



**SPECIFYING SPACE DEFENSE OPERATOR INTERFACES THROUGH THE
APPLICATION OF COGNITIVE SYSTEMS ENGINEERING AND
PROTOTYPING**

THESIS

Justin E. Oryschak, Captain, USAF

AFIT-ENV-MS-20-D-070

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENV-MS-20-D-070

SPECIFYING SPACE DEFENSE OPERATOR INTERFACES THROUGH THE
APPLICATION OF COGNITIVE SYSTEMS ENGINEERING AND PROTOTYPING

THESIS

Presented to the Faculty

Department of Systems Engineering and Management

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 Systems Engineering

Justin E. Oryschak, BS

Captain, USAF

October 2020

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT-ENV-MS-20-D-070

SPECIFYING SPACE DEFENSE OPERATOR INTERFACES THROUGH THE
APPLICATION OF COGNITIVE SYSTEMS ENGINEERING AND PROTOTYPING

Justin E. Oryschak, BS

Captain, USAF

Committee Membership:

Dr. Michael E. Miller
Chair

Dr. John M. McGuirl
Member

Dr. Richard G. Cobb
Member

Dr. Thomas C. Ford
Member

Abstract

The Department of Defense needs better tools to support its operators as they strive to defend its space assets. The growing sophistication of anti-satellite weapons increasingly challenges the nation's orbital communications and surveillance infrastructure. Operators face difficulties gathering useful information and dealing with the complexity of potential enemy actions. This research applied cognitive systems engineering and ecological interface design (EID) methodologies to create a prototype space mission management tool that enhances operator situation awareness and decision-making ability. Applied cognitive task analysis interviews were used to document space operator decision-making in their domain. Model-based systems engineering was applied to integrate work domain concepts into system models. EID methods were applied to inform user interface designs that support high-level decision making in addition to low-level tasks. User interface concepts were developed using rapid prototyping software, Axure 9.0, to satisfy the system requirements. The software prototypes were shown to space operators and assessed for validity. This process demonstrated how cognitive systems engineering can be used to derive system requirements and create system designs, the elements of which can be captured in a systems model and traced to operator goals, resulting in systems that are more capable of supporting operator needs in challenging environments.

To my wife, for her endless love, support, and patience

Acknowledgments

My utmost appreciation goes to Dr. Michael Miller, without whom I would not have been able to find a challenging and rewarding research topic to replace one that was irreparably disrupted by the coronavirus pandemic. I extend the same appreciation Dr. John McGuirl, Dr. Richard Cobb, and Dr. Thomas Ford for sharing with me their expertise and support. Special thanks to Captain Foster Davis for his assistance and knowledge as he worked on his own space operations research.

Justin E. Oryschak

Table of Contents

	Page
Abstract	1
Acknowledgments.....	iii
Table of Contents	iv
List of Figures	vii
List of Tables	x
I. Introduction	1
General Issue	1
Problem Statement.....	2
Research Objectives/Questions/Hypotheses	3
Research Focus.....	3
Investigative Questions	3
Methodology.....	4
Assumptions/Limitations.....	5
Implications	7
Preview	7
II. Literature Review	9
Chapter Overview.....	9
Defense in the Orbital Environment.....	10
Cognitive Systems Engineering and Work Analysis.....	16
Ecological Interface Design	19
Model-Based Systems Engineering.....	22
User Experience Survey	24
Summary.....	25

III. Using Cognitive and Model-Based Systems Engineering Methods for User Interface Design	27
Chapter Overview.....	27
Methodology.....	27
Results	34
Discussion and Conclusions	62
Chapter Summary	64
IV. User Interface Design Prototyping with Feedback into Cognitive System Models ...	65
Chapter Overview.....	65
Methodology.....	65
Results	72
Discussion and Conclusions	96
Chapter Summary	98
V. Conclusions and Recommendations	100
Chapter Overview.....	100
Conclusions of Research	100
Recommendations for Action.....	103
Recommendations for Future Research.....	105
Significance of Research	107
Summary.....	108
Appendix A – Applied Cognitive Task Analysis Interview Template	109
Appendix B – User Experience Survey	114
Prompt	114
Scenario Execution.....	114

Mission Card	115
Timeline.....	115
Probing Questions	117
Bibliography	118

List of Figures

	Page
Figure 1 Relationships of Systems Engineering Methods	10
Figure 2 Depiction of the Radial/In-Track/Cross-Track Coordinate Frame.....	13
Figure 3 Depiction of the Geocentric Inertial Coordinate Frame	15
Figure 4 Cycle Diagram for Iterative Cognitive System Development.....	24
Figure 5 Functional Abstraction Hierarchy for Orbital Defense Depicted in Cameo Systems Modeler 19.0.....	31
Figure 6 Custom Profile Extending SysML for CWA Studies.....	32
Figure 7 Example Decomposition of a Top-Level Goal into Information/Decision Requirements Met by a User Interface Implementation	33
Figure 8 Code System for Transcribed Operator Interviews from MAXQDA	34
Figure 9 Functional Abstraction Hierarchy with Exclusions Marked in Red.....	40
Figure 10 Relationship Complexity of Concrete Domain Functions.....	41
Figure 11 Relationship Complexity of Abstract Domain Functions	41
Figure 12 Domain Purpose and Domain Values Levels of Functional Abstraction Hierarchy	43
Figure 13 Domain Values and Domain Functions Levels of Functional Abstraction Hierarchy	44
Figure 14 Domain Functions and Physical Functions Levels of Functional Abstraction Hierarchy	45
Figure 15 Physical Functions and Physical Objects Levels of Functional Abstraction Hierarchy	46

Figure 16 Course of Action (COA) Decision Requirement.....	47
Figure 17 Payload Performance Decision Requirement (Left).....	48
Figure 18 Payload Performance Decision Requirement (Right)	49
Figure 19 Command Decision Requirement.....	50
Figure 20 Communication with Friendlies Decision Requirement	51
Figure 21 Intent Decision Requirement (Left).....	52
Figure 22 Intent Decision Requirement (Right)	53
Figure 23 Threat Evaluation Decision Requirement	54
Figure 24 Resource Management Decision Requirement	55
Figure 25 Mission Plan Decision Requirement	56
Figure 26 Orbital Environment Decision Requirement (Left).....	57
Figure 27 Orbital Environment Decision Requirement (Right)	58
Figure 28 Active/Passive Defense Decision Requirement	59
Figure 29 Safety Decision Requirement	60
Figure 30 Maneuver Decision Requirement (Left).....	61
Figure 31 Maneuver Decision Requirement (Right)	62
Figure 32 Screenshot of Status UI Prototype Development Using Axure RP 9.....	68
Figure 33 Expanded Accordion Panel	69
Figure 34 Defining User Interaction Logic in Axure RP 9.....	70
Figure 35 Maneuver Selection from Goal Criteria	73
Figure 36 Projected Changes to Mission Constraints	73
Figure 37 Axure RP 9 Scenario Operating in Firefox Browser 81.0.....	75
Figure 38 Bus Status Window	76

Figure 39 Command and Control Status Window	77
Figure 40 Engagement/Vulnerability Zone Display	79
Figure 41 Goals Panel.....	81
Figure 42 Intelligence Panel	82
Figure 43 Maneuver Planning Panel.....	84
Figure 44 Orbit Status Display	85
Figure 45 Payload Status Window.....	86
Figure 46 Situation Awareness Timeline.....	88
Figure 47 Situation Awareness Timeline (Continued)	89
Figure 48 Spec Sheet Display	90
Figure 49 Visualization Panel	91
Figure 50 Visualization Panel (Continued).....	92
Figure 51 Visualization Panel (Continued).....	93
Figure 52 Cognitive System Development within an Iterative Development Process...	105
Figure 53 Scenario Mission Card Presented to Participants	115

List of Tables

	Page
Table 1 Mapping of Interview Content to Domain Model	36
Table 2 Abstraction Level and Elements of Functional Abstraction Hierarchy	38
Table 3 Comparison of User Interface Prototyping Software	67
Table 4 Suggested Interface Improvements.....	95

SPECIFYING SPACE DEFENSE OPERATOR INTERFACES THROUGH THE APPLICATION OF COGNITIVE SYSTEMS ENGINEERING AND PROTOTYPING

I. Introduction

General Issue

The Department of Defense (DoD) needs better tools to support its operators as they strive to defend its space assets. The growing sophistication of physical and non-physical anti-satellite weapons poses an increasing challenge to the nation's orbital communications and surveillance infrastructure. Operators must be capable of quickly recognizing unprecedented threats in a heretofore uncontested domain to respond appropriately while carefully managing their satellite's limited resources and accomplishing their missions. Traditionally, the DoD has trained operators to perform routine satellite maintenance activities. However, as they operated their satellites in an uncontested domain, they have not developed the expertise necessary to respond to time-critical emerging threats (Ziarnick, 2018, p. 11). Building expertise in space combat tactics to deter aggression and maintain the nation's space superiority has recently been recognized as the Air Force Space Command's top priority (Raymond, 2018, pp. 5–6). To place greater emphasis on this mission, this responsibility has recently become the purview of a new military branch, the United States Space Force. Research is needed to study the challenge of defending space assets in a contested environment and to produce

new mission planning tools to support fast, effective responses to both expected and unforeseen threats.

Operators face great difficulties in gathering useful information and dealing with the complexity of potential enemy actions. The US has finite capability to track space objects, burdening operators with interpreting incomplete and irregular observations of adversarial satellites to identify changes in satellite behavior that could indicate a hostile intent (Ewart, 2018, p. 88). There are a variety of ways hostile satellites could threaten US assets, including kinetic and laser attacks, as well as more subtle approaches, including communications jamming and cyber-attacks (National Air and Space Intelligence Center, 2018, p. 15). These activities may be presaged by close approaches and proximity maneuvers, as demonstrated by a Russian military satellite launched in 2015 (Gruss, 2015). Operators must carefully consider the maneuvers they use to mitigate damage or thwart surveillance operations, as fuel is limited and irreplaceable. Domain literature elaborates on satellite defense strategies, the information and decision requirements necessary to execute them (Hanlon & Yakimenko, 2019), and how ecological interface design concepts for mission planning tools can assist operators with interpreting data and responding effectively (Vicente & Rasmussen, 1992).

Problem Statement

There is a need for space operators to recognize hostile actions and carefully manage their assets to deter surveillance and attacks. The complexity of maneuvering in orbit, the range of possible hostile activities, and the limited amount of information available to operators make it difficult to respond effectively and quickly. Without

adequate tools to support situation awareness and mission planning, operators cannot make informed decisions to defend their satellites.

Research Objectives/Questions/Hypotheses

This thesis seeks to provide insights towards answering the following question. How can ecological interface design methods be applied to create improved tools for space operators to identify, respond to, and deter hostile actions against U.S. assets in orbit?

Research Focus

The application of apply cognitive systems engineering and ecological interface design methods to create a prototype space mission management tool will enhance operator situation awareness and improve decision making ability. The system will be limited to a representation of two spacecraft in geostationary orbit to focus on building a basic foundational interface. The improved system will address difficulties operators have with making timely decisions in a challenging environment and inform design development for other DoD space mission planning systems.

Investigative Questions

- How can cognitive systems engineering be used to obtain the information and decision requirements for defending satellites?
- How can cognitive task analysis identify how operational context changes the ways operators interpret information and make decisions?

- What are the appropriate data representations that permit experienced operators to better understand the threats and available options to counter these threats?
- How can EID make mission planning challenges tractable?
- How can the knowledge of operators' information needs be captured, traced, and modeled in Model-Based Systems Engineering (MBSE) tools to support the derivation of system and training requirements?
- How can the operational impacts of EID be measured?

Methodology

This study will employ cognitive systems engineering methods to study the work that space operators perform. Cognitive task analysis (CTA) describes how operators perform specific tasks and models their decision-making process, the difference between novice and expert performance, and how mental workload changes throughout task performance. Cognitive Work Analysis (CWA) is a multi-stage framework that employs various methods used in CTA but focuses on how the operational environment affects and constrains the work conducted within it in order to specify the requirements for human-system integration (Vicente, 1999, pp. 114–119). Operator task performance will be measured through observations of operator interaction with prototype interface designs. Interview methods will be applied to assess the performance of the new designs. The artifacts created during the application of cognitive systems engineering methods will be used to generate requirements documents, diagrams, and models to serve as example inputs for operational space mission planning tool designs.

Research Tasks.

- Collect information on operator tasks through observation, interviews, and studies of system, training, and certification documentation
- Model tasks using the Systems Modeling Language (SysML) with cognitive systems engineering extensions
- Collect and analyze information on operator and workplace goals using subject matter expert interviews to learn how they create, change, and assess progress towards operational goals
- Model the functionality and goals of the space operations work domain, and the relationships between domain concepts, using a Functional Abstraction Hierarchy (FAH).
- Derive system information and decision requirements
- Build software UI implementations and show traceability requirements in the system model
- Present a realistic software simulation to operators and attain feedback on the UI and workflows

Assumptions/Limitations

1. The AFIT space warfare simulation tool suite (ProxBox and Maser) is an assumed accurate functional model of space operations. The developers' goal has been to create a tool suite that resembles the bespoke tools designed for specific space systems in various operational locations. Operators trained using this tool suite should find the skills transferrable to their work in these operational

locations. Accordingly, improvements to the user experience for the tool suite identified by this research will be suitable for a variety of types of space operations.

2. Existing cognitive systems engineering and EID methods will be applicable to DoD space operations. These methods will accurately describe operator performance and the impact of user experience design features through surveys, interviews, observation, and documentation. Previous studies have applied these methods to civilian Unmanned Aerial Vehicle (UAV) operations, which anecdotal evidence suggests are similar to DoD space operations. In UAV operations, one or more sensor operators command their sensors to look at targets of interest and optimize collection time between limited amounts of sensor and time resources. Space assets operate similarly: remote operators must ensure that they apply a limited number of sensors to collect on targets while spending the limited resources in the most optimal manner. The modeling methods used to describe these goals and constraints will be tailored to include parts of the DoD's mission set, such as military hierarchies and behaviors during combat operations.
3. The sample size for the user feedback survey is expected to be small (around 4 participants) due to lack of in-person availability so the survey results will not represent a broad spectrum of space operations experience. The individual experiences of the participants may bias their responses, such as if one has had more experience in leadership roles and another has spent more time operating payloads. Because the fidelity of the UI prototype will be low, participants will not be able to interact with it directly. This will limit the ability for the researcher

to identify novel interactions or problematic aspects of the UI that are not elicited with the probing questions. There will be no scored, standardized survey questions so the evaluation of the participant's responses will be left to the judgment of the researcher.

Implications

This research will study the application of cognitive systems engineering, MBSE, and ecological interface design methods to a space operational environment. Tailoring the methods for DoD operations will allow future researchers to apply them to other mission sets, enhancing their descriptive ability and contributing to a common language for DoD space operations designs. Using cognitive systems engineering methods to inform the MBSE design process demonstrates how operator goals can be captured early in the requirements development process and used to inform the design of a complex system. This design process may be used to emphasize the role of human systems integration in maximizing system performance for other areas of DoD space operations and other operational domains, such as the defense of remote-controlled or artificially intelligent aircraft. These domains share similar traits to space asset defense, with operations managers evaluating multiple types of data inputs and optimizing mission resource application.

Preview

Chapter I introduced the general issue facing space operators tasked with defending their satellites, defined a problem with a lack of robust situation awareness and mission planning tools, proposed research on their work domain using cognitive systems

engineering methods and ecological interface design to improve their tools, and described the methodology, limitations, and implications of the research process. Chapter II summarizes the academic discussion on the cognitive challenges facing computer system operators in dynamic work domains, the role ecological interface design plays in creating effective software user interfaces, and important concepts for the conduct of offense and defense operations in the orbital environment. Chapter III details the methodology for conducting domain research and performing ACTA interviews with space operators to document the goals, functions, and requirements for defending an asset in space. These concepts are used to create a model of the work domain by applying and extending the Systems Modeling Language (SysML), which aids the specification and creation of user interface prototypes for operator evaluation. Chapter IV describes how the prototype space operator UI was created and presented to operators as a functioning scenario for feedback. Chapter V concludes with an overall assessment of the research process and suggestions for implementing the results and performing future research with cognitive systems engineering and ecological interface design in the space domain. Relatively little work has been done to adapt engineering tools for space mission planning to meet the needs of operators in a rapidly evolving operational environment. Operators need better tools to manage and defend a growing variety of space vehicles meeting the tactical and strategic needs of the Department of Defense.

II. Literature Review

Chapter Overview

This literature review summarizes academic perspectives on the concepts used and built upon in this research. When defending space assets, operators contend with problems with solutions that are unintuitive and dissimilar to terrestrial warfighting domains. Cognitive systems engineering and work analysis methods allow systems engineers to translate operator goals and behavior into design requirements for new systems. Ecological interface design uses these goal- and behavior-based requirements to design system interfaces that permit operators to act on representations of their work domain, improving their ability to solve complex and novel problems. The relationships between these topics and traditional systems engineering are shown in Figure 1. The Systems Modeling Language (SysML), which is implemented in tools such as Cameo Systems Modeler, allow systems engineers to explore system designs that trace to operator needs. Rapid prototyping and user experience surveys allow users to provide feedback on potential interface designs prior to system deployment. The literature summary of these topics provides background information on the research problem and the methods used to address it.

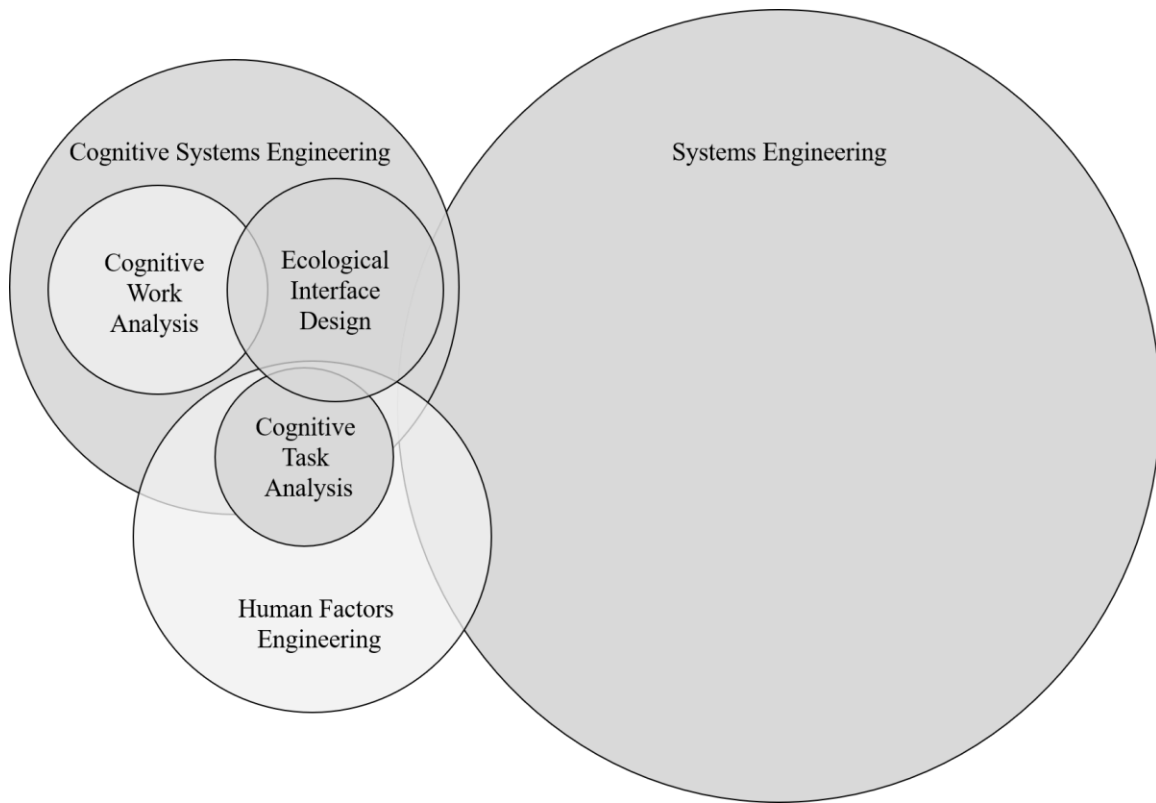


Figure 1 Relationships of Systems Engineering Methods

Defense in the Orbital Environment

The United States considers gaining and maintaining space superiority to be vitally important to the interests of the nation and its allies (Air Force Space Command, 2018, pp. 2–3). The safety of the US’s orbital assets is increasingly threatened by a growing number of counterspace weapons and tactics used on the ground or by spacecraft in orbit (National Air and Space Intelligence Center, 2018). Spacecraft operators must address these threats in a warfighting domain where measured escalation is untested, punitive actions are unprecedented, and making the wrong move can irreversibly damage an environment necessary for the conduct of a technological civilization (Frey, 2008). As

prior policy precluded the militarization of space, operators of the nation's current communications and observation satellites have limited resources to respond to hostile actions. Therefore, the prompt identification of hostile intent and efficient use of maneuvers for de-escalating dangerous situations is critically important to the continuity of space operations (Hanlon & Yakimenko, 2019). The literature on space defense describes the threats that satellites face, the political and legal environments decision-makers navigate, and the actions they can take to protect their assets.

Ranging from inspection and observation to directed energy and kinetic weapon attacks, counterspace weapons take many forms. In 2015, Russia launched the Kosmos 2504 satellite and demonstrated its ability to approach other objects and “maneuver extensively” around them (Gruss, 2015), an activity known as “rendezvous and proximity operations,” or RPOs. The same techniques that enable spacecraft to dock with the International Space Station also enable them to spy on other satellites, grapple and seize them with robotic arms, and intentionally cause destructive collisions. Other forms of co-orbital counterspace weapons which have been discussed in the open literature include laser and microwave attacks designed to disrupt electronics and blind sensors, as well as signals jamming and cyber-attacks designed to intercept data and seize control of satellites. China, Russia, and the U.S. have all demonstrated the ability to use these weapons and tactics, though seldom in conflict (National Air and Space Intelligence Center, 2018, pp. 15–19). The unprecedentedness of space conflict means that policy on direct response to threats and managing escalation is ill-defined, favoring caution and indirect defensive actions.

There have been multiple attempts to form international treaties that define the orbital environment as a space of common good, limit the use of weapons, and assign liability to nations for damage caused by their space vehicles. The 1967 United Nations Outer Space Treaty is the most notable, but it fails to create mechanisms for enforcing liability and only attempts to limit weapons of mass destruction, not the more subtle approaches that nations are developing today. Newer proposals, one of which Russia and China jointly proposed in 2008, were perceived as overly limiting peaceful activity and insufficient to account for the growing variety of counterspace weaponry (Defrieze, 2014, pp. 111–112). The weakness in international space policy enforcement means that the U.S. is left with the options of 1) defending its assets through military means, either in space or on the ground, or 2) using diplomatic and economic influence to deter adversary nations (Frey, 2008, pp. 79–81). To avoid escalating space conflicts into catastrophic situations, the U.S. can employ passive defenses to deny and deter adversary attacks, or else mitigate the damage they cause (Colby, 2016, p. 30). The legal and practical constraints of defending space assets, and defending the environment they operate in, limit the actions available to space operators to respond to aggressive actions by others.

If nations agree that destructive weapons are not in play, then space conflicts may take the form of remote observation, grappling, or cyber/electronic warfare methods that require proximity to accomplish. After determining that their asset may be a target, the first step in mounting a successful defense is identifying and locating an attacker (Hanlon & Yakimenko, 2019, pp. 3–4). Merely letting the adversary know that you’re watching them, through persistent space situation awareness provided by both ground and space observatories, may be sufficient to thwart their activities (Ewart, 2018). If the attacker

continues to move toward its target, then passive defense becomes dependent on evasive maneuvers. There are multiple ways of depicting the orbital environment for the purpose of planning and visualizing maneuvers.

The relative motion of two satellites can be depicted using the Radial / In-Track / Cross-Track (RIC) coordinate frame with an origin at the satellite of interest, where the radial axis points away from the center of the Earth, the in-track axis is along the spacecraft's velocity vector, and the cross-track axis is perpendicular to the orbital plane as shown in Figure 2.

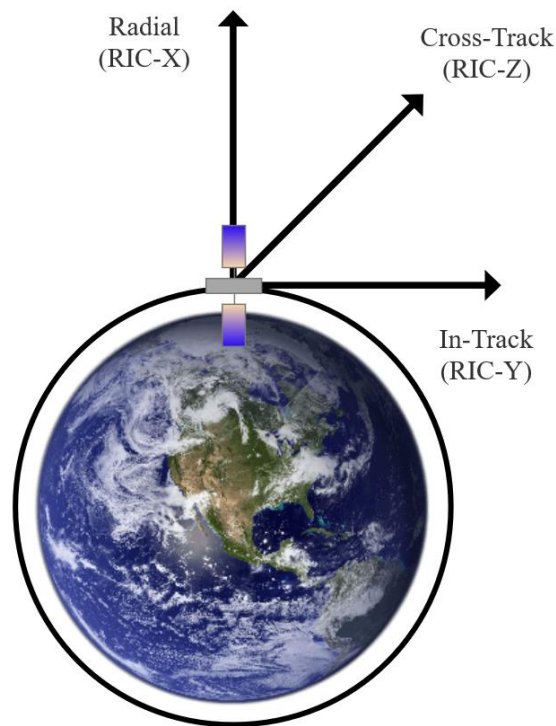


Figure 2 Depiction of the Radial/In-Track/Cross-Track Coordinate Frame

Applying thrust in any of these directions, or a combination thereof, allows a satellite to induce periodic motion that its pursuer must match. The defender's goal is to make the pursuer spend enough of its fuel to force it to disengage without spending enough of their own fuel to compromise mission effectiveness (Hanlon & Yakimenko, 2019, pp. 17–18). For an otherwise unprepared defender, observation and evasive maneuvers may be the only possible measures to defend against attack.

The motion of satellites about the Earth can be depicted using the Geocentric Inertial (GCI) coordinate frame (Figure 3). This coordinate frame has an origin at the center of the Earth with X, Y, and Z axes that are not fixed to the Earth's rotation and point to celestial objects, allowing coordinates to reference the position of satellites in orbit in a similar manner to the terrestrial latitude and longitude system (Boden, 1999). Visualizations using the GCI frame depict the orbital motion of satellites around the Earth as it rotates. This allows operators to see the full orbital period and orientation of one or more satellites, providing a strategic view of the orbital environment.

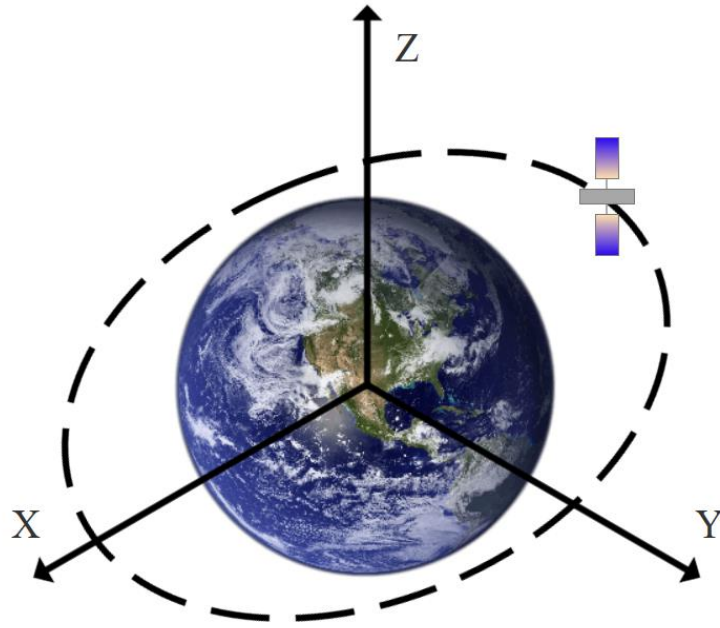


Figure 3 Depiction of the Geocentric Inertial Coordinate Frame

Space operators must consider the impact of limited resources (fuel) and other constraints when executing defensive maneuvers. In geostationary orbit, spacecraft must expend the equivalent of 50 m/s of fuel per year to avoid drifting away from their chosen location (Janson, 1993, p. 5). An effective maneuver may do the adversary's work for them by permanently reducing the defender's effective lifespan. Moving a satellite out of view of a ground station will sever communications links or reduce redundancy, affecting a satellite's ability to perform its mission. Solar panel angles are carefully pre-determined to provide the best power generation for the spacecraft's payloads, so operators must account for deviations from the optimum sun-solar panel angle as their spacecraft maneuvers around an adversary (Naasz, 2005, pp. 12–13). These constraints greatly

contribute to the challenge of selecting defensive responses that balance mission effectiveness against defense of the satellite.

The large variety of threats to U.S. spacecraft, the lack of an agreed-upon structure to orbital engagements, and the limited options available to defenders make space an extremely challenging warfighting domain. Research on these subjects suggests the need for creative and flexible approaches to defending threatened satellites in ways that avoid needless escalation into a kinetic conflict. To support space operators in challenging situations, cognitive systems engineering and task analysis methods may assist with defining requirements for mission planning and space asset visualization systems.

Cognitive Systems Engineering and Work Analysis

Designing systems that help space operators with complex problems requires new perspectives on the systems engineering process. Cognitive systems engineering (CSE) addresses complexity by designing systems that support operator problem solving and decision making in addition to rote tasks. To identify the kinds of problems space operators must solve, cognitive work and task analysis methods are used to model operators' work domains and thought processes. The resulting information and decision requirements necessary to accomplish their goals become design inputs for improved user interfaces through ecological interface design.

Cognitive systems engineering is an approach to designing computer or machine systems that focuses on supporting high-level problem solving and decision making (i.e., cognitive processes) (Hollnagel & Woods, 1982, p. 5). As people demand that their

computer systems solve ever more complex problems, the complexity of systems' operation threatened to outpace people's ability to use them (Hollnagel & Woods, 2005a, pp. 1–2). To address the complexity and unpredictability of modern tasks, cognitive systems engineering treats humans and computer systems as joint members of the problem-solving process, where these entities share problems, goals, and work in a joint cognitive system (Hollnagel & Woods, 2005a, pp. 21–24). The practice of studying goal-oriented behavior instead of rote mechanical labor, and applying the resulting design concepts, creates systems that are capable of helping operators use their natural cognitive abilities to reach their goals or supplying assistance when the task exceeds their abilities (Hollnagel & Woods, 2005b, pp. 7–8). The foundation of the cognitive systems engineering process is a study of the operators and their work domain, including their goals, the problems they solve in the presence of real-world or perceived constraints, and how they deal with unanticipated situations.

