a change in the trafficking meta-architecture. This essentially changes the environment that the counter-trafficking system is operating in. Figure 5.6 illustrates the change in constituent system values after the smuggling vessel changes described above. The smuggling vessel change decreased the stakeholder value of both the surveillance and interdiction alternatives. Only the Maximize Interdiction objective was influenced by the change to the DTO architecture change.

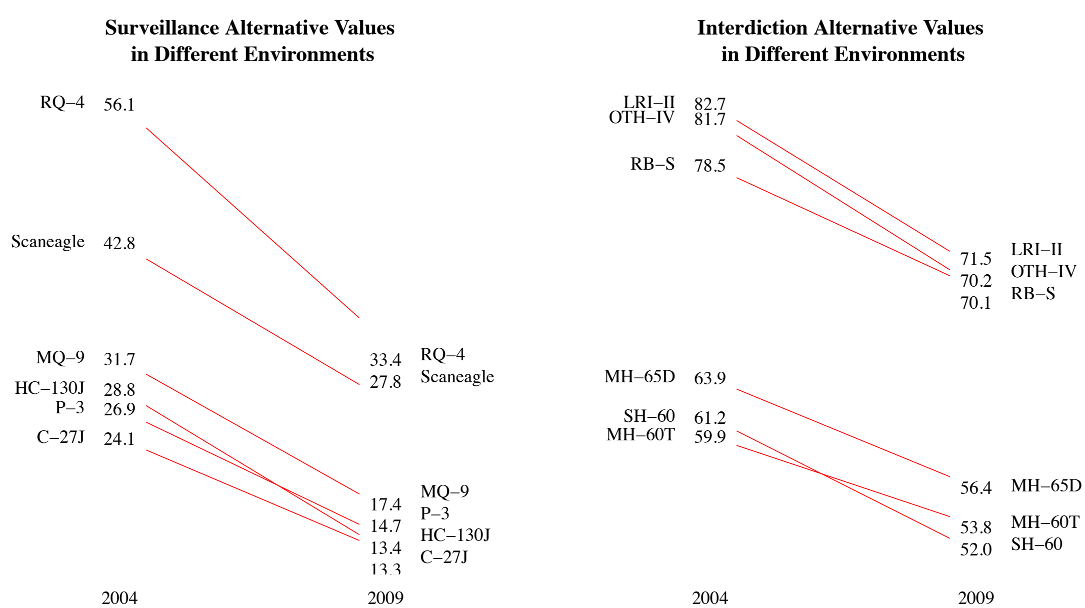
A comparison of the subset of objectives that were affected by this evolution in smuggling vessels is depicted in Figure 5.7. Almost all measures are affected for surveillance systems, and the interdiction capability of all interdiction alternatives was reduced.

Since the impact is on specific objectives (Detection and Limit Impacts for surveillance craft, and Interdiction for interdiction systems), comparing the initial SoS with the

SoS performance in the new environment for only those objectives affected allows decision makers to target new alternatives to increase the SoS performance and value delivery to SoS stakeholders.

5.4. COEVOLUTION AND ALTERNATIVE GENERATION

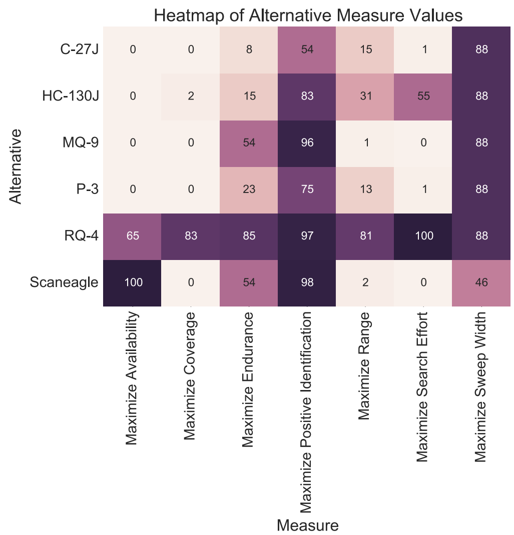
Section 5.3.2 demonstrated that the use of submersible vessels by DTOs significantly reduces the stakeholder value delivered by the existing counter-trafficking SoS. These smuggling vessels allow DTOs to evade current detection capabilities and avoid interdiction. Further, these vessels can stay submerged for over a week, requiring that counter-trafficking assets maintain contact, through electronic or other means, throughout



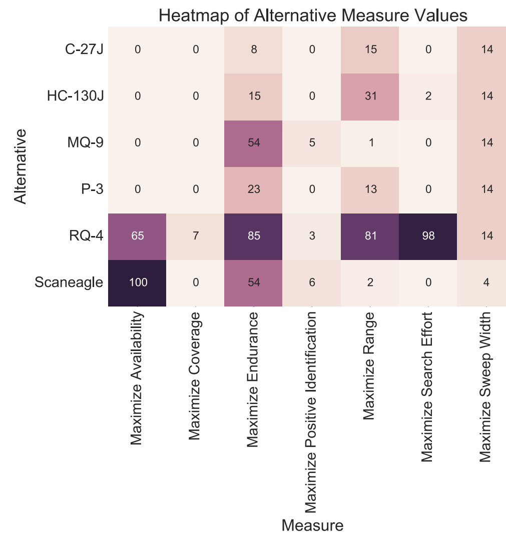
(a) Surveillance alternatives.

(b) Interdiction alternatives.

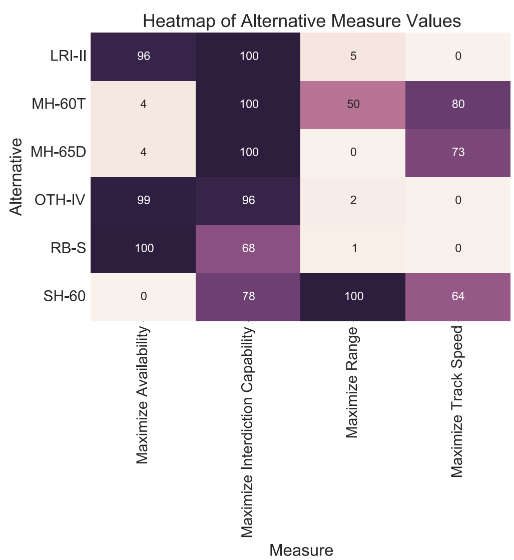
Figure 5.6. Alternative values before and after smuggling vessel evolution. The slopegraphs show the steep changes in surveillance (a.) and interdiction (b.) alternative values as a result of the different DTO smuggling environments from 2004 and 2009. In this instance, most alternatives retain their relative position with respect to the overall set of alternatives, with the exception of the SH-60 and the HC-130J.



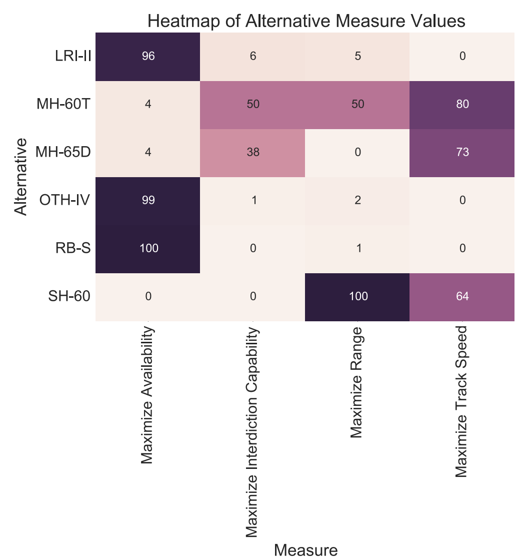
(a) Initial surveillance alternatives scorecard for detection and impacted objectives.



(b) New surveillance alternatives scorecard for detection and impacted objectives.



(c) Initial interdiction alternative interdiction and impacted objectives.



(d) New interdiction alternatives value for interdiction and impacted objectives.

Figure 5.7. Impact of smuggling vessel evolution on surveillance and interdiction constituents. The evolution of smuggling vessels results in degraded performance for all surveillance and interdiction constituent system alternatives for the counter-trafficking SoS.

this time. Maintaining contact requires the counter-trafficking system to use limited resources on a single effort for extended periods of time, reducing the availability of these systems for other missions. The reduced performance of the counter-trafficking SoS forces stakeholders to seek new capabilities in order to improve SoS performance in this new environment.

5.4.1. New SoS Capabilities. Section 5.2.3 showed the impact of the evolutionary behavior on the part of the DTOs by implementing new smuggling vessels. This change primarily impacted the Maximize Detection and Maximize Interdiction objectives. Other changes to the environment, such as new technologies that disrupt the control of autonomous aircraft, or new policies that change the operations and maintenance costs of SoS alternatives, would impact other SoS objectives. New alternatives should be developed that focus on the adversely impacted objectives. Creating a new SoS meta-architecture that increases SoS performance and stakeholder value requires that new alternatives focus on Maximizing Detection and Maximizing Interdiction in the new environment.

By design, submersibles have a low detection profile for surface and air sensors. By traveling underwater, these vessels evade radar and EO/IR sensors. However, they have unique sonar signatures. This requires the ability to detect the vessels while moving underwater, and coordinate with interdiction assets to seize the vessel. Current approaches by DTOs require that submersibles surface for a period of time in order to offload the illicit cargo for final transportation to land, where the drugs are distributed. Current interdiction alternatives are able to seize the submersible while it is surfaced. In addition, submersibles have limited operating ranges and cannot stay submerged indefinitely. Surfacing provides an opportunity to interdict the submersible, but requires that interdiction assets be aware of the location of the submersible when it does so.

Existing systems that are not part of the initial SoS meta-architecture could support improved submersible detection. These systems are currently used for other purposes. The USN has a submarine fleet that is used for defense missions. Submarines are capable of tracking targets underwater for extended periods of time. Submarines have unique capabilities and mission sets, but are cost-prohibitive to use for the counter-trafficking mission. The USN also uses sonobuoys for detecting underwater threats. Bailey [79] describes a buoy developed specifically for the counter-trafficking system. This buoy is designed to detect go-fast boats and operates at depths of 50-600 feet with a 6–12 month operational life. This buoy uses satellite communications links to report detection events within a 5 nm radius, but is not designed to track the target. With additional development, this type of sonobuoy may be designed to detect submersibles using either active or passive sonar and provide target tracks for a detected vessel.

Another alternative is the use of autonomous systems. Autonomous system use for land, sea and air missions is growing rapidly. Although these technologies are in the early stages of development, many are being matured to meet operational requirements and environmental challenges. Such systems could improve SoS performance for the counter-drug mission. Unmanned underwater vessels (UUVs) operate submerged and can be operate autonomously or semi-autonomously. The Massachusetts Institute of Technology has developed an UUV that is intended for use in port security operations [80]. This small vessel navigates autonomously to maritime targets, attaches to the target's hull, and uses an ultrasound sensor to scan the target's hull to detect hollow compartments storing contraband. While currently in development and prototype testing stages, a similar type of UUV with improved range and navigation could be used in the counter-trafficking mission. Zorn [55] also identifies a cutter-based UUV for use in maritime security and maritime domain awareness missions. These vessels have a range of around 80 nm, and could be delivered by cutter or interdiction assets. In the future, these types of vessels may be delivered by other systems such as surveillance aircraft. The UUVs described by Hardesty and Zorn are considered for the future notional counter-trafficking architecture.

5.4.2. Future SoS Alternatives. The following scenarios are considered for evolving the SoS meta-architecture. These scenarios, as a response to DTO adaptations in smuggling vessels, completes the coevolution example between the counter-trafficking SoS and the environment. This notional future concept is a multi-stage delivery, tracking and interdiction concept using UAVs, sonobuoys and UUVs in the counter-trafficking SoS meta-architecture.

- Cutter-based UUV Future Alternative: The first stage delivery system (e.g. NSC) carries a secondary stage delivery system (e.g. UAV) and the UUV autonomous system (e.g. an enhanced version of the one developed by MIT). Upon a target detection by an array of deployed sonobuoys, the secondary stage delivery system (e.g. MH-65D) navigates to the proximal detection location, and deploys the UUV. The UUV navigates to the target using on-board navigation systems and communications with the sonobuoy array. The UUV attaches to the target vessel once it is intercepted. The target is scanned to classify the vessel as threat or benign, and results are communicated to the C2 system. If the target is classified as a threat, the UUV tracks the target, and reports the target state while maintaining contact with the target. When the target surfaces, interdiction assets are coordinated to interdict the DTO vessel once it is no longer submerged. This notional operational concept omits the need for high-value counter-trafficking assets to maintain single-target detection and tracking for extended periods of time, and supports coordination of interdiction assets so that they are available at the right time and right place.

- UAV-based UUV Future Alternative: In this concept, a UAV, equipped with the same UUV detector, performs multiple missions simultaneously. The primary mission, ISR, is conducted as a surveillance platform. The secondary mission is to serve as the first stage delivery vehicle of the UUV. Suppose, as before, that a sonobuoy array target detection occurs. In this concept, the UAV navigates to within close proximity to the detection, and delivers the UUV to that location. The advantage in this concept is that the UAV can navigate to very close proximity to the target location, whereas in the first concept, delivery is limited to the range of the MH-60T. Once submerged, the

UUV navigates to the location of the target, which reduces the required range of the UUV (assuming it attaches to the target and uses the target propulsion for maneuvre). The rest of the scenario is as above, where communication to the counter-trafficking C2 system is maintained, and interdiction assets are coordinated once the target is no longer submerged. This concept is depicted graphically in Figure 5.8. Figure 5.9 depicts the OV-1 of this SoS meta-architecture alternative.

5.4.3. Assessing New SoS Meta-Architectures. The new SoS meta-architectures share interdependencies between constituent systems. In the new environment, interdiction assets rely on the detection and tracking of submersibles in order to function. Without interdiction systems capable of interdicting submersibles, the UUV has no significant role. Agent based modeling allows these interdependencies to be modeled and the SoS to be simulated in various environments. Chapter 6 describes the agent based model used for this analysis.

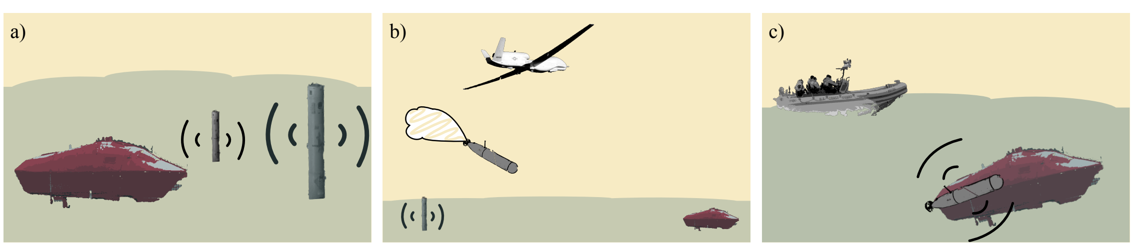


Figure 5.8. Future SoS meta-architecture evolution concept. An example operation with a new SoS architecture: a) Sonobuoys are deployed to detect, track and monitor DTO submersible movements. These systems communicate detection events and target tracks to the ISR system. b) Upon target detection and identification, an unmanned underwater vessel (UUV) is deployed in the vicinity of the target by a UAV. The UUV navigates to target using target tracks and an on-board tracking system. c) The UUV attaches to the target's hull, communicating location and target state to the ISR system. The target's track is monitored and interdiction assets are deployed when the submsersible surfaces.

