3-2022

Applied Agile Digital Mission Engineering for Cislunar Space Domain Awareness

Benjamin R. Williams

APPLIED AGILE DIGITAL MISSION ENGINEERING FOR CISLUNAR SPACE DOMAIN AWARENESS

THESIS

Benjamin R. Williams, Major, USSF

AFIT-ENY-MS-22-M317

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENY-MS-22-M317

APPLIED AGILE DIGITAL MISSION ENGINEERING FOR CISLUNAR SPACE DOMAIN AWARENESS

THESIS

Presented to the Faculty

Department of Aeronautics & Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Space Systems

Benjamin R. Williams, MBA, BS

Major, USSF

March 24, 2022

AFIT-ENY-MS-22-M317

APPLIED AGILE DIGITAL MISSION ENGINEERING FOR CISLUNAR SPACE DOMAIN AWARENESS

THESIS

Benjamin R. Williams, MBA, BS
Major, USSF

Committee Membership:

Lt Col Bryan Little, Ph.D
Chair

Lt Col Amy Cox, Ph.D
Member

David Meyer
Member

AFIT-ENY-MS-22-M317

Abstract

In the backdrop of an expansion into cislunar space and a digital transformation, the author synthesizes a methodology from agile system development, Digital Engineering (DE), Mission Engineering (ME), and Model-based Systems Engineering (MBSE) processes, methods, and tools to develop a basic Reference Architecture (RA) and Digital Thread (DT) for cislunar Space Domain Awareness (SDA) mission and system design. The Agile Digital Mission Engineering (DME) methodology is used to conduct basic requirements analysis, develop a concept, understand cislunar physics, study scenario geometries, and perform analyses and the DT implements, executes, and accelerates research by integrating descriptive, analytical, and simulation tools. This research demonstrates the value of automated Modeling, Simulation, and Analysis (MS&A) workflows and how DE can aid in meeting the Department of Defense (DoD) vision to prioritize speed of delivery within rapidly changing operational environments, on limited budgets, and in short timelines using a model-analyze-build methodology.

Table of Contents

Page

List of Figures

Figure Page

List of Tables

APPLIED AGILE DIGITAL MISSION ENGINEERING FOR CISLUNAR SPACE DOMAIN AWARENESS

I. Overview

Cislunar trajectories, such as the free-return trajectory developed for the Apollo missions, can easily be repurposed to hold near-Earth and terrestrial targets at risk while reducing the probability of detection and attribution [1]. This is a problem for the United States Space Force (USSF) since it has a mission to characterize, understand, and attribute objects within the space domain that "could affect space operations and thereby impact the security, safety, economy, or environment of [the United States of America]".[2, 3] Space Domain Awareness (SDA) provides this knowledge and characterization to promote "the safety, stability, and sustainability of operations in the space environment".[4] A cislunar SDA mission architecture must be able to detect and track an object within the cislunar 4π steradian search volume that extends to the Earth-Moon L_2 point, as seen in Figure 1, which encompasses roughly three times the search volume as the traditional geosynchronous Earth orbit (GEO) space situational awareness mission.[1]

The United States does not have a dedicated SDA system capable of attributing spacecraft beyond GEO despite the increase in cislunar operations. However, the USSF must field a system capable of conducting cislunar SDA to support space security and Freedom of Maneuver.[2] The Space Development Agency recently publicized the study and inclusion of a space deterrence layer within the National Space Defense Architecture, including potentially, some 200 satellites in low Earth orbit (LEO) designed to provide timely detection and attribution of threatening deep space activities. The emergence of this problem set also coincides with a strategy push from the Department of Defense (DoD) to implement Digital

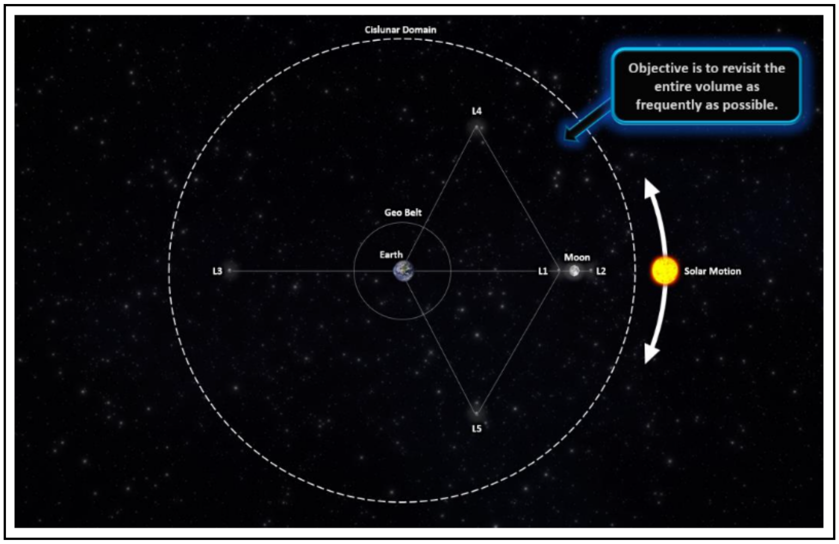

Figure 1: Depiction of the Cislunar Domain [1]

Engineering (DE).

The DoD is pursuing this transformation to modernize "defense systems and prioritize speed of delivery to be able to fight and win the wars of the future."[5] The goal is to balance design, delivery, and sustainment within rapidly changing operational environments, on limited budgets, and in short timelines. There is a desire to shift from a design-build-test to a model-analyze-build methodology in hopes of attaining this goal. The purpose of DE is to evolve at the conceptual phase to reduce design lock, limit mockups and physical testing, and promote use of digital representations of systems to communicate amongst stakeholders and across the life-cycle of defense systems. This digital transformation will use integrated virtual environments, or digital engineering toolchains, "to increase customer and vendor engagement, improve threat response timelines, foster infusion of technology, reduce cost of documentation, and impact sustainment affordability."[5] This integrated environment will serve to securely connect people, processes, and data across a digital enterprise. This virtual world will enable stakeholders to interact and solve problems in a geographically separated way, perfect for the dispersed nature of the military. The hope of DE is that it reduces "challenges associated with complexity, uncertainty, and rapid change in deploying and us-

ing" defense systems by providing a more agile and responsive development environment.[5] The expected benefits include better informed decision making, enhanced communication, and more efficient processes.

1.1 Problem

A comprehensive cislunar SDA system to maintain attribution of objects in cislunar space does not exist. Additionally, the Air Force Institute of Technology (AFIT) lacks access to a comprehensive digital ecosystem and in particular, Digital Threads (DTs) designed to accelerate research for cislunar SDA. Furthermore, there is not a published methodology that synthesizes agile managerial and Model-based Systems Engineering (MBSE) technical processes, methods, and tools to pursue the rapid development of a cislunar SDA architecture.

1.2 Hypothesis

The hypothesis of this thesis is that a comprehensive Agile Digital Mission Engineering (DME) methodology can accelerate life-cycle modeling and evaluation to create high-confidence, decision-making information for cislunar SDA mission and system designs. The primary research objectives are to:

1. Understand the implications the cislunar environment has on mission and system design,
2. Demonstrate the value of a Reference Architecture (RA) in the design process,
3. Automate the simulation and evaluation of a system model within a DT.

1.3 Research Questions

Due to the broad nature of this hypothesis, multiple high-level research questions are posed but are actually composed of multiple layers of more specific research questions that

will be addressed using the methodology described in Chapter 3.

The primary research questions are:

1. What methodology should be used to study the operational domain while adhering to system design principles?

2. Can a RA be created for cislunar SDA mission and system design?

3. Can a DT be constructed to evaluate the effectiveness of alternative systems?

1.4 Assumptions and Limitations

While there are non-materiel solutions capable of mitigating risks associated with nefarious uses of cislunar space within the spectrum of Doctrine, Organization, Training, materiel, Leadership, Personnel, Facilities, and Policy (DOTmLPFP), this research will focus on a materiel solution, specifically an Electro-optical (EO) sensing constellation, for performing SDA. While studies like Little's investigate sensor tasking algorithms, this research is focused on the more simplified geometric SDA detection problem by assuming the sensor has apriori knowledge of the target's location and the sensor's boresight is always aligned with the access vector to the target.[6]

Additionally, this research is further simplified to focus on the stated objectives and limit required computational resources by modeling dynamics within a two-body system and the Circular Restricted Three-Body Problem (CR3BP). For satellites orbiting near Earth, Keplarian motion is used, while motion of satellites in cislunar space are described using three-body Equations of Motion (EOM). The EOM for the CR3BP are used to generate first order approximations of Lagrange point orbits, and a multiple-shooting method is utilized to generate more realistic orbits.

Within the physical Electro-magnetic (EM) model, only the Sun's radiance in the visible band is used as an energy source, excluding albedo from both the Earth and the Moon.

This source energy irradiates a spherical threat object with a Lambertian reflection and all power reflected from the threat object is assumed to focus on a single pixel in the sensor's detector due to the extreme distances involved during the observations. Many of the assumptions presented within this thesis come from research Knister conducted to minimize variations during the verification process.[7] Additional assumptions are discussed throughout the methodology section as the DT is further elaborated.

1.5 Expected Outcomes and Impacts

An applied Agile DME methodology is expected to produce a DT capable of modeling, simulating, and analyzing the performance of multiple sensors against multiple threats transiting cislunar space. The resulting DT will include a cislunar SDA RA that contains mission, system, subsystem models and diagrams. This SDA RA will accelerate the development of future Mission Threads (MTs) and associated systems. The DT will also integrate a relevant physical environment to simulate the dynamics of spacecraft and analyze the detection performance of each spacecraft against threat representative systems. Demonstrating the utility of a DT provides additional evidence and motivation to pursue digital transformation within the DoD.

1.6 Thesis Overview

Digital capabilities will allow the DoD and its mission partners to understand and solve problems quicker, expediting the decision-making loop ahead of adversaries. This particular research focuses on the performance characteristics of systems that need to maintain attribution of uncooperative targets transiting cislunar space. This thesis is written in a comprehensive, incremental approach using the Agile DME methodology, with a complete draft thesis delivered at the end of each Sprint. The Overview, Chapter 1, contains research questions and hypotheses derived from the Mission Engineering (ME) Mission Definition ac-

tivity and the Object-Oriented Systems Engineering Method (OOSEM) Stakeholder Analysis activity. The Background, Chapter 2, discusses previous research conducted and principles, processes, and methods that have been used or could be used to pursue the research questions. The Methodology, Chapter 3, synthesizes various principles, processes, and methods into an overarching methodology and execution plan. Results and Analysis, Chapter 4, discusses the construction of the DT and the generated results. Finally, Conclusion, Chapter 5, provides answers to the research questions and verifies the proposed hypothesis.

II. Background

Designing a Space Domain Awareness (SDA) architecture requires the use of both a suitable architecting and Digital Engineering (DE) methodology. This research seeks to both enhance and expand upon previous space-based SDA architectures by implementing an architecting process and Digital Thread (DT) to explore the problem-solution trade space and rapidly converge on solution concepts. Scheurer asserts the Mission Engineering (ME), Systems Engineering (SE), and DE methodologies all share a common goal to create an architecture to satisfy a need.[8] The goal of architecting is to establish key concepts and principles to solve a problem and to prevent waste in creating a solution.

Architecting methodologies lend themselves to the exploration of novel problem sets, such as cislunar SDA. ME is focused on the coupling and interoperability of a System-of-Systems (SoS) architecture while Model-based Systems Engineering (MBSE) focuses on the functional and physical decomposition of a system development architecture. DE is focused on the process interfaces of the engineering workflow to include simulation and analysis. And, finally, Model-based Engineering focuses on physics-based element descriptions for a system.[8]

Section 2.1 provides an overview of ME and SDA functions and efforts to better understand the physical implications of cislunar space. This section investigates the dynamics and families of cislunar orbits and their potential usefulness in conducting cislunar SDA. Section 2.2 investigates how DE supports the architecting goals to reduce ambiguity, enable creativity, and manage complexity through modeling, simulating, and evaluating processes. Finally, Section 2.3 investigates how Agile development principles and practices facilitate the quick delivery of verified value to end-users.

2.1 Mission Engineering

A mission describes the context, the purpose, and logical behavior of a system.[9] DoD Joint Publication 3-0 defines mission as "the task, together with the purpose, that clearly indicates the action to be taken and the reason thereby."[10] ME focuses on mission objectives and associated measures of success and effectiveness with respect to a logical, capability-based architecture. ME analyzes mission goals and evaluates available and emerging capabilities to develop an architecture capable of meeting mission objectives. ME is associated with system-of-systems because missions are accomplished through the coordination and interoperability of heterogeneous systems.[11] The mission is accomplished by operational nodes completing one or more operational activities.[9]

Joint Publication 3-14 functionally decomposes SDA into "detect/track/identify, integrated tactical warning, and attack assessment and characterization" functions.[10] This research will use the ME process to evaluate the ability of an SDA SoS architecture to perform the detection function. At a high level, the ME process, depicted in Figure 2, contains the following activities:

1. define problem statement by generating a mission question,
2. characterize the mission by documenting assumptions and constraints,
3. define mission metrics and hypothesize a Doctrine, Organization, Training, materiel, Leadership, Personnel, Facilities, and Policy (DOTmLPFP) solution,
4. design an analysis,
5. and perform analysis to identify mission drivers.[11]

Delving into more detail, the Mission Definition and Characterization phase describes the entry conditions and boundaries such as the operational environment and the commander's desired intent or objectives for a particular mission as depicted in Figure 3.[11]

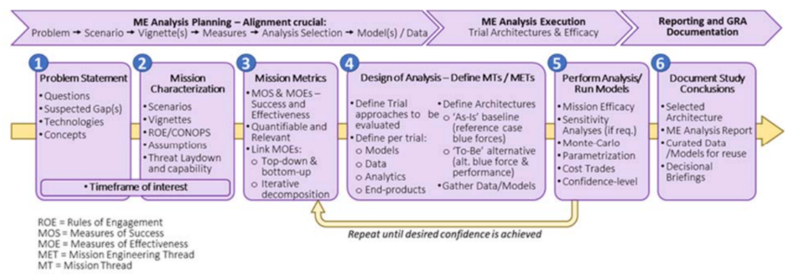

Figure 2-1. Mission Engineering Approach and Methodology

Figure 2: Mission Engineering Process [11]

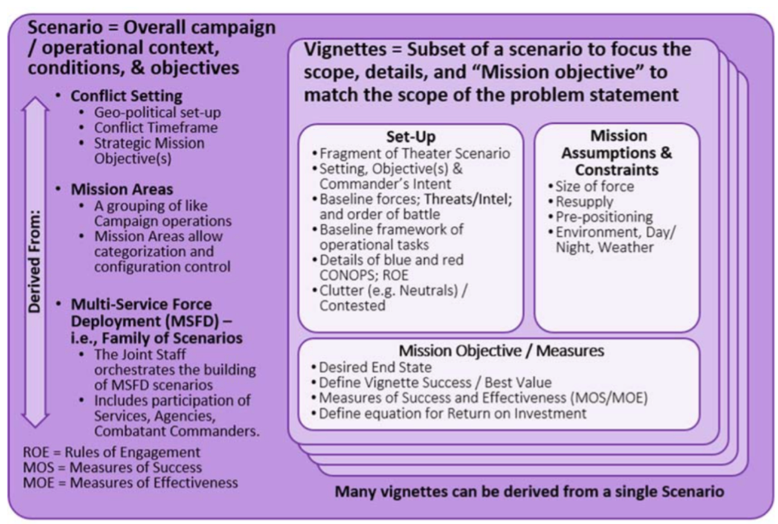

Figure 2-2. Mission Characterization and Mission Metrics

Figure 3: Mission Characterization [11]

Incorporating information on the operational context helps refine the problem statement; some of these details can be seen in Figure 4. Many of these details can be found in Operational Planning guidance, Joint Warfighting Publications, Defense Planning Scenarios, and Concept of Operations (CONOPs).[11]

Table 2-1. Categories of Mission Definition

Linkage to "Strategic" Constructs	
• National Defense Strategy (NDS) • Defense Planning Guidance (DPG) • National Military Strategy (NMS) • Defense Planning Scenario (DPS) (Strategic Scenario)	• Multi-Service Force Deployment (MSFD) (Operational Scenario derived from DPS) • Mission Areas • Campaign Objective • Joint or Service Concept of Operations (CONOPS) and Tactics, Techniques, and Procedures (TTPs)
Scenario and Vignette Specifics	
<u>Mission Setting</u> • Time frame (year) • Phase of conflict • Time available • Theater/setting • Objective(s) and Commander's Intent • Geopolitical considerations • Environmental (contested, dust, weather, terrain, day/night, weather, moon, solar, etc.) • Allied, commercial, and neutral forces • Civil considerations	<u>Threat (Red) Forces</u> • Threats/capabilities/intel • Baseline forces and Order of Battle (OOB) • Threat CONOPS, Rules of Engagement (ROE), doctrine, or TTPs <u>Blue (to include coalition) Forces Baseline Capability (in time frame of interest)</u> • Systems/capabilities/performance • CONOPS, return on investment, doctrine, etc. • Baseline framework of operational tasks
Mission Assumptions and Constraints	
• Performance or capabilities of systems • Force OOB and deployment, movement and sustainment (prepositioning); logistical considerations	• Constituents of the mission and how it must interact with other mission areas

Figure 4: Mission Categories [11]

Missions are hierarchical, from strategic to tactical. A mission is composed of one or more scenarios which provides the overall context, largely constituted from the mission definition documents. The scenario defines the overall mission objective and success criteria. Vignettes represent a more narrowly framed subset of the scenario and its purpose is to focus the analysis of blue and red capabilities in an operational environment. Once the details of the vignette are clearly understood, the analysis can be developed, metrics can be created to include Measure of Success (MOS), Measure of Effectiveness (MOE), and Measure of Performance (MOP).

A MOS is a value attributed to mission success. A MOE indicates a measurable attribute or target value for success within the overall mission and MOP indicates performance characteristics of individual systems.[11] These measures are related to each other, as depicted in Figure 5 and Figure 6, where each higher-level metric is a composition of lower-level metrics. Proposed metrics for the geometric SDA detection problem are discussed in the Methodology chapter.

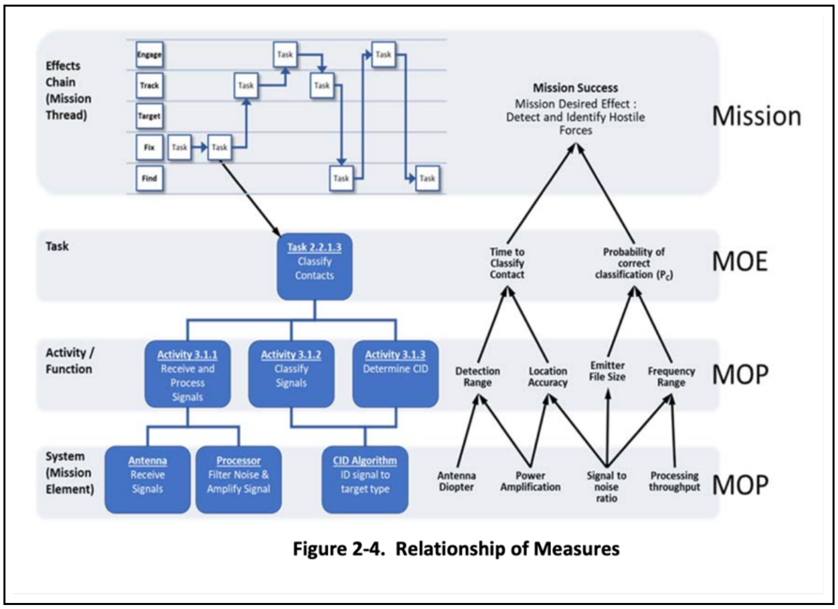

Figure 5: Relationship of Measures [11]

The design of the analysis is the next step of the ME process. ME evaluates the mission by examining the interaction between the operational environment, threats, activities, and systems-in-use. The systems-in-use can be captured in both "as-is" and "to-be" architectures to show connections among major elements of a mission construct. A mission thread comprises the end-to-end tasks or activities needed to accomplish an objective within a scenario or vignette as depicted in Figure 7.[11] Examples of end-to-end mission constructs include kill chains or effects chains.

The mission characterization, metrics, and architectures help identify what data is needed, what assumptions and constraints are needed, and what models are appropriate for analysis.

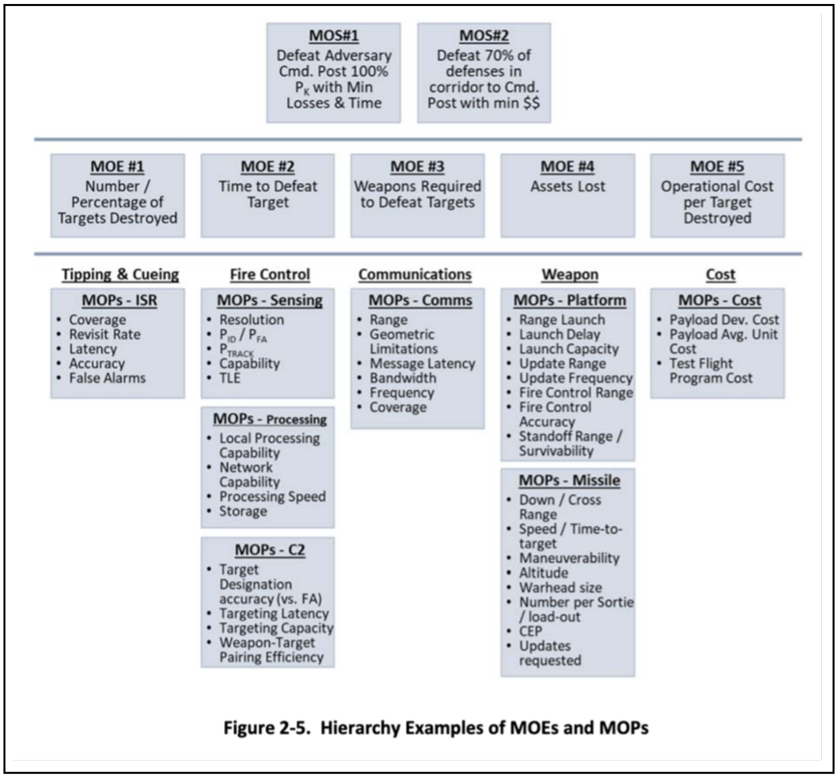

Figure 6: Hierarchy of Measures [11]

The models and series of tasks associated with the Mission Thread (MT) is called a Mission Engineering Thread (MET), which is what will be captured in a DT. The variables for the MET can be tasks, capabilities, technology or systems within a MT. For SDA, the input variables could consist of aperture size, detector characteristics, sensing orbit regimes, target characteristics, and time of year. Then, supporting models, data, and analytics (Figure 8) can be synthesized to perform the analysis depending on the level of fidelity needed and the types of domains or physics involved as discussed in Chapter 3. Another aspect of the analysis process is characterizing the uncertainty involved. The sources of error need to be identified and understood, as well as the effects of error propagation across the system of models. These results are then used to adequately define the confidence level in the mission analysis. Then, a sensitivity analysis can determine the overall influence of each variable on the output and can provide input to optimization routines. These analyses help answer the

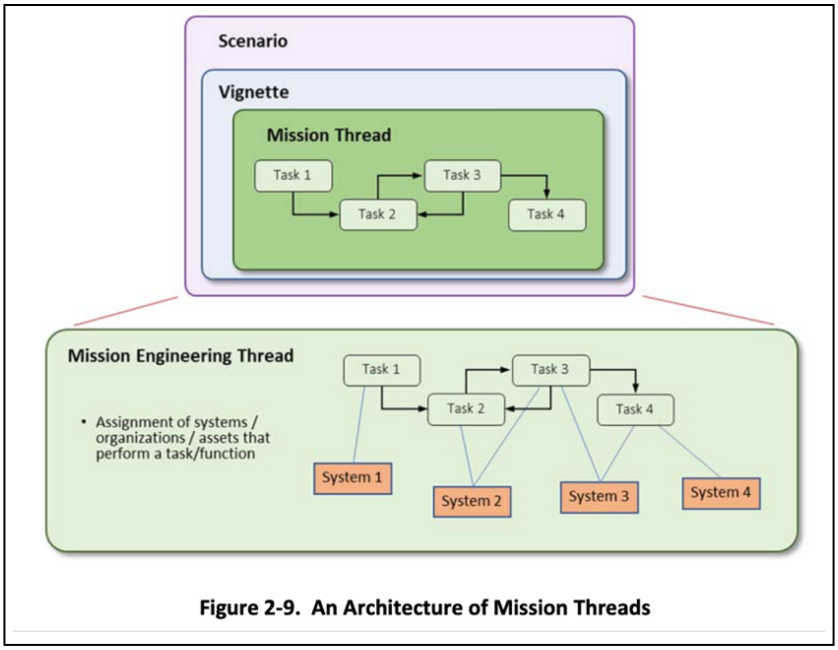

Figure 7: Mission Thread Architecture [11]

question of "how can we better accomplish and execute the mission" using alternatives from DOTmLPFP.

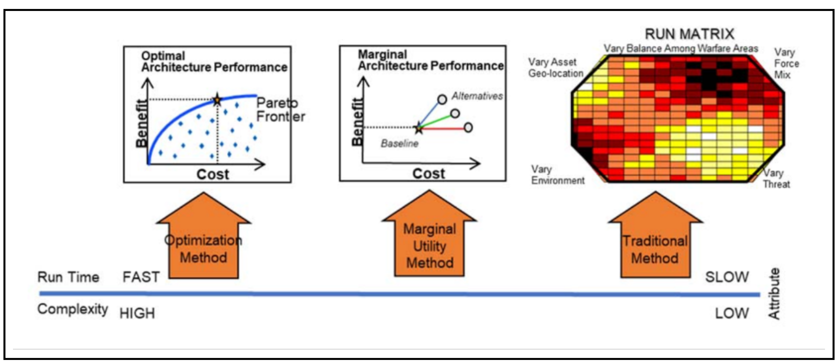

Figure 8: Type of Analysis [11]

Finally, observed trends, implications, relationships, and correlations from the results can be discussed, as it is in Chapter 5. The deliverables include analytical reports, visualizations, government Reference Architectures (RAs), and a preferred mission architecture based on the analysis.[11] Together, these products identify and quantify mission capability gaps and help

focus attention on technological solutions to meet future mission needs, inform requirements, and support capability portfolio management. Though there are multiple functions and systems required to perform SDA, this thesis will focus on the detection function as it is foundational for Orbit Determination (OD).

2.1.1 Orbit Determination

One of the primary operational SDA functions is OD. OD provides requisite knowledge to propagate satellite motion, and therefore understand possible future position and velocity, in order to generate a threat assessment. There are numerous challenges associated with using sensors to detect and initiate OD for an uncooperative target transiting cislunar space.[1] Too low of a Signal-to-Noise Ratio (SNR) value means the sensor cannot detect the target. Range, and a short custody period, cause a too-short-arc issue due to poor sampling of the target's in-track velocity that increases the error associated with the predicted orbit. This error is further exacerbated by observing constellation orbit instability and imprecise position, navigation, and timing due in part to a lack of systems like the Global Positioning System (GPS).[1] To avoid the complications involved with OD techniques, Knister asserts measuring detection and custody performance can be a more abstract way of studying the effectiveness of SDA systems.

Knister developed three performance metrics to evaluate SDA constellation performance to include Mean Detect Time (MDT), Mean Track Time (MTT), and Mean Time Between Tracks (MTBT), which are further defined in the Methodology chapter.[7] These metrics describe the ability of an architecture to detect and maintain custody of an object. MDT, MTT, and MTBT are abstractions of measurements that are more important for cislunar SDA (i.e., angle and range rates), so they must be carefully considered before they are implemented because they form the metrics which ultimately drive the identification of an optimal architecture.

2.1.2 Orbital Dynamics

As previously discussed, range affects the ability of a system to detect an object or perform OD, so it is prudent to consider various orbits or trajectories that could maximize SNR for any given object. There are geocentric, lunacentric, and Lagrange point families of periodic, semi-periodic, and quasi-periodic orbits within the cislunar environment. Each of these orbits offer different locations for observation platforms. The most stable orbits tend to have apoapsis close to the celestial body. Simplified two-body dynamics and an analytical method can be used to model a geocentric orbit with negligible perturbations from other sources. The resulting, simplified Keplerian motion assumes:

1. there are only two bodies,
2. there is an inertial coordinate system centered on the primary body,
3. the mass of the satellite is negligible and treated as a point mass with uniform density,
4. the primary body is treated as a point mass with uniform density,
5. and the only forces acting on the two bodies are from their mutual gravitational attraction.[12]

The two-body Equations of Motion (EOM) can be solved analytically using design variables called Classical Orbital Elements (COEs). There are five, time invariant COEs and one time-varying COE that characterize a geocentric orbit. The time invariant variables include semi-major axis, right-ascension of the ascending node, inclination, argument of perigee, and eccentricity while true anomaly is time varying. This method and dynamics model only works when other forces acting on a satellite are negligible. Once forces start to significantly perturb a satellite's motion, higher-fidelity dynamics are needed to describe motion.

