

**A FLEXIBLE, MODULAR APPROACH TO INTEGRATED SPACE EXPLORATION
CAMPAIGN LOGISTICS MODELING, SIMULATION, AND ANALYSIS**

by

Paul Thomas Grogan

B.S. Engineering Mechanics and Astronautics
University of Wisconsin, 2008

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements to the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2010

© Massachusetts Institute of Technology.
All rights reserved.

Signature of Author.....
Department of Aeronautics and Astronautics
July 30, 2010

Certified by.....
Olivier L. de Weck
Associate Professor of Aeronautics and Astronautics and
of Engineering Systems
Thesis Supervisor

Accepted by.....
Eytan H. Modiano
Associate Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Abstract

A space logistics modeling framework to support space exploration to remote environments is the target of research within the MIT Space Logistics Project. This thesis presents a revised and expanded framework providing capabilities to analyze a new set of explorations using a generalized resource flow through a time-expanded network to satisfy exploration demands. The framework is both flexible to model a wide range of destinations using mixed levels of fidelity and modular to enable future expansion through interfaces.

The SpaceNet software tool implements the space logistics modeling framework, providing integrated modeling and simulation capabilities for quantitative space exploration campaign analysis. Discrete event simulation identifies logistical infeasibilities and provides quantitative measures of exploration effectiveness to guide trade studies or other campaign analyses. SpaceNet 2.5, a Java executable with an extensive graphical user interface, has been publicly released under an open source license.

Four case studies are presented as examples of the modeling framework applied to relevant exploration campaigns. A resupply of the International Space Station from 2010-2015 includes 77 flights of seven different vehicles from six launch sites to investigate the supply capacity under existing resupply strategies. A near-Earth asteroid exploration details a two crew, 14-day tele-operated mission at 1999-AO10 to establish the feasibility requirements of using modified Constellation vehicle architectures. A lunar outpost exploration models the buildup of infrastructure and surface excursions leading to continuous human presence over 21 missions and seven years. Finally, a Mars surface exploration models the ten launches and in-space nuclear thermal rocket propulsion required to send a crew of four to the surface of Mars for a 531-day exploration.

Finally, a usability experiment is presented to demonstrate the usability and efficiency of the SpaceNet tool as compared to independent analysis methods. Seven test subjects were tested, five using SpaceNet and two control subjects using spreadsheet-based methods, to analyze and establish the feasibility of a near-Earth object mission. The median SpaceNet subject required 35 minutes to complete the analysis, compared to a median of 113 minutes for the control subjects.

Thesis Supervisor: Olivier L. de Weck

Title: Associate Professor of Aeronautics and Astronautics and of Engineering Systems

(This Page Intentionally Blank)

Acknowledgements

There are many individuals in need of recognition for their substantial impact on this project. First, I would like to thank Prof. Olivier de Weck for his advising and mentoring over the past two years. He has generously supported both my academic and professional development by providing countless opportunities to participate in projects, committees, conferences, and meetings, all of which have dramatically supplemented my research.

I would be lost without the help of my family and closest friends. I would like to thank my father Tom for encouraging me to always accept new challenges, my mother Kathy for showing what is truly important in life, my sister Sarah for her selfless attitude, and my brother Brian for his loyal companionship. Special recognition is reserved for Natasha Lewis, for whom frequent visits are the least I can offer for the support she has provided.

A substantial portion of this project is built on the work of others. In particular, I would like to recognize Nii Armar, Sarah Shull, and Afreen Siddiqi for their development contributions within the space logistics modeling framework. Also, I would like to thank my colleagues at MIT for their collaboration and assistance in case study development and usability testing. In particular, I would like to thank Ariane Chepko, Torin Clark, Ben Corbin, Ivo Ferreira, Alessandro Golkar, Jennifer Green, Abe Grindle, Arthur Guest, Tak Ishimatsu, Justin Kaderka, Greg O'Neill, Basant Sagar, Narek Shougarian, Daniel Selva, Anthony Wicht, and Howard Yue.

A portion my involvement was supported by NASA Strategic University Research Partnership (SURP) contract number 1344341. Our contacts at JPL including Gene Lee, Elizabeth Jordan, and Dr. Robert Shishko provided tremendous expertise and were instrumental in guiding the project in the right direction.

This thesis is dedicated to the memory of my grandfather, Prof. Paul J. Grogan (1918-2008). He served as an inspiration from my earliest childhood and contributed greatly to my love of mathematics applied through engineering.

(This Page Intentionally Blank)

Contents

1	Introduction	14
1.1	Background	15
1.1.1	The Vision for Space Exploration.....	15
1.1.2	MIT Space Logistics Project.....	16
1.1.3	Human Space Flight Review.....	17
1.2	Project Motivation	17
1.2.1	Space Logistics “-ilities”.....	18
1.2.2	Flexible Modeling Framework.....	18
1.2.3	Modular Implementation Architecture.....	20
1.2.4	User Community	22
1.3	Related Research and Literature	22
1.4	Project Overview	23
2	Domain Model	25
2.1	Network Components Model	25
2.1.1	Nodes	25
2.1.2	Edges.....	27
2.2	Resource Model	28
2.3	Element Model.....	31
2.3.1	Element Hierarchy	31
2.3.2	Element Parts	34
2.3.3	Element States.....	34
2.3.4	Element Demand Models.....	35
3	Campaign Modeling and Analysis	39
3.1	Network Model	39
3.2	Mission Model	40
3.2.1	Mission Demand Models	40
3.2.2	Mission Events.....	40
3.2.3	Spatial Simulator.....	43
3.3	Manifest Model.....	43
3.3.1	Demand Simulator	44
3.3.2	Demand Aggregation	48
3.3.3	Demand Packing	48
3.3.4	Container Manifesting.....	49
3.4	Campaign Analysis	49
3.4.1	Campaign Simulator	49
3.4.2	Measures of Effectiveness	50
4	SpaceNet 2.5 Implementation.....	51
4.1	Development and Release.....	51
4.1.1	Development Methodology.....	52
4.1.2	Development Tools.....	52
4.1.3	Public Release	53
4.1.4	Documentation	53
4.2	Data Sources	54
4.2.1	Data Editor	55
4.2.2	Element Sizing Tool.....	56
4.3	Graphical User Interface	56

4.3.1	Campaign Panel	57
4.3.2	Network Tab	58
4.3.3	Missions Tab	58
4.3.4	Demands Tab	59
4.3.5	Manifest Tab	63
4.3.6	Simulation Tab	65
5	SpaceNet 2.5 Applications	68
5.1	Case Studies	68
5.1.1	International Space Station Resupply	68
5.1.2	Near-Earth Object Sortie.....	79
5.1.3	Lunar Outpost Buildup.....	84
5.1.4	Mars Exploration.....	94
5.2	Usability Experimentation	103
5.2.1	Testing Procedure	103
5.2.2	Results and Discussion.....	105
5.2.3	Discussion	109
6	Conclusions	110
6.1	Significant Contributions	110
6.2	Future Work	110
6.3	Integrated Analysis Strategic Roadmap	111
6.3.1	Analysis Scope.....	112
6.3.2	Stochastic Analysis	114
Appendix A	Object Model Diagrams.....	117
Appendix B	Optimal Manifesting Methods	122
	Modeling Cargo Manifests	122
	Campaign Feasibility and Manifest Optimization.....	124
References.....		126

List of Figures

Figure 1: MIT space logistics project timeline. <i>The project has followed development and changes to the NASA Human Space Flight program.</i>	15
Figure 2: Use case scenario networks. (a) <i>Lunar sortie.</i> (b) <i>ISS resupply post shuttle retirement.</i> (c) <i>Lunar hub-spoke (outpost).</i> (d) <i>Lunar spoke-hub (global exploration).</i> (e) <i>Martian human-robotic exploration.</i>	20
Figure 3: Procedural programming diagram. <i>Procedures called by a monolithic GUI edit portions of a shared memory.</i>	21
Figure 4: Object-oriented programming diagram. <i>Objects reside in memory and carry functional capabilities limited in scope. Objects are accessed by parallel GUI components.</i>	21
Figure 5: Lagrange point positions. <i>L1, L2, and L3 are stable locations, L4 and L5 are unstable and require station-keeping. Not to scale.</i>	27
Figure 6: Surface edge example. <i>Locations 1, 2, and 3 are connected with three surface edges.</i>	27
Figure 7: Space edge examples. <i>Edges correspond to propulsive burn sequences. The burns are highlighted for the space edge between nodes LEO and LLPO.</i>	28
Figure 8: Flight edge examples. <i>Edges correspond to pre-determined flight architectures with crew and cargo limits.</i>	28
Figure 9: Element hierarchy. <i>A hierarchy of elements provides specialized capabilities extensible for future expansion while establishing a common set of inherited properties.</i>	32
Figure 10: Part commonality with shared resource types. <i>The Power Plant and Lander elements share a common Power Supply resource.</i>	34
Figure 11: Element reconfiguration sequence. <i>An element may experience several operational states throughout a campaign including dormant before delivery, active or quiescent during operation, and decommissioned state to signal availability of parts for scavenging.</i>	35
Figure 12: Discrete event time effect on linear demands. <i>Discrete time simulation aggregates linear demands at each event execution.</i>	36
Figure 13: Sparing by mass using a parts list. <i>The sparing by mass demand model optionally can access an element's part list to generate unspecified, pressurized, and unpressurized spares demands.</i>	37
Figure 14: Campaign modeling and analysis. <i>Campaigns are modeled in three parts: a network, missions, and a manifest. When the campaign is both spatially feasible and logistically feasible, analysis can provide visualizations and measures of effectiveness.</i>	39
Figure 15: Network bat chart. <i>A bat chart shows a time-expanded network and element movement over time.</i>	39
Figure 16: Spatial simulation process. <i>The spatial simulator sequentially executes events, updating the network state and adding events when necessary, and may generate spatial errors.</i>	43
Figure 17: Manifest model creation sequence. <i>Manifesting requires the generation of demands, aggregated to the location and time of demand. Resources are then packed into containers which subsequently are manifested onto carriers for transport.</i>	44
Figure 18: Demand simulation process. <i>The demand simulator executes events, updating the network state and adding events as necessary, may generate spatial errors, and logs unsatisfied demands for analysis and manifesting.</i>	45
Figure 19: Impact of discretization policy. <i>Sample demands for two elements with common discrete demands. AP is the aggregation policy value. (a) No discretization treats the demands as continuous. (b) Discretization by element aggregates items on a per-element basis. (c) Discretization by location aggregates items on a per-location basis, enabling spares pooling to locally decrease and smooth demands.</i>	46

