ACQUISITION INNOVATION
RESEARCH CENTER

Report on

Data-Driven Capability Portfolio Management Pilot

EXECUTIVE SUMMARY AND REPORT
OCTOBER 2022

PRINCIPAL INVESTIGATOR:
Dr. Daniel A. DeLaurentis, *Purdue University*

CO-PRINCIPAL INVESTIGATOR:
Dr. Jitesh H. Panchal, *Purdue University*

SPONSOR:
Ms. Nickee L. Abbott,
Director and Principal Advisor of Acquisition Intelligence Division, Strategic Advisor for Mission Engineering and Integration, DASD(AE), OUSD(A&S)
OFFICE OF THE UNDER SECRETARY OF DEFENSE FOR ACQUISITION AND SUSTAINMENT

RESEARCH TEAM

NAME	ORG.	LABOR CATEGORY
Daniel DeLaurentis	Purdue	Principal Investigator, Professor
Jitesh Panchal	Purdue	Co-Principal Investigator, Professor
Cesare Guariniello	Purdue	Research Scientist
Kshitij Mall	Purdue	Post-Doctoral Associate
Kris Ezra	Purdue	Research Scientist
Robert Campbell	Purdue	Lead Software Engineer
Clint Hanthorn	Purdue	Software Engineer
Joshua Fitch	Purdue	Graduate Research Assistant
Waterloo Tsutsui	Purdue	Senior Research Associate
Paul Grogan	Stevens	Assistant Professor, Director of the Collective Design Lab
Brian Chell	Stevens	Post-Doctoral Associate
Benjamin Stanley	Stevens	Graduate Research Assistant
Corey Batchelder	Stevens	Undergraduate Research Assistant
Daniel Browne	GTRI	Chief, Systems Engineering Research Division
Craig Arndt	GTRI	Mission Engineering Expert
Frank Patterson	GTRI	Senior Research Engineer
Santiago Balestrini	GTRI	Senior Research Engineer
Zach Welz	GTRI	Research Engineer
James Arruda	GTRI	Research Engineer
Joel Stansbury	GTRI	Research Scientist
Nicholas Bollweg	GTRI	Senior Research Engineer
Valerie Sitterle	GTRI	Principal Research Engineer
Stoney Trent	Virginia Tech	Principal advisor for research and innovation
Kurt Luther	Virginia Tech	Associate Professor
Hanumanthrao (Rao) Kannan	Virginia Tech	Assistant Professor
Justin Krometis	Virginia Tech	Research Assistant Professor
Peter Korfiatis	MITRE	Department Head
Joshua Mathews	MITRE	Systems Engineer
Lucas Karinshak	MITRE	Associate Systems Engineer

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

AIRC research under this task adapted a previously developed system-of-systems analytic workbench (SoS-AWB) of analytic tools to create a decision-support prototype effective for informing decisions in Integrated Acquisition Portfolio Reviews (IAPRs). Mission engineering analysis and portfolio optimization of an anti-surface warfare (ASuW) mission thread using personnel and munitions in the surface, aviation, and space domains was demonstrated via the prototype software on September 8, 2022. These advanced prototypes provide a broader range of insights (e.g., resource tradeoffs, cost-sensitivity analysis, and the most robust ASuW systems to be acquired in specific portfolios) for stakeholder decision making. Future work could improve the tools to identify: how risk aversion affects portfolio optimization, technical dependencies among systems, developmental dependencies, and portfolio performance effects from stakeholder decisions.

The following four aspects of the developed prototype are distinguishing features accomplished in the research in support of the sponsor's needs.

Portfolio-Centric Approach: The Department of Defense (DoD) has an increasing focus on Mission Engineering (ME) analysis and architecture development for modernization decisions, including investments and prioritization related to requirement development and selection of capabilities to support various concepts of employment and technological improvements. Typically, however, systems engineering tools focus on the system itself. That is, the tools may not translate the complexities of mission engineering analysis into the evaluation in a way that is both (a) meaningful to the requirements within the trade space of capabilities and (b) flexible, scalable, and configurable to integrate with other analyses. To this end, recommendations from an advisory panel suggested that the DoD approach should take a more holistic and portfolio-centric method for acquisitions rather than the current program-centric approach. In our prototype, systems and technologies are evaluated within an overall portfolio, exposing how each component plays a role in the realized capability while connecting the mission needs of warfighters with acquisition decisions. Continued development along these lines will eventually pave the way for the establishment of Acquisition Integration and Interoperability (AII), which should be based on mission and digital engineering, using data-driven methods.

Higher Fidelity Analysis through Discrete Event Simulation: It is imperative to evaluate the trades associated with architecting a mission in a model-based scenario that includes not just the blue systems but also the red systems and a specific mission context. To evaluate and compare successful portfolios of platforms and their associated capabilities, we must identify which portfolio concepts can be successful. Success is measured through scenario modeling in terms of performance in the context of a mission–performance against what and to accomplish what objectives. The Universal Platform for Simulating Tasks and Actors with Graphs and Events (UPSTAGE) was created specifically to help DoD stakeholders make decisions with respect to the design of Concepts of Employment (CONEMPs), force placement decisions, and investment of resources prior to kinetic action to achieve the desired effects across disparate potential scenarios. While multiple large-scale frameworks are available for simulations of tactics and operations, these frameworks implement higher fidelity representations that take significant time to set up and modify. Often, entity behaviors in these simulations are programmed at such a high level of fidelity that they are not reusable in other scenarios. To this end, UPSTAGE is different since it is a multi-resolution, hybrid simulation and wargaming framework to support quick-turn operational scenario analyses against evolving, competitive adversaries. UPSTAGE was designed to enable rapid explorations of CONEMPs, optimization, scenario re-definition, and other activities required for agile and iterative problem-solving. With UPSTAGE in our software prototype, multiple scenarios are more rapidly customizable and executable.

Space Domain Technology Injection: New technologies always emerge; thus, we must always find a way to incorporate the new technologies to our benefit. To this end, we focused on the injection of new technologies into an IAPR. More specifically, the new technology injection evaluates space surveillance domain assets (i.e., satellites) for an anti-surface warfare scenario. Two different systems are used in the IAPR: (a) older large satellites and (b) newer small satellites (i.e., smallsats). The former is legacy satellites and is generally larger, more capable, and more expensive. On the other hand, the latter is the new technology smallsats, which are generally newer, smaller, less capable, and less expensive. These two contrasting systems are placed into constellations consisting of several to many spacecrafts. Then, their ability to surveil the region of interest is evaluated with the Tradespace Analysis Tool for Constellations (TAT-C). TAT-C is an open-source mission engineering tool developed for these early design phase analyses. The legacy and smallsat systems are chosen to represent the types of systems which will be selected in the IAPR and have characteristics that show meaningful tradeoffs. The new technology injection workflow is integrated with the larger WRT-1049.5 toolset. The legacy and smallsats are evaluated with TAT-C, and these metrics are introduced into the AWB, which is then used to perform an RPO routine. The Pareto-optimal solutions to this routine demonstrate how decision-makers can obtain the best information for acquisitions in a portfolio-centric manner, including new technologies.

Anti-Surface Warfare Problem: Our adapted SoS-AWB prototype was successfully demonstrated to inform decisions in Integrated Acquisition Portfolio Reviews (IAPRs). More specifically, we demonstrated the use of SoS-AWB via mission engineering analysis and portfolio optimization of anti-surface warfare (ASuW) mission thread using personnel and munitions in the surface, aviation, and space domains. These advanced prototypes provide a broader range of insights (e.g., resource tradeoffs, cost-sensitivity analysis, and the most robust ASuW systems to be acquired in specific portfolios) for stakeholder decision-making. Our demonstration illustrated 1) what the integrated acquisition decision-support process looks like and 2) how provided data utilizes a model-based decision tool to produce insights on the most attractive ASuW systems to be acquired in specific portfolios. Our findings indicate that these approaches indeed do provide the stakeholders with a broader range of more accessible information, such as resource tradeoffs and cost sensitivity analysis. As a result, the stakeholders can incorporate the integrated acquisition process to improve ASuW mission engineering and defense acquisition performance. Future work could improve the tools to identify: how risk aversion affects portfolio optimization; technical dependencies among systems; developmental dependencies; and portfolio performance effects from stakeholder decisions.

INTRODUCTION

We developed a pilot/prototype capability to enhance data-driven decision-making regarding acquisition and sustainment programs, motivated by the context of the DoD's Integrated Acquisition Portfolio Review (IAPR) process. As DoD transforms its acquisition paradigm from centralized oversight of ACAT 1D programs to decentralized oversight delegated across Components, the Office of the Under Secretary of Defense for Acquisition & Sustainment (hereinafter referred to as OUSD(A&S)) must likewise shift its focus from traditional program oversight to enabling acquisition innovation and managing a portfolio of capabilities. OUSD(A&S) has made significant strides in acquisition innovation through the rollout of the Adaptive Acquisition Framework and Capability Portfolio Management. However, it has not fully realized the analytic capability necessary to underpin acquisition investment decisions with clear traceability to warfighter requirements.

As the WRT-1049.5 project title, Data-Driven Capability Portfolio Management Pilot, suggests, this research focuses on a portfolio-centric approach, which we implemented by enhancing and adapting an existing research product called the SoS Analytic Workbench (AWB). The SoS-AWB consists of several SoS tools, the primary of which are Robust Portfolio Optimization (RPO), Systems Operational Dependency Analysis (SODA), and Systems Developmental Dependency Analysis (SDDA). A significant part of the WRT-1049.5 research effort described herein included enhancements to the AWB elements. More specifically, we upgraded the scripts and functions representing the various AWB elements to a set of qualified Python packages. These upgrades enabled ease of continued development of the packages and their capabilities. The upgrades also made the components of AWB more friendly for both developers and users while also easing any future burden of transitioning the tools to the sponsor and their designees.

Research under WRT-1049.5 also explored the enablement of portfolio management from a mission engineering perspective. The two guiding principles for this subtask are 1) the demonstration of the viability of the Mission Engineering (ME) approach to support Joint acquisition decision-making and 2) the initiative for the development of a reusable Digital Engineering environment and methodology to support future Mission Engineering pilots, studies, and acquisition analyses. Furthermore, the subtask explored the transition from a paper-based (PowerPoint modality) review of various portfolios (e.g., EW Portfolio, NC3 Portfolio, or ASuW Portfolio) to a more model-based review of the portfolios, addressing questions such as: What form should this take? What information is key from a leadership perspective? How do we ensure a holistic review without being overwhelmed with complexity and information? This subtask included engagement with selected mission portfolio managers to understand their priorities and challenges to enable evidence/data-based portfolio management.

This report starts with a brief explanation of Analytic Workbench (AWB) development, wherein we describe the overall description of the AWB tools, followed by a discussion on the development of the AWB tools. The report also talks about higher fidelity analysis through Discrete Event Simulation (DES), a technique we used in the research. The report also discusses the injection of new technology. We chose to use space-domain applications (i.e., satellites) as we felt the need that space-domain applications are a crucial part of ME. Finally, the report concludes with the demonstration of the enhanced AWB in the anti-surface warfare (ASuW) domain.

ANALYTIC WORKBENCH DEVELOPMENT

1.1 THE SOS ANALYTIC WORKBENCH

The SoS Analytic Workbench (AWB) is a collection of methods and techniques that have been developed by researchers at Purdue University within SERC projects starting in 2011. Due to the complex and multifaceted nature of SoS modeling and analysis, the most effective approach is to develop different methodologies, each addressing one specific aspect of SoS, for example, emergence due to interactions or portfolio-wide considerations. The AWB implements this approach by providing a set of tools developed on purpose for modeling and analysis of SoS.

The AWB addresses complexities associated with interconnections that exist across physical, functional, and developmental SoS hierarchies. The idea is to support the "top-down integration, bottom-up implementation" paradigm at the SoS level. The analytical tools in the workbench account for the complex and highly interconnected nature of the systems that constitute the overall SoS. The analytical tools allow the user to:

- Quantify performance and risk for individual systems, links, and of overall SoS;
- Assess the impact that changes to SoS architecture (add/remove links and/or nodes) will have; and
- Quantitatively identify optimal sets of architectural solutions given constraints on cost, performance, and risk.

In general, when building tools to support decision-making in an SoS environment, the challenge is that such tools must address the technical and programmatic complexities of SoS, and yet remain domain-agnostic. It is up to researchers to find the appropriate balance between the need for tools that can be used on a broad spectrum of applications in various fields and the need for tools that can be easily tailored to specific applications and user requirements.

This project focused on three tools from the AWB: Robust Portfolio Optimization (RPO), Systems Operational Dependency Analysis (SODA), and Systems Developmental Dependency Analysis (SDDA). Figure 1 shows the inputs and outputs of tools in the Analytic Workbench and in the Decision Support Framework (DSF), the first framework that used RPO and SODA sequentially.

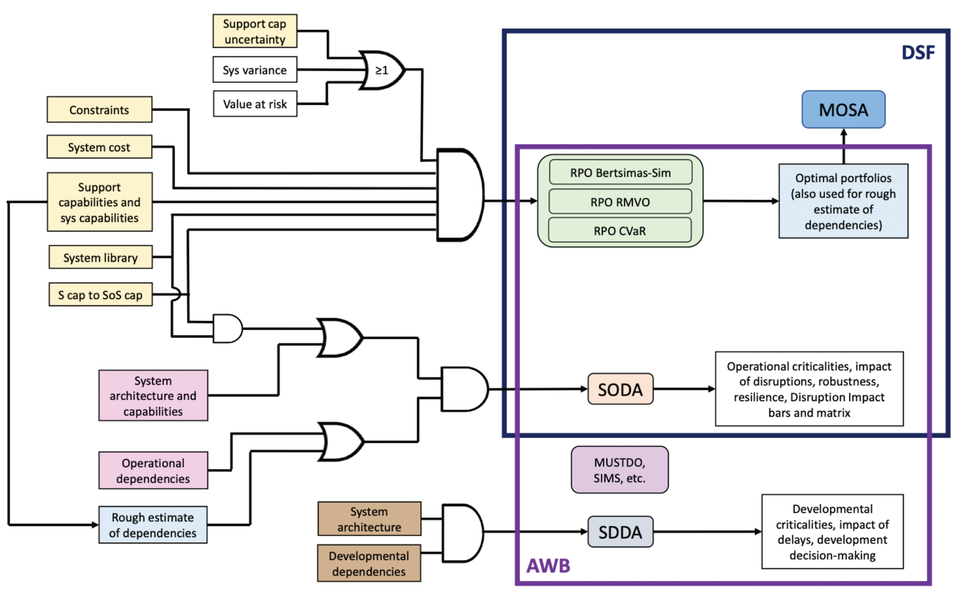

Figure 1 Inputs to AWB and DSF

Robust Portfolio Optimization (RPO)

The SoS modeling and analysis problem can sometimes be described as a combinatorial problem to determine the most promising portfolio of individual systems in which to invest to achieve a certain capability. For instance, in space systems architecture, mission designers need to find the best combination of spacecraft, launch systems, launch windows, commodities to be transported, existing space systems such as the International Space Station (ISS), and other capabilities to achieve the goal of a long journey and possible settlement on other space bodies. The process of selecting the optimal portfolio considers the Life Cycle Cost (LCC) and the capability of individual systems. The process keeps both cost and certain types of risk (e.g., developmental cost) within an acceptable level while accounting for the impact of various forms of data uncertainty. Implementation and solution of RPO for a particular SoS design problem yields a set of Pareto optimal solutions (each solution is a portfolio of systems) corresponding to a user-defined risk aversion factor. RPO analysis allows an SoS manager to explore the design space of available options when designing a mission. Additionally, for a chosen portfolio, further desired analysis can be carried out using other AWB tools.