Designed to help the United States Navy produce better training programs for its sailors, the Applied Cognitive Task Analysis (ACTA) is an interview method that helps researchers identify how operators solve complex problems in their workplace (Militello et al., 1997). When operators adapt to challenging tasks, their method for performing the work can differ from what the system designers envisioned (Miller & Feigh, 2019, p. 10). This makes interacting with operators essential to documenting how problems are solved in the operational environment. The ACTA method, designed to take only an hour to complete, has operators perform a brief task analysis of their high-level goals that leads into a knowledge audit with probing questions about anomaly resolution, problem solving, and coordination. Operators demonstrate their problem solving techniques

during a simulation of a stressful scenario they might encounter in the workplace (Militello & Hutton, 1998b, pp. 1620–1625). The ACTA is an abbreviated form of the Critical Decision Method (CDM) (Klein et al., 1989), which develops an extremely rich dataset on decision-making from interview sessions that can last up to 10 hours (Hoffman et al., 2002, p. 483). Because the ACTA targets specific cognitive requirements in the workplace, it can provide sufficient detail to continue the cognitive systems engineering process in a fraction of the time. The answers to the probing questions and “what if?” analysis in the simulation scenario drive the process of identifying decision and information requirements for a computer system to support.

Applied Cognitive Work Analysis (ACWA) is a method for deriving decision support system requirements from an analysis of operator goals. Starting with a functional abstraction network, the ACWA decomposes high-level goals into lower-level subgoals linked to the means operators use to achieve them (Potter et al., 2003, pp. 2–8). A simpler form of this goals-means decomposition diagram, used in this thesis, is the functional abstraction hierarchy (Rasmussen, 1985). The ACWA proceeds with a development of operator decision and information requirements, such as picking a move in a chess game, that are met with corresponding visualization requirements in the user interface of a decision support system (Potter et al., 2003, p. 13). This kind of detailed cognitive analysis process has been successfully used in applications ranging from improvements to aircraft systems and powerplants to workplace training (Roth, 2008, p. 478). The combination of ACTA and ACWA methods produces specific requirements for user interface design that are directly traceable to operator goals in the workplace, resulting in systems that support complex problem solving.

The cognitive systems engineering process addresses usability issues in complex systems by rooting designs in the problem solving and decision-making processes of human beings. This changes the focus on system design from rote calculation to joint problem solving between humans and the machines they use. The ACTA interview method helps researchers gather information on goals and problem solving in the workplace. Using the ACWA method, the information is used to derive subgoals and information and decision requirements for computer systems to support. The requirements guide the application of ecological interface design to create effective user interfaces.

Ecological Interface Design

Data Overload Problem.

Human-computer interaction researchers have studied issues that face the operators of complex systems. Humans and machines alike have trouble sorting through large amounts of data to piece together the information elements relevant for their work, a condition called “data overload”. Environmental context changes can significantly change the priority of data, like when alarm codes suddenly become critically important during a specific phase of space flight. Unanticipated events can stymie brittle automated solutions and challenge the operators who are primarily trained on and experience routine operations. Ecological interface design (EID) addresses these challenges with design methods that account for complexities in the underlying environment, as well as, human cognition and decision-making, including direct interactions based on learned behaviors and perception (skill-based behaviors), interactions based on applying rules according to

cues and signals (rule-based behaviors) and performing complex problem solving based on domain knowledge (knowledge-based behavior) (Vicente & Rasmussen, 1992). Over the long-term, EID reduces deviations from optimal operator performance and makes anomalous situations easier to handle (Christoffersen et al., 1996). The literature on these topics suggests that an EID-based mission planning system might help space operators manage the complex set of operations necessary to ensure the survival of their satellite in a space combat scenario.

Data overload is aptly defined as the result of the “data availability paradox” (Woods, Patterson, Roth, & Christoffersen, 1999, p. 23): people demand more data in an attempt to better understand the operation of increasingly complex systems, but attempting to interpret a flood of data inhibits their ability to apply the data successfully to perform goals and tasks. Having too much data creates three types of problems for operators: physically cluttering the screen and crowding out useful information, bogging down processing and creating a backlog of unprocessed data, and masking the significance of important subsets of data (Woods, Patterson, Roth, & Christoffersen, 1999, p. 25). During a space combat scenario, a satellite operator has to use their knowledge of the capabilities of their satellite and the adversary to select a viable evasion response from a large list of possibilities, each associated with different constraints, fuel costs, and impacts on the pursuer (Hanlon & Yakimenko, 2019, pp. 10–11). In an analogous case, developers created a system that assists Army command and control decisions by dynamically highlighting the most relevant information and directly showing the relationship between actions and impacts on a model of the warfighting environment (Bennett et al., 2008). Working representations of the operational domain in

the user interface can assist operators with addressing data overload, and they can also help operators adapt to changing situations.

Context Sensitivity Problem.

Environmental context changes can alter the relative importance of small pieces of data in large data sets and change how an operator needs to interact with their system. For example, one alarm code out of hundreds on a spacecraft might have little significance mid-flight but be of critical importance during a landing (Woods, Patterson, Roth, & Christoffersen, 1999, p. 27). In space operations, a satellite making a routine orbit correction is normally barely worth noting, but a correction in a new direction in response to your own satellite's maneuver indicates a potential threat. These context changes have historically been difficult for machines to recognize and point out to human users. To improve performance, system designers must create interfaces that highlight changing relationships and structure data in ways that people find easier to process (Woods et al., 2002, p. 34). In the field of computer network defense, researchers built graphical representations of hacking activity to show relationships and highlight patterns between physical items (e.g., computer servers and locations) and logical activities that drastically reduced the effort operators had to spend interpreting data (Bennett et al., 2018). Representing the work domain and highlighting relationships in fields of complex data are key to helping operators cope with unfamiliar situations.

The literature on data overload and context sensitivity describe two related issues that impact the design of complex systems. Complex computer systems must assist their operators with identifying important details in large fields of data, responding to changes in the operational environment, and selecting the most effective responses out of all

possible options. With ecological interface design methodologies, system designers can use their knowledge of the operator work environment and human cognition to develop systems that increase the chances of mission success.

Model-Based Systems Engineering

Model-based systems engineering (MBSE) is a system design approach that allows engineers to work on graphical representations of system components and behaviors that influence each other as the design is updated, eliminating the need to repeatedly review and refresh design documents (Long & Scott, 2011, pp. 65–67). Systems engineers use languages like the Systems Modeling Language (SysML), based on the Unified Modeling Language, as a standard way to depict the elements of a system design and the relationships among these elements (Holt & Perry, 2013). With computer tools like Cameo Systems Modeler, designers can document systems using SysML and analyze how design trades meet requirements and influence other components of the system.

SysML models decompose systems into two mutually dependent sets of representations: structural diagrams and behavioral diagrams. The structural diagrams, including the block definition, package, internal block, parametric, and requirement diagrams, describe the logical or physical structures of a system and their relationships to each other (e.g., ATMs contain network cards which satisfy the requirement to connect to a bank). The behavioral diagrams, including the use case, sequence, activity, and state machine diagrams, illustrate user interactions with the system as well as the internal or external interactions between system components (e.g., an ATM user withdraws money

by inputting a card and PIN which triggers a detailed software subroutine) (Holt & Perry, 2013, pp. 88–91). With tools like Cameo Systems Modeler, designers can easily show how changes to one aspect of a system (e.g., the ATM PIN pad is replaced with a touch screen) impact performance and requirements satisfaction (so now breakdowns are more frequent and blind people can no longer use the machine). Identifying these design trade-offs are a key part of the cognitive systems engineering process.

SysML and Cameo also allow designers to iteratively develop complex system architectures starting with nothing more than descriptions of user goals and the activities they perform to meet them (Lamm & Weilkiens, 2014). As shown in Figure 4, SysML diagrams can be applied to effectively tie cognitive systems engineering artifacts, like goal models and information/decision requirements, to software user interface implementations. Maintaining the link between user goals and implementation ensures that system designs are rooted in workplace requirements and allows designers to explore different approaches to meeting user needs, as well as understand which user needs are supported by different implementations.

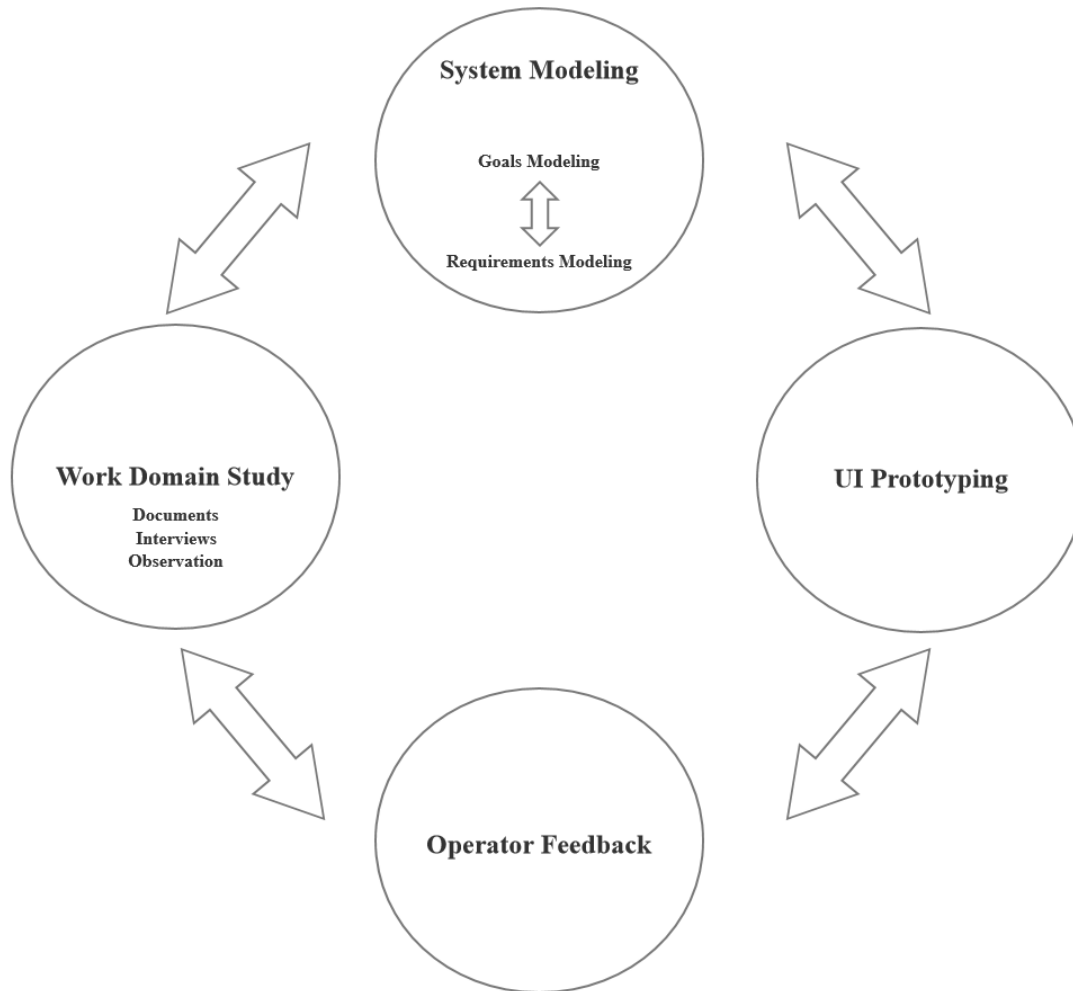


Figure 4 Cycle Diagram for Iterative Cognitive System Development

User Experience Survey

Assessing the user experience improvements afforded by the application of cognitive systems engineering and ecological interface design requires a user survey. The purpose of a user experience survey is to present potential users with a prototype of a user interface (UI) and gather suggestions or feedback, positive and negative, useful for validating and improving the design. Several candidate methods were considered for the

user experience survey. The System Usability Scale (SUS) (Brooke, 1996) and Usability Metric for User Experience (UMUX) (Finstad, 2010) are industry-accepted survey templates that ask users a series of questions with a 5 or 7-point response scale. These survey designs achieve statistically significant conclusions by comparing the scaled responses of large numbers of respondents. Few space operators are available for this research, so a more subjective approach is necessary. With the approach used herein, the researcher walks participants through a space operations scenario using the prototype UI and asks a series of probing, cognition-oriented questions derived from Mica Endsley's theoretical model of situation awareness (1995). The survey results and analysis provide an indication of whether users think the prototype offers a superior experience to their existing toolset, as well as provide them with a context to further understand or innovate features which might be useful within the intended domain.

Summary

This literature review summarized publications on concepts relevant to the understanding of this thesis. Defending assets in orbit is a challenging task, with a growing variety of weapons to contend with and limited means to respond. To develop computer systems to support space defense, cognitive systems engineering methods derive decision support requirements from an analysis of operator goals and problem-solving methods. These requirements guide the application of ecological interface design to create effective user interfaces that allow operators to apply their problem-solving skills on representations of the work domain that address common issues with interpreting and acting on complex data sets. System goals, requirements and behavior

are documented using the SysML modeling language in Cameo Systems Modeler, which allows system designers to explore design options and tie design specifications to high-level requirements. To assess how successful interface designs are at meeting operator goals, scenarios need to be presented to operators for their feedback. The following chapters describes the methodology used to document space operator goals and derive user interface requirements for a space operations system.

III. Using Cognitive and Model-Based Systems Engineering Methods for User Interface Design

Chapter Overview

This chapter describes the methods and tools used to model space operator goals and requirements to inform a prototype UI design. First, subject matter experts with space operations experience were interviewed using the ACTA process. These interviews were recorded, transcribed, and coded using MAXQDA 2018 Analytics Pro 2018. The outputs of this process were used to develop an operator goal model, functional abstraction hierarchy (FAH), and system information and decision requirements using Cameo Systems Modeler 19.0. This iterative process shows how cognitive analysis can be used to derive discrete system requirements that trace directly to operator goals and workplace decision-making. Chapter III describes the Work Domain Study and System Modeling portions of the process in Figure 4. Chapter IV will describe UI Prototyping and Operator Feedback to show how each system information requirement was developed into a prototype user interface (UI) that was implemented using Axure RP 9, presented to users, and documented in Cameo.

Methodology

Work Domain Study.

The work domain study began with a review of space operations literature, including discussions on conflict in space, international space policy, and issues facing organizations like Air Force Space Command and the United States Space Force. This process, called bootstrapping, drove the creation of the initial FAH to show how high-

level workplace goals are decomposed into smaller subgoals, processes, and functions with means-ends relationships. The Bootstrapping process allowed the vocabulary and system understanding to be developed which was necessary to design, conduct and interpret the results of the ACTA interview process developed by Militello & Hutton (1998a). In the ACTA interview process, operators participated in a series of activities that highlight concepts, decisions, and necessary information in their workplace.

Workplace information from the interviews was used to fill in any missing elements or relationships in the FAH which resulted from the bootstrapping process. In the current thesis, this FAH was developed in a model-based systems engineering tool, referred as Cameo Systems Modeler. This analysis then informed the development of a system requirements diagram. This diagram was customized to show the information operators need access to when making decisions in the workplace to satisfy their goals, as captured in the FAH.

Operator Interviews.

For the ACTA interviews, four USAF space operators with varying backgrounds were selected to provide a breadth of experience on how space operations are conducted and what operators think about when defending their assets. Each operator had performed at least one four-year tour in a space operations role. One operator had experience with managing spacecraft command and control ground stations and radar sites and had transitioned to a lead training role for junior operators. Another worked with highly maneuverable research and development spacecraft. The third helped coordinate space operations to meet the needs of joint forces personnel stationed overseas. The last participant oversaw operations for a large strategic-level asset. The operators were guided

through a tailored ACTA interview process designed to capture information relevant to deriving requirements for a new space operations interface.

The ACTA interview process was performed in three parts. The first asked participants to create a simple task diagram where they identified the steps necessary to perform major tasks in their workplace, such as maneuvering a satellite or performing their mission. During this process they were asked to identify which steps require the most complex decision-making, thought, or coordination to accomplish. Next, they participated in a knowledge audit where the interviewer asks probing questions to identify specific characteristics of expert performance in the cognitively demanding tasks. Finally, participants were walked through a simulation interview where they were presented with a realistic scenario that required stressful, high-importance decision-making to complete. They were asked to identify major events, point out key environmental cues, examine “what if?” events, and point out any mistakes they think a novice might make. The full ACTA interview process for this study is detailed in Appendix A. The interviews were video recorded for transcription and analysis by the researcher.

The video interviews were imported into a data analysis tool called MAXQDA Analytics Pro 2018. This tool allows researchers to transcribe video or audio interviews and categorize key passages in a process called coding. For this study, a code system was developed to highlight interview statements relevant for identifying operator goals, anomalous (challenging or unexpected) situations, real world examples, environmental cues, and specific decision and information requirements.

Work Domain Analysis.

The information from the document bootstrapping and operator interviews were used to develop an FAH for space operations. The FAH represents high-level goals, subgoals, and the processes and objects operators use to meet those goals. The top level, Domain Purpose, describes the highest level and ultimate purpose of workplace activities. Operators will prioritize Domain Values, such as maintaining or maximizing conditions of the system, to satisfy top-level goals to different degrees. The Domain Functions are the day-to-day workplace functions operators perform, which break down into Physical Functions and Physical Objects related to specific spacecraft components, administrative artifacts like briefings and documents, and physical activities. The overview of the FAH, including the many relationships between the elements, is shown in Figure 5.

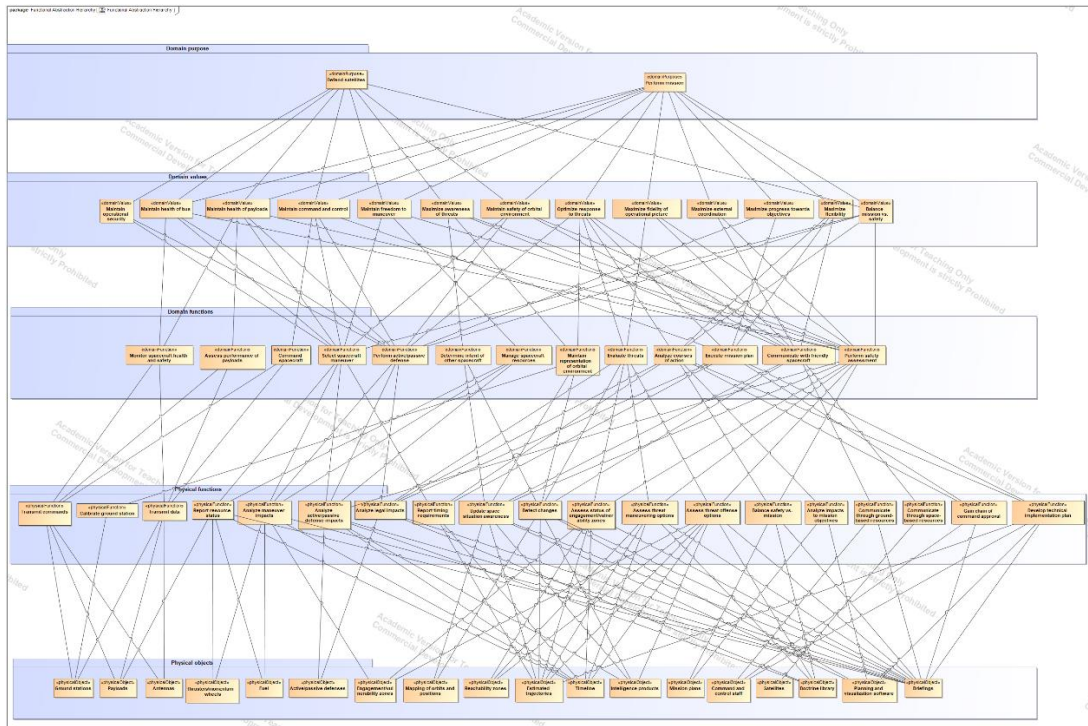


Figure 5 Functional Abstraction Hierarchy for Orbital Defense Depicted in Cameo Systems Modeler 19.0

To create an FAH in Cameo Systems Modeler, the SysML modeling language was extended with a new FAH diagram type as shown in Figure 6. The five levels in the hierarchy were represented as packages containing goals, processes, functions, and objects. These elements were also used in a new implementation of the built-in Requirements Diagram that featured additional information and decision requirement elements and interface implementation specifications as will be further described in Chapter IV. The new elements were created as stereotypes derived from the “Class” meta-class. This implementation preserved the ability to create relationships between the

elements in the requirements diagram and FAH for full traceability from requirements to cognitive processes and goals.

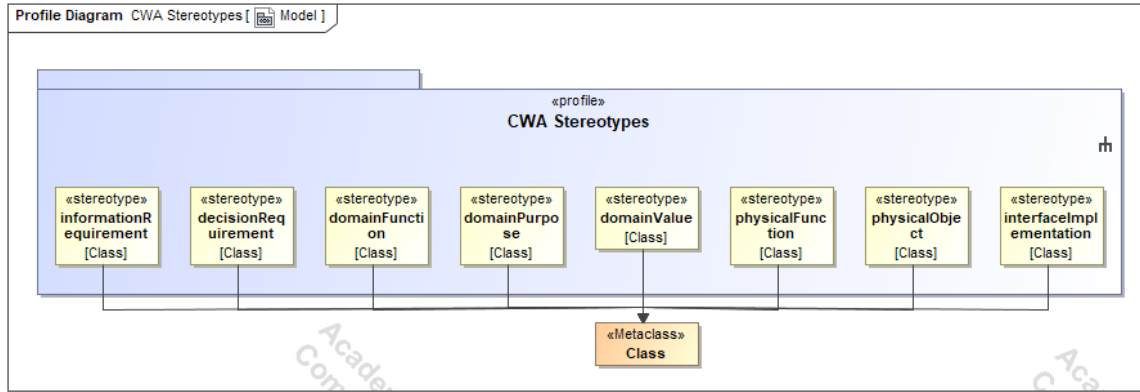


Figure 6 Custom Profile Extending SysML for CWA Studies

The completed FAH was used to establish goals and functions to be met by system requirements. The domain functions were each derived into decision requirements that were also informed by the operator interviews. These requirements, and the interface implementations described in Chapter IV, were represented with a SysML requirements diagram.

Requirements Modeling.

Once the FAH was completed, each of the domain functions was applied to derive system requirements. The domain function level of the hierarchy was chosen for this process because it represents the mid-point between physical functions and non-physical domain values. Requirements created at this level tie physical elements to abstract needs. Each domain function was associated with a single decision requirement that represents a

decision the operator must make about how they will perform that function. The decision requirement was supported by one or more information requirements, which represent what the operator would need to see in the interface to make their decision. Finally, user interface implementations were suggested to meet the information requirements. This process is shown generically in Figure 7. In Chapter IV, the process for creating prototype UI elements and assigning them to the interface implementations is shown.

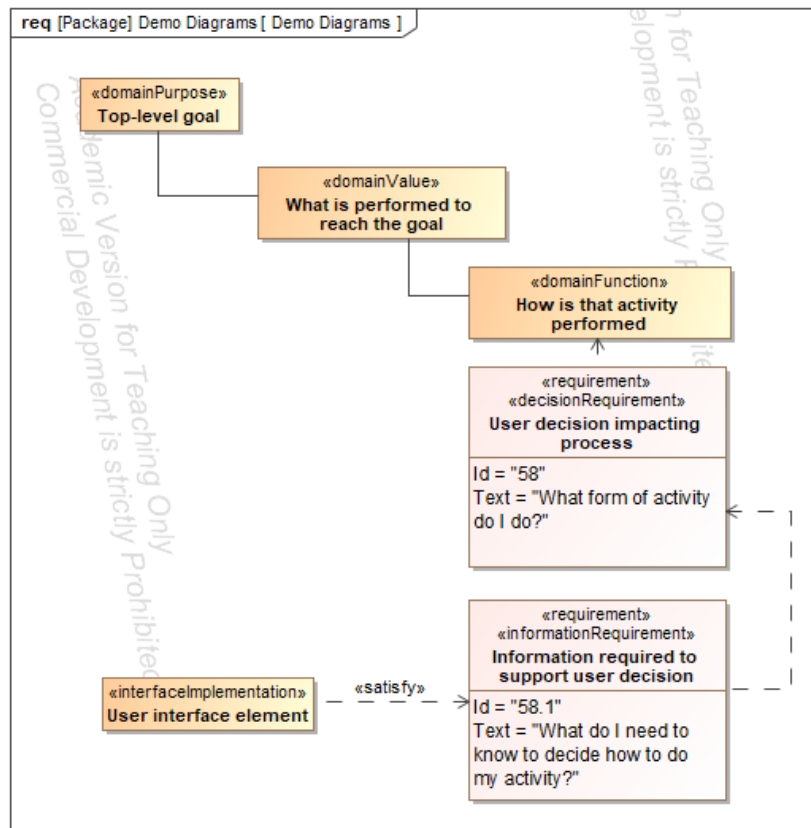


Figure 7 Example Decomposition of a Top-Level Goal into Information/Decision Requirements Met by a User Interface Implementation

Results

Interview Coding.

The operator ACTA interviews were recorded and transcribed according to the code system in Figure 8. Across the four interviews, 78% of the codes corresponded to goals, decision requirements, and information requirements that directly translated into elements of the FAH and requirements models. The remaining 22% corresponded to stories about system anomalies, personal experiences, example explanations, and environmental cues that assisted with the research process by providing context or illuminating information about the space operations domain. Although these responses were not part of the system model, they were nonetheless important for improving the researcher's understanding of how space operators perform their work and guiding the design of the user interface.

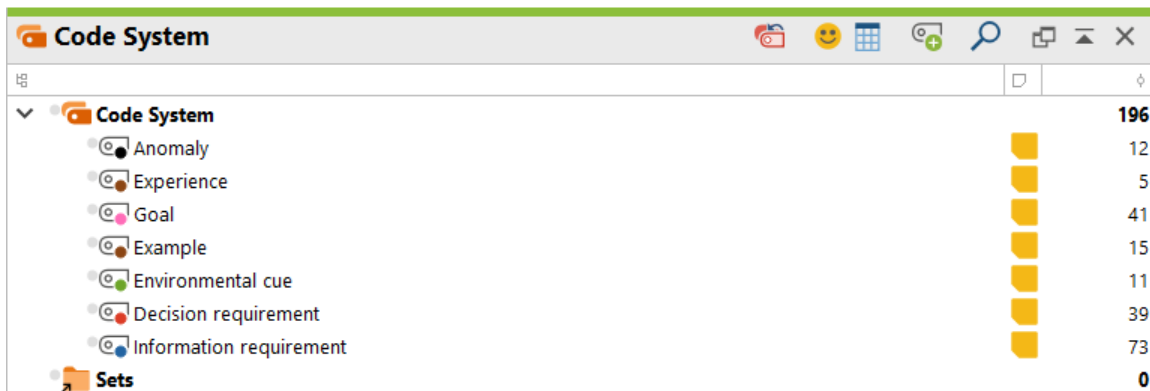


Figure 8 Code System for Transcribed Operator Interviews from MAXQDA

Coding the video interviews provided insights that were not obvious from reading space operations and policy literature. For example, one operator discussed how

performing maneuvers with multiple segments can be more efficient than performing one large maneuver. They described how mission plans must be coordinated around the availability of communications infrastructure to send commands. The operators suggested that performing their mission and ensuring the safety of their spacecraft were goals that had to be balanced against each other and were not always mutually achievable. These insights were critical for developing a representative workplace goal model and deriving system requirements.

Table 1 contains examples of the translation from coded interview segments to domain values, domain functions, and requirements. Each coded segment implied or explicitly stated one or more domain requirements or goals. When an operator stated that performing multiple maneuver segments may be a better approach than performing one large maneuver, this implied a major decision about what kind or how many maneuvers to execute, which implied that the operator would have to be able to weigh the costs and benefits of the different maneuvers. These statements were mapped to elements of the system model to improve its representation of the space operations domain.

Table 1 Mapping of Interview Content to Domain Model

Code	Interview Segment	Goal/Requirement Mapping
Decision Requirement	Interviewer: What would you say is the biggest indicator that someone is performing at an expert level with [planning], versus somebody just starting out? Participant: If they're looking ahead at how this maneuver sets them up to do another maneuver in six hours instead of just trying to do everything in one burn. Usually if you tell students to do something they'll just try to do it in one burn and get there when the more optimal way is to use multiple burns.	Decision Requirement -> Maneuver Decision: What kind of maneuver do I want to execute? Information Requirement -> Maneuver Segments: What's the tradeoff between one big maneuver and multiple smaller maneuvers?
Information Requirement	Interviewer: I want to know what you tell [operators] about what their goals are and how they achieve those. Let's say, in a 1-on-1 situation and your top-level goal is to defend an asset. How do you break that down? Participant: I would say do your mission plan. Your mission plan involves looking at the current state of the system, the geometry of the system, it involves looking at the sensors available. You're looking at your communications available - meaning, say we only have contact to give commands to this satellite every 3 hours or so, so plan for that.	Information Requirement -> Location of Other Spacecraft: What other spacecraft are in my environment and where are they? Information Requirement -> Update Rates: How often is the information on my own satellite and the orbital environment updated? Information Requirement -> Maneuver Constraints: What constraints, like sunlight angle, time, and fuel, impact my maneuver options?
Goal	Interviewer: So you're balancing what you do to defend the asset vs. how it impacts the mission? Participant: Yeah. I would also say that applies to how the defense is being applied as well. If the satellite is defending itself then certainly it's a straightforward defense, do you protect the mission or do you make a change that degrades mission but preserves longevity?	Domain Value -> Balance of Mission vs. Safety Domain Function -> Perform Safety Assessment

Functional Abstraction Hierarchy.

The information from the document bootstrapping and operator interviews were used to develop an FAH for space operations. The content of the five levels is represented in Table 3. In the top level, Domain Purpose, “Defend satellites” and “Perform mission” were the ultimate goals for space operators. As the operators pointed out, these can sometimes be contradictory goals: moving your satellite to defend against a threat can reduce its ability to perform its mission. The Domain Values were balanced between supporting the spacecraft and its payloads (e.g. Maintain health of payloads) and working strategically toward accomplishing the mission (e.g., Maximize fidelity of operational picture). The Domain Functions such as “Select spacecraft maneuver” and “Manage spacecraft resources” described spacecraft-oriented tasks while functions like “Determine intent of other spacecraft” implied the need for intelligence gathering and analysis. These functions broke into Physical Functions and Physical Objects related to spacecraft components like thrusters and organizational items like briefings and doctrine libraries.