Figure 20: Impact of aggregation policy. <i>The item aggregation policy determines at which point demands are aggregated into whole units. (a) No discretization treats demands as continuous. (b) A policy of 0.0 aggregates items at the first demand. (c) A policy of 0.5 aggregates items when half units are accumulated. (d) A policy of 1.0 aggregates items after unit demands.</i>	47
Figure 21: Campaign transports and exploration periods. <i>Each transportation arrival marks the start of its corresponding exploration period.</i>	48
Figure 22: Demand packing. <i>Aggregated demands are split into groups corresponding to how the demands are packed within individual resource containers.</i>	48
Figure 24: Campaign simulation process. <i>The campaign simulator executes events, updating the network (system) state and adding events as necessary, and logs measures of effectiveness.</i>	49
Figure 23: Multi-transport manifesting. <i>Containers manifested on transport 3 (T3) are demanded during exploration period 2 (E2) and must be supplied by either T1 or T2.</i>	49
Figure 26: SpaceNet 2.5 development timeline. <i>SpaceNet 2.5 R1 was released after initial delivery to NASA JPL. SpaceNet 2.5 R2 is currently under development.</i>	51
Figure 25: SpaceNet logo.....	51
Figure 27: Early SpaceNet GUI prototype mockup. <i>Prototype mockups assist development iterations by walking through the campaign definition without time-intensive coding investment.</i>	52
Figure 28: Data table relationships. <i>Data is divided across several tables to represent parent-child relationships between objects.</i>	54
Figure 29: SpaceNet data editor user interface. <i>The data editor provides a GUI for any data source implementation. This example edits an edge from a spreadsheet database.</i>	55
Figure 30: Element sizing model. <i>An element sizing model creates element designs using customized model input parameters.</i>	56
Figure 31: SpaceNet 2.5 user interface. <i>The SpaceNet GUI guides users through the process of modeling and analyzing exploration campaigns.</i>	57
Figure 32: SpaceNet network visualization. <i>Network for a lunar campaign including surface (yellow) and orbital (red) nodes, and surface (green), space (red), and flight (yellow) edges.</i>	58
Figure 33: Campaign visualizations. <i>(a) The process bar chart displays mission events, processes, and transports in time-expanded network. (b) The element bar chart displays individual element movement in time-expanded network.</i>	59
Figure 34: Spatial simulation errors. <i>Spatial simulation errors display (a) in the mission events list and (b) in the mission event dialog to alert the user of error conditions.</i>	59
Figure 35: Scenario feasibility visualization. <i>Cumulative raw and remaining transport capacity must always exceed estimated cumulative demands for feasible campaigns.</i>	60
Figure 36: Supply network visualization. <i>Line width represents net transport capacity and circle diameter represents aggregated exploration period demand mass.</i>	61
Figure 37: Element demand history visualization. <i>Plots demands for spares (COS 4) and packing mass estimates (COS 5) generated by a habitat element. The non-linear response results from reconfigurations of element state during crewed and un-crewed periods.</i>	61
Figure 38: Commonality and scavenging analysis. <i>Displays scavenged common parts filtered by source element and type of resource.</i>	62
Figure 39: Repair policy for optimal mass savings. <i>The set of items to repair to achieve optimal mass savings is found by sorting the repairable items in descending order by the benefit-cost ratio.</i>	63
Figure 40: Packing and manifesting interface. <i>The manifesting tab helps the user to create a valid cargo manifest by guiding through the process of packing and manifesting containers.</i>	63

Figure 41: Network state history. <i>The network state history shows the system state after each event execution. Elements can be inspected for contents including remaining propellant mass.</i>	66
Figure 42: Relative exploration capability history. <i>REC does not increase monotonically over a campaign due to non-exploration mass launched.</i>	66
Figure 43: Location resource history. <i>The resource history during an example campaign shows a build-up of COS 5, representing empty resource containers.</i>	67
Figure 44: ISS resupply network. <i>Visualization of Earth launch sites, landing zones, and the ISS in orbit.</i>	69
Figure 45: ISS resupply mission bat chart. <i>Green squares are element instantiations, yellow lines are flight transports, black squares are elements removed from simulation.</i>	75
Figure 46: ISS resupply scenario feasibility. <i>Cumulative plot of raw supply vehicle capacity, remaining capacity after pre-manifested elements, and aggregated demands indicating feasibility.</i>	76
Figure 47: ISS resupply feasibility with Cygnus COTS failure. <i>Without the Cygnus spacecraft, steady-state infeasibilities first start to appear in early 2013 and only worsen into 2014 and 2015.</i>	77
Figure 48: ISS resupply with advanced water recovery. <i>An advanced water recovery system is delivered in July 2013 reducing demands for water from 3.5 to 0.5 kg/person/day.</i>	78
Figure 50: NEO sortie network. <i>Visualization of space edges between launch, low-Earth orbit, the near-Earth object 1999 AO10, and splashdown in the Pacific Ocean.</i>	79
Figure 49: Flexible path strategy. <i>Timeline of milestones, destinations and capabilities of the Flexible Path strategy, adapted from [21].</i>	79
Figure 51: NEO sortie infeasibilities. <i>(a) The service module burn to return to Earth has a 526.8 m/s delta-v deficit without considering cargo. (b) The 111-day transit to 1999-AO10 alone exhausts the crew module cargo capacity.</i>	82
Figure 52: Modified NEO sortie service module resource history. <i>The science instrument payload (COS 6) is discarded. The service module propellant (COS 1) margin is 250 kg.</i>	83
Figure 53: Modified NEO sortie crew module resource history. <i>All crew consumables (COS 2) are used in the crew module. Packing materials (COS 5) total about 100 kg of the cargo.</i>	83
Figure 54: Lunar surface exploration network. <i>Visualization of flight edges to various surface locations and surface edges between excursion sites near LSP.</i>	85
Figure 55: Lunar outpost process bat chart. <i>Green squares are element instantiations, yellow lines are flight transports, green lines are surface transports, orange and pink squares are element movements and reconfigurations, blue lines are surface explorations, and black squares are elements removed from simulation.</i>	91
Figure 56: Lunar outpost feasibility at LSP. <i>Demands include science payloads, crew consumables and spares with packing mass estimates included.</i>	91
Figure 57: Lunar outpost demands at LSP. <i>COS 2: Crew Consumables, COS 3: Operational Items, COS 4: Maintenance and Spares, COS 5: Packing Mass (Estimated), COS 6: Science Payload, COS 7: Waste Disposal.</i>	92
Figure 58: Lunar outpost ISRU production and consumption. <i>Mass of oxygen available for consumption from (a) ISRU-1 and (b) ISRU-2.</i>	92
Figure 59: Lunar outpost measures of effectiveness. <i>(a) Crew surface days across all surface sites. (b) Exploration capability (crew-kg-days) across all surface sites.</i>	93
Figure 60: Mars exploration network. <i>Visualization of space edges utilizing propulsive burns to reach potential Martian surface sites.</i>	94
Figure 61: Mars exploration process bat chart. <i>Green squares are element instantiations, red lines are space transports, yellow lines are flight transports, orange squares are element movements and reconfigurations, and the blue line is a surface exploration.</i>	99

Figure 62: Baseline Mars exploration crew demands. <i>Under baseline crew member demand rates with a crew of six, demands total 8.1 tons during transport to Mars, 20.8 tons during surface operations, and 9.3 tons during return transport.</i>	100
Figure 63: Modified Mars exploration crew demands. <i>Using a 95% water closure loop and reusable hygiene and waste disposal items, demands are be reduced to 3.6 tons during transport to Mars, 7.6 tons during surface operations, and 4.1 tons during return transport.</i>	100
Figure 64: Modified Mars exploration transport capacity. <i>An additional CFC is included to satisfy demands during transport and the cargo capacity for the MDAV and SHAB have been increased to satisfy demands during surface operations.</i>	102
Figure 65: User experiment baseline spacecraft. (a) <i>Modified Ares V launch vehicle.</i> (b) <i>Modified Orion crew exploration vehicle.</i>	105
Figure 66: User experiment results. (a) <i>SpaceNet subject task milestones.</i> (b) <i>Comparison of treatment (SpaceNet) and control (spreadsheet) subjects' total task time. Whiskers show minimum and maximum recorded values.</i>	108
Figure 67: Manifesting problem diagram. <i>Optimization of the manifesting problem produces a cargo manifest to support exploration goals during simulation.</i>	113
Figure 68: Scheduling problem diagram. <i>Optimization of the scheduling problem produces a mission sequence and cargo manifest to support exploration goals during simulation.</i>	113
Figure 69: Architecting problem diagram. <i>Optimization of the architecture problem produces mission and element architectures, a mission sequence, and a cargo manifest to support exploration goals during simulation.</i>	114
Figure 70: Deterministic versus stochastic demands. (a) <i>Deterministic demands provide a single campaign valuation.</i> (b) <i>Stochastic demands require stochastic manifesting and result in a distribution of campaign valuations.</i>	115
Figure 71: Stochastic event simulation. <i>The manifesting process must be continuously revised during simulation to react to resolved uncertainties.</i>	116
Figure 72: Example object model diagram. <i>The Vehicle interface has two subclass objects, Car and Airplane. A car is comprised of an engine object and aggregates Passenger objects.</i>	117
Figure 73: Location objects. <i>Locations aggregate Elements. The Location interface is expanded by the Node and Edge interface, which both have several subclasses.</i>	118
Figure 74: Resource objects. <i>Three subclasses of the Resource interface include Discrete, Generic, and Continuous Resources.</i>	118
Figure 75: Element objects. <i>The Element interface is comprised of States and Parts. States are comprised of Element Demand Models and Parts aggregate Resources.</i>	118
Figure 76: Element demand model objects. <i>Element Demand Models aggregate Resources. Subclasses include Linear and Sparing by Mass Demand Models.</i>	119
Figure 77: Element object hierarchy. <i>All elements inherit attributes from the base Element class. Subclasses include Human Agents, Resource Containers, and Carriers. Resource Tanks are subclasses of Resource Containers and Surface and Propulsive Vehicles are subclasses of Carriers.</i>	119
Figure 78: Campaign objects. <i>A campaign model is comprised of a Network, Mission, and Manifest models.</i>	120
Figure 79: Mission demand model objects. <i>Mission demand models are very similar to element demand models in that they aggregate Resources.</i>	120
Figure 80: Core event objects. <i>Core element-based events include creation, movement, removal, and reconfiguration. Core resource-based events include addition, demand, and transfer.</i>	120
Figure 81: Composite event objects. <i>Composite events use sequences of core events to build up to complex functionality.</i>	121