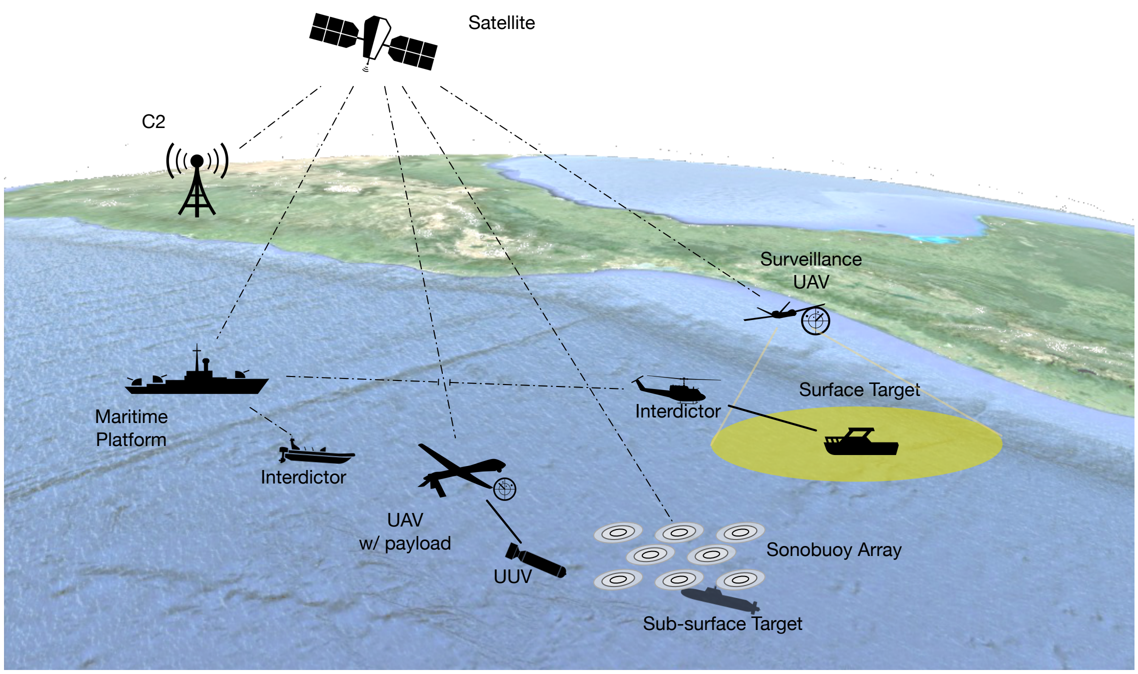


Figure 5.9. OV-1 for a future UAV-based UUV meta-architecture alternative. This alternative uses new technologies to address degraded SoS performance resulting from the new environment. Existing surveillance and interdiction alternatives are modified to enable communications and payload delivery (UUV) against new smuggling vessels (submersibles).

6. AGENT BASED MODEL FOR THE SOS META-ARCHITECTURE

Agent based modeling allows the exploration of sytem behavior by defining how agents interact with other agents and the environment. The ABM paradigm aids SoS meta-architecture performance assessment, particularly when system interactions are known generally, strict formulations describing system dynamics are unavailable, and individual behavior of constituent systems affects SoS performance as a whole. The counter-trafficking SoS can be modeled using ABM by describing agent rules and behaviors and assessing performance by varying the types of constituent systems that participate in the SoS.

6.1. STUDY QUESTIONS

The counter-trafficking SoS meta-architectures described in Chapter 5 are evaluated using ABM. Agents represent constituent systems, smuggling vessel targets and commercial vessels (benign traffic). Each agent has properties and rules that govern its behavior. ABM enables exploration of candidate SoS meta-architectures by varying the types of constituent systems operating in different environments. Environments are constructed by varying the frequency of DTO smuggling vessel use and amount of commercial boat traffic. Meta-architecture performance is based on the percent of each smuggling vessel type detected, the percent of each type interdicted, the percent of trafficked drugs interdicted, and the operating cost of the meta-architecture. The goal of this simulation is to provide insight into the SoS meta-architectures that perform best in varied operating environments.

The agent based model for this research was built using AnyLogic 7.2 Educational Version software. The agents, properties and behavior rules are described in Section 6.2. Figure 6.1 provides an animation snapshot of agent interactions during simulation runtime.

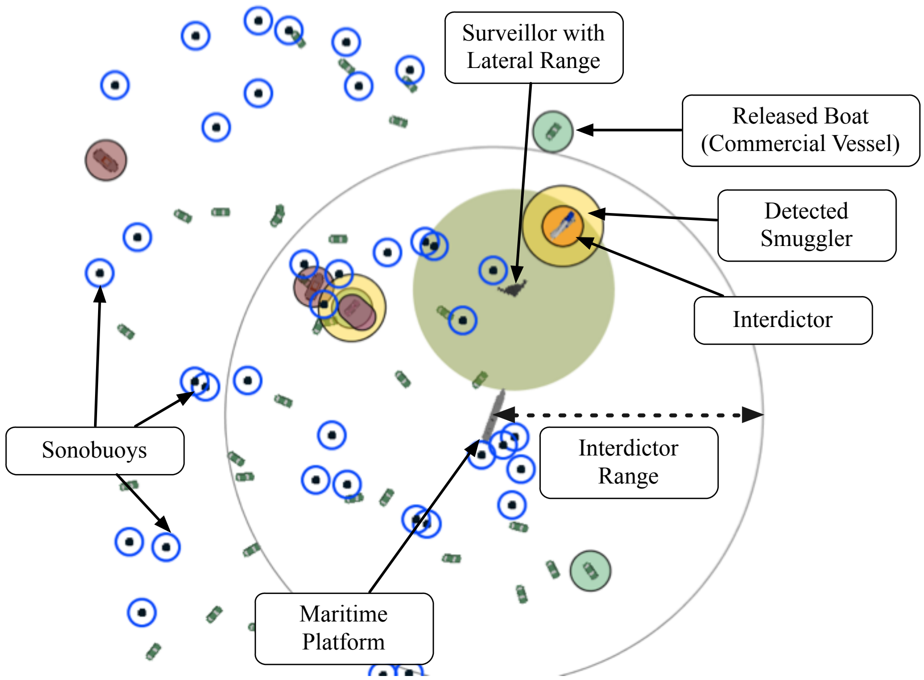


Figure 6.1. Agent based model runtime animation. Each agent in the ABM is identified, along with a few corresponding attributes, such as interdictor range and surveillance lateral range.

6.2. AGENTS, PROPERTIES AND BEHAVIOR RULES

The following sections describe the agent properties and statecharts for the ABM. Some agents are subclassed to define other agents, such as smuggling vessels that extend the Boats agent and helicopters that extend the Interdictor agent. Agent subclasses inherit the properties, states and behavior logic of the parent agent, but are assigned properties specific to the agent type that are assigned when the agent is created at runtime.

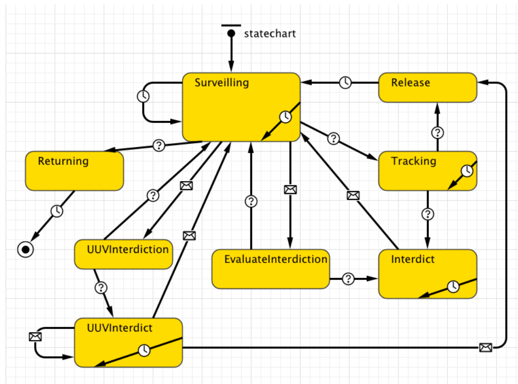
In this model, simple rules govern agent behavior. The logic for each agent is controlled by statecharts. Figure 6.2 depicts the statecharts for each of the primary agents in the ABM. Commercial and smuggling vessels are randomly assigned a starting location along a boundary. Each vessel chooses a random waypoint that is closer to the final destination at the opposite boundary. Each successive move is in the general direction of the vessel's final destination.

Surveillance aircraft are assigned a random starting location opposite the start of vessels. These agents fly a random search path across the simulation space. Waypoints are uniformly distributed across the simulation environment. If a smuggling vessel is within radar range, a detection occurs based on the lateral range curve. Upon detection, the craft flies to the target and maintains target visibility for a specified period of time. If a Maritime Platform vessel with an interdiction asset are available, an interdiction occurs, else the target is released. The simulation environment consists of a 1,000 nm x 1,000 nm boundary and runs for a continuous 4,000 hour simulated time period (approximately six months).

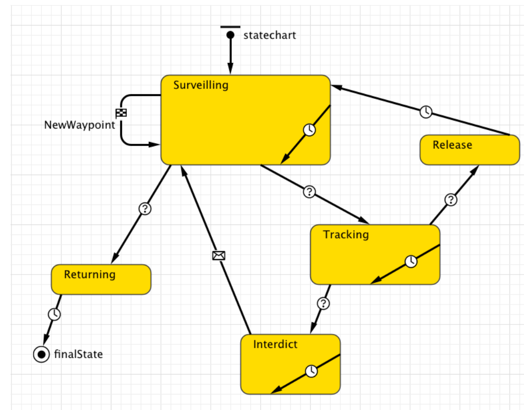
6.2.1. Maritime Platform Agent. The Platform agent models systems that can deliver interdictors and small surveillance systems. These include the National Security Cutter operated by the USCG, and the Oliver Hazard Perry Class Frigate operated by the USN. The Maritime Platform agent properties are described in Table 6.1. Figure 6.2a presents the Maritime Platform statechart, and agent behavior rules are described in Algorithm 1.

Table 6.1. Maritime platform agent properties.

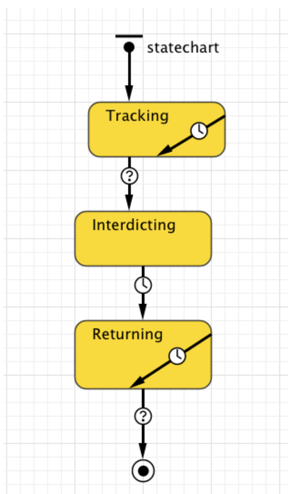
Name	Description	Units
cruiseSpeed	agent speed during normal operation	knots
maxSpeed	agent speed when reacting to an event	knots
range	distance the agent can travel	nm
endurance	time the agent is able to be continuously deployed	hours
lateralRange	radar detection range	nm
qtyInterdictors	array of the number of interdiction agents the platform can carry	count
qtySurveillors	array of the number of surveillance craft the agent can carry	count
location	current location of agent	lat/lon
nextWaypoint	the waypoint the agent transits to unless reacting to an event	lat/lon



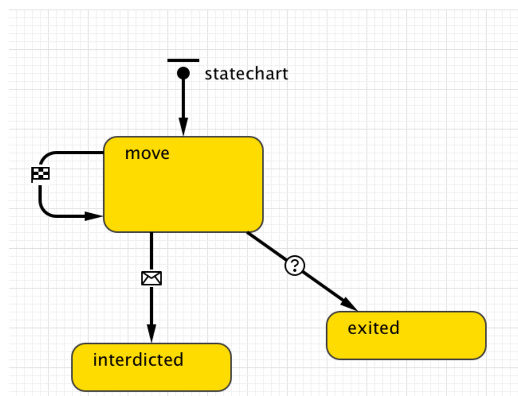
(a) The Platform statechart defines the conditional logic controlling agent states.



(b) The Surveillance statechart defines the conditional logic controlling Surveillance and subclassed agent states.



(c) The Interdictor agent statechart defines the conditional logic controlling Interdictor and subclassed agent states.



(d) The Boat statechart defines the conditional logic controlling Boat and subclassed smuggling vessel agent states.

Figure 6.2. Agent statecharts for the agent based model. Logic within each state controls agent behavior. The SoS tends to rely on the Maritime Platform agent to resolve detection and interdiction requests.

6.2.2. Surveillance Agent. The Surveillance agent models both manned and unmanned aerial surveillance systems. The Surveillance agent properties are described in Table 6.2. Figure 6.2b presents the Surveillance agent statechart, and agent logic is described in Algorithm 2.

Algorithm 1 Platform agent behavior.

```
initalize agent and set properties
goto surveilling
while Surveilling do
    navigate to random waypoint
    search for targets
    if ( target within range ) then
        goto Tracking
    end if
    if ( receive interdiction request ) then
        goto EvaluateInterdiction
    end if
    if ( reached endurance ) then
        return to port; goto initialize
    end if
end while
while Tracking do
    if ( interdiction asset available before timeout ) then
        launch available interdictor agent
        goto Interdict
    else release target; goto Surveilling
    end if
end while
while EvaluateInterdiction do
    if ( interdiction asset available and within range ) then
        goto Interdict
    else goto Surveilling
    end if
end while
while Interdict do
    navigate to target
    if ( recovered interdiction asset ) then
        goto Surveilling
    end if
end while
```

6.2.3. Interdictor Agent. The Interdictor agent represents interdiction systems that seize smuggling vessels. Interdictors include helicopters and interceptor boats. These are implemented as a subclass of the Interdictor agent. The Interdictor agent properties

Table 6.2. Surveillance agent properties.

Name	Description	Units
cruiseSpeed	agent speed during normal operation	knots
maxSpeed	agent speed when reacting to an event	knots
range	distance the agent can travel	nm
endurance	time the agent is able to be continuously deployed	hours
lateralRange	detection range of radar sensors	nm
payload	carrying capacity for surveillance agents	index
path	sequence of waypoints the agent navigates	lat/lon array
currentLocation	current location of agent	lat/lon
nextWaypoint	the waypoint the agent transits to unless reacting to an event	lat/lon

described in Table 6.3 are derived from Table 4.5. Figure 6.2c presents the Interdictor statechart and agent behavior is described in Algorithm 3.

6.2.4. Boat Agent. Boat agents represent commercial boats (benign targets), and smuggling boats. Each smuggling vessel type is a subclass of the Boat agent. This allows properties specific to fishing boats, go-fast boats and submersibles to be assigned at agent creation. Differences between these vessels, including speed, radar visibility and sonar visibility, affect Surveillance and Interdiction agent performance. Smuggling vessel properties are based on Table 4.2. Smuggler agent properties for the ABM are described in Table 6.4. Figure 6.2d presents the Boat agent statechart, and agent behavior is described in Algorithm 4. Boats move randomly in the same general lateral direction across the simulation environment and, unless interdicted, exit the simulation upon arrival to the final waypoint.

6.2.5. UUV Agent. The UUV agent models unmanned underwater vessels that are delivered by Mfaritime Platforms or Surveillance systems and detect and track submersible vessels. Once attached to the submersible, the UUV relays location information to the

Algorithm 2 Surveillance agent behavior.

```
initalize agent and set properties
goto Surveilling
while Surveilling do
    navigate to random waypoint
    search for targets
    if ( target detected ) then
        goto Tracking
    end if
    if ( reached endurance ) then
        return to port; goto initialize
    end if
end while
while Tracking do
    follow target
    if ( target resolved and interdiction asset available before timeout ) then
        request Interdictor from Platform agent
        goto Interdict
    else release target; goto Surveilling
    end if
end while
while Interdict do
    follow target
    if ( Interdictor agent releases Surveillance agent ) then
        goto Surveilling
    end if
end while
```

Maritime Platoform which launches the interdiction asset. The UUV agent has simple properties of speed and range. The agent behavior rules are described in Algorithm 5.

6.2.6. Sonobuoy Agent. The Sonobuoy agent models sonobuoys that detect submersibles and go-fast boats. Upon detection, the sonobuoy reports the target location to a Surveillance or Maritime Platform agent, depending on the architecture being modeled. The UUV agent has a single property of sonarRange which models the radius and the M-beta lateral range curve for the buoy. Sonobuoys are randomly deployed at stationary locations between 200–300 nm from the edge of simulation area nearest boat arrival lo-

Table 6.3. Interdictor agent properties.

Name	Description	Units
cruiseSpeed	agent speed during normal operation	knots
maxSpeed	agent speed when reacting to an event	knots
range	distance the agent can travel	nm
endurance	time the agent is able to be continuously deployed	hours
lateralRange	detection range of radar sensors	nm
availability	percent of time the asset is available to perform the interdiction mission	percent
seizeRange	range that a target agent must be within to be interdicted	nm
isSeizable	binary array of smuggler agents ability to be seized by the agent	array
payload	carrying capacity for sensor agents	index
currentLocation	current location of agent	lat/lon
targetLocation	location of the target agent (smuggler)	lat/lon

cations (x) and uniformly across the width of the simulation area (y). The agent behavior rules are described in Algorithm 6.

6.2.7. Main Agent. The Main agent is an artifact of the AnyLogic modeling environment. This agent defines simulation level properties, including the parameters varied across simulation trials, output files for simulation results, and the ABM presentation layer.