The risk can be characterized into three types: developmental, operational, and simulated. Based on these three types of risks, three flavors of RPO have been developed. Robust Mean Variance Optimization (RMVO) includes developmental risks, the Bertsimas-Sim method involves operational risks, while Conditional Value at Risk (CVaR) addresses simulated risks. Based on the problem at hand, the stakeholder needs to select a specific flavor of RPO.

Systems Operational Dependency Analysis (SODA)

SODA methodology addresses the operational domain of an SoS by providing an analysis of the impact of dependencies between constituent systems on the propagation of the effect of disruptions (Guariniello et al.). In SODA, a parametric model of system behavior is combined with a network representation for the system architecture. A small set of parameters is used to simplify the dependencies between each system. These parameters were chosen to represent aspects of the dependency of the operability of a system on the operability of another system. The Strength of Dependency (SOD) represents a linearized operational dependency between systems in the case of minor disruptions. The Criticality of Dependency (COD) represents the loss of operability due to major disruptions. The Impact of Dependency (IOD) models the boundary between the small disruption regime and the major disruption regime.

Based on the parameters of the model, SODA can quantify the cascading effect of disruptions in the architecture and constitutes a quantitative method of risk analysis that can be used to expand the traditional risk matrix. The algorithm can also model partial failures, both deterministic and stochastic, and multiple paths of propagation within the model. SODA thus provides early-stage feedback for the architecture's design, reducing the amount of simulation and other verification methods required to ensure mission feasibility and to identify criticalities and areas of potential emergent behavior (Guariniello et al.).

Systems Developmental Dependency Analysis (SDDA)

SDDA is the counterpart of SODA in the developmental domain. It is a parametric model of developmental dependencies and constitutes an extension of PERT/CPM techniques which adds partial parallel development and partial dependencies.

The outcome of SDDA modeling and analysis is a quantitative assessment of the beginning and completion time of activities in a project (e.g., development of technologies, systems, or SoS capabilities), accounting for the combined effect of multiple developmental dependencies and of possible delays in the development of predecessors. The lead time (i.e., the amount of time by which a system can begin to be developed before a predecessor is fully developed) is calculated based on the dependencies and the performance of predecessors.

SDDA allows for deterministic or stochastic analysis. In deterministic analysis, an amount of delay is assigned to each system, and SDDA evaluates the resulting schedule. In stochastic analysis, the amount of delay in each system follows a probability density function with the resulting beginning and completion time of each system also as a distribution. SDDA identifies the most critical nodes and dependencies with respect to overall development time and delay propagation, important decision support for both system managers and the SoS architect. Results from the analysis are used to compare different architectures in terms of development time, risk, and capability of absorbing delays.

1.2 ANALYTIC WORKBENCH INTEROPERABILITY, EXTENSIBILITY, AND USABILITY UPGRADES

A significant part of this research effort included upgrades to the Analytic Workbench (AWB). The original scripts and functions representing the various elements of AWB were upgraded to a set of qualified Python packages. This process included the implementation of industry-standard software control and revision processes and standards. These upgrades enable ease of continued development of the packages and their capabilities across academic, industry, and government teams. The upgrades also make the components of AWB more friendly for developers and users and ease any future burden of delivery.

A summary of the upgrades for the different components is outlined here:

- Robust Portfolio Optimization (RPO): RPO was upgraded to a fully Python-based application, removing the need for a MATLAB license. A set of input and output data control and validation methods was provided for interaction with RPO. RPO was also integrated into a controlled Python product with available pip and Anaconda packages. This process included the addition of unit and integration testing, static code analysis, and implementation of CI/CD. The input for RPO was converted into a compact text-based file format called JavaScript Object Notation (JSON) from a Microsoft Excel datasheet. Furthermore, Jupyter Notebooks were used for adding the input data, running the Python-based code, and analyzing the results on a webpage in an interactive manner.
- Systems Operational Dependency Analysis (SODA): SODA was integrated into a controlled Python product with available pip and Anaconda packages. This process included the addition of unit and integration testing, static code analysis, and implementation of CI/CD.
- Systems Developmental Dependency Analysis (SDDA): SDDA was integrated into a controlled Python product with available pip and Anaconda packages. This process included the addition of unit and integration testing, static code analysis, and implementation of CI/CD.
- AWB: AWB was integrated into a controlled Python product with available pip and Anaconda packages. Applicable automation was developed for AWB to ease the control and installation of necessary dependencies (i.e., RPO, SODA, and SDDA) across platforms. This process included the addition of unit and integration testing, static code analysis, and implementation of CI/CD.

Several other specific upgrades were implemented to make it easier for users to develop appropriate data to define the problems AWB is meant to address. These upgrades also support the ongoing work to develop an appropriate user interface (UI) for building AWB problem data. In particular, the team also improved the RPO package by reimplementing the code surrounding the data and optimizer handling logic. This included changes to the interface used to connect to RPO. The team made RPO features functional again within a UI in Python so that users can pass parameters to RPO, and RPO output plots and results can be displayed. The whole AWB UI allows the selection of tools such as RPO, SODA, SDDA, etc. Custom images or logos are displayed based on the tool selected. Input to the tool can be a built-in example, custom-build scenario (TBD), or file import. The RPO tool has tabs for setup and output tabs which allows for input selection, input setup, and display analysis output. SODA and SDDA have an Interactive tab with functionality that will be implemented in the future.

For all AWB tools, a Links widget will allow dependencies between nodes to be defined. A directed graph is used to show the dependencies between systems. More options for SODA and SDDA can be selected.

The RPO Output tab plots Pareto frontiers for cost vs. SoS performance index. There is also a table of allocations that shows the numbers of individual assets at each cost point. The SODA Output tab can show either repair impact or failure impact plots. SDDA output shows the resulting schedule of development based on SDDA analysis.

A side effect of the new RPO updates was the breaking of a convenience feature within DSF which allowed for quickly using RPO results as input into SODA. Here, a user-selected RPO allocation is "automatically" fed into SODA without parameter adjustment by the user. Therefore, further work was done to reconnect RPO output to SODA analysis. This enables SODA output plots to appear in addition to RPO output within the new UI.

1.3 RPO DATA VALIDATION OVERVIEW

The RPO software requires users to create instances of code classes (e.g., System, Capability) that have unique data requirements. Validating data is a common challenge in software development. The GTRI team has approached this problem by using JSON-based schema to capture information about data requirements so that only valid inputs are used to run scenarios within the RPO tooling. JSONSchema (a standard for developing these validations) is used. Generating the JSONSchema for all classes in the RPO library would require significant manual effort to create an additional effort to update each time the class definitions (i.e., data models) are modified. To alleviate this, all RPO data models are defined using a Python standard library (data classes), which can be used to automate the creation of JSONSchema for each class. This allows continuous updating of the schema for validating instances of objects against the expected representation without requiring a manual definition of the schema.

1.4 SCHEMA GENERATION SCRIPT

While the schema for class instances is not expected to change frequently, RPO needs a method to automate the process. To aid developers, a "generate_schema" script is included in the RPO scripts folder. Schema is checked into the repo to ensure that any changes undergo review by SERC developers. Once the script is called, users can commit changes, and all validation changes will propagate to the RPO tests. Invocation documentation can be found in the RPO README.

1.5 AUTOMATION OF SODA/SDDA DATA FROM RPO PROBLEM

A previous algorithm existed for converting the inputs used in RPO into those used in SODA and SDDA. Namely, the three dependency characteristic matrices of Strength of Dependency (SOD), Criticality of Dependency (COD), and Impact of Dependency (IOD). This pipeline utilizes the system support requirements and outputs to identify potential relationships among systems. For example, if system A produces resource R, and system B requires resource R, then B is assumed to depend on A. Similarly, we also capture relationships among systems and capabilities. Namely, if a system A has capability C, then C is said to depend on A. In reality, it is also possible for a system to require a capability as well as a capability to require another capability, but this information is not possible to deduce from the inputs of RPO alone. Furthermore, while the algorithm is designed to provide sensible values for each relationship's strength, criticality, and impact, it is generally accepted that expert opinion is necessary to achieve reliable results from SODA and SDDA analysis.

An expansion of this algorithm to approximate the inputs to SODA/SDDA has been designed as a proof-of-concept. The desire is to display approximated parameters to the user via an interactive data entry widget which will also provide the ability to correct them as needed. This project included the final stages of automating the algorithm and integrating it with the data structures used to run the Python implementation of RPO. An initial implementation of the data entry widget is being developed in parallel.

HIGHER FIDELITY ANALYSIS THROUGH DISCRETE EVENT SIMULATION (DES)

The DoD has an increasing focus on the concept of Mission Engineering, analyses, and architecture development for modernization decisions, including investments and prioritization related to requirements development and selection of capabilities to support various concepts of employment. Typically, however, systems engineering tools focus on the system itself. They do not bring the complexities of mission engineering analyses into the evaluation in a way that is (a) meaningful to the requirements understanding and tradespace of capabilities and (b) in a flexible, scalable, relatively rapidly configurable manner to integrate with other analyses at the 'speed of work.'

Pointedly, this requires evaluating the trades associated with architecting a mission in a model-based scenario that includes not just the blue systems but also the red systems and a specific mission context. To evaluate and compare successful portfolios of platforms and their associated capabilities, we must identify which portfolio concepts can be successful. Success is measured through scenario modeling in terms of performance in the context of a mission – performance against what and to accomplish what objectives.

GTRI's Universal Platform for Simulating Tasks and Actors with Graphs and Events (UPSTAGE) was created specifically to help DoD stakeholders make decisions with respect to the design of Concepts of Employment (CONEMPs), force placement decisions, and investment of resources prior to kinetic action to achieve the desired effects across disparate potential scenarios. While there are multiple large-scale frameworks available for simulations of tactics and operations, these frameworks implement higher fidelity representations that take significant time to set up and modify. Often, entity behaviors in these simulations are programmed at such a high level of fidelity that they are not reusable in other scenarios. UPSTAGE is different. It is a multi-resolution, hybrid simulation and wargaming framework to support quick-turn operational scenario analyses against evolving, competitive adversaries. UPSTAGE was designed to enable rapid explorations of CONEMPs, optimization, scenario re-definition, and other activities required for agile and iterative problem-solving. In the UPSTAGE environment, multiple scenarios are more rapidly customizable and executable.

These characteristics make UPSTAGE a preferred companion framework to support the capability portfolio analysis goals of this effort. UPSTAGE can provide enhanced, more realistic and mission-grounded dimensions of analytic capability as part of the overall system of integrated tools and approaches. This improves the tactical and operational relevance of the integrated analyses that will inform acquisition investment decisions and offers improved traceability to warfighter requirements.

UPSTAGE is intended to facilitate the development of discrete event simulation of complex systems using open-source efficient and scalable technologies. It is built atop SimPy, a bare-bones open-source discrete event simulation framework built in Python. The benefit of SimPy is that it does not require other software. It is a simple and minimalist framework that allows for extending and customizing. The challenge with using it is when the complexity of the model grows, handling interruptions and interactions between entities and events becomes increasingly more difficult. UPSTAGE was developed to try to address these by building helper methods and modeling constructs that help manage the complexity while allowing modelers to focus on value-added work rather than the overhead of developing discrete event simulations. This is not to say that modelers don't have to understand discrete event simulation, but for those that do and understand the approach proposed by UPSTAGE, they can reduce the amount of time spent checking the state of the simulation does not diverge into unrealistic conditions and can help the actors use rehearsal to make predictions of their future state, enabling more complex forward-looking decision making.

To illustrate how SimPy can be used and extended with UPSTAGE, the authors will employ a common example throughout the paper. This example is depicted in Figure 2. The figure describes (at a high level) a rescue system of systems (SoS). The SoS is comprised of a base with certain resources, from which rescue units operate to perform a certain mission in a mission area and to which ambulances arrive to heal patients. A resupply agent provides the base with resources in discrete intervals of time. The base has limited amounts of fuel, repair equipment, and crews for the helicopters. The question then becomes how to handle the development of the framework as stakeholders update the requirements and complexity of the behavior of the agents. The technical approach section will address how UPSTAGE improves this process.

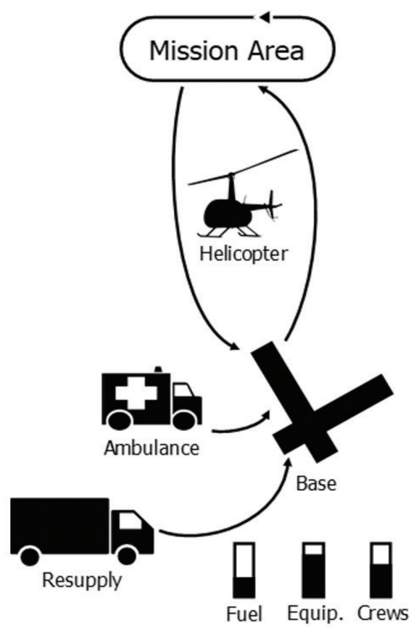

Figure 2 Example UPSTAGE problem

As the simulation's complexity increases, it is almost a certainty that an actor's tasking will have to be interrupted between scheduled events. As an example, Figure 3 depicts as an example one of the helicopters being re-tasked for a rescue mission after a higher priority mission arises after it took off to take care of the first mission objective. In order to model an interruption, the modelers must write code that: (1) updates the states of the actor, (2) sends out communications, and (3) implements whatever new tasking is called for. The additional code tends to be written ad-hoc for each task that needs to be interrupted and oftentimes is re-done in different steps of the task (e.g., interrupting on the outbound vs. inbound leg of the mission). This additional code required is a source of bugs, increases code length and complexity, and reduces the readability of the model's "business logic."

An additional concern of complex simulations is the need to assess throughout the simulation which agents (i.e., actors) can perform which tasks. For example, a planning agent will need to know which agents are good candidates for performing a given mission. This is often crucial when assigning tasks to actors by a planning agent. As the complexity of the model increases, the complexity of the tasks and the criteria that need to be assessed to determine if an actor can perform given tasks based on their current or projected state increases. The "don't repeat yourself" (or DRY) principle2 from software engineering is a valuable concept to keep in mind when developing a framework. When using SimPy, modelers tend to write methods to (1) define actor behavior and (2) determine if the actor can perform the behavior. A framework that allows modelers to use the same code to perform the task and assess an actor's ability to perform the task will reduce bugs and reduce code complexity. This is not a trivial problem because the state of the real simulation cannot be polluted by the "rehearsed" tasks.

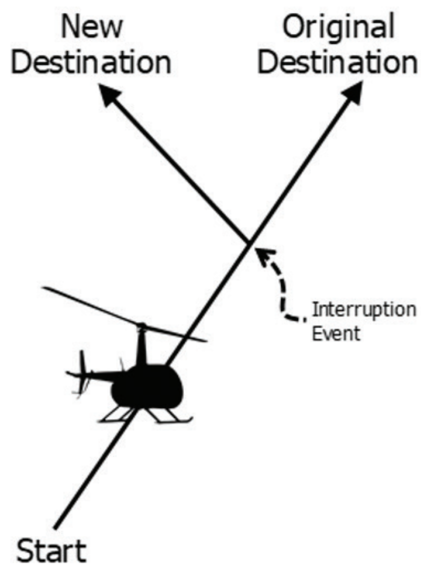

Figure 3 Travel change a task interruption example

The goal with UPSTAGE was to strike a balance between defining a set of constraints for developers while maintaining sufficient flexibility for them to model systems not considered in the original formulation. This is something that is still currently being evolved, but after multiple experimental applications, the elements defined and their properties and methods are being more refined and intuitive. Overall, the solution selected by the team was to formulate the minimal number of elements necessary that could provide the required functionality. The minimal set of elements is described in Figure 4.