Geocentric orbits with an apoapsis approaching the Moon's sphere-of-influence (SOI) tend to be unstable due to the effects of the Moon's gravity and require active orbital maintenance.

The exceptions are geocentric lunar resonant orbits which have a large apoapsis but remain stable because they leverage symmetry within three-body dynamics. A resonant orbit is any orbit with a period that creates an integer ratio with the period of the Moon; i.e. a satellite in a resonant orbit known as P/2, completes two orbits during a single orbit of the Moon about the Earth and creates a 2:1 ratio between the satellite's and Moon's periods. The benefit of high-altitude, geocentric orbits transiting cislunar space are their ability to increase coverage of the cislunar environment by reducing the distance between the target and the sensor. Additional orbits that increase coverage and can be stable are about the Lagrange points. The Lagrange point orbital families offer low-energy station-keeping locations useful for, among many things, observation.[1] These types of orbits can be designed and studied using various dynamics models and methods.

Cislunar space exacerbates perturbations to satellite motion due to the gravitational influence of other celestial bodies, namely the Moon, because Earth's gravity no longer completely dominates motion. Other perturbations could also result from the elliptical orbit of the Moon about the Earth, gravitational forces from other celestial bodies and solar radiation. Once Keplerian motion can no longer describe or predict a satellite's trajectory, a new method is required. Parker and Anderson describe the dynamics involved within a three-body system (Earth, Moon, satellite).[13] This motion is often described using the Circular Restricted Three-Body Problem (CR3BP) set of EOM. The CR3BP simplifies dynamics by assuming:

1. the satellite mass is infinitesimally small and is therefore considered massless,
2. the only force acting on the satellite are the gravitational forces from the two massive bodies,
3. and the two primary bodies are in circular motion about their barycenter.[13]

The barycenter is located at the center of mass of the system, on the plane which connects

the Earth and Moon, also called the synodic plane. Grebow derived simplified CR3BP equations of motion from Newton's Second Law with respect to the synodic plane, which are defined in Equation 1, Equation 2, and Equation 3.[14]

$$\ddot{x} - 2\dot{y} - x = -\frac{(1-\mu)(x-\mu)}{r_{13}^3} - \frac{\mu(x+(1-\mu))}{r_{23}^3} \tag{1}$$

$$\ddot{y} + 2\dot{x} - y = -\frac{(1-\mu)y}{r_{13}^3} - \frac{\mu y}{r_{23}^3} \tag{2}$$

$$\ddot{z} = -\frac{(1-\mu)z}{r_{13}^3} - \frac{\mu z}{r_{23}^3} \tag{3}$$

These equations of motion are particularly useful for modeling orbits around equilibrium point solutions to the CR3BP.[13] The stationary equilibrium points are known as libration points, or Lagrange points, and there are five for any given three-body problem, as depicted in Figure 9. These points lay on the synodic plane and are locations where an object's velocity and acceleration components theoretically equal zero.

The three types of Lagrange point orbits are planar, axial, and vertical. Planar orbits reside on the synodic plane, axial orbits intersect the x-axis at two points, and vertical orbits emanate from the Lagrange point.[1]. These orbits and trajectories are designed using a shooting method that utilizes a differential corrector. The method starts with an initial state vector, and then propagates the state of the motion through time with the State Transition Matrix (STM). The differential corrector is used in conjunction with the STM to adjust the initial state in order to reach a desired end state.

Several researchers have generated series of orbits using this method, to include Doedel, et al.[15]. Doedel, et al. describe entire families of orbits emanating from Lagrangian points as documented in Figure 10.

Where the families intersect are called bifurcation points. These bifurcation points belong

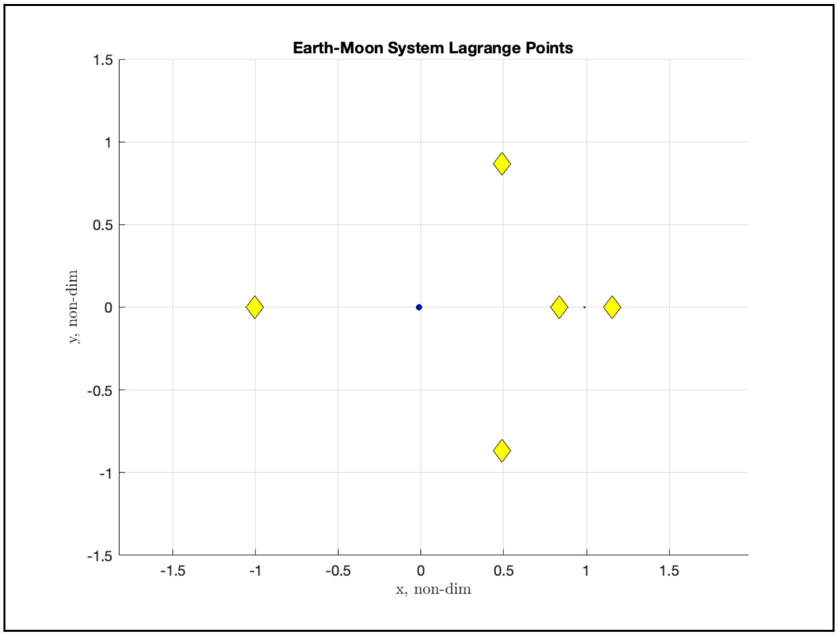

Figure 9: Earth-Moon Lagrange Points

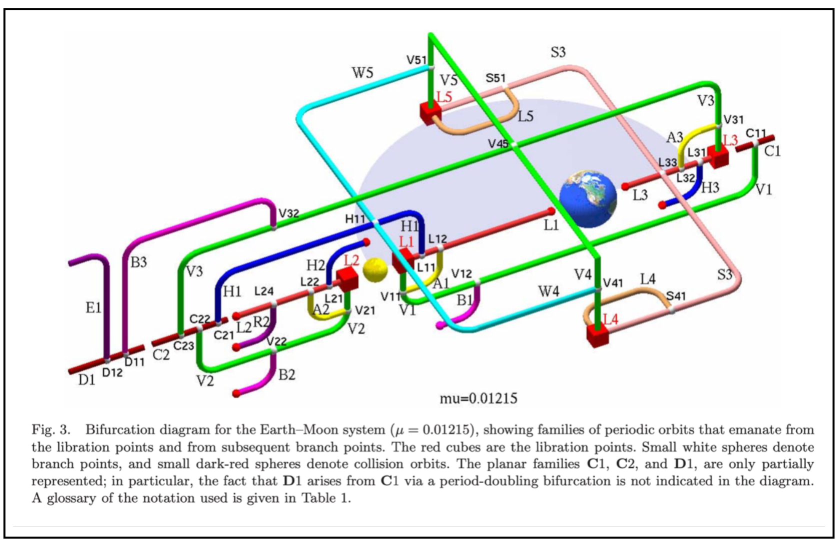

Figure 10: Orbital Families [15]

to at least two families of orbits and can be used to transition from one to another by perturbing an object's motion in a particular direction.

Knister implemented a type of a planar Lagrange point orbit, known as a Lyapunov orbit, as a representative orbit a target could occupy at L_1. The Lyapunov orbit is unique in

precluding perpendicular acceleration to the synodic plane, meaning the target would appear to follow a straight, horizontal line between points if observed from Earth. The Lyapunov orbit is generated using the linearized first order approximation of the CR3BP EOM as originally derived by Grebow.[14]

$$x' = \frac{A_{y'}}{\beta_3} cos(s\tau + \phi) \tag{4}$$

$$y' = -A_{y'} sin(s\tau + \phi) \tag{5}$$

$$z' = A_{z'} sin(\nu\tau + \psi) \tag{6}$$

where $A_{y'}$ and $A_{z'}$ are orbital amplitudes in the y' and z' directions and s and ν are orbital frequencies in the x-y plane and z-axis. For a Lyapunov orbit at L_1, the $A_{z'}$, β_3, s, and ν values are:

	$A_{z'}$	β_3	s	ν
L_1	0	3.5865	2.3344	2.2688

The orbital amplitudes and angular offsets can then be fed into an analytical analysis using the linearized approximation equations to generate an initial position and velocity estimate using a series of patch points. A Lyapunov orbit is symmetric across the x-z plane and is centered about a Lagrange point. The initial state vector for a Lyapunov orbit is:

$$\dot{X}_0 = [x_0, 0, 0, \dot{y_0}, 0, 0] \tag{7}$$

The initial conditions must then be modified by a differential corrector so the subsequent crossing of the x-z plan has a zero velocity in the y-direction, ensuring the orbit is periodic in nature. The design of this type of periodic orbit is only one aspect of designing a mission,

however.

2.1.3 Mission Architectures

An SDA observation architecture needs to provide diverse geometric collection to reduce uncertainty involved with OD.[1] An SDA architecture might be able to accomplish this with a heterogeneous architecture because a target can be detected using multiple passive and active sensing phenomena across multiple spectral bands from multiple trajectories. A couple of materiel concepts capable of fulfilling these functions include radio detection and ranging (RADAR) and Electro-optical (EO) detection and ranging. Interested in creating a heterogeneous architecture, Llavaria, et al. implemented the architecting method captured in Figure 11.

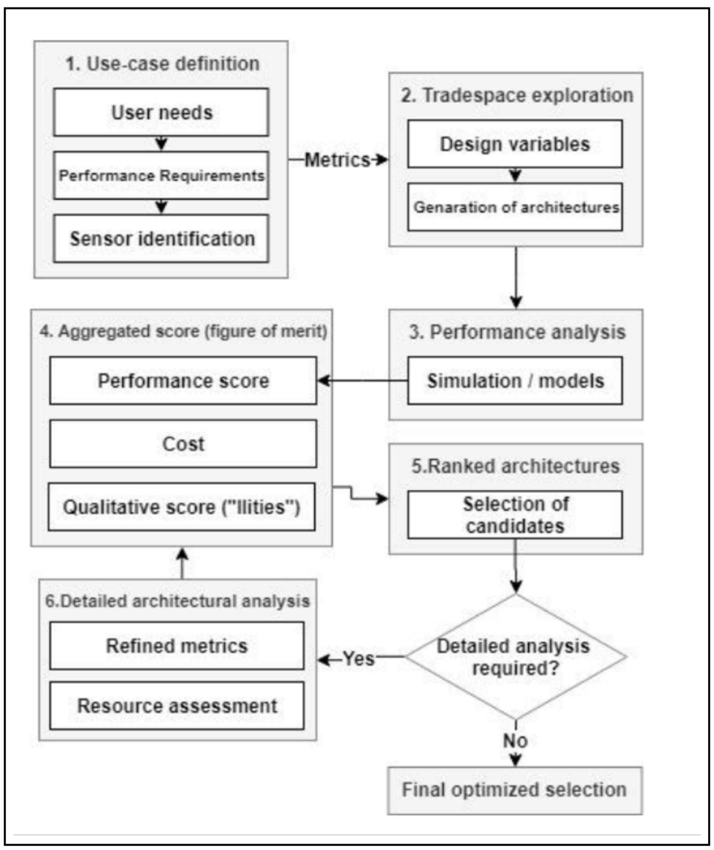

Figure 11: Architecture Process [16]

The use-case definition block in Figure 11 conducts stakeholder analysis and requirement derivation to ultimately define performance metrics. Of interest, this study included sensor definition early on to constrain the search space, adding a bottoms-up engineering approach to the top-down engineering approach associated with high-level architecting. This implementation reduces analytical complexity and may produce more realistic constellation alternatives. However, there is risk the solution space may be overly constrained to today's technology or assumptions. The next block, Tradespace exploration, determines and selects variables that will affect constellation concepts and generates alternative architectures combinatorically, what other texts describe as a process of architecture fragmentation and integrated concept creation [17]. These architectures are then modeled and simulated in the Performance analysis block. The Aggregated score block calculates performance and qualitative scores, along with parametric cost values. Metrics for these integrated constellation concepts are then combined into a weighted, normalized aggregate score in order to rank the architectures. The next step identifies whether more detailed analyses are required and, if so, what refinements are needed for the analyses. If there is not a need for additional analysis, then the method produces an optimal, heterogeneous constellation. Of note, Llavaria, et al. did not accomplish sensitivity analyses on the variables selected to determine which are more impactful and therefore require more deliberate decision-making. This analysis yielded many concepts of federated or fractionated satellite systems to meet SDA requirements. Generating pareto curves based on concept cost and individual performance metrics will yield additional understanding of which concepts perform most optimally for each requirement, providing additional understanding of the problem/solution space.

Knister implemented a similar process.[7] Knister proposed architecture fitness metrics and developed a corresponding model to evaluate several simple satellite constellations.[7] Knister evaluated and compared the satellite constellations presented in Figure 12 against a target in a Lyapunov orbit at $L_1$to determine the validity of the metrics proposed in the

OD section and determine the best alternative architecture.

		GEO	LEO	Ground
Reference Comparisons (number of sensors)				
1	Stern & Wachtel (Optimized)	4	4	8
2	GEO and LEO (both polar/equatorial)	4/4	4/4	NA
3	GEO (polar/equatorial)	4/4	NA	NA
4	GEO (polar)	4	NA	NA
5	GEO (equatorial)	4	NA	NA
6	GEO (synodic)	4	NA	NA
7	LEO (polar/equatorial)	NA	4/4	NA
8	LEO (polar)	NA	4	NA
9	LEO (equatorial)	NA	4	NA
10	LEO (synodic)	NA	4	NA

Figure 12: Satellite Constellation Trials[7]

It was helpful to anchor the proposed architectures' performance against a more conventional baseline. Stern and Wachtel's optimized terrestrial and space-based architecture described in trial one of Figure 12 formed the normalized baseline, where every weighted-sum objective function's final value was one.[18] Each metric for every proposed architecture was normalized to the corresponding metric value for the Stern and Wachtel architecture prior to calculating the overall metric. More optimal architectures have a value of less than one, and less optimal architectures have a value greater than one. Figure 13 provides an overview of the trial comparison between the Stern and Wachtel baselines and the other proposed space-based architectures.

Overall, low Earth orbit (LEO) constellations ranked higher than geosynchronous Earth orbit (GEO) constellations and combined LEO/GEO constellations. However, the GEO constellations had lower MDT and MTBT scores due to their orbit period. Some of the

		1		2		3		4		5		7		8		9	
		S&W		GEO and LEO		GEO Only		GEO (equ)		GEO (polar)		LEO Only		LEO (equ)		LEO (polar)	
Scenario 1	MDT	7.97%	1.000	7.97%	0.994	0.62%	9.334	0.55%	11.761	0.07%	27.583	12.75%	1.112	7.44%	0.780	5.92%	0.794
	MTT	2.63		2.63		10.25		12.11		4.67		2.90		2.47		2.68	
	MTBT	30.36		30.36		1510.38		1964.90		4936.00		19.79		30.74		42.54	
	Cost ($M)	1990		1952		2192		1096		1096		1712		856		856	
Scenario 2	MDT	3.18%	1.000	3.18%	0.994	0.00%	NA	0.00%	NA	0.00%	NA	3.18%	0.953	3.18%	0.810	0.00%	NA
	MTT	2.51		2.51								2.51		2.51			
	MTBT	75.91		75.91		19758.00		19758.00		19758.00		75.91		75.91		19758.00	
	Cost ($M)	1990		1952		2192		1096		1096		1712		856		856	
Scenario 3	MDT	6.72%	1.000	6.72%	0.994	0.00%	NA	0.00%	NA	0.00%	NA	12.78%	1.256	6.72%	0.810	7.11%	0.877
	MTT	2.53		2.53								3.41		2.53		2.99	
	MTBT	35.10		35.10		19758.00		19758.00		19758.00		23.26		35.10		38.97	
	Cost ($M)	1990		1952		2192		1096		1096		1712		856		856	
Scenario 4	MDT	8.34%	1.000	8.34%	0.994	0.18%	1.757	0.15%	4.061	0.03%	41.287	9.11%	0.987	8.20%	0.805	1.60%	1.127
	MTT	2.44		2.44		3.60		3.75		3.00		2.58		2.42		2.06	
	MTBT	26.83		26.83		1792.91		2192.00		6584.00		25.73		27.07		125.43	
	Cost ($M)	1990		1952		2192		1096		1096		1712		856		856	

Figure 13: Comparison [7]

GEO constellations did not detect the target at all for the duration of the scenario due to geometric implications. While LEO constellations performed better, the satellites only detected the target, at best, 7.11% of the time. The analysis indicates traditional orbital regimes and ground-based systems cannot provide consistent custody of a cislunar target. The analysis suggests observers must be placed in non-traditional locations (e.g., high semi-major axis Earth orbits, Lagrange points, the lunar surface) to improve performance.[7]

Knister characterized how traditional geocentric, LEO and GEO EO constellations performed over time. These types of constellations are challenged to keep SNR values above the threshold detection level, as depicted in Figure 14. The SNR for each observer drops below the threshold value for several days over the course of the synodic period as the solar phase angle, as opposed to range variations, increases to a maximum value of 180 degrees. The reflection from the target reaches a minimum at the maximum solar phase angle, therefore decreasing the signal. Range variations drive more minor SNR oscillations compared to solar phase angle and are more pronounced for geosynchronous satellites.[7]

Knister's analysis provided conclusive evidence that solar phase and lunar exclusion angles limit the ability of geocentric, space-based architectures in performing cislunar SDA. Kinister also found lunar exclusion angles render ground-based sensors useless, though the effects are mitigated for space-based sensors because the Earth, without an implemented

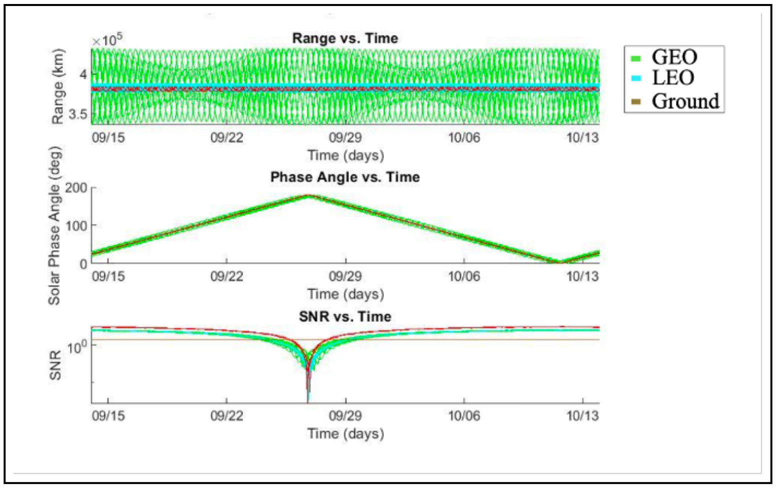

Figure 14: SNR through Time[7]

Earth-exclusion angle or consideration of Earth albedo, blocks the Moon from the sensor's Field-of-view (FOV) for a small portion of time when the object is in view.[7]

Bolden, et al. accomplished a similar architecting study for cislunar SDA where they considered different network distributions, sensing payloads, CONOPs, and OD algorithms to develop an optimal architecture.[1] Bolden, et al. modeled and evaluated different cued and un-cued CONOPs and architectures, as seen in Figure 15, and conducted domain coverage, line-of-sight to the target, and object custody analyses.[1]

Bolden, et al. discovered that LEO constellations provided more numerous revisits but had less coverage due to the limited geometric diversity LEO offers as depicted in Figure 16.[1] Additionally, large solar phase angles and long ranges hampered the regime's detection performance for targets between the Earth and Sun.

The conclusions from coverage and line-of-sight analyses, depicted in Figure 17, were then used to design and iteratively improve architecture detection performance. Bolden, et al. found a heterogeneous, multi-orbit type architecture provided the best coverage and revisit rates, as seen in Figure 17, and would likely perform the best against a hypothetical cislunar target.

Table 2: Constellation Orbit Type Scaling

Constellation Architectures	Number of Spacecraft per Orbit Type				
	LEO	GEO	Lagrange	Lunar Resonant P/3	Chaotic
LEO Only	75				
GEO Only		24			
Lagrange Only			L1, L2, L3, L4, L5		
P/3 Only				24	
Chaotic Only					24
P/3 & Lagrange			L1, L2, L3, L4, L5	24	
LEO & Lagrange	75		L1, L2, L3, L4, L5		
GEO & Lagrange		24	L1, L2, L3, L4, L5		
GEO & 6x Lagrange		24	6 x [L1, L2, L3, L4, L5]		
GEO, Lagrange, & P/3		24	L1, L2, L3, L4, L5	24	
GEO, 6x Lagrange, & P/3		24	6 x [L1, L2, L3, L4, L5]	24	
GEO, Lagrange, & Chaotic		24	L1, L2, L3, L4, L5		24
GEO, 6x Lagrange, & Chaotic		24	6 x [L1, L2, L3, L4, L5]		24

Figure 15: Bolden, et al. Trials [1]

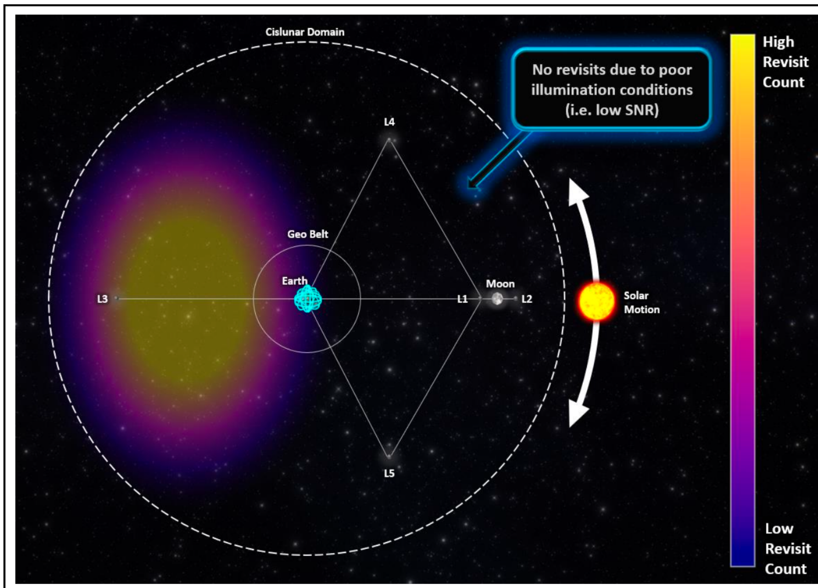

Figure 16: Illustration of Constellation with Insufficient Coverage [1]

Bolden, et al. investigated each orbital type's SNR performance against a target in a circumlunar free return trajectory.[1]. Bolden, et al. found LEO and GEO orbits performed poorly, with SNR values below an assumed threshold value of six, for the majority of the trajectory as seen in Figure 18.

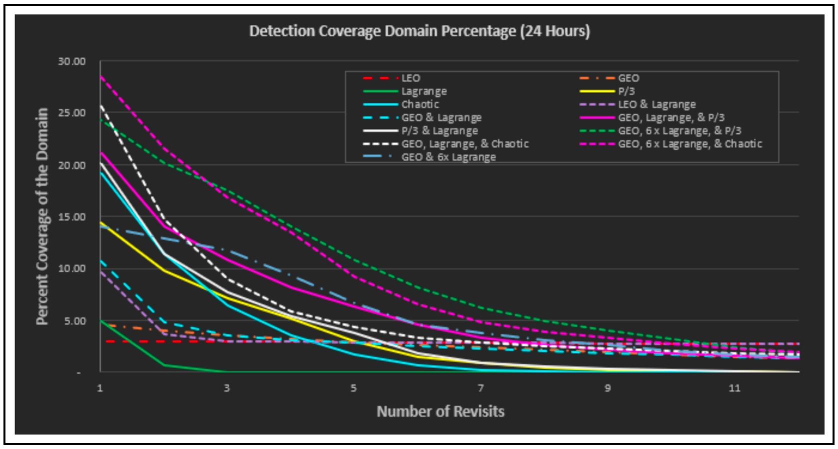

Figure 17: Detection Coverage Domain Percentage Architecture Comparison[1]

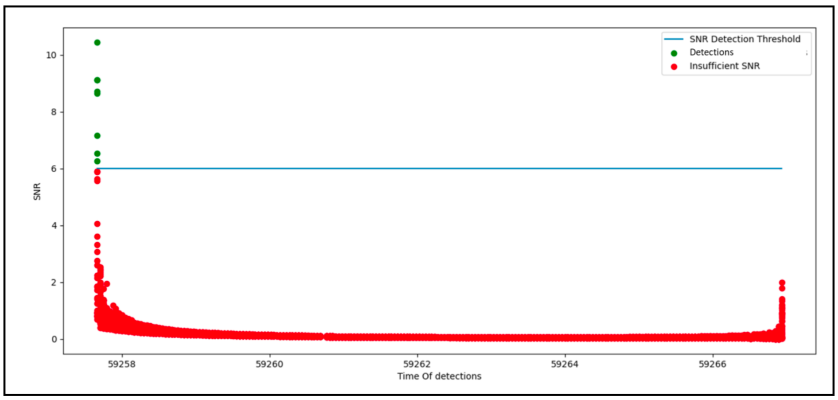

Figure 18: Example LEO Cued Custody Plot for Waxing Crescent Free Return Trajectory [1]

Lagrange point orbits were able to augment the LEO and GEO sensors by detecting the target when the LEO and GEO sensors were not, as depicted in Figure 19 and contrasted with Figure 18. However, there were approximately three days of detection gaps during the entire flight that needed to be addressed.[1]

Lunar resonant P/3 orbits provided higher coverage due to their altitude but less revisit opportunities. For this scenario, the P/3 orbit provided additional detections early in the flight, and also provided a second sensor detection event along with the sensor at L_5, but did not close the custody gaps. Chaotic orbits provided additional detections during the

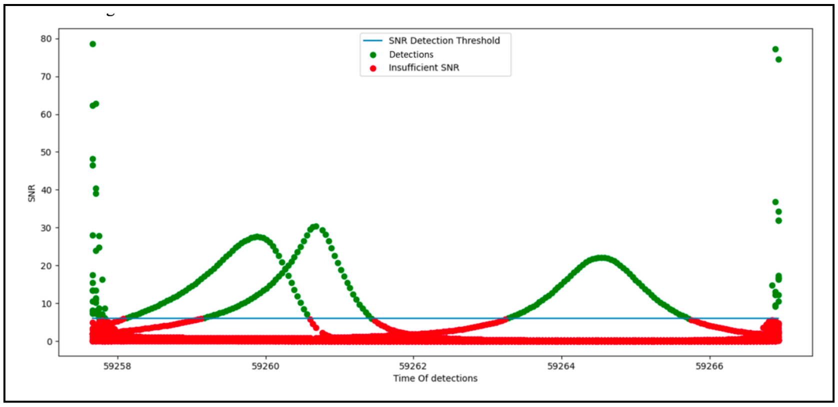

Figure 19: Example Cued Custody Plot for Waxing Crescent Free Return Trajectory [1]

initial and final days of the flight but a one day detection gap persisted in the middle of the flight. None of the architectures were able to maintain persistent coverage of a target in a circumlunar free return trajectory with assumed 42 cm optical apertures. The most effective architecture Bolden, et. al. found was a combination of satellites in GEO, Lagrange point, and chaotic orbits. This type of architecture would also be geometrically diverse and provide additional information to reduce uncertainties that could arise from near identical observations at LEO. However, Lagrange point and chaotic orbits present significant challenges for orbit stability and Position, Navigation, and Timing (PNT). Bolden, et. al. surmise another concept will need to be implemented to close this detection gap, whether it is another orbit type, sensing phenomenology, or a highly maneuverable satellite that could maintain a close range.

As corroborated by Knister, Bolden, et al. found that sensors too far away from the target experience low SNR values due to range loss, as well as encounter too-short-arc issues due to slow apparent angular motion.[1] These challenges suggest considering a set of space-based sensors distributed throughout cislunar space to reduce distances from sensor to target as well as implementing geometric diversity to solve the too-short-arc problem. Bolden, et al.'s architecture study revealed the best architecture to observe trajectories, like the circumlunar free return trajectory, would provide significant range and solar phase angle diversity.[1]

Observers placed in unstable Lagrange point and chaotic orbits significantly outperformed sensors in LEO and GEO orbits and, ultimately, a multi-orbit, multi-concept constellation will likely be necessary for continuously detecting targets on a cislunar trajectory.[1] Rosenof also discovered observers placed in cislunar orbits performed better against objects in cislunar trajectories than the same observers based in LEO.[19] These lessons learned provide heuristics that can be used to narrow the architecture study to those orbits most likely to satisfy cislunar SDA requirements. While the architecting process is well suited for novel problem sets like cislunar SDA, it is also directly applicable to developing and implementing digital engineering toolchains.