Algorithm 3 Interdictor agent behavior.

```
initalize agent and set properties
goto Surveilling
while Surveilling do
    navigate to random waypoint
    search for targets
    if ( target detected ) then
        goto Tracking
    end if
    if ( reached endurance ) then
        return to port; goto initialize
    end if
end while
while Tracking do
    follow target
    if ( target resolved and interdiction asset available before timeout ) then
        request Interdictor from Platform agent
        goto Interdict
    else release target; goto Surveilling
    end if
end while
while Interdict do
    follow target
    if (Interdictor agent releases Surveillance agent ) then
        goto Surveilling
    end if
end while
```

Algorithm 4 Boat agent behavior.

```
initalize agent and set properties
goto Moving
while Moving do
    navigate to random waypoint
    if ( interdicted by interdiction agent ) then
        goto Interdicted
    end if
    if ( reached final waypoint ) then
        goto Exited
    end if
end while
```

Table 6.4. Boat agent properties.

Name	Description	Units
cruiseSpeed	agent speed during normal operation	knots
maxSpeed	agent speed when reacting to an event	knots
capacity	carrying capacity for illicit drugs	index
detectability	ability of the agent to be detected; this value is $p(d)$ for the M-Beta detection model	percent
type	type of smuggling vessel: fishing vessel, go-fast boat, semisubersible, or fully-submersible	index
path	sequence of waypoints the agent navigates	lat/lon array
currentLocation	the current location of the agent	lat/lon
targetLocation	the target location of the agent	lat/lon
startingLocation	initial location of the agent	lat/lon
nextWaypoint	the waypoint the agent transits to unless reacting to an event	lat/lon
arrivalRate	rate that the agent enters the simulation	qty/hour

Algorithm 5 UUV agent behavior.

```
initalize agent and set properties
goto Moving
while Moving do
    navigate to target
    if ( reached target ) then
        stay with target
        message Maritime Platform of current location
    end if
    if (target interdicted) then
        goto Exited
    end if
end while
```

Algorithm 6 Sonobuoy agent behavior.

```
initalize agent and set properties
goto Surveilling
while Surveilling do
    if ( target detected ) then
        report location to nearest agent with UUV capability
    end if
end while
```

7. RESULTS

Chapter 5 demonstrated the impact of coevolution on the counter-trafficking constituent systems and described new systems to address the impacts of smuggling vessel use. Chapter 6 described the agent based model developed to assess multiple SoS meta-architectures using these new systems. The results of the agent based model allow exploration of these meta-architectures and their performance characteristics in different operating environments.

7.1. SIMULATION EXPERIMENTS

Table 7.1 describes the factors and levels of the simulation experiment for the counter-trafficking system. The performance characteristics used for this study include operating cost, percent of trafficking vessels detected, and percent of trafficking vessels interdicted. Each experiment sets the factors at a defined level and uses these parameters for the model settings. The performance characteristics are outputs from these settings.

The environment is represented by other variables outside of the control of the counter-trafficking SoS. For the simulation study, these variables are the percent of smuggling traffick of each vessel type: fishing vessels, go-fast boats, and submersibles.

Table 7.1. Agent based model parameter settings. The experiment parameter settings define the architecture simulated in each trial.

Variable	Minimum	Maximum	Step Size
Number of Platforms	1	2	1
Number of Sonobuoys	0	150	30
Number of Surveillance Systems	0	3	1
UUV Range	50	100	25
Interdictor Range	150	300	75

7.2. EXPLORATORY DATA ANALYSIS

Exploratory data analysis allows the behavior of the model to be better understood. Insights drawn from this analysis lead to decisions about how the model can be improved, new studies to undertake, and support decisions that affect overall SoS performance.

Cost affects most decisions, and operating cost is a driving component of total cost. Understanding the performance of the SoS based on different architectures is important. Figure 7.1 displays drug seizure performance versus operating cost for each of the simulated architectures. These architectures represent the cutter-based UUV concept from Chapter 5. Points are colored based on the fraction of smuggling vessel traffic that is conducted by submersibles. Submersibles have a strong influence on the overall performance of the SoS, where other smuggling vessels allow the SoS a greater ability to seize illicit cargo.

Another performance measure for the SoS is the percent of smuggling vessels interdicted. Figure 7.2a indicates that the number of surveillance craft (NumUAVs) and percent of smugglers using go-fast boats (PctGF) do not signifantly influence this performance measure. However, interdictor range tends to become more important as the percent of go-fast boats increases, and SoS architectures with larger range interdictors (helicopters) tend to outperform those with shorter range craft (such as interdictor boats).

The range of UUVs is another architecture characteristic simulated in the agent based model. These ranges are a result of uncertainty in the future operating capability of UUVs. Figure 7.2b shows the influence that this characteristic has on the overall drug seizure performance of the SoS. The frequency of DTO submersible use corresponds to different environments. UUV range becomes marginally more important as the frequency of submersible use grows. However, this property is far exceeded by submersible use, as seen in the negative trend across facets.

The agent based model simulated 3,888 different meta-architecture - environment scenarios. These architectures are composed by differing the numbers and features of constituent systems that comprise the SoS. The star plot, or kiviat chart, helps visualize the differences among these architectures. Figure 7.3 depicts a random sample of these architectures, ordered by increasing operating cost. Each star represents architecture features of number of maritime platforms, number of surveillance craft, and number of sonobuoys. Performance measures of percent of smugglers interdicted, percent of drugs seized and SoS operating cost over the six-month simulated time period are also included. In general, SoS meta-architectures with a greater number of constituent systems tend to perform well. However, some architectures outperform their more costly alternatives in terms of percent of smugglers and percent of drugs interdicted. These insights help support decision makers when considering trades among alternative SoS architectures.

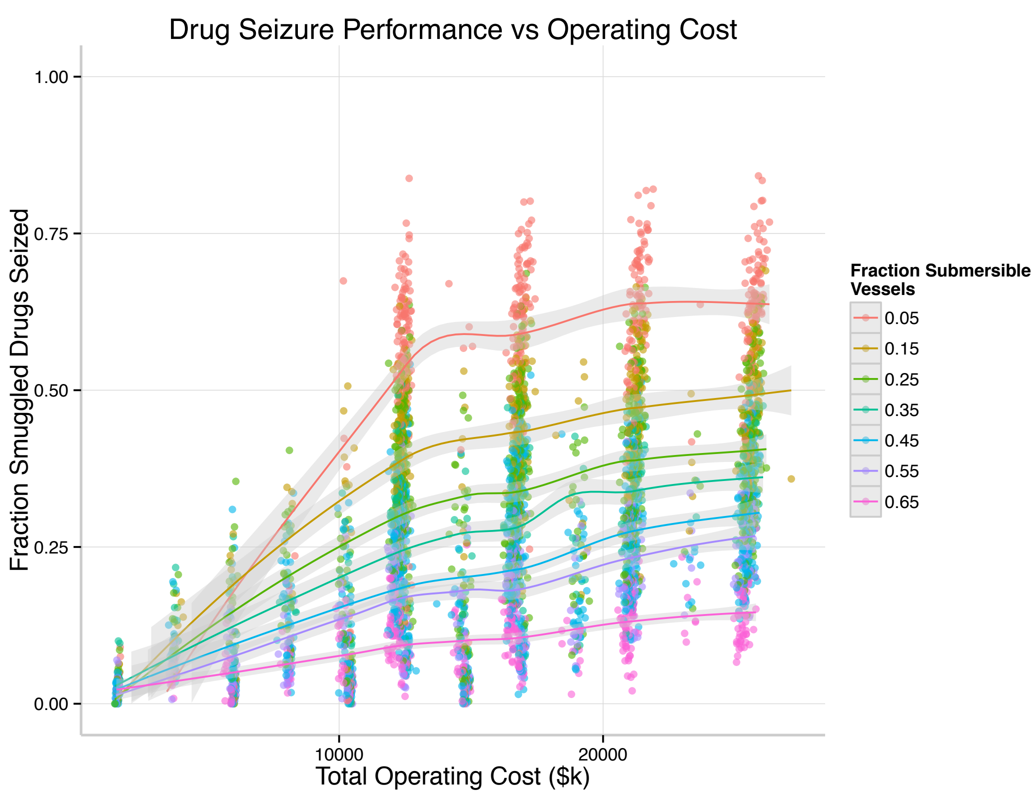
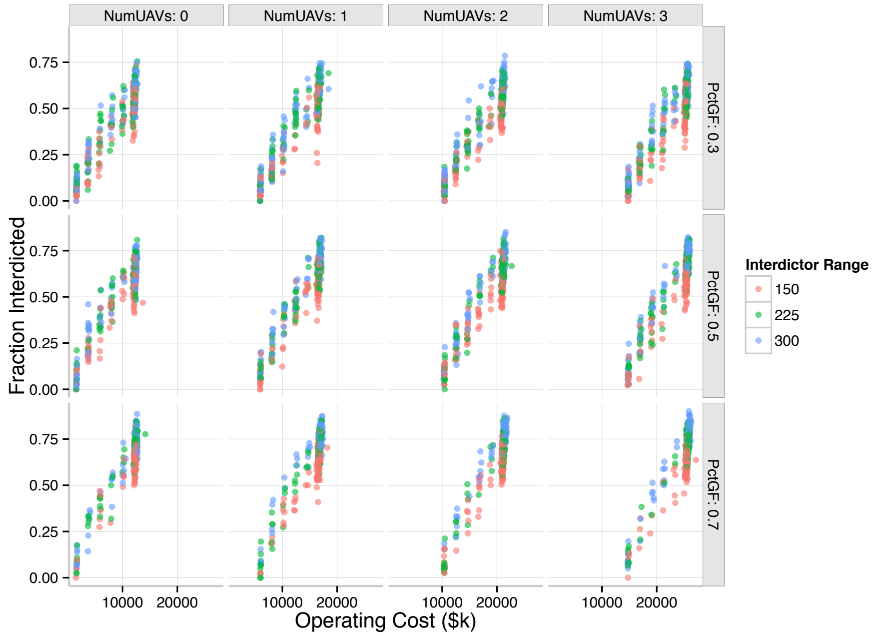
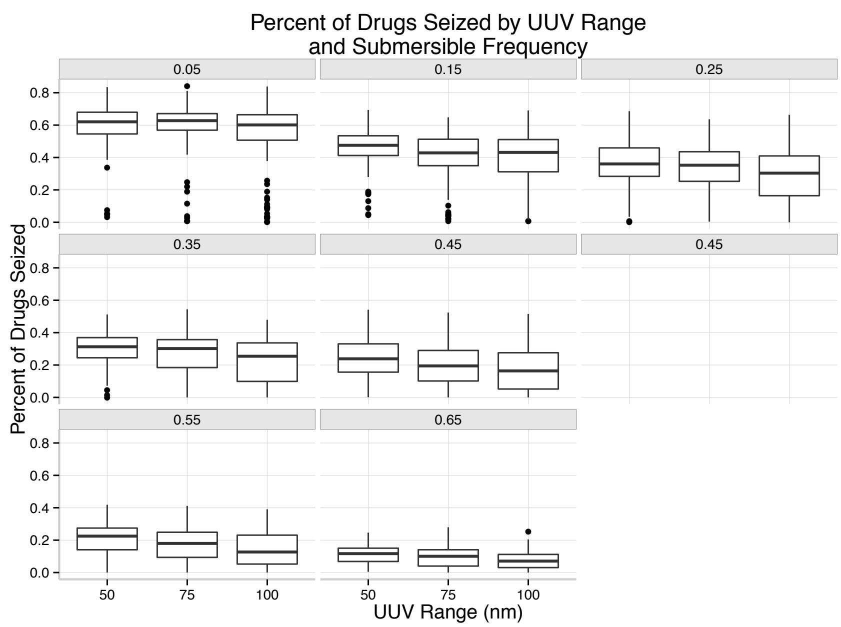


Figure 7.1. Seizure performance versus operating cost. LOESS curves have been fitted for different environments, defined by the fraction of smuggling vessels that are submersibles. Increased use of DTO submersibles significantly reduce SoS performance.



(a) Interdiction performance versus operating cost. Each facet corresponds to the number of surveillance craft (columns) and percent of DTO vessels that are go-fast boats (rows). As expected, additional surveillance assets increase to the SoS operating cost, but architectures with fewer surveillance craft have similar overall DTO smuggler interdiction performance.



(b) Seizure performance versus UUV range. Different environments are represented by submersible frequency. UUV range becomes marginally more important as the volume of submersibles grows. This effect is dwarfed by the trend in reduced drug seizure performance from increased volumes of submersible vessels.

Figure 7.2. Meta-architecture performance results.

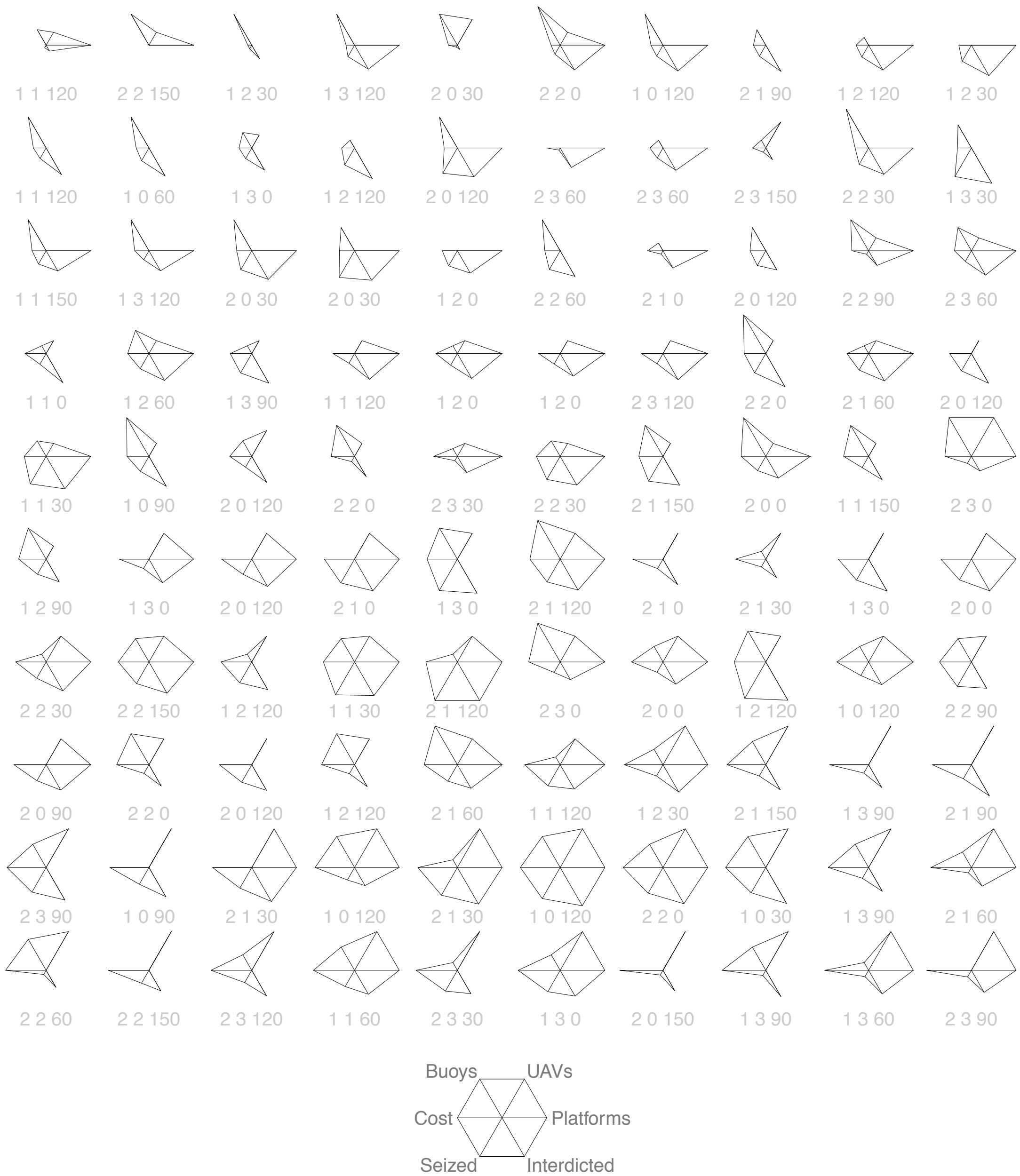


Figure 7.3. SoS meta-architectures and performance. A random subset of modeled SoS meta-architectures, ordered by increasing cost. These plots depict the number of sonobuoys, surveillance craft, and maritime platforms along the three upper-right axes. The other three (lower-left) axes depict meta-architecture performance in operating cost, percent of drugs seized, and percent of smuggling vessels interdicted. The meta-architecture encoding is included below each plot. Some meta-architectures with fewer platforms and surveillance craft still perform well. In general, increased drug seizure and vessel interdiction performance comes at increased operating cost.