The framework relies on re-engineering the concept of Event as formulated by SimPy. UPSTAGE Events contain functionality to facilitate the rehearsal of tasks. These Events are scheduled in the Environment in a nearly identical approach to how it is done in SimPy. The UPSTAGE environment is a simple extension of the SimPy Environment, with considerations for functioning as a singleton in order to avoid having the user create disconnected environments. UPSTAGE Tasks trigger UPSTAGE Events, and the completion of the Task is determined by the triggering of an Event. Tasks can launch other tasks, but more importantly, they are the means by which the State of an actor can change. Actors are entities whose condition over time is defined by the set of states they contain. These States can be frozen or changed throughout the simulation. UPSTAGE contains functionality to avoid having multiple Tasks change the State of an Actor unless explicitly stipulated by the modelers. This is critical to avoid unintended behaviors, where two or more Tasks may be inadvertently changing the state of an Actor, producing a nonsensical behavior that does not raise an error. These types of issues tend to be very difficult to debug because modelers oftentimes must evaluate complex series of events to understand which two or more processes are conflicting.

In addition to the functionality described above, Actors can perform and rehearse Tasks, which, as described previously, is a critical capability of a discrete event simulation framework for complex sociotechnical systems. The final element of the framework is the intelligent agents that can assign tasks to actors. These Directors were not fully implemented but prototyped for future development.

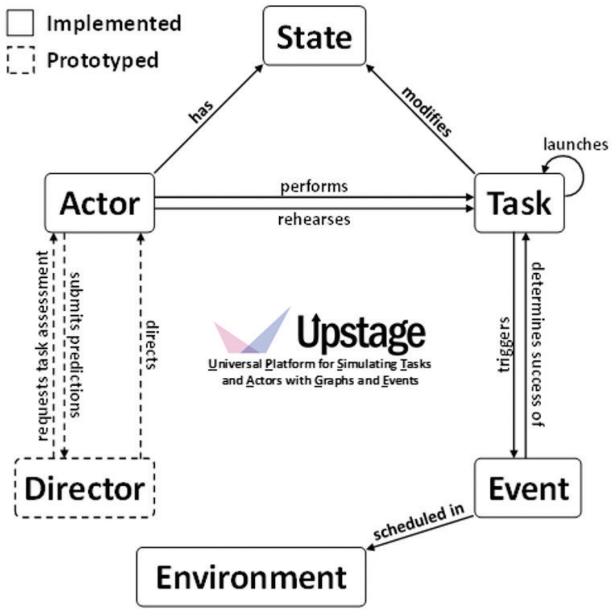

Figure 4 The UPSTAGE elements

The concept of State and how it relates to the Actor is illustrated in Figure 5. In it, a helicopter actor (Helo) has four states: **fuel** (the amount of fuel in the helicopter), fuel_**burn** (the rate at which the helicopter burns fuel), **location** (where the helicopter is located), and **speed**. The fuel is a state that linearly changes over time, and location offers similar functionality but for 2-dimensional space. An instance of the helicopter (helo) is instantiated and named "Helo 1". This instance has 100 units of fuel at the start, a speed of 5 units of distance per unit of time, a fuel burn of 12.1 units of fuel per unit of time, and an initial location at the origin of the 2-dimensional space. Figure 5 also illustrates how time-dependent states can be activated. This is the call that a task changing said states would implement, and by doing so, would ensure that no other task changed the relevant states. The States in UPSTAGE can be extended to address a project's specific needs, e.g., the LocationChangingState can be modified to work on a geodesic. All States keep a record of how their value changes over time, facilitating the analysis of a simulation's results.

The concept of Task rehearsal is illustrated in Figure 6. The example defines a Heal Task that can be performed by an ambulance Actor. Ambulance Actors have limited shift times. Therefore when checking to see if they will be able to heal an incoming patient, it is necessary to ensure they will have sufficient time in their shift to do so. The task is written in such a way that if the task receives a PLANNING_FACTOR_OBJECT instead of an actual patient, the time to perform the task will be an average time to heal. This allows a copy of the actor to be created, and as the Get Event is called, the framework will provide the PLANNING_FACTOR_OBJECT, allowing the task to run and provide an overall estimated shift time, which can be checked against the maximum shift length. This way, if there were multiple ambulances and patients, the planning agent could triage patients based on resources or call for additional ambulances.

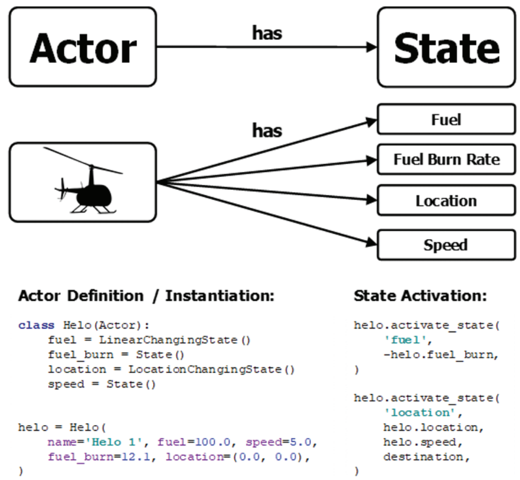

Figure 5 Actors and States

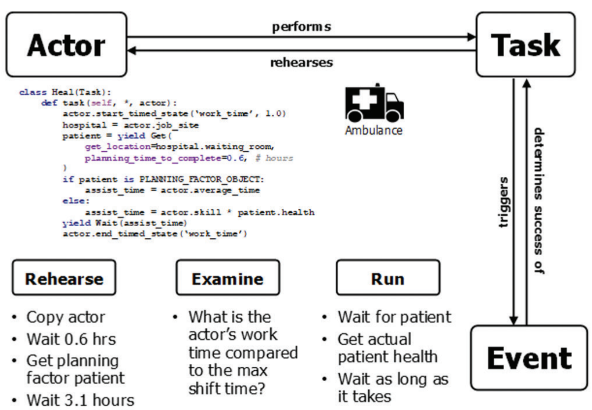

Figure 6 Task rehearsal example

Tasks can be interrupted, as illustrated by the example in Figure 7. In it, a helicopter Actor performs an Observe Task. The modeler describes the primary function of the Task in the task method, and on_interrupt, an additional method that is executed when the Task is interrupted. The regular task does not yield on a regular Event, but on a "marked event," a special method that allows the on_interrupt method to know under what conditions the task was interrupted. In this example, the task can be interrupted by either having to return home or because a survivor was observed.

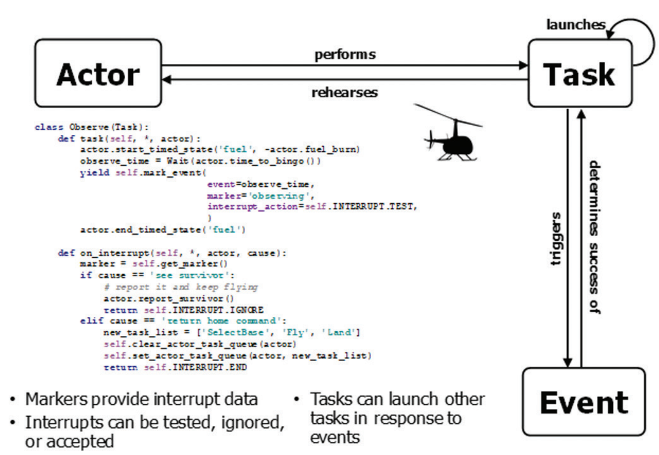

Figure 7 Task interruption example

SPACE DOMAIN TECHNOLOGY INJECTION

The interface between acquisitions and sustainment and technology development within the Department of Defense (DoD) could be made more robust by using mission engineering methods to connect them with a set of models. Recommendations from an advisory panel suggested that the DoD take a holistic, portfolio-centric method for acquisitions rather than current program-centric practices. In the portfolio-centric approach, systems are evaluated within an overall portfolio, allowing for more efficient use of resources when connecting the mission needs of warfighters with acquisitions. These evaluations should be informed by mission and digital engineering, using data-driven methods.

This research focuses on the injection of new technologies into an integrated acquisition portfolio review (IAPR). It specifically evaluates space surveillance domain assets for an anti-surface warfare (ASuW) scenario. Two different systems are used in the IAPR, the first being legacy satellites, which are generally larger, more capable, and more expensive. These are contrasted with the new technology smallsats, which are generally smaller, less capable, and less expensive. These two systems are placed into constellations consisting of a few to many spacecraft, and their ability to surveil the region of interest is evaluated with the tradespace analysis tool for constellations (TAT-C). TAT-C is an open-source mission engineering tool developed for early design phase analyses such as this one. The legacy and smallsat systems are chosen to represent the types of systems which will be selected in the IAPR, and have characteristics that show meaningful tradeoffs.

The new technology injection workflow is integrated with the larger WRT 1049.5 toolset. The legacy and smallsats are evaluated with TAT-C, and these metrics are introduced into the analytic workbench (AWB), which is then used to perform a robust portfolio optimization (RPO) routine. The Pareto-optimal solutions to this routine demonstrate how decision-makers can get the best information for acquisitions and sustainment in a portfolio-centric manner which includes new technology injections.

1.6 BACKGROUND

Mission Engineering and Digital Engineering

The Mission Engineering Guide and Defense Acquisition Guidebook describe Mission Engineering (ME) as "the deliberate planning, analyzing, organizing, and integrating of current and emerging operational and system capabilities to achieve desired warfighting mission effects" (Office of the Under Secretary of Defense for Research and Engineering), (Department of Defense, Defense Acquisition Guidebook). ME is also described as being part of systems engineering abstracted to a higher level (Zimmerman and Dahmann). As the systems problem is abstracted to higher levels, there are corresponding increases in complexity and uncertainty. This is due to missions spanning diverse sets of systems, scenarios, and stakeholders. Furthermore, these stakeholders have goals that do not necessarily align with one another and may directly be in conflict.

Digital Engineering (DE) is "an integrated digital approach using authoritative sources of system data and models as a continuum throughout the development and life of a system." DE can support ME by connecting models that previously existed in a siloed environment, capturing information generated by these models in an authoritative source of truth, and communicating that information to key decision-makers and other stakeholders. The DoD DE strategy seeks a transformation to digital engineering and expects to see "Increased efficiency in engineering and acquisition practices" from this transformation (Department of Defense, Digital Engineering Strategy).

The DoD DE strategy explicitly calls for the use of models to "inform enterprise and program decision making." These models should express system performance metrics in a probabilistic sense due to inherent uncertainties in warfare scenarios (Green and Stracener). Generally, the models used for DoD mission engineering activities will represent mission threads (or kill chains), where the system of systems' mission is given as a series of steps that will nominally achieve the desired effects. Digital threads are seen as a key enabler for reducing risks in acquisition workflows by helping decision-makers better understand the impact of design choices on cost, schedule, and mission outcomes (McDermott et al.). Within the military decision-making process, acquisitions are treated as a subsystem of mission engineering, with value-focused modeling as an effective method for acquiring the best system (Hernandez et al.).

Integrated Acquisition Portfolio Review and New Technologies

The DoD follows a three-step process to guide the acquisition process from design to deployment. Broadly, the three steps consist of: "identifying a need, establishing a budget, and acquiring the system" (SBIR-STTR). These activities are currently performed in a more siloed, program-centric workflow; however, the Advisory Panel on Streamlining and Codifying Acquisition Regulations recommends changing to a more portfolio-centric approach (Defense Technical Information Center). These recommendations do not suggest making a new acquisition system but rather transitioning the current one (Section 809 Panel, *Report of the Advisory Panel on Streamlining and Codifying Acquisition Regulations, Volume 2 of 3*). The General Accounting Office states that the DoD should follow industry best practices in integrated portfolio management, which addresses system investment at the enterprise level, allowing for a more optimal allocation of resources (U.S. Government Accountability Office). This will connect the requirements of warfighters with available resources, reduce redundancy in acquisitions, and aid in selecting the best options to accomplish a mission (Section 809 Panel, *Report of the Advisory Panel on Streamlining and Codifying Acquisition Regulations, Volume 3 of 3*). Furthermore, the National Defense Authorization Act for Fiscal Year 2018 (Thornberry) states that "data analysis, measurement, and other evaluation-related metrics" must be used to improve DoD acquisition outcomes.

Looking into the future, IAPRs are intended to keep US military capabilities on pace or better than adversaries and aware of changing environments (Bertuca). The dynamic environment is where new technology injection into a portfolio review is important. This research investigates methods for assessing new technologies within the IAPR process. This is inherently an ME problem, as consistent mission-level performance metrics are needed to evaluate tradeoffs within the portfolio. Focusing on the space domain, this analysis uses an ME tool to introduce a constellation of small satellites into the ASuW mission portfolio. This follows current trends in the space industry, where newer spacecraft tend to be smaller, cheaper, yet less capable than older, "legacy" spacecraft. The capability gap is overcome by launching more satellites and operating them in concert as a constellation.

1.7 SPACE DOMAIN MISSION SCENARIO

A fictional scenario has been created for a hypothetical conflict occurring in the Gulf of Mexico between forces from Florida against forces from Texas. Using this mission frame, the group worked to develop a simulation using a variety of observation points from satellites to determine the effectiveness of the mission.

A narrative was created for a series of offensive and surface-level attacks from Florida directed toward Texas through the Gulf of Mexico. This forced Texas to focus on various air and surface assets and their options for munitions. In addition, Texas would rely more on satellites that may provide data on fleet movements as well as use naval and anti-surface warfare, including space systems that will allow their leaders to make more informed strategic decisions. The following mission narrative and scenarios were used to perform tests with the Tradespace Analysis Tool for Constellations (TAT-C), focusing primarily on the interactions that may be observed from satellite imagery.

A few red force (Florida) mission scenarios or paths were created to simulate red force forces traveling through the Gulf of Mexico to attack the blue force in Texas. The model was created as a straight shot path (Figure 8) from Florida to Texas, starting at St. Pete Beach at 27.73 N, -82.75 W and ending in Galveston, Texas at 29.13 N, -95.08 W. The mission is approximately 645 nautical miles long, with an assumed vessel speed of 40.31 knots and a total traveling time of 16 hours.

The structured timeline of events for the mission narrative begins with Florida launching an attack & moving surface vessels towards Texas at 5:00 pm on July 1st. Texas satellites pick up a potential threat at 5:00 pm and ask for confirmation that the suspected target is a red force before moving defensive forces. Meanwhile, the Florida vessels continue moving forward. At 7:00 pm, Texas ground satellites confirm that the suspected target is a red force, and after a half hour, Texas moves defensive assets into position, so they will be prepared for Florida's attack when their ships arrive at approximately 5:00 am on July 2nd.

The alternative mission path can be seen in Figure 9 with multiple changes of directions after various distance/time intervals. This mission path would require a total traveling distance of 892 nautical miles, and it was assumed that the ships' collective speed would be constant at 34.75 knots which would make the trip a total of 26.66 hours long.

