SYM-AM-23-081

Excerpt from the Proceedings

of the

Twentieth Annual Acquisition Research Symposium

Acquisition Research:
Creating Synergy for Informed Change

May 10–11, 2023

Published: April 30, 2023

The research presented in this report was supported by the Acquisition Research Program at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

Acquisition Research Program
Department of defense management
Naval Postgraduate School

Model-Based Approach in Defense Portfolio Management: Data Preparation, Analysis, and Visualization of Decision Spaces

Waterloo Tsutsui—is a Senior Research Associate in the School of Aeronautics and Astronautics at Purdue University, IN. Tsutsui received his PhD in Aeronautics and Astronautics from Purdue University in 2017. Before Purdue, Tsutsui practiced engineering in the automotive industry for more than 10 years, with the last position involving the research and development of lithium-ion battery cells for electric vehicles. Tsutsui's research interests are systems engineering, mission engineering, energy storage systems, multifunctional structures and materials design, and the scholarship of teaching and learning. [wtsutsui@purdue.edu]

Cesare Guariniello—is a Research Scientist in the School of Aeronautics and Astronautics at Purdue University. He received his PhD in Aeronautics and Astronautics from Purdue University in 2016 and his master's degrees in Astronautical Engineering and Computer and Automation Engineering from the University of Rome "La Sapienza." Guariniello works as part of the Center for Integrated Systems in Aerospace led by DeLaurentis, and is currently engaged in projects funded by NASA, the DoD Systems Engineering Research Center (SERC), and the NSF. His main research interests include modeling and analysis of complex systems and SoS architectures—with particular focus on space mission architectures—aerospace technologies, and robotics. Guariniello is a senior member of IEEE and AIAA. [cguarini@purdue.edu]

Kshitij Mall—is a Post-doctoral Research Associate at the Center for Integrated Systems in Aerospace, Purdue University. He obtained his PhD and master's degrees from the School of Aeronautics & Astronautics, Purdue University. He was a Post-doctoral Research Fellow in the department of Aerospace Engineering at Auburn University in 2019. Previously, he completed B. Tech. in Mechanical Engineering at JSSATE Noida, India and then worked for a year at Infosys Technologies Ltd. as a Computer Systems Engineer Trainee. His research interests lie in the areas of Systems Engineering, Atmospheric Flight Mechanics, Explainable Artificial Intelligence, and Human-Class Mars missions.

Frank Patterson—is a Senior Research Engineer at the Georgia Tech Research Institute (GTRI) in the Systems Engineering Research Division (SERD). His current research includes the application of state of the art of computational methods and tools to the design and analysis of complex systems. He is also experienced in the development, integration, and use of multi-disciplinary simulation, modeling, and analyses for the design of systems under uncertainty. He has more than 15 years of experience supporting the DoD and warfighter as an engineer across various domains. Patterson earned his bachelor's degree, master's degree, and PhD in Aerospace Engineering at Georgia Tech.

Santiago Balestrini-Robinson—is the head of the Methods and Analysis Developments Branch (MADB) in the Electronic Systems Laboratory (ELSYS) of the Georgia Tech Research Institute (GTRI). His primary area of research is the development of collaborative quantitative and qualitative decision support tools and frameworks. Balestrini-Robinson has led teams supporting multibillion-dollar military acquisition programs, requirements analysis studies for novel operational and materiel concepts, as well as the development of general frameworks to support collaborative and executable Model-based Systems Engineering. He earned a BS, an MS, and PhD in Aerospace Engineering from the Georgia Institute of Technology in 2003, 2006, and 2009, respectively.

Jitesh Panchal—is a Professor of Mechanical Engineering at Purdue University. He received his BTech from Indian Institute of Technology (IIT) Guwahati, and MS and PhD from Georgia Institute of Technology. He is a member of the Systems Engineering Research Center (SERC) Council. He is a recipient of NSF CAREER award; Young Engineer Award and three best paper awards from ASME; and was recognized by the Schaefer Outstanding Young Faculty Scholar Award, the Ruth and Joel Spira Award from Purdue University. He is a co-author of two books and a co-editor of one book on systems design. [panchal@purdue.edu]

Daniel DeLaurentis—is Vice President for Discovery Park District Institutes and Professor of Aeronautics and Astronautics at Purdue University. He leads the Center for Integrated Systems in Aerospace (CISA) activities on research problem formulation, modeling, and systems engineering methods for aerospace systems and system-of-systems. DeLaurentis also serves as Chief Scientist of the U.S. DoD's SERC UARC to understand the systems engineering research needs of the defense community (primarily) and translate that to research programs that are then mapped to the nation's best researchers in the SERC's university network. He is a Fellow of the INCOSE and the AIAA. [ddelaure@purdue.edu]