7.3. REGRESSION METHOD

The challenge of understanding simulated SoS meta-architecture performance becomes a mulivariate nonlinear regression problem. Support vector machines and random forests are two methods that support multivariate nonlinear regression problems. Random forests are used for this problem because they have recently demonstrated considerable robustness in a range of classification and regression problems. These models use many subsets of the data to construct decision models, and then averages these models together to improve the overall estimate. Figure 7.4 depicts the error rate as a function of the number of trees in the random forest.

As demonstrated in Section 7.2, the volume of submersible vessels has a large influence on SoS performance. Table 7.2 displays the relative importance of each independent variable for the cutter-based UUV concept. The response variable for the random forest is percent of smuggling vessels seized. Surveillor quantity and maritime platform quantity tend to not impact the overall performance, suggesting that the number of sonobuoys and interdictor range are more important considerations when developing the counter-trafficking SoS architecture. DTO smuggling vessel use also strongly influences the SoS performance.

Table 7.2. Random forest variable importance.

Variable	Importance
Percent Submersible	0.0354
Percent Go-fast Boats	0.0046
Sonobuoy Quantity	0.0247
Interdictor Range	0.0014
UUV Range	0.0012
Surveillor Quantity	-0.0001
Platform Quantity	-0.0003

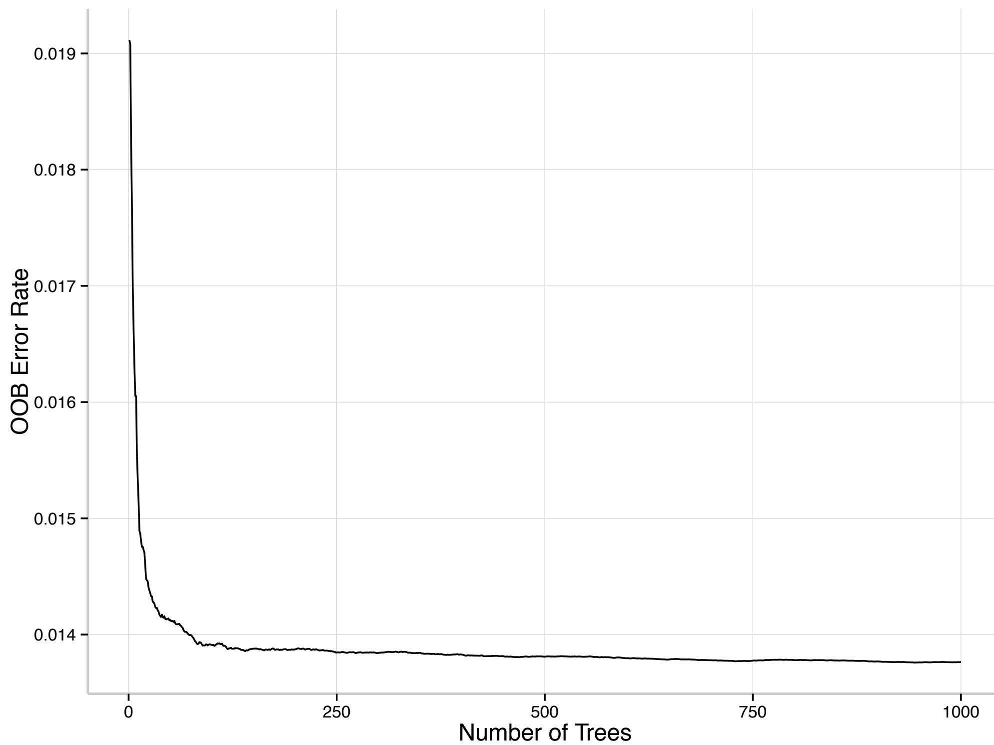


Figure 7.4. Random forest error rate by number of trees. The error rate is associated with the forecast percent of smuggling vessels seized, the response variable modeled in this random forest.

8. CONCLUSIONS AND FUTURE WORK

8.1. CONCLUSION

Modern systems continue to grow in complexity. Engineered systems-of-systems pose challenges for traditional systems engineering approaches due to complex stakeholder environments, asynchronous development of constituent systems, and changing operating environments. Many modern systems-of-systems must adapt to changes in the operating environment in order to maintain or improve performance. Mutual adaptation between the system and the environment lead to coevolution as both seek performance advantages. This behavior compounds the complexity of engineered systems-of-systems and further challenges traditional systems engineering approaches.

This work demonstrated an approach to assess a coevolutionary system-of-systems. A trade study of SoS constituent systems demonstrated the impact of an adaptive environment on stakeholder value. New SoS architecture concepts were created to address capability gaps and reduced stakeholder value. These concepts were explored in detail using agent based modeling, and the results demonstrated the usefulness of these architectures in the new environment. The results of this modeling demonstrated the substantial impact that the environment can have on SoS performance, regardless of SoS meta-architecture, if required capabilities are unavailable. The results also demonstrated that some meta-architectures with a smaller number of constituent systems had similar seizure and interdiction performance but reduced operating costs.

8.2. FUTURE WORK

Future work is needed to improve the analysis of coevolutionary systems. Opportunities to improve this type of analysis include modeling coevolution, extensions to traditional trade studies, improvements to the agent based model, and expanding the SoS analysis to include additional constituent systems or stakeholders.

8.2.1. Improved Modeling of SoS Coevolution. Accurate value models (or cost functions) of the environment or competing system could support improved system design. The value model provides a way to estimate likely adaptations in the environment or by the competing system. These changes can be modeled to understand the impacts on system performance. The result is testable architecture performance prior to development, and targeting the right set of system attributes for candidate architecture selection in this future environment.

For example, in the counter-trafficking SoS, the future implications of deploying increasing number of UAVs, UUVs or sonobuoys is unknown. However, DTOs have previously demonstrated adaptative behaviors through avoiding interdiction using faster boats and avoiding detection using submersible vessels. A successful change in the counter-trafficking architecture is likely to intiate future changes by DTOs. However, the specific changes they are likely to make are unknown at present. These changes could include new travel modes (UAVs), changes to smuggling routes, or including offensive measures to defeat unmanned counter-trafficking systems (such as detecting and destroying sonobuoys). The availability of technologies to support these adaptions is an important consideration for future adaptations.

This work likely requires abstract models to explore the complex intra-relationships between constituent systems, and inter-relationships between the SoS and the environment. Kauffman's *NKCS* model is one such model that requires the system to be encoded as a chromosome or bit string [40, 81]. The work done by Dagli et al. [82] and Giammarco [83] provide ways to encode such systems. Work by Ilachinski on complex systems, focused on defense applications, could also be used to explore emergent behavior between the SoS and the environment [84].

8.2.2. Modifications to Traditional Trade Studies. The approach demonstrated in this research considers a single objectives hierarchy for the SoS. Acknolwledged SoS consist of many differeent stakeholders and constituent systems necessary for other, unrelated missions. Constituent system stakeholders likely have objectives and values beyond those of an individual SoS. Additional work is needed to incorporate a hierarchy of stakeholder objectives or value functions. Doing so allows SoS objectives and constituent system stakeholder objectives to be considered in the SoS analysis. This disaggregatoin of value functions would help identify constitient systems most adept to participating in the SoS. Such a method could also expose gaps in needed capabilities.

8.2.3. Agent Based Modeling. The agent based model developed in this research is an abstraction of the SoS. Several assumptions could be relaxed to provide a more accurate SoS representation. Future work could include additional performance measures, availability of constituent system to support the counter-trafficking mission, and additional environmental variables that influence detection probability and interdiction capability. Operational considerations, such as traditional search patterns including parallel sweep or inward spriral patterns, could also be included. These search pattersn yield better detection performance for stationary targets than random searches.

Additional work is needed to validate and verify the agent based model. Operators and subject matter experts inform the logic behind the model and constituent system performance attributes to support validation. Empirical drug seizure data with greater fidelity than the UNODC data could be used to support model verification.

8.2.4. SoS Assessment. The counter-trafficking SoS includes other capabilities not studied in this work. This includes law enforcement efforts to curb cultivation and share information to increase smuggling interdiction. Inclusion of these other aspects of the SoS in the analysis allows other alternatives to be explored and prioritized. The role of information sharing, and network centric concepts of information reach, timeliness and quality are important consideration for coordination interdiction efforts. For example, modeling law enforcement resources could affect the volume and specificity of information available to the counter-trafficking SoS. Future models could include these aspects of the SoS to understand the affects of information sharing.

Finally, the results of the agent based model provide a mapping between input (SoS architecture and behavior rules) and output (performance measures). For complex systems, these relationships are likely nonlinear. Statistical methods such as multivariate nonlinear regression provide a way to construct a meta-model. Such methods define the mapping between dependent and independent variables. A meta-model provides a way to assess new architectures not explicity simulated. These models support decision making for the SoS architecture.

APPENDIX A

TRADE STUDY DETAILS

The surveillance and interdiction capability tables are included below. These capabilities are influenced by the types of smuggling vessels in the environment.

INITIAL AND ADAPTED ENVIRONMENT CAPABILITIES

The surveillance and interdiction capabilities differ across smuggling vessel types. Since the type and frequency of smuggling vessels changes between the initial and adapted environments, the SoS performance is impacted.

Surveillance capability depends on the alternative sweep width W and search effort Z. This analysis uses the “cookie-cutter”, or M-beta, detector model. For alternative i and smuggling vessel j, sweep width is

$$W_{ij} = p_{ij} R_i \tag{A.1}$$

where R_i is the lateral range and p_{ij} is the detection probability. Search effort is

$$Z_{ij} = W_{ij} L_i \tag{A.2}$$

where L_i is the range of alternative i. Coverage is

$$C_{ij} = Z_{ij}/A \tag{A.3}$$

where A is the search area being covered; $A = 1,000$ for this analysis. To account for search performance against multiple types of smuggling vessels, the sweep width, search effort and coverage are estimated using the fraction of each type of smuggling vessel λ_j:

$$W_i^* = \sum_j \lambda_j W_{ij} \tag{A.4}$$

$$Z_i^* = \sum_j \lambda_j Z_{ij} \tag{A.5}$$

$$C_i^* = \sum_j \lambda_j C_{ij} \tag{A.6}$$

The interdiction capability differs between the types of smuggling vessels. The interdiction capability, b_{ij}, is a relative score of the ability of the interdiction system i to interdicte smuggling vessel type j. For all combined smuggling vessels in the environment,

$$c_i = \sum_{i,j} p_j b_i j \tag{A.7}$$

where p_j is the fraction of total trafficking events using smuggling vessel type j. For the trade study, c_{ij} is the capability of the initial environment, and c'_{ij} is the capability in the new environment.

Table A.1. Search parameters for the initial environment.

	C-27J	HC-130J	P-3 Orion	MQ-9	Scaneagle	RQ-4
Cruise Speed, est. (knots)	220	374	328	80	55	130
Range (nm)	2675	5000	2380	675	809	12300
Endurance (hours)	12	14	16	24	24	34
Detection Range (km)	200	200	200	200	150	200
Lateral Range (nm)	108	108	108	108	81	108
P(d) - Fishing Boat	0.90	0.90	0.90	0.90	0.90	0.90
Sweep Width - Fishing Boat	97	97	97	97	73	97
Search Rate - Fishing Boat	21382	36350	31879	7775	4009	12635
Search Effort - Fishing Boat	259989	485961	231317	65605	58971	1195464
Coverage - Fishing Boat	260	486	231	66	59	1195
Positiive ID - Fishing Boat	0.70	0.90	0.70	0.85	0.95	0.95
P(d) - Go-fast Boat	0.90	0.90	0.90	0.90	0.90	0.90
Sweep Width - Go-fast Boat	97	97	97	97	73	97
Search Rate - Go-fast Boat	21382	36350	31879	7775	4009	12635
Search Effort - Go-fast Boat	259989	485961	231317	65605	58971	1195464
Coverage - Go-fast Boat	260	486	231	66	59	1195
Positive ID - Go-fast Boat	0.80	0.80	0.90	0.99	0.99	0.95
Fishing Boat Fraction	0.43	0.43	0.43	0.43	0.43	0.43
Go-fast Boat Fraction	0.57	0.57	0.57	0.57	0.57	0.57
W*	97.19	97.19	97.19	97.19	72.89	97.19
R*	21382	36349	31879	7775	4009	12634
Z*	259989	485961	231317	65604	58971	1195464
C*	259	485	231	65	58	1195
Positive Identification*	0.76	0.84	0.81	0.93	0.97	0.95

Table A.2. Search parameters for the adapted environment. The new capability uses the same search parameters for fishing boats and go-fast boats from the initial environment (Table B. 1).

	C-27J	HC-130J	P-3 Orion	MQ-9	Scaneagle	RQ-4
Cruise Speed, est. (knots)	220	374	328	80	55	130
Range (nm)	2675	5000	2380	675	809	12300
Endurance (hours)	12	14	16	24	24	34
Detection Range (km)	200	200	200	200	150	200
Lateral Range (nm)	108	108	108	108	81	108
P(d) LPV - Semi-submersible	0.20	0.20	0.20	0.20	0.20	0.20
Sweep Width - Semi-submersible	22	22	22	22	16	22
Search Rate - Semi-submersible	4752	8078	7084	1728	891	2808
Search Effort - Semi-submersible	57775	107991	51404	14579	13105	265659
Coverage - Semi-submersible	58	108	51	15	13	266
Positive ID - Semi-submersible	0.15	0.05	0.00	0.40	0.40	0.30
P(d) - Fully-submersible	0.00	0.00	0.00	0.00	0.00	0.00
Sweep Width - Fully-submersible	0	0	0	0	0	0
Search Rate - Fully-submersible	0	0	0	0	0	0
Search Effort - Fully-submersible	0	0	0	0	0	0
Coverage - Fully-submersible	0	0	0	0	0	0
Positive ID - Fully-submersible	0	0	0	0	0	0
Fishing Boat Fraction	0.12	0.12	0.12	0.12	0.12	0.12
Go-fast Boat Fraction	0.4	0.4	0.4	0.4	0.4	0.4
Semi-submersible Fraction	0.22	0.22	0.22	0.22	0.22	0.22
Fully-submersible Fraction	0.22	0.22	0.22	0.22	0.22	0.22
W*	55	55	55	55	41	55
R*	12164	20679	18136	4423	2281	7188
Z*	147905	276458	131594	37322	33548	680086
C*	147.9	276.5	131.6	37.3	33.5	680.1
Positive Identification*	0.44	0.44	0.44	0.59	0.60	0.56

Table A.3. The set of interdiction capability parameters for the initial and adapted environments.

	MH-60T	MH-65D	SH-60	LRI-II	OTH-IV	RB-S
Fishing Boat Interdiction Capability	0.99	0.99	0.99	0.99	0.9	0.8
Go-fast Boat Interdiction Capability	0.99	0.99	0.4	0.7	0.6	0.5
Fishing Boat Fraction	0.43	0.43	0.43	0.43	0.43	0.43
Go-fast Boat Fraction	0.57	0.57	0.57	0.57	0.57	0.57
Interdiction Capability — Initial Environment (c_{ij})	0.99	0.99	0.65	0.82	0.73	0.63
Fishing Boat Interdiction Capability	0.99	0.99	0.99	0.99	0.9	0.8
Go-fast Boat Interdiction Capability	0.99	0.99	0.4	0.7	0.6	0.5
Semi-submersible Interdiction Capability	0.4	0.3	0.15	0.4	0.2	0.1
Fully-submersible Interdiction Capability	0	0	0.0	0.0	0.0	0.0
Fishing Boat Fraction	0.12	0.12	0.12	0.12	0.12	0.12
Go-fast Boat Fraction	0.4	0.4	0.4	0.4	0.4	0.4
Semi-submersible Fraction	0.22	0.22	0.22	0.22	0.22	0.22
Fully-submersible Fraction	0.22	0.22	0.22	0.22	0.22	0.22
Interdiction Capability — Adapted Environment (c'_{ij})	0.6028	0.5808	0.3118	0.4868	0.392	0.318

VALUE FUNCTIONS

The value functions for the trade study use linear and sigmoid functions to translate measure space to stakeholder value space. The value functions for the surveillance and interdiction measures are included in Figures A.1 and A.2, respectively. Table A.4 and A.5 provide some parameters for these value functions. The full parameter set for these functions is provided in Section B.2.