2.2 Digital Engineering

DE addresses challenges of communicating user needs and system design by depicting and simulating dynamic user scenarios and concepts with semantically similar and traceable requirements.[20] A primary goal of DE should be to document knowledge in a human-readable, machine-interpretable and executable language within an integrated model representing an enduring, authoritative source of truth, what McDermott, et al. calls "owning the technical baseline".[20, 21] A DE framework uses models in the virtual world to facilitate rapid, iterative analyses. Ideally, these depictions and systems would reside in one integrated environment which would be used to understand a system model throughout the life-cycle.

The guiding principles for the Digital Enterprise are:

1. Model mission with threat information to enable mission level analysis.
2. Clearly understand the performance expectations and key system parameters.
3. Collect and control data to enable enterprise sharing.
4. Use common modeling languages to enable enterprise design.
5. Maximize the use of RAs and standards to accelerate solution development.

6. Leverage cloud capabilities to offload organic IT requirements and enable enterprise use.

7. Use digital tools to accelerate processes.[22]

The hypothesized benefits from implementing DE using DTs, Digital twins, and authoritative sources of truth include reduced technical and programmatic risks, as well as enhanced decision making.

2.2.1 Model-based Systems Engineering

Primary SE activities, namely stakeholder and requirement analysis, system design, and verification and validation, can be implemented with document-based and model-based approaches. However, document-based practices struggle to communicate complex systems, adapt to change, and are prone to human-errors, inconsistencies, and inaccurate representations. MBSE improves the effectiveness and efficiency of systems engineering practices, compared to document-based practices, by providing harmonized perspectives through a comprehensive, integrated set of models known as an authoritative source of truth. Friedenthal described five clear advantages for MBSE:

1. Precision of the system design, which reduces downstream design errors.

2. Efficient traceability from system requirements to verification, which enhances system integrity.

3. Flexibility to evolve system specification and designs.

4. Reuse across projects and utilization of RAs and reference models.

5. Shared source of information on the system to reduce miscommunication.[23]

MBSE is defined by INCOSE as "the formalized application of modeling to support system requirements, design, analysis" using a semantic modeling language, a modeling

method, and a modeling tool.[23] MBSE provides the technical process associated with system design and describes an integrated system's elements and relationships in a model-based context.[24] MBSE uses a standard modeling language and a modeling tool to perform a related set of processes and methods as prescribed by a modeling methodology. The primary artifact of MBSE is a descriptive model that specifies "what the system is, what it does, and how it does it".[23]. The two types of descriptive models are spatial and logical models. A spatial model, such as a three-dimensional Computer Aided Design (CAD), represents geometric relationships. A logical model describes functional, connectivity, and traceability relationships and dependencies, such as circuit design models.

A system model is a type of logical model and captures requirements, structure, behavior, and parametric constraints of a system and its environment. The system model further depicts and describes relevant system elements and relationships. These elements and relationships define a physical system and the system's functions to clearly communicate system design through many views.[23] A common MBSE method is required for a design team to ensure the team is building the model consistently, with common breadth, depth, and fidelity.[24] Selecting an MBSE method occurs in context of the problem-solution set and the skills of the team. Estefan conducted a survey to identify multiple MBSE available methods.[25]

2.2.2 Object-Oriented System Engineering Method

One MBSE methodology is Object-Oriented Systems Engineering Method (OOSEM). OOSEM integrates a top-down, scenario-driven, model-based SE approach with object-oriented concepts with a goal to architect more flexible and extensible systems that can accommodate changes, a goal shared with the agile methodology.[25, 23] OOSEM uses a SE process to focus on realizing systems at the differing levels of solution hierarchy which includes the following activities:

1. organize model,
2. analyze stakeholder needs,
3. analyze system requirements,
4. define logical architecture,
5. synthesize candidate physical architectures,
6. optimize and evaluate alternatives,
7. and manage requirements traceability.[23]

OOSEM utilizes Systems Modeling Language (SysML), a profile of the Unified Modeling Language, to support graphic specification, design, analysis, and verification of systems through standardized semantics representing requirements, behavior, structure, and properties of systems.[23] The architecture defines the operations, systems, and data flow within a mission and is efficiently modeled using a Block Definition Diagram (BDD). A block is used to represent a system or a component of the system, and any external systems. The block defines the values, operations, and classifications associated with that system. The BDD is used to show the relationship between different blocks, most notably, with a composition relationship.[23] General models, known as RAs, can be created with these elements to accelerate the production of more specific models.

2.2.3 Reference Architectures

The advantages of MBSE can be realized by capturing prior knowledge in a RA to provide a starting point for a particular type of system design. A RA aids in reducing ambiguity and managing complexity.[23] A RA can accelerate development timelines by focusing, guiding, and constraining downstream processes with prior knowledge.[17] The Office of the Assistant Secretary of Defense Networks and Information Integration describes architectures as “an

authoritative source of information about a specific subject area that guides and constrains the instantiations of multiple architectures and solutions".[26]

The RA should contain strategic purpose, principles, technical positions, patterns, and vocabulary. The purpose has goals, objectives, and a problem to address. Principles are high level statements that drive technical positions and patterns. Technical positions are technical guidance and standards that must be followed by solution architectures, such as interoperability standards. Patterns are representations that show relationships between elements and the vocabulary's acronyms, terms, and definitions communicate those relationships.[27] The Department of Defense (DoD) manages architectures with the Department of Defense Architectural Framework (DoDAF) by categorizing the different viewpoints an architect could create to depict different perspectives of the problem/solution set.[28]

These RAs typically contain source document derived requirements, structures, behaviors, and parametrics associated with the mission and/or system(s) of interest. Kelly, and others, have developed RAs for different contexts and at varying levels of solution development.[29] These RAs are created using a modeling language, method, and tool.[24] Kelly's cubesat RA is based on the Air Force Institute of Technology (AFIT)'s MBSE processes and conventions and was developed in Cameo Systems Modeler (CSM) using SysML and OOSEM. The RA contains source documentation, SE processes and artifacts, a library of generic parts and value properties, modifiable analysis patterns, and document generators to save time and improve satellite model quality for particular instances. The SE processes included in the RA are stakeholder analysis, requirement derivation, traces between requirements and allocated functional and physical solutions, trade studies, and verification and validation analyses of physical components for a particular cubesat instance. The structure is populated down to the component level of a cubesat with minimal modeling of external systems to the spacecraft.

In general, Kelly found the cubesat RA accelerated the creation of a cubesat instance

for the AFIT because it already contained prior knowledge of source documentation, model management, and the necessary model elements to describe a cubesat. However, Kelly did find the models and diagrams were fragile and that modifying an element or diagram in one area might require refactoring several other models or diagrams.[29] While this type of descriptive model is effective for human-interpretation, it is not an analytical model. Analytical models are typically mathematical and can be used to assess performance, and if simulated, assess performance as a function of time.

2.2.4 Simulation and Analysis

An analytical model represents the system, or environment, in terms of mathematical equations that describe a parametric relationships. Modeling the system and underlying phenomena can enable assessments or predictions of system performance or emerging attributes. Models can describe orbital dynamics, electromagnetic energy propagation, or even mass properties. Some analytical models can be solved via closed-form solutions, like the two-body dynamical EOM. Other models require numerical integration to determine the change in a state.[23] Parameter values describing these models can also be deterministic or probabilistic, introducing uncertainty to the analysis. Each of these considerations affect the fidelity of each analytical model, which means each model must be built based on the purpose of the model and the desired accuracy of the model. Propagating these mathematical models through time creates a simulation.

A simulation enables the analysis of complex interactions and system behavior through time. Bolden, Knister, and Brown all used simulation to propagate orbital dynamics and then analyze performance of various concepts. Each of these studies aimed to better understand the interactions of their concepts with the environment. For Bolden, et al.'s study, multi-body orbital dynamics were simulated and analyzed the performance of multiple spacecraft simultaneously using the open-source General Mission Analysis Tool and System Toolkit

(STK). Bolden, et al. were able to initiate a simulation during a waxing crescent phase of the Moon's orbit and then propagate for up to two months.[1] The information generated from this simulation allowed the researchers to develop trends and understand how different aspects of the environment affected the SNR performance of proposed sensing architectures. Similarly, Knister utilized an integrated MATLAB and STK toolchain to conduct simulation and analysis of simple LEO and GEO constellations performing cislunar SDA functions as seen in Figure 20.

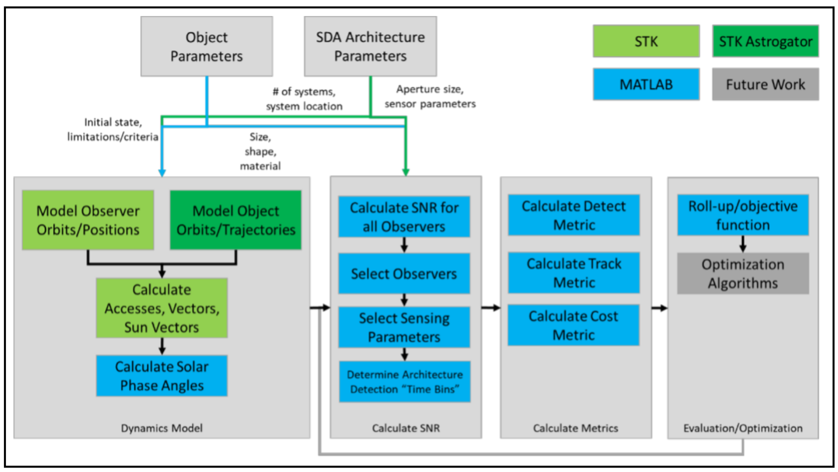

Figure 20: Data Flow [7]

Knister used STK to propagate three-body orbital dynamics of the target's L1 Lyapunov orbit and calculated constrained access times, due to Earth occlusion and sensor exclusion angles, and position vectors between objects and celestial bodies. This information, along with target parameters and sensor parameters, was sent automatically to an analytical model in MATLAB to calculate SNRs for each sensor at each time step, as well as each of the fitness metrics characterizing a sensors ability to detect and track the target. These metrics were rolled, along with cost estimates, into a weighted-sum function to rank each of the proposed sensor constellations based on their ability to detect and track the target.[7] The digital toolchain enabled Knister to accelerate and automate the production of information needed to characterize the performance of concept LEO and GEO based satellite constellations.

However, Knister's digital toolchain did not incorporate a descriptive modeling digital tool to enable the automatic verification of descriptive system requirements using a system model.

2.2.5 Digital Threads

As Noguchi states, an "effective MBSE implementation will harmonize descriptive and analytical models, ensuring the information exchanges rely on authoritative data and are automated to achieve greater efficiency."[30] DTs, also known as digital toolchains, combine purpose-built tools to create an "integrated digital approach that uses authoritative sources of system data and models as a continuum across disciplines to support life cycle activities from concept through disposal."[5] The models and series of tasks associated with a MT is called a MET, which will be implemented via a DT. The DT provides a conducive environment and infrastructure for the work performed with ME and OOSEM processes. Friedenthal describes an approach and key considerations for integrating a SysML-based tool with other models and tools in a systems development environment, to include the different types of tools, their relationships, logical interfaces between tools, configuration management concepts, data exchange approaches, and criteria for selecting a SysML tool.[23] The system model is the authoritative source of the "system specification, design, analysis, and verification information", and maintains traceability and rationale for key decisions.[23] The system model relates the text requirements to the system design which can then feed multi-disciplinary analysis as depicted in Figure 21.

A model is a representation of a concept in the physical world. Models can represent systems and the environment with varying degrees of fidelity, as previously discussed in the system development and dynamics sections. DTs help manage complexity, increase efficiency, enable repeatability, and accelerate understanding by automating the interactions of these descriptive and analytical models. Several SDA studies have implemented rudimentary DTs to achieve these benefits, though none of them reference their automated sequencing of digital

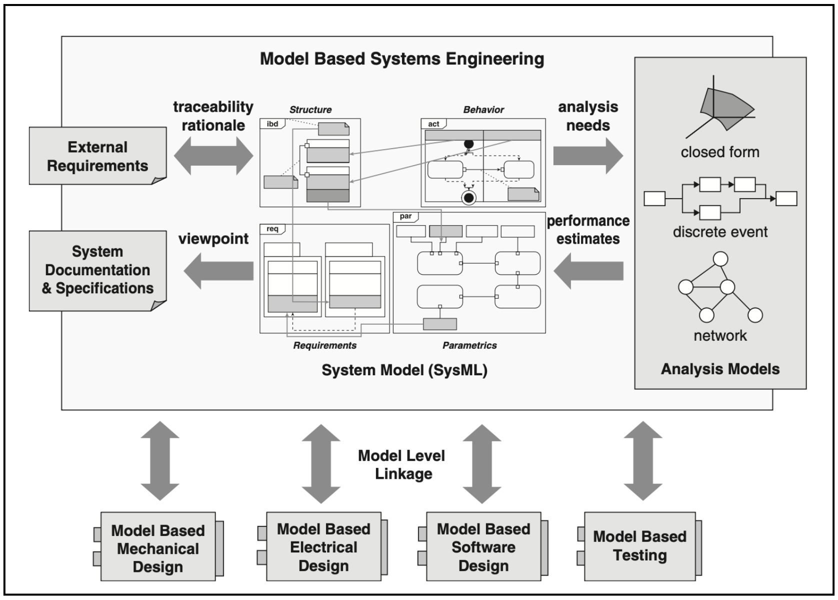

Figure 21: System Model as a Framework for Analysis and Traceability [23]

tools as a DT.

Brown's research provided the next iteration of a digital toolchain that integrated a digital modeling tool with analytical and physics-based tools. Brown focused on investigating a payload's performance during a mission by modeling the system descriptively and then simulating the system in a proposed mission using simulation software.[31] The resulting Payload Analysis Tool (PAT) could analyze up to four payloads on a single cubesat bus to understand power and data rates, as well as fault modes during an imaging mission scenario with one ground station and two targets.[31] Brown incorporated previously developed behavior models captured in Matlab's Stateflow and Simulink tools instead of behavior models in Cameo. A proof-of-concept state machine was developed in Cameo but revealed the fragility of the DT architecture as the Stateflow and Simulink diagrams would have required a large refactoring effort Brown did not undertake.[31] This fragility was exacerbated due to the difficulty of Cameo providing limited information for debugging errors. This resulted in a parametric diagram with only one constraint function, instead of multiple constraint functions, which were then executed in a comprehensive Matlab script because it was easier

to debug. However, the resulting PAT demonstrated the value of adding a descriptive model to a digital toolchain as it enabled the verification and validation of payload design. This was completed through the descriptive tool's traceability function between parametric values and text based requirements.

An additional issue with the PAT was that it was a static analysis. Even though the PAT used a simulation tool, the simulation would run in its entirety and then the data would be provided to Stateflow and Simulink to understand the performance of the Payload as it *would have performed* rather than how it *was performing*. Though this design decision did not prevent Brown from accomplishing the research objectives, it would be an issue if the behavior of a system required a change in the mission scenario. A dynamic, synchronous DT would be preferred so that at every time step, data is passed from the simulation to the behavior models to assess state changes that could be passed back to the simulation.

Brown's research highlights the challenges associated with designing and implementing a DT.[31] Legacy digital engineering tools have historically been disconnected, leading to incompatibilities, increased error, and long processing times. Models tend to be tightly coupled to the simulation tool to address one aspect of understanding dynamic performance and behavior, but the majority of digital engineering tools are not inherently designed for integration as described in Brown's research.[20, 31] Integrations are also largely tool-to-tool versus tool-to-formal knowledge and require significant time for integration as described by Brown and Knister.[20, 7, 31] Additionally, these tools embed different formal languages which drives compatibility issues. The vision of DE will likely be realized when knowledge experts can graphically describe and simulate a concept without the need to develop scripting interfaces.[20] Ideally, the DT would be modular enough to "drag and drop" or "plug and play" alternatives to assess their compatibility, interoperability, and performance at the mission, system, and subsystem levels. Though challenges remain, there are benefits to gain by implementing a DT.

DTs efficiently performed Brown's, Bolden's, and Knister's methodology in hours instead of days, proving the benefit of automating simulation and analysis processes with DTs in the future.[7] Efficiency can be further improved by implementing parallel computing during the evaluation process, as the study by Colombi, et al. demonstrated.[18] Colombi, et al. used Python and STK in coordination with high performance computing to accomplish an architecting study for GEO space situational awareness. Their study included 28 design variables for a large-scale, multi-domain, multi-orbit, disaggregated architecture design. Additionally, they used a genetic algorithm to search for a global, optimal architecture. The genetic algorithm produced 320,000 architectures with their multi-variable, multi-objective optimization problem set, but parallel processing accomplished 27 years of Central Processing Unit (CPU) time in three days.[18]

2.2.6 Integrated Development Environment

A DT is built in an Integrated Development Environment (IDE). An IDE refers to computer-based, interoperable, and networked digital tools and repositories that enable collaborative engineering and automated, executable workflows that can include physics-based simulation, analysis, and mission scenario visualization.[32, 23] The strategy for the IDE is to provide an affordable, stable environment where digital tools can model threats and solution concepts, simulate dynamics, and analyze concept performance via data transfer. Each type of tool should enable the functionality needed for the different parts of the system life-cycle process, such as requirements, configuration, and project management or system design, simulation, analysis, and verification. A SoS modeling tool may include support for architecture frameworks. A system modeling tool supports the development of the system model. Simulation and analysis tools support trade studies, sensitivity analysis, optimization, characterization and prediction.

This research shows how digital modeling, simulation, and analysis, contained in a DT,

can accelerate the creation of ME and SE technical information. However, these technical methodologies do not describe or prescribe how research questions should be prioritized. Cislunar SDA is a complex, complicated, and evolving problem space that needs to be managed with an operator focus. A proper management methodology is also needed to ensure the most value is being created for the operator. Agile development is a paradigm shift, one that has been gaining traction across industry for the past twenty years, whose principles and practices focus on the customer and the information or products most important to them.

2.3 Agile Development

Agile principles were created to develop software better by preferring individual interactions, working software, customer collaboration, and responsiveness over processes, tools, documentation, contracts, and following a predefined plan, though the later still holds value and must be completed as enablers for the project.[33] The twelve principles of Agile development are:

- The highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity – the art of maximizing the amount of work not done – is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.[33]

Several methodologies have been created in pursuit of the Agile principles to include Scrum, Lean, Kanban, Crystal, Extreme Programming, Feature Driven Development, and Dynamic System Development.[34]

2.3.1 Scrum

Scrum helps people generate value through the use of a lightweight, empirical framework that includes accountability, events, artifacts, and rules that bind them together. The Product Owner (PO) continuously orders the work of a complex problem into a Product Backlog. The Scrum Team turns a Sprint Backlog, selected during Sprint Planning, into an Increment of value during a Sprint using Daily Scrum. The Scrum Team and its stakeholders inspect the results during a Sprint Review and adjust for the next Sprint during the Sprint Retrospective.[35] The Scrum Master fosters an environment for the Scrum Team to perform their work based on shared values and principles. Together, these roles focus on creating a usable, concrete step towards the Product Goal, called an Increment.

An Increment is created once the work completed during the Sprint satisfies the Definition of Done. The Definition of Done is a formal quality description for the product. Each Increment builds on previous increments to provide thoroughly verified, additive value and the sum of the Increments is presented at the Sprint Review.[35] While Increments, or working systems, are the primary measure of progress, there are multiple practices for forecasting progress, like burn-downs, burn-ups, or cumulative flows, in addition to Scrum Agile metrics and even SE leading indicators.[36]

Of interest are the SE leading indicators for complex mission and system design. These leading indicators might provide additional transparency into the maturation of the mission and system design when used in conjunction with Agile metrics. An additional consideration and vision of this thesis is the scaled implementation of Scrum for a team-of-teams environment working on a system-of-systems design. Due to the short duration of this thesis, only the basic Scrum framework will be leveraged to manage this research, but the Scaled Agile Framework (SAFe) may be introduced as appropriate during larger scale concept design because it has proven results of improving engagement, time-to-deploy, productivity, and quality.[37]

2.4 Conclusion

There are many operational questions associated with maintaining attribution of uncooperative cislunar targets. Previous research has focused on the geometric target detection problem and has demonstrated that near-Earth sensing constellations lack the ability to maintain custody of cislunar targets. Detection performance, particularly for a circumlunar free-return trajectory, can be enhanced by placing passive sensing platforms in orbits about the Lagrange points. The ME process is useful for capturing mission information and deriving SoS architectures. Previous research demonstrated how the MBSE Object-Oriented Systems Engineering (OOSE) methodology can be tailored and can aid in research by de-

scribing the problem/solution pair from the mission level to the component level. Previous research also demonstrated how digital simulation and analysis can accelerate the creation of high-confidence information in a repeatable manner and how the tools associated with those functions can be integrated with a descriptive tool to form a Digital Thread. The ME, DE, SE, and Scrum methodologies will be synthesized in Chapter 3, the Methodology.

III. Methodology

For this research, multiple principles, practices, methods, and processes are synthesized into an Agile Digital Mission Engineering (DME) methodology to pursue research questions pertaining to the design and evaluation of a Space Domain Awareness (SDA) mission concept's ability to detect uncooperative targets transiting cislunar space. The Agile DME methodology developed in this chapter is implemented to meet the research objectives to deliver verified value quickly, with discipline, while scaling to meet the vision of a complete SDA mission and system Reference Architecture (RA)s. This research implements recursive, time-boxed Sprints to enable flexibility to pursue adjacent research questions and incorporates Mission Engineering (ME) and Systems Engineering (SE) technical activities to move from problem definition to concept design in a model-based context. In particular, this research utilizes the Object-Oriented Systems Engineering Method (OOSEM) to form the basis of the technical process and methods due to its compatibility with the chosen management process. While a new modeling method could theoretically be proposed to incorporate ME activities, this research instead maps the ME process to the OOSEM activities, as shown in Figure 22, to provide flexibility to move between mission abstraction through component realization within an agile context without having to change the modeling method.

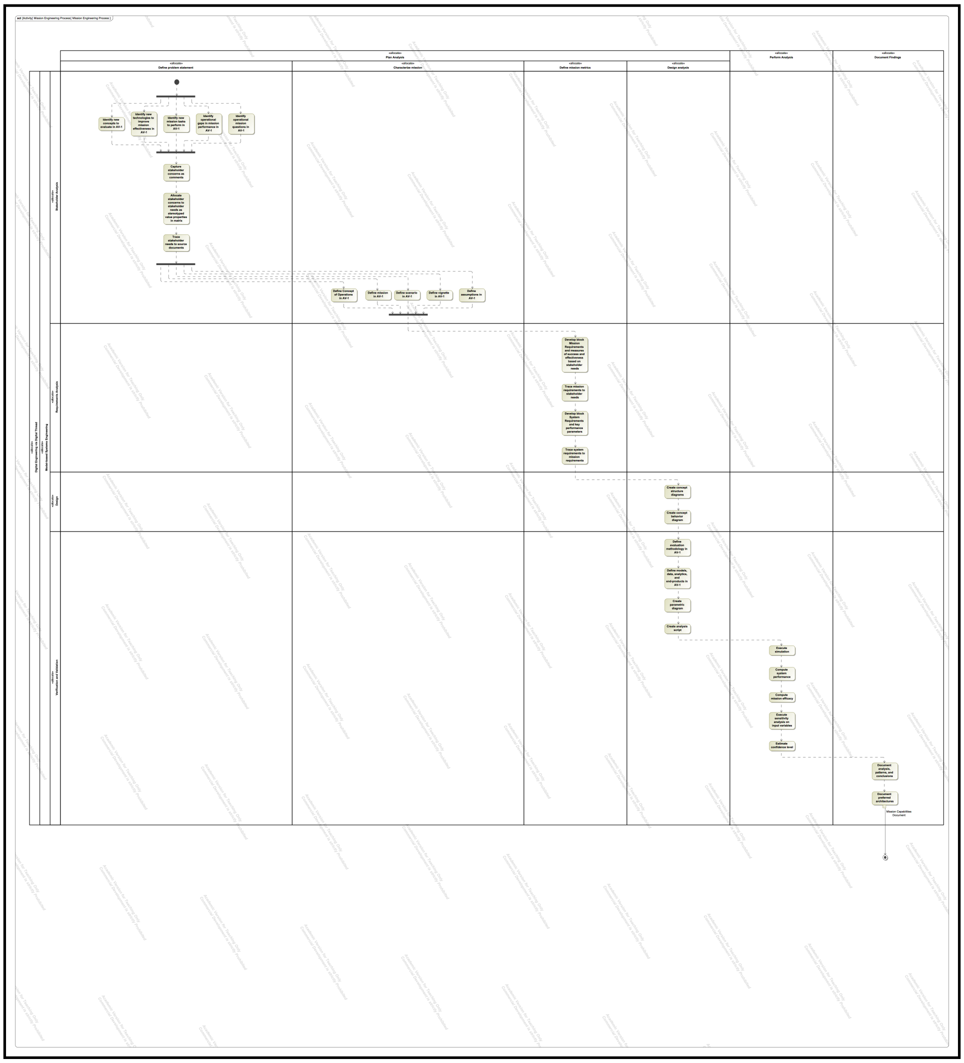

Figure 22: The Mission Engineering process mapped to the Object-Oriented System Engineering process

3.1 Sprint 1: High-altitude Threat and the Digital Thread

3.1.1 Sprint Planning

Within Sprint planning, research questions and hypotheses replace the typical work description style (Epic, Feature, User Story) but are used for a similar purpose. For this Sprint, the backlog research questions are:

1. Can a reference vignette be modeled in Cameo with the ME process?
2. Can a high altitude, highly elliptical threat be simulated using a Digital Thread (DT)?

The hypothesis is a cislunar threat can be modeled descriptively using the ME process and automatically simulated in a relevant physical environment. The following acceptance criteria establish quality requirements for the work conducted within the Sprint:

1. The ME process is executed through the vignette step
2. The ME information is contained in the Cameo model
3. A DT is created consisting of Cameo, Matlab, and System Toolkit (STK)
4. The threat's value properties are provided as inputs in a parametric diagram in Cameo
5. The threat model is simulated through a parametric diagram in Cameo
6. The scenario is generated in STK
7. Cameo receives data to verify scenario creation
8. An inspection of the STK scenario verifies scenario and value properties exist.

Key enablers are activities conducted that are not valuable in and of themselves, but enable the creation of value eventually. The enablers for Sprint 1 include:

1. complete ME process through vignette definition,

2. map the data flow and transformations,
3. create modeling objectives,
4. align digital tools to perform the transformations and identify interfaces,
5. and conduct digital tool training,
6. create a physical system model of the threat and add orbital parameters.

3.1.2 Daily Scrum

The Daily Scrum discusses the activities for the Sprint and describes how the DT and the threat's physical model are built. The first step of OOSEM is to set up the descriptive model in Cameo. Cameo Systems Modeler (CSM) is a descriptive modeling tool and is used for this research because the cubesat RA Kelly developed was built in CSM, CSM is available, and parallel efforts are using CSM. Kelly's cubesat RA is modified to adjust for the context of cislunar SDA, incorporate the ME process, a mission-level model, and additional physical models of systems in the Mission Thread (MT). Then, ambiguity is reduced by defining the problem statement, formatted as a mission question.

The initial ME activity defines the problem by asking, "What is the operational question? Is there a mission gap?" The mission gap is cislunar space is not monitored, enabling threat evasion and therefore a reduced probability of maintaining attribution. The next question is, "What mission, scenario, and vignette describes the context?" The threat vignette for Sprint 1 is characterized by a threat that exceeds an altitude greater than geosynchronous orbit ($> 35,786km$), has a perigee less than geosynchronous orbit ($< 35,786km$), and has the same physical characteristics (Table 1) as Knister's target to enable comparisons.

The threat's orbital parameters (Table 2) take the form of Keplerian Classical Orbital Elements (COEs) to simplify the modeling effort since the threat does not approach the Moon's sphere-of-influence (SOI).

Table 1: Physical Characteristics of a Reference Threat[7]

Reflectivity (aluminum)	C_D	0.88
Shape		Sphere
Size (radius)	r	1 meter

Table 2: Orbital Parameters of a Reference Threat

Semi-major Axis	α	100000 km
Eccentricity	e	0.8
Inclination	i	0 deg
Right Ascension of the Ascending Node	Ω	0 deg
Argument of Periapsis	ω	0 deg
True Anomaly	ν	0 deg

These orbital and system characterization parameters, along with string text, are captured in Cameo as value properties associated with a threat's mission and structure models. This completes the ME problem statement and mission characterization activities for Sprint 1.