Abstract

The research team adapted a previously developed system-of-systems analytic workbench to address Integrated Acquisition Portfolio reviews via mission engineering analysis. The team illustrated the findings to date in developing decision-support tools tailored to the needs of these reviews and the insights they produce for improved acquisition outcomes. The essence of the prototype acquisition decision-support tools we are developing is a combination of portfolio optimization and mission engineering. We explore the interactions between candidate systems to acquire and existing systems to identify capability gaps and features of portfolios that optimally cover a family of mission threads. Moreover, we investigate the role of digital engineering in facilitating this process to shift the stakeholders' mindset from the traditional forms of acquisition decision-making to a predominantly model-based approach, from data preparation, analysis, and visualization of the decision spaces. Preliminary 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. Longer-term goals include a more comprehensive model-based acquisition decision-support system, with consistent data definitions extracted from "authoritative sources of truth," thereby connecting all models with common data definitions.

Introduction

The research team developed a pilot/prototype capability to enhance data-driven decision-making regarding acquisition and sustainment programs, motivated by the context of the Department of Defense's (DoD) Integrated Acquisition Portfolio Review (IAPR) process. As the DoD transforms its acquisition paradigm from centralized oversight of Acquisition Category (ACAT) 1D programs to decentralized oversight delegated across Components, the Office of the Under Secretary of Defense for Acquisition & Sustainment (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.

The research team focuses on a portfolio-centric approach, which we implemented by enhancing and adapting an existing research product called the System-of-Systems Analytic Workbench (AWB). The 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 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.

The research team also explored the enablement of portfolio management from a mission engineering perspective. The two guiding principles for this research 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 research 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? The research included engagement with selected mission portfolio managers to understand their priorities and challenges to enable evidence/data-based portfolio management.

This paper starts with an explanation of the AWB development, wherein we describe the overall description of the AWB tools, followed by a discussion on the development of the AWB tools. The paper also demonstrates the enhanced AWB using a notional anti-surface warfare (ASuW) application to illustrate its application to a non-trivial domain.

Analytic Workbench Development

The AWB is a collection of methods and techniques 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.

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, 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 AWB and in the Decision Support Framework (DSF), the framework that used RPO and SODA sequentially.

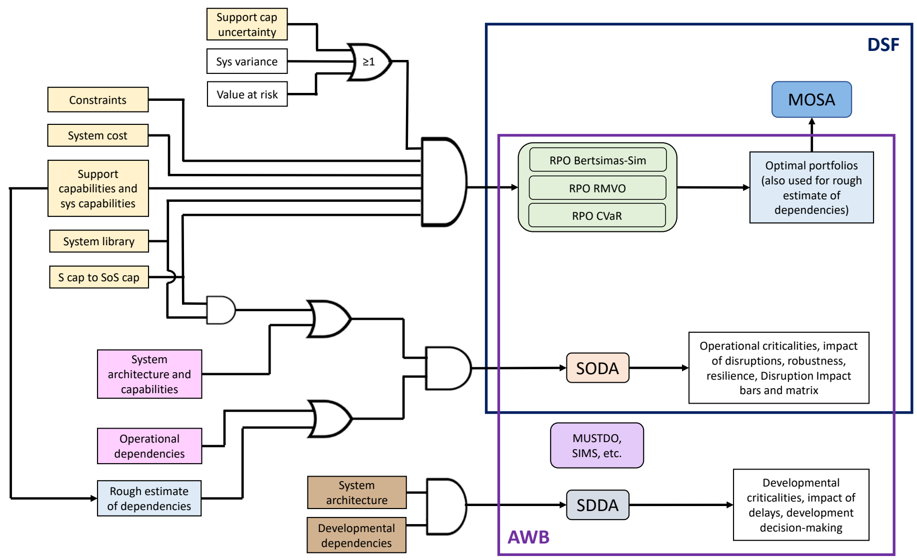

Figure 1. Inputs to AWB and DSF

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: (1) Robust Mean Variance Optimization (RMVO) includes developmental risks (Davendralingam & Delaurentis, 2015; Rubinstein, 2002), (2) the Bertsimas-Sim method involves operational risks (Bertsimas & Sim, 2004; Davendralingam & DeLaurentis, 2013), and (3) Conditional Value at Risk (CVaR) addresses simulated risks (Davendralingam & DeLaurentis, 2014; Shah et al., 2015). 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., 2019). 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., 2019).