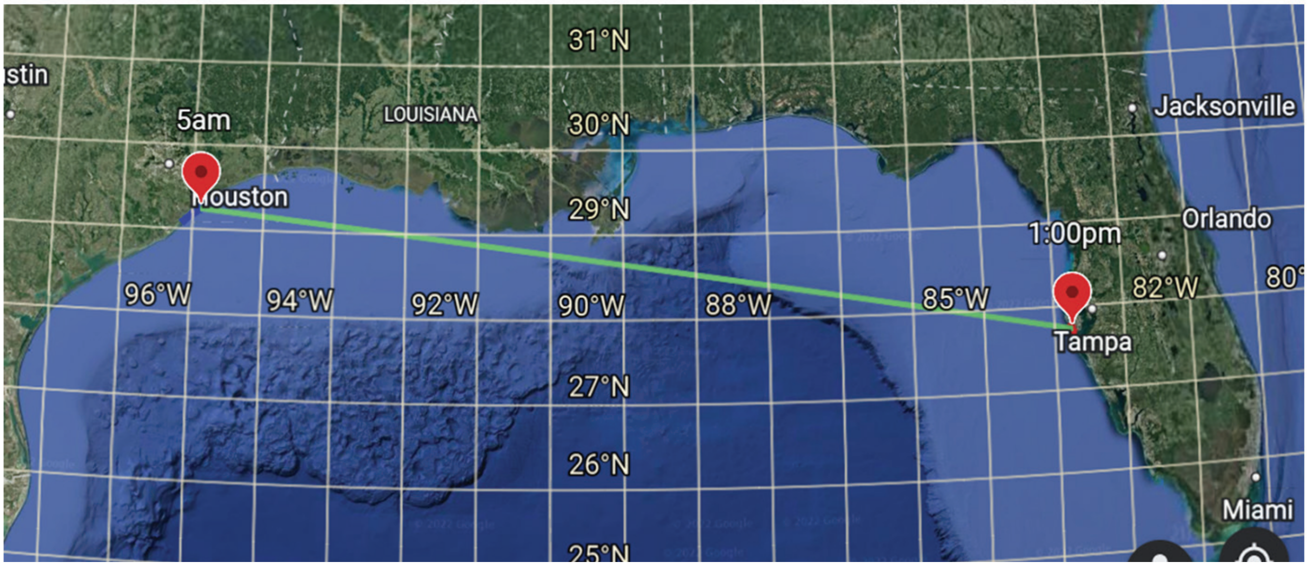

Figure 8 Direct boat path from Florida to Texas

Different metrics have been analyzed to measure the performance of the blue force. The main metric is the speed with which different satellite constellations can identify red force movements. Mission cycle time depends on the time to detect versus identify the threat. This includes the time to obtain data, create information, and decide via the Command, Control, Communications, Computers Intelligence, Surveillance and Reconnaissance (C4ISR) process, which is then disseminated across the force. Another performance metric analyzed was the probability of identifying within one hour. This could be called the access time or the time to identify red assets. On the other hand, the time needed to request and confirm a red force threat before calling for a defensive response would be the revisit time. Lastly, another metric may be how quickly the blue force can provide another observation after an initial confirmation request.

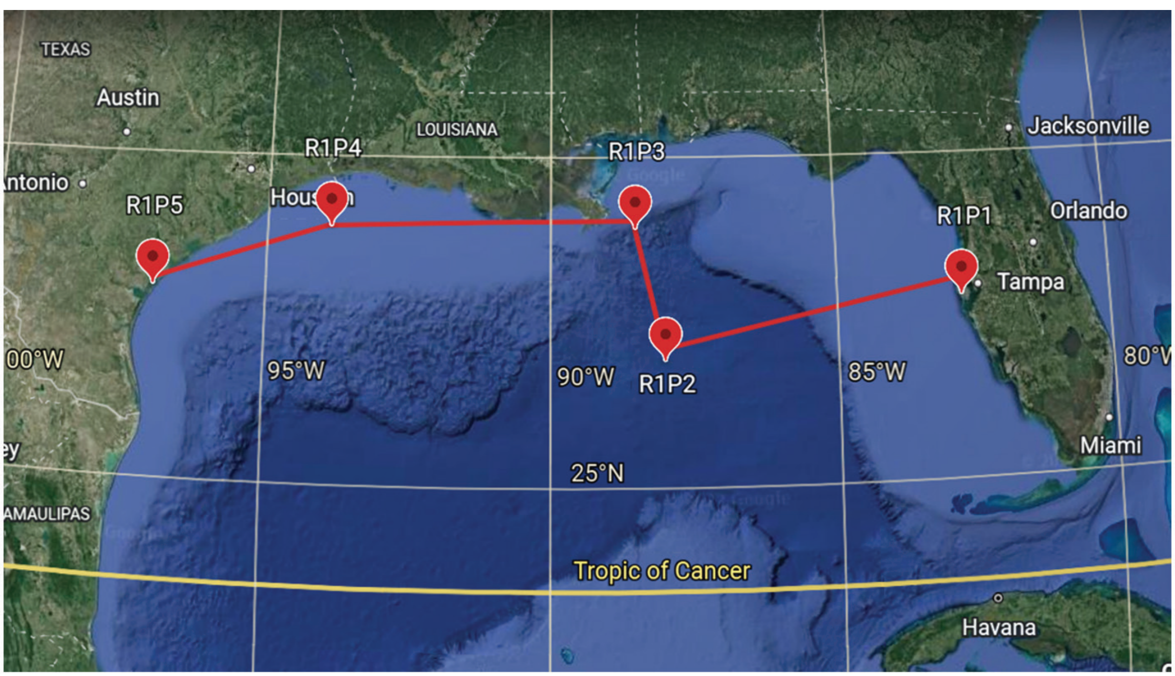

Figure 9 Representative boat path from Florida to Texas

1.8 TRADESPACE ANALYSIS TOOL FOR CONSTELLATIONS (TAT-C)

The Tradespace Analysis Tool for Constellations (TAT-C) is an open-source Earth science space mission analysis software tool developed in part by Stevens Institute of Technology with support from NASA's Earth Science Technology Office (ESTO). TAT-C computes key performance metrics for new Earth science mission concepts focused on satellite constellations and distributed observing systems. It serves as a bridge between scientists and engineering communities during early stage (pre-Phase A) conceptual design activities by explaining how alternative system architectures can achieve scientific objectives with respect to spatial and temporal coverage of geophysical phenomena. TAT-C version 3.1.2 was released in July 2022, and version 3.1.3 was released in August 2022 and is available via the conda-forge distribution system.

TAT-C consists of three components: 1) a low-level Python analysis library that defines object schemas and analysis functions, 2) a backend application that processes HTTP requests across a network of parallel task worker machines, and 3) a web-based frontend that visualizes analysis results using a web-based 3D geospatial visualization tool. Figure 10 (left) illustrates a snapshot of a five-satellite constellation with wide (red) and narrow (orange) field of view instruments. Figure 10 (right) depicts coverage statistics for a global observation mission integrated over a 30-day period.

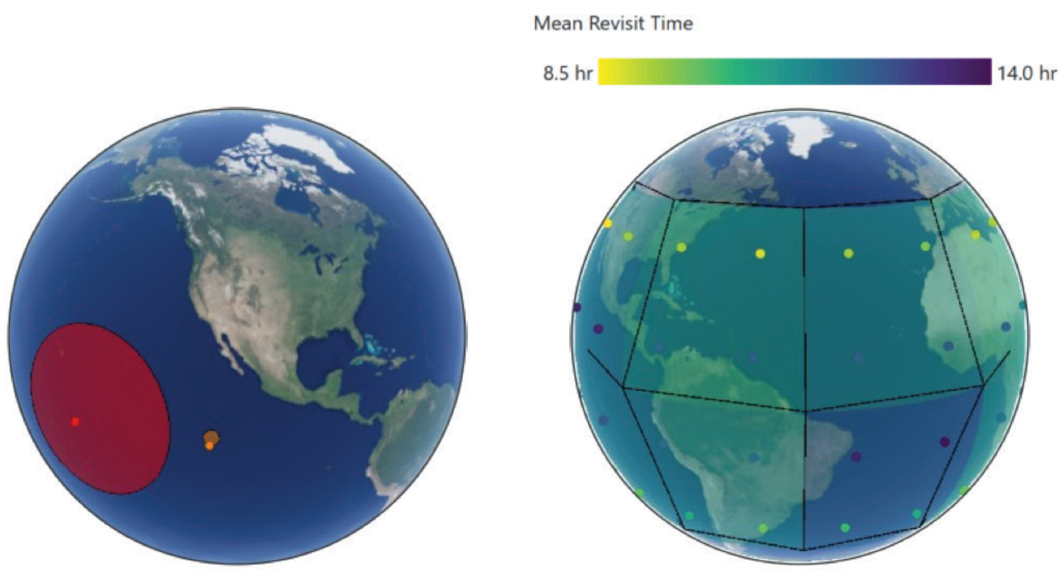

Figure 10 TAT-C simulation on satellite constellation observations (left) and computation on coverage statistics for regions of interest (right)

This project focuses only on the analysis library (NASA Earth Science Technology Office), which depends on several other Python libraries including: skyfield and sgp4 for astronautical calculations and spacecraft orbit propagation, geopandas, shapely, and pyproj for geospatial data structures, including coordinate system transformation and planar geometry manipulation, and geoplot and contextily for geospatial visualization.

TAT-C object schemas model an observing mission consisting of satellites and ground stations as well as points-of-interest on the Earth's surface. Key object schemas include:

- **Point:** a point-of-interest on the Earth's surface in the WGS 84 coordinate system.
- **Ground Station:** location of a ground station for communication with satellites.
- **Orbit:** orbital trajectory used by a satellite (subclasses: Two Line Elements, Circular Orbit, Sun-synchronous Orbit, and Keplerian Orbit).
- **Instrument:** sensor used by a satellite to collect data about the Earth's surface
- **Satellite:** combination of an orbit and instrument(s).
- **Constellation:** configuration of multiple satellites in coordinated orbits (subclasses: Train Constellation, Walker Constellation).

The TAT-C object schemas serve as inputs to analysis functions. Key analysis functions include:

- **Collect Observations:** enumerate the observation opportunities for an instrument hosted on a satellite to view a point-of-interest over a mission period.
- **Aggregate Observations:** merge observations windows across a constellation of satellites.
- **Reduce Observations:** compute descriptive statistics for collected observations.
- **Collect Orbit Track:** list the trajectory of a satellite orbit over a mission period.
- **Collect Ground Track:** list the trajectory of the projected coverage area over a mission period.
- **Collect Downlinks:** enumerate the opportunities for communication between a satellite and ground station network over a mission period.
- **Compute Latencies:** estimate the duration between observation and the first available downlink for given points of interest, satellites, and ground network.
- **Reduce Latencies:** compute descriptive statistics for computed latencies.

This project applies TAT-C analysis functionality to the mission scenario using a Jupyter notebook. Analysis functions are used in combination with custom scripts to enumerate and evaluate alternative satellite constellation designs.

1.9 MISSION ENGINEERING MODEL IMPLEMENTATION

Introducing new technology into the AWB first requires a consistent, system-level evaluation of the new (smallsat) and existing (legacy) assets. For the ASuW scenario, three different spacecraft constellations are evaluated:

- Low-cost smallsat constellation with global coverage (modeled after Capella-1 orbit),
- High-cost legacy satellite constellations with global coverage (modeled after Sentinel-1 orbit),
- Low-cost smallsat constellation with dedicated coverage (modeled after Capella-1 orbit with the modified inclination – 29 degrees, the northern edge of the Gulf of Mexico – to ignore higher latitudes not relevant to the mission scenario)

Only the first two of these constellations are introduced into the AWB, the smallsats representing the new technology injection and the legacy satellites representing existing assets. The "bespoke" constellation, which only covers the region of interest, is included due to the Space Force's stated interest in a "tactically responsive launch" capability, where a contractor would deploy small satellites to specific orbits on very short notice (Erwin).

Modeling the mission scenario requires first setting locations for the ships along their path towards Texas and then evaluating how well each space asset can surveil that location. A representative ship path was first chosen, and the Jupyter notebook then breaks this path into a set of points for analyzing observation opportunities. Sequential lat/long points were the inputs, and those were used to create a linestring. This linestring was then projected onto a local cartesian map in order to approximate distances more accurately. An input for the speed of the ship was used to calculate how long it would take the ship to traverse this path. Points were generated based on where the ship would be every ~30 minutes. Time values were assigned to these points to indicate when the ship would be at each one.

Next, the differing satellite constellations were simulated with TAT-C to find the coverage statistics for the varying ship locations. The small satellite constellations were equipped with instruments with a 45-degree field of regard (FOR) and the legacy constellations with a 90-degree FOR. The small satellite options were tested with constellations of 1-48 satellites, and the legacy option was tested with 1-24 satellites. Each specific combination of satellite type and constellation size was simulated 48 different times with varying start times spanning an entire day at half-hour intervals. Table 1 shows the inputs and output for the mission scenario. This was done because the start time of the simulation can greatly impact the results, especially for smaller constellations. The results shared below are based on averages of these simulations that differ only by start time.

Table 1 Mission engineering model inputs and outputs with nominal values

Input/Output (I/O)	Nominal Values/Ranges
I: Number of Satellites	1-48
I: Type of Constellation	Legacy, global smallsat, dedicated smallsat
I: Boat Path	Ordered latitude/longitude coordinates starting on the Florida gulf coast and ending on the Texas coast
O: Observation Opportunities	Coordinates, start and end times

Results show a strong linear relationship between the number of satellites and the number of observation opportunities in all cases. The legacy satellite constellations (modeled after Sentinel-1) show a performance advantage over the global smallsat constellations (modeled after Capella-1) due to more capable instruments (increased field of view) and a slightly higher orbit that further increases viewing area.

The dedicated smallsat constellations are also modeled after Capella-1; however, they are placed in a less-inclined orbit - 29° - to ignore latitudes not relevant to the mission scenario. It was superior to the legacy constellation despite having the same less capable instruments modeled on the global smallsat constellations – see Figure 11. The slope of each line, enumerated in Table 2, represents how many additional observations are gained for each additional satellite. The AWB inputs are derived from the relative relationship here. Legacy is chosen as a baseline case and assigned the value of 100 cueing opportunities per satellite. Other constellations would be represented in the AWB in comparison to that baseline number. For the figures in Table 2, this would mean values of 31 for the smallsat global constellation and 177 for the dedicated constellation.