The modeling objectives for this Sprint are to map the process, create a simplified threat representative system model, and pass value properties to the simulation software to execute the simulation. The data process is mapped in an activity diagram which depicts the value types and transformations that occurs in the DT. The DT is the software system that implements the data process and creates information to answer the research questions. Within the DT, Cameo is aligned to capture data in the form of orbital parameters and system characterization parameters that become inputs for the parametric model. Binding connectors are used to equate the threat's value properties to constraint parameters used in the constraint function.

Cameo is integrated with Matlab through a plug-in and executes Matlab scripts from a constraint block within a Cameo-based parametric diagram. Matlab is used for analysis because it also acts as an integration layer between Cameo and STK. The Matlab script referenced in the constraint block creates and executes the STK mission scenario. Script

created by Kelly and Knister is modified and appends satellite parameters to a satellite object. Matlab stores the COE and system parameters in the workspace and then passes parameters to STK, a simulation tool that propagates astrodynamics and can be controlled with external scripts. The Matlab script then generates a Boolean value based on whether the scenario is created, stores it in workspace memory, and then passes the value to Cameo at the end of the constraint function to verify the scenario is generated. This work constructs a basic DT of descriptive, simulation, and analytical tools that enables the simulation of a reference threat vignette and verifies basic DT functionality.

A Sprint Review is held to demonstrate the DT to stakeholders and discuss the results of the Sprint. Finally, a Sprint Retrospective is conducted to assess progress and to adapt the methodology before the next Sprint.

3.2 Sprint 2: Threat at Lagrange Point 1

Sprint 2 implements a threat vignette of an object in a Lyapunov orbit at L_1 to initiate the verification activity of the DT against previous research results.

3.2.1 Sprint Planning

For Sprint 2, one of the major management process additions is including estimates of effort to manage workloads in a more accurate way. The Scrum practice uses relative story points, capacity, and velocity to plan this work. The work associated with the research question below is estimated and refined to align with the available capacity. The research question selected for Sprint 2 is:

1. Can a threat orbiting Lagrange Point 1 be modeled in a descriptive tool and be simulated using a digital thread?

The hypothesis mirrors the initial hypothesis in Sprint 1 and states a cislunar threat can be modeled descriptively, using the ME process, and automatically simulated in a relevant

physical environment.

The following acceptance criteria established quality requirements for the work conducted with respect to the research question:

1. The ME process is executed through vignette step
2. The ME information is contained in the Cameo model
3. The DT has additional value properties associated with orbital parameters from a second threat in the parametric diagram
4. The threat model is simulated through a parametric diagram in Cameo
5. The scenario is generated in STK
6. Cameo receives data to verify scenario creation
7. An inspection of the STK scenario verifies scenario and value properties exist.

The enablers for Sprint 2 include:

1. complete ME process through vignette definition
2. create modeling objectives
3. map the data flow and transformations
4. align digital tools to perform the transformations and identify interfaces
5. conduct digital tool training.

3.2.2 Daily Scrum

The threat vignette for Sprint 2 is characterized by a threat in a Lyapunov orbit about L_1 with the initial analytical conditions of $Ay' = 15,000$ km, $Az' = 0$ km, $\phi = 0$ degrees,

and $\psi = 0$ degrees and the same physical characteristics (Table 1) as the High-altitude, highly-elliptical (HAHE) threat.[7]

The modeling objectives include mapping the process, creating a multi-threat model, and passing value properties to the simulation software to execute the simulation. The data process is mapped in an activity diagram which depicts the value types and transformations that occurs in the DT. New value properties are created in the Cameo mission model to capture initial conditions. This type of orbit does not follow Keplerian motion so new scripts are needed to design and implement the orbit in STK. The parameters associated with the value properties are used to calculate orbital patch conditions via a modified Matlab script that implements a process using Equations 4-6. Then, an STK scenario and Mission Control Sequence is generated and STK's differential corrector varies velocity parameters until the initial state vector yields a periodic, Lyapunov orbit. This work builds an additional reference threat vignette and continues the ME mission characterization activity. The DT is demonstrated during the Sprint review and the methodology is modified during the Sprint Restrospective.

3.3 Sprint 3: Modeling a Sensing Constellation

Sprint 3 continues the work of verifying the DT against previous research results by refining the problem definition and mission metrics for a sensing architecture focused on maintaining attribution of a threat at L_1.

3.3.1 Sprint Planning

For Sprint 3, the work associated with the research question below was estimated and refined to align capacity available with work to be done. The last Sprint saw the creation of a threat in the same Lyapunov orbit Knister used about L_1.[7] Sprint 3's focus is to create a sensing constellation Knister used as one of the alternate sensing concepts to verify the DT.

The research question selected for Sprint 3 is:

1. Can a geosynchronous Earth orbit (GEO)-based sensing constellation be modeled in a descriptive tool and simulated using a digital thread?

The hypothesis mirrors the previous hypothesis and states a blue GEO constellation can be modeled descriptively, using the mission engineering process, and automatically simulated in a relevant physical environment. The following acceptance criteria establish quality requirements for the work conducted with respect to the research question:

1. The ME process is executed
2. The ME information is contained in the Cameo model
3. The DT has additional value properties associated with orbital parameters for the sensing constellation in the parametric diagram
4. The sensing constellation model is simulated through a parametric diagram in Cameo
5. The scenario is generated in STK
6. Cameo receives data to verify scenario creation
7. An inspection of the STK scenario verifies scenario and value properties exist.

The enablers for Sprint 3 include:

1. create modeling objectives
2. map the data flow and transformations
3. align digital tools to perform the transformations and identify interfaces
4. conduct digital tool training.

3.3.2 Daily Scrum

The ME process is executed and implemented with OOSEM activities and methods to develop a derive the mission context for the sensing architecture. The modeling objectives include creating proper views of the mission as refined during the ME process, creating a sensing architecture model, mapping the analysis process, and passing value properties to the simulation software to execute the simulation.

A problem statement is generated by creating a hypothetical commander's intent and needs, goals, gaps, or opportunities for cislunar SDA. The mission is then characterized by describing entry conditions and boundaries such as the operational environment, scenarios, vignettes, assumed rules of engagement, and Concept of Operations (CONOPs). Some of these details are captured in an All Viewpoint (AV)-1 document and their relationships are established in a modified Block Definition Diagram (BDD) diagram adapted from the Model-based Systems Engineering (MBSE) Guidebook.[38] Then, key capabilities of the MT are depicted in a use-case diagram and an external value function is created and depicted in an Operational Viewpoint (OV)-5b activity diagram. The use-case diagram and the OV-5b are used to derive key capabilities and constraints which are captured in a requirements diagram as stereotyped block elements. Next, a nominal model of the current SDA architecture is depicted in a System Viewpoint (SV)-1 BDD to depict the system-of-interest amongst other elements of the enterprise architecture. Since a cislunar SDA sensing architecture does not currently exist, mission requirements are not defined for any of the sensing architectures to provide flexibility in discovering the primary drivers in the tradespace. However, mission metrics are still defined to analyze the effectiveness and performance of materiel concepts but are traced to mission goals and objectives instead of requirements.

A Measure of Success (MOS) is related to the mission architecture's ability to accurately attribute or maintain attribution of a threat in cislunar space. However, the MOS would require orbit determination and object characterization routines, as well as statistical

analysis. Since this thesis is focused on the sensing architecture and in particular, the detection capability, Measure of Effectiveness (MOE) and Measure of Performance (MOP) are more appropriate. Since requirements have not been identified, the chosen measures selected enable comparative analysis of different concepts as opposed to discrete requirements the sensing architecture has to meet for the mission. The analysis incorporates MOEs Knister previously defined including Mean Detect Time (MDT), Mean Track Time (MTT), and Mean Time Between Tracks (MTBT) since they are indicative of a sensing architecture's effectiveness.[7] Typically, the quantity of measurements drives the accuracy of an Orbit Determination (OD), a key task of SDA to predict future positions and velocities, and thereby determine threat risks. This heuristic is captured in the first lower-level MOE, MDT. MDT is the amount of time the Signal-to-Noise Ratio (SNR) of the target(s) remains above the threshold value during the simulation. MDT is calculated using Equation 8:

$$MDT = \frac{\sum_{i=1}^{n_{obj}} \frac{t_{detected,i}}{t_{scenario}}}{n_{obj}} \tag{8}$$

where, $n_{objects}$ is the total number of targets, $t_{detected,i}$ is the number of detection time bins for the i^{th} target, and $t_{scenario}$ is the total number of time bins in the scenario. The next metric is MTT as shown in Equation 9. MTT is the average period of the time the SNR remains above the threshold value which is valuable for describing the ability of a sensor to maintain custody of the target:

$$MTT = \frac{\sum_{i=1}^{n_{obj}} \frac{\sum_{j=1}^{n_{obs,i}} T_{obs,j,i}}{n_{obs,i}}}{n_{obj}} \tag{9}$$

where n_{obj} is the total number of targets to be tracked in the scenario, $n_{obs,i}$ is the number of observations for the i^{th} target, $T_{obs,j,i}$ is the time period of the j^{th} observation of the i^{th} target.

The accuracy of the OD estimate decreases as the time from successive detections in-

creases due to orbital instability and inherent error in the measurements. MTBT, as shown in Equation 10, captures this performance requirement and is the average gap in time between detections:

$$MTBT = \frac{\sum_{i=1}^{n_{obj}} \frac{\sum_{j=1}^{n_{gaps,i}} T_{gaps,j,i}}{n_{gaps,i}}}{n_{obj}} \quad (10)$$

where n_{obj} is the total number of targets to be tracked in the scenario, $n_{gaps,i}$ is the number of the observation gaps for the i^{th} target, $T_{gaps,j,i}$ is the time period of the j^{th} observation gap of the i^{th} target. Each of these MOEs is a function of lower-level capability or system MOP. The primary MOP in this analysis is SNR because all of the MOE are a function of a detector's SNR value in a given scenario. A successful detection occurs when the SNR is above an assumed threshold value of six.[7]

$$SNR = \frac{N_e}{\sqrt{N_e + \eta(N_d)n_{pixel} + N_r^2}} \quad (11)$$

Where N_d is dark noise, N_r is read noise, and N_e is discretized electrons. The discretized electrons (N_e) are a function of power received at the detector, detector efficiency, integration time, average wavelength detected, Planck's constant, the speed of light, optical transmittance, aperture diameter, atmospheric transmittance, and range. These parameters are captured as value properties of certain entities and the equations are captured in constraint blocks for use in a parametric analysis.

Traceability is maintained by generating requirement diagrams that connect measures all of the way back to source documentation. At this point, the ME problem statement definition, mission characterization, and mission metric definition activities are now complete. Then, the analysis process is mapped in an activity diagram, a parametric diagram is created, and the mission scenario is simulated to answer the Sprint 3 research question and to meet the acceptance criteria. A Sprint Review and Sprint Retrospective are performed

to assess progress as well as adapt the methodology before the next Sprint.

3.4 Sprint 4: Verifying the Digital Thread

Sprint 4 focuses on verifying the DT against trial results from Knister's research. The chosen trial is a GEO-based, four satellite constellation concentrated on the object that was previously modeled in the L_1 Lyapunov orbit.

3.4.1 Sprint Planning

For Sprint 4, the work associated with the research question below is estimated and refined to align capacity available with work to be done. The previous Sprint saw the completion of the ME process through metric definition and the creation of modeling artifacts depicting the need for a cislunar SDA capability. Sprint 4's focus is to evaluate a sensing constellation Knister used as one of the alternate sensing concepts to verify the DT. The research question selected for Sprint 4 is:

1. Can the cislunar SDA DT be verified?

The hypothesis is the DT can be verified against previous data generated by a previous evaluation of a GEO-based sensing constellation.

The following acceptance criteria establish quality requirements for the work conducted with respect to the research question:

1. The ME process is executed
2. The ME information is contained in the Cameo model
3. The sensing constellation model is evaluated through a parametric diagram in Cameo
4. Performance metrics are calculated for the sensing constellation
5. Cameo receives data to verify scenario creation.

The enablers for Sprint 4 include:

1. create modeling objectives

2. modify existing Matlab scripts

3. conduct digital tool training.

3.4.2 Daily Scrum

The ME Design of Analysis enables creativity and manages complexity by hypothesizing and modeling a Doctrine, Organization, Training, materiel, Leadership, Personnel, Facilities, and Policy (DOTmLPFP) solution and implements an analytical experiment to measure and compare the performance of alternative concepts. The sensing constellation chosen for this mission scenario includes four satellites, spaced every 90 degrees in the same plane, with COE of:

Table 3: Sensing Constellation Orbital Parameters

Semi-major Axis	α	$42,164$ km
Eccentricity	e	0
Inclination	i	0 deg
Mean Anomaly	ν	$0, 90, 180, 270$ deg

The modeling objectives include mapping the process, modeling the parametric analysis, and passing value properties to the simulation software to execute the simulation, and analyzing the data created by the simulation. This thesis models the signal chain for Electro-optical (EO) systems that collect and digitize Electro-magnetic (EM) radiation. The mathematical model for the signal chain has to account for the signal source radiance and albedo effects, as well as, various propagation, reflection, and optical losses. Since this mission scenario is designed verify the DT, the mathematical model and parameter values mirror those used in the analysis Knister conducted. For this scenario, the Sun is the EM radiation source,

and the radiation is reflected from the cislunar object to the observing sensor. The Sun's radiance at the average orbital radius of the Earth is $S = 1366.1 Wm^{-2}$.[39] This value will vary periodically for cislunar objects during the course of their orbit around the Earth, but a steady-state value will be used to simplify analysis. Additional radiance from Earth and Moon albedo are also ignored to simplify the analysis. For the object at L_1, the total power of the radiation upon a a spherical object is a multiplicative of the intensity of the solar irradiance and a geometric term as described in Equation 12.

$$P = I\pi {r_{threat}}^2 \tag{12}$$

This power is reduced once it reflects off of the object. The reduction is determined by the Lambertian reflection coefficient which is dependent on threat satellite's the surface reflectance, C_D, and varies as the solar phase angle, β, changes at each time step. This coefficient is described by Equation 13.

$$\psi = \frac{2C_D}{3\pi}(sin(\beta) + (\pi - \beta)cos(\beta)) \tag{13}$$

The power term from Equation 12 is then multiplied with a Lambertion reflection coefficient, ψ, to calculate the total reflected power from the object as defined by Equation 14.

$$P_{reflected} = P\psi \tag{14}$$

The power received at the sensor sensor aperture can be calculated as a simple path loss resulting from propagation of light over a distance, R:

$$P_{in} = \frac{P_{reflected}}{R^2} \tag{15}$$

Then, the telescope apparatus is modeled via Equation 16 to calculate the power received

at the detector:

$$P_{detector} = P_{in}\tau_{opt}\pi {d_{aperture}}^2 \tag{16}$$

where optical transmittance, τ_{opt}, accounts for the signal loss of the system and is set to 0.9, while the aperture diameter, ${d_{aperture}}^2$ in the geometric term accounts for the optical gain of the telescope and is set to 0.3 meters. The power received by the detector can then be used to compute the number of N_e counted by the sensor:

$$N_e = \frac{P_{detector}\eta\lambda_{avg}}{hc}t_{int} \tag{17}$$

where ν is detector efficiency, t_{int} is integration time, λ_{avg} is the average wavelength detected, h is Planck's constant, and c is the speed of light, with associated values in the table below:

Table 4: Discretized Electron Calculation Parameters

Detector efficiency	ν	0.65
Integration time	t_{int}	1 sec
Average wavelength	λ_{avg}	600 nm

The discretized electrons can then be used in conjunction with noise terms, dark noise (N_d) and read noise (N_r), as well as detector efficiency, ν, and the number of pixels, n_{pixel}, to calculate the sensor's SNR MOP, as defined in Equation 11. The values used in this analysis for N_d and N_r are 12 electrons per pixel per second, and 6 electrons per pixel per second, respectively. It is assumed the power spectral density of the object observed by the sensor falls within the instantaneous field-of-view of a single pixel ($n_{pixel} = 1$) due to the extreme distances associated with the observation. Defining this signal chain allows for the SNR value to be calculated at each time step of the simulation, providing the capability to ascertain the effectiveness of the proposed solution using the time-dependent MOE.

The process is depicted in an activity diagram and describes how the input parameters are used and transformed by the constraint functions to execute simulations and analyses to calculate the four metrics. The input value type variables for this parametric analysis define payload, orbit, target, and environmental parameters. The DT is modified to include scripts that request STK to calculate access opportunities, ranges, and vectors between satellites and celestial bodies as constrained by Earth occlusion and sensor exclusion angles. This data is then returned to Matlab scripts to solve the mathematical EO model, the MOP, and the MOE. These measures and their associated values are then stored in an instance table within Cameo to verify requirements and enable the comparison of multiple mission and system concepts.

Functional tests are performed to verify the DT's ability to evaluate a proposed solution against data Knister generated in Figure 13. There were inconsistencies noted between the literature review and the available source code between the values used for source radiation incident on the object, the design parameters of the Lyapunov orbit, and the altitude of the GEO-based sensing constellation, as summarized in Table 5:

Table 5: Inconsistencies Between Literature and Code

		Literature	Code	Unit
Radiance (pg. 19)[7]	S	1366.1	1600	W/m^2
Lyapunov Orbit Parameters	A_y'	15,000	5,000	km
(pg. 40)[7]	A_z'	20,000	0	km
	θ	180	0	deg
GEO Altitude	alt	35,786	42,164	km

Of particular note is the difference of A_z' since the threat object would be in a Lissajous orbit rather than a Lyapunov orbit, influencing the impact of the sensors' five degree lunar exclusion angle. These inconsistencies require performing a total of 32 trials to confidently assess the functionality of the DT, as any of these variables influence the computation of the MOP and MOE. These trials are captured in an Instance Table (Figure 23) associated with the parametric analysis. These 32 instances are a combination of the five variables with two

alternative values.

#	Name	.Source Intensity : Real	.lagrangePoi	.theta : degrees[angul (deg)	.y-amplitude : distance[km] (km)	.z-amplitude : distance[km] (km)	.altitude : distance[m] (m)
1	4 Blue GEO vs Threat Vignette (L1 Lyapunov)	1366.1	L1	0 deg	5000 km	0 km	3.5786E7 m
2	4 Blue GEO vs Threat Vignette (L1 Lyapunov)1	1366.1	L1	0 deg	5000 km	0 km	4.2164E7 m
3	4 Blue GEO vs Threat Vignette (L1 Lyapunov)2	1366.1	L1	0 deg	5000 km	20000 km	3.5786E7 m
4	4 Blue GEO vs Threat Vignette (L1 Lyapunov)3	1366.1	L1	0 deg	5000 km	20000 km	4.2164E7 m
5	4 Blue GEO vs Threat Vignette (L1 Lyapunov)4	1366.1	L1	0 deg	15000 km	0 km	3.5786E7 m
6	4 Blue GEO vs Threat Vignette (L1 Lyapunov)5	1366.1	L1	0 deg	15000 km	0 km	4.2164E7 m
7	4 Blue GEO vs Threat Vignette (L1 Lyapunov)6	1366.1	L1	0 deg	15000 km	20000 km	3.5786E7 m
8	4 Blue GEO vs Threat Vignette (L1 Lyapunov)7	1366.1	L1	0 deg	15000 km	20000 km	4.2164E7 m
9	4 Blue GEO vs Threat Vignette (L1 Lyapunov)8	1366.1	L1	180 deg	5000 km	0 km	3.5786E7 m
10	4 Blue GEO vs Threat Vignette (L1 Lyapunov)9	1366.1	L1	180 deg	5000 km	0 km	4.2164E7 m
11	4 Blue GEO vs Threat Vignette (L1 Lyapunov)10	1366.1	L1	180 deg	5000 km	20000 km	3.5786E7 m
12	4 Blue GEO vs Threat Vignette (L1 Lyapunov)11	1366.1	L1	180 deg	5000 km	20000 km	4.2164E7 m
13	4 Blue GEO vs Threat Vignette (L1 Lyapunov)12	1366.1	L1	180 deg	15000 km	0 km	3.5786E7 m
14	4 Blue GEO vs Threat Vignette (L1 Lyapunov)13	1366.1	L1	180 deg	15000 km	0 km	4.2164E7 m
15	4 Blue GEO vs Threat Vignette (L1 Lyapunov)14	1366.1	L1	180 deg	15000 km	20000 km	3.5786E7 m
16	4 Blue GEO vs Threat Vignette (L1 Lyapunov)15	1366.1	L1	180 deg	15000 km	20000 km	4.2164E7 m
17	4 Blue GEO vs Threat Vignette (L1 Lyapunov)16	1600	L1	0 deg	5000 km	0 km	3.5786E7 m
18	4 Blue GEO vs Threat Vignette (L1 Lyapunov)17	1600	L1	0 deg	5000 km	0 km	4.2164E7 m
19	4 Blue GEO vs Threat Vignette (L1 Lyapunov)18	1600	L1	0 deg	5000 km	20000 km	3.5786E7 m
20	4 Blue GEO vs Threat Vignette (L1 Lyapunov)19	1600	L1	0 deg	5000 km	20000 km	4.2164E7 m
21	4 Blue GEO vs Threat Vignette (L1 Lyapunov)20	1600	L1	0 deg	15000 km	0 km	3.5786E7 m
22	4 Blue GEO vs Threat Vignette (L1 Lyapunov)21	1600	L1	0 deg	15000 km	0 km	4.2164E7 m
23	4 Blue GEO vs Threat Vignette (L1 Lyapunov)22	1600	L1	0 deg	15000 km	20000 km	3.5786E7 m
24	4 Blue GEO vs Threat Vignette (L1 Lyapunov)23	1600	L1	0 deg	15000 km	20000 km	4.2164E7 m
25	4 Blue GEO vs Threat Vignette (L1 Lyapunov)24	1600	L1	180 deg	5000 km	0 km	3.5786E7 m
26	4 Blue GEO vs Threat Vignette (L1 Lyapunov)25	1600	L1	180 deg	5000 km	0 km	4.2164E7 m
27	4 Blue GEO vs Threat Vignette (L1 Lyapunov)26	1600	L1	180 deg	5000 km	20000 km	3.5786E7 m
28	4 Blue GEO vs Threat Vignette (L1 Lyapunov)27	1600	L1	180 deg	5000 km	20000 km	4.2164E7 m
29	4 Blue GEO vs Threat Vignette (L1 Lyapunov)28	1600	L1	180 deg	15000 km	0 km	3.5786E7 m
30	4 Blue GEO vs Threat Vignette (L1 Lyapunov)29	1600	L1	180 deg	15000 km	0 km	4.2164E7 m
31	4 Blue GEO vs Threat Vignette (L1 Lyapunov)30	1600	L1	180 deg	15000 km	20000 km	3.5786E7 m
32	4 Blue GEO vs Threat Vignette (L1 Lyapunov)31	1600	L1	180 deg	15000 km	20000 km	4.2164E7 m

Figure 23: Trials in an Instance Table

The results of the various trials, with all other variables held constant for a scenario starting on September 14, 2019, are then compared to those Knister calculated using another constraint property that identifies whether the calculated measures are within a tolerance of 2% of Knister's values. Once these steps are performed, the primary ME activities to design and perform the analysis are complete for the proposed operational and engineering research questions.

3.5 Conclusion

The Agile DME methodology created in this chapter is implemented to answer the research questions and meet research objectives. The technical process and associated methods are performed using digital tools to model processes and systems, accelerate analyses, and capture data. The resulting DT performs several functional tests to verify the ability to evaluate proposed cislunar SDA concepts as modeled in a mission-level RA. The results generated during the implementation of the methodology are presented in Chapter 4.

IV. Results and Analysis

Applying an Agile Digital Mission Engineering (DME) methodology to particular cislunar Space Domain Awareness (SDA) operational questions yield several sections of results. Each section contains comprehensive results with respect to each two week Sprint conducted.

4.1 Sprint 1 Review

Sprint 1 accomplished multiple objectives in pursuit of answering two research questions. The SDA Reference Architecture (RA) model was set up, the Mission Engineering (ME) Problem Statement and Mission Characterization activities were executed through vignette definition, and a reference threat vignette was modeled and simulated in a digitally relevant environment.

The modification of the cubesat RA resulted in the addition of the ME process to the Guidance package, as depicted in Figure 24.

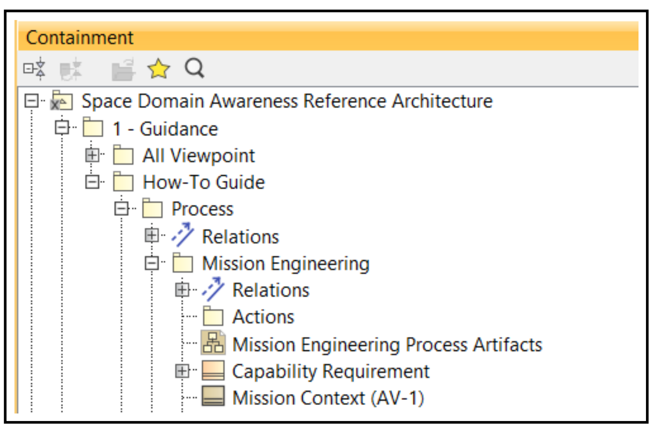

Figure 24: The Mission Engineering Process Model in the Reference Architecture

In addition, the artifacts associated with the ME process were modeled, as depicted in Figure 25. This model depicts the relationships between source documents, derived requirements, and evaluation activities.

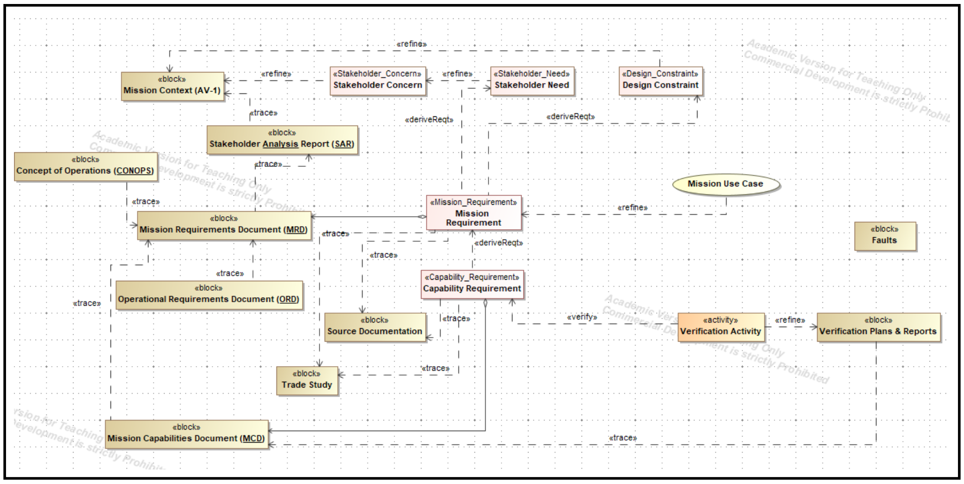

Figure 25: The Artifacts Associated with the Mission Engineering Process

The Stakeholder Analysis activity enabled traceability by documenting the research questions as comments (Figure 26) and relating those to Stakeholder Needs (Figure 27), as seen in Figure 28.

Establishing a relationship between the research questions and tasks provided the ability to trace derived requirements and models in the future.

A generic mission model was then created along with additional physical models of systems in the Mission Thread (MT), as seen in Figure 29.

A cislunar SDA mission model was also created. This is where the research questions and cislunar threat vignette reside. Orbital, system, and simulation parameters were captured as value properties within the high-altitude, highly-elliptical threat vignette model, as depicted in Figure 30, completing the stakeholder analysis and problem statement activities for Sprint 1.

Then, the data process was mapped as part of the modeling objectives as depicted in Figure 31. A parametric diagram was populated with part properties and their associated value properties, as seen in Figure 32, which were then connected to a constraint block using binding connectors.

Though the parametric model can execute with default values, an instance table was

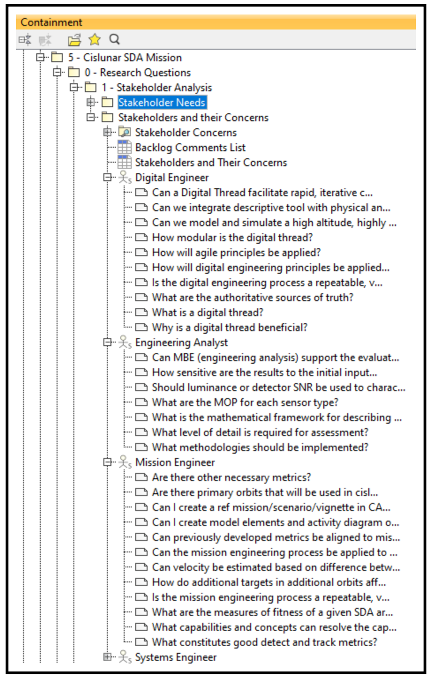

Figure 26: Research Questions in Comment Form

created for the High-altitude, highly-elliptical (HAHE) satellite as seen in Figure 33.