Table 2 Abstraction Level and Elements of Functional Abstraction Hierarchy

Abstraction Level	Elements
Domain purpose	Defend satellites; Perform mission
Domain Values	Balance mission vs. safety; Maintain command and control; Maintain freedom to maneuver; Maintain health of bus; Maintain health of payloads; Maintain operational security; Maximize safety of orbital environment; Maintain awareness of threats; Maximize external coordination; Maximize fidelity of operational picture; Maximize flexibility; Maximize progress towards objectives; Optimize response to threats
Domain Functions	Analyze courses of action; Assess performance of payloads; Command spacecraft; Communicate with friendly spacecraft; Determine intent of other spacecraft; Evaluate threats; Execute mission plan; Maintain representation of orbital environment; Manage spacecraft resources; Monitor spacecraft health and safety; Perform active/passive defense; Perform safety assessment; Select spacecraft maneuver
Physical Functions	Analyze active/passive defense impacts; Analyze impacts to mission objectives; Analyze legal impacts; Analyze maneuver impacts; Assess status of engagement/vulnerability zones; Assess threat maneuvering options; Assess threat offense options; Balance safety vs. mission; Calibrate ground station; Communicate through ground-based resources; Communicate through space-based resources; Detect changes; Develop technical implementation plan; Gain chain of command approval; Report resource status; Report timing requirements; Transmit commands; Transmit data; Update space situation awareness
Physical Objects	Active/passive defenses; Antennas; Briefings; Command and control staff; Doctrine library; Engagement/vulnerability zones; Estimated trajectories; Fuel; Ground stations; Intelligence products; Mapping of orbits and positions; Mission plans; Payloads; Planning and visualization software; Reachability zones; Satellites; Thrusters/momentum wheels; Timeline

The overview of the FAH, including the many relationships between the elements, is shown in Figure 9. From top to bottom, associations show how a higher-level goal, value, or function is implemented (the “how”). From bottom to top, associations

show the higher-level purpose a goal, value or function has (the “why”). Values, functions, and objects that were not represented in the final prototype UI are highlighted in red. In the Domain Functions level, “Perform active/passive defense” and “Communicate with friendly spacecraft” were outside the scope of this research and were not represented. The physical and organizational functions and objects supporting these goals, such as legal analyses and weaponry, were also not included. The missing elements represent 30% of the overall diagram, but the remainder sufficiently fulfilled the higher-level goals and values. Incorporating the missing elements would give operators additional options to meet their goals, information helpful for prioritizing them, or provide additional constraints on their actions. They could be met by additional functionality in the prototype UI or by other software altogether.

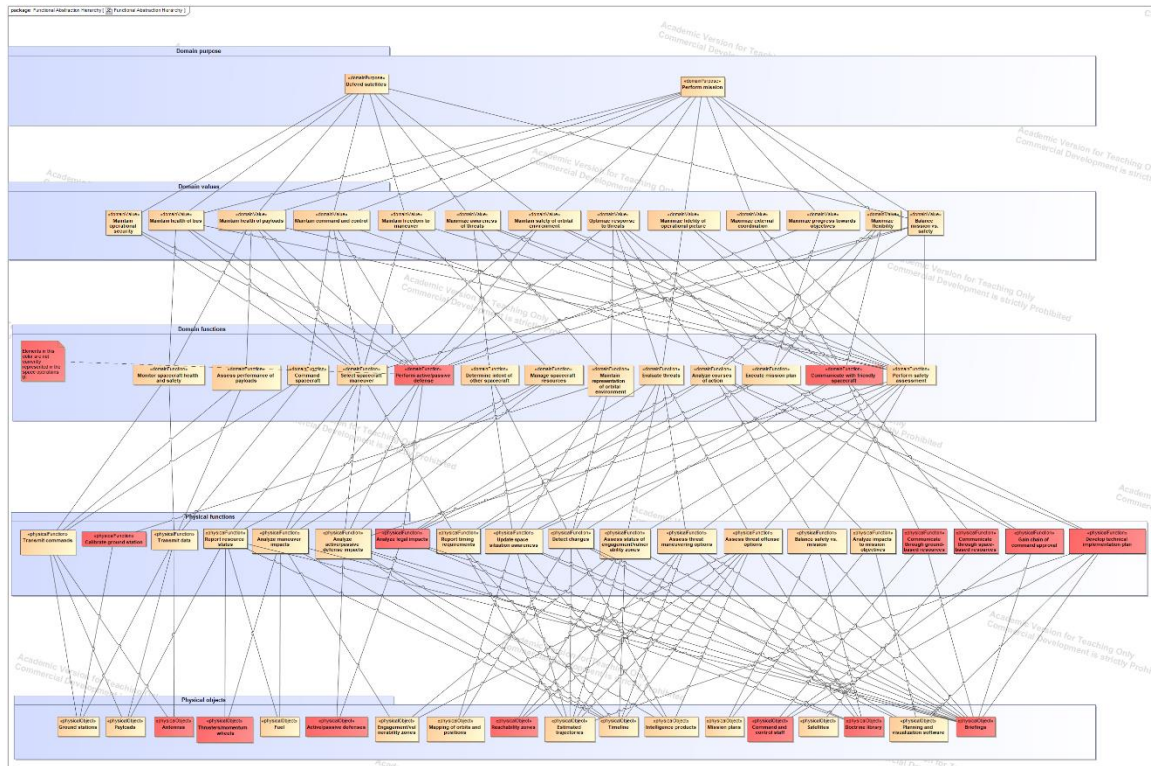


Figure 9 Functional Abstraction Hierarchy with Exclusions Marked in Red

Completing the FAH showed the number and complexity of goals, functions, and objects to be implemented in a prototype operator UI. Examining the number of relationships between diagram elements showed that concrete functions related to spacecraft operations, such as monitoring health and safety or payload performance, met a small number of higher-level goals and could be implemented with fewer processes than the more abstract functions, as shown in Figure 10. The abstract functions, such as performing safety assessments (“Is my spacecraft safe from harm?”), supported a much larger set of goals and, accordingly, needed support from a larger set of physical functions (Figure 11). Examining the complexity of the relationships indicated operator needs that the user interface would need to work harder to meet. Representing the temperature of a sensor could be done with a single interface element, but showing the

user that their spacecraft was safe from harm could require multiple, coordinated elements that showed how the friendly spacecraft position, enemy spacecraft position, resource loads, and possible maneuvers contribute to the notion of safety.

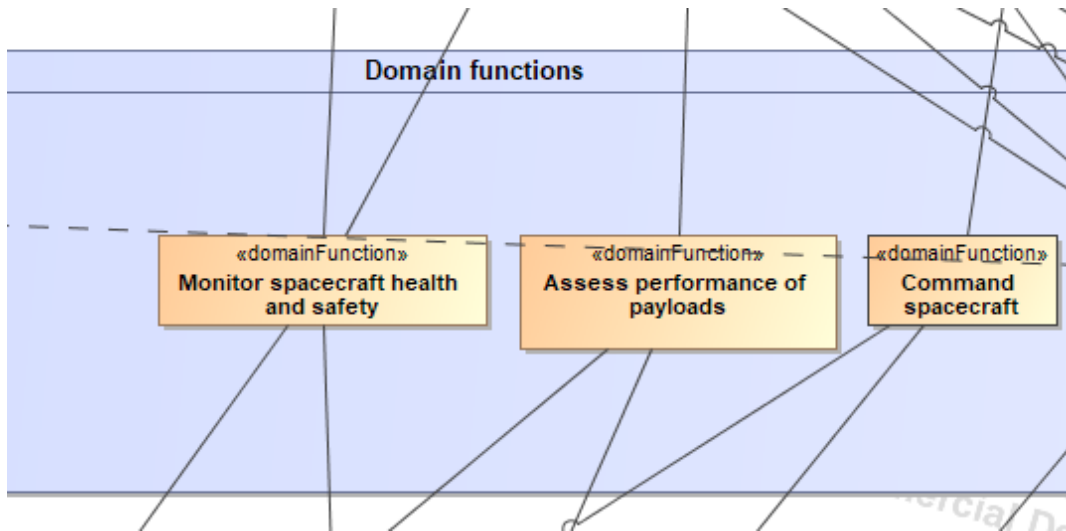


Figure 10 Relationship Complexity of Concrete Domain Functions

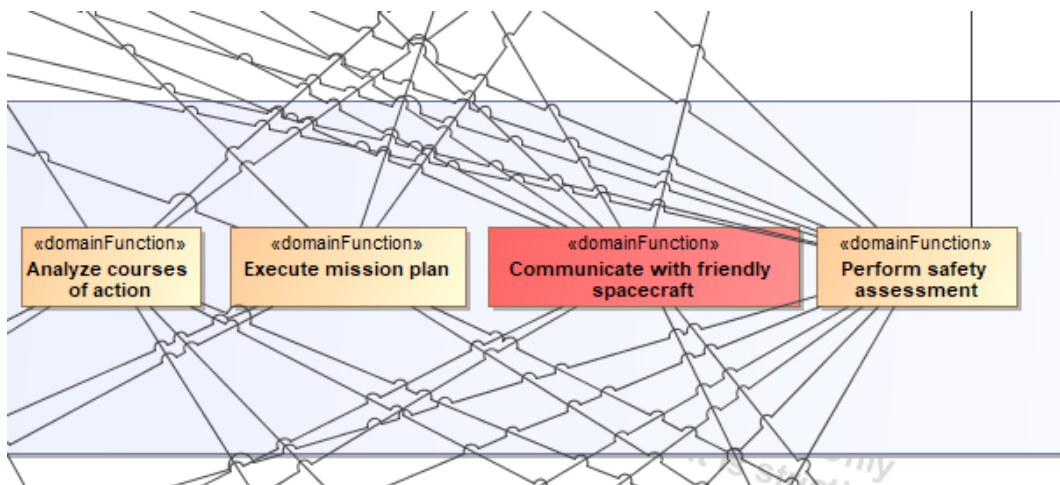


Figure 11 Relationship Complexity of Abstract Domain Functions

The full set of associations between elements of the FAH are shown in matrices in Figures 12-15. Associations are indicated with slash marks between items in the upper level (rows) and the lower level (columns) for each level in the hierarchy. Elements that have not been implemented in the prototype UI are highlighted in red to clearly show which goals and functions the software does not support.

There was substantial overlap between the values linked to the “Defend satellites” and “Perform mission” top-level goals in Figure 12. The satellite can neither survive nor perform its mission if its payloads are not functioning. However, there were five values linked exclusively to performing the satellite’s mission: maximizing external coordination, maximizing flexibility, maximizing fidelity of the operational picture, maximizing progress towards objectives, and optimizing response to threats. These values are not essential to the operation of the satellite or its payloads but are essential for accomplishing the satellite’s mission. The lack of overlap implied that functions dedicated to mission analysis would not have to interact with or share visual space with those dedicated to supporting the satellite, which informed the layout of the prototype interface.

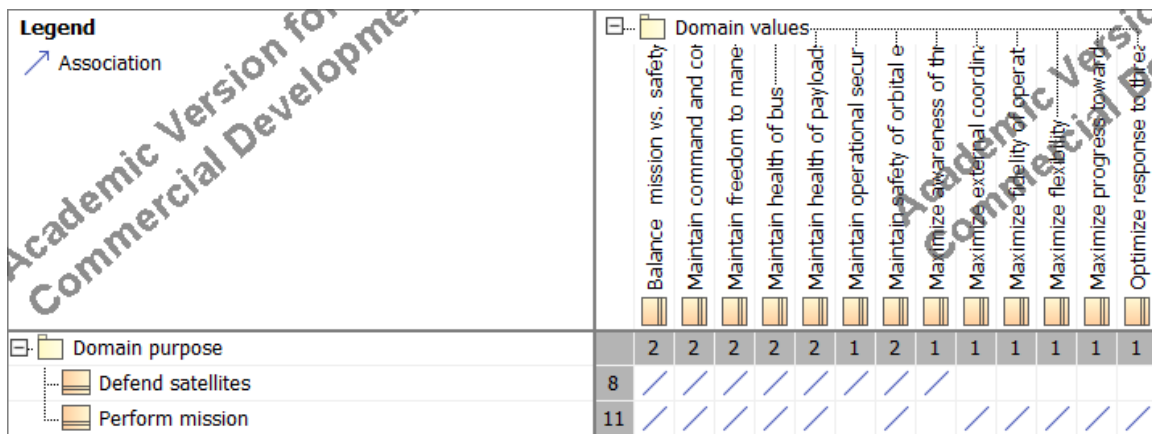


Figure 12 Domain Purpose and Domain Values Levels of Functional Abstraction

Hierarchy

The goal-oriented impact of not representing communication with friendlies or active/passive defense in the system is shown in Figure 13. Although each domain value was associated with at least one domain function, there were 12 values that were incompletely supported because these two functions were not selected for implementation in the prototype UI. The inclusion of a single function could enhance an operator's ability to optimize many domain values considered necessary to achieve their goals. The number of associations between values and functions also provided an indicator of interface complexity. Domain functions supporting single values, such as "Assessing performance of payloads," could be represented with a single, simple interface element. Those associated with multiple values, such as "Perform safety assessment," would need multiple and/or complex interface elements to support operator decision-making.

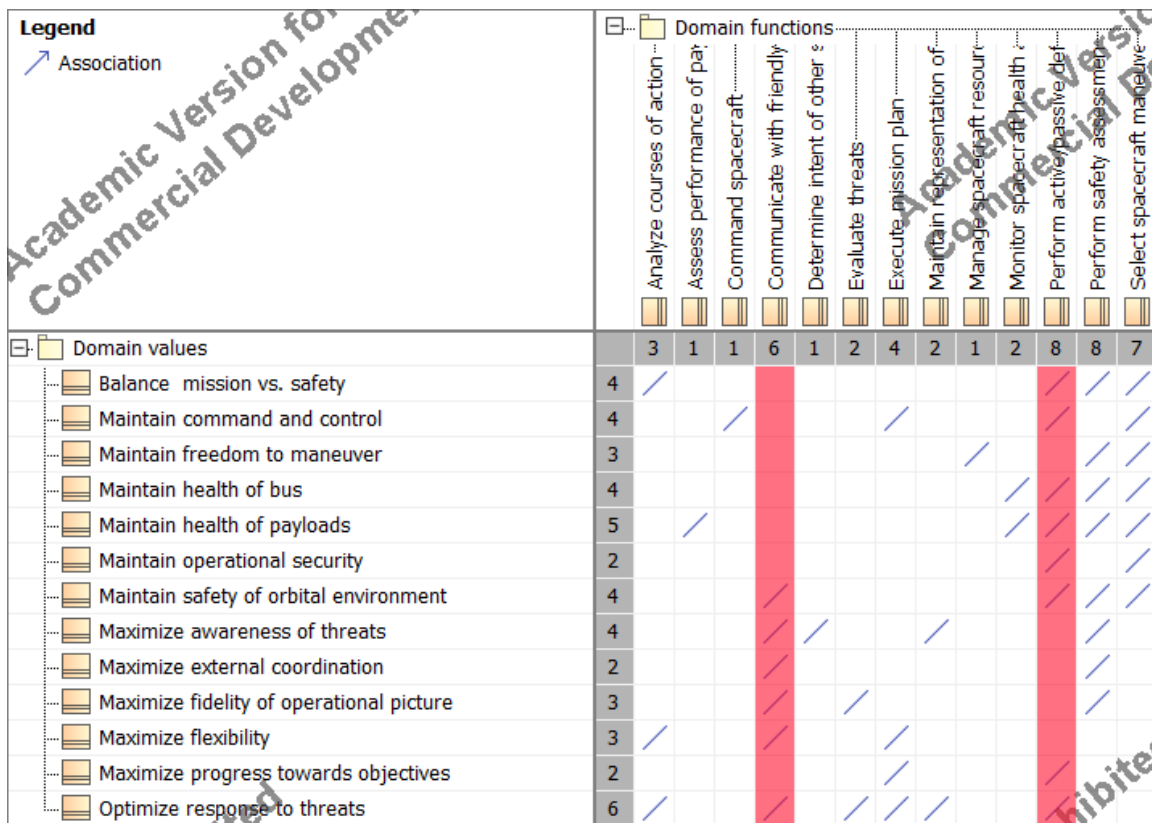


Figure 13 Domain Values and Domain Functions Levels of Functional Abstraction Hierarchy

Figures 14 and 15 show the physical functions and objects association with space operations tasks. Although some elements had many associations with elements in higher levels, that did not indicate their level of importance. For example, briefings supported 13 different functions by providing guidance, context, and constraints. However, while a spacecraft may be operated without organizational guidance it absolutely cannot be

operated without sufficient fuel. The implied complexity of interface representations was not an indicator of relative importance.

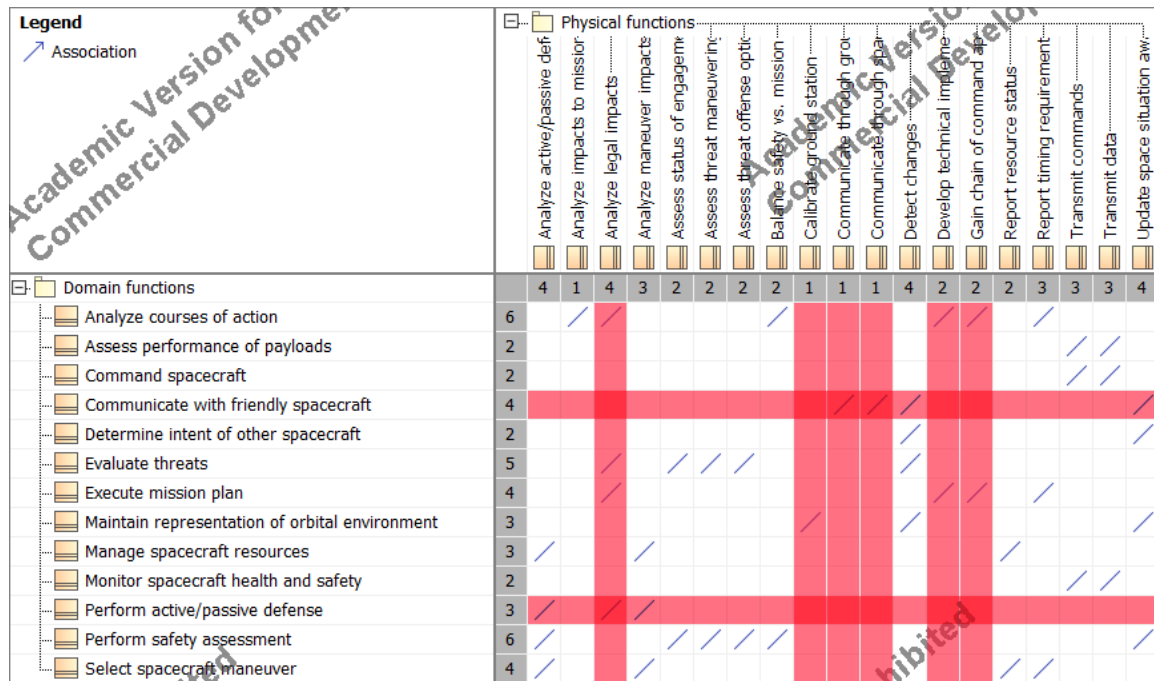


Figure 14 Domain Functions and Physical Functions Levels of Functional Abstraction Hierarchy

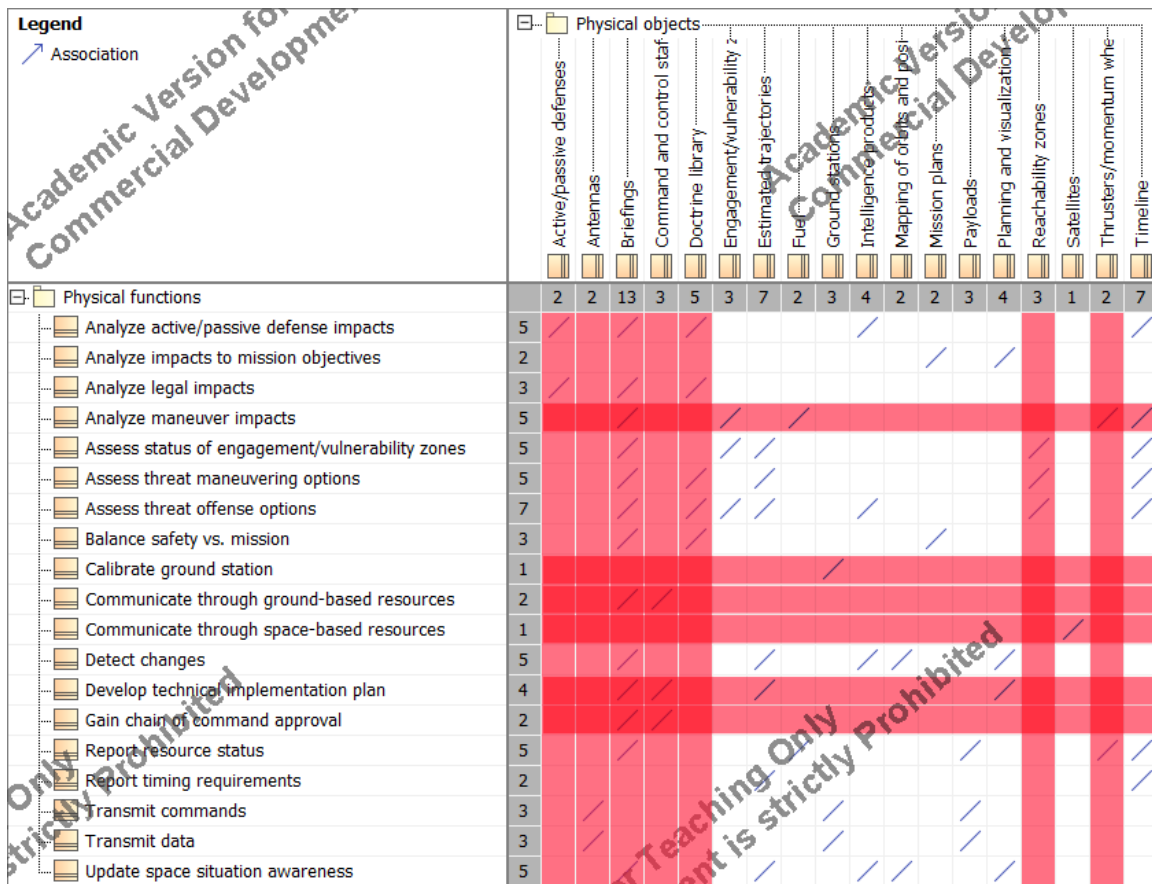


Figure 15 Physical Functions and Physical Objects Levels of Functional Abstraction Hierarchy

Requirement Diagrams.

The operator interviews and FAH were used to derive decision and information requirements associated with each domain function. Each decision requirement was supported by one or more information requirements that tied to suggested interface implementations. The requirement diagrams became the main input to the interface design process discussed in Chapter IV.

The Course of Action (COA) decision in Figure 16 asks, “What kind of response do I (the operator) want to make?” This decision supports domain values in the FAH that correspond to comparing the consequences of multiple possible actions. To make this decision, the operator would need to be able to visualize the current and future states of the orbital environment and plan maneuvers accordingly.

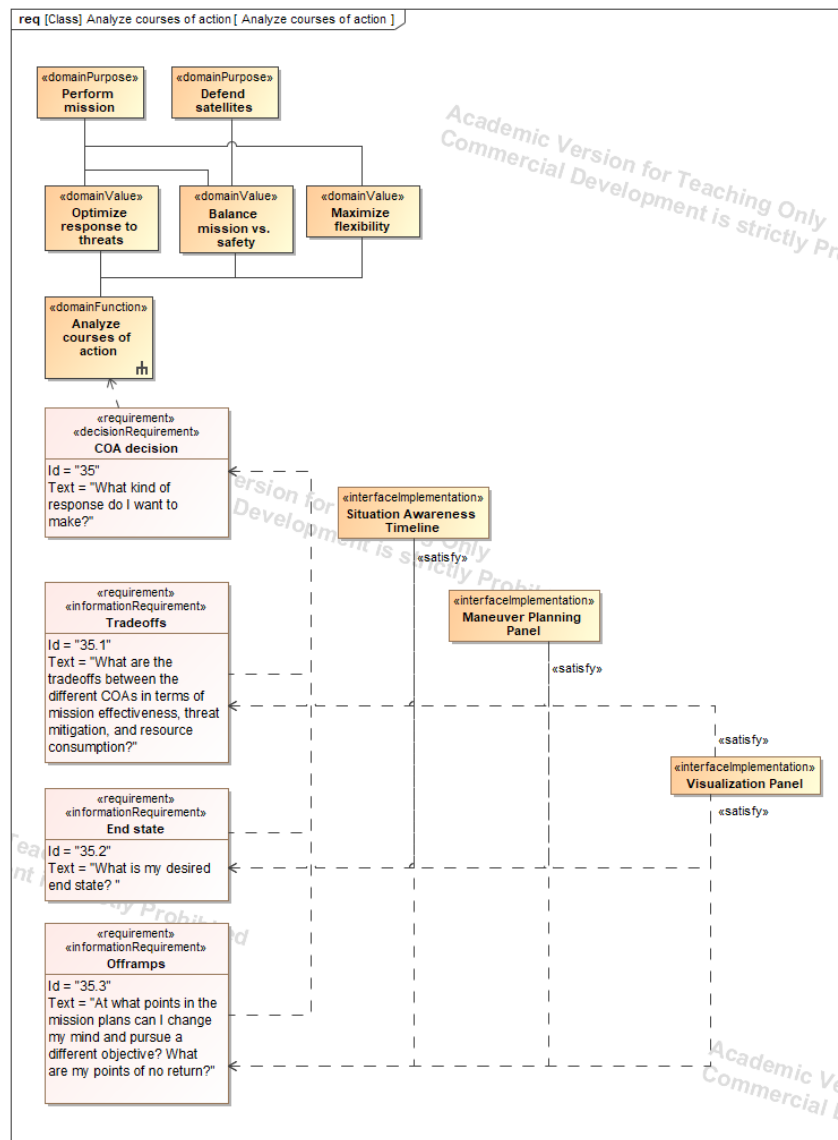


Figure 16 Course of Action (COA) Decision Requirement

The “payload performance” decision in Figures 17 and 18 asks, “How do I address any issues with my payload performance?” This decision supports the domain value in the FAH to maintain the health of the payloads. To make this decision, the operator would need to know the state of health of each payload as well as how each payload is contributing to their mission goals.

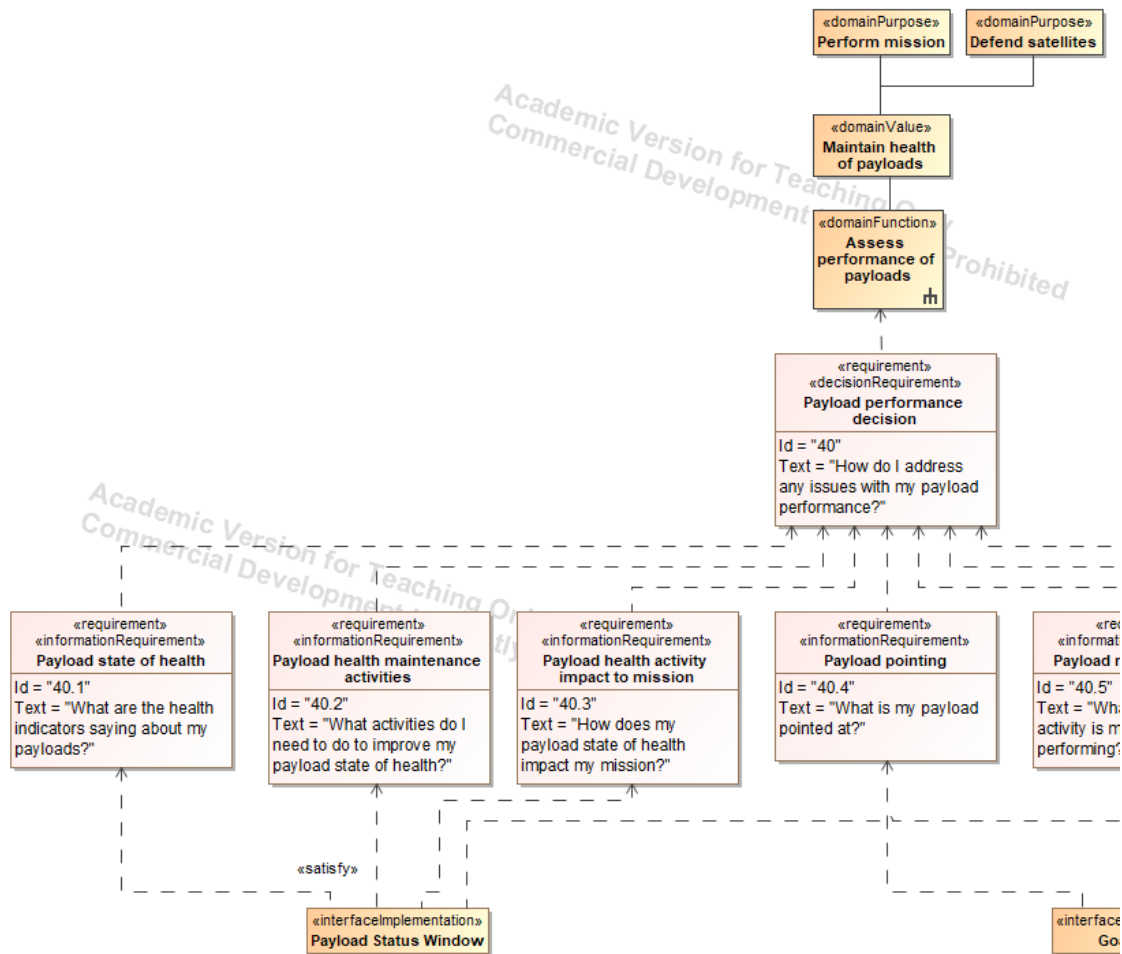


Figure 17 Payload Performance Decision Requirement (Left)

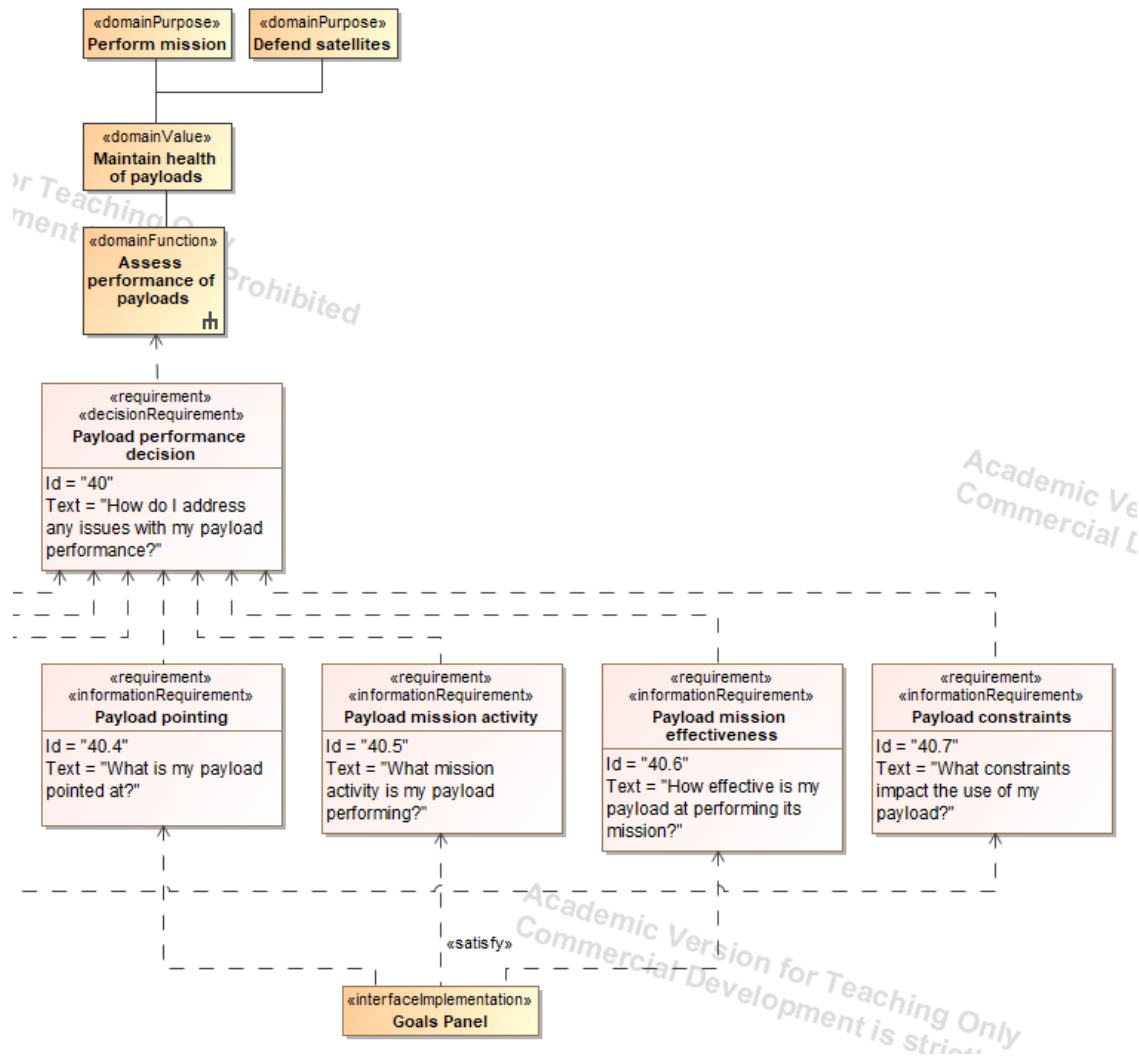


Figure 18 Payload Performance Decision Requirement (Right)

The “command” decision in Figure 19 asks, “How do I send commands to my spacecraft?” This decision supports the domain value in the FAH to maintain command and control of the spacecraft. To make this decision, the operator would need to be able to visualize any issues impacting their ability to send and receive commands through ground control stations.