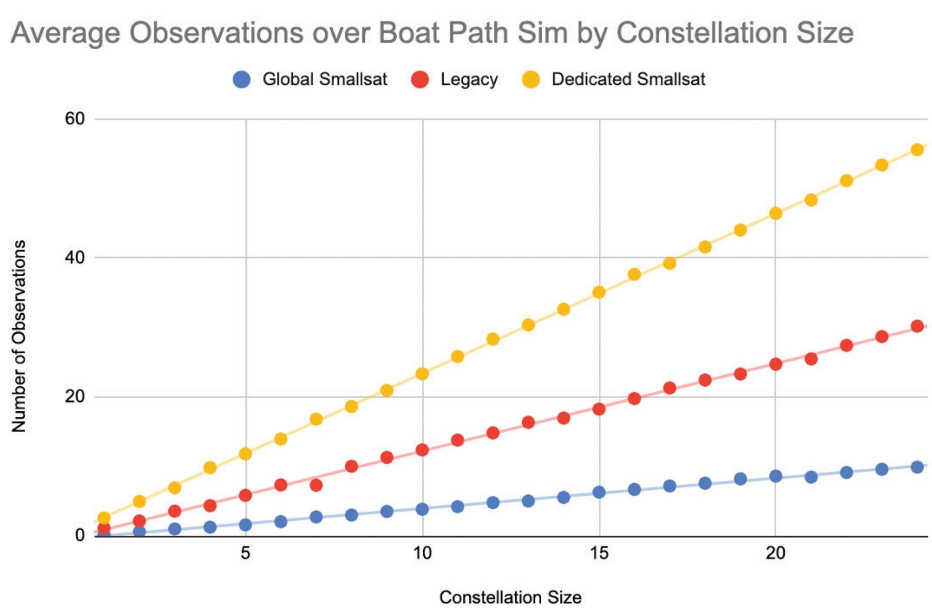

Figure 11 Coverage statistics for all three constellation types.

Table 2 Aggregated results for the value of each additional satellite

Constellation	Observations for each additional satellite
Global Smallsat	0.4
Legacy	1.3
Dedicated Smallsat	2.3

In testing the impact of different paths, we found that the results for global constellations were unaffected. The dedicated smallsat constellation, though, produced varied results. The northerly path used for the results above was much stronger than a southern path which produced results more in line with the legacy global constellation.

The final iteration of the Jupyter Notebook delivered with this report automates the entire process described in this section and has the following upgrades:

- Only runs simulations for the first two and last two numbers of satellites for each constellation (1, 2, 47, 48) to greatly reduce computation time. Validation cases showed that the results were identical after removing the intermediate values.
- Creates a random start time within a one-month window for each of 100 iterations for each combination of boat path, constellation type, and size of the constellation. This eliminates the possibility of temporal resonance effects.
- Averages result from different boat paths. Outputs were changed from observations-per-simulation to observations-per-unit-time in order to make accurate comparisons between simulations of different lengths.

Assumptions and limitations

A simplifying assumption was made to consider the position of the ship as stationary for ~30-minute intervals while the satellites collected observations. We considered this to be an acceptable simplification, given that the discrepancy in speed between the ships and the satellites is so large. A more realistic approach would be to allow the observation target to move in real-time and add a triangular tail representing the detectable wake. Further work would be necessary to determine the value of this added level of detail.

The ships were assumed to progress along their paths at a constant rate of 40 miles per hour. In reality, they would almost certainly not have a constant velocity due to weather and blue force interactions.

A consistent seed start time of July 1, 2021, was used for the simulations. Because TLEs are used to determine orbits, and they are specific to a certain point in time, the results become less accurate the further away you are from the time associated with the orbital elements. Validation testing was done to show that moving backward in time produced negligible differences due to the exponential process of orbit decay. Future work could improve the accuracy of this tool by assigning the center point of the random start times to the date taken from the TLE.

Analysis was based solely on observation opportunities. Further work could introduce additional variables or thresholds, such as the minimum time required for a viable detection opportunity, the probability of a successful detection given a viable opportunity, and a tolerance level for false positives that would influence the probability of successful detection.

The assumption of having perfectly-spaced constellations also limits the reality of the space surveillance assets within the AWB. Unlike other systems in the portfolio, which can be redeployed at a reasonable cost, space assets have a limited ability to change their orbits. The detection capabilities of space assets are highly dependent on their orbits, and changing orbits once deployed can be extremely expensive (in that they use a lot of fuel and significantly shorten the lifespan of the asset) or even impossible. It was assumed here that, for any number of satellites, they were evenly distributed in a preplanned constellation architecture. However, this is not realistic for a scenario wherein a particular number of satellites are launched to begin with, and more satellites are added to the portfolio later. Put another way, it sometimes makes more sense to think of space assets as fixed infrastructure than mobile re-deployable assets. This may change as we see advances in on-orbit servicing that can refuel or reposition existing satellites or as we see an increasing shift towards more distributed systems with shorter planned life cycles.

1.10 MISSION ENGINEERING MODEL INTEGRATION

After performing mission-level analyses on the various space domain surveillance constellations with TAT-C, they are introduced into the broader ASuW scenario. Originally, there was a single SAR Satellite entry in the AWB. This was later replaced by the a) legacy SAR Satellite representing current systems and b) a small SAR Satellite representing very near future space surveillance capabilities.

The TAT-C simulations for the space assets are used to create inputs for the AWB. These inputs must be consistent with the rest of the systems within the ASuW portfolio so that the RPO solutions show real tradeoffs. Reasonable assumptions are used for some of the AWB inputs. In particular, required SATCOM bandwidth and costs for each system based on assumptions to show correct relationships between systems and force meaningful tradeoffs.

1.11 SUMMARY

This research introduces a method for injecting new technologies into an IAPR. Using a notional ASuW scenario, a space domain mission engineering tool (TAT-C) is used to evaluate how a new constellation of smallsats compares to a constellation of larger, more expensive, and established satellites. The surveillance capabilities of these systems have differing orbits and observation instruments, showing the types of tradeoffs seen in a real-world scenario. The system-of-system level performance metric of total observations over the mission period is used to introduce a mission-level performance metric into the AWB. Results show that moving to a smaller, more distributed architecture provides roughly one-third of the capability of the existing architecture. This may be a preferred strategy, depending on the differences in resource requirements.

ANTI-SURFACE WARFARE (ASUW) PROBLEM

1.12 PROBLEM FORMULATION

The Anti-Surface Warfare (ASuW) problem selected is intended to be notional while remaining illustrative of an ASuW problem that may be relevant to the US Navy. The basic model has two surface threats traversing a body of water, (1) a Surface Action Group (SAG) and a Fast Attack Craft (FAC) group. The SAG is composed of surface combatants (e.g., frigates, destroyers, and cruisers). The FAC group is composed of small and fast (40+ kts) vessels. The blue force must complete the Find, Fix, Track, Target, Engage, Assess (F2T2EA) kill chain against these threats. Therefore the blue force must include sensors, shooters, and a command and control element that coordinates and decides how to task the other elements. The sensors can be space, airborne, and surface.

For this problem, the blue architecture does not include subsurface elements (e.g., submarines and underwater arrays). The ASuW problem selected is intended to be notional while remaining illustrative of an ASuW problem that may be relevant to the US Navy. This is a realistic warfighting problem and one that is addressed by a complex portfolio of assets from across multiple platforms and weapons. The basic model has two surface threats traversing a body of water: a Surface Action Group (SAG) and a Fast Attack Craft (FAC) group. The SAG is composed of surface combatants (e.g., frigates, destroyers, and cruisers). The FAC group is composed of small and fast (40+ kts) vessels. The blue force must complete the Find, Fix, Track, Target, Engage, Assess (F2T2EA) kill chain against these threats. Therefore, the blue force must include sensors, shooters, and a command and control (C2) element that coordinates and decides how to task the other elements. The sensors can be space, airborne, and surface. For this example of the problem space, the blue force architecture does not include subsurface elements (e.g., submarines, underwater arrays). Subsurface elements could be added to this analysis at a later date without a change in the methodology. A simple basic architecture is depicted in Figure 12.

A more comprehensive construct based on a richer kill web is depicted in Figure 13. This construct increases multidomain effects and interdependencies as it includes additional sensors (i.e., Maritime Patrol Aircraft (MPA), Helicopter, Radar from Surface Combatants) and shooters (i.e., MPA, Attack Aircraft, Helo). Sensors are categorized into two sets: Electro-Optical/Infra-Red (EO/IR) sensors and Radio Frequency (RF) sensors. In this scenario, EO/IR sensors are primarily used for target identification, while RF sensors are used for target detection and potentially for cueing other sensors. All data and concepts illustrated in this notional example are derived from open-source data, primarily wikidata.org.

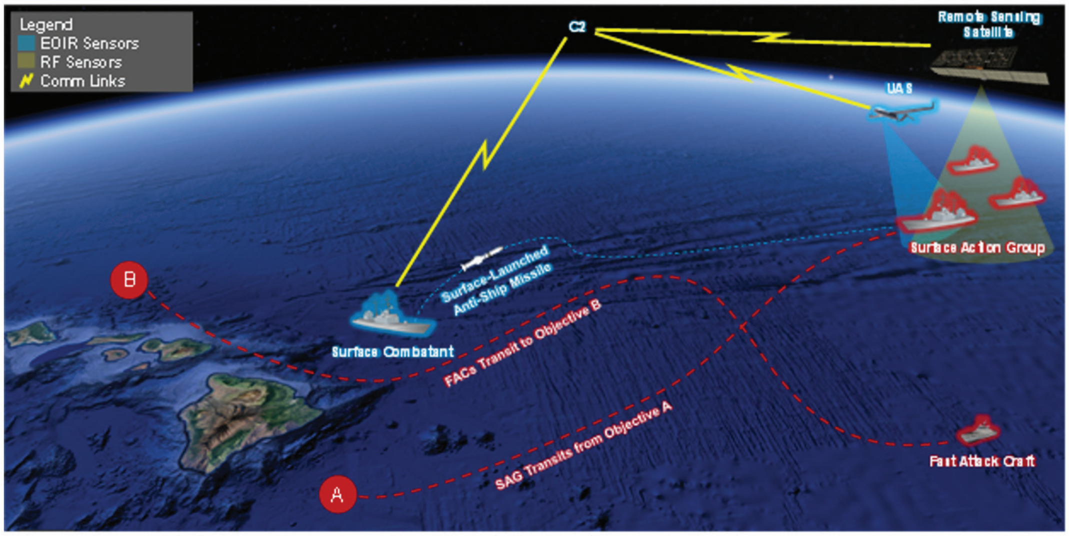

Figure 12 OV-1 of the simple notional Anti-Surface Warfare scenario

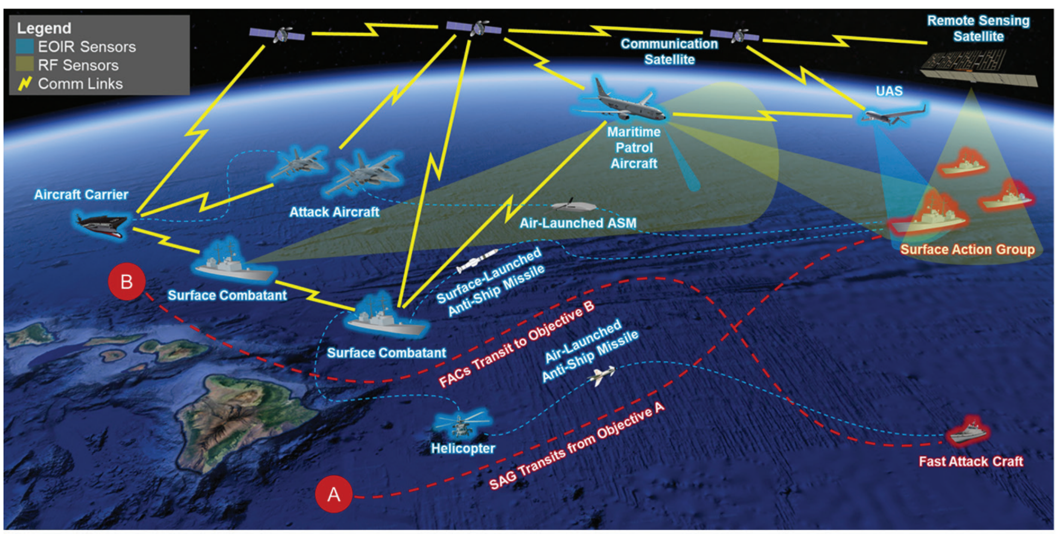

Figure 13 OV-1 of a more comprehensive Anti-Surface Warfare scenario

All the data for the assets and weapons described below was obtained or derived from wikidata.org. The roles/responsibilities and the specific values are not intended to be overly accurate or complete but capture coarse-level capabilities that illustrate the potential tradeoffs that the Analytical Workbench can assess.

The blue force ASuW kill chain is based on the Find, Fix, Track, Target, Engage, Assess (F2T2EA) kill chain, with some simplifications to ensure the example remains unclassified.

- **Find:** This is the task of doing the initial detection of the red surface vessels. The result is a cue to other sensors to do the additional assessment on the potential target.
- **Fix:** In this formulation, the fix is primarily concerned with identifying the potential target. It requires distinguishing a target from its surroundings and is doctrinally described as "identifying an emerging target as worthy of engagement and determining its position and other data with sufficient fidelity to permit engagement." This requires a cue from another sensor.
- **Track:** In this formulation, the process of formulating tracks for targets is highly abstracted. In reality, this task can require complex processes to fuse different sources of data and assess the error, maintaining custody of a target across one or multiple assets until a

target solution is determined. Adding to the complexity, most systems today that can fix can also track. In reality, this task can require complex processes to fuse different sources of data and assess the error.

- **Target:** This step involves defining/selecting a capability to take action against an identified target, inclusive of the weapons, platforms with those weapons, other resources, and authorities. In the formulation defined for this effort, the targeting phase is highly simplified and presumed to be done to a high degree of certainty. In this formulation, the targeting phase is highly simplified and is presumed to be done with a high degree of certainty.
- **Engage:** For the purposes of this model, the engage phase primarily consists in launching a weapon against the target and evaluating a random chance of the weapon finding the target and killing it. This makes the problem tractable and able to produce outcomes measurable by the integrated set of methods. In reality, however, increased stand-off ranges in contested environments inject time into the F2T2EA, something not accounted for in traditional kill chain analyses in general.
- **Assess:** the assessment phase of the F2T2EA kill chain is critical. However, for the purposes of this simulation, the process is highly simplified. Any surviving targets remain alive in the simulation and can be picked up by other sensors.

The assets required to complete the kill chain are listed in Table 3 below. The potential assets considered in the architecture are grouped by their domain. Weapons and personnel are the two other categories of elements considered in the architecture mix analysis. For ease of understanding, real names of assets are used, but all the properties and tactics, techniques, and procedures (TTPs) used for the assets are notional and unclassified.