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.

AWB Interoperability, Extensibility, and Usability Upgrades

A significant part of this research effort included upgrades to the 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 (GitHub Resources, 2022). 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 (Jupyter, n.d.) 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 the tool suite 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.

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. JSON Schema (a standard for developing these validations; JSON Schema, n.d.) is used. Generating the JSON Schema 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 JSON Schema 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.

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.

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 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.

Anti-Surface Warfare (ASuW) Problem

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 U.S. Navy (Broadfoot et al., 2018; Kaymal, 2013; Neumann, 2021). 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 U.S. 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 2.

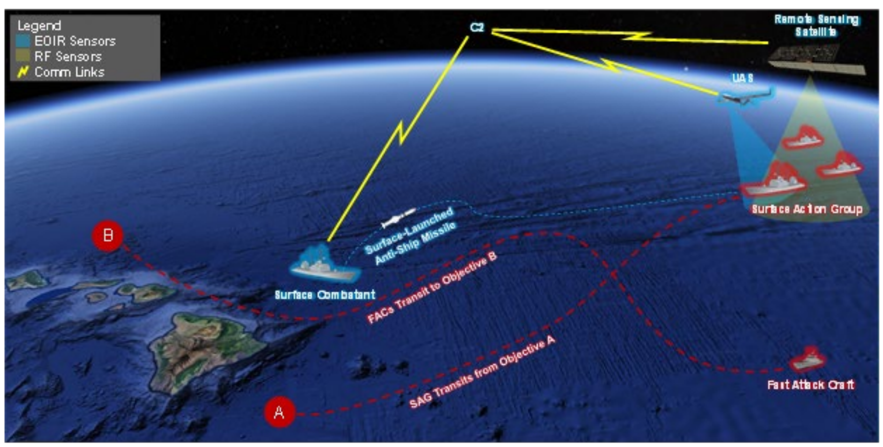

Figure 2. OV-1 of the Simple Notional Anti-Surface Warfare Scenario

A more comprehensive construct based on a richer kill web is depicted in Figure 3. 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.

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.

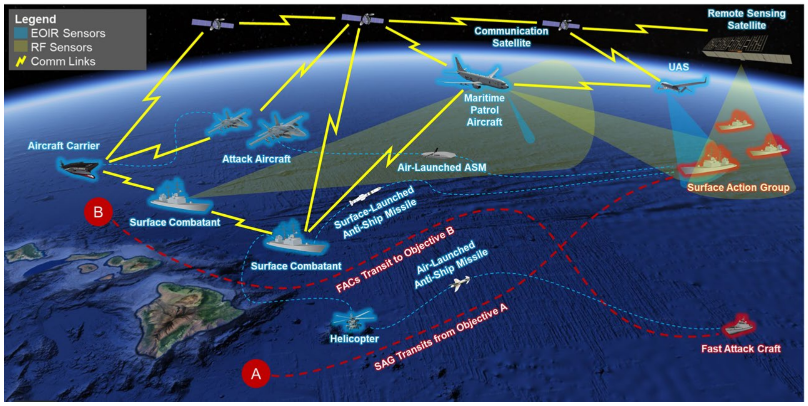

Figure 3. OV-1 of a More Comprehensive Anti-Surface Warfare Scenario

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 of 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 with 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 standoff 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 1. 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, the 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 1. 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*, n.d.)

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 2) was to illustrate that a notional capability/cost tradeoff could be captured by the 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 2. 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*, n.d.)

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 3). These are utilized by RPO and can be passed to supporting analysis (e.g., the Discrete Event Simulation developed using UPSTAGE; Arruda, 2018) 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 4). These SoS Capabilities are groupings of the individual system capabilities.

Table 3. 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	Standoff Range	Weapon Range (nm)	nmi
SC 5	Disable Surface Combatant	P_{hit} SC	%
SC 6	Damage Surface Combatant	$P_{kill\|hit}$ SC	%
SC 7	Disable Fast Attack Craft	P_{hit} FAC	%
SC 8	Damage Fast Attack Craft	$P_{kill\|hit}$ FAC	%
SC 9	Quickness	Airspeed	kts
SC 10	Coverage	Flight Range	nmi
SC 11	Power Projection	Capacity	#

Table 4. SoS Capabilities Defined for the Overall Scenario

	SC 1	SC 2	SC 3	SC 4	SC 5	SC 6	SC 7	SC 8	SC 9	SC 10	SC 11
SoS-Capability	**Maritime Surveillance**	**Identify Surface Contacts**	**Jam Ship Radars**	**Standoff 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 4). 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.