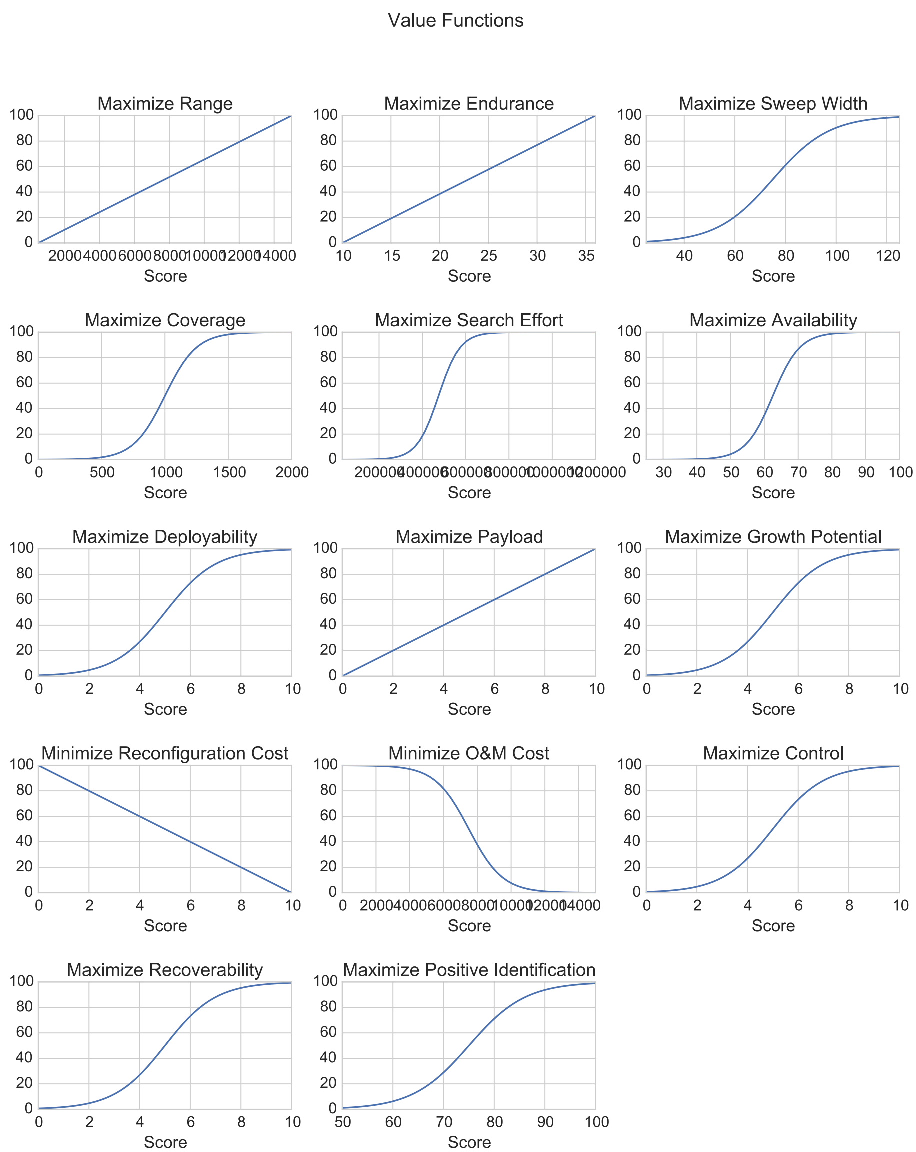


Figure A.1. Surveillance alternative value functions. The trend of each function corresponds to a minimization (decreasing) or maximization (increasing) objective.

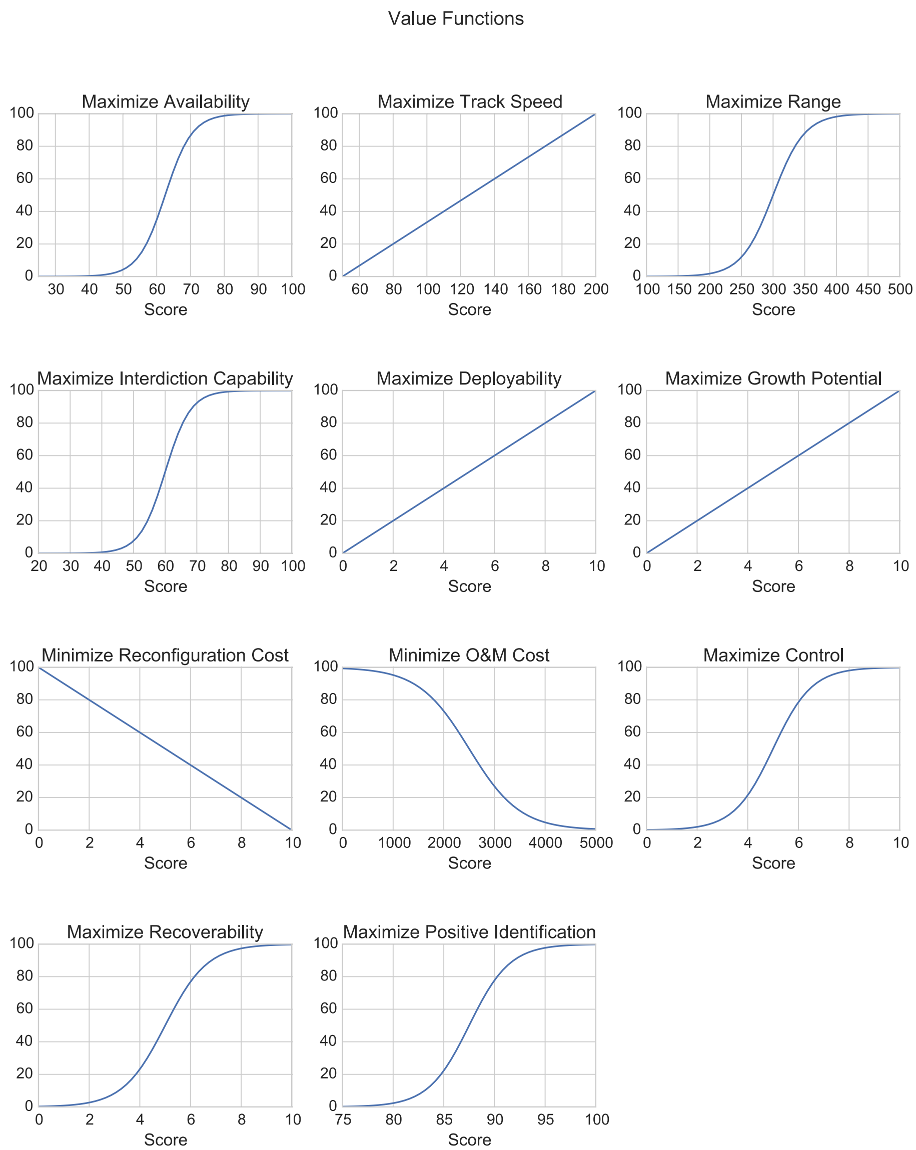


Figure A.2. Interdiction alternative value functions. The trend of each function corresponds to a minimization (decreasing) or maximization (increasing) objective.

Table A.4. Objectives for surveillance alternatives.

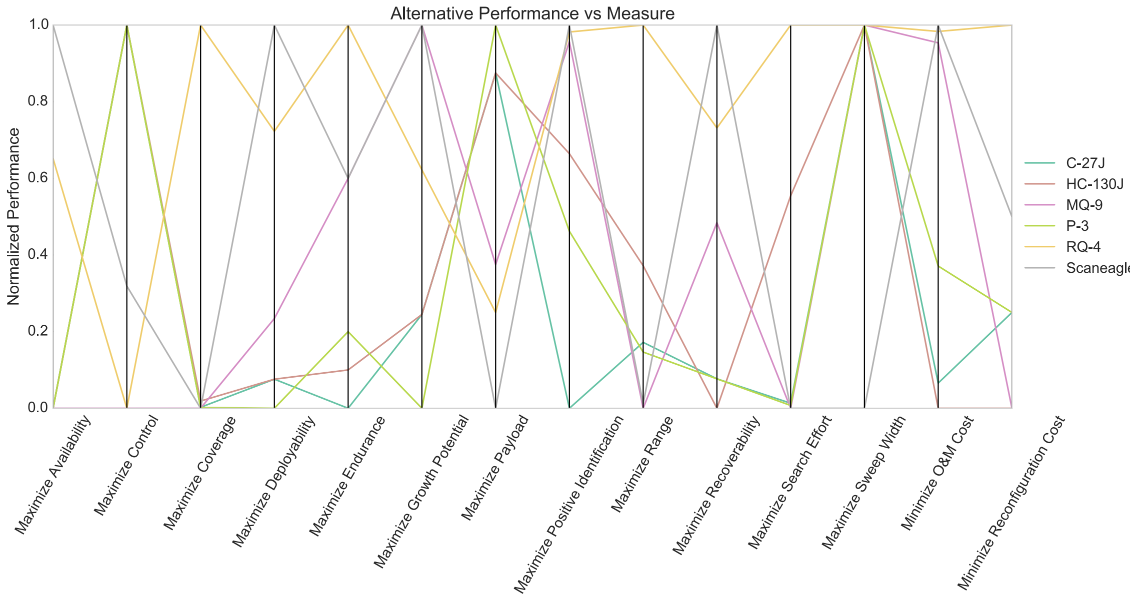
Objective	Measure	Importance	Minimum	Maximum	Ideal	Units
Detection	Maximize Range	55	500	15000	15000	nm
Detection	Maximize Endurance	90	10	36	36	hours
Detection	Maximize Sweep Width	85	25	125	125	nm
Detection	Maximize Coverage	95	0	2000	2000	unitless
Detection	Maximize Search Effort	80	30000	1200000	1200000	sq. nm
Detection	Maximize Availability	50	25	100	100	percent
Flexibility	Maximize Deployability	75	0	10	10	index
Flexibility	Maximize Payload	60	0	10	10	index
Flexibility	Maximize Growth Potential	50	0	10	10	index
Cost	Minimize Reconfiguration Cost	35	0	10	0	index
Cost	Minimize O&M Cost	65	0	15000	0	USD/hour
Limit Losses	Maximize Control	15	0	10	10	index
Limit Losses	Maximize Recoverability	30	0	10	10	index
Limit Impacts	Maximize Positive Identification	55	50	100	100	percent

Table A.5. Objectives for interdiction alternatives.

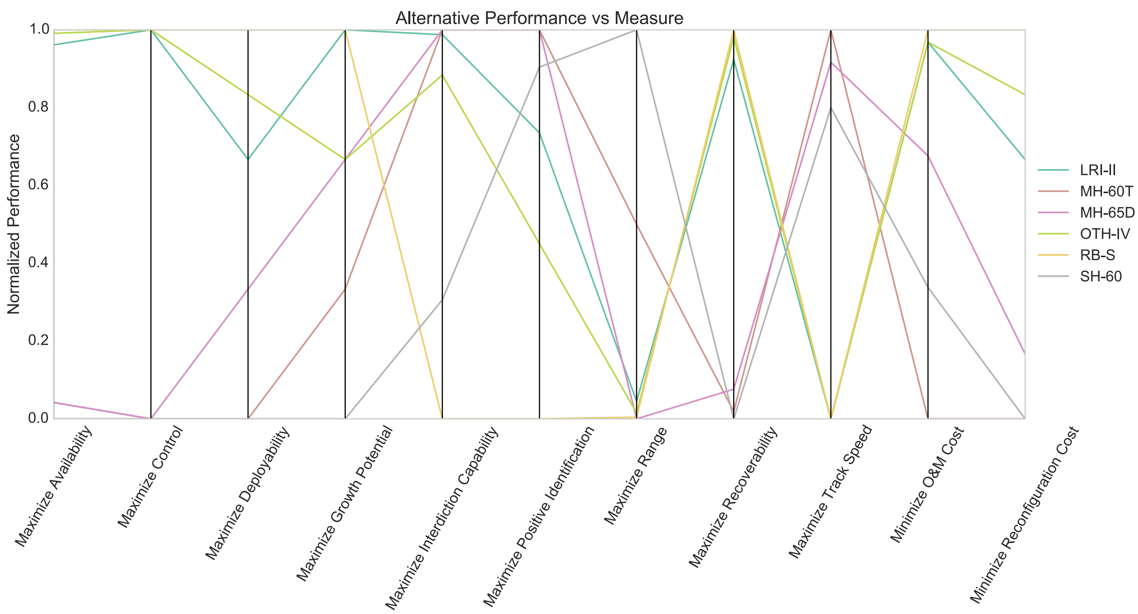
Objective	Measure	Importance	Minimum	Maximum	Ideal	Units
Interdiction	Maximize Speed	75	50	200	200	knots
Interdiction	Maximize Range	80	100	500	500	nm
Interdiction	Maximize Interdiction Capability	85	1	10	10	index
Interdiction	Maximize Availability	60	25	100	100	percent
Situational Awareness	Maximize Information Sharing	55	1	10	10	index
Situational Awareness	Maximize Coordination	80	1	10	10	index
Flexibility	Maximize Deployability	75	1	10	10	index
Flexibility	Maximize Payload	45	1	10	10	index
Flexibility	Maximize Upgradability	65	1	10	10	index
Cost	Minimize Acquisition Cost	20	100	1200	100	$K
Cost	Minimize Reconfiguration Cost	40	3	150	3	$K
Cost	Minimize O&M Cost	35	2.5	15	2.5	$K
Limit Losses	Maximize Control	50	1	10	10	index
Limit Losses	Maximize Recoverability	45	1	10	10	index
Limit Impacts	Maximize Positive Identification	90	50	100	100	percent

RESULTS

The overall results of the trade study are presented in Section 5.2.3. The contribution of each alternative against each measure, in terms of stakeholder value, are useful to compare the relative performance across alternatives for each measure. Alternative values for each performance measure are normalized for the parallel coordinates plot in Figure A.3. These results are for the SoS architecture in the initial (non-submersible) environment.



(a) Surveillance alternatives



(b) Interdiction alternatives

Figure A.3. Parallel coordinates plots for surveillance and interdiction alternative values for each performance measure. Measure values are normalized to show comparisons by value contribution of each alternative. Performance values for each measure are normalized independently.

APPENDIX B

PYTHON IMPLEMENTATION

The alt-swing software tool was developed as part of this research. The code is provided in Appendix B. The goal of this tool is to enable more rapid trade space exploration when performing a trade study of a set of alternatives. The software is developed as an IPython Notebook, and is made available on GitHub as open source software under the MIT license. The tool generates an HTML report based on user input from text and CSV files to generate a formatted HTML report. An example HTML report is included in Section B.4.

DESCRIPTION

The following code, developed as part of this research, is an implementation of the Systems Engineering tradeoff study framework [23, 25]. The code is made available at https://github.com/gm4/alt-swing. A User Guide is available at http://gm4.github.io/alt-swing/.