Table 3 Potential assets for the ASuW scenario

Domain	Asset	Description
Space	Legacy Synthetic Aperture Radar (SAR) Satellite	Larger Space-based Remote Sensor that uses Synthetic Aperture Radar to detect surface vessels from their wake. Primary function: cueing.
	Small SAR Satellite	Smaller, more affordable, and less capable Space-based Remote Sensor that uses Synthetic Aperture Radar to detect surface vessels from their wake. Primary function: cueing.
	Electro-Optical/Infra-red (EO/IR) Imaging Satellite	EO/IR space-based remote sensing capability that may be able to identify surface vessels. Primary function: target identification. It may provide cueing but not the primary function.
	Communications Satellite	Space-based communications relay provides over-the-horizon communications.
Air	MQ-4C	Unmanned reconnaissance aircraft. Primary function: target identification, secondary function: target detection.
	P-8A	A Maritime Patrol Aircraft (MPA) can detect and identify surface targets.
	EA-18G	A Standoff Electronic Attack Aircraft that can passively detect surface targets.
	F/A-18E/F	An attack/fighter fixed-wing aircraft that can launch anti-ship weapons. Requires a CVN to launch from.
	MH-60S	A rotary wing aircraft that can detect, identify targets at close range, and launch short-range anti-ship weapons with limited lethality.
	F-35B	Short Take-off and Vertical Landing (STOVL) stealth aircraft can be operated from amphibious assault ships (e.g., LHA, LHD).
	F-35C	Carrier-capable fixed-wing stealth aircraft can only be launched from CVNs.
Surface	FREEDOM (LCS-1)	Mono-hull Littoral Combat Ship (LCS), a small, more affordable, but less capable surface combatant.
	INDEPENDENCE (LCS-2)	Multi-hull Littoral Combat Ship (LCS), a small, more affordable, but less capable surface combatant.
	ARLEIGH BURKE (DDG-51)	First generation (Flight I) of a modern missile-guided destroyer, no capability to support helo operations.
	MAHAN (DDG-72)	Second generation (Flight II) of a modern missile-guided destroyer, limited capability to support helo operations.
	OSCAR AUSTIN (DDG-79)	Third generation (Flight IIA) of a modern missile-guided destroyer, full capability to support helo operations.
	JACK LUCAS (DDG-125)	Future generation (Flight III) of a modern missile-guided destroyer, full capability to support helo operations and improved sensing/weapon systems.
	ZUMWALT (DDG-1000)	Best-in-class guided missile destroyer with improved sensors and weapon systems, signature management capabilities, but limited quantities of anti-ship weapons.
	TICONDEROGA (CG-47)	Legacy cruiser with moderate sensing capability but large missile capacity.
	BUNKER HILL (CG-52)	Modern cruiser with modern sensing capabilities and large missile capacity.
	WASP (LHD-1)	A small aircraft carrier that can support STOVL aircraft operations.
	AMERICA (LHA-6)	A small aircraft carrier that can support STOVL aircraft operations.
	FORD (CVN-78)	A large aircraft carrier that can support carrier-based aircraft operations.

Resource: (Wikidata.Org)

As with the assets, the weapons are notional, with numbers obtained from unclassified sources. However, real weapon names are used to facilitate the understanding of the scenario and the results produced by the framework. The goal of the Anti-Ship Missile (ASM) weapon mix (Table 4) was to illustrate that a notional capability/cost tradeoff could be captured by the Analytical Workbench (AWB) and the discrete event simulation.

The conduct of the operations and employment of the different assets in the ASuW scenario model are dependent on a wide range of factors. This includes the physical deployment of the different assets, the operation rules of engagement, and the actions of the red force actors (there may be more than one working at some level of coordination). For this analysis, we are assuming that the SAG and the FAC group are one red-force actor and are working in full coordination.

Operationally, for the blue force to be successful in the kill chain, their actions must result in the red force losing mission capability and/or deterring the red force from future engagement. Set-based methods containing tactical and intelligence input and assessment results will be required to evaluate the amount of reduction of red force capability needed for blue force success. Within the ASuW mission area, metrics of success are primarily the reduction of red force capability, weapons expended, and blue force casualties.

Table 4 Anti-ship weapons

Designation	Name	Launcher Domain	Range (nmi)	Speed (kts)	Cost (k$)
AGM-114L	Hellfire	Air	6	864	150
AGM-119	Penguin	Air	100	633	800
AGM-158C	LRASM	Air	300	633	3,960
AGM-158D	JASSM-XR	Air	970	1026	1,500
AGM-84D	Harpoon	Air	50	461	500
AGM-84F	Harpoon	Air	170	461	600
AGM-84H/K	SLAM-ER	Air	150	461	3,300
BGM-109 Blk V	Maritime Strike Tomahawk	Surface	1350	493	1,409
RGM-184A	Naval Strike Missile	Surface	100	600	2,194
RGM-84F	Harpoon	Surface	150	461	600
RIM-174	Standard Extended Range Active Missile	Surface	130	2315	4,318

Resource: (Wikidata.Org)

Measuring SoS Capability

A relatively simple set of system capabilities was developed to measure how the various systems contribute to the overall ASuW scenario (Table 5). These are utilized by RPO and can be passed to supporting analysis (like the DES UPSTAGE model) for more detailed analysis. The better a system performs for each of these capabilities, the more likely it is to be allocated when it is SoS capability. These capabilities are intended to be notional and illustrative of the types of characteristics that may be used to assess how well an ASuW System-of-Systems performs. RPO performs optimization of system allocation against SoS performance. To facilitate this, five System of System Capabilities were defined for the overall scenario (Table 6). These SoS Capabilities are groupings of the individual system capabilities.

Table 5 System capabilities

System Capability	Name	Measurement	Measurement Units
SC 1	Maritime Surveillance	Notional Area Surveillance Capability	1/3/9: Low/Med/High
SC 2	Identify Surface Contacts	Notional ID Capability	1/3/9: Low/Med/High
SC 3	Jam Ship Radars	Notional Jamming Capability	1/3/9: Low/Med/High
SC 4	Stand-off Range	Weapon Range (nm)	nmi
SC 5	Disable Surface Combatant	Phit SC	%
SC 6	Damage Surface Combatant	Pkill|hit SC	%
SC 7	Disable Fast Attack Craft	Phit FAC	%
SC 8	Damage Fast Attack Craft	Pkill|hit FAC	%
SC 9	Quickness	Airspeed	kts
SC 10	Coverage	Flight Range	nmi
SC 11	Power Projection	Capacity	#

Table 6 SoS capabilities defined for the overall scenario

SoS-Capability	SC 1	SC 2	SC 3	SC 4	SC 5	SC 6	SC 7	SC 8	SC 9	SC 10	SC 11
	Maritime Surveillance	Identify Surface Contacts	Jam Ship Radars	Stand-off Range	Disable Surface Combatant	Damage Surface Combatant	Disable FAC	Damage FAC	Quickness	Coverage	Power Projection
ASuW Offensive	X	X	X	X	X	X	X	X	X		
ASuW Defensive	X		X	X	X	X	X	X	X	X	X
ASuW Near Peer	X	X	X	X	X	X			X	X	X
ASuW Non-State Actor	X	X					X	X	X	X	
Maritime Awareness	X	X									

ASuW-specific mission analysis of optimized portfolios with UPSTAGE

For this effort, the GTRI team utilized UPSTAGE to develop a more complex network for an ASuW mission, one incorporating both blue and red platforms and capabilities, to produce an analysis that would inform the AWB framework with improved mission fidelity. This will offer a much more complex representation than the tools without these dynamics being considered. Even so, the models for this effort are intended to demonstrate the general capability but will still fall short of real-world dynamic complexity. This unclassified implementation of UPSTAGE to the ASuW problem simulates a notional scenario where Red (Florida) is set to carry out a strike mission against Blue (Texas) launch assets deployed along the coast (Figure 14). A Red Surface Action Group (SAG) and Fast Attack Craft (FAC) group move from their ports through northern and southern routes, respectively, to reach Blue's home shore.

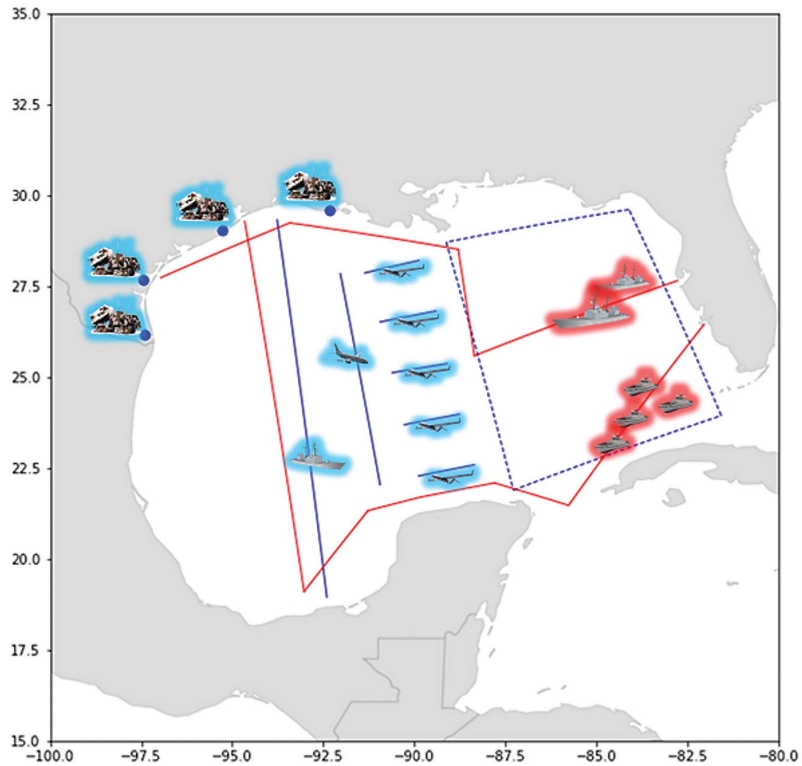

Figure 14 Red (Florida) vs. Blue (Texas)

Blue forces are arranged to provide a layered defense of their shore. Unmanned Aircraft Systems (UAS) fly patrol routes in the eastern portion of the Blue sea, while Maritime Patrol Aircraft (MPA) fly a patrol route to the west of the UAS. Further west, Blue naval assets such as DDGs and CVNs conduct patrols. The exact makeup of the Blue patrols and of the patrolling system's attributes are input into the UPSTAGE simulation. As Red forces move through Blue waters, Blue satellites may detect them in the dotted region in Figure 14. The satellites' detection capabilities are also inputs to the simulation, and it is possible that false targets can be found and reported to the Blue command.

Difficulty in analyzing parameterized force structures is parameterizing command and control (C2). UPSTAGE mitigates this difficulty through entity grouping and rehearsal features to support C2's selection of friendly assets based on user-defined capabilities.

The F2T2EA kill chain is abstracted in the ASuW simulation to follow this general flow (Figure 15). The Blue C2 will receive information from the systems given to it – based on a portfolio – and that information can have a variable certainty as a function of the system that performed the detection. Low certainty information will cause Blue C2 to follow up with UAS or MPA tasking to provide higher-quality track information. If the track quality is high enough, Blue C2 will initiate a fires mission from one of the available fires systems.

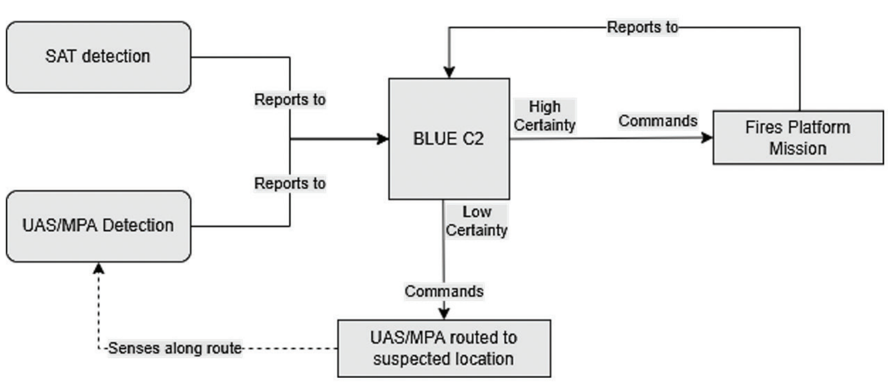

Figure 15 F2T2EA kill-chain

A fires mission will generally involve a flyout to a known or best-predicted position of a Red system using a platform with enough of a weapon class to ensure success. If only surface assets are available, they will be selected, but they are not preferred. Individual fires-capable systems are allowed to detect and fire on Red systems of their own volition. These include any land-based batteries or surface ships. The simulation will run until Red systems are destroyed or they reach Blue shores. The time it takes to complete the scenario and the success/failure are the primary outputs. Secondary outputs can include resource usages, such as fuel and munitions, comms requirements, and other interactions.

1.13 INITIAL ASUW RESULTS

While the process of setting up the full ASuW problem formation is well in progress, initial results show some interesting trades. Running the problem through RPO and examining the results show a continuous improvement of the SoS performance score as the cost constraint is raised, as expected. On closer inspection of the allocations, more interesting results are seen. The parameters used for the initial results are listed in Table 7.

Table 7 Initial results run parameters

	Minimum	**Maximum**	**Steps**
Cost ($MUSD)	50.0	800.0	15
Risk (n.d)	0.2	1.2	3

The results of the RPO run can be viewed in Figure 16, where SoS Performance Index is a non-dimensional measure indicating performance across all selected SoS Capabilities. For this initial example, all 5 of the SoS Capability measures were analyzed simultaneously. In the future, more nuanced results could be achieved by optimizing the SoS Capabilities individually.

The increase in overall SoS capability, as more money is spent, is a fairly obvious and expected result. The most noticeable trend in this chart is the divergence of performance at higher costs when more risk (lower conservatism) is allowed in the solution. More interesting results can be observed in the full allocation table, however. Table 8 shows how many of each system was purchased for the points plotted in Figure 16, a run of RPO on the ASuW scenario.

From Table 8, it is apparent that the preferred low-cost solution (allocations 0 – 2) involves an investment in LCS ships carrying the Hellfire Longbow (AGM-114L), a currently experimental solution, with limited allocations of aircraft, DDGs, or dedicated anti-surface missiles like the AGM-84. However, as the cost constraint is relaxed and the optimizer can afford more expensive systems, it quickly shifts to a solution based largely on amphibious assault ships, F-35Bs, and JASSM-XRs. In this middle range, RPO also demonstrates that Fix, Tracking, and Targeting can be largely based on unmanned assets.

As more money is allowed to be invested, the strategy again shifts to provide more SoS performance. This time once a full carrier is affordable, the investment strategy quickly switches to carriers, F/A-18E/Fs and AGM-84Ds. F-35Cs and JASSM-XRs are preferred if they can be afforded and mixed in with the F-18s as more money is allocated and if more risk is allowed. At that point, if more money is allocated, the same strategy is repeated, mixing in some Arleigh Burke destroyers until another carrier can be afforded.

This initial trend will be further analyzed as the team is able to integrate more tools (SODA, SDDA) with the ASuW scenario. As space domain-specific technology injection is integrated, the effects of satellite technologies on the allocations and analysis will be explored. Higher fidelity will also be executed via UPSTAGE to future predict how these different investment strategies might play out in a simulation. The team expects to include these results in the final report.

Even though the data used for this analysis is notional, some other interesting trends in the mix include the use of the Tomahawk Maritime Strike (MST) missile (i.e., BGM-109 Blk V). More conservative portfolios tend to use fewer MSTs as they have higher uncertainty in their ability to hit targets than the other missile options. AGM-84Ds tend to be preferred because of their cost-effectiveness and because the risk to the launching asset is not captured by the analysis.