The orbit, system, and simulation parameters became the input variables for the parametric diagram that simulated the threat vignette and confirmed the System Toolkit (STK) scenario was created by generating a true Boolean value. An inspection of the STK scenario also verified the scenario was generated properly, as seen in Figure 34. During the Sprint 1 Retrospective, the consensus was to start estimating the effort associated with the work selected because several of the original backlog questions selected for Sprint 1 had to be delayed to a future Sprint.

Within Sprint 1, the ME process was completed through the threat vignette, Object-Oriented Systems Engineering (OOSE) methods were used to document and model the information, and a descriptive tool executed a simulation automatically through an analytical tool, satisfying acceptance criteria and resulting in a foundational Digital Thread (DT) to further study cislunar SDA operational questions.

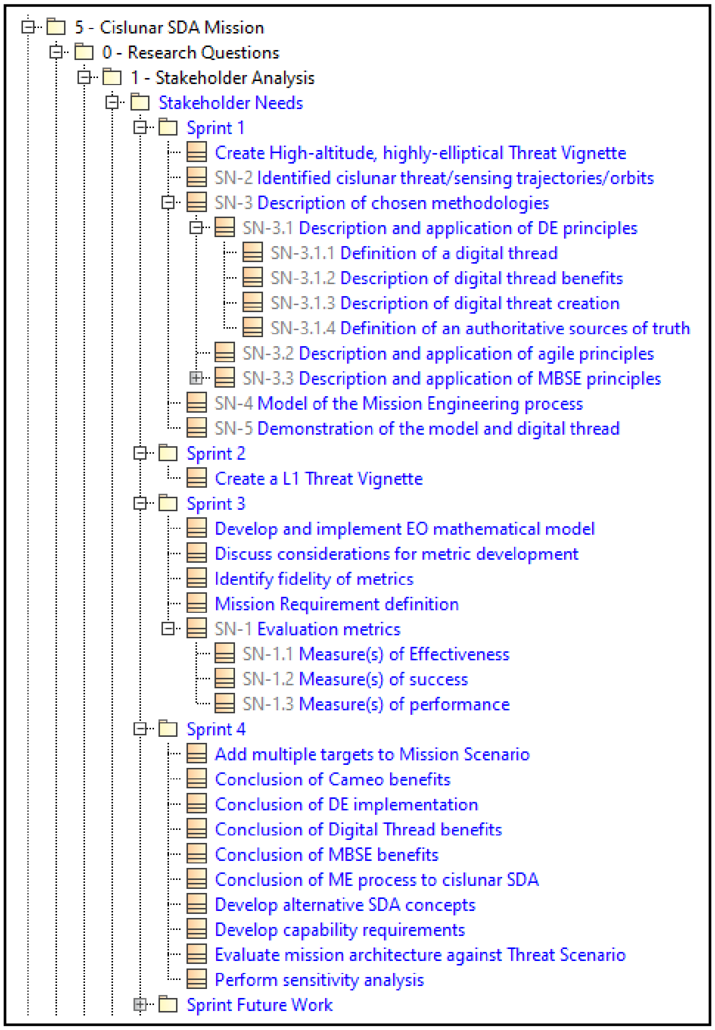

Figure 27: Stakeholder Needs Derived from Research Questions

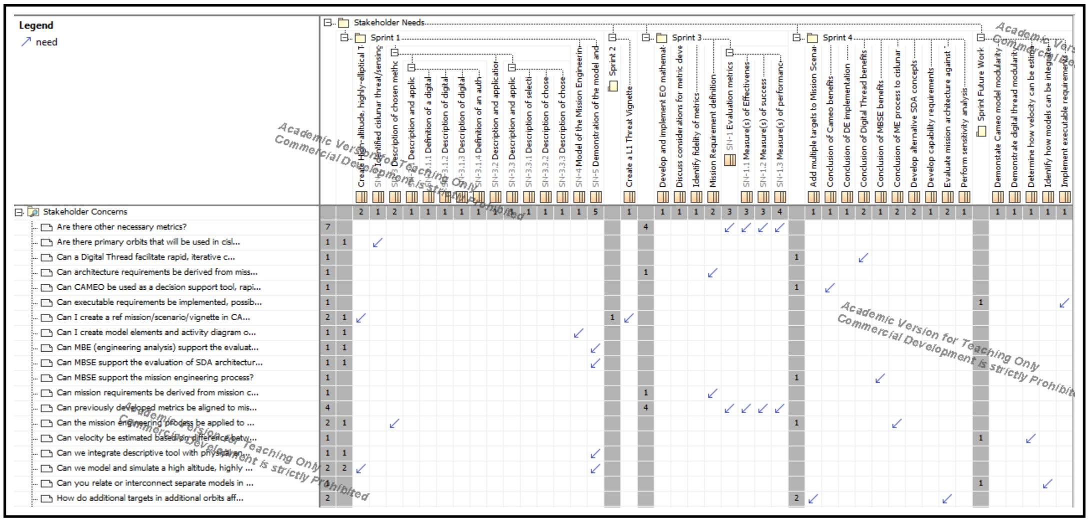

Figure 28: Stakeholder Needs Traced to Research Questions

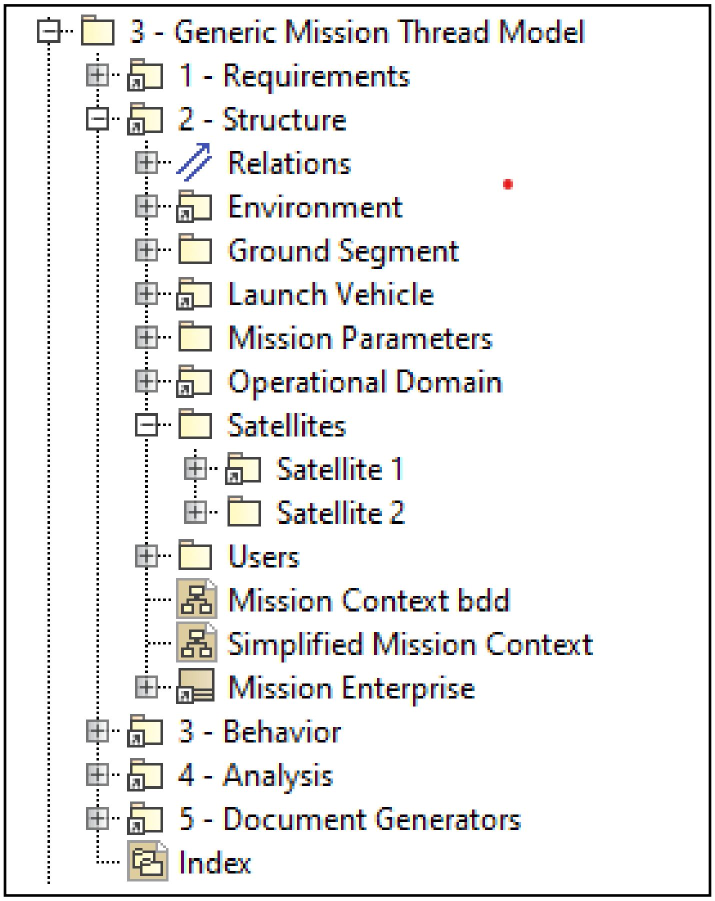

Figure 29: Mission Thread Model

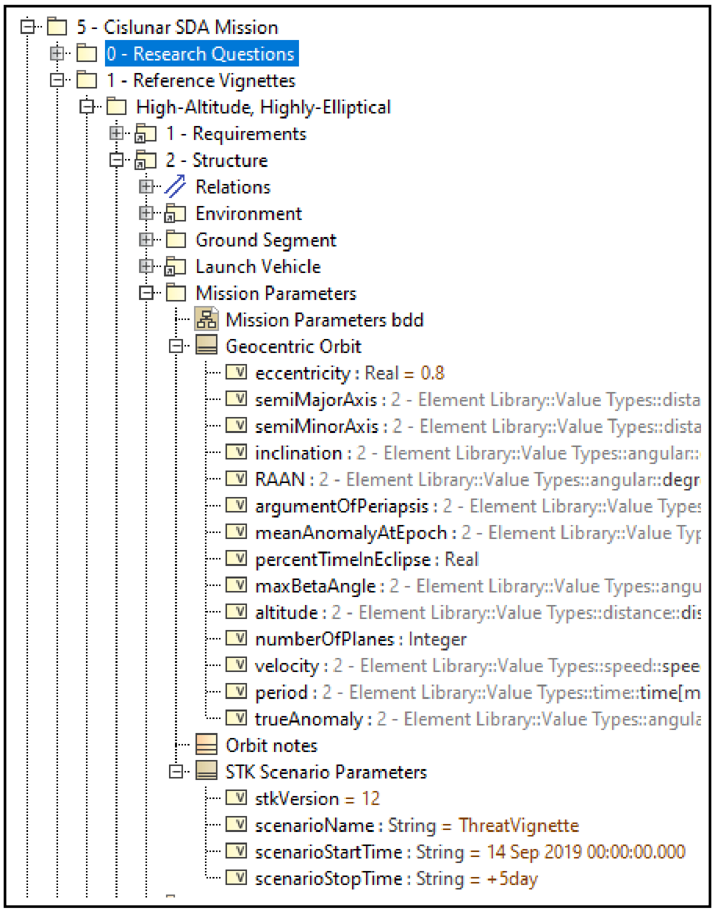

Figure 30: High-altitude, Highly-elliptical Threat Model

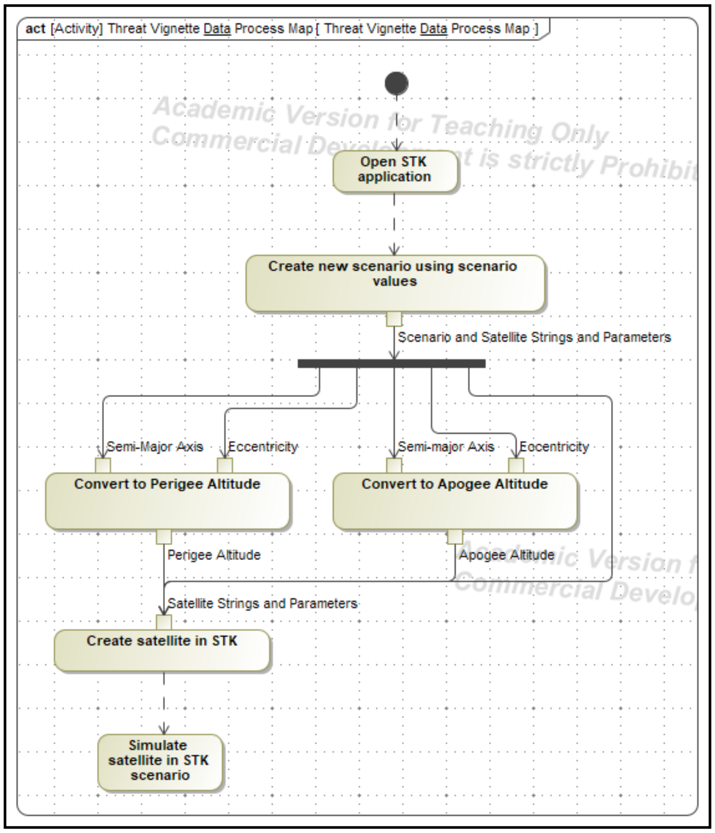

Figure 31: HAHE Threat Vignette Data Flow and Transformations

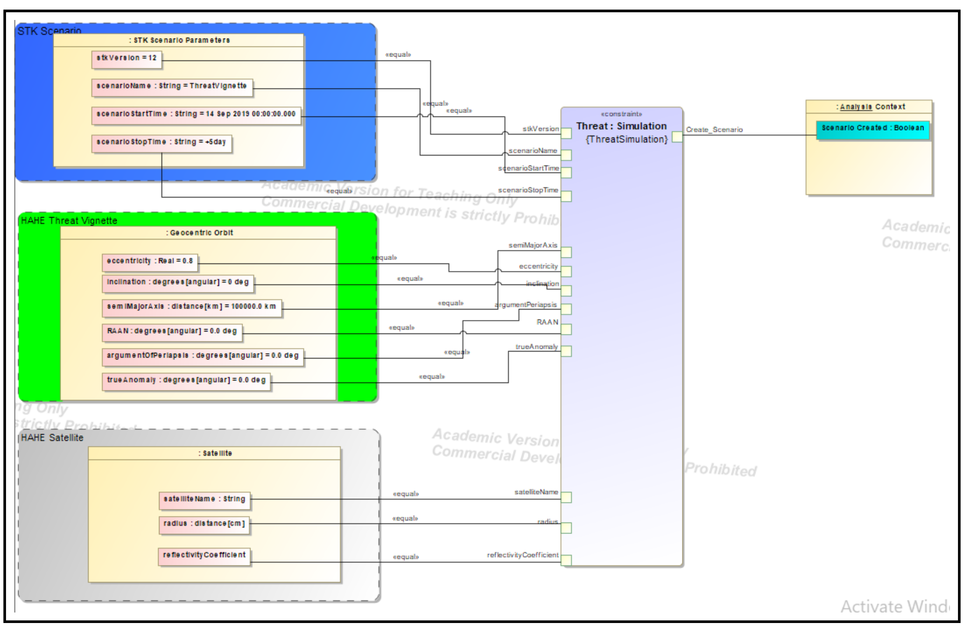

Figure 32: HAHE Parametric Model

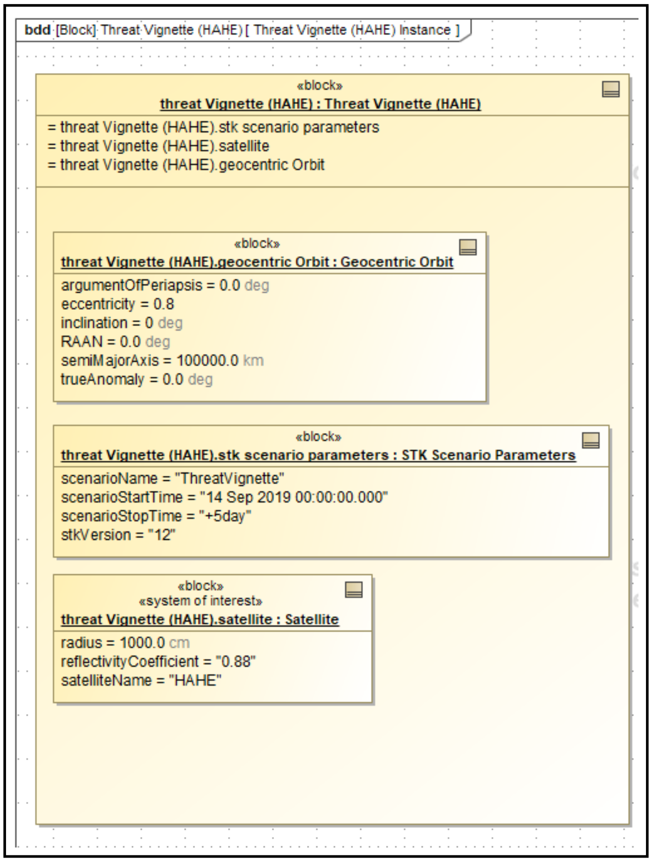

Figure 33: HAHE Instance

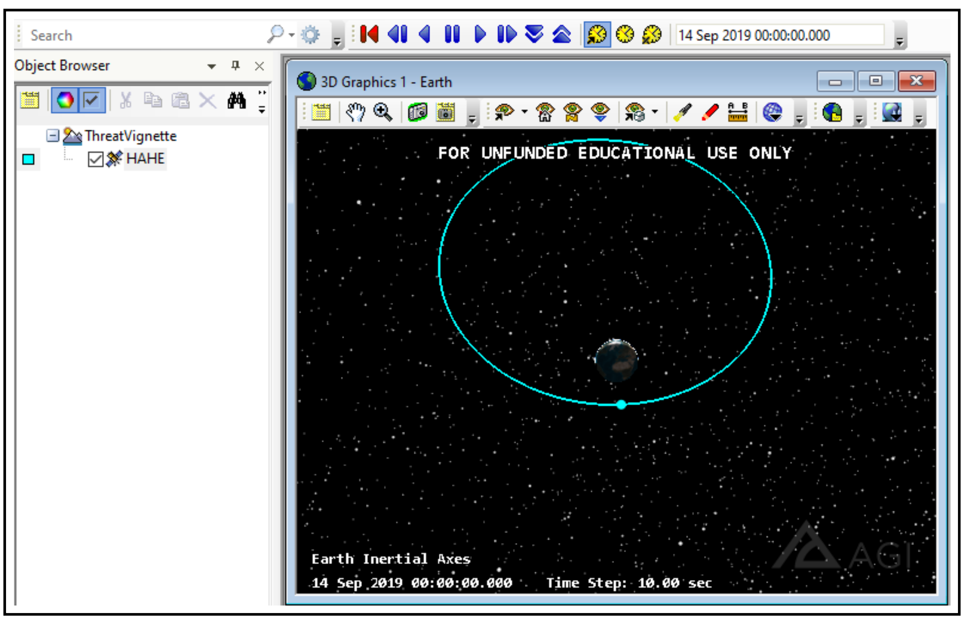

Figure 34: HAHE Orbit in STK

4.2 Sprint 2 Review

Sprint 2 executed the ME process through vignette definition and created a reference L_1 threat model and a parametric model to simulate the threat in a relevant environment. The L_1 threat model now contains the value properties associated with the Lyapunov orbit in addition to other satellite and simulation parameters as seen in Figure 35 and 36.

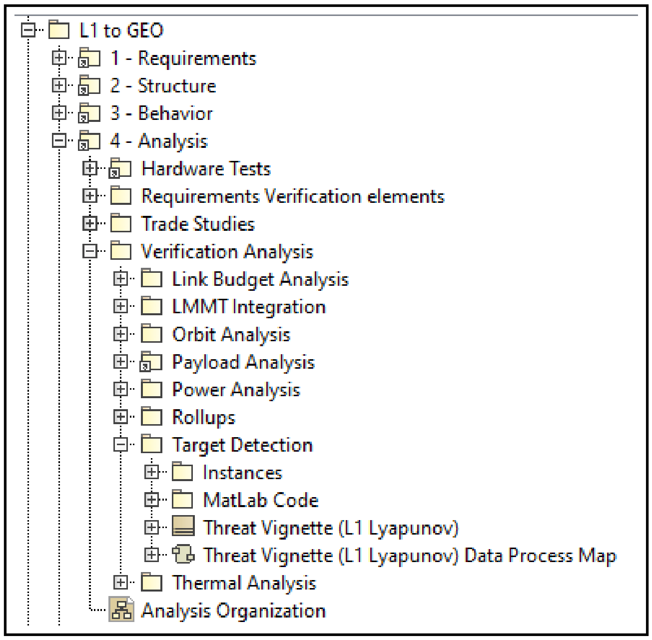

Figure 35: L1 Threat Model

The data process was mapped, resulting in Figure 37. The data process was much more complex for the threat at L_1 because an initial state vector and patch point conditions needed to be calculated using the linearized Equations of Motion (EOM) and the initial Lyapunov orbit design variables. These patch point conditions then had to be passed to a STK Mission Control Sequence that applied a differential corrector to create a periodic Lyapunov orbit.

A parametric diagram was populated with part properties and their associated value properties, as seen in Figure 38, which were then connected to a constraint block using binding connectors.

The orbit, system, and simulation parameters became the input variables for the para-

metric diagram that simulated the threat vignette and confirmed the STK scenario was created by generating a true Boolean value. An inspection of the STK scenario verified the scenario was generated properly, as seen in Figure 39 and Figure 40.

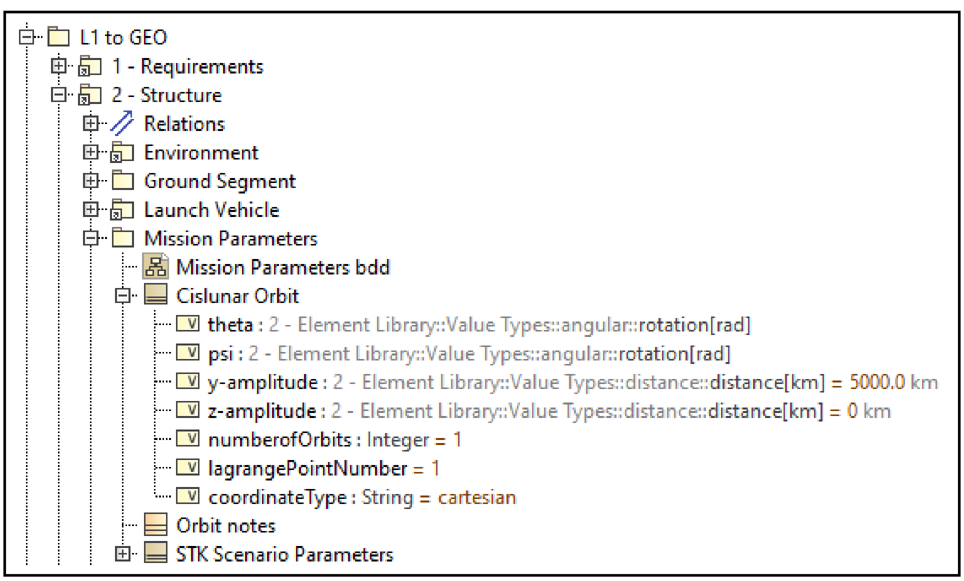

Figure 36: L1 Threat Model Value Types

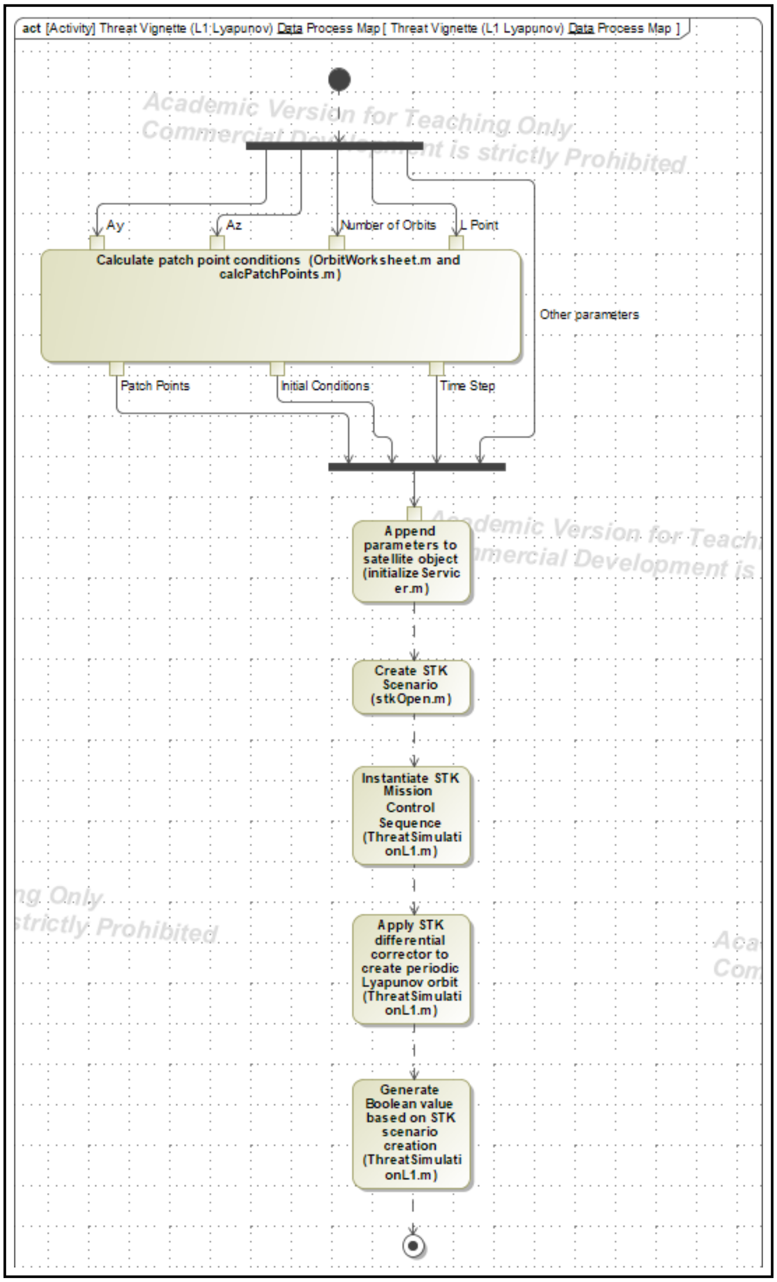

Figure 37: L1 Threat Vignette Process Map

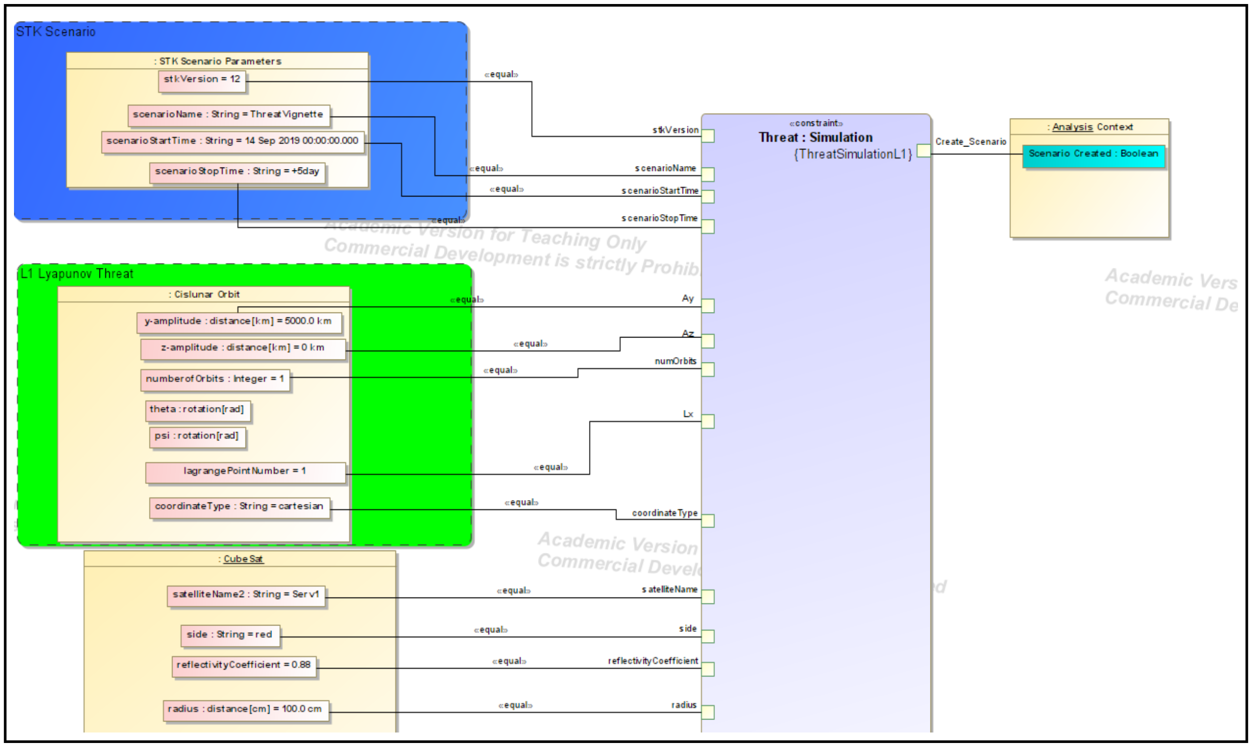

Figure 38: L1 Threat Vignette Parametric Model

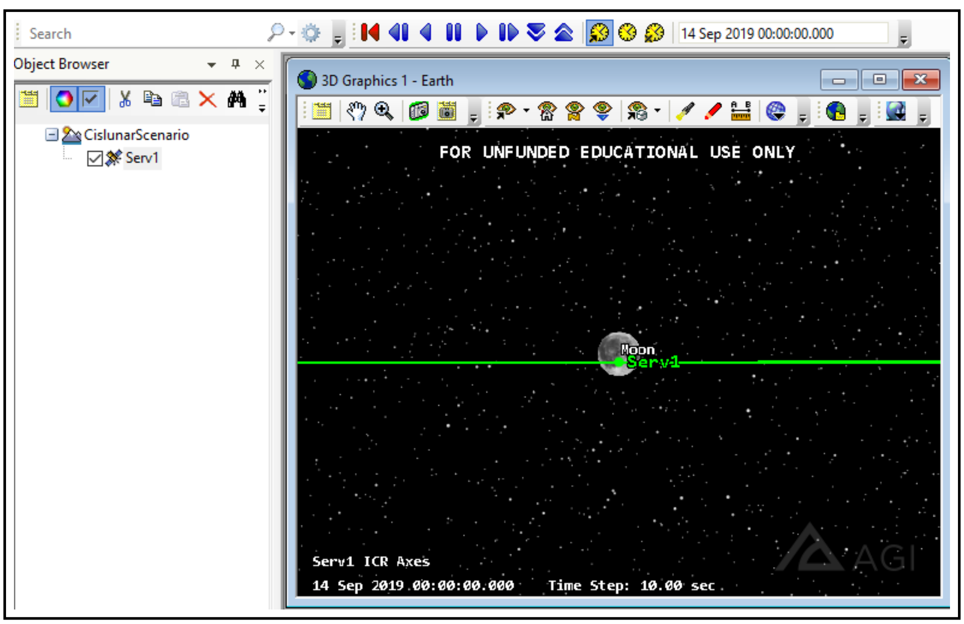

Figure 39: Lyapunov Planar View

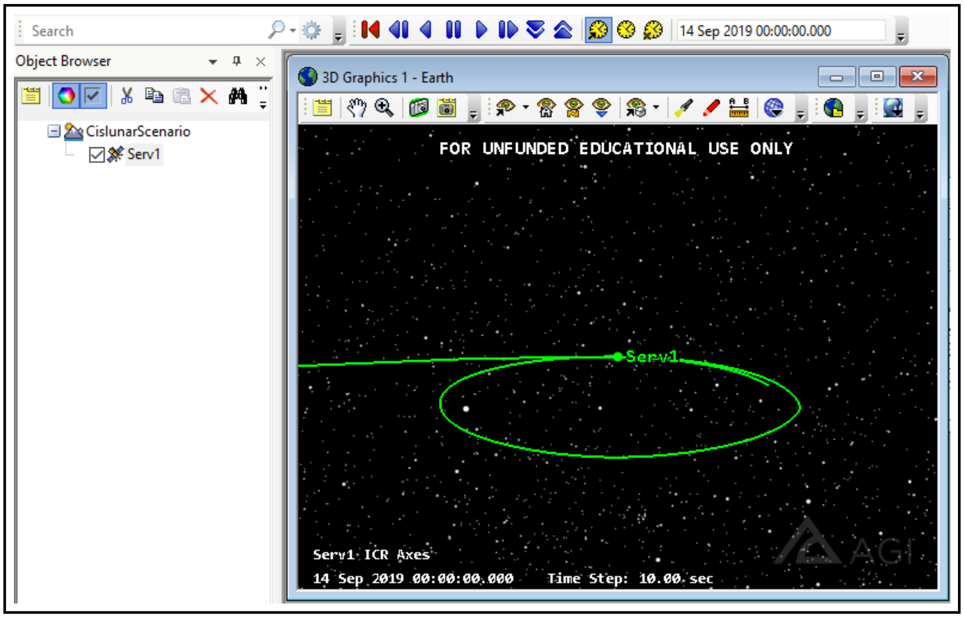

Figure 40: Lyapunov Top View

4.3 Sprint 3 Review

The initial activity for Sprint 3 produced an All Viewpoint (AV) package (Figure 41) with overview and summary information along with a repository for enterprise information that serves to guide the SDA architecture development and evaluation process.