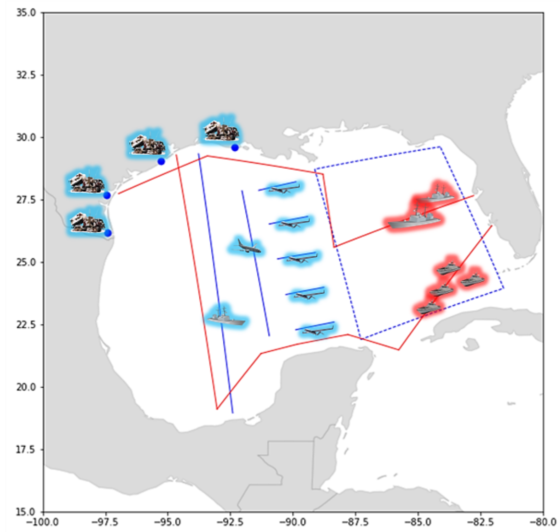

Figure 4. Red (Florida) vs. Blue (Texas)

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 4. 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 5). 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.

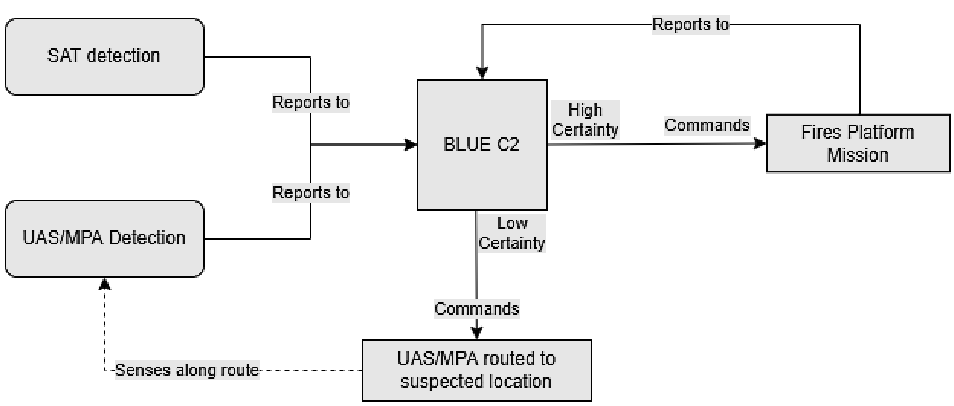

Figure 5. F2T2EA Kill-Chain

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 usage, such as fuel and munitions, comms requirements, and other interactions.

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 5.

The results of the RPO run can be viewed in Figure 6, where SoS Performance Index is a non-dimensional measure indicating performance across all selected SoS Capabilities. For this initial example, all five 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 6 shows how many of each system were purchased for the points plotted in Figure 6, a run of RPO on the ASuW scenario.

Table 5. Initial Results Run Parameters

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

From Table 6, 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.