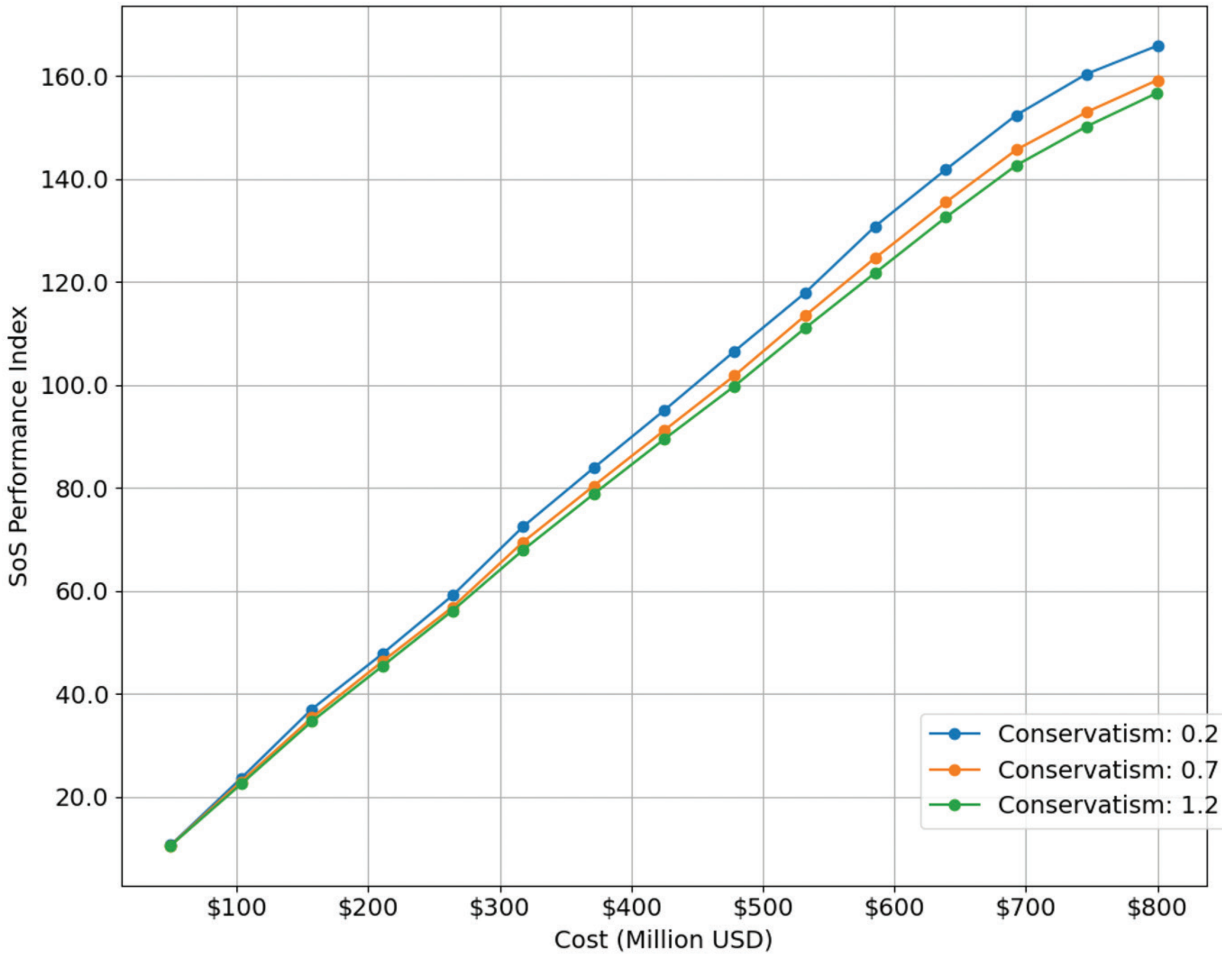

Figure 16 Initial results: cost vs. SoS performance

Table 8 Initial results allocations

	Allocations															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Objective Value	10.7	10.7	10.7	22.6	22.2	21.8	35.4	34.3	33.8	46.9	45.7	45.2	58.1	57.3	56.9	70.0
Cost	$ 50.00	$ 49.94	$ 49.96	$ 103.53	$ 103.53	$ 103.57	$ 157.14	$ 157.14	$ 157.14	$ 210.71	$ 210.71	$ 210.71	$ 264.28	$ 264.19	$ 264.28	$ 317.85
Max Conservatism	0.2	0.7	1.2	0.2	0.7	1.2	0.2	0.7	1.2	0.2	0.7	1.2	0.2	0.7	1.2	0.2
Legacy SAR Satellite	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Small SAR Satellite	1	1	1	2	2	2	3	3	3	4	3	3	3	5	5	6
EO/IR Satellite	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Comm Satellite	2	2	2	4	4	4	6	6	6	7	7	7	8	8	8	9
MQ-4C	12	12	12	15	15	20	20	20	20	20	20	20	20	20	19	20
P-8A	4	4	4	3	4	4	4	4	4	4	4	4	4	4	4	4
EA-18G	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
F/A-18E/F	0	0	0	0	0	0	0	0	0	0	0	0	0	71	68	56
MH-60S	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0
F-35B	0	0	0	8	8	0	20	20	19	20	18	19	20	0	0	0
F-35C	0	0	0	0	0	0	0	0	0	0	0	0	0	6	11	24
INDEPENDENCE (LCS-2)	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0
FREEDOM (LCS-1)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ARLEIGH BURKE (DDG-51)	1	1	1	1	1	2	1	1	1	5	5	5	6	0	0	3
MAHAN (DDG-72)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OSCAR AUSTIN (DDG-79)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
JACK LUCAS (DDG-125)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ZUMWALT (DDG-1000)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TICONDEROGA (CG-47)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BUNKER HILL (CG-52)	0	0	0	1	1	0	1	1	1	0	0	0	0	2	2	1
WASP (LHD-1)	0	0	0	1	1	0	1	1	1	1	1	1	1	0	0	0
AMERICA (LHA-6)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FORD (CVN-78)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
AGM-84H/K	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BGM-109 Blk V	5	5	5	0	0	28	2	8	12	23	32	34	55	0	0	11
RIM-174	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AGM-158D JASSM-XR	0	0	0	27	25	0	45	39	36	45	38	36	45	28	29	49
AGM-158C LRASM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AGM-84D	6	6	6	6	7	7	8	5	7	6	8	6	5	149	144	120
AGM-84F	2	2	2	0	1	1	0	3	1	2	0	2	3	1	0	0
RGM-84F	8	8	8	16	16	16	15	16	16	40	40	39	47	16	16	32
AGM-119	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RGM-184A (NSM)	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0
AGM-114L	19	19	19	0	0	26	0	0	0	0	0	0	0	0	0	0
Navy Officer Personnel	64	66	67	133	137	109	145	154	160	221	228	234	249	422	424	468
Navy Enlisted Personnel	439	419	420	1277	1285	747	1334	1321	1322	2149	2183	2165	2444	3080	3081	3617
Navy Flight Personnel	9	11	12	15	19	15	31	31	31	29	31	31	29	91	95	89

Table 8 Initial results allocations (Continued)

	Allocations														
	16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
Objective Value	68.9	68.4	81.3	80.2	79.4	92.4	91.2	90.7	103.6	102.5	101.8	115.0	113.4	112.8	126.1
Cost	$ 317.86	$ 317.86	$ 371.43	$ 371.43	$ 371.42	$ 424.95	$ 424.94	$ 425.00	$ 478.51	$ 478.57	$ 478.57	$ 532.14	$ 532.07	$ 532.14	$ 585.68
Max Conservatism	0.7	1.2	0.2	0.7	1.2	0.2	0.7	1.2	0.2	0.7	1.2	0.2	0.7	1.2	0.2
Legacy SAR Satellite	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
Small SAR Satellite	6	6	6	6	6	6	6	6	6	5	6	5	6	6	5
EO/IR Satellite	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Comm Satellite	9	9	10	10	10	12	12	12	13	13	13	14	15	15	16
MQ-4C	20	20	20	20	19	20	20	20	20	20	20	20	20	20	20
P-8A	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
EA-18G	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
F/A-18E/F	68	65	59	63	60	20	68	65	41	67	61	119	68	64	82
MH-60S	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
F-35B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F-35C	12	15	21	17	20	60	12	15	39	12	19	41	12	15	78
INDEPENDENCE (LCS-2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FREEDOM (LCS-1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ARLEIGH BURKE (DDG-51)	3	3	5	5	5	5	3	3	5	3	5	3	3	3	3
MAHAN (DDG-72)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OSCAR AUSTIN (DDG-79)	0	0	1	1	1	1	0	0	1	0	1	1	0	0	1
JACK LUCAS (DDG-125)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ZUMWALT (DDG-1000)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TICONDEROGA (CG-47)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BUNKER HILL (CG-52)	1	1	0	0	0	0	1	1	0	1	0	2	1	1	2
WASP (LHD-1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AMERICA (LHA-6)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FORD (CVN-78)	1	1	1	1	1	1	1	1	1	1	1	2	1	1	2
AGM-84H/K	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BGM-109 Blk V	22	23	39	46	49	55	93	95	102	131	120	19	165	168	36
RIM-174	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AGM-158D JASSM-XR	33	33	46	37	36	83	33	33	63	33	36	65	33	33	100
AGM-158C LRASM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AGM-84D	142	135	125	127	118	45	141	137	88	142	125	245	140	136	171
AGM-84F	2	3	1	7	10	3	3	1	2	0	5	1	4	0	1
RGM-84F	31	32	48	48	48	48	32	32	48	32	48	48	32	32	48
AGM-119	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RGM-184A (NSM)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AGM-114L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Navy Officer Personnel	473	487	521	528	545	526	484	484	529	500	543	853	491	504	852
Navy Enlisted Personnel	3627	3630	4183	4195	4200	4207	3663	3666	4218	3678	4238	6583	3697	3701	6609
Navy Flight Personnel	91	92	89	91	93	89	93	92	89	93	94	169	91	92	171

Table 8 Initial results allocations (Continued)

	Allocations													
	31	32	33	34	35	36	37	38	39	40	41	42	43	44
Objective Value	124.7	124.0	137.3	135.6	134.9	148.5	146.7	146.1	159.6	157.6	156.5	169.5	166.7	164.6
Cost	$ 585.68	$ 585.71	$ 639.29	$ 639.28	$ 639.26	$ 692.86	$ 692.86	$ 692.86	$ 746.40	$ 746.36	$ 746.42	$ 800.00	$ 800.00	$ 800.00
Max Conservatism	0.7	1.2	0.2	0.7	1.2	0.2	0.7	1.2	0.2	0.7	1.2	0.2	0.7	1.2
Legacy SAR Satellite	0	0	1	0	0	1	0	0	1	1	0	0	0	0
Small SAR Satellite	6	5	5	6	6	5	6	6	5	5	8	8	8	8
EO/IR Satellite	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Comm Satellite	16	16	17	17	18	19	19	19	20	20	20	20	20	20
MQ-4C	20	20	20	18	20	20	20	20	20	20	20	7	1	0
P-8A	4	4	4	4	4	4	4	4	4	4	4	4	4	4
EA-18G	0	1	0	0	0	0	0	0	0	0	3	1	1	1
F/A-18E/F	63	60	101	60	61	65	63	61	83	114	107	92	88	87
MH-60S	0	0	0	0	0	0	1	0	1	1	0	0	1	0
F-35B	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F-35C	17	19	59	20	19	95	17	19	77	46	50	67	71	72
INDEPENDENCE (LCS-2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FREEDOM (LCS-1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ARLEIGH BURKE (DDG-51)	5	5	3	5	5	3	5	5	3	2	2	3	3	3
MAHAN (DDG-72)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OSCAR AUSTIN (DDG-79)	1	1	1	1	1	1	0	0	1	0	0	1	0	0
JACK LUCAS (DDG-125)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ZUMWALT (DDG-1000)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TICONDEROGA (CG-47)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BUNKER HILL (CG-52)	0	0	2	0	0	2	1	1	2	3	3	2	3	3
WASP (LHD-1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AMERICA (LHA-6)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FORD (CVN-78)	1	1	2	1	1	2	1	1	2	2	2	2	2	2
AGM-84H/K	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BGM-109 Blk V	190	193	82	228	227	100	261	263	145	174	179	211	228	230
RIM-174	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AGM-158D JASSM-XR	37	36	82	38	36	116	37	36	99	61	60	77	66	61
AGM-158C LRASM	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AGM-84D	133	127	208	120	125	138	133	125	174	236	221	191	183	113
AGM-84F	1	1	2	8	5	0	1	5	0	0	1	1	1	69
RGM-84F	48	48	48	48	48	48	48	48	48	40	40	48	48	48
AGM-119	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RGM-184A (NSM)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AGM-114L	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Navy Officer Personnel	550	557	855	547	557	863	557	577	864	847	863	849	851	854
Navy Enlisted Personnel	4267	4279	6644	4283	4299	6643	4315	4315	6668	6395	6409	6630	6655	6633
Navy Flight Personnel	91	95	169	91	92	169	93	92	171	173	180	183	175	174

CONCLUSIONS

The WRT-1049.5 research led by Purdue University adapted a previously developed system-of-systems (SoS) analytic workbench (AWB) of analytic tools to inform decisions in Integrated Acquisition Portfolio Reviews (IAPRs). The AWB we developed and enhanced supports OUSD(A&S) for the rollout of the Adaptive Acquisition Framework and Capability Portfolio Management since our software can provide the analytic capability that is necessary to provide a solid foundation for acquisition investment decisions with clear traceability.

The WRT-1049.5 Final Technical Report herein started with a brief explanation of the AWB software development, in which we discussed the overall description of the AWB tools, followed by the discussion on the specific AWB development. Moreover, the report talked about higher fidelity analysis through Discrete Event Simulation (DES), a technique we used in the WRT-1049.5 research. The report also discussed the new-technology injection into AWB, wherein we demonstrated the possibility of how we were able to incorporate the space-domain applications (i.e., older/larger satellites and newer/smaller satellites) as a part of the mission engineering and decision-making process. Finally, the report demonstrated the AWB via mission engineering analysis and portfolio optimization of an anti-surface warfare (ASuW) mission thread using personnel and munitions in the surface, aviation, and space domains

These advanced prototypes provide a broader range of insights (e.g., resource tradeoffs, cost-sensitivity analysis, and the most robust ASuW systems to be acquired in specific portfolios) for stakeholder decision-making. Future work could improve the tools to identify the following: how risk aversion affects portfolio optimization; technical dependencies among systems; developmental dependencies; and portfolio performance effects from stakeholder decisions. As a result, future work could assist the establishment of Acquisition Integration and Interoperability (AII) within OUSD(A&S).

PROJECT TIMELINE & TRANSITION PLAN

Original Research Contract Performance Period

The original research contract had the period of performance ending at the end of June 2022. To this end, Figure 17 depicts the project timeline for the original research contract.