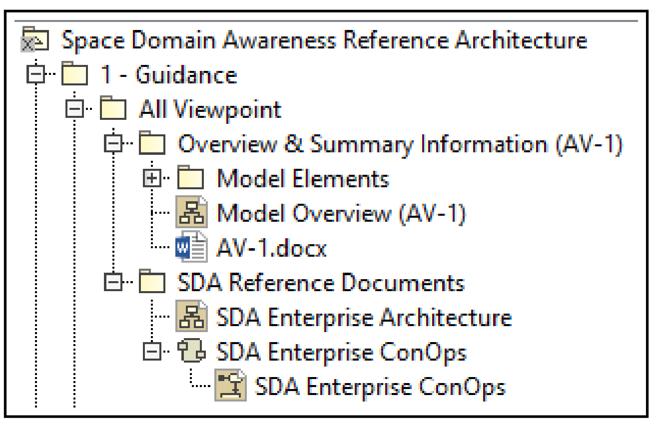

Figure 41: AV-1 Package

The adapted AV-1 Block Definition Diagram (BDD) diagram (Figure 42)provides the

relationships between the source and reference artifacts to the rest of the solution definition and assessments. The reference artifacts were synthesized in the AV-1 document shown in Figure 41.

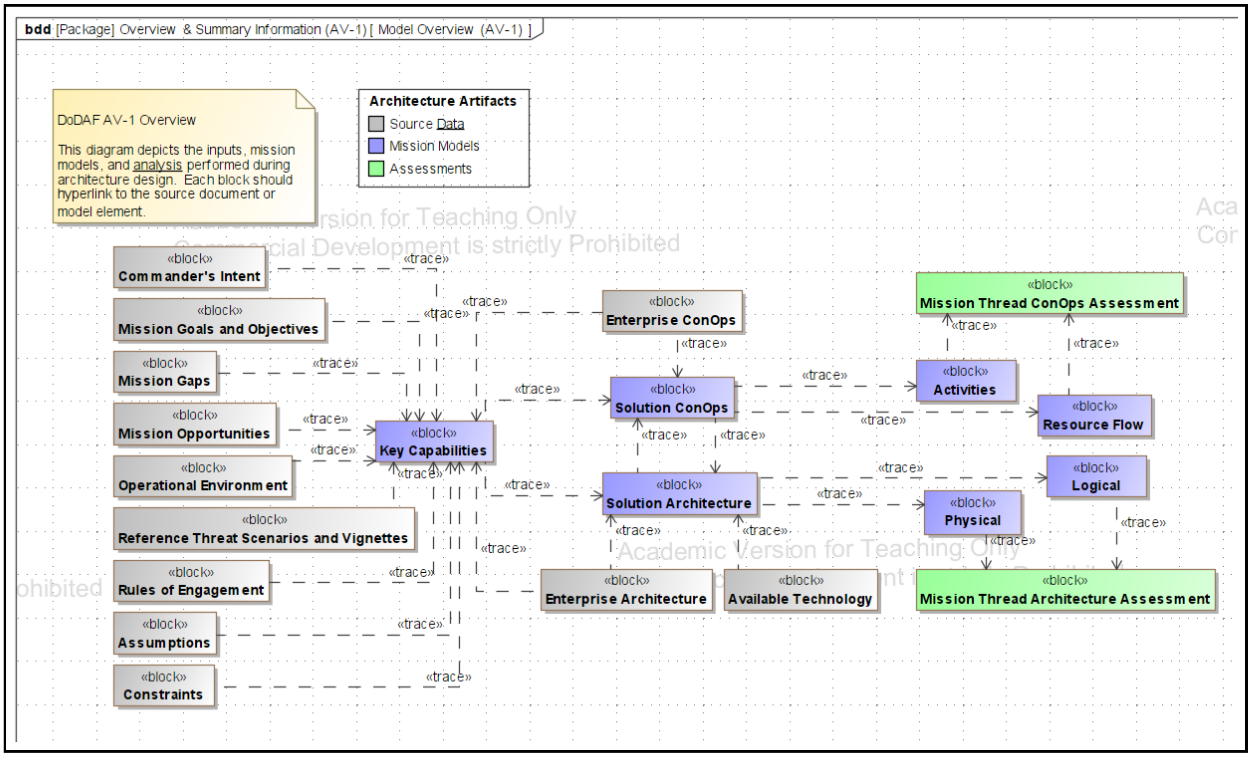

Figure 42: AV-1 Diagram

The use-case diagram in Figure 43 depicts some of the key capabilities of sensing architecture as they relate to the overall goal. The external value function is captured in an Operational Viewpoint (OV)-5b activity diagram, as seen in Figure 44. Any sensing architecture must be able to perform these activities to satisfy the objectives of cislunar SDA. Required capabilities and a mission constraint were derived from the use-case and activity diagrams and were captured in a requirements diagram to graphically depict the relationships using a containment hierarchy (Figure 45).

An additional benefit is the requirements are now in an element form that can be related to other elements, such as requirements, structures, activities, and analyses. All of these

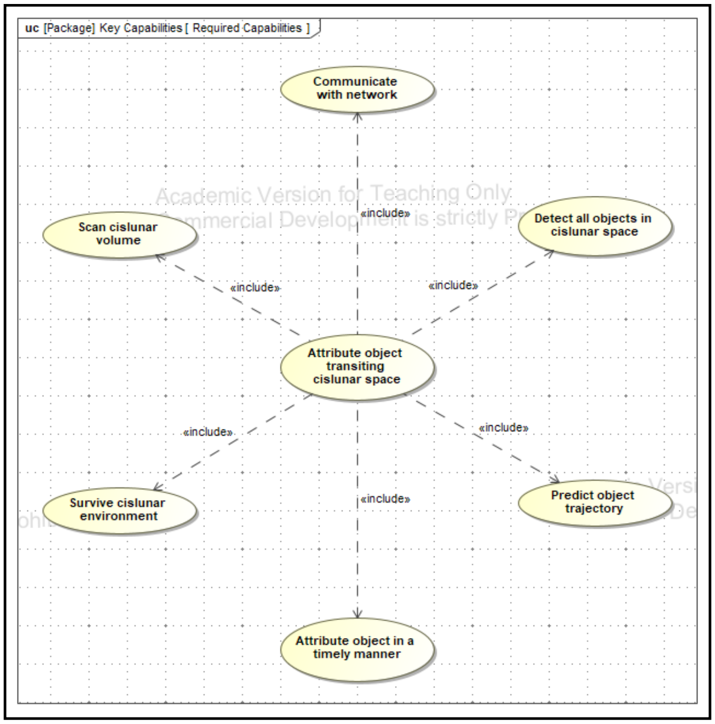

Figure 43: Key Capabilities

elements can related back to these original text-based requirements using derive, satisfy, verify, refine, trace, and copy relationships. An example is in Figure 46 where the required capabilities and design constraint are traced to source documents.

A nominal SDA enterprise architecture was modeled in a System Viewpoint (SV)-1 BDD, as seen in Figure 47. This BDD depicts the entire cislunar SDA domain which is composed of several entities such as space-based and terrestrial sensors, a communications network, and the physical environment that includes objects of interest. Electro-optical (EO) and Synthetic Aperture Radar (SAR) are subclasses that specialize the more general space-based sensor kind. The included value properties are used in the analyses of the space-based EO sensing architecture system-of-interest. Figure 48 depicts these value properties and the object and control flow the DT implements.

This data process is similar to the process developed in Sprint 2 but instantiates the sensing satellites in the STK scenario. After this, a requirements package was created to

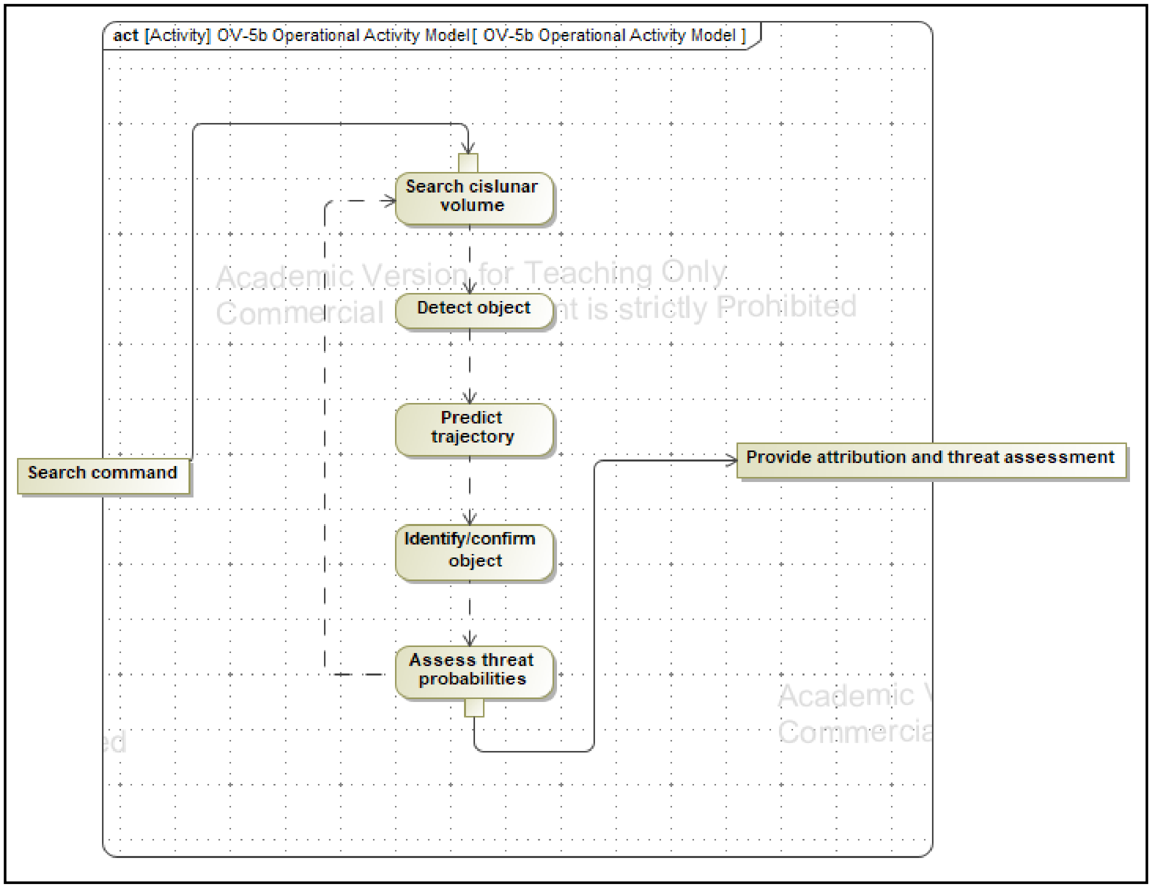

Figure 44: OV-5b Operational Activity Model

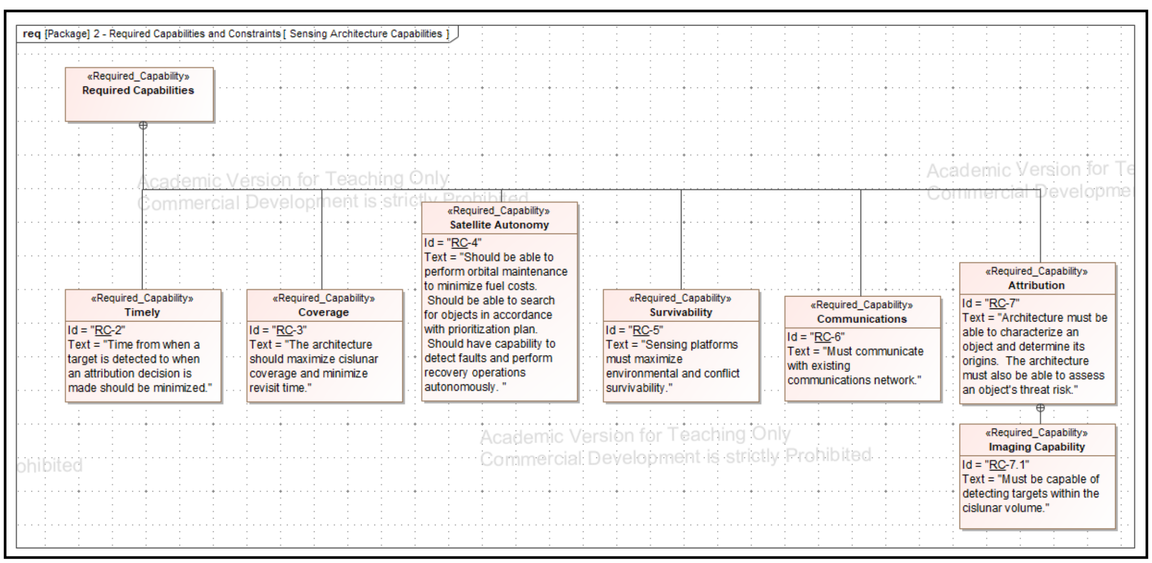

Figure 45: Capabilities as Requirements

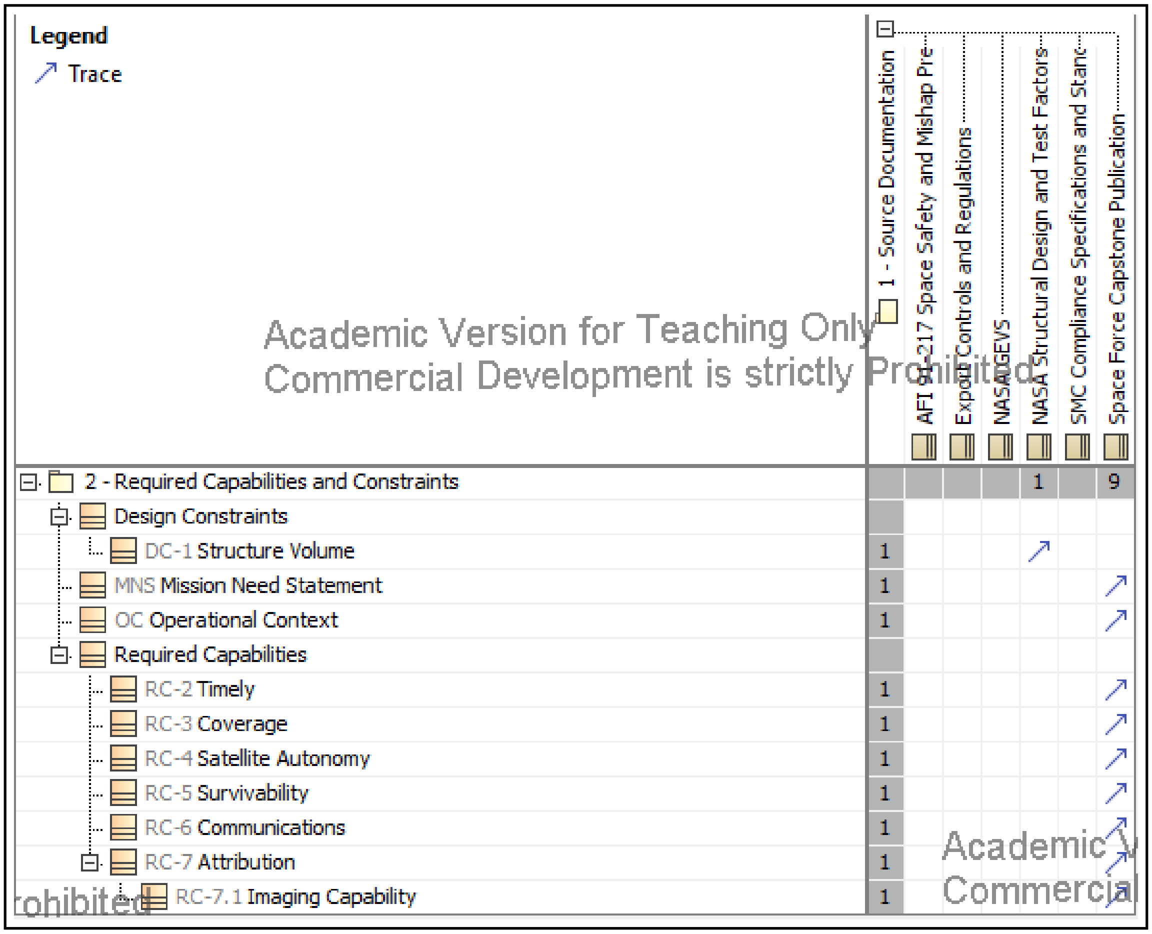

Figure 46: Capabilities and Constraint Traced to Source Documentation

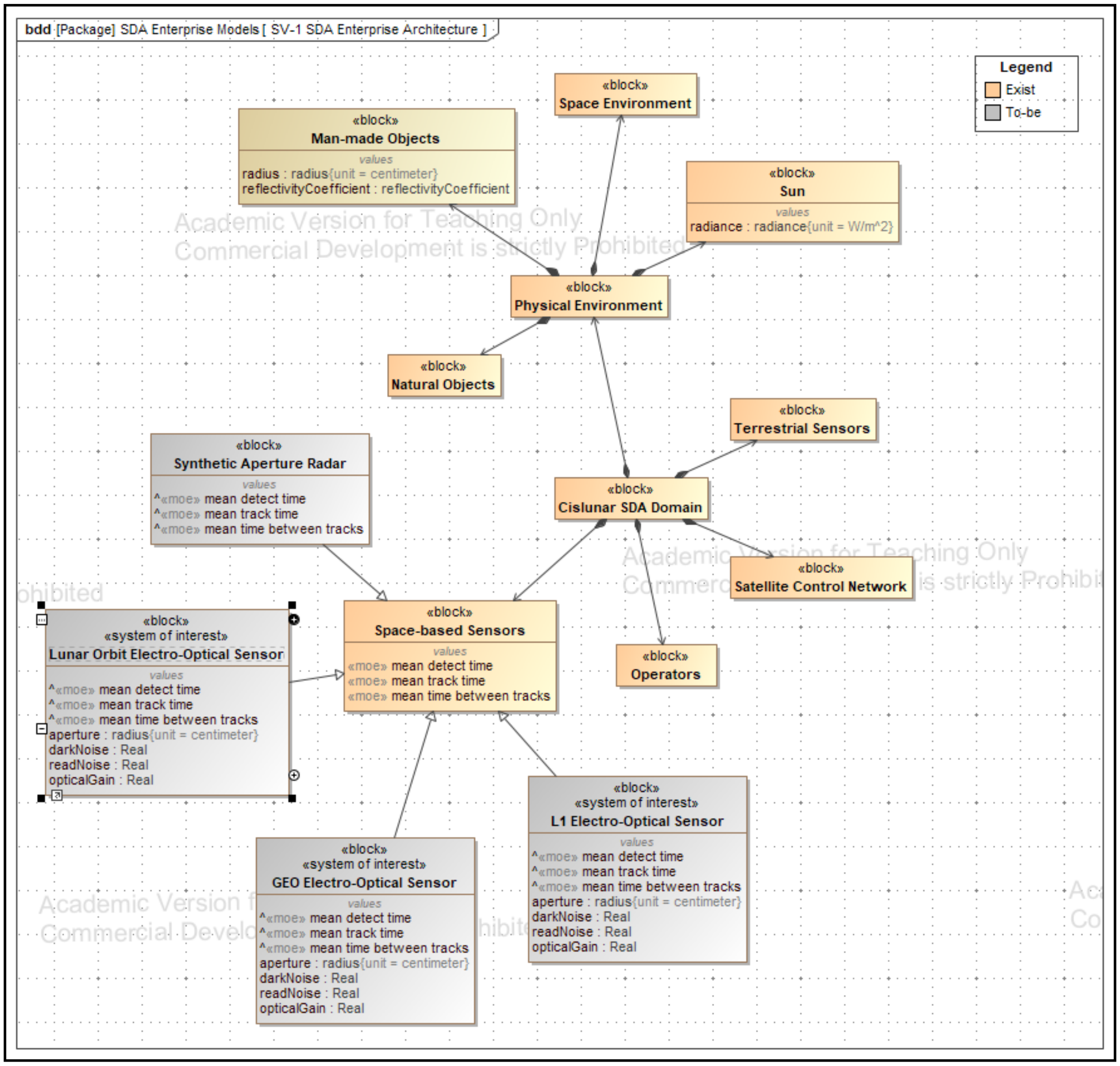

Figure 47: Nominal SDA Enterprise Architecture

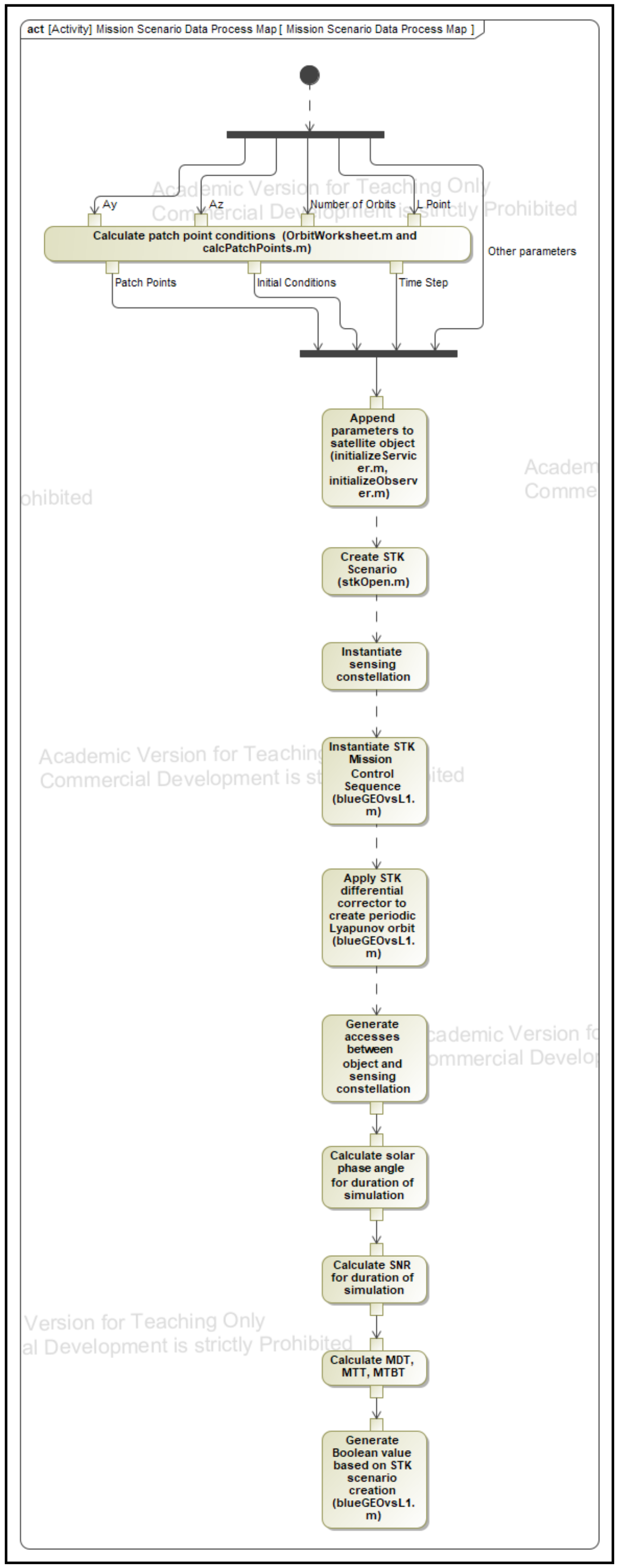

Figure 48: Mission Scenario Data Process Map

depict the relationships between elements in Figure 49, justifying goals, key capabilities, activities, and analysis by linking them to the source document.

Sprint 3 was successful in implementing the ME process with Object-Oriented Systems Engineering Method (OOSEM) activities and methods to capture details relating to the problem statement of needing to detect objects in cislunar space.