List of Tables

Table 1: SpaceNet development history.	16
Table 2: Development use cases summary.	19
Table 3: Surface node examples.	26
Table 4: Orbital node examples.	26
Table 5: Lagrange node examples.	27
Table 6: Common resource classes of supply.	29
Table 7: Continuous resource examples.	30
Table 8: Discrete resource examples.	30
Table 9: Common element classes of supply.	31
Table 10: Resource container examples.	33
Table 11: Resource tank examples.	33
Table 12: Annual demands using a sparing by mass model with a parts list.	38
Table 13: Crew consumables mission demand model parameters.	40
Table 14: Element-based core events list.	41
Table 15: Resource-based core events list.	41
Table 16: Composite events list.	42
Table 17: Campaign measures of effectiveness.	50
Table 18: Campaign options list.	57
Table 19: ISS resupply nodes.	69
Table 20: ISS resupply edges.	70
Table 21: ISS resupply elements.	71
Table 22: ISS resupply missions 0-22 (2010-2011).	72
Table 23: ISS resupply missions 23-45 (2011-2013).	73
Table 24: 2014 ISS resupply missions 46-68 (2013-2015).	74
Table 25: ISS resupply missions 69-77 (2015).	75
Table 26: NEO sortie nodes.	80
Table 27: NEO sortie edges.	80
Table 28: NEO sortie elements.	81
Table 29: NEO sortie mission events.	81
Table 30: Lunar outpost nodes.	86
Table 31: Lunar outpost edges.	86
Table 32: Lunar outpost primary elements.	87
Table 33: Lunar outpost secondary elements.	88
Table 34: Lunar outpost missions 1-10.	89
Table 35: Lunar outpost missions 11-21.	90
Table 36: Mars exploration nodes.	95
Table 37: Mars exploration edges.	95
Table 38: Mars exploration elements.	96
Table 39: Mars exploration mission events.	98
Table 40: Nuclear thermal rocket specific impulse sensitivity.	99
Table 41: User experiment subject comparison.	103
Table 42: Propulsive fuel margins.	106
Table 43: Post-test questionnaire results.	107
Table 44: User experiment feasible designs.	108
Table 45: Levels of space exploration analysis.	112

1 Introduction

As humans grasp the reigns of technology and ride beyond our planet's tenuous atmosphere to more distant destinations over extended time periods, the importance of strategic planning becomes paramount to ensuring the safety and success of future space exploration missions. Over the next 50 years, mission architectures are expected to transition from single, independent sorties to tightly-integrated campaigns spanning many years and involving several stakeholder organizations. As a system-of-systems, a space exploration campaign will require sophisticated logistics and supply chain planning to maintain human presence in remote, hostile environments.

Following the announcement of a new era of human spaceflight and exploration in 2004, the American Institute of Aeronautics and Astronautics (AIAA) reorganized the Space Logistics Technical Committee, defining space logistics as [1]:

The theory and practice of driving space system design for operability, and of managing the flow of materiel, services, and information needed throughout the space system lifecycle.

Addressing the logistics of space exploration is challenging due to two major differences from terrestrial analogs. First, the physics of rocket propulsion provide only a minute fraction of launch mass (typically well below 1%) for resources and items needed during exploration. This narrow margin forces careful selection of what cargo to bring and makes multi-level packing and packaging a high priority. Second, the dynamics of orbital trajectories significantly constrains transportation schedule and duration. If a critical item fails during an exploration, it may take weeks or months to deliver a replacement with no alternatives for resupply.

The first grand experiment in space logistics is already underway. Over the next ten years, the completed International Space Station will serve as the exploration frontier in low Earth orbit, receiving up to eight different vehicles from six launch sites to support a crew of six and over 350 tons of infrastructure. Future campaigns to return to the Lunar or Martian surface will require even more substantial planning to manage long-duration transportation, time-dependent launch windows, and high radiation environments while providing robustness to inevitable failures and maintaining high exploration capability.

The goal of this thesis is to expand on an existing modeling framework developed under the MIT Space Logistics Project to address the challenges of space logistics to evaluate exploration feasibility and quantify exploration value for new and interesting space exploration campaigns.

1.1 Background

The efforts to model, simulate, and analyze the logistics behind space exploration campaigns are closely tied with NASA's human space flight program as well as existing and continuing research areas at MIT. The following sections describe the background research and human spaceflight events leading up to this project, summarized in Figure 1.

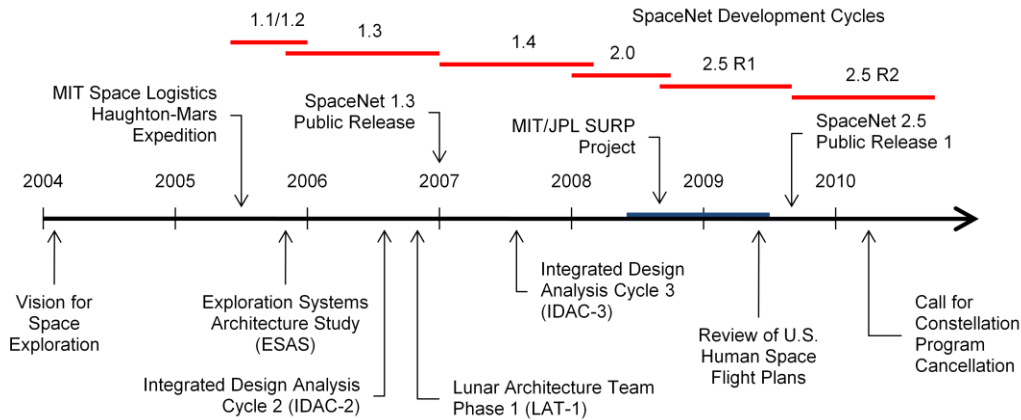


Figure 1: MIT space logistics project timeline. The project has followed development and changes to the NASA Human Space Flight program.

1.1.1 The Vision for Space Exploration

A new era of space exploration was announced by President George W. Bush in January 2004, putting the United States on the path to long-term human and robotic exploration including a return to the Moon's surface in preparation for a human exploration to Mars [2]. To achieve the goals, a tentative timeline was set to develop a crew exploration vehicle by 2008, conduct the first human spaceflight mission by 2014, and explore the Moon with robotic missions by 2008 and with crewed missions by 2020 [3].

Early architectural studies performed by NASA shaped the agency's plan of action to achieve the Vision's goals. The Exploration Systems Architecture Study (ESAS), completed in 2005, developed the *Constellation Program*, which included a general mission architecture of two launch vehicles: one human-rated vehicle to carry crew (Ares I), and one heavy-lift vehicle (Ares V) to carry in-space vehicles, infrastructure, and cargo [4]. The Lunar Architecture Team (LAT) Phase I, completed in 2006, developed on a *Global Exploration Strategy* promoting multi-national cooperation for lunar explorations of increasing duration, focusing on utilizing lunar *in-situ* resources to develop technology necessary for a sustainable Mars mission [5].

1.1.2 MIT Space Logistics Project

In coordination with NASA’s architectural studies, MIT started the Space Logistics Project to build a research base supporting interplanetary supply chain management and logistical analysis necessary for extended exploration campaigns. The project initially studied several terrestrial analogs to space exploration, including operations in remote terrestrial environments such as the Arctic and Antarctic, commercial supply chains, and military logistics operations, culminating in the development of a space logistics framework [6, 7].

SpaceNet, a software tool implementing the space logistics framework, supports campaign analyses and trade studies. SpaceNet models space exploration from a supply chain and logistics perspective, and has been under continuous development over the past five years, as summarized in Table 1. As of 2008, SpaceNet was a fully functional model resulting from two years of development and application within the NASA community. Its analytical capabilities were demonstrated in the Constellation Program (CxP) Integrated Design Analysis Cycle 2 (IDAC-2) in 2006, where SpaceNet was used to trade launch architectures and propellant types and quantify performance drivers for the lunar campaign [8,9].

Table 1: SpaceNet development history.

Designation	Timeframe	Comments
SpaceNet 1.1	2005-2006	Prototype
SpaceNet 1.2	2006	Established visualizations and database
SpaceNet 1.3	2006-2007	Public release, scenarios focused on lunar sorties
SpaceNet 1.4	2007-2008	Scenarios focused on lunar campaigns, demand modeling
SpaceNet 2.0	2008	Internal prototype, code migration to Java, advanced visualizations, SQL database

SpaceNet 1.3 was released to the public in 2007 as a MATLAB® application and graphical user interface (GUI) supported by an Excel database [10]. The overall framework of space logistics campaign analysis was further expanded upon and summarized in Shull’s conference paper and thesis, establishing a solid foundation for future work efforts [11, 12]. SpaceNet 1.4 included additional development to improve the ability to analyze long-duration lunar surface campaigns in support of CxP IDAC-3 [13, 14].

In 2008, there was an effort to migrate SpaceNet from MATLAB® to a cross-platform, web-accessible implementation. SpaceNet 2.0 served as an internal Java Web Start prototype that utilized modular aspects of object-oriented programming to provide a platform for future extension and development. The flexibility of the revised architecture was demonstrated by Armar’s paper on cargo revenue management techniques for determining optimal manifests [15].

1.1.3 Human Space Flight Review

Under increasing pressure to reconcile the forward budget for the Constellation Program and the Vision for Space Exploration, the Obama Administration requested an independent review of NASA's Human Space Flight (HSF) program in May 2009 [16]. The major concern was the federal budget allocated to NASA could not support both the sustainment of the International Space Station *and* the development of the new systems required to meet the program of record schedule. NASA established a blue-ribbon panel of experts, chaired by Norman Augustine, to perform the review over the summer of 2009 [17].

The "Augustine Commission" submitted a final report in October 2009 outlining several options for the future of the human space flight program [18]. The committee provided three classes of options: a "constrained" option adhering to the forward budget profile but would not return to the moon before 2030, a "moon-first" option using an expanded budget to support a campaign similar to the program of record with a lunar return in the mid-2020's, and a "flexible path" option using an expanded budget but would first focusing on non-lunar missions, such as near-Earth objects (NEOs).

In February 2010, President Obama released the 2011 fiscal year budget, influenced by the findings of the Augustine Committee [19]. Major decisions included extending funding for the International Space Station though 2020, cancelling the Constellation Program, and renewing focus on commercial space transportation. Overall, NASA's budget was to increase by \$6.0 billion over five years, bringing the total funding for NASA to \$100 billion. Administrator Bolden was quoted that the new investments "...will enable our path beyond low Earth orbit through development of new launch and space transportation technologies, nimble construction capabilities on orbit, and new operations capabilities..." and may include "...new and novel approaches to spaceflight such as in-orbit fuel depots and rendezvous and docking technologies, and closed-loop life support systems..." [20]. However, as of the time of writing, no final decisions have been approved by Congress as to the future of the HSF program.

1.2 Project Motivation

The space logistics framework and code used in SpaceNet resulted from several years' worth of iterative effort. Also, the targeted analysis focus had shifted over time, from investigating individual sorties (single missions) to long-duration campaigns and from lunar-specific campaigns to more general exploration scenarios. Several areas were identified to motivate improvements to the existing model to progress research and accommodate the programmatic changes within the HSF program.

1.2.1 Space Logistics “-ilities”

One of the topics of interest emerging from the CxP IDAC-3 analysis was the ability to model and “quantify the ‘-ilities’ as part of an end-to-end campaign scenario, such as the reconfigurability, reusability, commonality, and repairability of elements” [13]. Representing emergent lifecycle properties of the space exploration system, the “-ilities” could contribute to decisions in the early design phase of the system elements. This addition was developed under a Strategic University Research Partnership (SURP) with Caltech’s Jet Propulsion Laboratory.