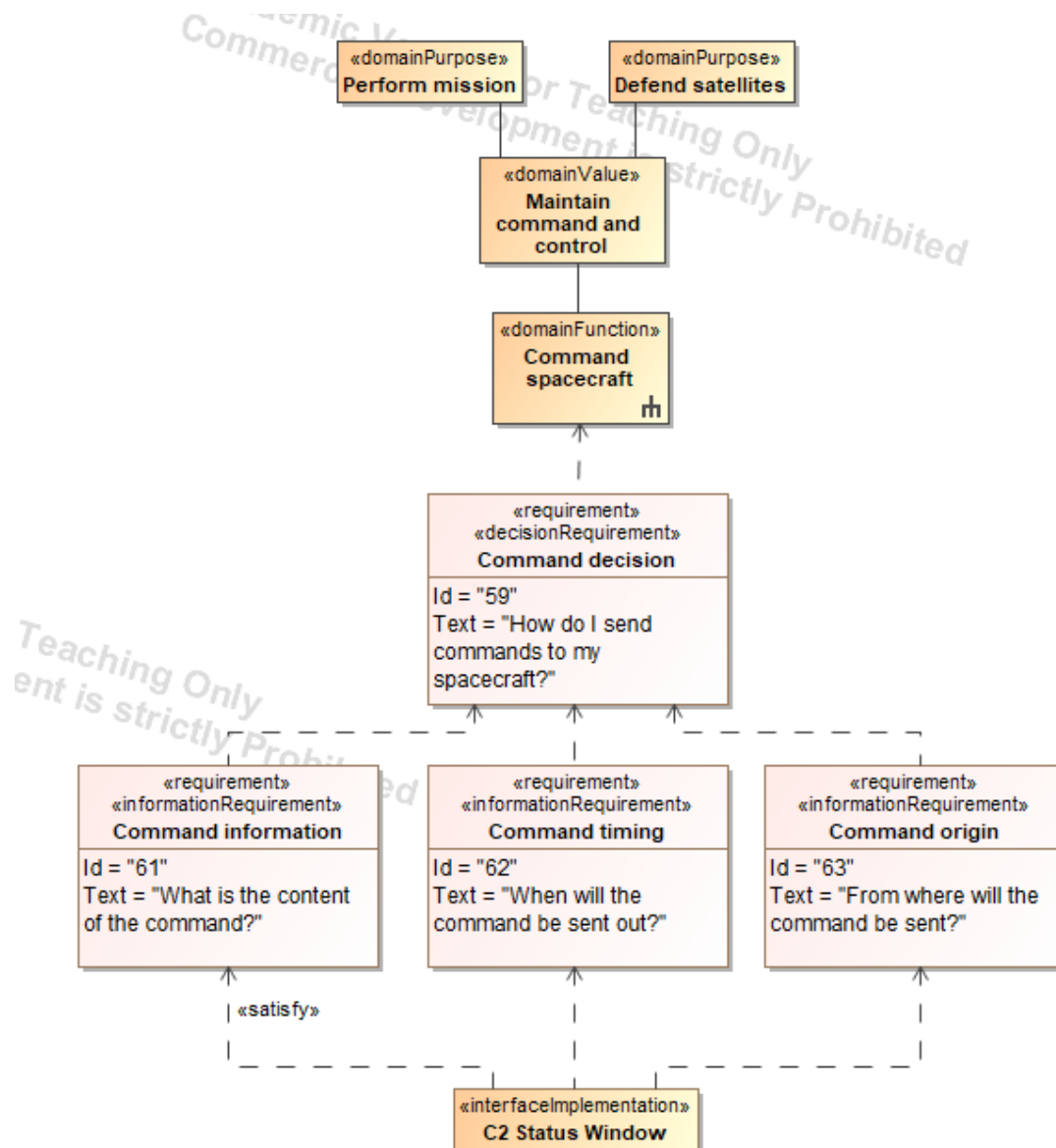


Figure 19 Command Decision Requirement

The “communication with friendlies” decision in Figure 20 asks, “Should I ask for help or warn my friends?” This decision supports domain values in the FAH that correspond to knowledge of the orbital environment and responding to threats within it. Representing multiple friendly spacecraft was outside the scope of the space operator UI prototype and is not associated with any suggested interface implementations.

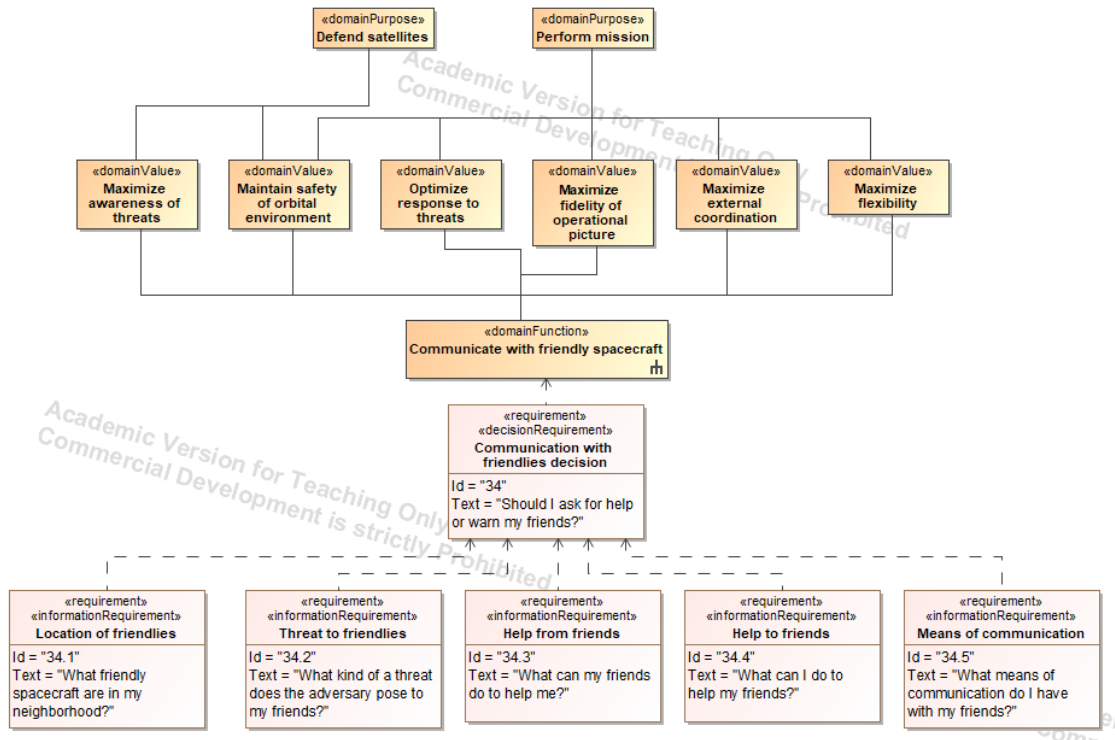


Figure 20 Communication with Friendlies Decision Requirement

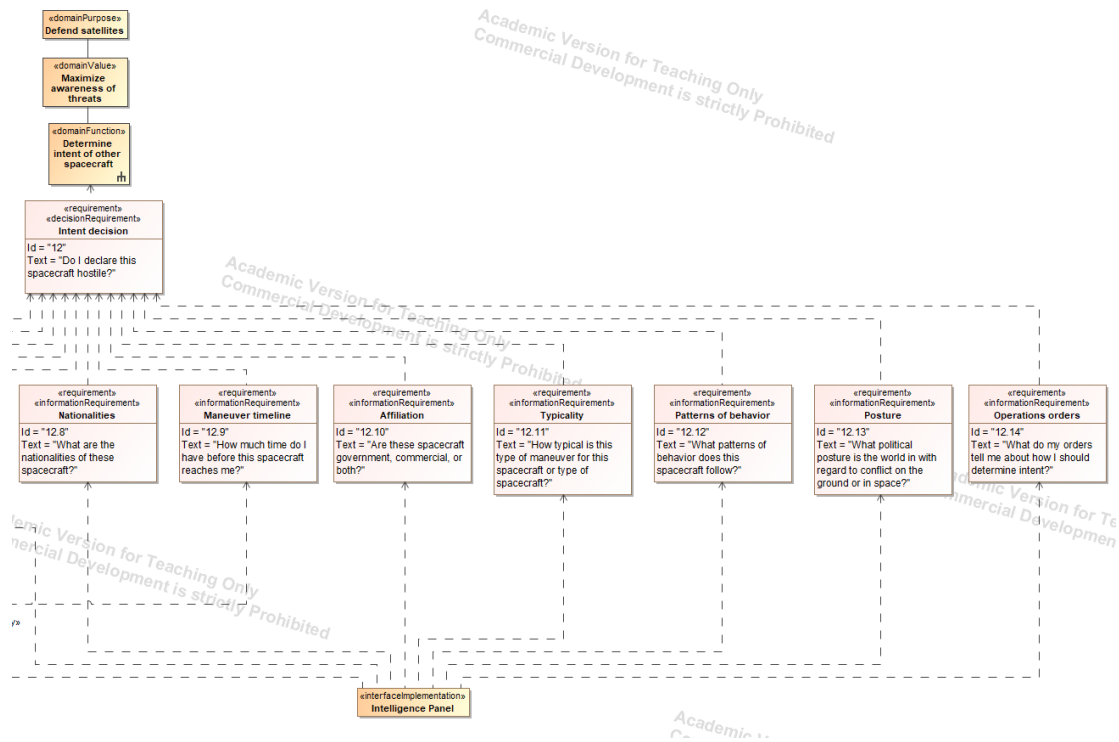


Figure 22 Intent Decision Requirement (Right)

The “threat evaluation” decision in Figure 23 asks, “What kind of threat does the adversary pose?” This decision supports domain values in the FAH that correspond to knowledge of the orbital environment and responding to threats within it. To make this decision, the operator would need intelligence on the enemy spacecraft, its payloads, and possible hostile maneuvers.

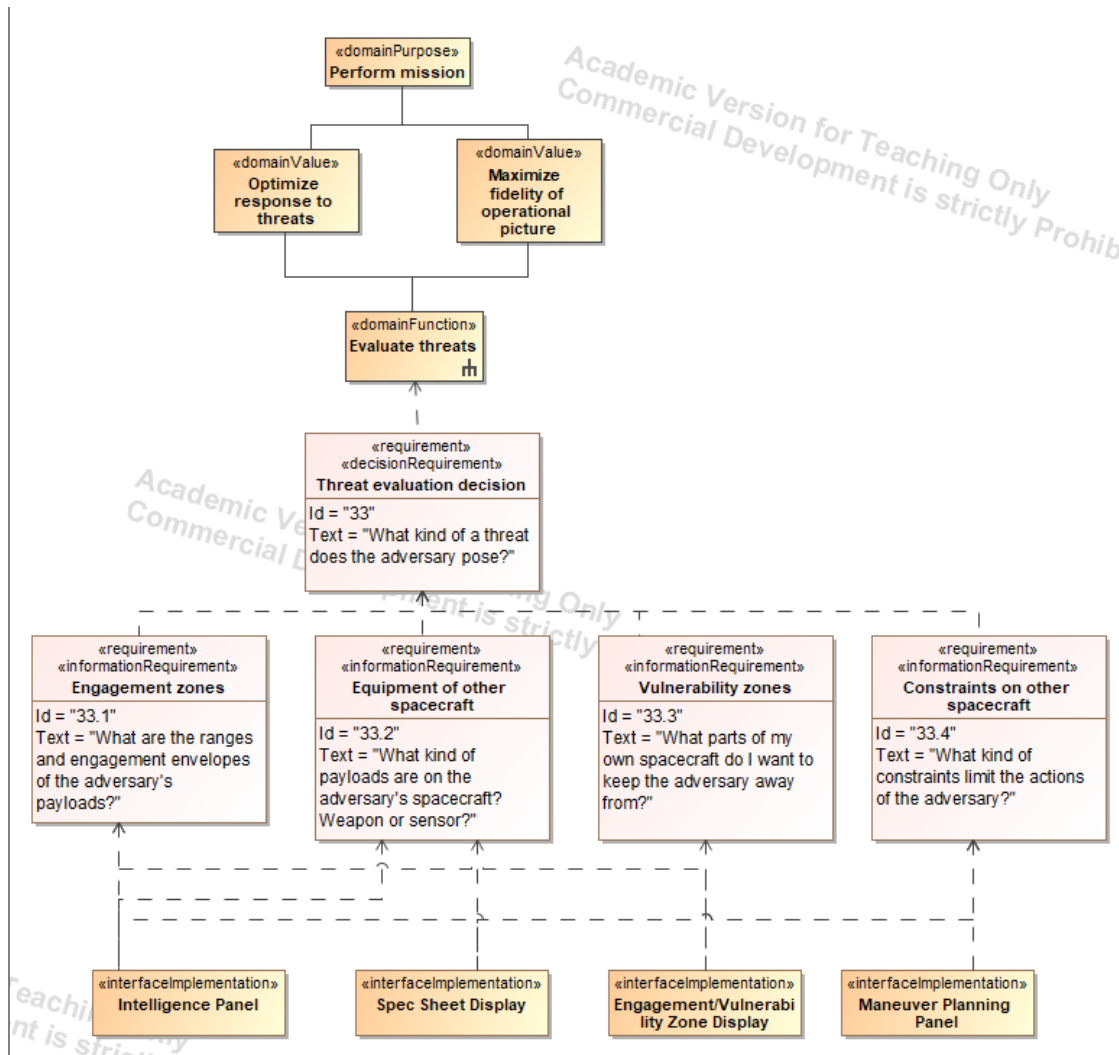


Figure 23 Threat Evaluation Decision Requirement

The “resource management” decision in Figure 24 asks, “How do I base my strategy on resource consumption?” This decision supports the domain value in the FAH to maintain freedom to maneuver. To make this decision, the operator would need to be able to consider their current resource levels and the impacts of different maneuver plans.

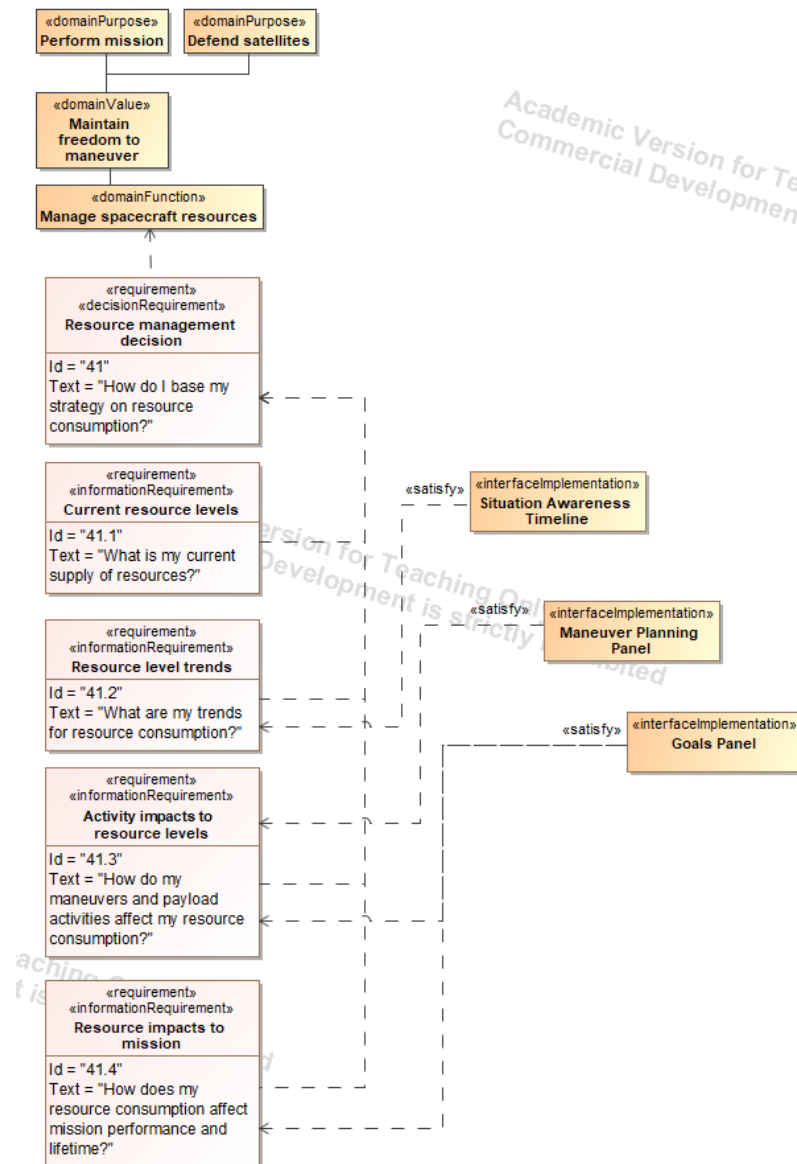


Figure 24 Resource Management Decision Requirement

The “mission plan” decision in Figure 25 asks, “How do I execute my mission plan?” This decision supports domain values in the FAH that correspond to threat response and maintaining flexibility while progressing towards mission objectives. To make this decision, the operator would have to create plans for multiple spacecraft maneuvers.

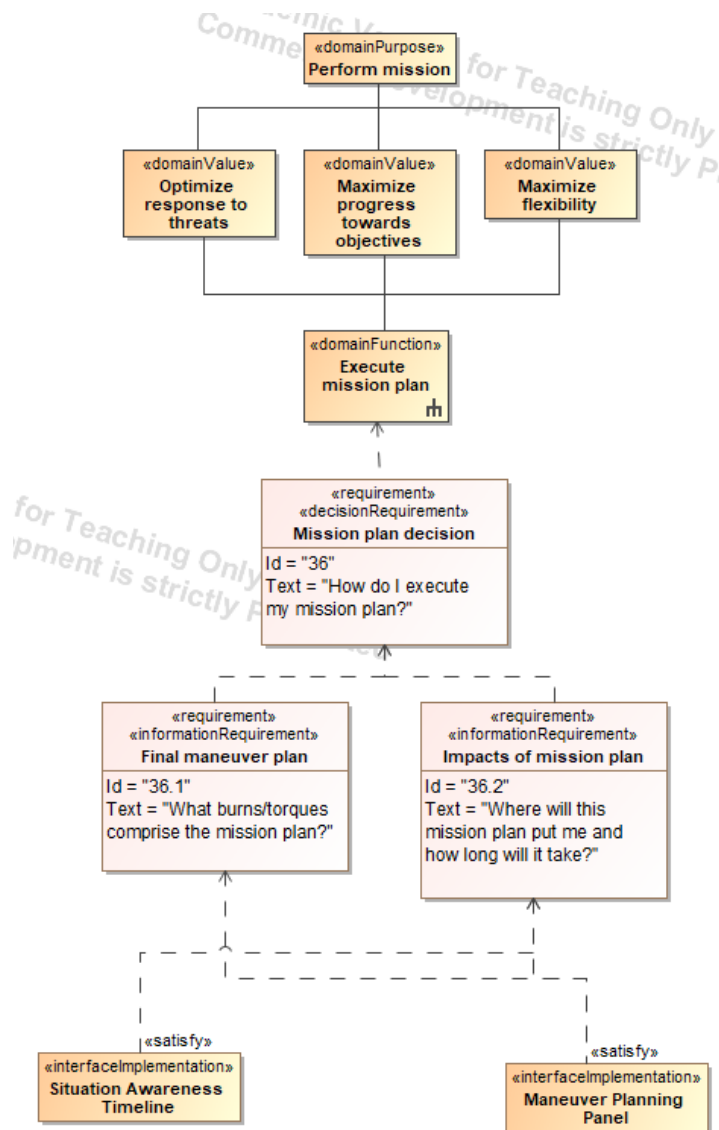


Figure 25 Mission Plan Decision Requirement

The “orbital environment” decision in Figures 26 and 27 asks, “Is my understanding of the orbital environment sufficient?” This decision supports domain values in the FAH that correspond to threat awareness and response. To make this decision, the operator would need to be able to visualize the current and future states of the orbital environment while accounting for upcoming friendly and adversary maneuvers.

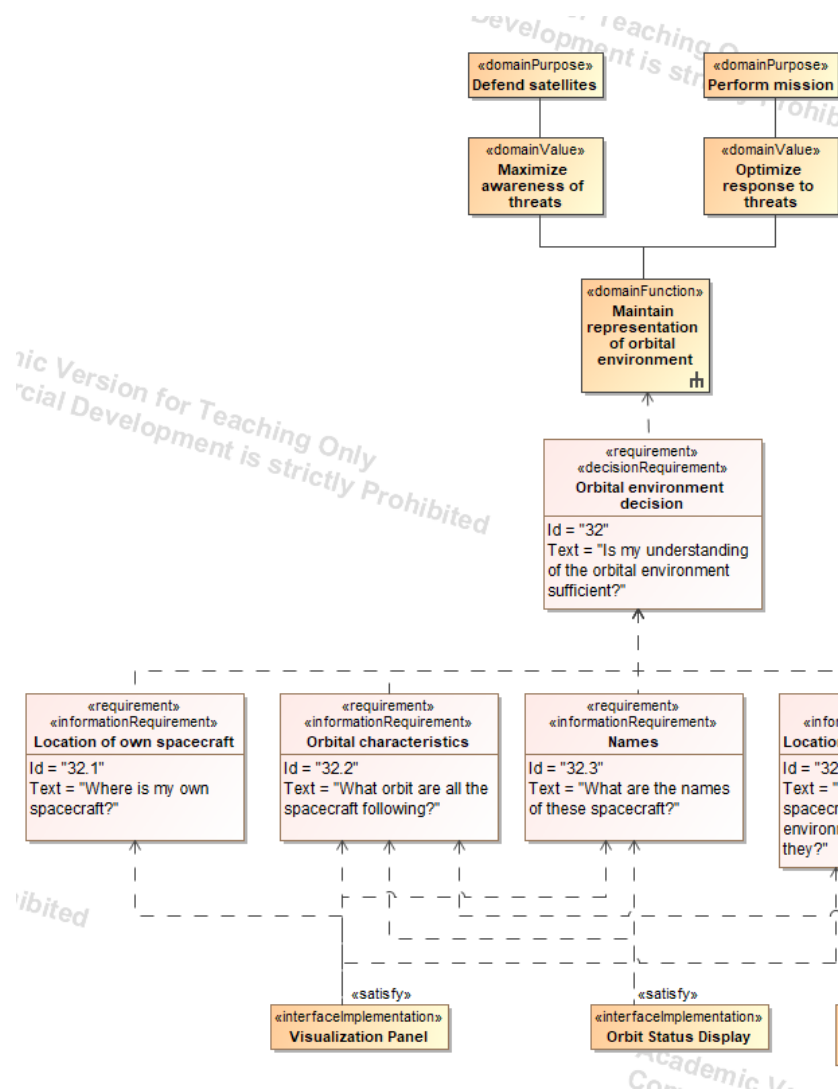


Figure 26 Orbital Environment Decision Requirement (Left)

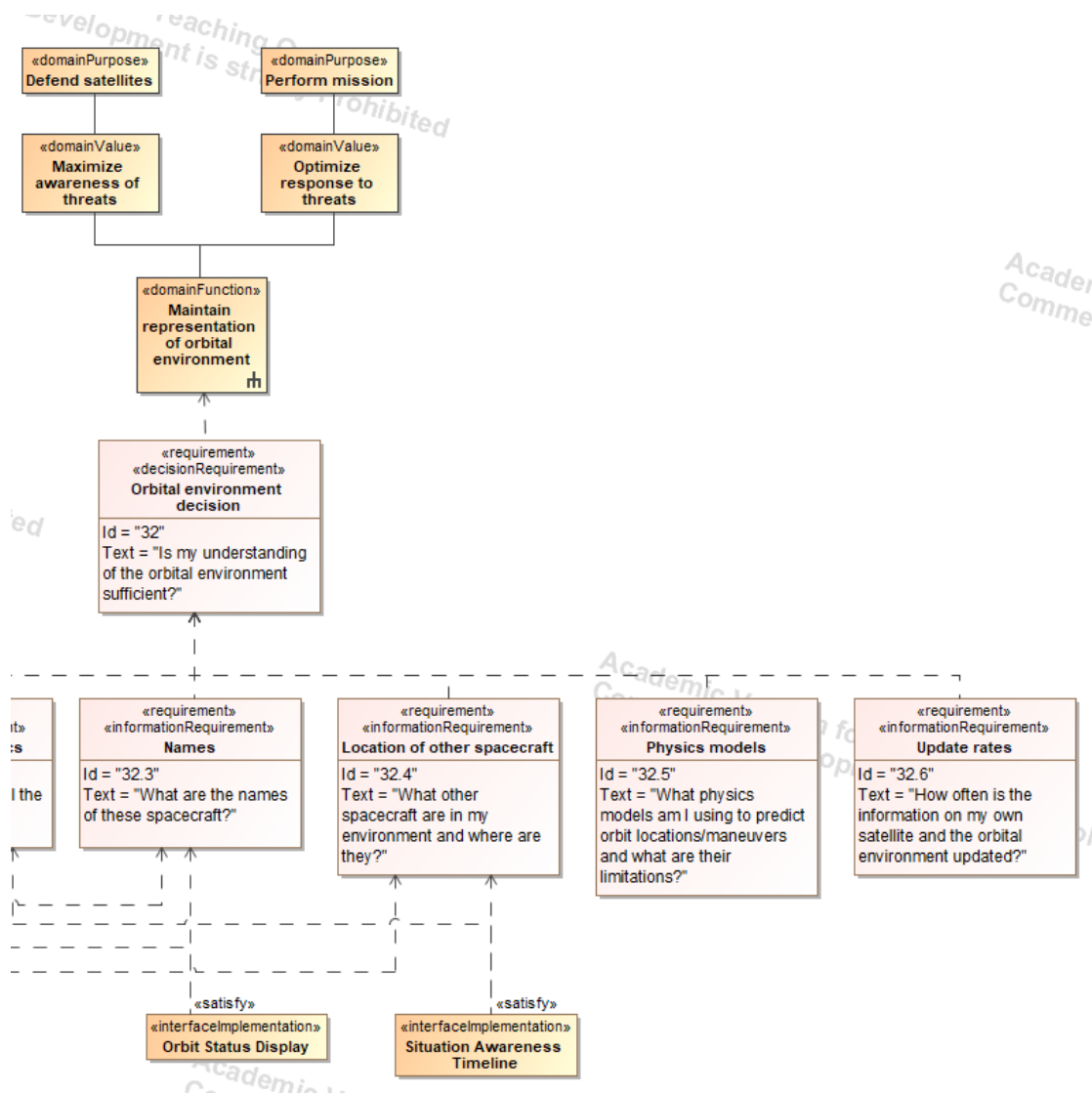


Figure 27 Orbital Environment Decision Requirement (Right)

The “active/passive defense” decision in Figure 28 asks, “Which of my defensive options do I use?” This decision supports domain values in the FAH that correspond to mission accomplishment, maintaining command and control, maintaining spacecraft and payload safety, and threat response. The use of active or passive defenses was not in the scope of the space operator UI and is not associated with any interface implementation suggestions.