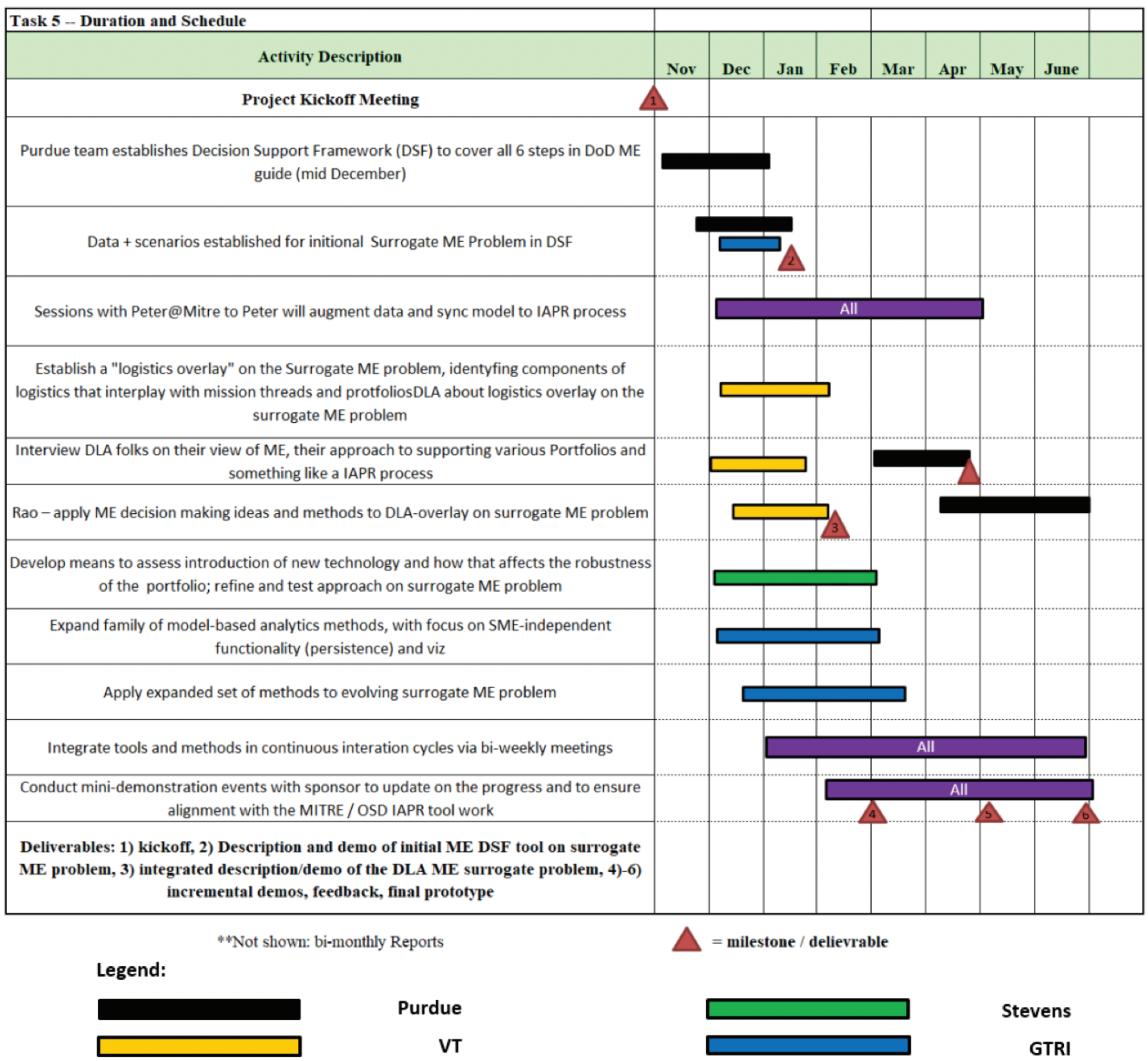

Figure 17 Project timeline for the original contract

Based on the Statement of Work for the original research contract, the participating institutions (i.e., Purdue, Virginia Tech, Stevens, and GTRI) had the following research responsibilities:

Purdue: Dan DeLaurentis and Jitesh Panchal
The Purdue team will be responsible for the main project goal of a prototype decision-support tool (DST) for the Integrated Assessment Portfolio Reviews (IAPR), including a) Architecture, development, & integration of prototype elements into the decision-support tool, b) Interfacing on data exchanges w/ongoing pilots (including especially those led by MITRE for the A&S sponsor) and c) deriving stakeholder requirements by engaging with stakeholders. As the lead for the Task, Purdue will also track key success metrics, raise risks with the sponsor if they arise, and adapt the research plan in response to needs. In particular, the Purdue team will:

- Primary interface with sponsor and team
- Lead relevant interviews and interactions with IAPR stakeholders, and specifically be responsible for ensuring detailed outcomes are communicated
- Lead the architecting and software development, and integration of the prototype DST
- Tailor and evolve prior developed analysis and optimization methods and tools to include those from SERC SoS Analytic Workbench and other work
- Track progress and risk factors associated with the DST prototype development and development courses of action to recommend to sponsor when necessary
- Set up and conduct engagement sessions with stakeholders when possible to garner feedback during the course of the effort.

Virginia Tech: Stoney Trent
Provide a human-centric, tactical approach to help engage program offices and request data in a way that is intuitive for these offices. Identify those program and portfolio artifacts most important to strategic mission engineering decision-support. Provide expertise to the research team to continuously mitigate the risk factor of insufficient data to develop the prototype ME decision-support tool. In particular:

- Attend and serve as "translator" for all interviews and interactions with IAPR stakeholders
- Define specific requirements for the prototype DST as it relates to data needs, any data modification, and ROI items for program/portfolio lead so that the DST outputs are worth the data sharing
- Help the team define the "degraded mode" functioning of the DST in cases where data is deficient as a means to address risk in IAPR

Stevens: Paul Grogan
Support rigor and innovation in models for the portion of the prototype tool that pertains to situations where assessments involve programs distributed across multiple threads and portfolios (e.g., how to resolve conflicts between objectives of multiple stakeholder entities). In particular:

- Attend relevant interviews and interactions with IAPR stakeholders
- Define the workings of and tailor for DST use methodology (and software module) for assessing the impact of new technology infusion on mission effectiveness, leveraging what he developed for NASA using game theoretic models in distributed systems
- Based on the IAPR context, define a tech infusion sensitivity input template (what data is needed, what is good to have, etc.) as well as the definition of output formats necessary for the larger DST
- Develop and integrate software that implements this tech infusion impact assessment into the DST

GTRI: Danny Browne, Craig Arndt, and Frank Patterson
Leverage experience to guide our prototype DST toward Web-Based Decision Support and Dashboards to include a combination of human and technical data in decision analysis. In particular:

- Attend relevant interviews and interactions with IAPR stakeholders, and specifically be responsible for understanding user desires from a visualization perspective
- Define the workings of and tailoring for DST using methodology (and software implementations) for visualization and dashboard realizations of the IAPR DST prototype
- Based on the IAPR context, define a data requirements template for the visualization in a manner that is evolvable as new user requirements arise
- Provide the software elements that implement this visualization and dashboard functionality into the DST

Contract Extension Performance Period

In addition to the original contract described above, we received an extension to the original contract period. The research extension allowed us to conduct research through the end of September 2022. Thus, we created a separate project timeline for the contract extension (Table 9).

Table 9 Project timeline for the contract extension

Milestone	Purdue	GTRI	Stevens	MITRE	Completion Date
Develop space domain working narrative			×		7/01/2022
Complete space domain performance analysis			×		7/15/2022
Complete space initial domain analysis interfacing with AWB		×	×		7/29/2022
Complete disassociation of RPO with MATLAB (initial Python implementation)	×	×			8/01/2022
Complete ASuW problem formulation	×	×	×	×	8/01/2022
Fully automate AWB with sub-repo version control	×	×			8/01/2022
Develop DES ASuW Model		×			8/15/2022
Complete proof-of-concept integration of external tools	×	×	×		8/25/2022
Deliver the initial draft of the final project report	×	×	×	×	9/01/2022
Continue integration of RPO optimization flavors	×	×			9/15/2022
Deliver the final AWB software prototype for DoD use	×	×	×	×	9/30/2022

Transition Plan

As of September 30, 2022, we are investigating the possibility of supporting OUSD(A&S) to establish Acquisition Integration and Interoperability (AII). Per OUSD(A&S), the AII initiative will enable greater interoperability and integration across the Services and Agencies to create a resilient Joint Force and defense ecosystem. We are hoping that supporting the AII initiative for OUSD(A&S) will pave the way for the next phase of the research.

APPENDIX A: DLA-AVIATION SUPPLY-CHAIN ANALYSIS BY VIRGINIA TECH

In addition to the research results presented in the body of the final Technical Report, Virginia Tech conducted research on the supply chain system at the Defense Logistics Agency (DLA)-Aviation. The research provided insightful information on how the supply chain data flows at DLA-Aviation. Below, we provide the summary slide deck of Virginia Tech's research (Figure 18).

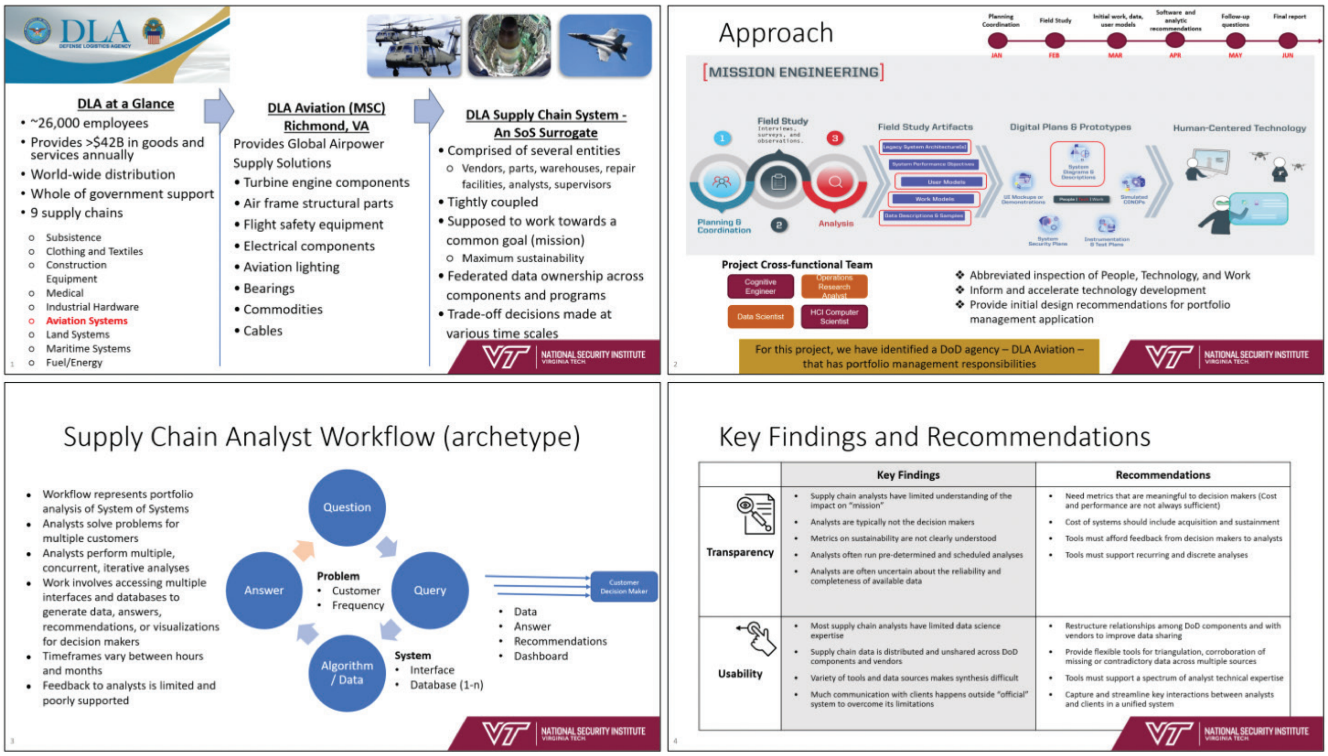

Figure 18 DLA-Aviation Supply Chain System

APPENDIX B: LIST OF PUBLICATIONS RESULTED

As of 9/28/2022, there are no publications.

APPENDIX C: CITED AND RELATED REFERENCES

Bertuca, T. *DoD Embarks on New Acquisition Portfolio Reviews*. https://www.proquest.com/docview/2578723661/fulltext/ABAF82F3637F41F8PQ/1?accountid=14052. Accessed 20 Aug. 2022.

Defense Technical Information Center. *Section 809 Panel – Defense Technical Information Center*. https://discover.dtic.mil/section-809-panel/. Accessed 20 Aug. 2022.

Department of Defense. *Defense Acquisition Guidebook*. 2013, https://at.dod.mil/sites/default/files/documents/DefenseAcquisitionGuidebook.pdf.---. *Digital Engineering Strategy*. 2018, https://man.fas.org/eprint/digeng-2018.pdf.

Erwin, S. *Space Force Looking at U.S. Needs for 'Responsive Space' - SpaceNews*. https://spacenews.com/space-force-looking-at-u-s-needs-for-responsive-space/. Accessed 20 Aug. 2022.

Green, J., and J. Stracener. "Proceedings of the Sixteenth Annual Acquisition Research Symposium." *A Framework for a Defense Systems Effectiveness Modeling and Analysis Capability: Systems Effectiveness Modeling for Acquisition*, 2019, https://dair.nps.edu/bitstream/123456789/1757/1/SYM-AM-19-067.pdf.

Guariniello, C., et al. "Quantifying the Impact of Systems Interdependencies in Space Systems Architectures." *70th International Astronautical Congress* (IAC), 2019, pp. 21–25.

Hernandez, A. S., et al. "Mission Engineering and Analysis: Innovations in the Military Decision Making Process." *Hernandez, A.S., Karimova, T., Nelson, D.H., Ng, E., Nepal, B. and Schott, E.,* 2017, https://www.researchgate.net/publication/322448165.

McDermott, T., et al. *Enterprise System of Systems Model for Digital Thread Enabled Acquisition*. 13 July 2018, https://apps.dtic.mil/sti/pdfs/AD1056642.pdf.

NASA Earth Science Technology Office. *Tradespace Analysis Tool for Constellations (TAT-C)*. https://tatc.code-lab.org/docs. Accessed 20 Aug. 2022.

Office of the Under Secretary of Defense for Research and Engineering. *Mission Engineering Guide*. 2020, https://ac.cto.mil/wp-content/uploads/2020/12/MEG-v40_20201130_shm.pdf.

SBIR-STTR. *Department of Defense (DoD) Acquisition Basics, Course 11, Tutorial 1*. https://www.sbir.gov/sites/all/themes/sbir/dawnbreaker/img/documents/Course11-Tutorial1.pdf. Accessed 20 Aug. 2022.

Section 809 Panel. *Report of the Advisory Panel on Streamlining and Codifying Acquisition Regulations, Volume 2 of 3*. 2018, https://discover.dtic.mil/wp-content/uploads/809-Panel-2019/Volume2/Sec809Panel_Vol2-Report_Jun2018.pdf.

---. *Report of the Advisory Panel on Streamlining and Codifying Acquisition Regulations, Volume 3 of 3*. 2019, https://discover.dtic.mil/wp-content/uploads/809-Panel-2019/Volume3/Sec809Panel_Vol3-Report_Jan2019_part-1_0509.pdf.

Thornberry, M. H.R.2810 - 115th *Congress (2017-2018): National Defense Authorization Act for Fiscal Year 2018*. 2017, https://www.congress.gov/bill/115th-congress/house-bill/2810#:~:text=The National Defense Authorization Act,Department of Energy (DOE).

U.S. Government Accountability Office. *Best Practices: An Integrated Portfolio Management Approach to Weapon System Investments Could Improve DOD's Acquisition Outcomes | U.S. GAO*. https://www.gao.gov/products/gao-07-388. Accessed 20 Aug. 2022.

Wikidata.Org. https://www.wikidata.org/.

Zimmerman, P., and J. Dahmann. *Digital Engineering Support to Mission Engineering*. 2018, https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2018/systems/Wed_21339_DahmanZimmerman.pdf.

DISCLAIMER

The Acquisition Innovation Research Center is a multi-university partnership led and managed by the Stevens Institute of Technology and sponsored by the U.S. Department of Defense (DoD) through the Systems Engineering Research Center (SERC)—a DoD University-Affiliated Research Center (UARC).

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD(A&S)) and the Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E)) under Contract HQ0034-19-D-0003, TO#0309.

SYSTEMS ENGINEERING RESEARCH CENTER

AIRC ACQUISITION INNOVATION RESEARCH CENTER

STEVENS INSTITUTE OF TECHNOLOGY 1870