Implementation of these “-ilities,” however, extended beyond the existing domain of the space logistics framework. To account for the lower-level nature of the analysis, the task was separated into two components. The first would model the “-ilities” at the subsystem level; the second would propagate their impacts back to the system and campaign level. The following features were targeted in SpaceNet to handle this new domain of analysis:

1. Model the *reconfigurability* of systems such that duty cycles, functional capabilities, and operational states can be chosen dynamically depending on the mission plan. Challenges include sufficiently describing operational states and dynamically changing the underlying models generating demands during simulation.
2. Analyze the *reuse* of systems beyond the utilization of static, pre-deployed assets, specifically surface mobility systems. Challenges include decoupling elements from missions and persisting elements and their effects on demands throughout the simulation.
3. Analyze the impact of *commonality* and scavenging between systems such as surface rovers, habitation modules, and propulsive stages. Challenges include adding the ability to scavenge components from existing elements.
4. Model surface *repair* activities and how they impact system availability, spare parts and tool demands, and the time available for exploration. Challenges include adding the ability to choose repair actions and evaluating the impact on the overall scenario.

1.2.2 Flexible Modeling Framework

Several “use case” scenarios were identified during the development phase to develop a flexible modeling framework capable of adapting to new scenarios including those emerging from future programmatic changes. Five cases of increasing difficulty, summarized in Table 2, were created to guide development and prevent hard-coding assumptions.

Table 2: Development use cases summary.

Use Case	Primary System	Flights	New Challenges
Lunar Sortie	Earth-Moon	1	Match existing modeling capability
ISS Resupply 2010-2015	Earth	40	Non-zero initial state Complex models may be required for realistic demands
Lunar Hub-Spoke (Outpost)	Earth-Moon	20	Long duration, detailed campaign High level of element reuse Dependence on ISRU technology
Lunar Spoke-Hub (Global)	Earth-Moon	20	Surface-to-surface transportation
Martian Human-Robotic	Earth-Mars	16	Infrequent launch windows Variable flight durations Consideration of low-thrust propulsion Long-term crew health concerns

The Lunar Sortie models a single Constellation-class mission for a seven-day duration exploration of the Lunar South Pole (LSP), matching existing modeling capability.

International Space Station (ISS) Resupply for 2010-2015 models the sustainment of ISS after shuttle retirement utilizing a combination of International Partners (IP) and Commercial Orbital Transportation Services (COTS).

Lunar Hub-Spoke (Outpost) models the build-up and sustainment of an outpost located on the rim of Shackleton Crater at the Lunar South Pole. The surface infrastructure and mobility elements are delivered over the span of seven years, leading up to a continuous human presence on the moon. The campaign is influenced by mission architectures investigated by the Constellation Architecture Team (CxAT).

Lunar Spoke-Hub (Global Exploration) relies on reusable surface mobility elements to model a global exploration of the moon over a seven year period. Initial missions are targeted at building up infrastructure at the Lunar South Pole. Later missions perform portions of a long-term traversal to the North Pole using mobile habitation elements.

The Martian use case models a precursor robotic exploration of the Martian surface, followed by human-crewed missions based on Mars Design Reference Architecture 5.0 [21]. The robotic precursors are sent to select one of three target surface locations for the human missions.

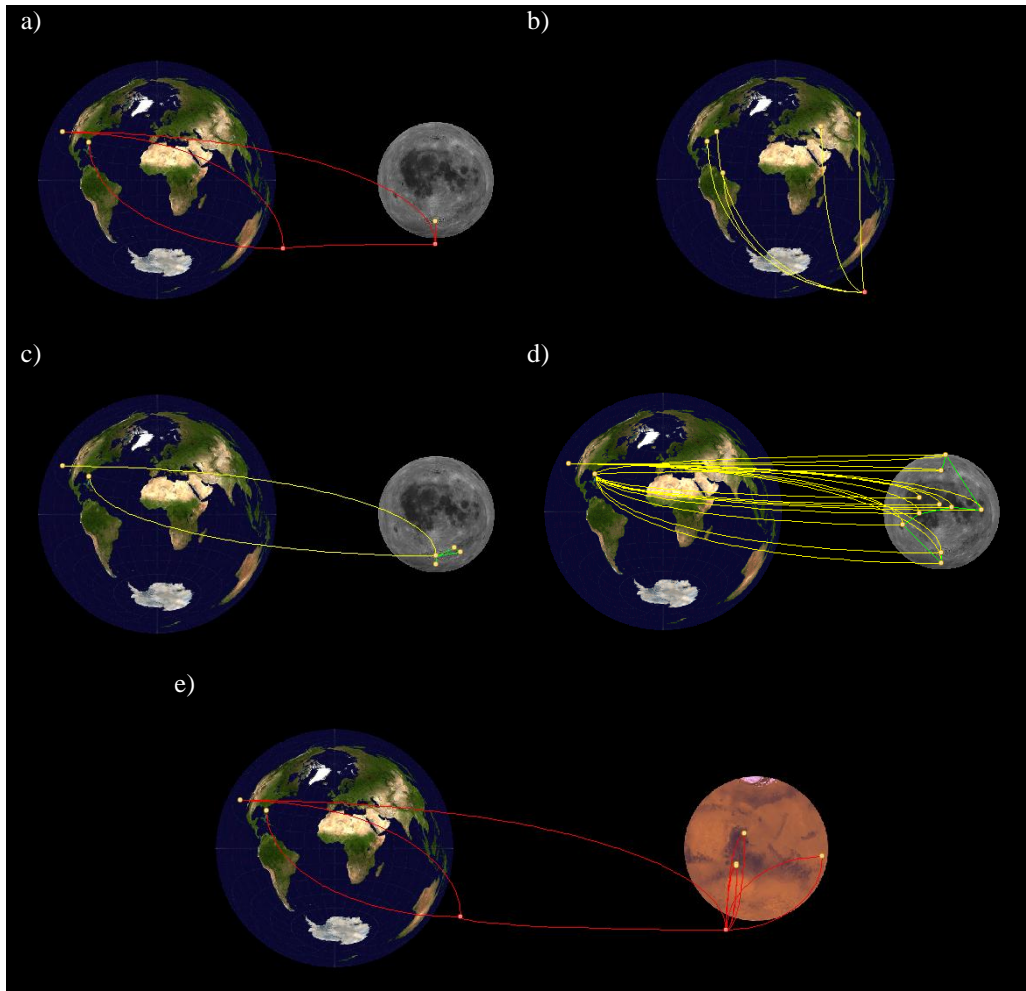


Figure 2: Use case scenario networks. (a) *Lunar sortie.* (b) *ISS resupply post shuttle retirement.* (c) *Lunar hub-spoke (outpost).* (d) *Lunar spoke-hub (global exploration).* (e) *Martian human-robotic exploration.*

1.2.3 Modular Implementation Architecture

Aside from a flexible modeling framework, it was also a priority to design the tool in a modular fashion to provide a platform from which to establish future extensions and development. The development of SpaceNet up to version 1.4 used procedural programming methods in MATLAB®, common for engineering computation. The underlying structure of procedural programs resembles a bus for both memory and the user interface, shown in Figure 3. Data is managed in memory accessible to any function called by the user via a monolithic user interface.

In typical development progression, the organization of memory often does not keep up with the creation of procedures. Shortcuts enabling access to previously unrelated data contribute to a code base that is difficult to understand and more challenging to track, as potentially every procedure could alter shared

memory. In addition, there may be limited documentation or definition of interfaces between procedures, complicating extensions or future development.

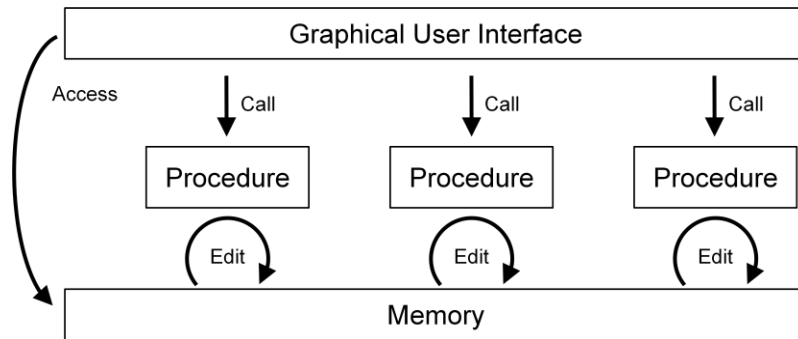


Figure 3: Procedural programming diagram. Procedures called by a monolithic GUI edit portions of a shared memory.

SpaceNet 2.0 took the first steps into the development of an object-oriented architecture. Object-oriented methods establish object classes to couple memory structure and function and separate GUI functionality, as shown in Figure 4. Memory is not widely accessible; rather it is limited to the scope of each object so unrelated objects cannot alter each other’s data, equivalently, no knowledge of unrelated objects is necessary to understand an object. Interfaces between objects are also inherently defined, helping to insulate against change propagation and enabling easier extensions.

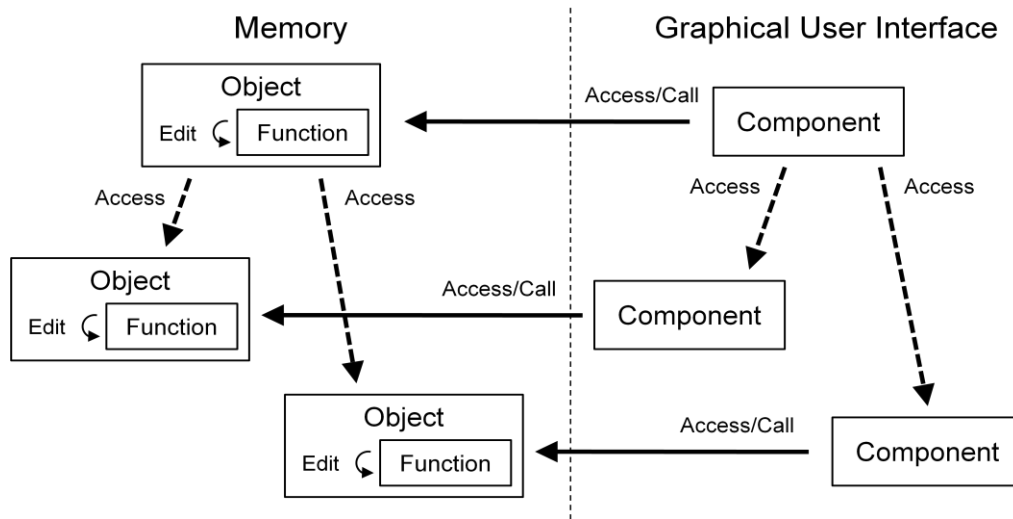


Figure 4: Object-oriented programming diagram. Objects reside in memory and carry functional capabilities limited in scope. Objects are accessed by parallel GUI components.

The graphical user interface (GUI) can be divided into components to mirror the object structure providing specialized interfaces to individual objects. As the GUI is separated from the underlying modeling code, changes and updates can be easily implemented without a substantial update or revision to

the model. This is especially important when considering serialized data (saved files), which often requires identical models for compatibility but has no restriction on modifications to the GUI.

1.2.4 User Community

Another goal motivating additional SpaceNet development was the desire to open the MIT Space Logistics Project to a wide user community, potentially extending to applications outside field of space exploration logistics. A three-fold strategy was undertaken towards this goal.