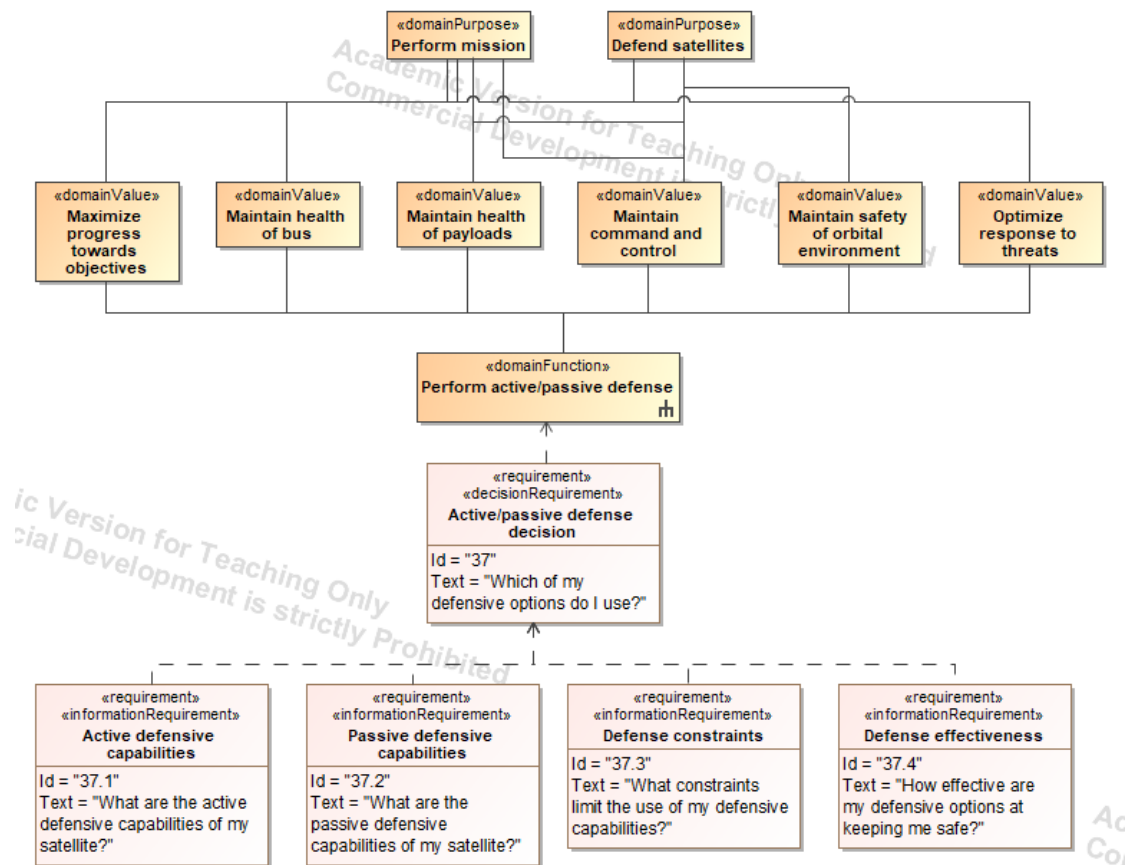


Figure 28 Active/Passive Defense Decision Requirement

The “safety” decision in Figure 29 asks, “Am I safe to continue my mission?”

This decision supports domain values in the FAH that correspond to knowledge of the orbital environment, spacecraft and payload health, threat awareness, and freedom to maneuver. To make this decision, the operator would need to be able to visualize the results of their maneuvers as well as the current and future status of their spacecraft and its payloads.

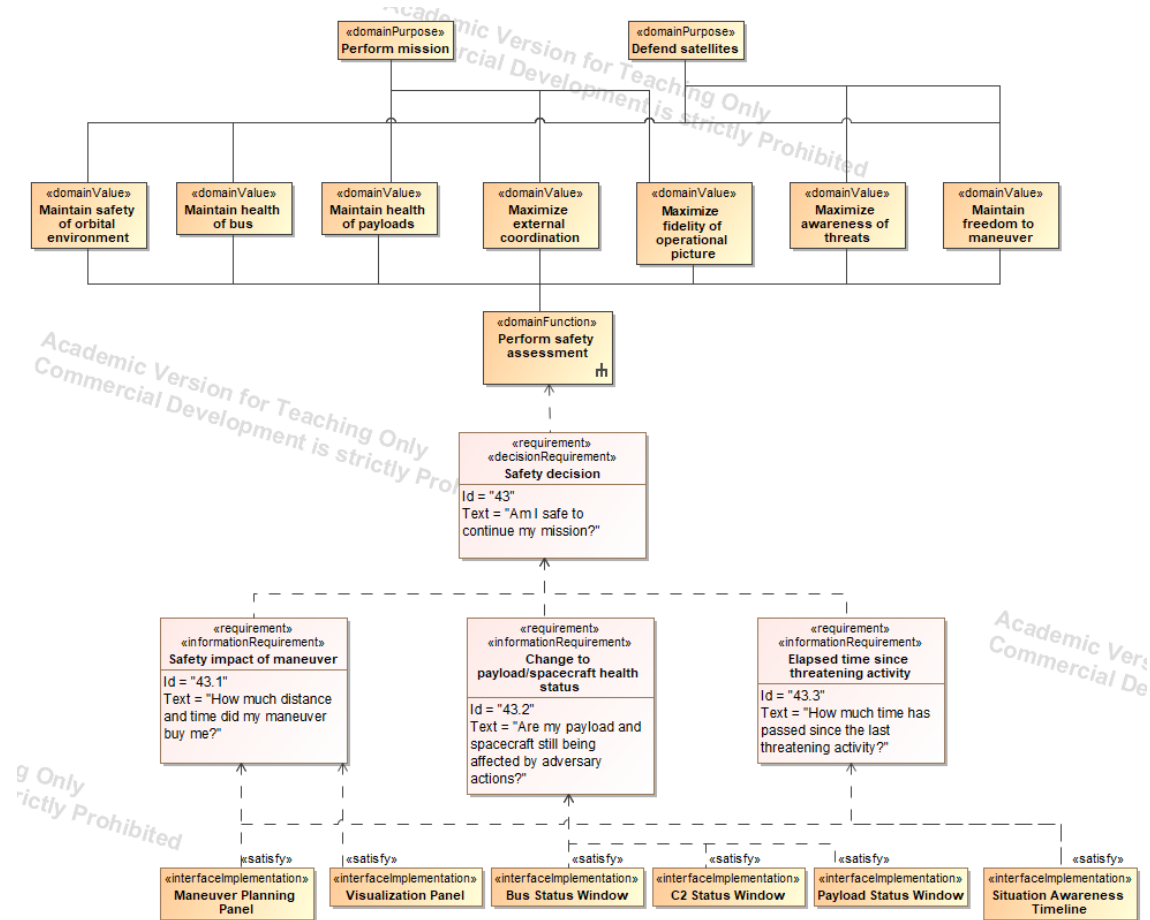


Figure 29 Safety Decision Requirement

The “maneuver” decision in Figures 30 and 31 asks, “What kind of maneuver do I want to execute?” This decision supports domain values in the FAH that correspond to knowledge of the orbital environment, spacecraft and payload health, threat awareness, and freedom to maneuver. To make this decision, the operator would need interfaces that provide the information necessary to plan maneuvers, visualize their impacts, and understand any constraints.

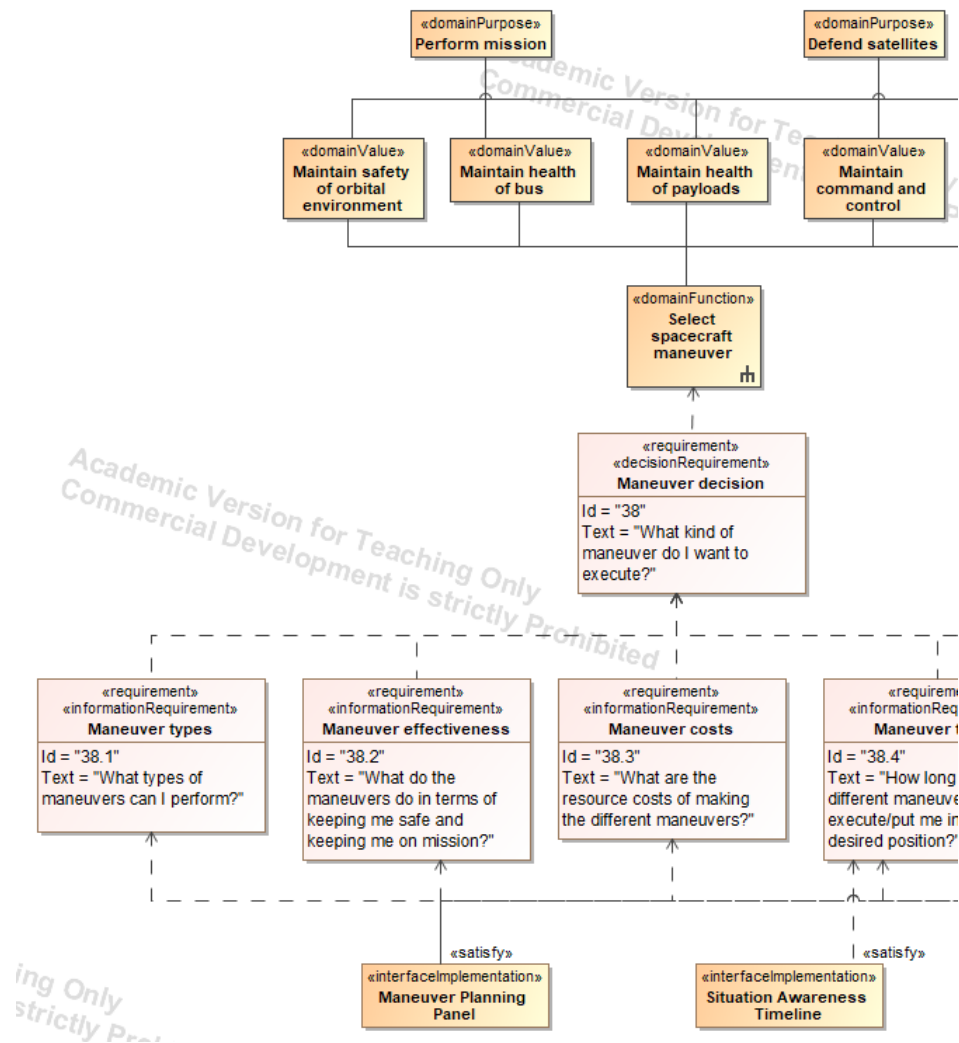


Figure 30 Maneuver Decision Requirement (Left)

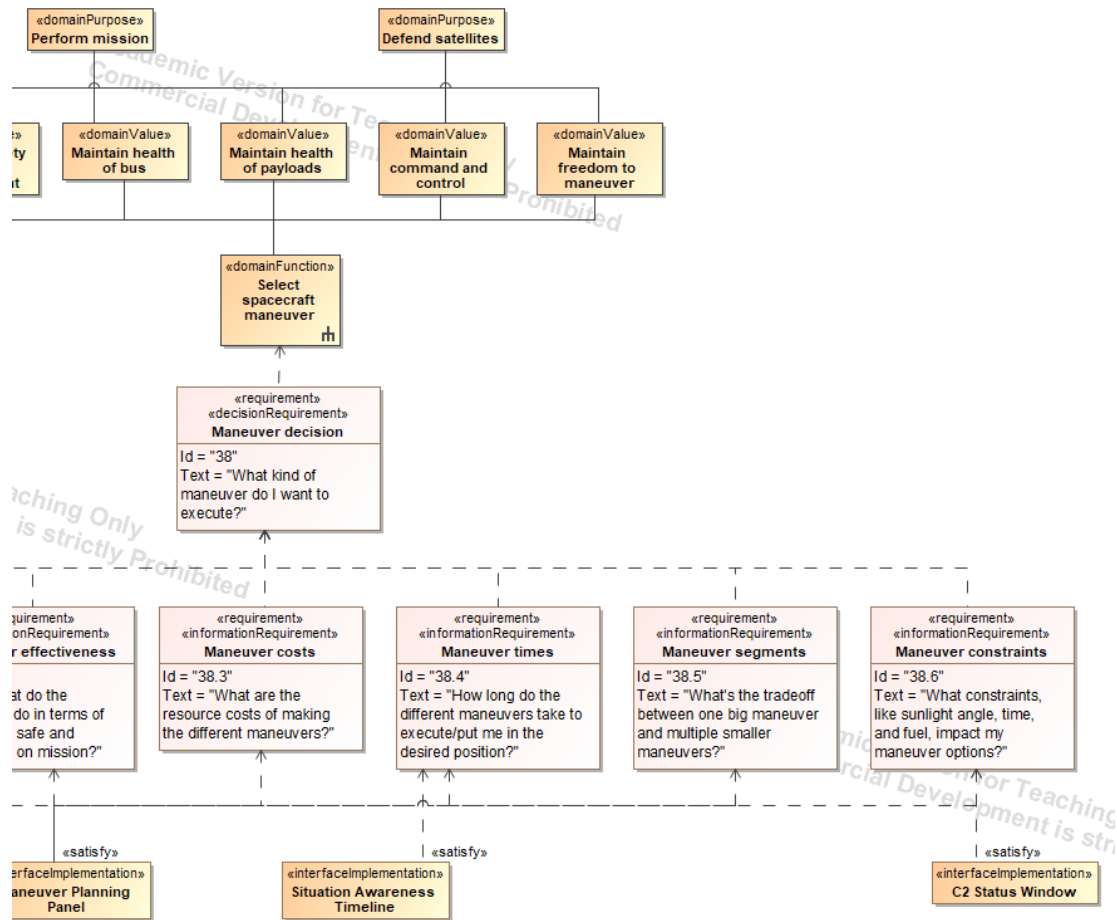


Figure 31 Maneuver Decision Requirement (Right)

Discussion and Conclusions

The operator interview process was extremely useful and provided key insights that were not discovered during the domain literature review. Operators described tradeoffs between major goals that dramatically influence how they perform their work in both nominal and stressful conditions. These kinds of tradeoffs must be represented in a space operations interface to assist operators with making those kinds of decisions. Representing the complicated interplay between physical actions and goal-seeking ability is one kind of software challenge that EID methods can assist operators with. Using an

iterative, cognitive systems engineering process for identifying operator needs and deriving requirements ensures that designers understand how to prioritize pieces of software functionality.

The FAH diagrams help system designers visualize the complexity of the system they are creating for operators. In this research, the diagrams showed operator goals that were both complementary and contradictory depending on the situation. Some space operations functions contributed to a much larger set of goals than others, and likewise needed the cooperation of a greater number of physical and organizational components. To assist operators with performing these functions, a space operations UI would need to depend on multiple, coordinating parts to show the relationships between low-level interactions and high-level goal fulfillment. Although it's important to keep track of the battery level of a satellite, that is a much simpler task than integrating payload status, host vehicle status, the orbital environment, and intelligence reports to decide whether a friendly asset is in a condition to safely continue its mission.

The requirements diagram showed how all the domain functions in the FAH could be broken down into individual operator decision requirements backed by one or more information requirements. In this representation, the space operator interface assumes the role of providing operators the right information at the right time to assist them with making decisions like how to maneuver their spacecraft or whether another spacecraft is acting suspiciously. In a time-sensitive situation, space operators would need to sift through a large amount of data to make decisions with potentially long-term impacts on their spacecraft. In the case of strategically managing resource consumption, different pieces of information may be provided by multiple interface components. This poses the

challenge of how to create an interface that provides multiple perspectives on a problem without overwhelming the operator. Using rapid UI prototyping to gain operator feedback on design decisions would be an effective way of resolving interface issues early in the design process.

Chapter Summary

This chapter described how space operator goals and requirements were modeled with the assistance of cognitive systems engineering methods, including the ACTA interview process. The interviews were analyzed for concepts and observations that were based on the operators' personal experience and not described in the available domain literature. These data formed the initial inputs into the space defense operations FAH diagram, a model of domain goals and how operators attempt to meet those goals. Domain functions from the FAH diagram were used to derive decision and information requirements necessary for a space defense operations UI to assist operators with their work. This process was designed to be iterative, so the goal models and system requirements can be continually updated based on research, new requirements, and operator feedback on the design process. Chapter IV will describe the latter two portions of the process in Figure 4, UI prototyping and operator feedback. In this phase, the space operator system requirements and interface implementation specifications are turned into semi-functional prototypes. Elements of these prototypes are traced to elements of the system model in Cameo, along with operator feedback on how well they satisfy the workplace goals.

IV. User Interface Design Prototyping with Feedback into Cognitive System Models

Chapter Overview

This chapter describes the process of creating user interface (UI) prototypes from a system model containing operator goals, processes and functions and derived decision/information requirements. The suggested interface implementations for each system information requirement are turned into semi-functional interface prototypes using prototyping software. The resulting space operator interface prototype is demonstrated to operators to gain feedback on how well the UIs meet their requirements and support goal-seeking activity. Their feedback is then prepared for incorporation back into the system model as updated domain goals, processes, and UI designs. This process constitutes the latter two steps of the iterative cognitive system design process in Figure 4, UI Prototyping and Operator Feedback.

Methodology

Selecting Prototyping Software.

Rapid UI prototyping allows system designs to demonstrate system functionality and user interactions to operators early in the design process when changes can be made with the least impact on budget and schedule. This study also shows how attaining operator feedback through rapid prototyping supports the maturation of workplace goal models, ensuring that the outcomes of the cognitive systems engineering process could be used to improve that process in an iterative fashion. To achieve this, the prototyping software had to support rapid design creation by the researcher with sufficient fidelity to gain meaningful feedback from the potential users.

Three popular commercial software packages used for UI prototyping were considered for this study: Adobe XD, Axure RP 9, and Microsoft PowerPoint (Table 2). It was important for the software to support elaborate UI designs similar to what a user would encounter in the real world, and for those designs to support complex user interactions like dragging, sliding, typing and passing information from one screen to another. Newer software products emphasize internet-based collaboration, but extensive support for local storage and publishing was necessary for products the DoD considers for official use only (FOUO). Most importantly, the software had to support rapid content creation by designers with little programming familiarity. Based on these criteria, Axure RP 9 was selected for this research because it allowed for the design of variable-based and state-based user interactions that made the prototypes perform more like real software. An interaction that resulted in different outcomes based on a spacecraft's current fuel load, for example, could not be created in either Adobe XD or Microsoft PowerPoint.

Table 3 Comparison of User Interface Prototyping Software

Criteria	Adobe XD	Axure RP 9	Microsoft PowerPoint
Supports Elaborate UI Designs	X	X	
Models Complex User Interactions		X	
Supports Rapid Content Creation	X	X	X
Local Storage to Support FOUO Products		X	X

Axure RP 9 was selected to create functional UI designs fulfilling each of the system information requirements. Software logic was attached to each UI component to simulate user interactions and their outcomes in an example space defense scenario. All UI components were integrated into a full space operator UI example to be used for further development and test.

Creating and Applying Logic to Interface Components.

The prototyping software allowed UI components to be built from scratch using primitive shapes or picked from a library of common UI patterns. Behaviors were assigned to the UI components to simulate common user interactions with familiar displays, shown in Figure 32. For example, the spacecraft status UI was built in the form

of an “accordion” panel. This form of UI allows users to see a high-level status overview that can be expanded into a more-detailed display if additional attention is necessary as shown in Figure 33. Markers on the accordion panel will call the user’s attention to potential current or upcoming problems and status changes when the detail is not shown, providing necessary information while reducing clutter. Adding logic to the UI components allowed them to interact with each other, allowing for the creation of realistic scenarios.

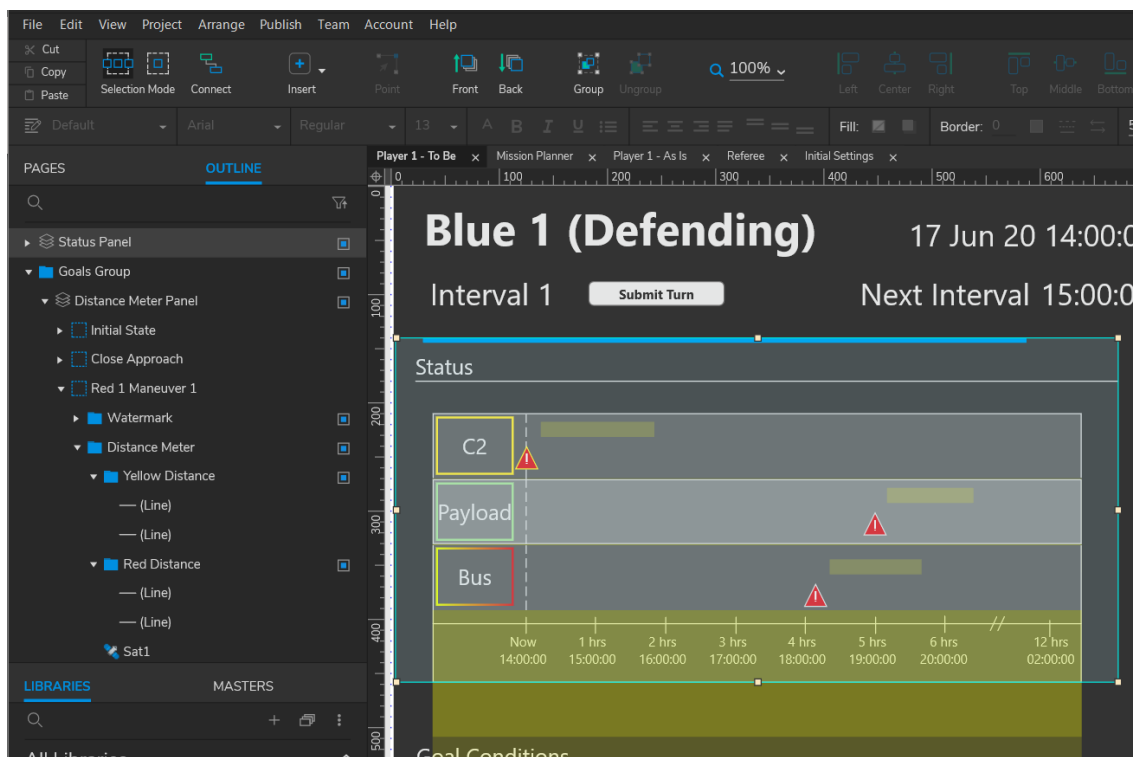


Figure 32 Screenshot of Status UI Prototype Development Using Axure RP 9

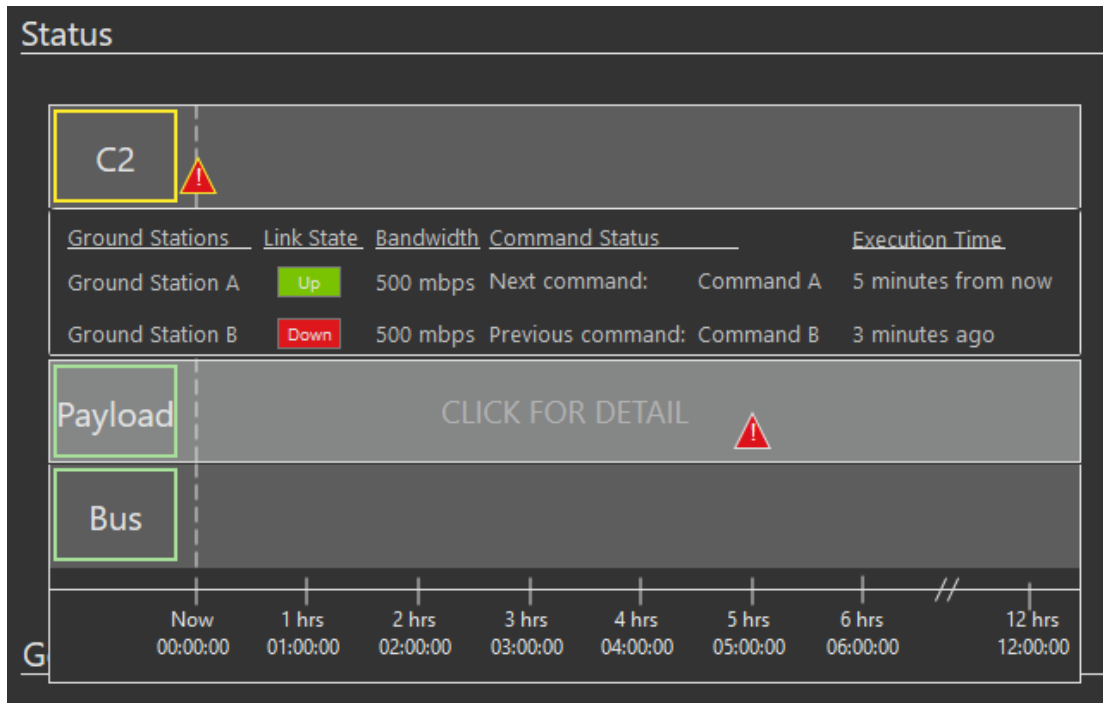


Figure 33 Expanded Accordion Panel

Individual UI elements, such as images or shapes, were assigned logic and behaviors to simulate real-world interfaces. For example, the Simulation Awareness Timeline shown in Figure 34, illustrates planned and upcoming spacecraft maneuvers, hovering over a spacecraft icon shows the name tag associated with the maneuver. Right clicking on the icon displays a context menu with options for editing the maneuver. Left clicking on the icon displays a context menu with options for editing the maneuver. Left clicking the icon performs a much more elaborate action: it simulates the act of traveling forward in time, so the user can visualize the future state of the orbital environment based on predictive calculations. In a real application this would be performed algorithmically. In the prototype, this is accomplished by triggering all impacted displays to move to a pre-programmed state for the selected maneuver. Although the prototype does not have

the fidelity of a mathematical simulation, the scripting functions allow it to approximate real-world behavior well enough to present operators with example scenarios for feedback on the design.

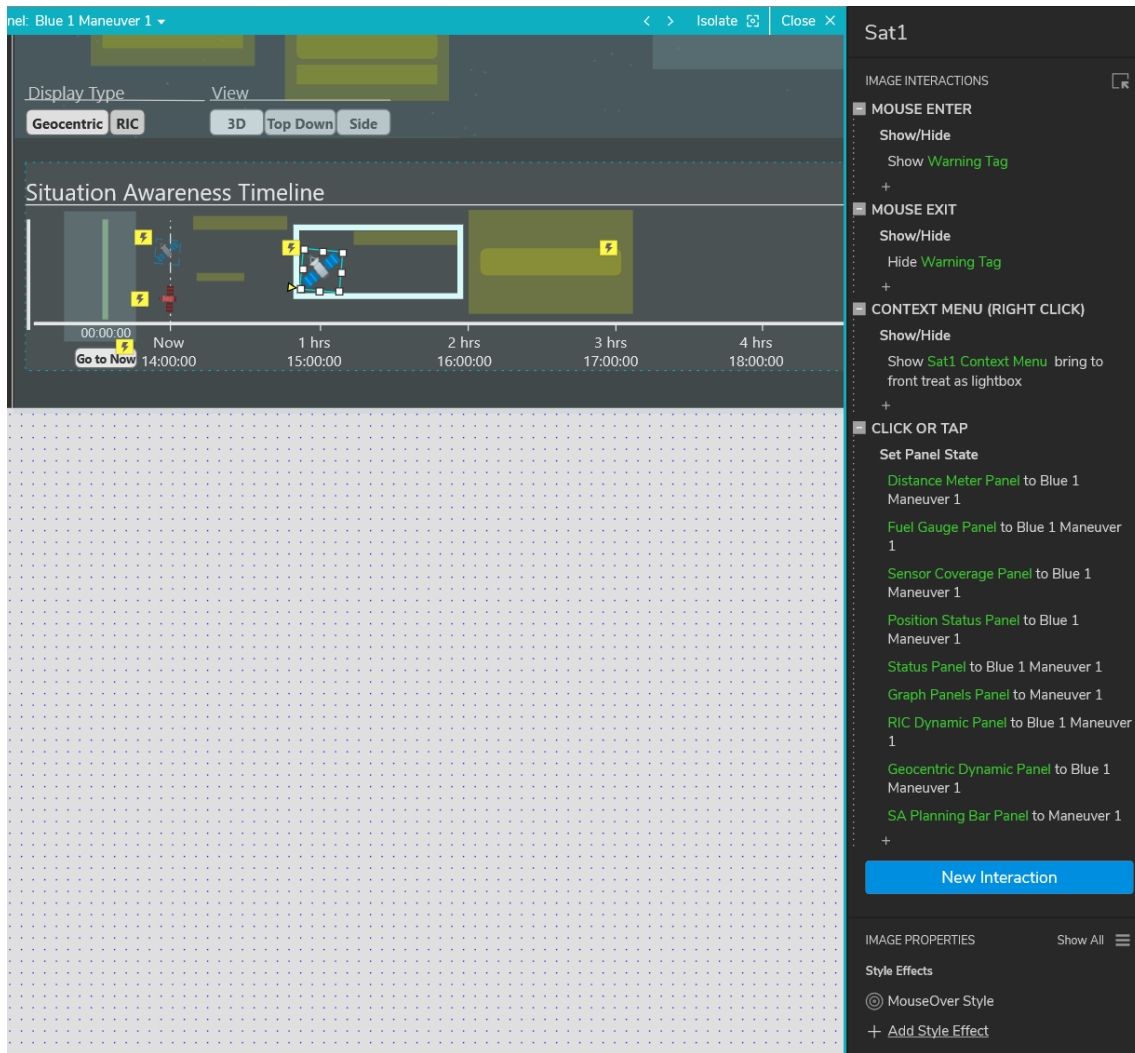


Figure 34 Defining User Interaction Logic in Axure RP 9

Traceability of UI Implementation.

Once completed, the interface designs were imported to Cameo and assigned to the interface implementation elements for each decision requirement. A single interface design could satisfy one or more decision requirements, each with one or more information requirements. Showing the graphical interface and noting how each portion meets these requirements provides verification that the space operator UI has addressed that need. Multiple UIs can be assigned to the requirements to show different options for further exploration in the prototyping software. The attachment of prototype UIs system requirements provides full traceability from high-level goals to specific functionality early in the design process before proceeding to full development, which allows systems engineers to incorporate changes without costly rework. To obtain feedback on how well the UIs met space operator needs, a full scenario was presented to individuals with space operations experience.

User Experience Survey.

Participants were selected from members of the Air Force Institute of Technology (AFIT) Orbital Warfare Research Group. Each participant had spent one four-year tour in a space operations-related role for the USAF. Once selected, they met individually with the researcher, who walked through a space operations scenario built using the UI. The researcher asked probing questions to assess how effective the participant perceived the UI prototype to be at simplifying cognitive tasks and providing valuable situation awareness. The full scenario description and list of probing questions are in Appendix B.

Results

Ecological Interface Designs for Spacecraft Maneuver Planning.