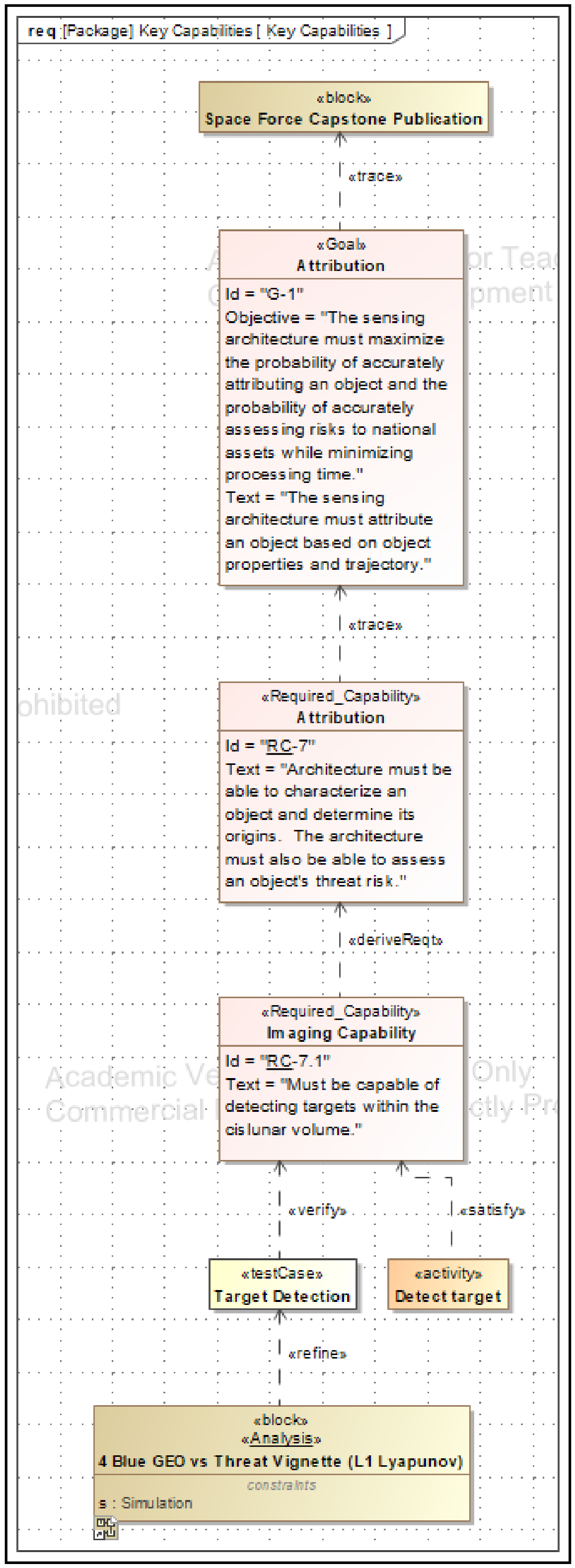

Figure 49: Requirements Trace

4.4 Sprint 4 Review

Sprint 3 saw the creation of a model supporting the early activities of the ME process; Sprint 4 focuses on verifying the DT that is used to evaluate the proposed solution to the mission's problem statement. The data flow was extended from the previous maps to include the additional mathematical models and measures used in the evaluation process, as depicted in Figure 50.

The initial parametric diagram, depicted in a parametric diagram in Figure 51, was built with a singular constraint property that executed the simulation in STK and performed the mathematical routines associated with evaluating the sensing constellation's performance.

While there were a couple of ways to instantiate the satellites with STK, to include explicitly creating a constellation, the parametric diagram created in Sprint 2 was instead modified to incorporate and instantiate three additional, separate satellites which can be seen in Figure 52.

This initial proof-of-concept parametric diagram reduced the risk associated with verifying the DT, though it minimizes the value of the parametric diagram since the data flow and transformations are not visible. This design enabled easier debugging during the scripting process, though it removes the benefit of easily communicating the process as there is a single constraint block instead of multiple constraint blocks with their associated inputs and outputs. Of note, the parametric analysis can only solve expressions in a one-way direction only. The parametric engine will execute the constraint and next constraint properties of the block. If an input constraint parameter of constraint property, A, is connected to the output of another constraint property, B, B will be executed first. This is important to understand because it impacts the integration of Cameo with STK; specifically, it means the evaluation is static rather than dynamic, where the simulation and associated data are accomplished first and then analyzed, rather than being analyzed in near-real time. Possible remedies to this shortcoming are presented in Chapter V. However, a static analysis is still useful since

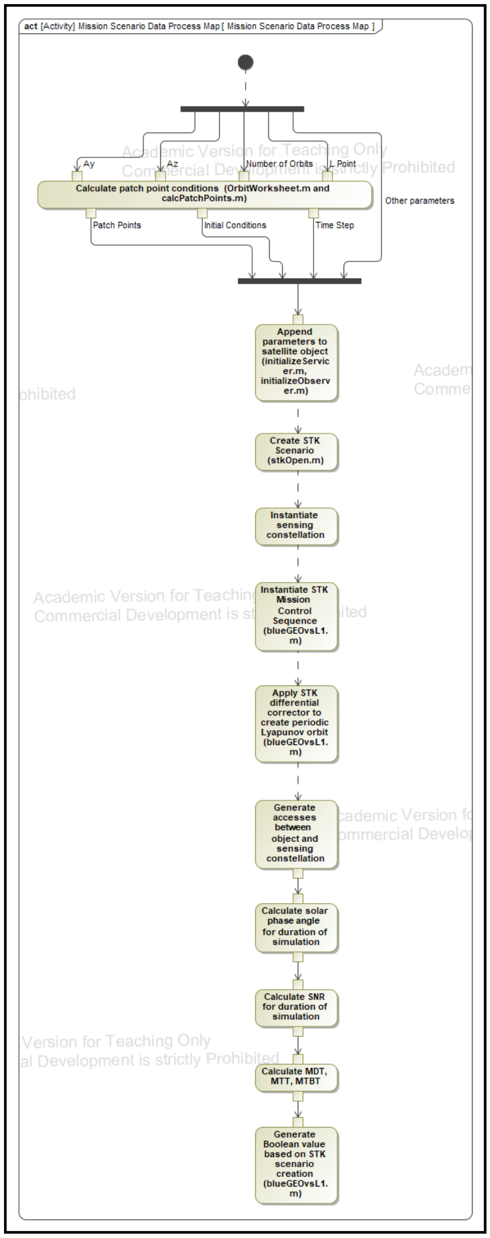

Figure 50: Mission Scenario Evaluation Process Map

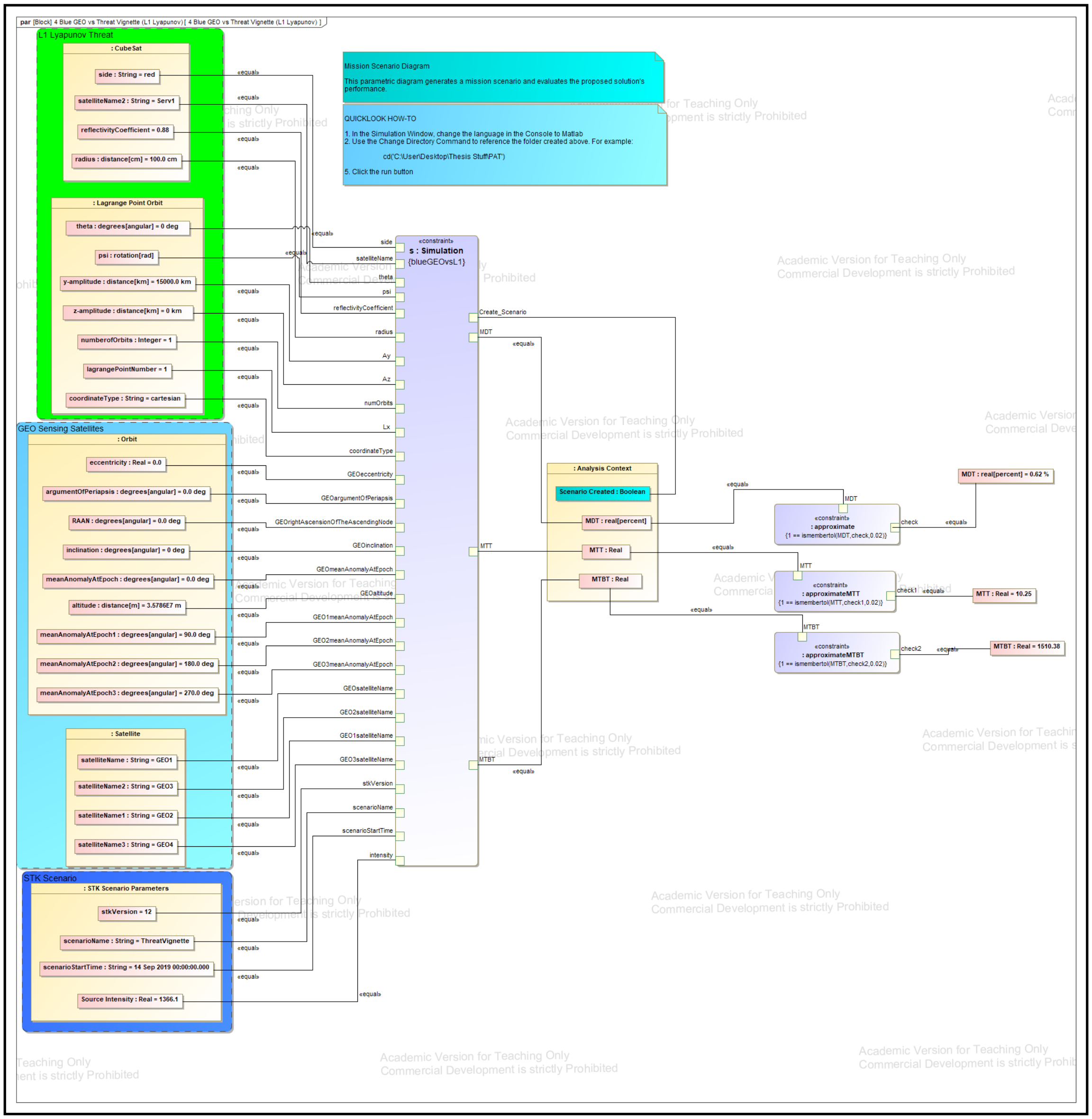

Figure 51: Sensing Constellation Parametric Diagram

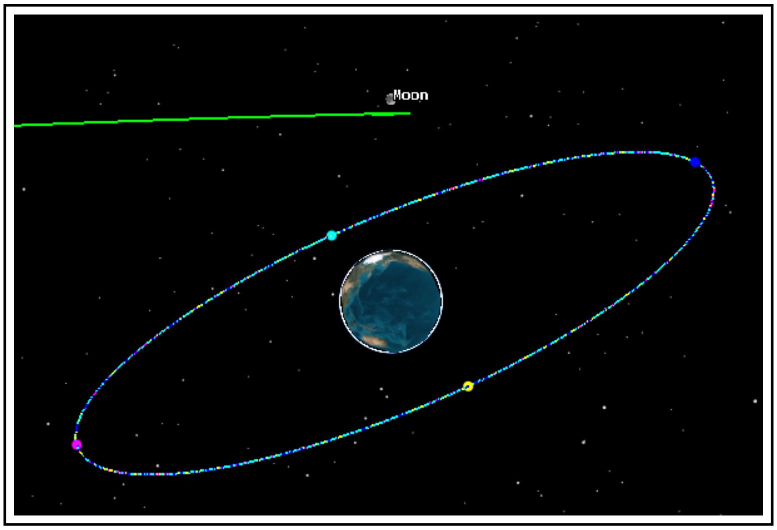

Figure 52: Mission Scenario

this parametric analysis solely focuses on performance rather than behavior. The initial results (Figure 53) from this parametric analysis do not match Knister's results for the same scenario, as seen in Table 6.

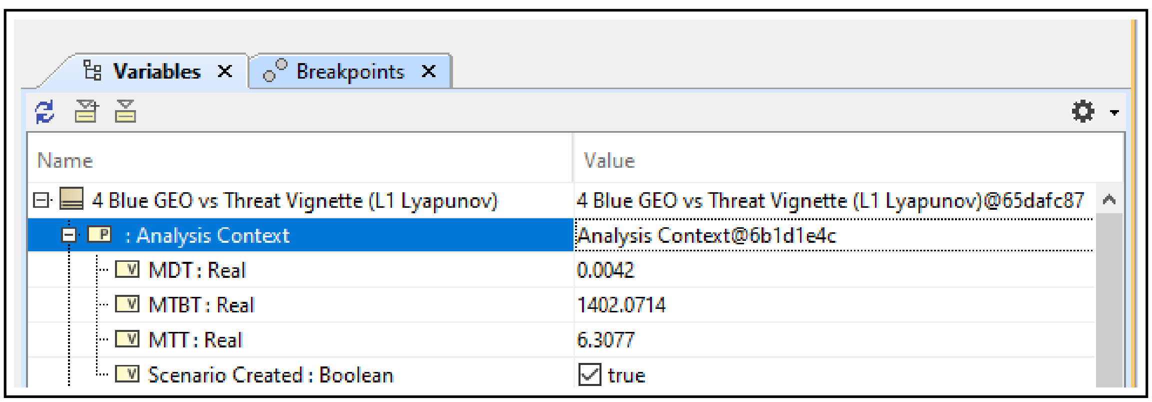

Figure 53: Initial Results

Table 6: Initial Comparison

	Knister	Calculated
MDT	0.55%	0.42%
MTT	12.11	6.31
MTBT	1964.9	1402.07

Therefore, all of the instances from Figure 23 were executed and the calculated measures were compared to the values Knister calculated. If the calculated measures were within tolerance, the cell turns green; if not, the cell turns red. As displayed in Figure 54, none of the 32 trials produced results within a 2% tolerance of Knister's values.

#	Name	.Source Intensity : Real	.lagrangePoint	.theta : degrees[angul (deg)	.y-amplitude : distance[km] (km)	.z-amplitude : distance[km (km)	.altitude : distance[m] (m)	.Scenario Created : Boolean	.MDT : real[percent] (%)	.MTBT : Real	.MTT : Real
1	4 Blue GEO vs Threat Vignette (L1 Lyapunov)	1366.1	L1	0 deg	5000 km	0 km	3.5786E7 m	true	0.1268 %	1968.6	2.7778
2	4 Blue GEO vs Threat Vignette (L1 Lyapunov)1	1366.1	L1	0 deg	5000 km	0 km	4.2164E7 m	true	0.0862 %	2813.4286	2.8333
3	4 Blue GEO vs Threat Vignette (L1 Lyapunov)4	1366.1	L1	0 deg	15000 km	0 km	3.5786E7 m	true	0.416 %	1402.0714	6.3077
4	4 Blue GEO vs Threat Vignette (L1 Lyapunov)5	1366.1	L1	0 deg	15000 km	0 km	4.2164E7 m	true	0.3754 %	1785.1818	7.4
5	4 Blue GEO vs Threat Vignette (L1 Lyapunov)8	1366.1	L1	180 deg	5000 km	0 km	3.5786E7 m	true	0.1268 %	1968.6	2.7778
6	4 Blue GEO vs Threat Vignette (L1 Lyapunov)9	1366.1	L1	180 deg	5000 km	0 km	4.2164E7 m	true	0.0862 %	2813.4286	2.8333
7	4 Blue GEO vs Threat Vignette (L1 Lyapunov)12	1366.1	L1	180 deg	15000 km	0 km	3.5786E7 m	true	0.416 %	1402.0714	6.3077
8	4 Blue GEO vs Threat Vignette (L1 Lyapunov)13	1366.1	L1	180 deg	15000 km	0 km	4.2164E7 m	true	0.3754 %	1785.1818	7.4
9	4 Blue GEO vs Threat Vignette (L1 Lyapunov)16	1600	L1	0 deg	5000 km	0 km	3.5786E7 m	true	0.1268 %	1968.6	2.7778
10	4 Blue GEO vs Threat Vignette (L1 Lyapunov)17	1600	L1	0 deg	5000 km	0 km	4.2164E7 m	true	0.0862 %	2813.4286	2.8333
11	4 Blue GEO vs Threat Vignette (L1 Lyapunov)20	1600	L1	0 deg	15000 km	0 km	3.5786E7 m	true	0.416 %	1402.0714	6.3077
12	4 Blue GEO vs Threat Vignette (L1 Lyapunov)21	1600	L1	0 deg	15000 km	0 km	4.2164E7 m	true	0.3754 %	1785.1818	7.4
13	4 Blue GEO vs Threat Vignette (L1 Lyapunov)24	1600	L1	180 deg	5000 km	0 km	3.5786E7 m	true	0.1268 %	1968.6	2.7778
14	4 Blue GEO vs Threat Vignette (L1 Lyapunov)25	1600	L1	180 deg	5000 km	0 km	4.2164E7 m	true	0.0862 %	2813.4286	2.8333
15	4 Blue GEO vs Threat Vignette (L1 Lyapunov)28	1600	L1	180 deg	15000 km	0 km	3.5786E7 m	true	0.416 %	1402.0714	6.3077
16	4 Blue GEO vs Threat Vignette (L1 Lyapunov)29	1600	L1	180 deg	15000 km	0 km	4.2164E7 m	true	0.3754 %	1785.1818	7.4
17	4 Blue GEO vs Threat Vignette (L1 Lyapunov)2	1366.1	L1	0 deg	5000 km	20000 km	3.5786E7 m	true	0.1725 %	2186.3333	4.25
18	4 Blue GEO vs Threat Vignette (L1 Lyapunov)3	1366.1	L1	0 deg	5000 km	20000 km	4.2164E7 m	true	0.1522 %	2811.5714	5
19	4 Blue GEO vs Threat Vignette (L1 Lyapunov)6	1366.1	L1	0 deg	15000 km	20000 km	3.5786E7 m	true	0.4363 %	1635.4167	7.8182
20	4 Blue GEO vs Threat Vignette (L1 Lyapunov)7	1366.1	L1	0 deg	15000 km	20000 km	4.2164E7 m	true	0.3906 %	2181.5556	9.625
21	4 Blue GEO vs Threat Vignette (L1 Lyapunov)10	1366.1	L1	180 deg	5000 km	20000 km	3.5786E7 m	true	0.1725 %	2186.3333	4.25
22	4 Blue GEO vs Threat Vignette (L1 Lyapunov)11	1366.1	L1	180 deg	5000 km	20000 km	4.2164E7 m	true	0.1522 %	2811.5714	5
23	4 Blue GEO vs Threat Vignette (L1 Lyapunov)14	1366.1	L1	180 deg	15000 km	20000 km	3.5786E7 m	true	0.4363 %	1635.4167	7.8182
24	4 Blue GEO vs Threat Vignette (L1 Lyapunov)15	1366.1	L1	180 deg	15000 km	20000 km	4.2164E7 m	true	0.3906 %	2181.5556	9.625
25	4 Blue GEO vs Threat Vignette (L1 Lyapunov)18	1600	L1	0 deg	5000 km	20000 km	3.5786E7 m	true	0.1725 %	2186.3333	4.25
26	4 Blue GEO vs Threat Vignette (L1 Lyapunov)19	1600	L1	0 deg	5000 km	20000 km	4.2164E7 m	true	0.1522 %	2811.5714	5
27	4 Blue GEO vs Threat Vignette (L1 Lyapunov)22	1600	L1	0 deg	15000 km	20000 km	3.5786E7 m	true	0.4363 %	1635.4167	7.8182
28	4 Blue GEO vs Threat Vignette (L1 Lyapunov)23	1600	L1	0 deg	15000 km	20000 km	4.2164E7 m	true	0.3906 %	2181.5556	9.625
29	4 Blue GEO vs Threat Vignette (L1 Lyapunov)26	1600	L1	180 deg	5000 km	20000 km	3.5786E7 m	true	0.1725 %	2186.3333	4.25
30	4 Blue GEO vs Threat Vignette (L1 Lyapunov)27	1600	L1	180 deg	5000 km	20000 km	4.2164E7 m	true	0.1522 %	2811.5714	5
31	4 Blue GEO vs Threat Vignette (L1 Lyapunov)30	1600	L1	180 deg	15000 km	20000 km	3.5786E7 m	true	0.4363 %	1635.4167	7.8182
32	4 Blue GEO vs Threat Vignette (L1 Lyapunov)31	1600	L1	180 deg	15000 km	20000 km	4.2164E7 m	true	0.3906 %	2181.5556	9.625

Figure 54: Comparative Analysis

The largest Mean Detect Time (MDT) value any of the trials generated was 0.436 % which occurred when the threat object was in a Lissajous orbit and the sensing constellation was at a geosynchronous Earth orbit (GEO) altitude of 35,786 km, irrespective of the source intensity or angular offset values. The largest Mean Track Time (MTT) value generated by the trials was 9.625, which occurred while the threat object was in a Lissajous orbit with the sensing constellation at an altitude of 42,164 km, regardless of the values of the source intensity or angular offset. The nearest Mean Time Between Tracks (MTBT) value generated by the trials was 1968.6, when the sensing constellation was at a GEO altitude of 35,786 km and the threat object was in a Lyapunov orbit with an amplitude, A_y', of 5,000 km, and A_z' of 0 km. Again, as with the previous measures, varying the source intensity

did not affect the value of MTBT. All three of the nearest values came from separate trials, further obfuscating the reason the actual results do not match the expected results.

4.5 Conclusion

Applying an Agile DME methodology led to the creation of a MT RA which reduces ambiguity, enables creativity, and manages complexity and DTs that can simulate threat vignettes and can evaluate a sensing constellation's performance and suitability. However, a verification analysis of the evaluation DT revealed stark differences between previously generated results and the DT results, with no immediate identification of the cause. Chapter 5 will summarize the findings from this research and explain their significance, address the limitations and applicability of the findings, acknowledge difficulties, provide answers to the research questions, describe future work, and provide additional recommendations for future Digital Engineering (DE) efforts.

V. Conclusions

As described in the introduction, the entire trade-space associated with mission and system design to conduct cislunar Space Domain Awareness (SDA) is not fully understood. Additionally, the Air Force Institute of Technology (AFIT) lacks access to a comprehensive digital ecosystem and in particular, Digital Threads (DTs) designed to accelerate research for cislunar SDA. The hypothesis of this research was that a comprehensive Agile Digital Mission Engineering (DME) methodology could produce a foundational DT that can accelerate the creation of high-confidence, decision-making information for cislunar SDA mission and system designs. In this chapter, the findings from this research are summarized and their significance is explained. Additionally, the limitations and applicability of the findings are also addressed and difficulties are acknowledged. Finally, research questions are answered, future work is described, and recommendations are made for future Digital Engineering (DE) efforts.

5.1 Findings

This research found many commonalities between the United States Air Force (USAF)'s Mission Engineering (ME) and traditional Systems Engineering (SE) activities, leading to a framework that enables a team to work through a spectrum of system design phases, buttressed by object-oriented concepts. The ME process aids in defining the problem and context, generating Mission Thread (MT) (or System-of-Systems (SoS)) concepts, and analyzing the problem/solution tradespace to identify key drivers. This is significant because it is a repeatable process that can be initiated by any operational question, which is beneficial since there are still many operational questions concerning cislunar SDA operations. Additionally, this research demonstrated how the ME process can be accomplished digitally and ultimately generate a cislunar SDA Reference Architecture (RA).

This research found a DT can integrate distinct descriptive, analytical, and physics-based digital tools to automate the simulation and analysis of system models. This automation not only saves time but also reduces data transfer errors associated with human manipulation. However, this research provides additional evidence that building a DT requires its own system design process. The implemented scenario is highly complex, leveraging multiple mathematical models and multiple digital tools. The comparative analysis during the DT verification activity highlighted the challenge associated with building this type of virtual environment and the need to implement a design process that verifies the digital system starting at the "component" level. While the DT is functional, there is currently low confidence in the accuracy of the DT due to a lack of a proven, high-confidence reference scenario to compare the generated results to. However, the creation of a cislunar SDA RA and foundational, parametric-based DT within a few months time proves the value of the Agile DME methodology.

5.2 Challenges

While there were many difficulties encountered during the development of the RA and DT, only the most impactful will be discussed. Issues associated with programming and the integration of the digital tools caused the greatest delays in deriving answers to research questions, some of which were also documented by Brown, but which can largely be attributed to a lack of experience with the digital tools.[31] Modifying existing code would often times require several rounds of troubleshooting to address cascading failures. Furthermore, troubleshooting issues in Cameo were sometimes difficult to debug, to include how Cameo handles object-oriented programming variables in Matlab. The result of this issue was the majority of code was written in Matlab script, outside of Cameo, to more easily debug routines.

Attempting to verify the DT was another challenge. Matlab and System Toolkit (STK)

versions differed from the versions Knister used, and the inherited code was not able to be verified as the exact code Knister used to produce the scenario data. While the received version of the code is certainly a late iteration, there were several parts of the code that did not seem to match the descriptions in the thesis or appeared incomplete. However, the code is functional and a more compact version was used for the geosynchronous Earth orbit (GEO)-based sensing constellation evaluation. The veracity of the DT is called into question since the 32 trials did not find a corresponding set of variables that generated matching data. The challenges were not only associated with programming and the overall evaluation process, but also with the descriptive modeling effort.

There were difficulties encountered in reusing model elements in the descriptive tool. Often times, duplicating an element and changing the duplicate would change the original element; or, modifying or deleting the original would modify or delete the duplicate residing in a different package or diagram. While multiples challenges were encountered during this research, they should not detract from the overall demonstration of the value a DT has in simulating and evaluating systems constructed in a more human readable format.

5.3 Limitations and Applicability

The findings of this research are limited to the assumptions, mathematical models, and the digital tools used to evaluate system performance during cislunar SDA operations. The DT simulates a high-altitude, highly eccentric threat using specific digital tools, to include a tool specifically designed to propagate motion adhering to two-body Equations of Motion (EOM)s. It also simulates a threat in a Lyapunov orbit at L_1, along with a GEO-based sensing constellation observing that particular threat. Ultimately, the overarching Agile DME methodology is expected to be applicable to any mission or system design.

5.4 Research Questions

The individual Sprints in this thesis posed more specific research questions to the three broad research questions proposed in the beginning. It is through the collective insights of each of these Sprints that the more broad research questions can be answered. In particular:

1. *What methodology should be used to study the operational domain while adhering to system design principles?* An Agile DME methodology was synthesized from the USAF ME guide, USAF DE transformation documents, and the Object-Oriented Systems Engineering (OOSE) method, and was validated by applying the synthesized process, methods, and tools to cislunar SDA research, resulting in the creation of a cislunar SDA RA and DT which were rapidly assembled to design and evaluate a sensing constellation's performance against a threat vignette.

2. *Can a DT be constructed to develop system alternatives and evaluate their effectiveness?* Yes. A DT was designed and constructed by integrating a descriptive modeling tool, an analytical tool, and a simulation tool, which enabled the creation of modeling artifacts during the ME process, automated simulation and analysis, and calculated metrics that traced to mission goals and capabilities.

3. *Can an RA be created for cislunar SDA mission and system design?* Yes. A cislunar SDA RA was established by modifying an existing cubesat RA. The existing RA already had a majority of the primary requirement, structure, behavior, and analytical elements which were then modified to form a basic RA for the cislunar SDA context.

5.5 Future Work and Recommendations

This research was not performed because it is distinctively novel, since modeling, simulation, analysis, and automation have been a mainstay in engineering for the past couple of decades. Instead, this research was conducted to bolster the body of knowledge resident at

AFIT and to provide additional evidence to the transformative effects of DE in a burgeoning military mission set. Therefore, the future work described below does not necessarily pertain to unexplored academic questions, but is based on the perception of the highest value tasks that could be performed to expand the usefulness of a DT for cislunar SDA research and system design.

5.5.1 Simulation and Analysis Environments

The most important activity is to verify the implementation of the Electro-optical (EO) mathematical model and evaluation process. Perhaps the best approach would be to perform verification activities of the DTs "subsystems", similar to any other system verification process of verifying the performance of lower-level system before integrating them and verifying the higher-level system. This could be accomplished by creating reference scenarios via manual calculations of the process. These reference scenarios could be created based on a Design of Experiments setup. This would accomplish two objectives. The first would be to develop a comprehensive understanding of the process and script, driving a higher confidence level of the DT. The second is this activity would enable more effective, efficient, resilient, and modular code to be written. These modular constraint blocks could then be used to clearly define the mathematical model in the parametric diagram, similar to Figure 55 and 56, that depict a higher-fidelity model of the High-altitude, highly-elliptical (HAHE) threat vignette. However, this introduces another challenge of how independent Matlab scripts manage control of the simulation tool.