The `alt-swing` Python code has the following requirements:

- Python 2.7
- IPython notebook
- scipy (0.15.1)
- numpy
- pandas (0.16.1)
- matplotlib
- seaborn (0.5.1)
- markdown
- jinja2

PYTHON CODE

```python
# coding: utf-8

# # Alternative Analysis Using the Swing Weight Matrix

# An IPython Notebook implementation of the Systems Engineering trade study method
# described at http://sebokwiki.org/wiki/Decision_Management.
```

```python
# get_ipython().magic(u'matplotlib inline')
from __future__ import division
import numpy as np
import scipy as sp
from scipy.special import expit
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style = 'whitegrid') # plot aesthetics
import markdown
from jinja2 import Environment, FileSystemLoader
# --------
# distribution short names
sigString = 'sigmoid'
linString = 'linear'
powString = 'power'
triString = 'triangular'
# Value function range [0, vfRange]
vfRange = 100.0
# Weights for Importance and Swing for Swing Weight calculation
impWt = 0.65
swWt = 1.0 - impWt
# For large figures with subplots
numPlotCols = 3 # number of columns
# HTML output
htmlReport = True
# #### Define the Allowable Value Functions
# the linear, or scaled, function
def scale(x, xMin, xMax):
    """ Returns x between [0.0, 1.0] from original domain of [xMin, xMax]. """
    return (x - xMin) / (xMax - xMin)
# the triangular function
```

```
def triangular(x, l, c, r):
    """ Returns the [0,100] scaled triangular value function evaluated
    at x for (l)eft, (c)enter, (r)ight triangular parameters."""
    return vfRange * max(min(((x - l)/(c - l)), ((r - x)/(r - c))), 0.0)
# the bell function
def bell(x, a, b, c):
    """ Returns the [0,100] scaled generalized bell curve
    evaluated at x for (c)enter and shape parameters a and b."""
    return vfRange * 1.0 / (1.0 + pow(np.abs((x - c)/a), (2.0*b)))
# the sigmoid function
def sigmoid(x, a, c):
    """Returns the [0,100] scaled sigmoid function evaluated
    at x for (a)lpha and (c)enter."""
    return vfRange * (1.0 / (1.0 + np.exp(-1.0 * a * (x - c))))
# --------
# ## Read Input Files
# ### Objectives and Measures
# Read directly into a `pandas` DataFrame
# Define the set of value function families to use:
# Family  |      Value Function Form
# ------- | -------------------------------------------
# Linear  |  $$ f(x) = mx + b $$
# Power   |  $$ f(x) = mx^a $$
# Sigmoid |  $$ f(x) = \frac{a}{b + e^{-ax/2}} $$
objDF = pd.read_csv('./input/surveillance-objectives.csv',
    index_col = ['Objective', 'Measure'])
# objDF = pd.read_csv('./input/surveillance-objectives-subset.csv',
    index_col = ['Objective', 'Measure'])
# objDF = pd.read_csv('./input/interdiction-objectives.csv',
    index_col = ['Objective', 'Measure'])
# objDF = pd.read_csv('./input/interdiction-objectives-subset.csv',
    index_col = ['Objective', 'Measure'])
```

```
# objDF   # Uncomment to view the DataFrame inline
# #### Build the Value Functions for Each Objective and Measure
# This section creates plots of the value functions defined for each measure.
# The result
# is a series of figure with subplots of each value function.
# ----------
# **Note:** *If you defined additional value functions above,
# you will need to add these to
# the below loop to make sure they are evaluated.*
# ---------
tmpDF = pd.DataFrame(columns=['Measure', 'Score', 'Value'])
# Get the number of unique Measures to plot
numPlotRows = int(round(
                np.ceil(len(objDF.index.levels[1].unique()) / numPlotCols), 0))
fig, axs = plt.subplots(numPlotRows, numPlotCols, figsize = (13, 15))
# Loop through the subplots and objDF indices
for ax, idx in zip(axs.flat, objDF.index):
    vals = objDF.loc[idx] # get the dataframe columns for this index
    axMin = float(vals.Minimum)
    axMax = float(vals.Maximum)
    domain = np.linspace(axMin, axMax)
    # Build the corresponding value function
    if vals.Family == 'sigmoid':
        valFunc = [sigmoid(i, float(vals.Param1), vals.Param2) for i in domain]
        if float(vals.Slope) == -1.0:
            valFunc[:] = [vfRange - i for i in valFunc]
    elif vals.Family == 'linear':
        valFunc = [vfRange * scale(i, axMin, axMax) for i in domain]
        if float(vals.Slope) == -1.0:
            valFunc[:] = [vfRange - i for i in valFunc]
    elif vals.Family == 'power':
        valFunc = [vfRange * np.power(scale(i, axMin, axMax),
                vals.Param1) for i in domain]
```

```
        if float(vals.Slope) == -1.0:
            valFunc[:] = [max(valFunc) - i for i in valFunc]
    elif vals.Family == 'triangular':
        valFunc = [triangular(i, axMin, vals.Param1, axMax) for i in domain]
        if float(vals.Slope) == -1.0:
            valFunc[:] = [max(valFunc) - i for i in valFunc]
    else:
        valsFunc = [0]*len(domain)
        print 'This value function family is not yet implemented.'
    tmpDF.Measure = str(idx[1]) # Assign the Measure
    tmpDF.Score = domain
    tmpDF.Value = valFunc
    plotTitle = str(idx[1])
    tmpDF.plot(ax=ax, x = 'Score', y = 'Value', title = plotTitle, legend = False)
    plt.subplots_adjust(hspace = 0.7)
for ax in axs.flat[axs.size - 1:len(objDF.index) - 1:-1]:
    ax.set_visible(False)
plt.suptitle('Value Functions', fontsize = 16)
plt.savefig('./html_report/images/value-functions.png', bbox_inches='tight')
plt.savefig('./html_report/images/value-functions.pdf', bbox_inches='tight')
plt.show()
# ### Alternatives
# Read directly into a `pandas` DataFrame
altDF = pd.read_csv('./input/surveillance-alternatives-0.csv',
    index_col=['Objective', 'Measure', 'Alternative'])
# altDF = pd.read_csv('./input/surveillance-alternatives-1.csv',
    index_col=['Objective', 'Measure', 'Alternative'])
# altDF = pd.read_csv('./input/surveillance-alternatives-0-subset.csv',
    index_col=['Objective', 'Measure', 'Alternative'])
# altDF = pd.read_csv('./input/surveillance-alternatives-1-subset.csv',
    index_col=['Objective', 'Measure', 'Alternative'])
# altDF = pd.read_csv('./input/interdiction-alternatives-0.csv',
```

```
    index_col=['Objective', 'Measure', 'Alternative'])
# altDF = pd.read_csv('./input/interdiction-alternatives-1.csv',
    index_col=['Objective', 'Measure', 'Alternative'])
# altDF = pd.read_csv('./input/interdiction-alternatives-0-subset.csv',
    index_col=['Objective', 'Measure', 'Alternative'])
# altDF = pd.read_csv('./input/interdiction-alternatives-1-subset.csv',
    index_col=['Objective', 'Measure', 'Alternative'])
altDF['Consequence'] = np.NaN
# altDF   #  Uncomment to view inline
# --------
# ## Score each Alternative against each Objective and Measure
for item, val in altDF.iterrows():
    idxString = list(item) # convert this alternative's index to a list
    print '\n\nidxString is: ', idxString
    print 'val is: \n -----\n', val, '\n-----'
    # drop the 'Alternative' from the index used for the objective DataFrame
    objIdx = idxString[0:2]
    print 'objective index is: \n', objIdx
    # get the corresponding objective for this index
    obj = objDF.ix[objIdx[0],objIdx[1],]
    print 'objDF row is: \n', obj
    score = np.NaN
    funcFamily = str(obj['Family']) # get the corresponding value function family
    # the measured value for this alternative
    paramX = altDF.ix[idxString[0], idxString[1], idxString[2]]['Score']
    paramSlope = objDF.ix[objIdx[0],objIdx[1]]['Slope'] # the slope
    paramXMin = objDF.ix[objIdx[0],objIdx[1]]['Minimum'] # minimum acceptable
    paramXMax = objDF.ix[objIdx[0],objIdx[1]]['Maximum'] # maximum desirable
    paramX1 = objDF.ix[objIdx[0],objIdx[1]]['Param1'] # 1st function parameter
    paramX2 = objDF.ix[objIdx[0],objIdx[1]]['Param2'] # 2nd function parameter
    paramX3 = objDF.ix[objIdx[0],objIdx[1]]['Param3'] # 3rd function parameter
    if funcFamily == sigString:
```

```
        print str(idxString[0:2]) + ' is ' + sigString
        score = sigmoid(paramX, paramX1, paramX2)
        if float(paramSlope) == -1.0:
            score = vfRange - score
    elif funcFamily == linString:
        print str(idxString[0:2]) + ' is ' + linString
        score = vfRange * scale(paramX, paramXMin, paramXMax)
        if float(paramSlope) == -1.0:
            score = vfRange - score
    elif funcFamily == powString:
        print str(idxString[0:2]) + ' is ' + powString
        score = np.power(scale(paramX, paramXMin, paramXMax), paramX1)
        if float(paramSlope) == -1.0:
            score = vfRange - score
    elif funcFamily == triString:
        print str(idxString[0:2]) + ' is ' + triString
        score = triangular(paramX, paramXMin, paramX1, paramXMax)
        if float(paramSlope) == -1.0:
            score = vfRange - score
    else:
        print 'The "',funcFamily, '" value function family is not yet implemented.'
    if score > 100.0:
        score = 100.0
    elif score < 0.0:
        score = 0.0
    print 'Value against this measure is ', score
    altDF.loc[(idxString[0], idxString[1], idxString[2]),'Consequence'] = np.round(
            float(score), 3)
print altDF['Score'].dropna('index')   # Uncomment to view the DataFrame inline
# -----------
# ## Calculate the Swing Weight for each Objective and Measure
objDF.loc[:,'Swing'] = np.NaN
```

```
for idx, val in objDF.iterrows():
    try:
        objMin = min(altDF.loc[(idx[0], idx[1]),'Consequence'])
        objMax = max(altDF.loc[(idx[0], idx[1]),'Consequence'])
    except:
        print "\nNo Score found for ", idx
        objMin = np.NaN
        objMax = np.NaN

    swing = objMax - objMin
    objDF.loc[(idx[0], idx[1]),'Swing'] = swing
    print idx, ' min: ', objMin, '  max: ', objMax, '  swing: ', swing
objResults = objDF.copy()
objResults.reset_index(inplace=True)
print objResults
sns.lmplot(x="Swing", y="Importance", data=objResults, fit_reg=False,
           hue = "Measure", aspect=1.3, scatter_kws={"s": 100},
           palette=sns.color_palette("Paired", n_colors=16, desat=.5))
plt.xlim(-0.1,101)
plt.ylim(-0.1,101)
plt.title("Measure Swing vs. Importance")
plt.savefig('./html_report/images/swing-importance.png', bbox_inches='tight')
plt.savefig('./html_report/images/swing-importance.pdf', bbox_inches='tight')
# ### Calculate the Unnormalized Weight
objDF.loc[:,'Weight'] = impWt * objDF.Importance + swWt * objDF.Swing
# ### Calculate the Normalized Weight
objDF.loc[:,'NormdWt'] = objDF.loc[:,'Weight'] / objDF.loc[:,'Weight'].sum()
# objDF   # Uncomment to view the DataFrame inline
# -------------
# ## Evaluating the Alternative's Value
# ### Calculate Total Value for Each Alternative
# $$
```

```
# V(x) = \sum_{i = 1}^{n}w_{i} v_{i}(x_{i})
# $$
# where $V(x)$ is the total value, $i$ is the index of the objective/measure,
# $w_i$
# is the normalized weight for objective/measure $i$, $x_i$ is the
# alternative's score
# for objective measure $i$, and $v_i(x_i)$ is the corresponding value of $x_i$.
for idx, vals in altDF.iterrows():
    altDF.loc[:,'WtdConsequence'] =
    objDF.loc[(idx[0],idx[1]),'NormdWt'] * altDF.Consequence
# altDF    # Uncomment to view the DataFrame inline
print(altDF['WtdConsequence'].dropna('index'))
# ------------
# ## Visualizing Output
# #### Heatmap (Consequences Scorecard)
# Display the relative performance of each Alternative against each Measure.
heatDF = altDF.drop(['Score', 'Units', 'WtdConsequence'], axis = 1)
heatDF.reset_index(inplace=True)
summaryDF = altDF.drop(['Score', 'Consequence'], axis=1).groupby(
                        level = 'Alternative').agg(sum)
summaryDF.columns = ['Value']
# summaryDF   # Uncomment to view the DataFrame inline
tmpDF = altDF.drop(['Units', 'Consequence'], axis = 1)
# tmpDF   # Uncomment to view the DataFrame inline
for idx, cols in tmpDF.iterrows():
    val = summaryDF.loc[(idx[2]), 'Value']
    tmpDF.loc[idx,'Value'] = val
heat_rect = heatDF.pivot('Alternative', 'Measure', 'Consequence')
# heat_rect.dropna("columns")   # Uncomment to view the DataFrame inline
print(heat_rect)
sns.heatmap(np.round(heat_rect.dropna('columns'), 0),
            annot=True, fmt='g', cbar=False)
```

```
plt.title("Heatmap of Alternative Measure Values")
plt.savefig('./html_report/images/value-scorecard.png', bbox_inches='tight')
plt.savefig('./html_report/images/value-scorecard.pdf', bbox_inches='tight')
# #### Trellis Plot of Alternative Value vs Measure Score
# Allows a quick comparison of the total value and original score of
# Alternatives against all Measures.
summaryDF.reset_index(inplace=True)
sns.barplot('Alternative', 'Value', data=summaryDF, palette='muted')
plt.ylabel("Value")
plt.ylim(0,60)
plt.title('Total Alternative Value')
plt.savefig('./html_report/images/value-barplot.png', bbox_inches='tight')
plt.savefig('./html_report/images/value-barplot.pdf', bbox_inches='tight')
print(summaryDF)
tmpDF.reset_index(inplace=True)
grid = sns.FacetGrid(tmpDF, col="Measure", hue="Alternative", col_wrap=3, size=4,
                    legend_out = True, sharex=False, sharey=True)
grid.map(plt.plot, "Score", "Value", marker="o", ms=14, alpha=0.6)
grid.fig.tight_layout(w_pad=1)
sns.set_context("paper", font_scale=1.6)
grid.add_legend()
grid.savefig('./html_report/images/measure-trellis.png', bbox_inches='tight')
grid.savefig('./html_report/images/measure-trellis.pdf', bbox_inches='tight')
# Format the DataFrame to produce the parallel coordinates plot.
fooDF = heat_rect.copy()
# normalize the measure values for the parallel coordinates plot
for col in fooDF.columns:
    fooDF[col] = (
        fooDF[col] - np.min(fooDF[col])) / (np.max(fooDF[col]
                                            ) - np.min(fooDF[col]))
print(fooDF)
# reset the index, but keep 'Alternative' as a column
```

```
fooDF = fooDF.reset_index(level=0,drop = False)
fooDF.index.name = None
plt.figure(figsize = (15,6))
pd.tools.plotting.parallel_coordinates(fooDF, 'Alternative', colormap = 'Set2')
plt.xticks(rotation=60)
plt.ylabel("Normalized Performance")
plt.title('Alternative Performance vs Measure')
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.savefig('./html_report/images/parallel-coordinates.png', bbox_inches='tight')
plt.savefig('./html_report/images/parallel-coordinates.pdf', bbox_inches='tight')
# -----------------
# ## The creation of the HTML Report is availabe from the online version.
#    It requires additional input files that the user can modify to tailor the
#    report.
# -----------
```

INPUT DATA

The input data can be generated in spreadsheet software. However, the Python code requires this data in comma separated values (CSV) files.

1. Surveillance Alternatives. The initial set of surveillance alternatives include the following data stored as a comma separated values (CSV) file.