To assist operators with making goal-oriented decisions in the space operator interface, EID methodologies were used in the visual and behavioral design of several interface components. EID supports challenging decisions, such as selecting evasive maneuvers based on their potential impacts to the mission, by allowing operators to examine problems from the perspective of learned behaviors and perception (skill-based behaviors), the application of rules according to cues and signals (rule-based behaviors) and complex problem solving using domain knowledge (knowledge-based behavior) (Vicente & Rasmussen, 1992). In the spacecraft maneuver selection window, users may craft maneuvers in a traditional, rule-based manner with specific parameters for thruster impulses in each direction. They may also take advantage of a new type of maneuver selector based on a desired goal state for their spacecraft as shown in Figure 35: attaining a desired level of sensor coverage, separating themselves from a pursuer, or drawing in close to the other spacecraft to perform a remote inspection. This allows users to focus on high-level goals while the underlying simulation engine calculates and provides different maneuver options for them to select. Each one can be placed on the Situation Awareness Timeline and visually examined for its impacts on mission success criteria as shown in Figure 36. By allowing operators to plan maneuvers based on goal states instead of specific control inputs, operators may find it easier to make complex mission decisions and analyze tradeoffs in a demanding situation.

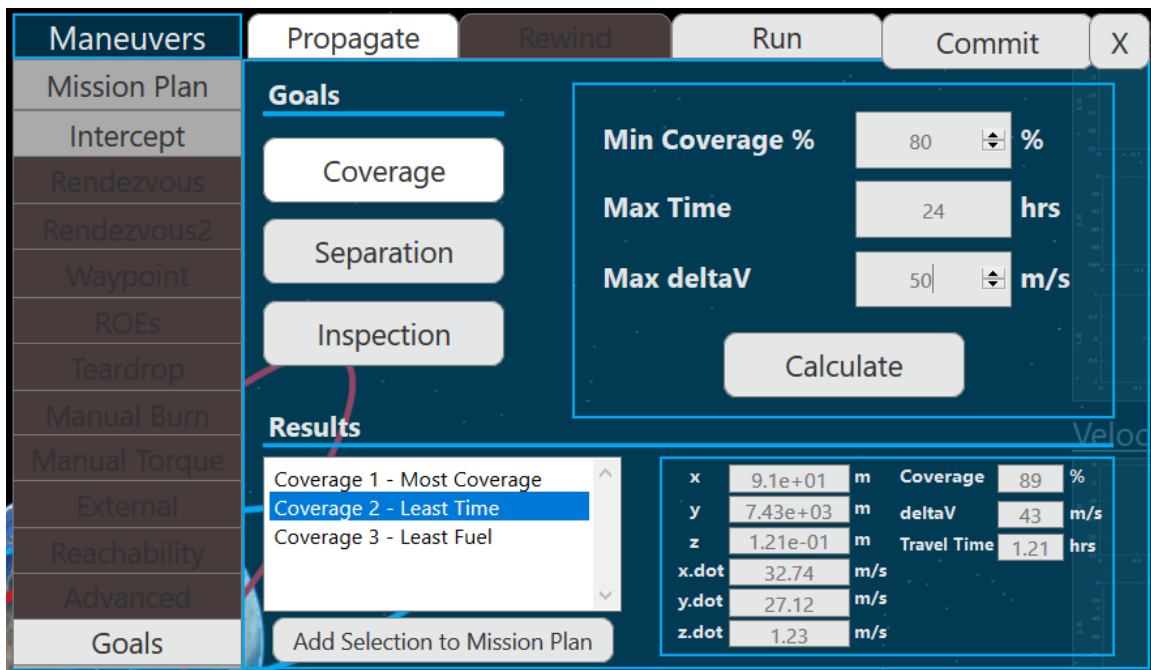


Figure 35 Maneuver Selection from Goal Criteria

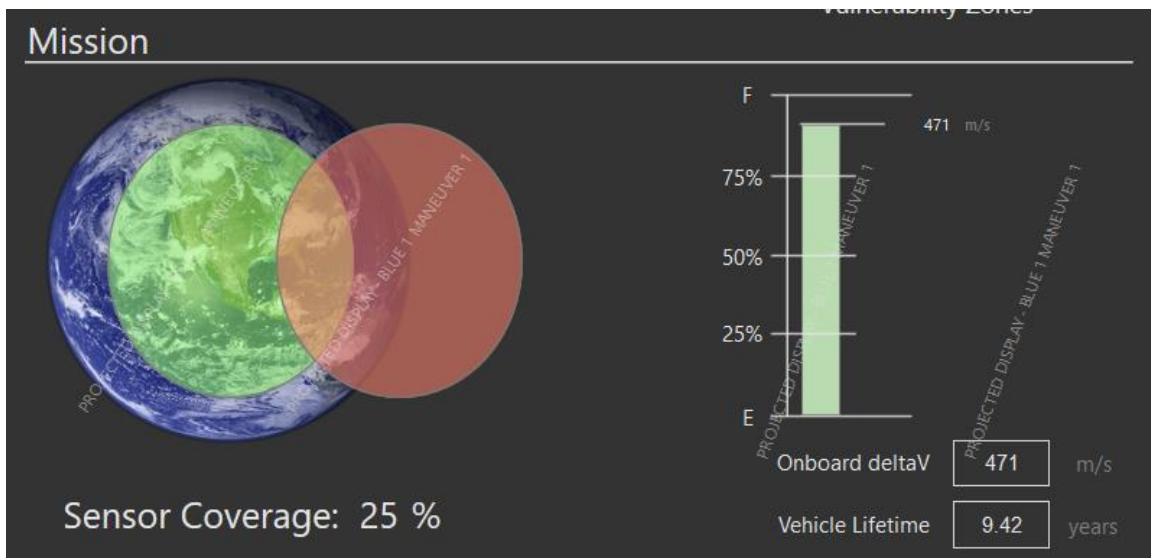


Figure 36 Projected Changes to Mission Constraints

Interface Implementations.

The completed interface designs were assembled into the full UI prototype shown in Figure 37. The prototype included a sample data set generated using an existing space mission management toolset provided by AFIT. Each interface element was interactive within the limits of the scenario developed for the user survey. The individual elements were documented in Cameo and comments were made to show how they satisfied the system information requirements. No interfaces were developed for the “Perform active/passive defense” and “Communicate with friendly spacecraft” values, which were identified in Chapter III and were outside the scope of this research. This left 14% of the 64 information requirements for a future iteration of the software that could account for spacecraft weaponry and simulate the presence of more than two spacecraft. Additionally, the “Command and Control Staff,” “Doctrine library,” and “Briefings” objects were not considered inside the scope of the mission management interface. These objects, though necessary for a complete mission management workflow, would be accomplished by other means in the workplace such as inter-personal communication and general document storage.

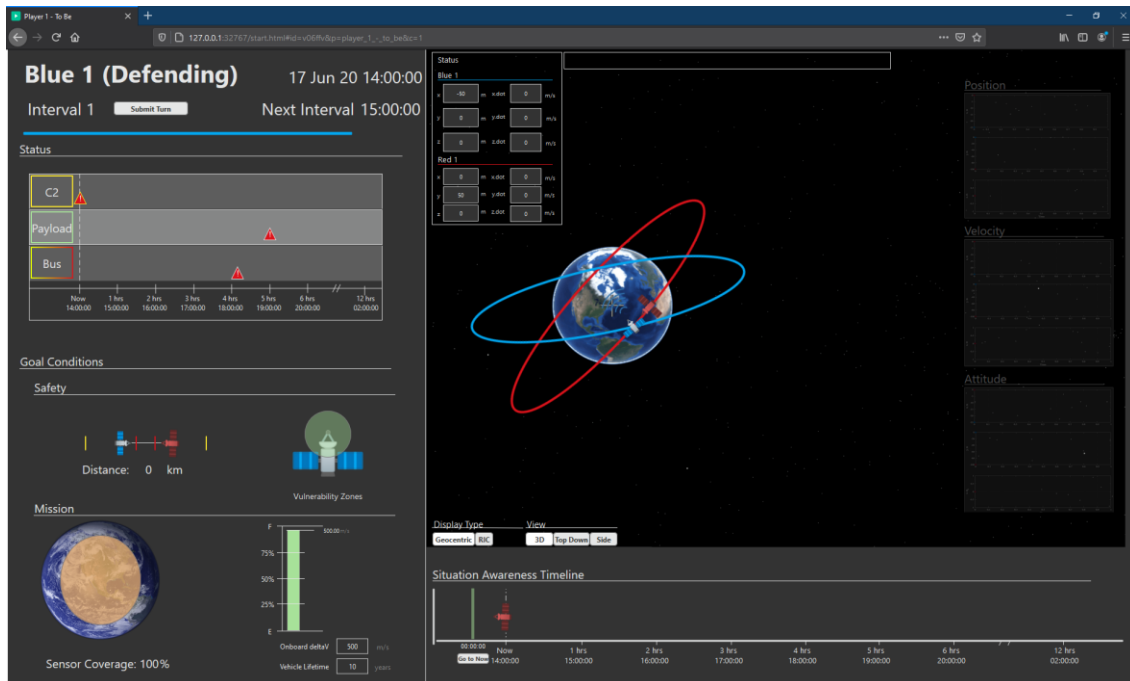


Figure 37 Axure RP 9 Scenario Operating in Firefox Browser 81.0

Figure 38 shows the specification and prototype for the Bus Status Window interface implementation. The information requirements associated with the Bus Status Window are shown above the UI prototype developed in Axure. Lines are drawn between the requirements and labels that describe how the interface displays and user interactions satisfy those requirements. The implementation of EID principles are discussed in terms of Vicente and Rasmussen's EID framework (1990). Figures 39 through 51 describe all interface implementations in the same manner. The Bus Status Window in Figure 38 shows the current and anticipated status for the spacecraft bus. The status icons move left as time advances, with the dashed vertical line indicating the current time. The user may click on the accordion headings to see a more-detailed status report on specific spacecraft

components and trends for state of health indicators. In terms of EID principles, the user is able to directly perceive the evolution of emergency situations as icons appear and change with time. The high-level and expanded detail views offer context and suggest approaches to address an issue, providing the user with flexibility.

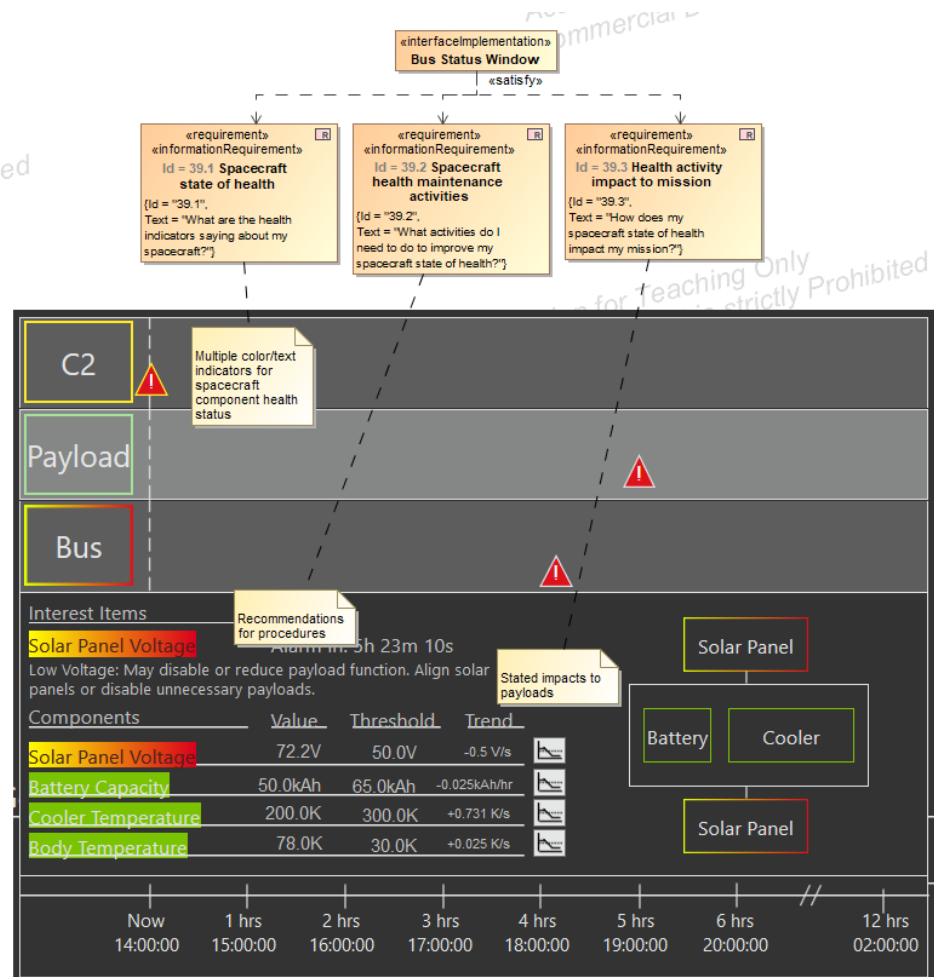


Figure 38 Bus Status Window

The Command and Control Status Window in Figure 39 shows the current and anticipated status for the spacecraft bus. The status icons move left as time advances, with the dashed vertical line indicating the current time. The user may click on the accordion headings to see a more-detailed status report on the ground stations, with link status, bandwidth, and anticipated command execution times.

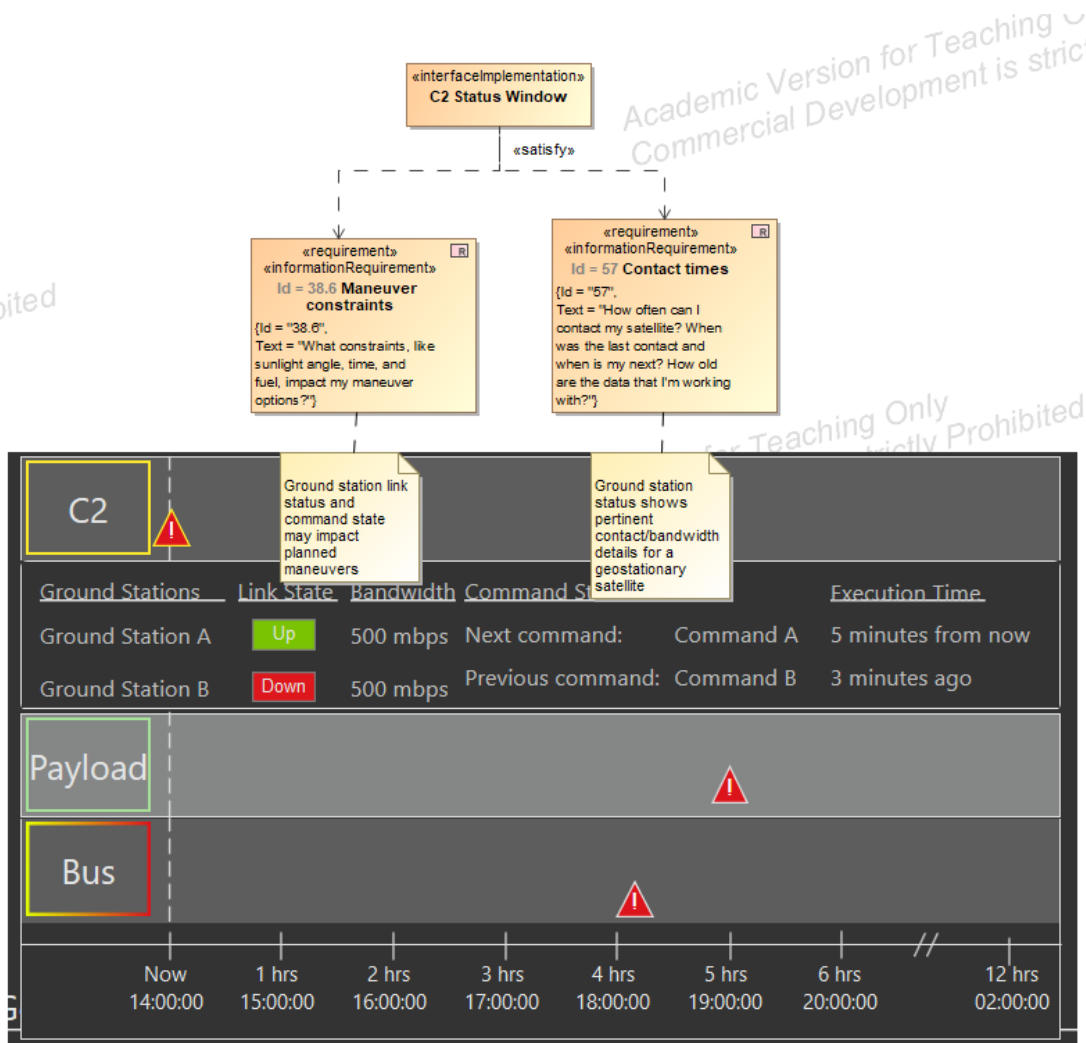


Figure 39 Command and Control Status Window

The Engagement/Vulnerability Zone Display in Figure 40 marks the engagement and vulnerability zones of the different spacecraft. The engagement zone is used to indicate what direction an adversary spacecraft is pointing a payload like a camera, and the distance at which the payload can function. The vulnerability zone indicates an area where a defending spacecraft cannot tolerate the entry of an adversary's engagement zone. With the EID approach, the user is able to perceive the magnitude and orientation of each's spacecrafts zones in the orbital environment. This affords a rapid understanding of the current situation and tradeoffs between potential maneuver options without the need for numerical analysis.

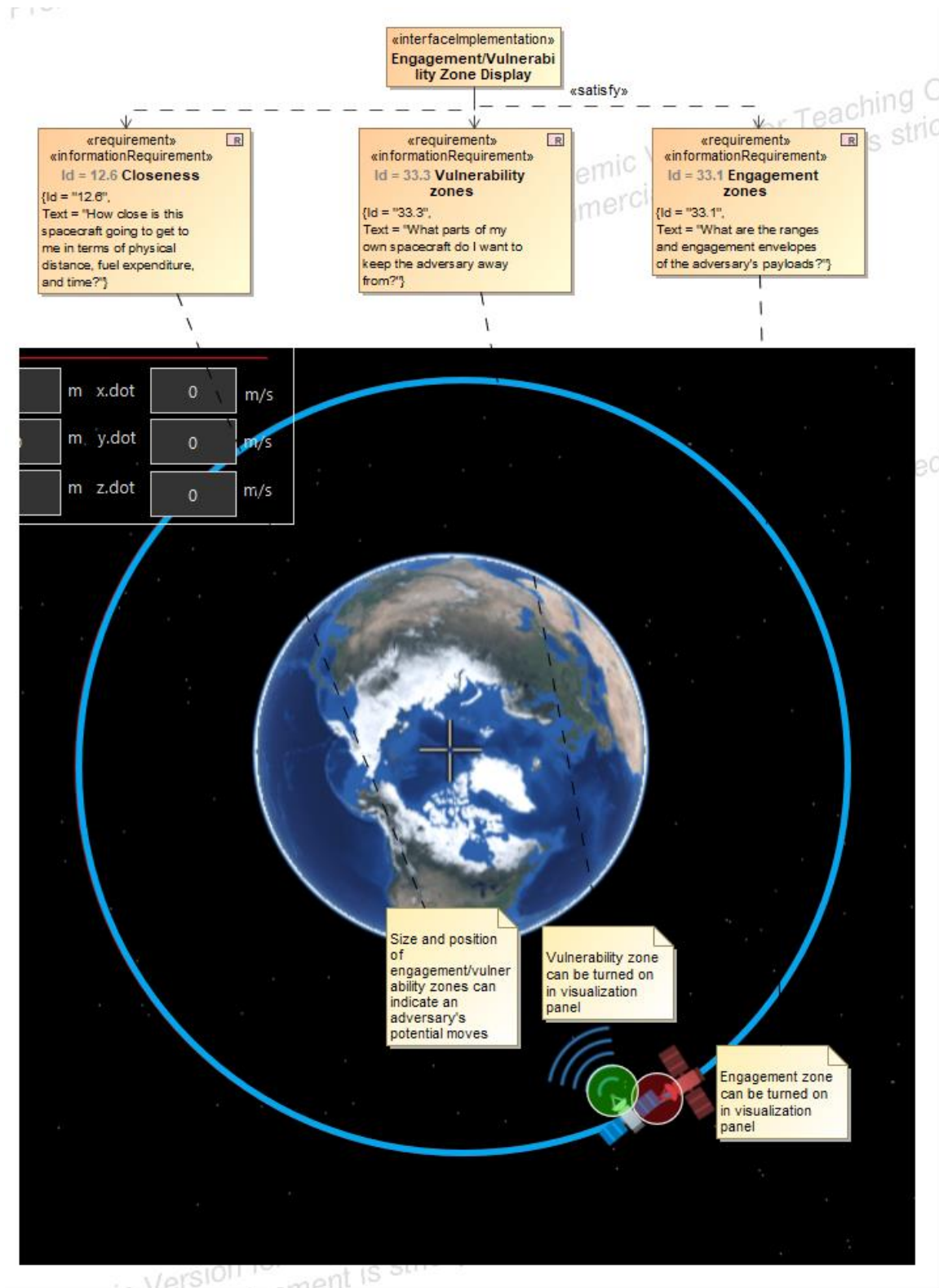


Figure 40 Engagement/Vulnerability Zone Display

The Goals Panel in Figure 41 has four major indicators of goal and mission achievement status. The distance meter in the top left shows the relative distance between a defending and adversary spacecraft. The vulnerability zone meter in the top right shows green, yellow, or red depending on how close the adversary spacecraft's engagement zone is to the defending spacecraft's vulnerability zone. The sensor coverage indicator in the bottom left shows how far the sensor payload has deviated from its ideal target coverage. The fuel gauge in the bottom right shows the fuel reserves and estimated mission life of the spacecraft. In EID principles, these displays show the environmental constraints on the user's decisions. The impact of user interactions on the constraints are shown using domain language: the distance between satellites and the earth-covering footprint of a sensor.

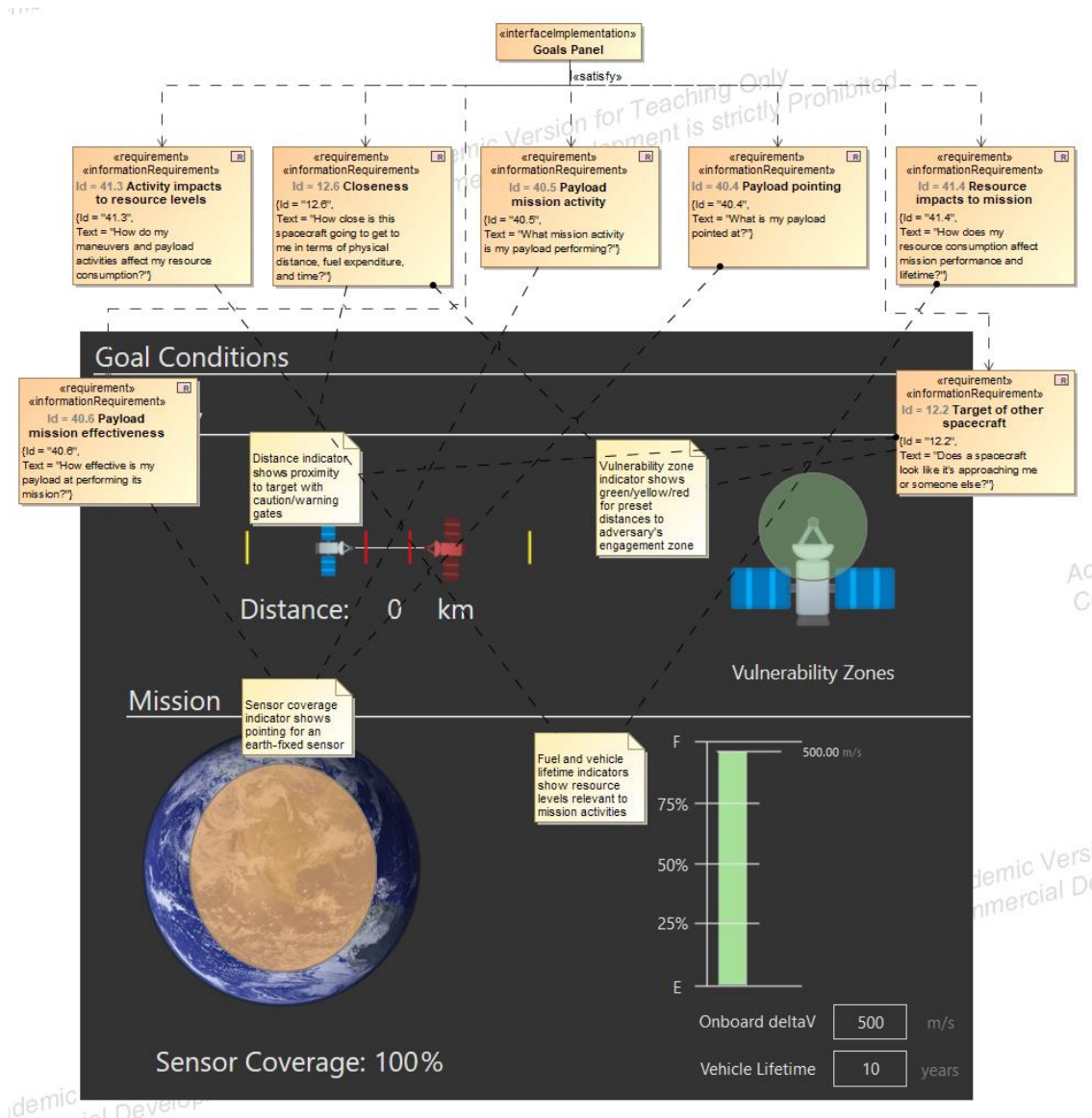
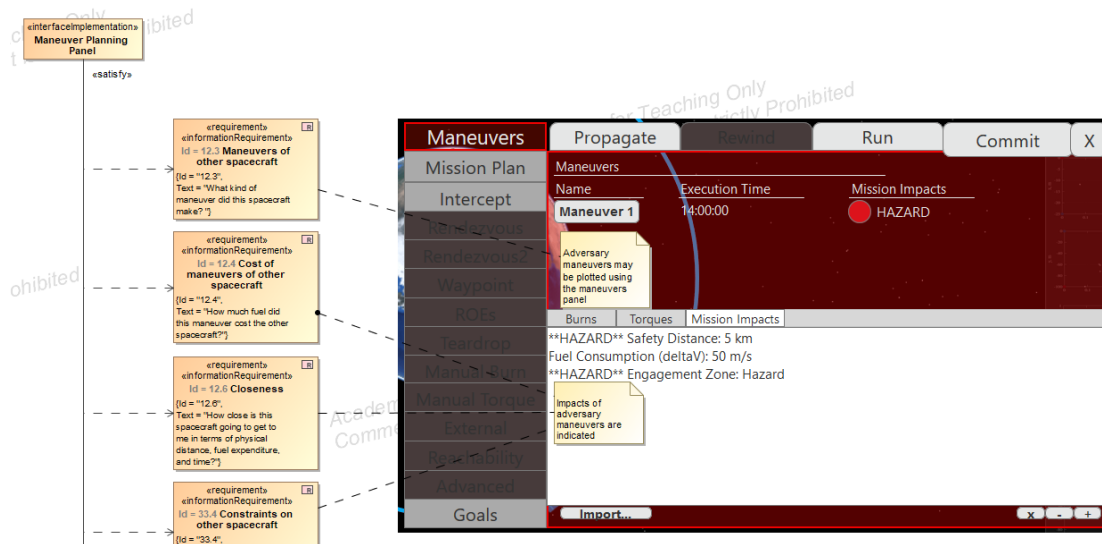


Figure 41 Goals Panel

Figure 42 Intelligence Panel

The Maneuver Planning Panel in Figure 43 allows the operator to plan maneuvers of multiple types with multiple segments for both the defending and adversary spacecraft. Once created, the maneuvers are assessed for the mission impacts associated with their end state in terms of fuel consumption, safety distance, and sensor coverage. The Goal maneuver type allows operators to create maneuver plans based on their desired end state instead of manually inputting mathematical parameters. In EID terms, this interface provides the ability to perform tradeoff analyses in terms of means-end relationships. The environmental constraints, such as fuel or time costs, help the user to determine the type of maneuver to be performed and are directly linked to varying levels of goal accomplishment.



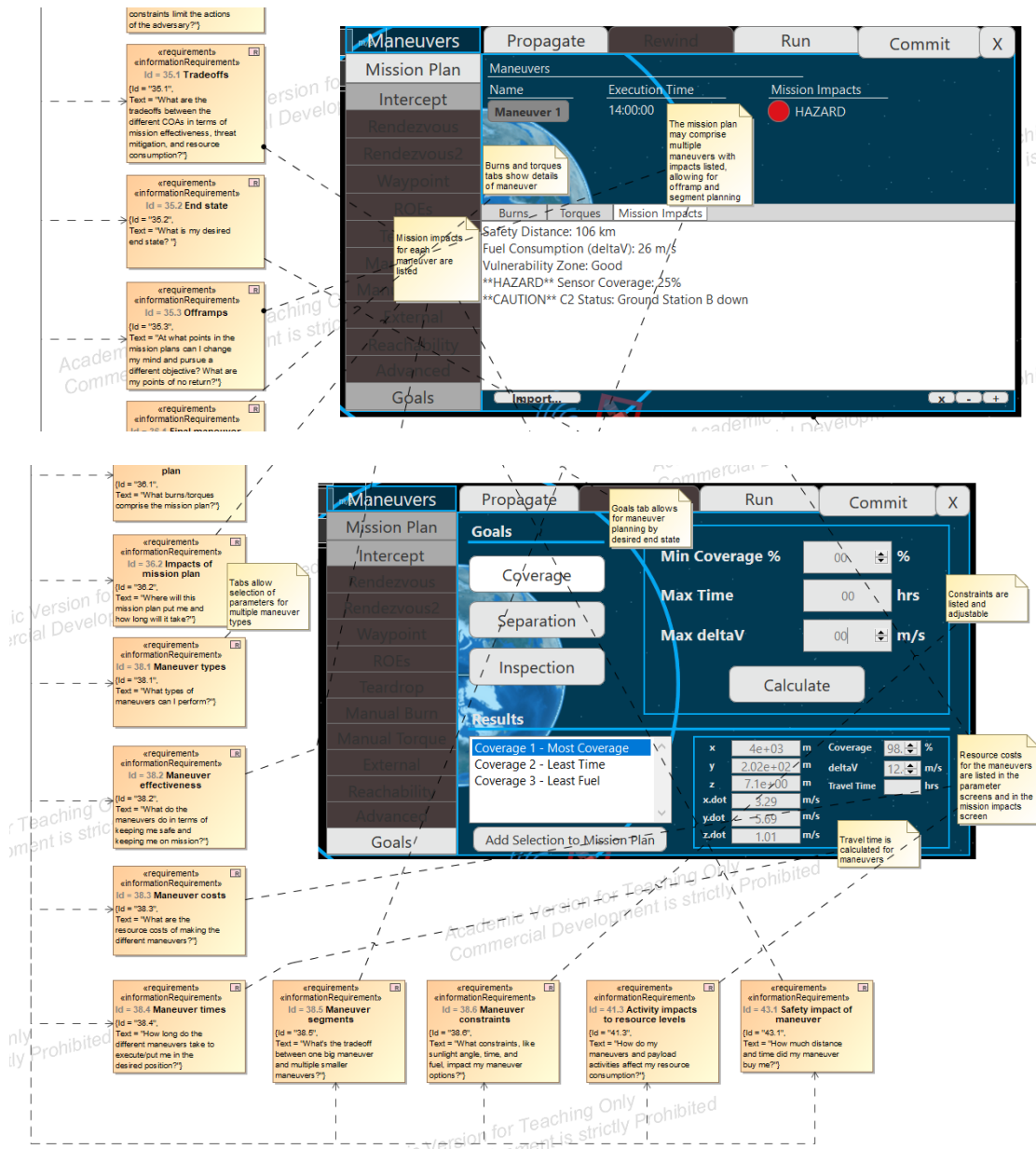


Figure 43 Maneuver Planning Panel

The Orbit Status Display in Figure 44 shows the names and orbital characteristics of the defending and adversary spacecraft.