These constraint blocks can then be added to a parametric diagram to drive greater visibility to the process and how the model elements interact with each other. However, this will introduce another challenge. The default Cameo execution process is to accomplish constraints based on their variable dependence. What this means in practice is the modeler has no control over how Cameo executes a parametric analysis. An execution configuration

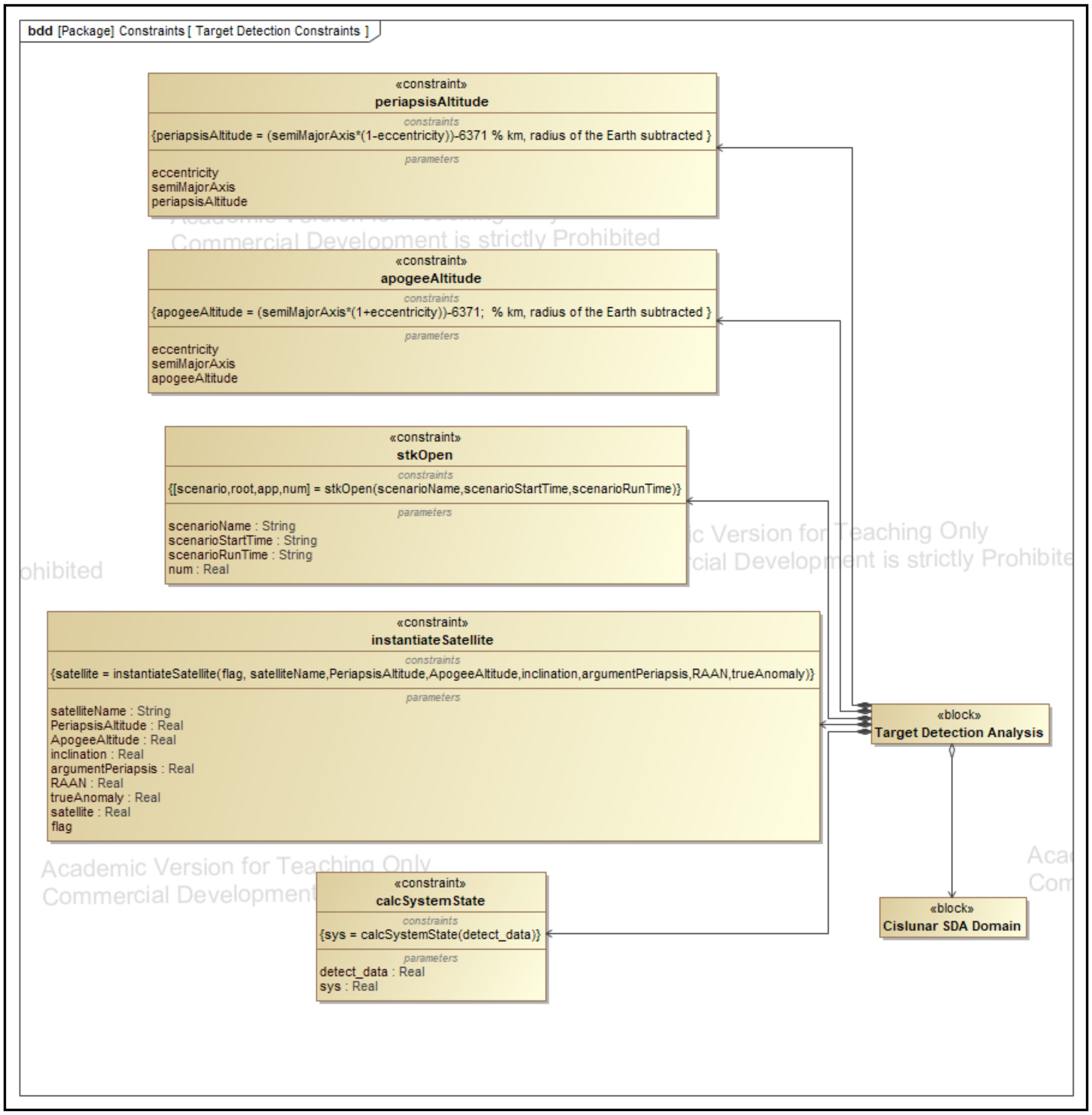

Figure 55: Target Detection Constraints

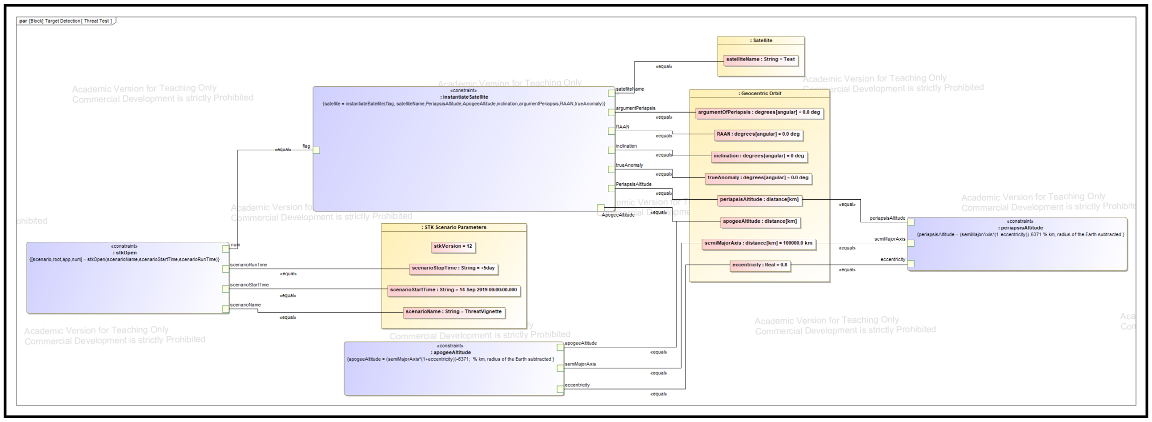

Figure 56: Higher-fidelity Parametric Diagram

should be studied for possible remedies to this challenge, because it could also lead to a breakthrough in creating a dynamic analysis where the system is evaluated at each time step and data is passed back and forth between the simulation and analysis environments. This analysis process can be accelerated by using the STK engine to eliminate the processing demands of the graphical user interface. Analytical Graphics, Inc. (AGI) estimates STK engine can reduce processing times by 70%. This process could be accelerated faster using parallel computing and optimization to simulate and analyze of multiple trials, similar to what Stern and Wachtel accomplished.[18]

Once the DT is verified, additional analysis can be conducted using the same evaluation framework, but pairing sensing concepts against different threat vignettes, like the HAHE threat, or the object at L_1 transiting back to GEO, or a threat in a circumlunar free return trajectory, as seen in Figure 57. This threat vignette was created through a patch-conic routine and modeled in STK using a mission control sequence. It is also already modeled in a parametric diagram, as seen in Figure 58, but the Matlab script needs to be finished in order to integrate the descriptive and simulation tools.

As for enhancing the analysis process, the Measure of Effectiveness (MOE) and Measure of Performance (MOP), to include any Measure of Success (MOS), should be revisited to determine whether there are more applicable measures to use in this analysis. The Find, fix,

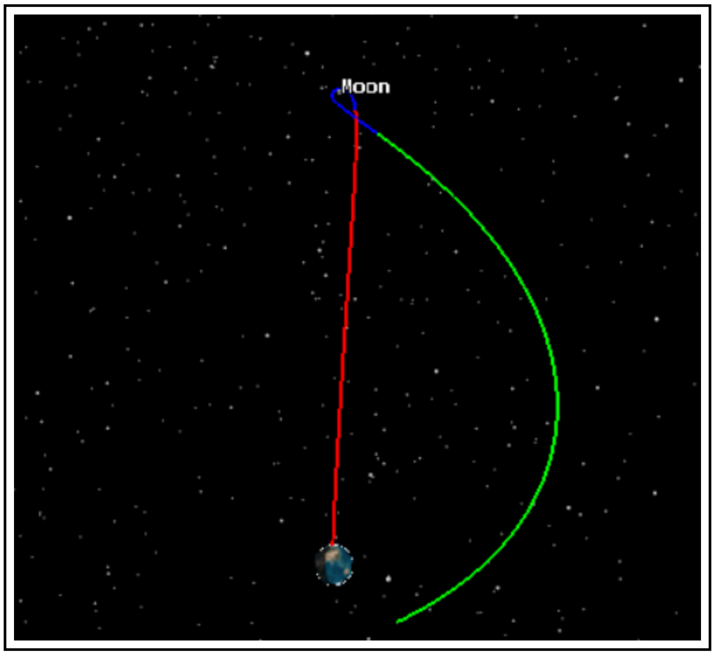

Figure 57: Circumlunar Free Return Trajectory Visualization

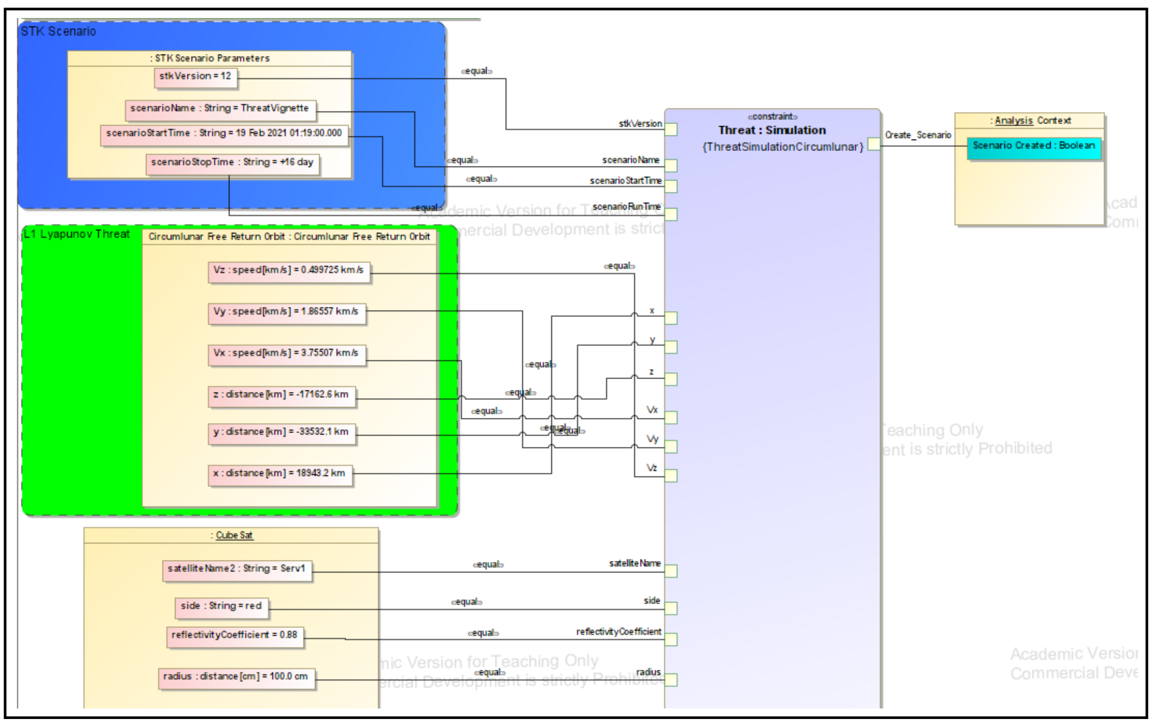

Figure 58: Circumlunar Free Return Parametric Diagram

track, target, engage, assess (F2T2EA) guide could be referenced for additional ideas related to mission satisfaction, cost, coverage, readiness, uncertainty, mission Return-on-Investment (ROI), and probability of accurately classifying objects; perhaps even orbit stability and Orbit Determination (OD).

The collinear L_1, L_2, and L_3 are unstable while L_4 and L_5 are stable in the Circular Restricted Three-Body Problem (CR3BP). This means any small force will cause an object to drift away from one of the collinear libration points. The monodromy matrix characterizes an orbit's stability using rational and irrational eigenvalue pairs. An objects motion around a Lagrange point can be characterized with these eigenvalues in conjunction with the eigenvectors, also known as manifolds. The monodromy matrix can be used during the design and evaluation processes of possible sensing concepts to better understand the effects on OD and station keeping requirements. While detection and stability analysis can assist with sensor locations and sensor requirements, an OD model would capture the ultimate objective for a SDA sensing constellation. Some techniques, such as Gauss' method, require multiple angles for each observation to compute an initial OD, something this analysis does not include.

Other ways to enhance the ME effort and analysis of the mission or system architecture are by conducting a sensitivity analysis and characterizing uncertainty. A sensitivity analysis is useful to identify specific parameters that drive mission effectiveness and system performance. Statistical analysis can provide uncertainties involved in the simulation and evaluation process, providing additional confidence levels for the generated data. Furthermore, statistical analysis could reveal the level of fidelity needed in each of the various models, whether multi-body dynamics should be used or whether additional considerations should be added to the EO model. The statistical analysis could also aid in defining the confidence level associated with each of the individual models, as well as the comprehensive model-of-models. Furthermore, it would be beneficial to investigate other integration oppor-

tunities, leveraging applications like ModelCenter, to understand how that type of capability could mitigate some of the challenges encountered while building the DT.

5.5.2 Modeling Environment

Transitioning the the descriptive model, it seems prudent the mission scenario parametric diagram be anchored to a mission level block, such as the "Mission Enterprise" block in the RA, as opposed to a separate block just for the parametric analysis. The benefit is the ownership of the parametric analysis would be more accurate and the properties of the mission thread would already be present, instead of having to make them additional properties to a different block, which in this thesis was titled, "4 Blue GEO vs Threat Vignette (L1 Lyapunov)". Another way the RA could be made more accurate, with increased detail, is by detailing different Department of Defense Architectural Framework (DoDAF) viewpoints.

Stakeholders and beneficiaries could be captured in an Operational Viewpoint (OV)-4. Needs and goals could be further detailed in a Capability Viewpoint (CV)-1. A solution neutral function could be developed and captured in an OV-1, OV-5b, and OV-6c. A concept's functions could be captured in a logical architecture in System Viewpoint (SV)-1, SV-4, and SV-5a/b activity diagrams and the form could be captured in a physical architecture in OV-2 and OV-3 Block Definition Diagram (BDD)s. Complexity can be managed by tracing these functions and physical parts to mission goals and/or requirements. Additionally, since this research only focused on evaluating the "to-be" sensing architecture, an "as-is" architecture, like the one Bierschbach modeled, could be integrated to accomplish more of the Object-Oriented Systems Engineering Method (OOSEM).[40]

Another opportunity is to use this cislunar SDA RA as a starting point to manage and design a system like Cislunar Highway Patrol System (CHPS). The mission thread exists in limited detail and the simulation code is already present, to include physical system models of

Synthetic Aperture Radar (SAR) or other Electro-magnetic (EM) related sensing technologies that could be used for SDA. Additionally, the threat object could be further enhanced by characterizing its reflectivity in other EM bands and whether it emits any radio frequency energy. These are just some of the many ways to add fidelity or increase the breadth of the DT. While it is conceivable an individual researcher could pursue these tasks, there are several organizational recommendations that could further enhance the development and sustainment of the DT based on the collective body of evidence discussed in the findings, challenges, and future work sections.

5.6 Recommendations

The findings suggest building a DT, a type of digital system, should be accomplished using a system design process. The system design process could leverage the modeling objectives presented in this thesis as initial requirements which could then be allocated to different model types. An analysis of alternatives could be accomplished, followed by the development and verification of the individual models and then the model-of-models. The design process needs to consider and manage the nuances of the tool interfaces and programming languages to ensure the DT is reliable, resilient, and efficient.

The construction of the DT and the operational environment contained aspects of astronautical engineering, systems engineering, computer engineering, operations research, and physics. Based on future work, there are also aspects related to data science, statistics, and networking. Based on these multiple disciplines, it seems prudent to recommend the research and the construction and maintenance of the DT be managed by a multi-functional team, perhaps one facilitated by AFIT's Center for Space Research and Assurance (CSRA).

In general, CSRA could manage research questions posed by mission partners and could organize the research objectives for the multi-functional team. CSRA could also provide the infrastructure, platform, and digital tools/applications for the research team, similar to

what the Naval Systems Engineering Resource Center offers. Alternatively, CSRA could leverage a digital environment hosted by a mission partner. CSRA could also provide access to resident experts for "just in time" learning to prevent the researchers from spending an inordinate amount of time trying to build or maintain aspects of the DT. Additionally, CSRA could manage style guides for different aspects of the research projects to ensure the team is adhering to common design formats, similar to how Application Programming Interfaces (APIs) facilitate easier integration between applications. It is in these ways that CSRA could aid in the development and maintenance of the DT to accelerate innovation and develop a robust digital ecosystem that researchers from many different disciplines could leverage to accelerate research.

5.7 Conclusion

This research synthesized and applied a methodology that was constructed to deliver value quickly, and to provide additional, demonstrable evidence that DE fosters an environment that enhances human readability while automating processes using machine intelligible mathematical models for cislunar SDA mission and system design. The result is a basic RA customized for cislunar SDA and a DT that enables evaluations and assessments of materiel concepts to address SDA needs. While multiple challenges were encountered during this research, many of these challenges can be overcome by using a system design process executed by a multi-functional team with access to a digital environment, "just in time" training resources, and expert help. It is within this framework the future work can be pursued successfully.

Bibliography

1. M. Bolden, T. Craychee, and E. Griggs, "An evaluation of observing constellation orbit stability, low signal-to-noise, and the too-short-arc challenges in the cislunar domain," *Advanced Maui Optical and Space Surveillance Technologies Conference*, 2020. [Online]. Available: www.amostech.com

2. "Space capstone publication: Spacepower doctrine for space forces," 2020.

3. Space Operations Command, "Delta 2 fact sheet," 2020. [Online]. Available: https://www.spoc.spaceforce.mil/About-Us/Fact-Sheets/Display/Article/2334029/space-delta-2

4. "Space policy directive 3," 2018. [Online]. Available: https://www.whitehouse.gov/presidential-actions/space-policy-directive-3-national-space-traffic-management-policy/

5. "Department of defense digital engineering strategy," 2018. [Online]. Available: www.acq.osd.mil/se

6. B. D. Little and C. E. Frueh, "Multiple heterogeneous sensor tasking optimization in the absence of measurement feedback," *The Journal of the Astronautical Sciences*, vol. 67, no. 4, pp. 1678–1707, 2020.

7. S. R. Knister, "Evaluation framework for cislunar space domain awareness (sda) systems," Air Force Institute of Technology, Tech. Rep., 2020. [Online]. Available: https://scholar.afit.edu/etd/3243

8. R. Scheurer, "Mission engineering, digital engineering, mbse, and the like: The one underlying essential attribute." [Online]. Avail-

able: https://www.incose.org/docs/default-source/midwest-gateway/events/incose-mg_2018-11-13_scheurer_presentation.pdf;11/15/2018;Boeing

9. R. Giachetti and A. Hernandez, "Mission engineering." [Online]. Available: https://www.sebokwiki.org/wiki/Mission_Engineering

10. "Joint publication 3-14 space operations," 2018. [Online]. Available: https://www.jcs.mil/Portals/36/Documents/Doctrine/pubs/jp3_14.pdf

11. "Mission engineering guide," 2020. [Online]. Available: https://ac.cto.mil/wp-content/uploads/2020/12/MEG-v40_20201130_shm.pdf

12. D. A. Vallado, *Fundamentals of Astrodynamics and Applications.* Microcosm Press, 2007.

13. J. S. Parker and R. L. Anderson, *Low-energy Lunar Trajectory Design.* John Wiley & Sons, 2014, vol. 12.

14. D. J. Grebow, "Generating periodic orbits in the circular restricted three-body problem with applications to lunar south pole coverage," Purdue University, Tech. Rep., 2006.

15. E. J. Doedel, V. A. Romanov, R. C. Paffenroth, H. B. Keller, D. J. Dichmann, J. Galán-Vioque, and A. Vanderbauwhede, "Elemental periodic orbits associated with the libration points in the circular restricted 3-body problem," *International Journal of Bifurcation and Chaos*, vol. 17, pp. 2625–2677, 2007.

16. D. Llaveria, C. Araguz, A. Camps, and E. Alarcon, "Architecting optimized spaceborne earth observation missions," *2019 IEEE International Conference on Systems, Man and Cybernetics*, 2019.

17. B. Cameron, E. Crawley, and D. Selva, *Systems Architecture. Strategy and product development for complex systems.* Pearson Education, 2016.

18. J. M. Colombi, J. L. Stern, S. T. Wachtel, D. W. Meyer, and R. G. Cobb, "Multi-objective parallel optimization of geosynchronous space situational awareness architectures," *Journal of Spacecraft and Rockets*, vol. 55, pp. 1453–1465, 11 2018.

19. M. S. Rosenof, "Assessing observer stations for cislunar space domain assessing observer stations for cislunar space domain awareness awareness," Air Force Institute of Technology, Tech. Rep., 2020. [Online]. Available: https://scholar.afit.edu/etd/4730

20. G. Haun, "Ufos and physics-integrated executable architectures for digital engineering," 2021.

21. T. McDermott, P. Collopy, M. Nadolski, and C. Paredis, "The future exchange of digital engineering data and models: An enterprise systems analysis," pp. 260–267, 2019.

22. "Digital guide quick start." [Online]. Available: https://usaf.dps.mil/teams/afmcde/SitePages/Digital-Guide-Quick-Start.aspx?csf=1&web=1&e=llx59z

23. S. Friedenthal, R. Steiner, and A. Moore, *A Practical Guide to SysML*. Morgan Kaufmann OMG Press, 2009.

24. L. Delligatti, *SysML distilled: A brief guide to the systems modeling language*. Addison-Wesley, 2013.

25. J. A. Estefan *et al.*, "Survey of model-based systems engineering (mbse) methodologies," *Incose MBSE Focus Group*, vol. 25, no. 8, pp. 1–12, 2007.

26. O. of the Assistant Secretary of Defense Networks and I. I. (OASD/NII), "Reference architecture description," 2010.

27. R. Cloutier, G. Muller, D. Verma, R. Nilchiani, E. Hole, and M. Bone, "The concept of reference architectures," *Systems Engineering*, vol. 13, no. 1, pp. 14–27, 2010.

28. "Department of defense architecture framework version 2.0," 2009.

29. S. Kelly, "A reference architecture for rapid cubesat development," Air Force Institute of Technology, Tech. Rep., 2021.

30. R. Noguchi, R. Minnichelli, and M. Wheaton, "Architecting success in model based systems engineering pilot projects," 2019.

31. K. Brown, "Logic-based mission modeling tool evolution for cubesat payload analysis using model-based systems engineering," Air Force Institute of Technology, Tech. Rep., 2021.

32. "Digital transformation." [Online]. Available: digital.dps.mil

33. "Agile manifesto." [Online]. Available: https://agilemanifesto.org

34. N. Tyagi, "7 types of agile methodologies," Sept 29, 2020. [Online]. Available: https://www.analyticssteps.com/blogs/7-types-agile-methodologies

35. "What is scrum?" [Online]. Available: https://www.scrum.org/resources/what-is-scrum

36. E. M. Shirley, "Application of system engineering leading indicators to scrum agile projects," Air Force Institute of Technology, Tech. Rep., 2016.

37. "Scaled agile framework." [Online]. Available: scaledagileframework.com/case-studies

38. "Mbse guidebook for early concept development." [Online]. Available: https://usaf.dps.mil/

39. V. L. Pisacane, *The Space Environment and its Effects on Space Systems.* American Institute of aeronautics and Astronautics, 2008.

40. C. Bierschbach, "Using model-based systems engineering to describe a notional space domain awareness architecture and its reaction to stimuli," Air Force Institute of Technology, Tech. Rep., 2022.

Acronyms

N_d dark noise. 58

N_e discretized electrons. 54, 58

N_r read noise. 58

AFIT Air Force Institute of Technology. 3, 32, 33, 89, 93, 99

AGI Analytical Graphics, Inc.. 95

API Application Programming Interface. 100

AV All Viewpoint. 52, 74, 75

BDD Block Definition Diagram. 31, 52, 74, 76, 98

CAD Computer Aided Design. 30

CHPS Cislunar Highway Patrol System. 98

COE Classical Orbital Element. 15, 46, 48, 56

CONOPs Concept of Operations. 10, 24, 52

CPU Central Processing Unit. 38

CR3BP Circular Restricted Three-Body Problem. 4, 16, 17, 19, 97

CSM Cameo Systems Modeler. 32, 46

CSRA Center for Space Research and Assurance. 99, 100

CV Capability Viewpoint. 98

DE Digital Engineering. iv, 1, 2, 7, 28, 29, 37, 42, 88, 89, 92, 93, 100, 1

DME Digital Mission Engineering. iv, 3, 5, 43, 61, 62, 88, 89, 90, 91, 92, 1

DoD Department of Defense. iv, 1, 2, 5, 32, 1

DoDAF Department of Defense Architectural Framework. 32, 98

DOTmLPFP Doctrine, Organization, Training, materiel, Leadership, Personnel, Facilities, and Policy. 4, 8, 13, 56

DT Digital Thread. iv, 3, 4, 5, 6, 7, 12, 29, 35, 36, 37, 38, 45, 46, 47, 48, 49, 50, 51, 55, 56, 59, 61, 64, 76, 83, 88, 89, 90, 91, 92, 93, 95, 98, 99, 100, 1

EM Electro-magnetic. 4, 56, 99

EO Electro-optical. 4, 20, 23, 56, 59, 76, 93, 97

EOM Equations of Motion. 4, 15, 16, 19, 33, 70, 91

F2T2EA Find, fix, track, target, engage, assess. 95

FOV Field-of-view. 24

GEO geosynchronous Earth orbit. 1, 22, 23, 25, 26, 27, 28, 34, 38, 51, 55, 59, 87, 91, 95

GPS Global Positioning System. 14

HAHE High-altitude, highly-elliptical. 50, 64, 93, 95

IDE Integrated Development Environment. 38

LEO low Earth orbit. 1, 22, 23, 24, 25, 26, 27, 28, 34

MBSE Model-based Systems Engineering. iv, 3, 7, 29, 30, 31, 32, 35, 41, 52, 1

MDT Mean Detect Time. 14, 53, 87

ME Mission Engineering. iv, 5, 7, 8, 11, 35, 39, 41, 42, 43, 45, 46, 47, 48, 49, 50, 51, 52, 54, 55, 56, 60, 62, 64, 70, 81, 83, 89, 92, 97, 1

MET Mission Engineering Thread. 12, 35

MOE Measure of Effectiveness. 10, 11, 53, 54, 58, 59, 95

MOP Measure of Performance. 10, 11, 53, 54, 58, 59, 95

MOS Measure of Success. 10, 11, 52, 95

MS&A Modeling, Simulation, and Analysis. iv

MT Mission Thread. 5, 12, 35, 46, 52, 63, 88, 89

MTBT Mean Time Between Tracks. 14, 53, 54, 87, 88

MTT Mean Track Time. 14, 53, 87

OD Orbit Determination. 14, 15, 20, 22, 24, 53, 97

OOSE Object-Oriented Systems Engineering. 41, 64, 92

OOSEM Object-Oriented Systems Engineering Method. 6, 30, 31, 32, 35, 43, 46, 52, 81, 98

OV Operational Viewpoint. 52, 75, 98

PAT Payload Analysis Tool. 36, 37

PO Product Owner. 40

RA Reference Architecture. iv, 3, 4, 5, 13, 28, 29, 31, 32, 43, 46, 61, 62, 88, 89, 90, 92, 98, 100, 1

RADAR radio detection and ranging. 20

ROI Return-on-Investment. 97

SAFe Scaled Agile Framework. 41

SAR Synthetic Aperture Radar. 76, 99

SDA Space Domain Awareness. iv, 1, 3, 4, 5, 7, 8, 11, 12, 14, 20, 21, 23, 24, 28, 34, 35, 39, 43, 46, 52, 53, 55, 61, 62, 63, 64, 74, 75, 76, 89, 90, 91, 92, 93, 97, 98, 99, 100, 1

SE Systems Engineering. 7, 29, 30, 32, 39, 41, 42, 43, 89

SNR Signal-to-Noise Ratio. 14, 15, 23, 25, 27, 34, 53, 54, 58

SOI sphere-of-influence. 15, 46

SoS System-of-Systems. 7, 8, 38, 41, 89

STK System Toolkit. 33, 34, 38, 45, 47, 48, 49, 50, 51, 59, 64, 70, 71, 76, 83, 90, 95

STM State Transition Matrix. 17

SV System Viewpoint. 52, 76, 98

SysML Systems Modeling Language. 31, 32, 35

USAF United States Air Force. 89, 92

USSF United States Space Force. 1

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
24–03–2022	Master's Thesis	Sept 2020 — Mar 2022

4. TITLE AND SUBTITLE

APPLIED AGILE DIGITAL MISSION ENGINEERING FOR CISLUNAR SPACE DOMAIN AWARENESS

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Benjamin Williams, Maj, USSF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

AFIT-ENY-MS-22-M317

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In the backdrop of an expansion into cislunar space and a digital transformation, the author synthesizes a methodology from agile system development, DE, ME, and MBSE processes, methods, and tools to develop a basic RA and DT for cislunar SDA mission and system design. The Agile DME methodology is used to conduct basic requirements analysis, develop a concept, understand cislunar physics, study scenario geometries, and perform analyses and the DT implements, executes, and accelerates research by integrating a descriptive tool with analytical and simulation tools. This research demonstrates the value of automated Modeling, Simulation, and Analysis (MSA) workflows and how DE can aid in meeting the DoD's vision to prioritize speed of delivery within rapidly changing operational environments, on limited budgets, and in short timelines using a model-analyze-build methodology.

15. SUBJECT TERMS

Agile system development, digital engineering (DE), mission engineering (ME), model-based systems engineering (MBSE), digital thread (DT), reference architecture (RA), cislunar space domain awareness (SDA)

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	UU	133	Lt Col Bryan Little, AFIT/ENY
					19b. TELEPHONE NUMBER *(include area code)* (937) 255-6565 x4901; bryan.little@afit.edu