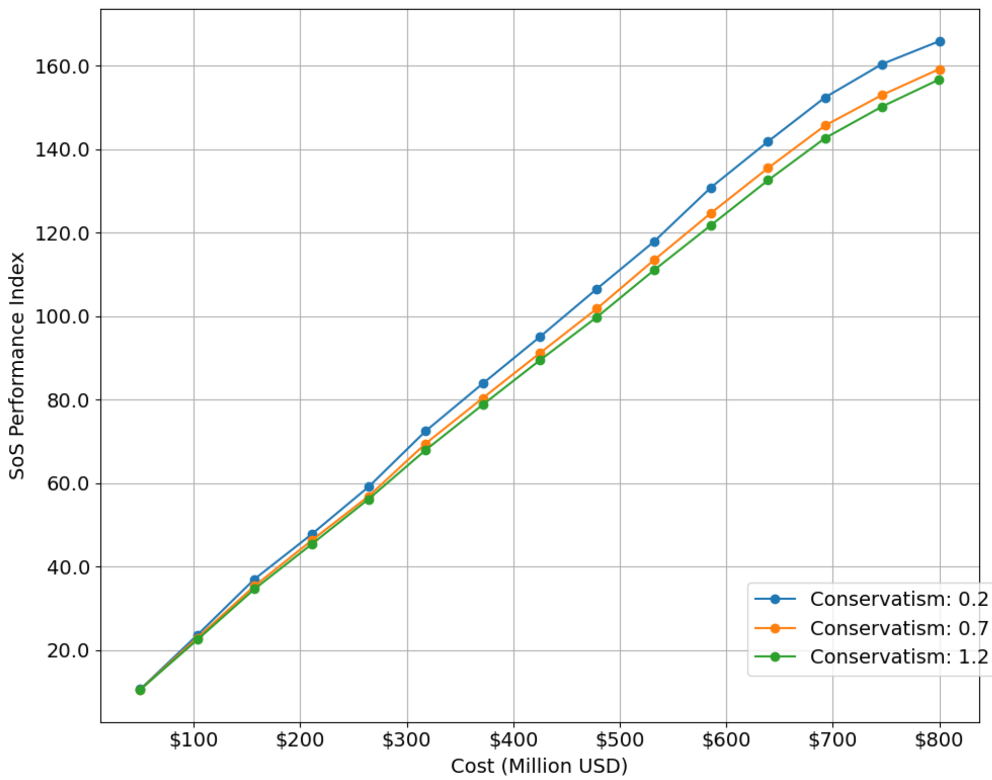

Figure 6. Initial Results: Cost vs. SoS Performance

Table 6. Initial Results Allocations

rnative	0	1	2	3	4	5		39	40	41	42	43	44
Objective Value	10.7	10.7	10.7	22.6	22.2	21.8		159.6	157.6	156.5	169.5	166.7	164.6
Cost	$ 50.00	$ 49.94	$ 49.96	$ 103.53	$ 103.53	$ 103.57		$ 746.40	$ 746.36	$ 746.42	$ 800.00	$ 800.00	$ 800.00
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
Legacy SAR Satellite	0	0	0	0	0	0		1	1	0	0	0	0
Small SAR Satellite	1	1	1	2	2	2		5	5	8	8	8	8
EO/IR Satellite	0	0	0	0	0	0		0	0	0	0	0	0
Comm Satellite	2	2	2	4	4	4		20	20	20	20	20	20
MQ-4C	12	12	12	15	15	20		20	20	20	7	1	0
P-8A	4	4	4	3	4	4		4	4	4	4	4	4
EA-18G	0	0	0	0	0	0		0	0	3	1	1	1
F/A-18E/F	0	0	0	0	0	0		83	114	107	92	88	87
MH-60S	0	0	0	0	0	1		1	1	0	0	1	0
F-35B	0	0	0	8	8	0		0	0	0	0	0	0
F-35C	0	0	0	0	0	0		77	46	50	67	71	72
INDEPENDENCE (LCS-2)	0	1	1	0	0	1		0	0	0	0	0	0
FREEDOM (LCS-1)	1	0	0	0	0	0		0	0	0	0	0	0
ARLEIGH BURKE (DDG-51)	1	1	1	1	1	2		3	2	2	3	3	3
MAHAN (DDG-72)	0	0	0	0	0	0		0	0	0	0	0	0
OSCAR AUSTIN (DDG-79)	0	0	0	0	0	0	Alternatives 6-38 were omitted from this figure	1	0	0	1	0	0
JACK LUCAS (DDG-125)	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
TICONDEROGA (CG-47)	0	0	0	0	0	0		0	0	0	0	0	0
BUNKER HILL (CG-52)	0	0	0	1	1	0		2	3	3	2	3	3
WASP (LHD-1)	0	0	0	1	1	0		0	0	0	0	0	0
AMERICA (LHA-6)	0	0	0	0	0	0		0	0	0	0	0	0
FORD (CVN-78)	0	0	0	0	0	0		2	2	2	2	2	2
AGM-84H/K	0	0	0	0	0	0		0	0	0	0	0	0
BGM-109 Blk V	5	5	5	0	0	28		145	174	179	211	228	230
RIM-174	0	0	0	0	0	0		0	0	0	0	0	0
AGM-158D JASSM-XR	0	0	0	27	25	0		99	61	60	77	66	61
AGM-158C LRASM	0	0	0	0	0	0		0	0	0	0	0	0
AGM-84D	6	6	6	6	7	7		174	236	221	191	183	113
AGM-84F	2	2	2	0	1	1		0	0	1	1	1	69
RGM-84F	8	8	8	16	16	16		48	40	40	48	48	48
AGM-119	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
AGM-114L	19	19	19	0	0	26		0	0	0	0	0	0
Navy Officer Personnel	64	66	67	133	137	109		864	847	863	849	851	854
Navy Enlisted Personnel	439	419	420	1277	1285	747		6668	6395	6409	6630	6655	6633
Navy Flight Personnel	9	11	12	15	19	15		171	173	180	183	175	174