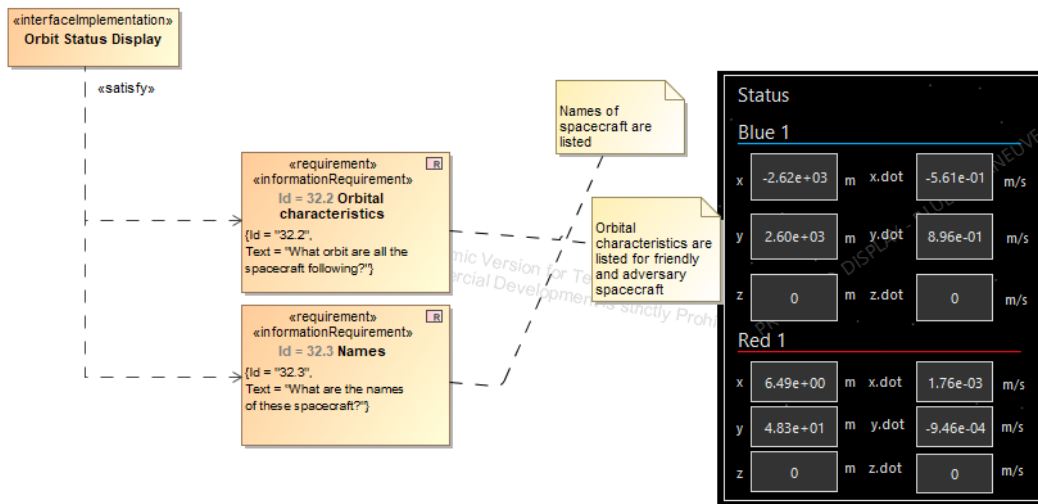


Figure 44 Orbit Status Display

The Payload Status Window in Figure 45 shows the current and anticipated status for the spacecraft bus. The status icons move left as time advances, with the dashed vertical line indicating the current time. The user may click on the accordion headings to see a more-detailed status report on specific payloads and trends for state of health indicators. The user may also perform scheduled and unscheduled payload maintenance tasks. The EID approach for this interface is similar to that in Figure 38.

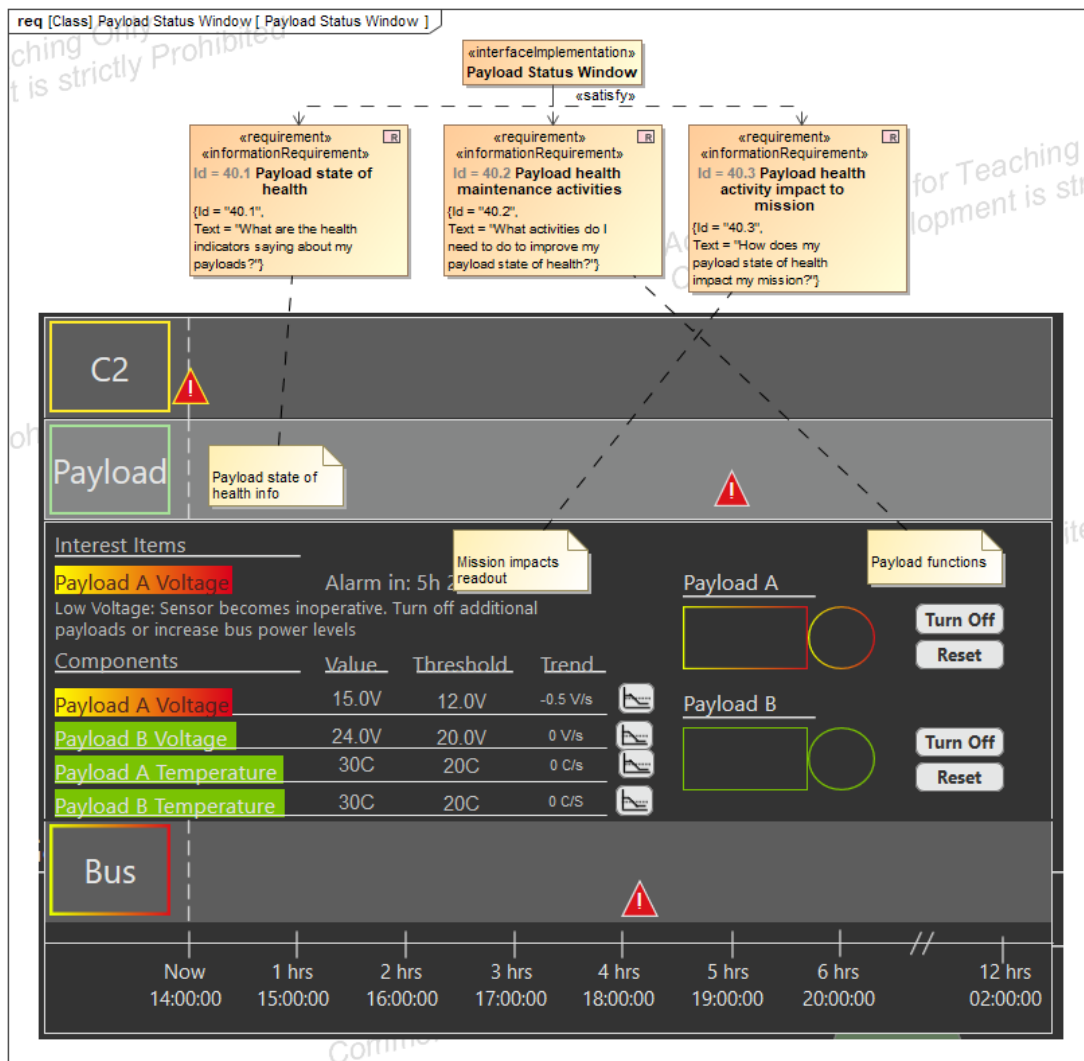


Figure 45 Payload Status Window

The Situation Awareness Timeline in Figure 46 allows the user to predict the future state of the orbital environment, the spacecraft, and their status and mission parameters. Maneuvers plotted with the Maneuver Planning Panel show up on the Situation Awareness Timeline. Clicking on them brings the UI to the predicted state when the maneuvers start or finish. The green bar can be dragged and dropped to pick an arbitrary future state for the UI. When predicting a future state, watermarks are placed on the affected UI elements to ensure that the user is aware that they are not seeing the current state of the environment. In EID principles, this interface provides an implementation of a perception-action cycle. The user selects maneuvers (acts), sees the changes to the future state (perceives), and changes or confirms the maneuver selection as necessary. The future environment is not an invisible quantity but something the user sees and interacts with as part of their decision-making process. The interface also supports knowledge-based decision-making: when there are no explicit rules to guide performance or learned skills to lean on, the operator must rely on their general domain knowledge. The combination of the Maneuver Planning Panel and Situation Awareness Timeline allows operators to visualize and compare maneuvers and orbital states with the parameters they choose.

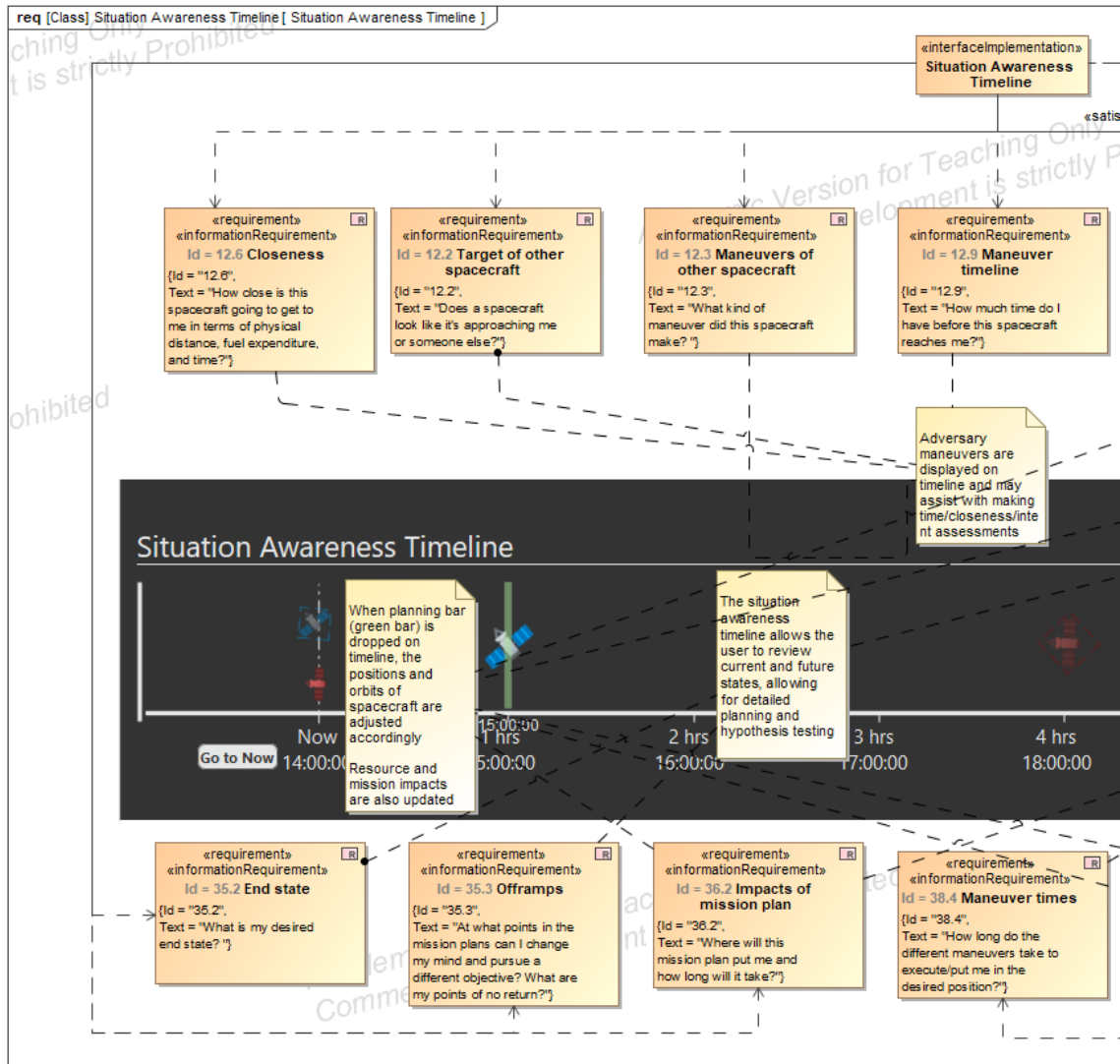


Figure 46 Situation Awareness Timeline

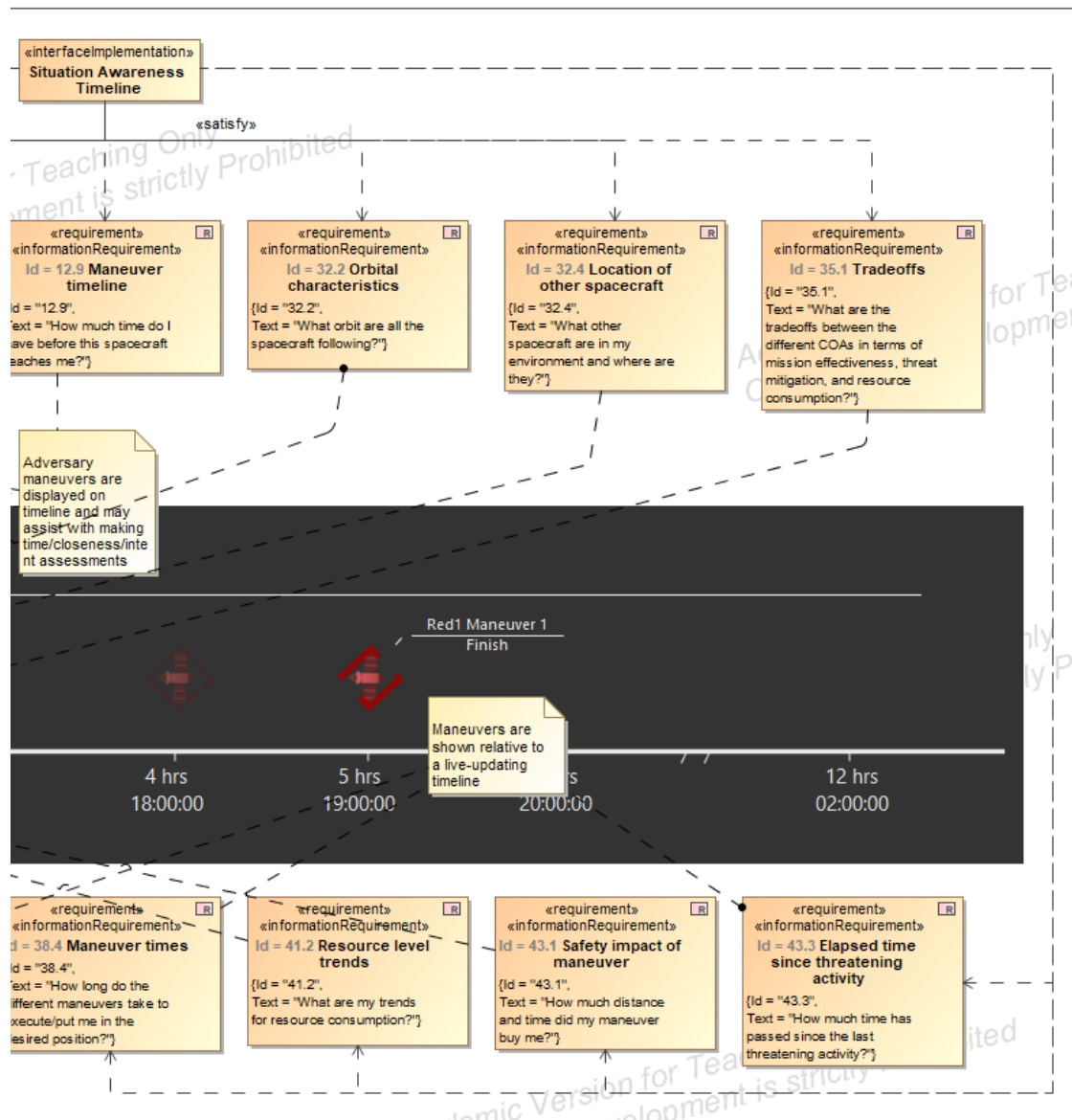


Figure 47 Situation Awareness Timeline (Continued)

The Spec Sheet Display in Figure 48 shows the estimated or known fuel and payload parameters for the adversary spacecraft. These can be adjusted to simulate changing capabilities or as more information is known about the status of the adversary.

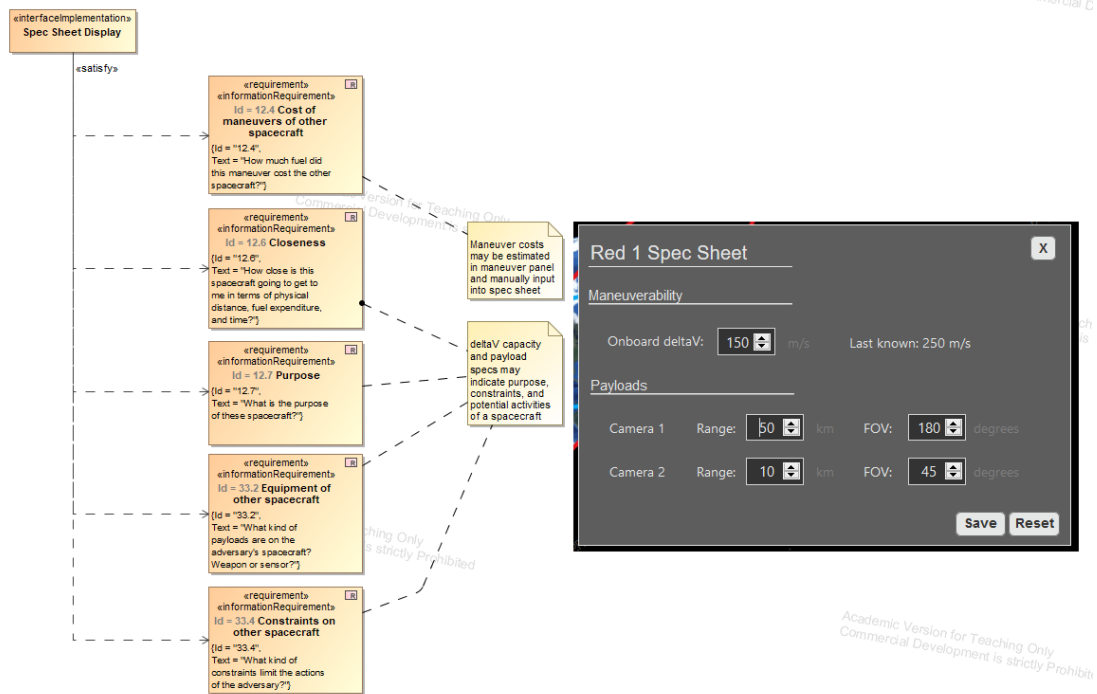


Figure 48 Spec Sheet Display

The Visualization Panel in Figures 49 through 51 contains geocentric and RIC displays of the orbital environment and the spacecraft. Each display can be adjusted in 3D or set to different, fixed coordinate system views. The spacecraft relative position, velocity, and attitude are shown with panels that overlay but do not obscure the main visualization. These visualizations provide familiar perceptual cues to support the perception action cycle while interacting with other interface elements. Relative distances, spacecraft zones, and absolute positions are shown and updated according to changes in the environment or the user's plans.

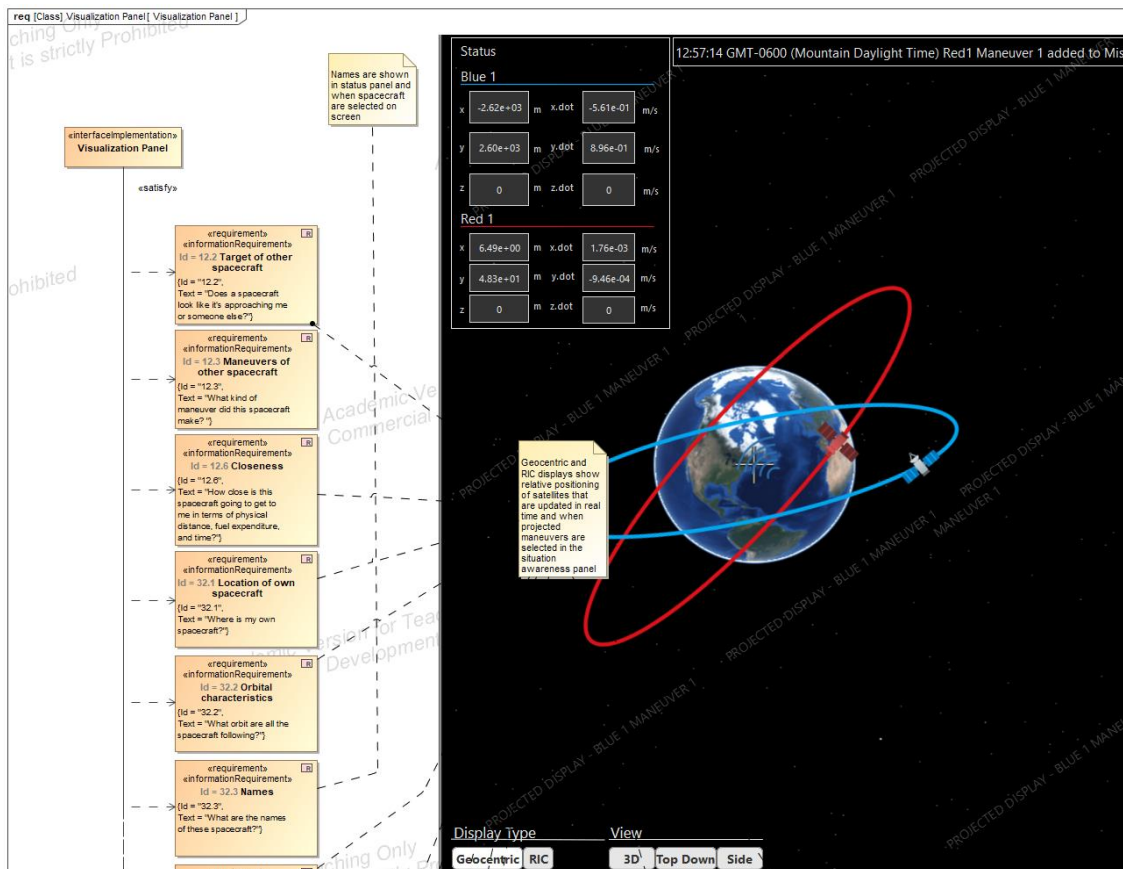


Figure 49 Visualization Panel

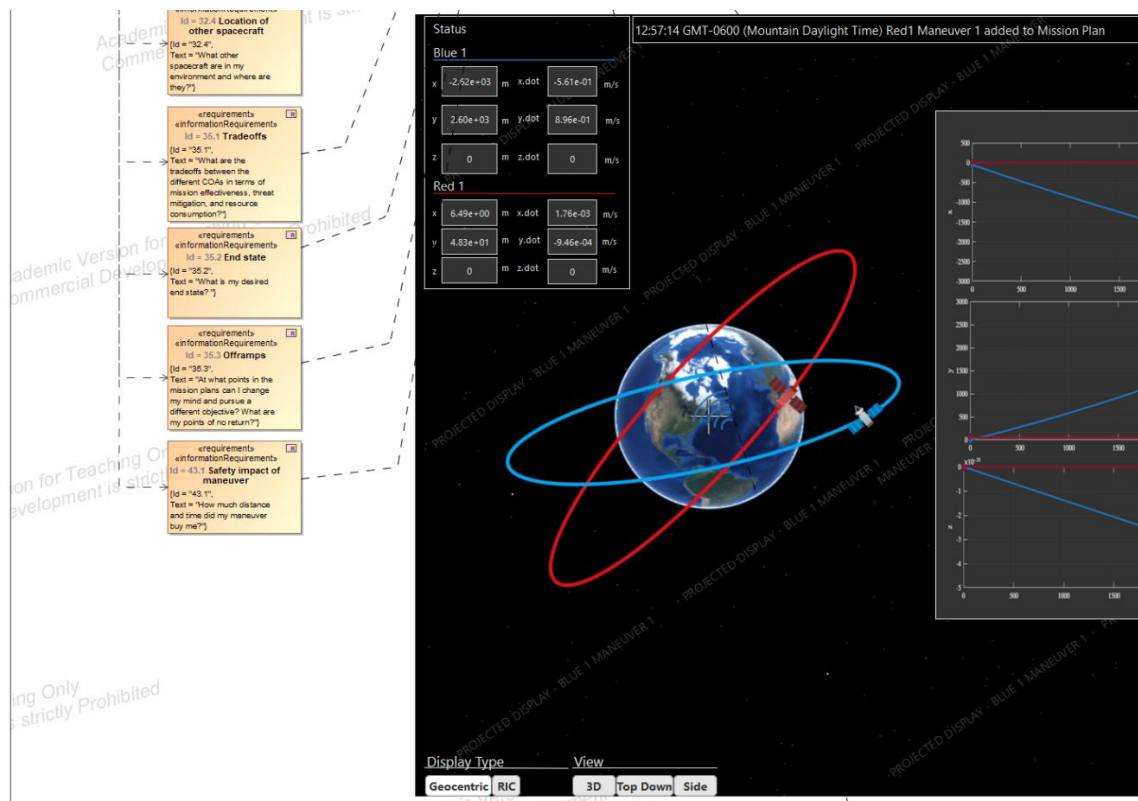


Figure 50 Visualization Panel (Continued)

User Feedback.

The responses from the survey participants suggested multiple areas for improvement, from improved iconography to the addition of a new high-level goal. The full list and their potential impacts to the interface design are shown in Table 4. Although the need for a sun vector visualization was not apparent to the researcher from the initial literature review, multiple operators quickly identified its absence and explained why it was a necessary part of planning maneuvers for evading an aggressor who might be dependent on specific lighting conditions to accomplish an observation mission. Operators also identified a deficiency in the distance condition for the safety goal: it shows the relative distance between the spacecraft based on a straight line between the two, but not how far apart they are above, below, and to the side. Based on orbital mechanics, reducing distance in one direction might be faster or more fuel efficient than closing the same distance in another. In addition to suggesting specific interface changes, operators discussed how their planning would also include considerations for the impact of a maneuver or defensive action on the safety of the orbital environment, a new high-level goal that would bear its own decision and information requirements for the UI to support. This feedback can be addressed with new versions of the system model and UI prototype and shown to operators again, emphasizing the value of cognitive modeling and rapid prototyping for identifying requirements changes early in the design process.

Table 4 Suggested Interface Improvements

Suggested Feature	Impact to Design
Indicate required cooldown times for thruster burns	New constraints listed in the Spec Sheet and Maneuver Planning displays
Highlight expected recovery times for bus/C2/payload outages in status accordion	Updated iconography in Bus/C2/Payload Status panel headers
Show reachability volumes to indicate where a spacecraft might be able to maneuver	Expand Visualization Panels and Maneuver Planning displays to calculate and show reachability volumes for different maneuver types
Indicate spacecraft collision risks	Expand Visualization Panels and Maneuver Planning displays to calculate collision risks and trigger collision warnings
Show the vectors between spacecraft and the sun	Expand Visualization Panels, Maneuver Planning displays, and Goal panels to show the effects of sun vector angles on mission performance for the defending satellite and the adversary
Show fuel reserve amount	Update fuel gauge display to highlight a set fuel reserve amount
Rework relative position diagram to show both distance and direction along the three axes	Incorporate additional 2D bars for the axial distances or create a new 3D representation in the Goals Panel
Show intelligence events like space launches in the Situation Awareness Bar	Add new event types to the Situation Awareness Bar
Introduce displays for spacecraft and payload attitude and articulation	Indicate attitude and articulation in the Goals and Visualization panels and expand the Goal Maneuver section in the Maneuver Planning displays
Improve indication that the simulation has moved into a speculative state	Rework “future state” watermarks to draw more attention without obscuring mission data
Introduce uncertainty indicators for spacecraft and payload positions	Create new options in the Visualizations Panel to show uncertainty volumes for positions based on propagation error
Allow the selection of different algorithms for mission planning and simulation	Add an algorithm selection feature that impacts the amount of uncertainty shown for maneuvers and visualizations
Show the last known “good” state of the spacecraft and its payloads	Identify last known good states and times on the Status, Goal, and Situation Awareness panels
Support goal to preserve the safety of the orbital environment	See “Indicate Collision Risk” suggestion

Discussion and Conclusions

The selection of capable prototyping software was critical for the process of creating prototype operator UIs. The software needed to balance ease of use with a capability for creating complex designs. Although there is value in gaining operator feedback on visual elements of the design, allowing for interactivity within and between UI elements in a realistic scenario provides a much richer level of feedback on how well the software works at helping operators accomplish their goals. Axure RP 9 supported the rapid creation of a prototype UI with a sufficient level of interactivity to portray a realistic space operations scenario. Although not explored in this study, Axure RP 9 has the capability to perform mathematical operations on global and local variables, which could allow future versions of the prototype to simulate the effects of user interactions with less scripting, making scenarios more realistic.

Applying ecological interface design methods to the UI elements highlighted the connections between operator interactions and high-level mission goals. In existing space mission management tools, operators must plan maneuvers based on a specific set of inputs generated by orbital mechanics experts. This creates a wide gulf between novice and expert space operator performance, where novices may not have an idea of what they're telling the spacecraft to do but experts may be able to visualize the future state of a spacecraft and plan accordingly. Supporting the creation of maneuvers based on a desired mission or goal state may allow inexperienced operators to visualize the impacts of their actions in the same manner as an expert, which would speed up the decision-making process with less training required.

Providing traceability from UIs to requirements satisfaction up to high-level goal fulfillment is key for showing how cognitive systems engineering methods can drive a full system design and implementation process. As a result, this chapter has demonstrated how the information requirements were translated into functional UI components that support operator decision-making. If, through additional operator interviews or workplace observation, a domain goal is determined to depend on additional domain functions then the UIs that need to be changed can be found quickly by following the links between domain functions, decision requirements, and information requirements to the interface implementation specifications. If a UI can be expanded to meet additional goals, then that process can be reversed to tie the interface implementation to those goals in the FAH. This end-to-end, reversible, and iterative process simplifies the act of making changes to a design prior to delivery to operations, both further defining the operator's goals and information requirements as well as providing structured refinement of any interface implementation. Importantly, the underlying FAH, domain functions, decision requirements, and information requirements can be reused to aid the design or evaluation of alternate UIs by determining which of the information requirements and domain functions are supported in any UI concept.

The user experience survey provided a rich source of user feedback in a small amount of time, with each survey requiring only an hour to present the operational scenario and collect answers to the probing questions. The responses varied from suggestions to improve shapes, colors, or other basic elements of the UI to the introduction of new visualizations and high-level goals. Using a list of probing questions based on situation awareness and cognitive systems engineering concepts ensured that the

feedback could be directly used to improve the work domain model and system requirements. The improvements can then be turned into interface improvements for another prototype iteration or incorporated into the developmental version of the software.