First, the usability of the software tool was targeted for dramatic improvement. From the user's perspective, user-centric and task-centric methods should be embraced to improve the user interface. From the developer's perspective, a modular architecture should be developed, isolating and abstracting components likely to be expanded upon in the future. Second, the software tool should be migrated to a license-free platform as many potential users do not have access to a MATLAB® license. Finally, an open source coding strategy was desired to provide transparency of the space logistics framework and allow the user community to self-sponsor modifications or expansions as necessary.

1.3 Related Research and Literature

Aside from the previous references within the MIT Space Logistics Project, several research papers investigate topics related to space exploration campaign modeling and simulation.

Cirillo provides an overview of a strategic analysis methodology that "... provides integrated analysis of system performance over the full system life cycle..." [22]. Strategic analysis investigates three main components of space exploration scenarios: performance, affordability, and risk. He identifies the differences between macro-logistics (resource movement between locations) and micro-logistics (resource movement within a location) and the interplay between deterministic and probabilistic analysis. The methodology is demonstrated with an example scenario focusing on lunar surface exploration.

Stromgren takes a closer look at micro-logistics analysis by modeling gas/liquid and solid logistics usage using system dynamics for a lunar outpost exploration scenario [23]. Pressurized Logistics Modules (PLMs) and Cargo Transfer Bags (CTBs) are used as logistics carriers. A visualization module also enables a graphical depiction of the consumables during simulation.

Andraschko presents Campaign Analysis Manifest Tool (CMAT), a deterministic software model to support strategic analysis activities within the Constellation Architecture Team – Lunar (CxAT-Lunar) [24]. CMAT is very similar to SpaceNet in that exploration scenario definitions are used as inputs and

logistics feasibility is evaluated via an iterative manifesting process. Logistics considered include pressurized crew logistics (food, hygiene, clothing, operational supplies, etc.), pressurized and unpressurized logistics and maintenance, pressurized and unpressurized science, and consumables (oxygen, nitrogen, and water). After the CxP IDAC-3 study in which SpaceNet 1.4 was used, CMAT was selected as the primary tool for future analysis of lunar architectures.

Within the context of space logistics “-ilities,” Siddiqi investigates the spare parts requirements for missions utilizing reconfigurability and commonality between elements [25]. She describes the benefits of commonality for scavenging spares from non-operational elements and presents a method for determining spare parts demands for reconfigurable states where the element may transition between several operational states with different demand profiles. The theory is demonstrated with the discrete event simulation of an example Mars exploration mission.

Kline describes a hybrid parametric-analytic sparing model to estimate the spare parts needed to achieve specified system availability [26]. Commonality of spare parts is investigated both within elements and across elements. The sparing model is implemented in the Spacecraft Sustainability Model™ (SSM) and demonstrated with an example Mars exploration mission based on a NASA design reference mission.

1.4 Project Overview

A decision was made to restructure the space logistics modeling framework from the ground up to achieve the project goals. The modeling framework to represent space exploration scenarios is defined using object-oriented methods. The lowest-level domain objects including locations, resources, and elements at the core of any logistical analysis regardless of application are discussed in Chapter 2. The higher-level campaign objects more specific to space exploration including missions, events, and manifests and the discrete event simulation engines used to drive analysis are discussed in Chapter 3.

Parallel to the efforts in developing modeling frameworks, logic flow diagrams and modular prototypes developed from a user’s perspective guide how a scenario should be constructed and analyzed. A user interface was generated by breaking down an exploration campaign into the high-level modules a user would define, including the network definition, mission and transportation specification, demand modeling, cargo manifesting, and simulation and visualization. The resulting software implementation, SpaceNet 2.5, is discussed in Chapter 4.

Applications of SpaceNet are presented in Chapter 5. Four case studies illustrating the flexibility to model vastly different scenarios are presented including initial results and analysis. The case studies include the

resupply of the International Space Station from 2010-2015, a sortie mission to a near-Earth object for human tele-operated exploration, the buildup of a Lunar outpost leading to continuous human presence, and a crewed surface exploration on Mars. In addition, a controlled user experiment is presented to highlight potential benefits of using the SpaceNet tool over independent analysis methods. Finally, Chapter 6 provides some concluding thoughts, including a summary of the primary contributions of this project and a strategic outline towards expanding the space logistics framework in future research.

2 Domain Model

The core of the space logistics modeling framework is focused on the *domain*, or physical state of components within an exploration campaign. Components of the domain have evolved development within the MIT Space Logistics Project. A typical iteration loop starts with specialized capabilities later expanded to more general cases through abstraction. For example, SpaceNet 1.4 generalized the SpaceNet 1.3 model to allow multi-mission campaigns. As a substantial change from previous models, however, object-oriented methods are used to emphasize the concepts of modularity and flexibility. Object model diagrams illustrating the modeling framework visually are available in Appendix A.

The domain model is divided into three categories: network components, resources, and elements. Although specific to space logistics, the domain representation of the modeling framework is extensible to new applications because of its focus on physical representation. In other words, any model for logistics, exploration, or related applications will incorporate similar domain objects.

2.1 Network Components Model

Network components represent locations and paths containing elements and nested resources within a time-expanded network. Using network terminology, nodes and edges are two types of network components that serve as interfaces to specialized components. New implementations of an interface would be interchangeable with the existing objects in any generic application, such as those in most underlying simulation code.

2.1.1 Nodes

Nodes represent time-invariant locations in the campaign. For some nodes, such as surface locations on a planetary body, this definition is clear as the surface location does not move over time. In other nodes, such as stable elliptical orbits, the time-invariant nodal representation corresponds to the entire orbit.

The primary purpose of nodes is to set the limits of resource sharing. Any resources at a node are assumed to be equally accessible to all co-located elements to satisfy demands. This definition means that resources are shared at the level of nodes, or equivalently, demands for resources are aggregated to nodes. In campaigns more focused on micro-logistics, this assumption may not be desired – for example, resources may not be explicitly shared between space station nodes, or habitats at a surface outpost may require independent resources to satisfy demands. In these cases, pseudo-nodes can be defined to handle the desired level of sharing or demand aggregation even though the nodes may be in close proximity.

Subclasses of nodes include surface nodes corresponding to locations on a planetary body, orbital nodes corresponding to stable orbits, and Lagrange nodes corresponding to Lagrange points between celestial bodies. Nodes of any subclass may also be designated as *source nodes*, meaning they may act as a source of resources. Launch sites such as the Kennedy Space Center (KSC) are most often designated as source nodes. Source nodes play an important part in the manifesting process, detailed in Section 3.3.

Surface Nodes

Surface nodes represent a location on a planetary body given by a latitude and longitude. The latitude and longitude values are primarily used to drive projections of the surface node onto the planetary body in visualizations, but future development may utilize this information to support other calculations. Table 3 lists commonly-used surface nodes.

Table 3: Surface node examples.

Surface Node	Abbrev.	Body	Latitude	Longitude
Kennedy Space Center	KSC	Earth	28.6°N	80.6°W
Lunar South Pole	LSP	Moon	89.9°S	0.0°E
Apollo 17 Landing Site	A17	Moon	20.2°N	30.8°E
Viking 1 Landing Site	VK1	Mars	22.5°N	50.0°W

Orbital Nodes

Orbital nodes represent elliptical orbits about a celestial body given by an inclination and altitude apoapsis and periapsis. As orbital nodes represent time-invariant locations, two elements co-located at an orbital node are assumed to be co-positioned in orbit, though multiple pseudo-nodes could be created with similar orbital parameters to enforce differentiation. Similar to surface nodes, the orbital parameters presently drive visualizations rather than serving as inputs to calculations or logical instructions. Table 4 provides examples of commonly-used orbital nodes.

Table 4: Orbital node examples.

Orbital Node	Abbrev.	Body	Inclination	Apoapsis [km]	Periapsis [km]
Low Earth Orbit	LEO	Earth	28.5°	296	296
International Space Station	ISS	Earth	51.6°	360	347
Low Lunar Polar Orbit	LLPO	Moon	90.0°	100	100
Low Mars Equatorial Orbit	LMEO	Mars	0.0°	500	500

Lagrange Nodes

Lagrange nodes represent the five libration points (zones of no net gravitational acceleration from the celestial bodies), as illustrated for reference in Figure 5. Though current use of Lagrange points for logistics is limited to observational satellites, future missions may use them to maintain fuel or other supply depots. Table 5 provides examples of commonly-used Lagrange nodes.

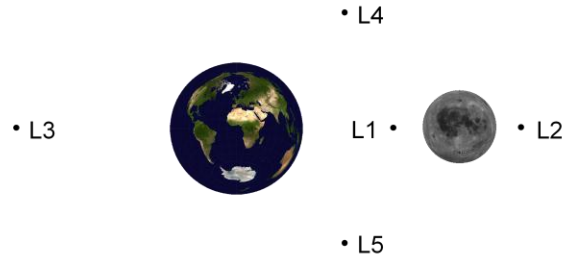


Figure 5: Lagrange point positions. *L1, L2, and L3 are stable locations, L4 and L5 are unstable and require station-keeping. Not to scale.*

Table 5: Lagrange node examples.

Lagrange Node	Abbrev.	Major Body	Minor Body	Position
Sun-Earth L1	SEL1	Sun	Earth	1
Sun-Earth L2	SEL2	Sun	Earth	2
Earth-Moon L2	EML2	Earth	Moon	2

2.1.2 Edges

Edges represent time-invariant connections between nodes. During simulation, elements traverse edges using a *transport event* (see Section 3.2.2). Although an edge is a time-invariant path, the transport event may impose time-dependent characteristics. Elements co-located on an edge are assumed to share resources in the same way that elements co-located at a node share resources.

Subclasses of edges include surface edges corresponding to paths between surface nodes, space edges corresponding to impulsive propulsion trajectories, and flight edges corresponding to known transportation architectures simplified to eliminate propulsive requirements.

Surface Edges

Surface edges represent paths between two surface nodes, parameterized by a distance and traversed using a surface transport. Surface edges are not directional, meaning traversal is possible in both directions. Surface edges enable movement of previously-deployed elements to new locations while modeling demands

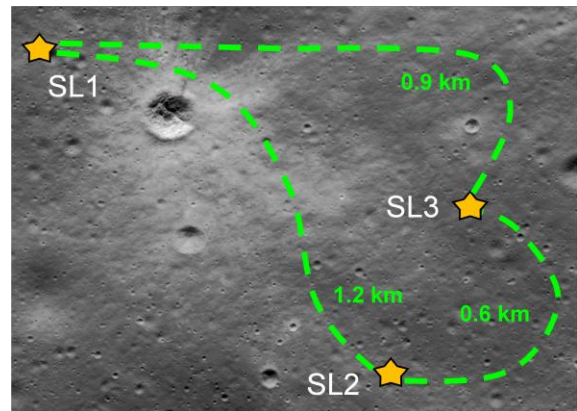


Figure 6: Surface edge example. *Locations 1, 2, and 3 are connected with three surface edges.*

during transport. Figure 6 illustrates surface edges between three surface nodes distances in kilometers.