2. Interdiction Alternatives. The following objectives and measures correspond to the interdiction capability of the counter trafficking SoS:

```
Objective,Measure,Importance,Minimum,Maximum,Ideal,Units,Family,Slope,Param1,Param2,Param3
Interdiction,Maximize Speed,75,50,200,200,knots,linear,1,1,,
Interdiction,Maximize Range,80,100,500,500,nm,sigmoid,1,0.04,300,
Interdiction,Maximize Interdiction Capability,85,10,100,10,percent,sigmoid,1,0.1,55,
Interdiction,Maximize Availability,60,25,100,100,percent,sigmoid,1,0.25,67.5,
Situational Awareness,Maximize Information Sharing,55,1,10,10,index,linear,1,,,
Situational Awareness,Maximize Coordination,80,1,10,10,index,linear,1,,,
```

```
Flexibility,Maximize Deployability,75,1,10,10,index,linear,1,,,
Flexibility,Maximize Payload,45,1,10,10,index,sigmoid,1,1.15,6,
Flexibility,Maximize Upgradability,65,1,10,10,index,linear,1,,,
Cost,Minimize Acquisition Cost,20,100,1200,100,$K,sigmoid,-1,0.01,550,
Cost,Minimize Reconfiguration Cost,40,3,150,3,$K,linear,-1,,,
Cost,Minimize O&M Cost,35,2.5,15,2.5,$K,sigmoid,-1,0.3,9,
Cost,Minimize Retirement Cost,25,5,120,5,$K,linear,-1,,,
Limit Losses,Maximize Control,50,1,10,10,index,sigmoid,1,1.3,5,
Limit Losses,Maximize Recoverability,45,1,10,10,index,sigmoid,1,1,5,
Limit Impacts,Maximize Positive Identification,90,50,100,100,percent,sigmoid,1,0.3,75,
```

The following input corresponds to the initial interdiction alternatives

```
Alternative,Objective,Measure,Score,Units
MH-60T Jayhawk,Interdiction,Maximize Speed,170,knots
MH-60T Jayhawk,Interdiction,Maximize Range,300,nm
MH-60T Jayhawk,Interdiction,Maximize Interdiction Capability,7,index
MH-60T Jayhawk,Interdiction,Maximize Availability,70,percent
MH-60T Jayhawk,Situational Awareness,Maximize Information Sharing,6,index
MH-60T Jayhawk,Situational Awareness,Maximize Coordination,6,index
MH-60T Jayhawk,Flexibility,Maximize Deployability,1,index
MH-60T Jayhawk,Flexibility,Maximize Payload,7,index
MH-60T Jayhawk,Flexibility,Maximize Upgradability,7,index
MH-60T Jayhawk,Cost,Minimize Acquisition Cost,,$K
MH-60T Jayhawk,Cost,Minimize Reconfiguration Cost,,$K
MH-60T Jayhawk,Cost,Minimize O&M Cost,,$K
MH-60T Jayhawk,Cost,Minimize Retirement Cost,,$K
MH-60T Jayhawk,Limit Losses,Maximize Control,9,index
MH-60T Jayhawk,Limit Losses,Maximize Recoverability,5,index
MH-60T Jayhawk,Limit Impacts,Maximize Positive Identification,80,percent
MH-65D Dolphin,Interdiction,Maximize Speed,160,knots
MH-65D Dolphin,Interdiction,Maximize Range,150,nm
MH-65D Dolphin,Interdiction,Maximize Interdiction Capability,7,index
```

```
MH-65D Dolphin,Interdiction,Maximize Availability,90,percent
MH-65D Dolphin,Situational Awareness,Maximize Information Sharing,6,index
MH-65D Dolphin,Situational Awareness,Maximize Coordination,6,index
MH-65D Dolphin,Flexibility,Maximize Deployability,6,index
MH-65D Dolphin,Flexibility,Maximize Payload,5,index
MH-65D Dolphin,Flexibility,Maximize Upgradability,5,index
MH-65D Dolphin,Cost,Minimize Acquisition Cost,,$K
MH-65D Dolphin,Cost,Minimize Reconfiguration Cost,,$K
MH-65D Dolphin,Cost,Minimize O&M Cost,,$K
MH-65D Dolphin,Cost,Minimize Retirement Cost,,$K
MH-65D Dolphin,Limit Losses,Maximize Control,9,index
MH-65D Dolphin,Limit Losses,Maximize Recoverability,5,index
MH-65D Dolphin,Limit Impacts,Maximize Positive Identification,60,percent
LRI-II,Interdiction,Maximize Speed,38,knots
LRI-II,Interdiction,Maximize Range,225,nm
LRI-II,Interdiction,Maximize Interdiction Capability,3,index
LRI-II,Interdiction,Maximize Availability,95,percent
LRI-II,Situational Awareness,Maximize Information Sharing,4,index
LRI-II,Situational Awareness,Maximize Coordination,4,index
LRI-II,Flexibility,Maximize Deployability,10,index
LRI-II,Flexibility,Maximize Payload,3,index
LRI-II,Flexibility,Maximize Upgradability,6,index
LRI-II,Cost,Minimize Acquisition Cost,,$K
LRI-II,Cost,Minimize Reconfiguration Cost,,$K
LRI-II,Cost,Minimize O&M Cost,,$K
LRI-II,Cost,Minimize Retirement Cost,,$K
LRI-II,Limit Losses,Maximize Control,9,index
LRI-II,Limit Losses,Maximize Recoverability,8,index
LRI-II,Limit Impacts,Maximize Positive Identification,99,percent
OTH-IV,Interdiction,Maximize Speed,40,knots
OTH-IV,Interdiction,Maximize Range,200,nm
OTH-IV,Interdiction,Maximize Interdiction Capability,3,index
OTH-IV,Interdiction,Maximize Availability,95,percent
```

```
OTH-IV,Situational Awareness,Maximize Information Sharing,4,index
OTH-IV,Situational Awareness,Maximize Coordination,4,index
OTH-IV,Flexibility,Maximize Deployability,10,index
OTH-IV,Flexibility,Maximize Payload,5,index
OTH-IV,Flexibility,Maximize Upgradability,7,index
OTH-IV,Cost,Minimize Acquisition Cost,,$K
OTH-IV,Cost,Minimize Reconfiguration Cost,,$K
OTH-IV,Cost,Minimize O&M Cost,,$K
OTH-IV,Cost,Minimize Retirement Cost,,$K
OTH-IV,Limit Losses,Maximize Control,9,index
OTH-IV,Limit Losses,Maximize Recoverability,8,index
OTH-IV,Limit Impacts,Maximize Positive Identification,99,percent
RB-S,Interdiction,Maximize Speed,45,knots
RB-S,Interdiction,Maximize Range,175,nm
RB-S,Interdiction,Maximize Interdiction Capability,4,index
RB-S,Interdiction,Maximize Availability,95,percent
RB-S,Situational Awareness,Maximize Information Sharing,4,index
RB-S,Situational Awareness,Maximize Coordination,4,index
RB-S,Flexibility,Maximize Deployability,10,index
RB-S,Flexibility,Maximize Payload,4,index
RB-S,Flexibility,Maximize Upgradability,6,index
RB-S,Cost,Minimize Acquisition Cost,,$K
RB-S,Cost,Minimize Reconfiguration Cost,,$K
RB-S,Cost,Minimize O&M Cost,,$K
RB-S,Cost,Minimize Retirement Cost,,$K
RB-S,Limit Losses,Maximize Control,9,index
RB-S,Limit Losses,Maximize Recoverability,8,index
RB-S,Limit Impacts,Maximize Positive Identification,99,percent
```

EXAMPLE HTML OUTPUT

The below figures present the HTML output from the `alt-swing` code. The goal of this tool is to enable more rapid trade space exploration when performing a trade study for a set of alternatives. The formatted HTML report is optionally generated based on user input from text and CSV files. Narrative descriptions under each section of the report use markdown syntax text which should be modified by the user. The Python packages markdown and jinja2 are used in the background to transform the text, tables and figures into the HTML report automatically. The following figures are examples of this output.

Default alt-swing Report

Author Name

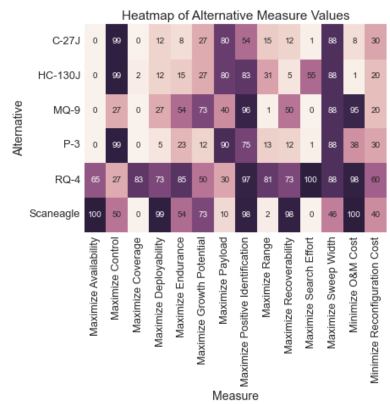
Introduction

This section introduces the trade study problem.

Bottom Line Results

To get straight to the point.

	Value
Alternative	
C-27J	24.113833
HC-130J	28.802832
MQ-9	31.720632
P-3	26.935759
RQ-4	56.085392
Scaneagle	42.754911



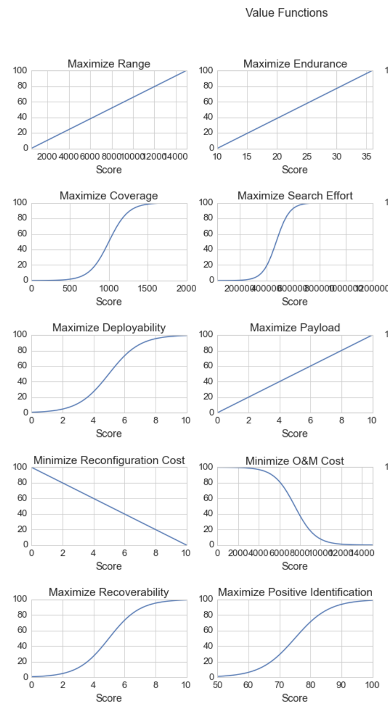
Objectives and Measures

This is boilerplate text to put in your HTML report. Describe the objectives and measures that you use, why you selected them, etc. You may want to include a subsection on stakeholders as well.

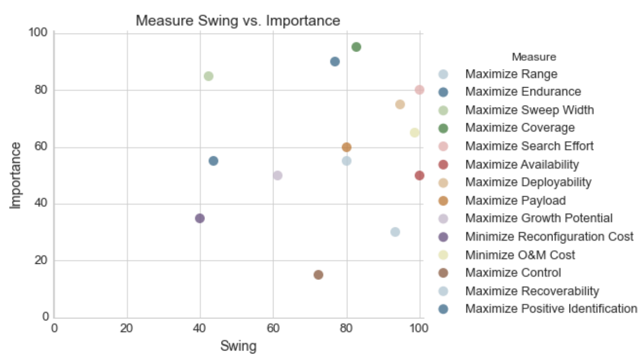
The following table provides the Objectives and Measures used in this analysis.

		Importance	Minimum	Maximum	Ideal	Units	Swing	NormdWt
Objective	**Measure**							
Detection	**Maximize**							

	Range	55	500	15000	15000	nm	80.172	0.069426
	Maximize Endurance	90	10	36	36	hours	76.923	0.092941
	Maximize Sweep Width	85	25	125	125	nm	42.356	0.076242
	Maximize Coverage	95	0	2000	2000	unitless	82.581	0.098631
	Maximize Search Effort	80	30000	1200000	1200000	sq. nm	99.976	0.094647
	Maximize Availability	50	25	100	100	percent	99.969	0.073429
Flexibility	**Maximize Deployability**	75	0	10	10	index	94.588	0.089060
	Maximize Payload	60	0	10	10	index	80.000	0.072896
	Maximize Growth Potential	50	0	10	10	index	61.186	0.058660
Cost	**Minimize Reconfiguration Cost**	35	0	10	0	index	40.000	0.039984
	Minimize O&M Cost	65	0	15000	0	USD/hour	98.751	0.083573
Limit Losses	**Maximize Control**	15	0	10	10	index	72.437	0.038192
	Maximize Recoverability	30	0	10	10	index	93.458	0.056805
Limit Impacts	**Maximize Positive Identification**	55	50	100	100	percent	43.641	0.055515



Nunc vel gravida dui, ac aliquam augue. Vivamus eu ultrices mauris, sit amet dictum diam.



Alternatives

This is boilerplate text to put in your report.

You may want to describe the alternatives that you use and any assumptions that you made.

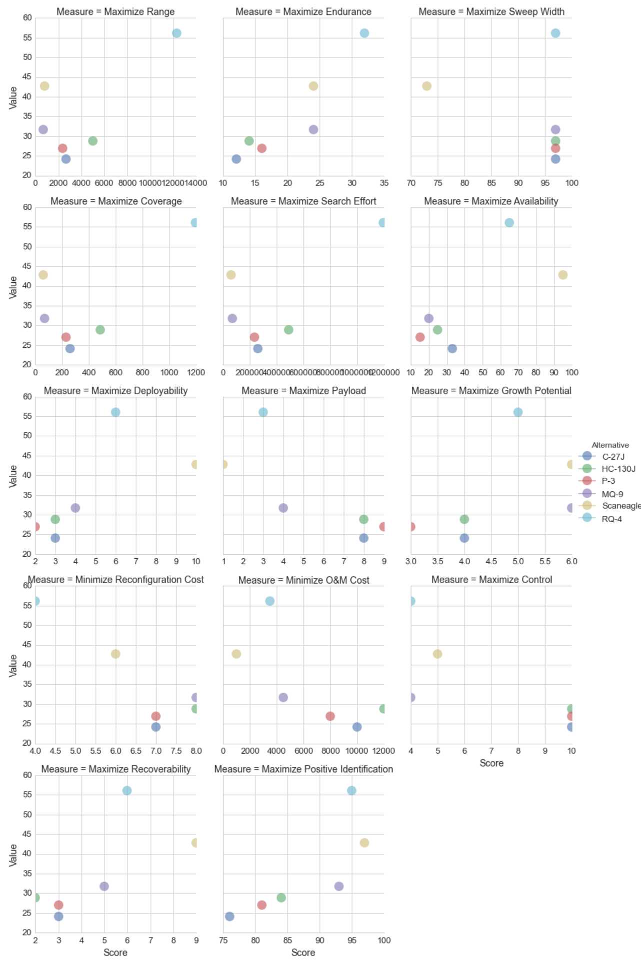
Alternative A provides ...

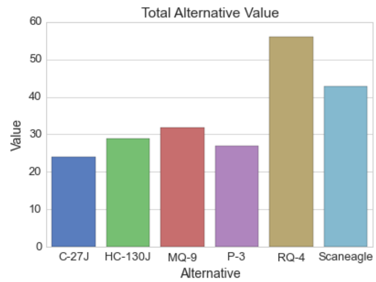
Alternative B uses ...

These alternatives, and the corresponding scores for each Objective and Measure are included in the table below:

	Score										
Objective	**Detection**						**Flexibility**			**Cost**	
Measure	**Maximize Range**	**Maximize Endurance**	**Maximize Sweep Width**	**Maximize Coverage**	**Maximize Search Effort**	**Maximize Availability**	**Maximize Deployability**	**Maximize Payload**	**Maximize Growth Potential**	**Minimize Reconfiguration Cost**	**Mini O&M Cos**
Units	**nm**	**hours**	**nm**	**unitless**	**sq nm**	**percent**	**index**	**index**	**index**	**index**	**USD**
Alternative											
C-27J	2675	12	97	260	259989	33	3	8	4	7	1000
HC-130J	5000	14	97	486	485961	25	3	8	4	8	1200
MQ-9	675	24	97	68	65604	20	4	4	6	8	4500
P-3	2380	16	97	231	231317	15	2	9	3	7	8000
RQ-4	12300	32	97	1195	1195464	65	6	3	5	4	3500
Scaneagle	809	24	73	59	58971	95	10	1	6	6	1000

Nunc vel gravida dui, ac aliquam augue. Vivamus eu ultrices mauris, sit amet dictum diam.


Measure = Maximize Range
Measure = Maximize Endurance
Measure = Maximize Sweep Width
Measure = Maximize Coverage
Measure = Maximize Search Effort
Measure = Maximize Availability
Measure = Maximize Deployability
Measure = Maximize Payload
Measure = Maximize Growth Potential
Measure = Minimize Reconfiguration Cost
Measure = Minimize O&M Cost
Measure = Maximize Control
Measure = Maximize Recoverability
Measure = Maximize Positive Identification
Value
Score
Alternative
C-27J
HC-130J
P-3
MQ-9
Scaneagle
RQ-4



Vivamus eu ultrices dui, ac aliquam augue.

Built from alt-swing using Skeleton.

BIBLIOGRAPHY

[1] M. Jamshidi, "Introduction to system of systems," *System of Systems Engineering. Innovations for the 21st Century; Jamshidi, M., Ed*, pp. 1–43, 2009.

[2] Office of the Deputy Under Secretary of Defense for Acquisition and Technology, *Systems Engineering Guide for Systems of Systems*. ODUSD(A&T)SSE, 2008, vol. 36. [Online]. Available: http://www.acq.osd.mil/se/docs/SE-Guide-for-SoS.pdf

[3] M. Jamshidi, "System of Systems Engineering - New Challenges for the 21st Century," *IEEE A&E Systems Magazine*, pp. 4–19, 2008.