The first iteration of the system modeling and UI prototyping process revealed additional information that should be used to expand the FAH into the Functional Abstraction Network (FAN) described by Potter et al. (2003). The FAN iterates on each high-level goal to identify lower levels of sub-goals and processes that can each impact the goals, sub-goals, and processes elsewhere on the diagram. These cross-links are discovered through repeated cycles of building and presenting the FAN to operators for feedback. For this research, the simpler FAH diagram was chosen to accelerate the process of building a prototype and demonstrating it to operators for feedback. During the survey process, the operators identified interactions between goals and processes that suggest the need for the cross-link representations. For example, the underlying orbital dynamics model for the mission management software impacts the precision of various functions and representations. The outcome of one function or visualization can drive an operator to choose a more involved dynamics model and repeat or change procedural steps. Describing these relationships in an FAN would ensure that future UI iterations account for interactions across goal hierarchy levels.

Chapter Summary

This chapter discussed how UI components were developed to satisfy the system information requirements derived in Chapter III and then presented to operators for

feedback. Several pieces of commercial UI prototyping software were compared and Axure RP 9 was selected for the creation of the prototype space operator UI. This prototype explored user workflows and interactions through an example space defense scenario. The UI implementations were documented in the system SysML model to show traceability from operator goal definition to UI implementation. Finally, the prototype scenario was presented to operators for feedback. The operators identified multiple areas of improvement, from better representations of relative distance between spacecraft to the addition of a high-level goal for preserving the safety of the orbital environment during a conflict. The next chapter summarizes the research conducted for this thesis and provides recommendations for action for the space operator UI prototype as well as suggestions for further research in defining user roles, machine-based assistants, and training applications.

V. Conclusions and Recommendations

Chapter Overview

This chapter discusses how the research conducted fulfills the research objectives and answers the investigative questions posed for the design of a system to assist space operators with identifying and deterring hostile actions toward their satellites. The research results indicate actions that the Department of Defense should take to improve the state of space operations tools. There are also recommendations for further research in the area with the aim of improving space operations expertise for the United States and its allies. The chapter concludes with a discussion of the significance of this research project in terms of the benefits to program offices involved in software system design.

Conclusions of Research

This thesis began with the question “How can ecological interface design methods be applied to improve tools for space operators to identify, respond to and deter hostile actions in orbit?” From that overarching question a series of specific investigative questions were derived. These questions aimed to address the issues of how to implement cognitive systems engineering processes to identify space operator needs, build system designs to help them meet those needs, and determine the effectiveness of those designs. The research presented in Chapters III and IV described the investigation process and provided answers to the questions.

The first question posed was, “*How can cognitive systems engineering be used to obtain the information and decision requirements for defending satellites?*” The information and decision requirements are derived from space operator goals and the

means, in terms of processes and functions, by which the operators achieve those goals. Chapter III described cognitive systems engineering methods that were used to define those goals and derive information and decision requirements from them. User interface elements were built to satisfy those information requirements and represented in the model to show requirements verification. An example was shown in Table 1. The model of space operations goals and functions was shown in Table 2.

The next question was, *“How can cognitive task analysis identify how operational context changes the ways operators interpret information and make decisions?”* In a space conflict, operators must make decisions that balance the mission effectiveness of their satellite against their need to safeguard it from hostile actions. Chapter IV provided examples of how operators could select maneuvers based on a desired goal state and examine the implications to their mission effectiveness in Figure 43. These forms of ecological interface designs assist operators with interpreting complex fields of information by allowing them to make decisions based on the high-level state of the operational environment in addition to low-level details.

Next was, *“What are the appropriate data representations that permit experienced operators to better understand the threats and available options to counter these threats?”* Space operators will not necessarily know much about a potential adversary from mere observation. Chapter III described the information requirements space operators have for effectively assessing threats and planning their responses, with an example shown in Table 1. Chapter IV showed how those information requirements were built into a prototype user interface and demonstrated to operators for their

feedback. The results of the feedback generation process for the prototype UI were shown in Table 4.

The fourth question asked, *“How can ecological interface design make mission planning challenges tractable?”* As mentioned before, space operators must balance mission versus safety when planning evasive maneuvers. They must make these plans in an environment where their adversary is making its own maneuvers, and it may be difficult to establish a complete and accurate picture of all objects in the orbital environment. Chapters III and IV showed how interface design requirements were built and implemented to assist with these challenges in Figures 38 through 51. The combination of goal-based mission planning and situation awareness features can simplify the process of making mission plans and give operators a greater awareness of what the orbital environment will look like at some point in the future.

The fifth question asked, *“How can the knowledge of operators’ information needs be captured, traced, and modeled in Model-Based Systems Engineering (MBSE) tools to support the derivation of system and training requirements?”* This is accomplished using the SysML modeling language, with extensions to support cognitive systems engineering concepts, in a modeling tool like Cameo Systems Modeler. Chapter III described the process of modeling operator goals and processes with SysML extensions. These goals were then used to derive system decision and information requirements to be satisfied with user interface designs. Chapter IV described how those designs were documented in the overall system model to provide traceability from user interface implementations to system requirements and operator goal satisfaction.

The final question asked, “*How can the operational impacts of ecological interface design be measured?*” Interface designs can only be validated through exposure to operators. Ideally, the designs should be presented to operators for feedback as early as possible in the design process before delivery. This would be possible in an iterative software development process where the design is continually presented to operators for feedback as it matures, and their feedback is incorporated in future pre-release increments. In a traditional waterfall process it would be possible to present designs to operators but incorporating their feedback could require extensive re-work and contract scope changes that are less compatible with the rapid prototyping concept. Chapter IV described a method for obtaining operator feedback throughout the development process. Rapid, functional UI prototypes can be developed and shown to operators for their feedback. This feedback can then be incorporated as new elements in or tweaks to the system model and built into new UI prototypes. Accommodating operator feedback early in development can reduce the risk that the system will not meet requirements upon deployment. The results of the feedback generation process for the prototype UI were shown in Table 4.

Recommendations for Action

The cognitive system model and UI prototype developed for this thesis can inform ongoing research at the Air Force Institute of Technology, as well as operator training and education for the United States Space Force. The UI prototype should be matured into a higher-fidelity form so operators can interact with it without strict scenario scripting. This will allow researchers to capture novel interactions and better identify

unintuitive or challenging workflows. If desired for operator training, the prototype should be tailored for specific mission types and satellite platforms (e.g., with accurate specifications for the Space-Based Infrared System missile warning satellites). To improve the operator experience, the ecological interface design concepts should be expanded upon to include manipulations of spacecraft within the graphic visualizations to develop mission plans and predict maneuvers. This process would be easier and faster for both novice and expert users. It was not attempted during this research because of technical limitations with the available prototyping software. These improvements would allow the products of this research to improve the state of space operations training and education.

The cognitive system development cycle described in this research should be paired with iterative system development. While work domain studies and prototyping would provide some value in a traditional waterfall process, iterative development and regular feedback are necessary to capture changes to the system model and deliverables prior to delivery. An illustration of how the cognitive system development cycle fits into an iterative process is shown in Figure 52. The system architecting process leads to a cycle of work domain studies, modeling, prototyping, and feedback that can be repeated as many times as necessary to capture operator feedback. UI prototypes inform the incremental development process which creates functional software that is fed back into the operator feedback process and delivered to operations when ready. Accounting for changing operator goals, decisions, and information requirements throughout the early design process would ensure that the final product is usable and well-adapted to the operational environment.

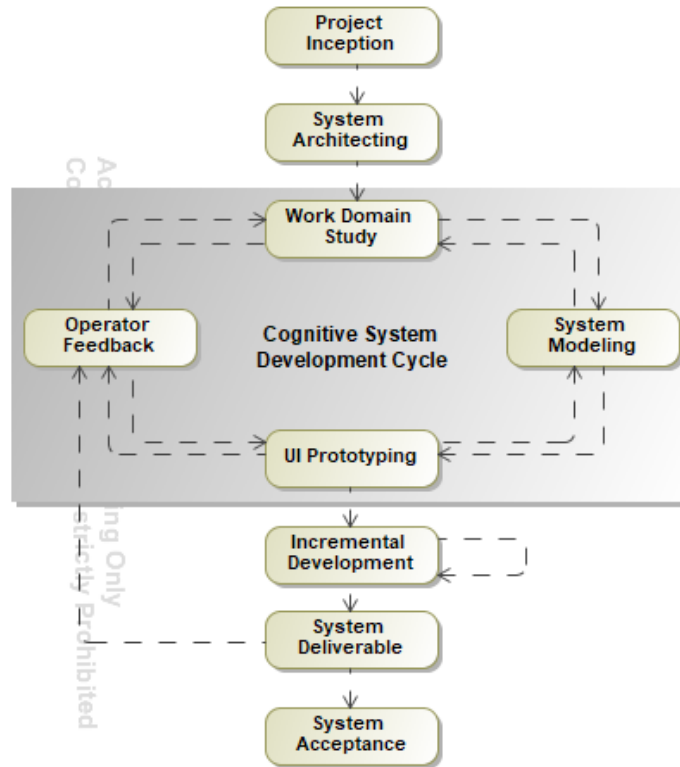


Figure 52 Cognitive System Development within an Iterative Development Process

Recommendations for Future Research

The research performed for this project identified additional avenues for investigators to contribute to the improvement of tools for space operations. These research areas should be considered for investigation within the United States Air Force and Space Force.

Explore Different User Roles.

The prototype space operations UI provided tools for mission planning and situation awareness. However, the model of space operations goals included elements for the use of weaponry, legal analysis, and decision-making at different levels of the military chain of command. Satisfying these goals would likely require different

approaches to interface design that better meet the needs of users with different operational and organizational roles. Additional studies on how these roles interact within the space operations domain could drive the creation of new and improved tools for these users.

Automation Design for Operator Assistance.

While ecological interface design and other user experience improvements can assist space operators with their missions, the domain remains cognitively challenging. The challenges in the space domain increase dramatically with the number of spacecraft and other orbital objects that operators must keep track of, as well as the number of actions available to them. Additional decision-making assistance, in the form of intelligent machine agents, could assist operators with these challenges. Machine teammates could assist human operators with interpreting the information available to them as well as perform cognitively challenging functions to reduce the workload on human operators or make them more effective at achieving their operational goals.

Concepts for Training Design.

This research focused on modeling goals and creating a system design for an operational domain. Users in a space operations training environment may have a different set of goals that start simple and become more complex as they grow increasingly familiar with concepts during their instruction. Their instructors will have their own set of goals to motivate and challenge students at different points in a curriculum. Additional research should be performed to examine how a space operations interface should be designed to support the progressive development of knowledge while making the experience engaging for the students.

Significance of Research

This research demonstrated how cognitive system engineering processes can be incorporated into model-based systems engineering tools and used to inform user interface designs. Linking each interface element to system information and decision requirements, all tracing back to major operator goals and subgoals, ensures that every piece of the design is grounded in what the operator needs to accomplish in their work. With model-based systems engineering tools, it is simple to identify UI components that are overly complex or superfluous to the necessary function of the system, as well as unmet operator needs. This process could provide two major boons to program offices: interfaces that operators find more effective and easier to learn, and leaner software designs that take less effort to build and maintain.

The use of interface prototyping tools also provides a valuable source of feedback into an iterative, cognitive systems engineering design process. Interface components that are built and tested in prototyping tools can be displayed in the system models to prove requirements verification or show the tradeoffs between multiple designs. These designs, with operational scenarios built around them, can be provided to operators during task analysis interviews to assess how well the designs support decision-making and goal-seeking. Operator feedback from these interviews (as shown in Table 4) is easily incorporated into the system model to identify the impacts of design changes. The iterative process of rapidly prototyping designs and folding operator feedback into the system model to inform new designs makes it easier for program offices to accommodate changing requirements during system development, reducing the risk of an unsatisfying transition to operations. Future software upgrades can take advantage of mature models

of operator goals and decision/information requirements, which only need to be updated if the scope or nature of the operator's work changes.

These processes are applicable to a variety of system types beyond the domain of space operations. Any software-oriented system design process would benefit from better methods to define system scope and respond to operator needs. Using rapid prototyping to identify cognitive performance issues can ensure that systems are capable of supporting operators in dynamic, uncertain environments.

Summary

This thesis introduced problems facing space operators who need to maneuver their spacecraft to evade orbital threats while continuing to perform their mission. Cognitive systems engineering methods and ecological interface design were suggested as ways to design software user interfaces that help operators identify and deter threats while maintaining mission capability. Research showed how model-based systems engineering and interface prototyping tools integrate with cognitive systems engineering processes to produce effective designs that address the investigative questions. Additional areas of research were suggested to develop these concepts in the areas of user role definition and the integration of machine agent-based automation. Finally, the significance of the research was described in terms of the benefits to program offices working in software system design.

Appendix A – Applied Cognitive Task Analysis Interview Template

Process

The ACTA will be conducted in three phases: Task Diagram, Knowledge Audit, and Simulation Interview.

1. Task Diagram (Length: 10 minutes)

Participants will be asked to identify the steps necessary to perform their tasks. The number of steps should be limited to no more than 6 and no less than 3 to avoid spending time on minute details. Once complete, they will be asked to identify which steps are the most cognitively challenging, meaning those that require complex decision-making, though, or coordination to accomplish (Militello & Hutton, 1998b, p. 1620).

2. Knowledge Audit (Length: 30 minutes)

Participants will be asked a series of probing questions to identify specific characteristics of expert performance in cognitively demanding tasks. These questions will be developed ahead of time to focus on areas research identifies as important to expert decision-making (Militello & Hutton, 1998b, pp. 1621–1622).

3. Simulation Interview (Length: 20 minutes)

Participants will be shown a presentation that goes through a real-world historical event that required stressful, high-importance decision-making. They will be asked to identify major events, give their assessment of the situation, point out key environmental cues, examine “what if?” events, and explain mistakes they think a novice might make (1998b, pp. 1623–1624).

Probing Questions for Knowledge Audit

These questions are adapted from Militello and Hutton's "Applied Cognitive Task Analysis (ACTA): A Practitioner's Toolkit for Understanding Cognitive Task Demands" (1998b) and McDermott et al.'s "Human-Machine Teaming Systems Engineering Guide" (2018, pp. 13–19). This guide, written for The MITRE Corporation, aims to help systems engineering practitioners gather subject matter expert data appropriate for deriving requirements for systems with human-machine team interactions. The questions are categorized according to the type of insights they provide for system design. The final list of questions has been selected and tailored for the space defense operational environment.

1. Past & Future

These questions elicit information about how the system changes future states and how operators depend on past knowledge to make predictions of the future.

- 1.1. How predictable are the missions you accomplish with your system? What changes over the course of a day, and what changes with operational tempo?
- 1.2. How does historical information help you plan for the future?

2. Big Picture

These questions elicit information about how the operator accomplishes their work.

- 2.1. Can you give me an example of what is important about the Big Picture for this task? What are the major elements you must know and keep track of?
- 2.2. What kind of decisions and actions do you have to make in a typical day of work? How about in a stressful situation?

3. Noticing

These questions elicit information about what the operator notices in their environment to help them perform their role.

3.1. What are the key signs or cues in your environment that tell you you're on track to accomplish the mission? How about when you're not on track?

3.2. Have there been times that you wish you were notified of new or changing information? Would that have made a difference in your decision making?

4. Anomalies

These questions elicit knowledge about how the operator diagnoses and responds to system issues.

4.1. What kind of things do you notice that tell you that coordination and/or performance in your workplace are not going as planned?

4.2. Can you describe an instance when you spotted a deviation from the norm, or knew something was amiss?

5. Improvising

These questions elicit information about how an operator deals with unexpected situations.

5.1. What are some example situations in which you have had to rapidly improvise a plan? For example, to handle a threat or take advantage of an opportunity.

6. Self-Monitoring

These questions elicit information about how an operator monitors their own performance or that of the people they work with.

6.1. Can you describe a situation when you knew you were task saturated/overloaded and had to ask for help?

6.2. If a new person on your team were to take over your job, what would you be most concerned about? What part of your job would you feel most uneasy about if a new or inexperienced person was doing it?

7. Job Smarts

These questions elicit information about what determines good or bad performance in the role.

7.1. When you do this task, are there ways of working smart or accomplishing more with less – that you have found especially useful?

8. Equipment Difficulties

8.1. Have there been times when the equipment pointed in one direction, but your own judgment told you to do something else? Or when you had to rely on experience to avoid being led astray by the equipment?

Probing Questions for Simulation Interview

The following questions are tailored from the list presented in Klein et al.'s "Critical Decision Method for Eliciting Knowledge," (1989, p. 466) the work that defined the methods from which the ACTA interviews were derived. These questions are meant to help the interviewer guide the participant as they walk through a presentation on a challenging operational event.

1. "What would be your goal when responding to this event?"

2. “How could a visualization and planning tool help you in responding to this event?”
3. “What mistakes were possible at this point?”
4. “What information in your environment would help you make your decision?”
5. “How about external to your environment?”
6. “Are there any past events that would help you make your decisions?”
7. “What information do you think is lacking that could help you?”
8. “What made this incident special?”
9. “What would you do if [aspect of event] played out differently?”

Appendix B – User Experience Survey

Prompt

The user is in control of Blue 1, a strategic missile warning satellite in geostationary orbit. They are responsible for ensuring that the satellite remains mission capable. To do this, they have the ability to perform satellite and payload maintenance functions as well as to plan defensive or mission-oriented maneuvers. At the beginning of this scenario, a potential aggressor, Red 1, has sidled into orbit close to Blue 1.

Scenario Execution

To begin the scenario, the researcher will launch the UI prototype and share the window with the participant via a desktop streaming service. The researcher will then recite the scenario prompt, present the mission card summary (Figure 53), and begin walking the participant through the scenario script. At each step, the researcher will ask the participant one or more probing questions intended to assess how effective the participant perceives the UI prototype to be at simplifying cognitive tasks and providing valuable situation awareness. The research will document the participant's responses.

Mission Card

Game Card: Observation and Evasion Scenario

Scenario Overview

- Satellite Class
 - Blue 1: Strategic earth observation satellite
 - Red 1: Aggressor satellite with known imaging capabilities
- Satellite Tasking
 - Blue 1: Prevent imaging while remaining on mission
 - Red 1: Close distance and image Blue 1 for as long as possible
- Scenario Duration and Orbital Regime
 - Duration: 2 days (12-hr increments)
 - Orbital regime: Geostationary (GEO)
- Mission Constraints
 - Blue 1: Retain enough fuel for 5 years of mission lifespan
 - Red 1: Attempt no direct/kinetic hostile action

Satellite Conditions

- Blue 1
 - Delta-V max: 5 m/s; delta-V total: 500 m/s
 - Initial State: Fixed in orbit and pointed at mission target
 - Vulnerability Zone: **km sphere
- Red 1
 - Delta-V max: 2 m/s; delta-V total: 250 m/s
 - Initial State: Co-orbital with Blue 1; 200 km ahead in in-plane direction
 - Engagement Zone: 50m cone; 30 deg half-angle
- Mission Constraints
 - Blue 1: Retain enough fuel for 5 years of mission lifespan
 - Red 1: Attempt no direct/kinetic hostile action

Proceed to Scenario

Figure 53 Scenario Mission Card Presented to Participants

Timeline

- Day 1, 0001 - 1200:
 - Blue 1: One of the two ground stations in contact with this satellite is experiencing an issue that has taken it out of operations. The operator perceives that there will be an impending payload issue at 0500 that requires their attention.
 - Red 1: The satellite is seen executing a maneuver that will bring it to within 50km of Blue 1
- Day 1, 1201 – 2400:
 - Blue 1: The operator has addressed the payload issue and both ground stations are fully operational. They see that Red 1 has come within 50km

of their satellite and review the available intelligence reports. The operator considers four possible maneuvers that Red 1 could make within its estimated capabilities to reach Blue 1's vulnerability zone. They select "Most Time at Min" to plot a maneuver that has Red 1 maximize the time spent at a minimum distance of 15km.

- Red 1: The satellite is in an orbit that places it a minimum of 50km away from Blue 1 for half of its period.
- Day 2, 0001 – 1200:
 - Blue 1: A complete ops center outage has occurred and severed all situation awareness and contact with Blue 1. Estimated recovery time is 12 hours.
 - Red 1: The satellite makes a maneuver that brings it to within 5 km of Blue 1.
- Day 2, 1201 – 2400:
 - Blue 1: The ops center has returned to full capability and Red 1 is seen within 5 km of Blue 1. The operator considers emergency maneuvers to remove Blue 1 from observation distance. They also update their estimates of Red 1's fuel load and plan their maneuver accordingly.
 - Red 1: The satellite takes no further action.
- Scenario Complete

Probing Questions

1. How does this UI help you understand the basic elements, such as position, speed, capabilities, and available resources, in the situation?
2. In what ways is the UI lacking in helping you understand the basic elements?
3. How do the different elements of the UI help you achieve the goal of defending your satellite?
4. How do the different elements of the UI help you achieve the goal of completing your satellite's mission?
5. What additional tools or displays would help you achieve these goals?
6. What kind of issues do you foresee with this UI if you had to manage a situation where the time pressure was much greater and you had to make decisions on the order of minutes, not hours?
7. Given the elements in play for this scenario, including the two satellites, their orbits, and their missions, how could this UI be improved to help you predict how the situation will progress?
8. What kind of decisions might a mission planner have to make that are not supported by this UI?
9. What kind of space operations goals, in addition to defending your satellite and achieving your mission, are not supported by this UI?
10. What additional feedback do you have for this UI?

Bibliography

- Air Force Space Command. (2018). *Air Force Space Command Commander's Strategic Intent*.
- Bennett, K. B., Bryant, A., & Sushereba, C. (2018). Ecological Interface Design for Computer Network Defense. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 60(5), 610–625. <https://doi.org/10.1177/0018720818769233>
- Bennett, K. B., Posey, S. M., & Shattuck, L. G. (2008). Ecological Interface Design for Military Command and Control. *Journal of Cognitive Engineering and Decision Making*, 2(4), 349–385. <https://doi.org/10.1518/155534308x377829>
- Boden, D. G. (1999). Introduction to Astrodynamics. In J. R. Wertz & W. J. Larson (Eds.), *Space Mission Analysis and Design* (Third Edit, p. 133). Microcosm Press and Springer. https://doi.org/10.1007/978-94-011-2692-2_6
- Brooke, J. (1996). SUS: A quick and dirty usability scale. In P. W. Jordan, B. Thomas, I. L. McClelland, & B. Weerdmeester (Eds.), *Usability Evaluation in Industry* (pp. 189–194). Taylor and Francis.
- Christoffersen, K., Hunter, C. N., & Vicente, K. J. (1996). A Longitudinal Study of the Effects of Ecological Interface Design on Skill Acquisition. *Human Factors*, 38(3), 523–541.
- Colby, E. (2016). *FROM SANCTUARY TO BATTLEFIELD : Deterrence Strategy for Space*. January.
- Defrieze, B. D. C. (2014). Defining and Regulating the Weaponization of Space. *Joint Force Quarterly*, 74(August), 110–116.
- Endsley, M. R. (1995). Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1), 32–64. <https://doi.org/10.1518/001872095779049543>
- Ewart, R. (2018). Persistent Space Situation Awareness for the Guardians of the High Frontier. *Air & Space Power Journal*, 32(2), 84.
- Finstad, K. (2010). The usability metric for user experience. *Interacting with Computers*, 22(5), 323–327. <https://doi.org/10.1016/j.intcom.2010.04.004>

- Frey, A. E. (2008). Defense of US Space Assets. *Air and Space Power Journal*, Winter, 75–85. <http://www.au.af.mil/au/afri/aspj/airchronicles/apj/apj08/win08/frey.html>
- Gruss, M. (2015, July). Maneuvering Russian Satellite Has Everyone's Attention. *SpaceNews*. <https://spacenews.com/maneuvering-russian-satellite-has-everyones-attention/>
- Hanlon, E., & Yakimenko, O. (2019). Introduction to Space Dogfighting - What Matters in Space Engagements. *IEEE Aerospace Conference Proceedings, 2019-March*. <https://doi.org/10.1109/AERO.2019.8741819>
- Hoffman, R. R., Coffey, J. W., Carnot, M. J., & Novak, J. D. (2002). An Empirical Comparison of Methods for Eliciting and Modeling Expert Knowledge. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 46(3), 482–486. <https://doi.org/10.1177/154193120204600356>
- Hollnagel, E., & Woods, D. D. (1982). *Cognitive Systems Engineering: New Wine in New Bottles*. <https://doi.org/10.1016/j.lisr.2017.07.009>
- Hollnagel, E., & Woods, D. D. (2005a). *Joint Cognitive Systems - Foundations of Cognitive Systems Engineering*.
- Hollnagel, E., & Woods, D. D. (2005b). *Joint Cognitive Systems - Patterns in Cognitive Systems Engineering*.
- Holt, J., & Perry, S. (2013). SysML for systems engineering: 2nd edition: A model-based approach. In *SysML for Systems Engineering: A Model-Based Approach*. <https://doi.org/10.1049/PBPC010E>
- Janson, S. W. (1993). The on-orbit role of electric propulsion. *29th Joint Propulsion Conference and Exhibit*. <https://doi.org/10.2514/6.1993-2220>
- Klein, G. A., Calderwood, R., & Macgregor, D. (1989). Critical Decision Method for Eliciting Knowledge. *IEEE Transactions on Systems, Man and Cybernetics*. <https://doi.org/10.1109/21.31053>
- Lamm, J. G., & Weilkiens, T. (2014). Method for Deriving Functional Architectures from Use Cases. *Systems Engineering*, 17(2), 225–236. <https://doi.org/10.1002/sys>
- Long, D., & Scott, Z. (2011). *A Primer for Model-Based Systems Engineering* (2nd ed.). Vitech Corporation.

- Mcdermott, P., Dominguez, C., Kasdaglis, N., Ryan, M., Mitre, I. T., & Nelson, A. (2018). Human-Machine Teaming Systems Engineering Guide. *The MITRE Corporation*.
- Militello, L. G., & Hutton, R. J. B. (1998a). Applied cognitive task analysis (ACTA): a practitioner ' s. *Ergonomics*.
- Militello, L. G., & Hutton, R. J. B. (1998b). Applied cognitive task analysis (ACTA): A practitioner's toolkit for understanding cognitive task demands. *Ergonomics*.
<https://doi.org/10.1080/001401398186108>
- Militello, L. G., Hutton, R. J. B., Pliske, R. M., Knight, B. J., & Klein, G. (1997). *Applied Cognitive Task Analysis (ACTA) Methodology*.
- Miller, M. J., & Feigh, K. M. (2019). Addressing the envisioned world problem: A case study in human spaceflight operations. In *Design Science* (Vol. 5).
<https://doi.org/10.1017/dsj.2019.2>
- Naasz, B. (2005). Safety Ellipse Motion with Coarse Sun Angle Optimization. *595 Flight Mechanics Symposium; October 18-20, 2005; Greenbelt, MD; United States*, 1–13.
- National Air and Space Intelligence Center. (2018). *Competing in Space*.
https://www.nasic.af.mil/Portals/19/documents/Space_Glossy_FINAL--15Jan_Single_Page.pdf?ver=2019-01-23-150035-697
- Potter, S., Gualtieri, J., & Elm, W. (2003). Case studies: Applied cognitive work analysis in the design of innovative decision support. *Handbook for Cognitive Task Design*, 1–35.
- Rasmussen, J. (1985). The Role of Hierarchical Knowledge Representation in Decision making and System Management. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-15(2), 234–243. <https://doi.org/10.1109/TSMC.1985.6313353>
- Raymond, J. W. (2018). *Air Force Space Command Commander's Strategic Intent*.
- Roth, E. M. (2008). Uncovering the Requirements of Cognitive Work. *Human Factors: The Journal of the Human Factors*, 50(3), 475–480.
<https://doi.org/10.1518/001872008X288556>
- Vicente, K. J. (1999). *Cognitive Work Analysis Toward Safe, Productive, and Healthy Computer-Based Work*. Lawrence Erlbaum Associates, Publishers.

- Vicente, K. J., & Rasmussen, J. (1990). The Ecology of Human-Machine Systems II: Mediating “Direct Perception” in Complex Work Domains. *Ecological Psychology*, 2(3), 207–249. https://doi.org/10.1207/s15326969eco0203_2
- Vicente, K. J., & Rasmussen, J. (1992). Ecological Interface Design: Theoretical Foundations. *IEEE Transactions on Systems, Man and Cybernetics*, 22(4), 589–606. <https://doi.org/10.1109/21.156574>
- Woods, D. D., Patterson, E. S., Roth, E. M., & Christoffersen, K. (2002). Can We Ever Escape from Data Overload? A Cognitive Systems Diagnosis. *Cognition, Technology & Work*, 4, 22–36. <https://doi.org/10.1177/154193129904300310>
- Ziarnick, B. (2018). The Coming Revolution in Military Space Professionalism. *Air & Space Power Journal*, 32(2), 9–20.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 074-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 30-10-2020		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From – To) March 2020 – October 2020	
TITLE AND SUBTITLE Specifying Space Defense Operator Interfaces through the Application of Cognitive Systems Engineering and Prototyping				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Oryschak, Justin E., Captain, USAF				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/ENY) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENV-MS-20-D-070	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Intentionally Left Blank				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED					
13. SUPPLEMENTARY NOTES This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.					
14. ABSTRACT The Department of Defense needs better tools to support its operators as they strive to defend its space assets. The growing sophistication of anti-satellite weapons increasingly challenges the nation's orbital communications and surveillance infrastructure. Operators face difficulties gathering useful information and dealing with the complexity of potential enemy actions. This research applied cognitive systems engineering and ecological interface design (EID) methodologies to create a prototype space mission management tool that enhances operator situation awareness and decision-making ability. Applied cognitive task analysis interviews were used to document space operator decision-making in their domain. Model-based systems engineering was applied to integrate work domain concepts into system models. EID methods were applied to inform user interface designs that support high-level decision making in addition to low-level tasks. User interface concepts were developed using rapid prototyping software, Axure 9.0, to satisfy the system requirements. The software prototypes were shown to space operators and assessed for validity. This process demonstrated how cognitive systems engineering can be used to derive system requirements and create system designs, the elements of which can be captured in a systems model and traced to operator goals, resulting in systems that are more capable of supporting operator needs in challenging environments.					
15. SUBJECT TERMS CSE, HSI, human factors, MBSE, prototyping, space operations, SysML, task analysis, UIs					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 136	19a. NAME OF RESPONSIBLE PERSON Michael E. Miller, AFIT/ENV
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code) (937) 255-3636 x4651 (michael.miller@afit.edu)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18