Space Edges

Space edges represent directional paths between two nodes requiring a series of propulsive burns for traversal. Each burn is designated by an execution time relative to the start of the transport, required change in velocity (delta-v), and a specification of using either a propulsive vehicle Orbital Maneuvering System (OMS) or Reaction Control System (RCS). Figure 7 illustrates space edges used in lunar transportation network to represent launch, trans-lunar injection/lunar orbit insertion, descent, ascent, and trans-Earth injection.

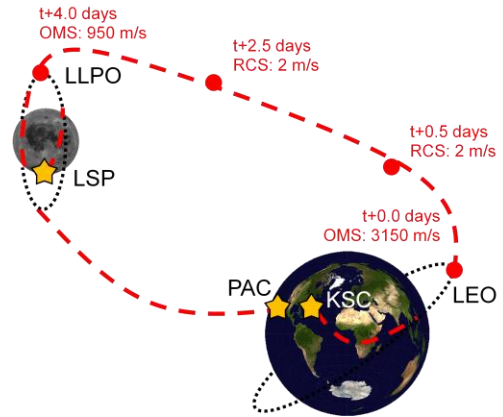


Figure 7: Space edge examples. Edges correspond to propulsive burn sequences. The burns are highlighted for the space edge between nodes LEO and LLPO.

Flight Edges

Flight edges represent an abstracted path between two nodes without relying on propulsive burns. They commonly use a standard flight architecture known to feasibly deliver a set amount of cargo and crew (e.g. a launch vehicle). Flight edges are used when detailed analysis of individual space edges and burns is not needed; common applications include extended-duration campaigns with many missions utilizing similar transportation methods between common nodes. A flight is described by its duration, the crew capacity, and the cargo capacity. Figure 8 illustrates an example network using flight edges to abstract a series of space transports.

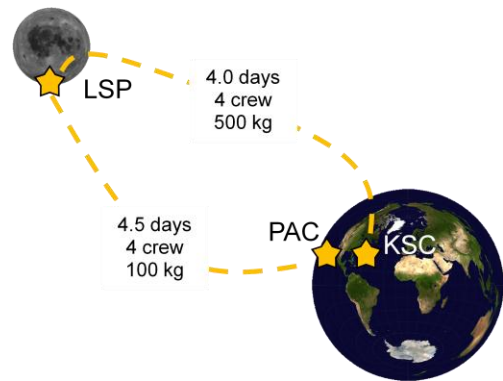


Figure 8: Flight edge examples. Edges correspond to pre-determined flight architectures with crew and cargo limits.

2.2 Resource Model

Resources are substances consumed to satisfy demands. Resources only specify types of substances rather than the instantiated items expended during simulation – the name “resource type” has been used in previous modeling frameworks to make this distinction. Resources must always be contained by elements, specifically resource containers, which maintain the quantity of each resource.

All resources are assigned a functional *class of supply* (COS) based on military and NASA techniques for classifying cargo by its function [27]. Classes of supply are used to abstract and group similar resources to simplify demand models and visualizations. There are ten primary classes of supply and many sub-classes of supply that define cargo in more detailed terms. As a general guideline, classes 1-4 and 7 are commonly used for resources and 5, 6, 8, 9, and 10 for elements. Table 6 lists the current set of classes of supply commonly used for resource classification.

Table 6: Common resource classes of supply.

COS	Description and Sub-Classes
1	Propellants and Fuels 101: Cryogenics, 102: Hypergols, 103: Nuclear Fuel, 104: Petroleum Fuels, 105: Other Fuels, 106: Green Propellants
2	Crew Provisions 201: Water and Support Equipment, 202: Food and Support Equipment, 203: Gases, 204: Hygiene Items, 205: Clothing, 206: Personal Items
3	Crew Operations 301: Office Equipment and Supplies, 302: EVA Equipment and Consumables, 303: Health Equipment and Consumables, 304: Safety Equipment, 305: Communications Equipment, 306: Computers and Support Equipment
4	Maintenance and Upkeep 401: Spares and Repair Parts, 4011: Pressurized Spares, 4012: Unpressurized Spares, 4013: Repair Parts, 402: Maintenance Tools, 403: Lubricants and Bulk Chemicals, 404: Batteries, 405: Cleaning Equipment and Consumables
7	Waste and Disposal 701: Waste, 702: Waste Management Equipment, 703: Failed Parts

In addition to a class of supply, resources are described with a unit of measurement, unit mass (kilograms per unit), and unit volume (cubic centimeters per unit). These additional properties allow customizable resources with arbitrary units.

There are three subclasses of resources used in analysis: generic, continuous, and discrete. All three share the same interface enabling a mixture of different levels of abstraction in a single analysis. For example, it is possible to use generic resources for crew demands, continuous resources for propellant demands, and discrete resources for spares demands.

Generic Resources

Generic resources are the simplest type of resource which is directly tied to a specific class of supply. They are most useful for performing high-level analysis using approximations for demands by class of

supply. Generic resources are always expressed in units of kilograms with a unit volume estimated from the class of supply's average density, though volume is can be disregarded for high-level analysis.

Continuous Resources

Continuous resources are customized resources having continuously-variable quanta such as propellants, liquids, and gases. Continuous resources are usually measured in units of kilograms but other units can be defined. Zero-mass resources such as electricity could be modeled using continuous resources having zero unit mass and volume. Table 7 lists examples of continuous resources.

Table 7: Continuous resource examples.

Continuous Resource	COS	Units	Unit Mass [kg/unit]	Unit Volume [cm ³ /unit]
Ammonium Perchlorate Fuel	105	kg	1	500
Potable Water	201	kg	1	1000
Lunar Samples	603	lb	0.45	200
Electricity	105	kW-h	0	0

Discrete Resources

Discrete resources are customized resources having indivisible units. Examples of discrete resources may include supply items such as a packaged meal or a toothbrush, or components used as spare parts. Depending on the level of analysis, the line between discrete resources and elements may be blurred as both represent indivisible objects. Resources, however, are the product of demand rather than instantiation, and only elements can generate demands.

The units for discrete resources are always “item” though the unit mass and unit volume can be set to correspond to the individual items. Utilization of discrete resources is often different from continuous resources, as the item may not actually be used in the sense that it no longer exists (e.g. maintenance tools). In this sense, utilization is understood to be the demand for the existence of a new item. Table 8 lists examples of discrete resources.

Table 8: Discrete resource examples.

Discrete Resource	COS	Units	Unit Mass [kg/unit]	Unit Volume [cm ³ /unit]
Leisure Book	206	item	0.15	700
Repair Wrench	402	item	0.24	200
Computer Motherboard	4011	item	0.65	4500

2.3 Element Model

Elements have a rich hierarchy to represent a variety of specialty capabilities. Elements are each uniquely identified upon instantiation in a campaign. They may contain parts, incorporating resources to help drive demands for spares and other activities such as repair or scavenging, and may have one or more operational states of several categories, enabling multi-modal demand generation.

Similar to resources, elements are also assigned a class of supply. Table 9 lists classes of supply commonly assigned to elements. Elements also have a specified mass, volume, and stowage environment (pressurized or unpressurized) which determines allowable nesting relationships.

Table 9: Common element classes of supply.

COS	Description and Sub-Classes
5	Stowage and Restraint 501: Cargo Containers and Restraints, 502: Inventory Management Equipment
6	Exploration and Research 601: Science Payloads and Instruments, 602: Field Equipment, 603: Samples
8	Habitation and Infrastructure 801: Habitation Facilities, 802: Surface Mobility Systems, 803: Power Systems, 804: Robotic Systems, 805: Resources Utilization Systems, 806: Orbiting Service Systems
9	Transportation and Carriers 901: Carriers, Non-propulsive Elements, 902: Propulsive Elements

2.3.1 Element Hierarchy

Figure 9 illustrates a hierarchy of elements developed to account for specialized functions while allowing as many common attributes as possible. The base element generates demands for resources using parts and states. Subclasses of elements include human agents (crew), robotic agents (reserved for future expansion), resource containers and tanks containing quantities of resources, and carriers containing other elements. Subclasses of carriers include propulsive vehicles which can traverse space edges using propulsive burns and surface vehicles which can traverse surface edges at a specified speed.

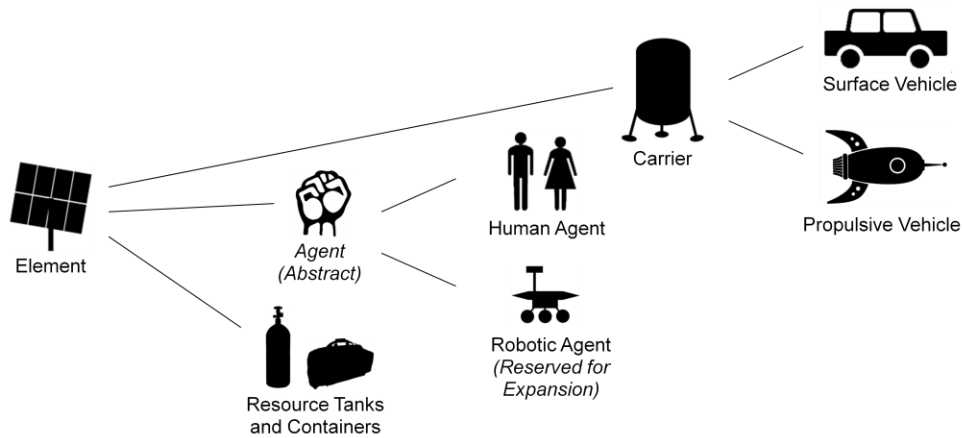


Figure 9: Element hierarchy. A hierarchy of elements provides specialized capabilities extensible for future expansion while establishing a common set of inherited properties.

The hierarchy of elements is likely to be expanded with future development by building on the existing functionality. For example, if low-thrust (ion propulsion) vehicles were desired, a new sub-class of the Carrier interface could be created similar to the existing Propulsive Vehicle.

Elements

The simple element forms the base of the element hierarchy. Elements cannot contain other elements or resources or traverse edges under their own power but may generate or contribute to demands for resources. Common objects modeled as elements include infrastructure objects such as solar panels, radio communication links, and scientific payloads.

Human Agents

Human agents represent crew members. Presently, the only difference from basic elements is that human agents specify a fraction of time that the agent is available (typically estimated at 2/3). Human agents count towards crew capacity constraints for carriers, and are also accounted for in several measures of effectiveness (see Section 3.4.2).

Future development is targeted at the “agent” interface, which both human agents and future robotic agents will implement. Human and robotic agents are distinguished from other elements in that they will be able to perform *tasks*, such as preventative or corrective maintenance.

Resource Containers

Resource containers are elements with the primary purpose of containing one or more types of resources up to a mass or volume limit. Table 10 lists resource containers examples including cargo transfer bags (CTBs) and SPACEHAB Oceanering Space Systems (SHOSS) boxes for spare parts.

Table 10: Resource container examples.