[4] S. Agarwal, L. E. Pape, N. Kilicay-Ergin, and C. H. Dagli, "Multi-agent Based Architecture for Acknowledged System of Systems," *Procedia Computer Science*, vol. 28, pp. 1–10, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050914000647

[5] L. Pape, K. Giammarco, J. Colombi, C. Dagli, N. Kilicay-Ergin, and G. Rebovich, "A fuzzy evaluation method for system of systems meta-architectures," *Procedia Computer Science*, vol. 16, pp. 245–254, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2013.01.026

[6] L. Pape, S. Agarwal, K. Giammarco, and C. Dagli, "Fuzzy optimization of acknowledged system of systems meta-architectures for agent based modeling of development," *Procedia Computer Science*, vol. 28, pp. 404–411, 2014.

[7] C. H. Dagli, D. Enke, N. Ergin, D. Konur, R. Qin, A. Gosavi, R. Wang, L. Pape, S. Agarwal, and R. D. Gottapu, "Flexible and Intelligent Learning Architectures for SoS (FILA-SoS): Volume 1 - Integrated Model Structure," Systems Engineering Research Center (SERC), Tech. Rep., 2015.

[8] C. Dagli, J. Colombi, N. Ergin, K. Giammarco, D. Enke, A. Gosavi, D. Konur, R. Qin, S. Agarwal, K. Haris, and L. Pape, "An Advanced Computational Approach to System of Systems Analysis & Architecting Using Agent-Based Behavioral Model," Systems Engineering Research Center, Tech. Rep., 2013. [Online]. Available: http://www.sercuarc.org/uploads/files/SERC Final Technical Report_2013-TR-021-3_RT-44c Phase 2_20131118.pdf

[9] A. Mour, C. R. Kenley, N. Davendralingam, and D. Delaurentis, "Agent-Based Modeling for Systems of Systems," no. 1991, p. 15, 2013.

[10] R. K. Garrett, S. Anderson, N. T. Baron, and J. D. Moreland, "Managing the Interstitials, a System of Systems Framework Suited for the Ballistic Missile Defense System," *Systems Engineering*, vol. 14, 2010.

[11] A. Chepko and O. de Weck, "A Modeling Framework for Applying Discrete Optimization to System Architecture Selection and Application to In-Situ Resource Utilization," *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, no. September, pp. 1–10, 2008. [Online]. Available: http://strategic.mit.edu/docs/3_114_AIAA-2008-6058-Optimal-ISRU.pdf

[12] N. Ricci, a. M. Ross, and D. H. Rhodes, "A Generalized Options-based Approach to Mitigate Perturbations in a Maritime Security System-of-Systems," *2013 Conference on Systems Engineering Research*, vol. 16, pp. 718–727, 2013.

[13] A. Alfaris, "The Evolutionary Design Model (EDM) For the Design of Complex Engineered Systems," Dissertation, Massachusetts Institute of Technology, 2009.

[14] D. S. Alberts, J. J. Garstka, and F. P. Stein, *DoD C4ISR Cooperative Research Program*, 1999. [Online]. Available: http://www.dodccrp.org/files/Alberts_NCW.pdf

[15] T. M. Schuck, "An Extended Enterprise Architecture for a Network-Enabled, Effects-Based Approach for National Park Protection - Transitioning Military Paradigms," *The International C2 Journal*, vol. 3, no. 2, 2009.

[16] J. R. Cares, R. J. Christian, and R. C. Manke, "Fundamentals of Distributed, Networked Military Forces and the Engineering of Distributed Systems," Naval Undersea Warfare Center (NAVSEA), Newport, Rhode Island, Tech. Rep. May, 2002.

[17] N. Boccara, *Modeling Complex Systems*. Springer Science & Business Media, 2010.

[18] J. C. Bohorquez, S. Gourley, A. R. Dixon, M. Spagat, and N. F. Johnson, "Common ecology quantifies human insurgency." *Nature*, vol. 462, pp. 911–914, 2009.

[19] M. Spagat, N. F. Johnson, J. a. Restrepo, O. Becerra, J. C. Bohórquez, E. M. Restrepo, and R. Zarama, "Universal patterns underlying ongoing wars and terrorism," pp. 1–69, 2006.

[20] G. S. Parnell, P. J. Driscoll, and D. L. Henderson, *Decision making in systems engineering and management*. John Wiley & Sons, 2011, vol. 81.

[21] B. S. Blanchard and W. J. Fabrycky, *Systems Engineering and Analysis*. Prentice Hall Englewood Cliffs, New Jersey, 2008, vol. 5.

[22] A. Singh and C. H. Dagli, "Using quality attributes and computational intelligence to generate and evaluate system architecture alternatives," *2010 IEEE International Systems Conference Proceedings, SysCon 2010*, pp. 347–352, 2010.

[23] INCOSE, "Guide to the Systems Engineering Body of Knowledge (SEBoK), version 1.3.1," 2014. [Online]. Available: http://sebokwiki.org/wiki/Main_Page

[24] R. L. Keeney and D. Von Winterfeldt, "A Value Model for Evaluating Homeland Security Decisions," *Risk Analysis*, vol. 31, no. 9, pp. 1470–1487, 2011.

[25] G. S. Parnell and T. E. Trainor, "Using the swing weight matrix to weight multiple objectives," *19th Annual International Symposium of the International Council on Systems Engineering, INCOSE 2009*, vol. 1, pp. 283–298, 2009. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-84878077732&partnerID=tZOtx3y1

[26] M. V. Cilli and G. S. Parnell, "Systems Engineering Tradeoff Study Process Framework," 2014.

[27] National Drug Intelligence Center, "The Economic Impact of Illicit Drug Use," pp. 1–123, 2011.

[28] S. Hyland, "The Shifting Terrain of Latin American Drug Trafficking," 2011. [Online]. Available: http://origins.osu.edu/article/shifting-terrain-latin-american-drug-trafficking

[29] I. A. Chindea, "Man, The State and War Against Drug Cartels: A Typology of Drug-Related Violence in Mexico," *Journal Article*, vol. March, 2014. [Online]. Available: http://smallwarsjournal.com/printpdf/15430

[30] R. M. Yeatman, "JIATF-South: Blueprint for Success," no. 42, pp. 26–27, 2006.

[31] C. R. Seelke, L. S. Wyler, and J. S. Beittel, "Latin America and the Caribbean: Illicit Drug Trafficking and U.S. Counterdrug Programs," Congressional Research Service, Tech. Rep., 2011.

[32] I. S. Davis, C. L. Worth, and D. W. Zimmerman, "A Theory of Dark Network Design," 2010.

[33] J. S. Beittel, "Mexico's Drug Trafficking Organizations: Source and Scope of the Violence," Congressional Research Service, Tech. Rep., 2012.

[34] I. S. Davis, C. L. Worth, and D. W. Zimmerman, "A Theory of Dark Network Design (Part Two): Type - I Dark Network: Opportunistic - Mechanical."

[35] R. W. Kurrle, "The Effective Business Practices of Mexican Drug Trafficking Organizations (DTOs)," Thesis, Naval Postgraduate School, 2013.

[36] M. Kenney, "The Architecture of Drug Trafficking: Network Forms of Organisation in the Colombian Cocaine Trade," *Global Crime*, vol. 8, no. 3, pp. 233–259, 2007.

[37] P. Corcoran, "Mexico's shifting criminal landscape: Changes in gang operation and structure during the past century," *Trends in Organized Crime*, vol. 16, pp. 306–328, 2013.

[38] United States Government Accountability Office, "Coast Guard Resources Provided for Drug Interdiction Operations in the Transit Zone, Puerto Rico, and the U.S. Virgin Islands," Tech. Rep. June, 2014.

[39] A. Ilachinski, "Land Warfare and Complexity, Part I: Mathematical Background and Technical Sourcebook," Center for Naval Analyses, Alexandria, VA, Tech. Rep., 1996.

[40] S. A. Kauffman, "Antichaos and adaptation." *Scientific American*, vol. 265, pp. 78–84, 1991.

[41] K. J. Thomas, "A Model for Success in the Drug War," 2011. [Online]. Available: http://leb.fbi.gov/2011/february/a-model-for-success-in-the-drug-war

[42] United Nations Office on Drugs and Crime, "World Drug Report 2014," New York, Tech. Rep., 2014.

[43] E. O. Williams, "Surveillance and interdiction models: a game-theoretic approach to defend against VBIEDS," Thesis, Naval Postgraduate School, 2010.

[44] R. Wood, "Deterministic Network Interdiction," *Mathematical and Computer Modelling*, vol. 17, no. 2, pp. 1–18, 1993.

[45] K. J. Cormican, D. P. Morton, and R. K. Wood, "Stochastic Network Interdiction," *Operations Research*, vol. 46, no. 2, pp. 184–197, 1998.

[46] R. L. Steinrauf, "Network Interdiction Models," Thesis, Naval Postgraduate School, 1991.

[47] E. Shieh, B. An, R. Yang, and M. Tambe, "PROTECT: A deployed game theoretic system to protect the ports of the United States," *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, vol. 1, pp. 13–20, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2343578

[48] J. Pita, M. Jain, C. Western, C. Portway, M. Tambe, F. Ordonez, S. Kraus, and P. Parachuri, "Depoloyed ARMOR Protection: The Application of a Game-Theoretic Model for Security at the Los Angeles International Airport," *AAMAS-08*, 2008.

[49] A. R. Washburn, *Two-person Zero-sum Games*. INFORMS, 2003.

[50] F. Pan, "Stochastic Network Interdiction: Models and Methods," Dissertation, University of Texas at Austin, 2005.

[51] A. R. Washburn, *Search and Detection*, 3rd ed. INFORMS, 1996.

[52] J. R. Frost, "The theory of search: a simplified explanation," Soza & Company, Ltd. and U.S. Coast Guard Office of Search and Rescue, Fairfax, VA, Tech. Rep., 1996.

[53] H. R. Richardson, "Search theory," Alexandria, VA, 1986.

[54] K. L. Ruegger, "Architecting a Net-Centric Operations System of Systems for Multi-Domain Awareness," Thesis, Naval Postgraduate School, 2008.

[55] J. B. Zorn, "A Systems Engineering Analysis of Unmanned Maritime Systems for U.S. Coast Guard Missions," Thesis, Naval Postgraduate School, 2013.

[56] S. P. Hayes and E. P. Paulo, "Architecture for comparing alternative designs of a tactical naval command and control system using discrete-event simulation," *Proceedings - Winter Simulation Conference*, pp. 1817–1822, 2009.

[57] W. H. Bong, "Application of Systems Engineering to Understand and Replicate Interagency Coordination in Support of Combatant Commands," Thesis, Naval Postgraduate School, 2012.

[58] B. Abeto, "An Application of Systems Engineering to Analyze the Interagency Coordination Aspect of Counter Trafficking and Terrorism," Thesis, Naval Postgraduate School, 2012.

[59] INCOSE, *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 3rd ed., San Diego, CA, 2012.

[60] M. W. Maier, *The Art of Systems Architecting*. CRC press, 2011.

[61] U.S. DoD CIO, "DoD Architecture Framework Version 2.02," pp. 1–2, 2013. [Online]. Available: http://dodcio.defense.gov/Library/DoDArchitectureFramework.aspx

[62] A. P. Sage and C. D. Cuppan, "On the Systems Engineering and Management of Systems of Systems and Federations of Systems," *Inf. Knowl. Syst. Manag.*, vol. 2, no. 4, pp. 325–345, 2001.

[63] J. S. Dahmann, G. Rebovich, and J. A. Lane, "Systems Engineering For Capabilities," *CrossTalk*, 2008. [Online]. Available: http://www.dtic.mil/dtic/tr/fulltext/u2/a489896.pdf

[64] H. Kopetz, "System-of-Systems Complexity," *arXiv preprint*, vol. arXiv:1311, pp. 35–39, 2013. [Online]. Available: http://arxiv.org/abs/1311.3629v1

[65] Y. Bar-Yam, "About engineering complex systems: Multiscale analysis and evolutionary engineering," *Proc. International Product Focused Software Development and Process Improvement conference*, pp. 16 – 31, 2005. [Online]. Available: http://www.springerlink.com/index/m46m15r22vf04x9k.pdf

[66] Z. Fang and D. DeLaurentis, "Dynamic Planning of System of Systems Architecture Evolution," *Procedia Computer Science*, vol. 28, pp. 449–456, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050914001185

[67] A. Meilich, "System of Systems (SoS) Engineering & Architecture Challenges in a Net Centric Environment," in *Proceedings of the 2006 IEEE/SMC International Conference on System of Systems Engineering*, 2006, pp. 1–5.

[68] R. Kewley, J. Cook, N. Goerger, D. Henderson, and E. Teague, "Federated Simulations for Systems of Systems Integration," in *Proceedings of the 2008 Winter Simulation Conference*, 2008.

[69] W. C. Baldwin, B. Sauser, and R. Cloutier, “Simulation Approaches for System of Systems: Events-based versus Agent Based Modeling,” *Procedia Computer Science*, vol. 44, pp. 363–372, 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1877050915002689

[70] C. H. Dagli, “Flexible and Intelligent Learning Architectures for SoS (FILA-SoS),” Los Angeles, CA, pp. 1–65, 2014.

[71] M. W. Maier, “Architecting Principles for Systems-of-Systems,” *Systems Engineering*, vol. 1, pp. 267–284, 1998.

[72] United States Southern Command, “Countering Transnational Organized Crime.” [Online]. Available: http://www.southcom.mil/ourmissions/Pages/Countering Transnational Organized Crime.aspx

[73] United States Government Accountability Office, “Agencies Need to Plan H ighlights,” United States Government Accountability Office, Tech. Rep. GAO-06-200, 2005.

[74] White House Office of National Drug Control Policy, “Transit Zone Operations.” [Online]. Available: http://www.whitehouse.gov/ondcp/transit-zone-operations

[75] United States Government Accountability Office, “Drug Control: Cooperation with Many Major Drug Transit Countries Has Improved, but Better Performance Reporting and Sustainability Plans Are Needed,” Tech. Rep. July, 2008.

[76] United Nations Office on Drugs and Crime, “Criminal Intelligence Manual for Analysts,” Tech. Rep., 2011.

[77] S. Campos III, “An Analysis of Mathematical Models to Improve Counter-Drug Smuggling Operations,” Thesis, Naval Postgraduate School, 2014.

[78] B. Ramirez and R. J. Bunker, “Narco-Submarines: Specially Fabricated Vessels Used For Drug Smuggling Purposes,” Tech. Rep., 2014.

[79] M. E. Bailey, “Counter-narcotic acoustic buoy (CNAB),” *Proceedings of SPIE*, vol. 5417, pp. 109–115, 2004. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=844030

[80] L. Hardesty, “Underwater robot for port security,” 2014. [Online]. Available: http://news.mit.edu/2014/underwater-robot-for-port-security-0926

[81] S. A. Kauffman and S. Johnsen, “Coevolution to the edge of chaos: coupled fitness landscapes, poised states, and coevolutionary avalanches.” *Journal of Theoretical Biology*, vol. 149, pp. 467–505, 1991.

[82] C. H. Dagli, A. Singh, J. P. Dauby, and R. Wang, “Smart Systems Architecting: Computational Intelligence Applied To Trade Space Exploration and System Design,” *Systems Research Forum*, vol. 03, no. 2, pp. 101–119, 2009.

[83] K. Giammarco, "An Advanced Computational Approach to System of Systems Analysis & Architecting Using Agent-Based Behavioral Model," no. March, pp. 1–157, 2013. [Online]. Available: http://www.sercuarc.org/uploads/files/SERC Final Technical Report_2013-TR-021-3_RT-44c Phase 2_20131118.pdf

[84] A. Ilachinski, "Complex Adaptive Systems, Multiagent-based Models, and some Heuristics Regarding Their Applicability to URW," pp. 199–216.

VITA

George Muller is an engineer at the U.S. Department of Energy's Pacific Northwest National Laboratory. He applies his knowledge of systems modeling and analysis, optimization, and operational simulation to challenges ranging from detecting nuclear smuggling to epidemiological modeling. He has experience in diverse problem sets, including modeling pandemic influenza spread, assessing international air passenger screening, and operational modeling of nuclear materials detection. Prior to becoming an engineer, he was a cavalryman, carpenter, groundskeeper and farmhand. He earned his M.S. in Systems Engineering from Missouri University of Science and Technology in May 2016 and his B.S. in Industrial Engineering from Southern Illinois University Edwardsville in May 2007.