Conclusions

The research team adapted a previously developed SoS-AWB to inform decisions in 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 suite can provide the analytic capability that is necessary to provide a solid foundation for acquisition investment decisions with clear traceability. These advanced prototypes provide broader insights (e.g., resource tradeoffs, cost-sensitivity analysis, and the most robust ASuW systems to be acquired in specific portfolios) for the stakeholder's decision-making process. 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 in the activities for the new Acquisition Integration and Interoperability Office within OUSD(A&S).

Acknowledgments

The authors acknowledge financial support from the U.S. Department of Defense through SERC/AIRC on research task WRT 1049.5, contract no. HQ0034-19-D-0003 and report no. AIRC-2022-TR-007.

In addition, the authors thank the following individuals who have significantly contributed to the development of the SoS tools in various SERC and AIRC projects. Contributing to the original development of the Analytic Workbench: Dr. Karen Marais, Dr.

Payuna Uday, Dr. Navindran Davendralingam, Dr. Zhemei Fang, Mr. Rakshit Chandrahasa, Dr. Judith Dahmann, Mr. Scott Lucero. Supporting further development and improvement: Dr. Kris Ezra, Mr. Robert Campbell, Mr. Clint Hanthorn, Dr. Paul Grogan, Dr. Brian Chell, Mr. Benjamin Stanley, Mr. Corey Batchelder, Dr. Daniel Browne, Dr. Craig Arndt, Dr. Zachary Welz, Mr. James Arruda, Mr. Joel Stansbury, Mr. Nicholas Bollweg, Dr. Valerie Sitterle, Mr. Joshua Mathews, Mr. Lucas Karinshak, and Dr. Peter Korfiatis.

References

Arruda, J. (2018). UPSTAGE: Universal platform for simulating tasks and actors with graphs and events. In *86th MORS Symposium*.

Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research*, *52*(1), 35–53.

Broadfoot, M., Bush, C., Harpel, B. L., Lajoie, T., Laube, P. H., O'Grady, M. R., Overman, E. A., & Parcus, A. (2018). *Examining operational and design effects of MH-60S with enhanced weapon systems in anti surface warfare missions*.

Davendralingam, N., & DeLaurentis, D. (2013). A robust optimization framework to architecting system of systems. *Procedia Computer Science*, *16*, 255–264.

Davendralingam, N., & DeLaurentis, D. (2014). An analytic portfolio approach to system of systems evolutions. *Procedia Computer Science*, *28*, 711–719.

Davendralingam, N., & Delaurentis, D. A. (2015). A robust portfolio optimization approach to system of system architectures. *Systems Engineering*, *18*(3), 269–283.

GitHub Resources. (2022). *The fundamentals of continuous integration in DevOps*. https://resources.github.com/devops/fundamentals/ci-cd/integration/

Guariniello, C., Grande, M., Brand, C., Durbin, L., Dai, M., Das-Stuart, A., Alexander, R., Howell, K., & DeLaurentis, D. (2019). Quantifying the impact of systems interdependencies in space systems architectures. *70th International Astronautical Congress*.

JSON Schema. (n.d.). *JSON schema*. Retrieved March 31, 2023, from https://json-schema.org/

Jupyter. (n.d.). *Jupyter notebook*. Retrieved March 31, 2023, from https://jupyter.org/

Kaymal, T. (2013). *Assessing the operational effectiveness of a small surface combat ship in an anti-surface warfare environment*.

Neumann, N. (2021). *The shifting threats driving anti-surface warfare (ASuW) capabilities*. Naval Technology. https://www.naval-technology.com/features/anti-surface-warfare-asuw/

Rubinstein, M. (2002). Markowitz's "portfolio selection": A fifty-year retrospective. *The Journal of Finance*, *57*(3), 1041–1045. http://www.jstor.org/stable/2697771

Shah, P., Davendralingam, N., & DeLaurentis, D. A. (2015). A conditional value-at-risk approach to risk management in system-of-systems architectures. *2015 10th System of Systems Engineering Conference (SoSE)*, 457–462.

wikidata.org. (n.d.). https://www.wikidata.org/

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net