Resource Container	COS	Tare Mass [kg]	Volume [m ³]	Max Cargo Mass [kg]	Max Cargo Volume [cm ³]
CTB	501	1.8	0.0529	45.4	52900
Double CTB	501	3.6	0.1058	90.7	105800
Half CTB	501	1.0	0.0248	4400	24800
SHOSS Box	501	120	0.4444	200	444400

Resource Tanks

Resource tanks are elements similar to resource containers but can only contain one type of resource up to a set amount of units of resource. Since the capacity does not depend on mass or volume limits, resource tanks can be used for zero-mass resources. Typical applications for resource tanks are for fuel, propellant, water, or gas storage. Table 11 provides examples of a few resource tanks.

Table 11: Resource tank examples.

Resource Tank	COS	Tare Mass [kg]	Volume [m ³]	Resource	Max Amount [units]
Gas Tank	501	108	2.75	Generic COS 203	100 [kg]
Gas Tank Derivative	501	10.8	0.275	Generic COS 203	10 [kg]
Liquid Tank	501	34.4	0.0748	Generic COS 201	74.8 [kg]
Liquid Tank Derivative	501	11.5	0.0249	Generic COS 201	24.9 [kg]
Notional Battery	501	90	0.05	Electricity	15.6 [kW-h]

Carriers

Carriers are elements containing, contain, or carry, other elements up to a capacity limits for crew size, mass, and volume. Though similar in function to a vehicle or habitat, the term “carrier” was selected because “vehicle” inferred the capability of independent movement and “habitat” inferred capacity for at least one crew member. Carriers provide a cargo environment (pressurized or unpressurized) which defines what elements can be carried. As each carrier may only have either pressurized or unpressurized cargo, multiple carriers can be used to model multiple cargo environments. Common applications for carriers include habitation modules, non-propulsive in-space vehicles, and logistics and stowage modules. Carriers are also often used for abstracted elements used in flight transports.

Surface Vehicles

Surface vehicles are a subclass of carriers having surface mobility capabilities to traverse surface edges. They also have an integrated resource tank to carry fuel independent from cargo capacities.

Propulsive Vehicles

Propulsive vehicles are a subclass of carriers having propulsion capabilities needed to traverse space edges. Propulsive vehicles have up to two integrated resource tanks, one to supply an Orbital Maneuvering System (OMS) engine, and one to supply a Reaction Control System (RCS) engine. Propulsive vehicles may have one or both systems, availability specified by a specific impulse, and may share a common fuel tank for both systems.

2.3.2 Element Parts

Parts represent quantities of resources applied in an element. Though assigning parts to an element does not change spares demands alone, the parts list may be used to drive demands in sparing models. Additionally, parts can be assigned a mean time to failure (MTTF) to further inform detailed sparing models. If a part is designated as repairable, it is assigned a mean repair time (MRT) and mean repair mass (MRM) to cover low-level components and tooling, though the decision of whether to repair is made at the campaign level.

Parts are inferred as common if multiple elements are assigned the same underlying resource. Common parts are advantageous for spares pooling and scavenging from decommissioned elements. Supplying an exhaustive list of parts is often not feasible, especially for abstracted elements, high-level studies, or yet-to-be-designed components. In practice, a few notional parts that exhibit desired features such as commonality and repairability can be used to perform analysis. Figure 10 illustrates a notional case where two elements, a power plant and a lander, share a common power supply resource type. In this case, 10% of the power plant by mass is common with 20% of the lander by mass.

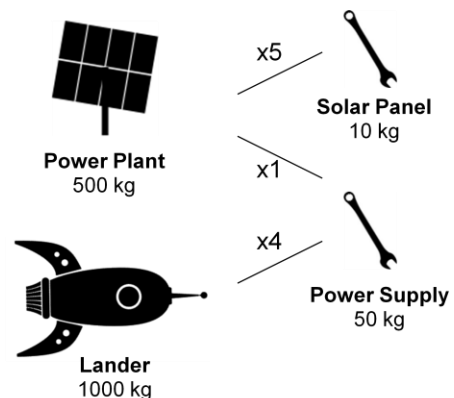


Figure 10: Part commonality with shared resource types. The Power Plant and Lander elements share a common Power Supply resource.

2.3.3 Element States

States model the operational capability of an element, allowing changes to the demand profile for an element during exploration. Element operational states fall under one of the

five following categories:

- **Active:** nominal operational level
- **Special:** short-duration increased operational level, often used to represent EVAs
- **Quiescent:** reduced operational level
- **Dormant:** minimal operational level, often used for elements before activation or deployment
- **Decommissioned:** permanent non-operational level; the element's parts are available to scavenge

An element can have any number of possible states and one current state which changes through undergoing a reconfiguration. Each state specifies a set of demand models that generate demands for the element while it is in that state. Figure 11 depicts a typical reconfiguration sequence.



Figure 11: Element reconfiguration sequence. *An element may experience several operational states throughout a campaign including dormant before delivery, active or quiescent during operation, and decommissioned state to signal availability of parts for scavenging.*

2.3.4 Element Demand Models

Element demand models generate demands for resources during simulation from the perspective of each element, which are then aggregated by node. This allows multi-fidelity element-level demand models to supplement the traditional method of modeling demands using parameterized functions on a global perspective (see Section 3.2.1). As element states are changed over the course of a campaign, different demand models are utilized to create a complex composite demand model that would be difficult to achieve from a global perspective.

Two examples of simple element demand models are presented below. Additional demand models are likely to evolve in future development to represent more complicated phenomena such as boil-off, shelf life, and advanced sparing models.

Linear Demand Model

The linear demand model produces demands for a set of resources linearly with time. Its cumulative demands for resource i can be expressed as a function of the simulation time t in Eq. (1).

$$D_i(t) = a_i + b_i \cdot t \quad (1)$$

However, as the simulation runs in discrete time intervals, the demands generated at each event execution are expressed as a function of the number of days since the previous event execution Δt , shown in Eq. (2). Functionally, the check for the first aggregation is performed using a Boolean flag that is initialized to zero and set to one after the first aggregation.

$$d_i(\Delta t) = \begin{cases} a_i + b_i \cdot \Delta t & \text{if first aggregation n} \\ b_i \cdot \Delta t & \text{otherwise} \end{cases} \quad (2)$$

The resulting demands resemble a step function, illustrated in Figure 12, though the demands are being produced at a linear rate. In practice, the constant terms a_i are used for fixed demands such as tools, safety, or health equipment and the linear terms b_i are used for recurring demands such as food, water or oxygen for human agents, or electricity for other elements.

Sparing by Mass Demand Model

The sparing by mass demand model is a simple sparing model assuming the mass of spare parts demanded by an element annually is proportional to a percentage of its dry mass. The spare parts rates can be split by environment into pressurized spares (COS 4011) and unpressurized spares (COS 4012), lumped into an unspecified category (COS 401), or a combination of both.

The basic formulas for deriving spares demands is an extension of the linear demand model generating demands for generic resources. Given rates r_i representing unspecified, pressurized, and unpressurized rates for class of supply $i = 401, 4011, \text{ and } 4012$ and the element mass $m_{element}$, the generic resource demands $d_{i,generic}$ aggregated after Δt days of simulation time are shown in Eq. (3).

$$d_{generic,i}(\Delta t) = r_i \cdot m_{element} \cdot \frac{\Delta t}{365.25} \quad (3)$$

If an element has a parts list defined, the sparing by mass model can optionally use non-generic resource types in proportion to each part's mass fraction of the overall element, separating unspecified, pressurized

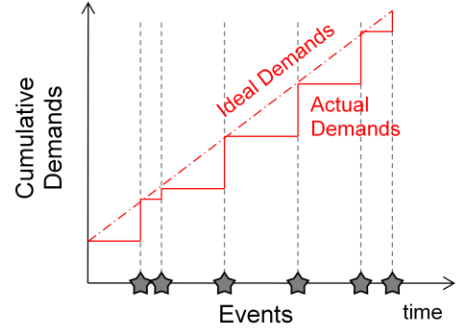


Figure 12: Discrete event time effect on linear demands. Discrete time simulation aggregates linear demands at each event execution.

and unpressurized demands. Figure 13 illustrates a parts list and corresponding resource type breakdown for a notional 100 kilogram surface rover with three specified parts totaling 50 kilograms.

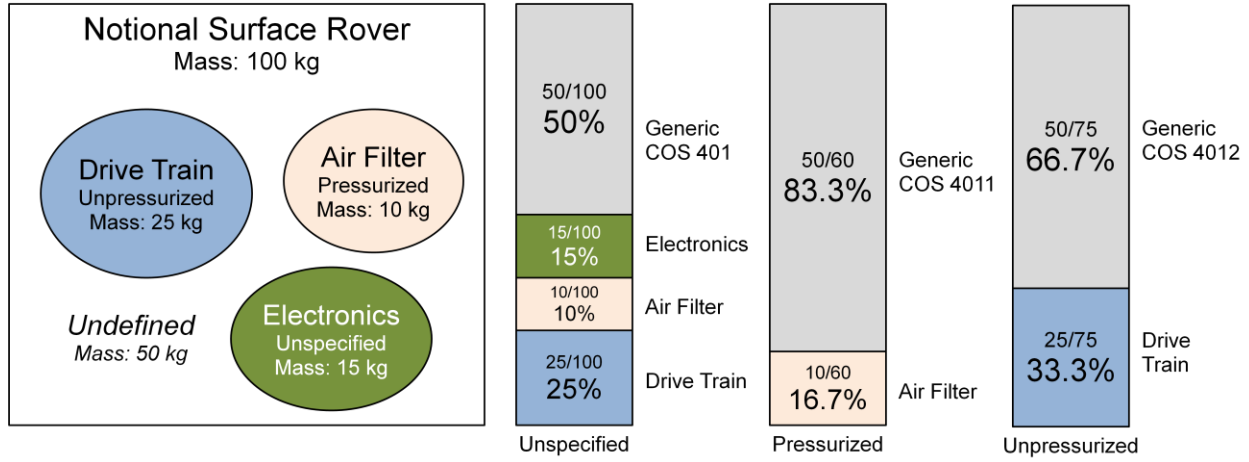


Figure 13: Sparing by mass using a parts list. The sparing by mass demand model optionally can access an element's part list to generate unspecified, pressurized, and unpressurized spares demands.

There are two fractions that drive the spares distribution when using parts lists. First, the element class of supply fraction $f_{cos,i}$ identifies the portion of an element comprised of class of supply i . For all elements, $f_{cos,401} = 1$, and both $f_{cos,4011}$ and $f_{cos,4012}$ are calculated by finding the fraction of element mass not accounted for by parts of other classes of supply. In the example in Figure 13, $f_{cos,4011} = 0.6$ and $f_{cos,4012} = 0.75$. Second, the part fraction $f_{part,j}$ identifies the portion of an element comprised of part j . In the example in Figure 13, $f_{part,1} = 0.25$, and $f_{part,2} = 0.10$ and $f_{part,3} = 0.15$ for the drive train, air filter, and electronics respectively.

Using the two fractions, Eq. (4) shows the modified functions to calculate demands for each part type j . All part types are considered instances of class of supply 401, but only pressurized parts are instances of class of supply 4011, and only unpressurized parts are instances of class of supply 4012.

$$d_{part,j}(\Delta t) = \sum_i r_i \cdot m_{element} \cdot f_{cos,i} \cdot f_{part,j} \cdot X_{i,j} \cdot \frac{\Delta t}{365.25} \quad (4)$$

where $X_{i,j} = \begin{cases} 1 & \text{if part } j \text{ is an instance of class of supply } i \\ 0 & \text{otherwise} \end{cases}$

Finally, Eq. (5) shows the adjusted generic resource demands after accounting for parts demands. If no parts list is defined, Eq. (5) reduces to Eq. (3).

$$d_{generic,i}(\Delta t) = r_i \cdot m_{element} \cdot f_{cos,i} \left(1 - \sum_j f_{part,j} \right) \cdot \frac{\Delta t}{365.25} \quad (5)$$

Table 12 lists the breakdown of annual demands for the notional surface rover example by class of supply and part under spares demands rates of 5% unspecified, 10% pressurized, and 5% unpressurized. As most parts will represent discrete resources, fractional units may be aggregated using an item discretization policy and aggregation settings, handled during simulation (see Section 3.3.2).

Table 12: Annual demands using a sparing by mass model with a parts list.

Category \ Resource [kg]	Generic COS 401	Generic COS 4011	Generic COS 4012	Electronics	Air Filter	Drive Train	Total
Unspecified Demands (5% of element mass / year)	2.50	0	0	0.75	0.50	1.25	5.00
Pressurized Demands (10% of element mass / year)	0	8.33	0	0	1.67	0	10.00
Unpressurized Demands (5% of element mass / year)	0	0	3.33	0	0	1.67	5.00
Total	2.50	8.33	3.33	0.75	2.17	2.92	20.00

3 Campaign Modeling and Analysis

A space exploration model is comprised of three components: a network, missions, and a cargo manifest. The network describes the nodes and edges that are accessible during the exploration. The mission sequence defines the sequence of events to create and move elements within the network. The cargo manifest accounts for demands occurring during the mission sequence by manifesting resource containers into carriers. Figure 14 illustrates the process used to model and analyze a campaign, including iteration loops in the manifest definition to allow logistics strategies such as pre-positioning.

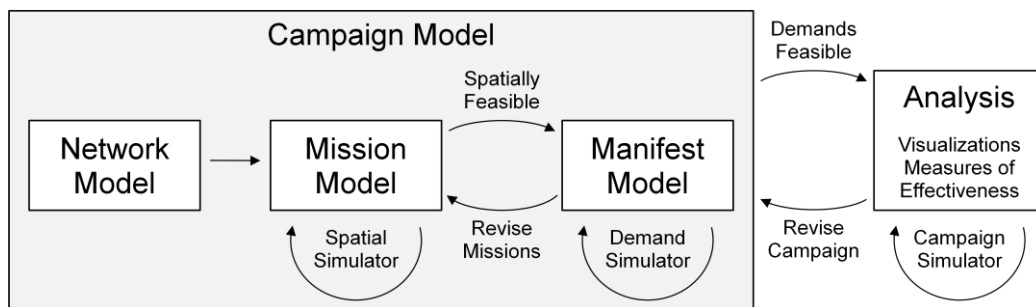


Figure 14: Campaign modeling and analysis. Campaigns are modeled in three parts: a network, missions, and a manifest. When the campaign is both spatially feasible and logistically feasible, analysis can provide visualizations and measures of effectiveness.

3.1 Network Model

The network model uses a set of nodes and edges to define the space over which the campaign will operate. Any combination of nodes and edges from any available subclasses are allowed provided every edge has a defined origin and destination node. The network is referred to as a *time-expanded network*, as all locations are expressed in time-invariant forms. A time-expanded network is visualized in a plot of time versus location, shown in Figure 15. This plot is referred to as a bat chart when showing elements moving through the network due to similarities in appearance to bats perching on a ceiling.

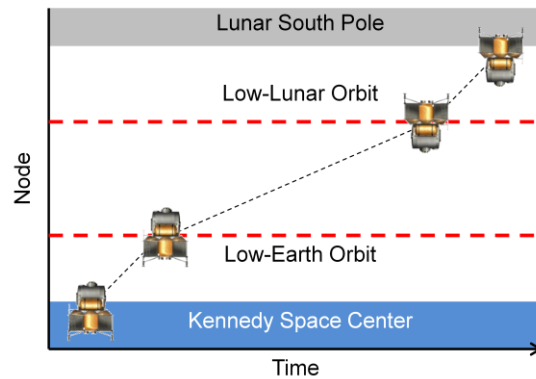


Figure 15: Network bat chart. A bat chart shows a time-expanded network and element movement over time.

3.2 Mission Model

Missions make up the majority of a particular space exploration campaign definition. Each mission contains a set of optional demand models to generate aggregated demands and a list of events to drive simulation. Missions can be checked for spatial correctness using a spatial simulator.

3.2.1 Mission Demand Models

Mission demand models generate aggregated demands from a global perspective, as opposed to element demand models which generate demands for individual elements (See Section 2.3.4). They are the traditional choice for campaign analysis and are typically parameterized by a large number of input variables. The consumables model used in SpaceNet 1.4 used parameters similar to those in Table 13 as inputs to equations for demand estimation. Many of the mission and environment parameters can be inferred by inspecting existing elements and planned mission events in a mission model, however some including the ECLSS parameters are outside the existing model scope and would be required as inputs.

Table 13: Crew consumables mission demand model parameters.

Mission Parameters	Environment Parameters	ECLSS Parameters
Crew Size	Habitat Volume	Waste Water Recovery Rate
Surface Duration	Habitat Air Pressure	Solid Water Recovery Rate
Non-crewed Surface Duration	Habitat Leak Rate	Brine Recycling (y/n)
Transit Duration	Airlock Volume	Brine Recycling Rate
Supply Reserves Duration	Airlock Efficiency	Sabatier Reaction (y/n)
Number In-transit EVAs		Electrolysis (y/n)
Number Surface EVAs		Methane Reformer (y/n)
Crew per EVA		Recover EVA CO ₂ (y/n)
Average EVA Duration		Launder Clothes (y/n)
		ISRU O ₂ Production Rate

In practice, mission demand models work well for aggregating mission impacts on demands but struggle to account for element-related impacts relying on parameters describing each element type in detail. Most campaign analysis will use a combination of both mission and element demand models to represent aggregated and individual demands.

3.2.2 Mission Events

Each mission defines a list of events to control the instantiation and movement of elements to accomplish exploration operations. There are a wide variety of events to represent the various activities that occur during a space exploration campaign, however there are a set of seven core events from which all events

can be derived. Four events, shown in Table 14, relate to the lifecycles of elements and three events, shown in Table 15, relate to the usage of resources. Composite events, shown in Table 16, automate the use of one or more core events to perform a complex event with minimal input.

Table 14: Element-based core events list.

Event Name	Description	Spatial Error Conditions
Instantiate Elements	Instantiates elements at a location or nested inside a carrier.	The target carrier has not been instantiated or does not exist The target carrier does not have sufficient capacity An element to be instantiated already exists in the simulation
Move Elements	Instantaneously moves elements to a new location or carrier.	The target carrier has not been instantiated or does not exist The target carrier does not have sufficient capacity An element to be moved has not been instantiated or does not exist at the expected location of origin
Remove Elements	Permanently removes elements from the scope of simulation.	An element to be removed has not been instantiated or does not exist at the expected location
Reconfigure Elements	Changes elements' operational state.	An element to be reconfigured has not been instantiated or does not exist at the expected location An element to be reconfigured does not contain the target reconfiguration state An element to be reconfigured is already in an unchangeable decommissioned state

Table 15: Resource-based core events list.

Event Name	Description	Spatial Error Conditions
Add Resources	Adds resources to an existing resource container.	The resource container has not been instantiated or does not exist at the expected location The resource container has insufficient capacity
Transfer Resources	Transfers resource from an existing origin resource container to a co-located destination resource container.	The origin or destination resource container has not been instantiated or does not exist at the expected location The origin resource container has insufficient resources The destination resource container has insufficient capacity
Remove Resources	Creates a demand for resources originating from a target element.	The target element does not exist or does not exist at the expected location The resource container has insufficient resources

Each mission event is assigned an execution time relative to the start of the mission and a priority between one and five. The priority is used to assign preference to simultaneous events where dependence is required. For example, elements may be instantiated at a node at the same time a transport begins, but clearly the instantiation must occur first. The seven core events and propulsive burns are modeled as instantaneous events at a single location. EVAs and explorations are modeled as processes occurring over

a set duration at a single location. Space, surface, and flight transports are modeled as transports occurring over a set duration between an origin and destination location.

Table 16: Composite events list.

Event Name	Description	Events Used
Propulsive Burn	Performs an impulsive burn using one or more propulsive vehicles' OMS or RCS engines with staging to achieve a target delta-v. Propellant is expended according to the rocket equation.	Remove Resources Remove Elements
Space Transport	Transports a set of elements (the stack) across a space edge using a series of propulsive burns to achieve required delta-v specified by the space edge.	Move Elements Propulsive Burn
Surface Transport	Transports a surface vehicle and any nested elements across a surface edge at a constant speed and given duty cycle. The surface vehicle is reconfigured to an optional transport state before the start of the transport.	Reconfigure Elements Move Elements
Flight Transport	Transports a set of elements (the stack) up to a nested mass and crew capacity across a flight edge.	Move Elements
Extravehicular Activity (EVA)	Schedules an extravehicular activity during which crew members are reconfigured to an EVA state if specified and moved external to the habitat (carrier). After the EVA duration, crew members are returned to the habitat and reconfigured to the previous state.	Reconfigure Elements Move Elements
Exploration	Schedules a specified number of EVA events having equal duration and the same crew over an exploration period. EVAs are scheduled with equal time separation before and after all events.	EVA

Both OMS and RCS propulsive burns utilize a specified burn/stage sequence and the rocket equation to calculate demands for fuel using the assumption of impulsive burns. Eq. (6) is used to calculate the maximum delta-v achievable $\Delta v_{achievable}$ for a stack of mass m_{stack} and vehicle with fuel mass m_{fuel} and specific impulse I_{sp} . If the achievable delta-v is less than the target delta-v, all fuel is consumed, any specified elements are staged, and the next propulsive vehicle in the sequence repeats the same analysis.

$$\Delta v_{achievable} = I_{sp} \cdot 9.81 \frac{\text{m}}{\text{s}^2} \cdot \log \left(\frac{m_{stack}}{m_{stack} - m_{fuel}} \right) \quad (6)$$

Alternatively, if there is excess fuel in a particular propulsive vehicle, the remaining delta-v is decreased to zero and Eq. (7) determines the amount of fuel to consume in the burn m_{burn} .

$$m_{burn} = m_{stack} \left(1 - \exp \left(\frac{-\Delta v_{target}}{I_{sp} \cdot 9.81 \text{ m/s}^2} \right) \right) \quad (7)$$

3.2.3 Spatial Simulator

It is important to identify error conditions as soon as possible as missions are populated with events. The types of errors occurring during this phase of campaign modeling are called spatial errors, meaning the error is related to the location or placement of one or more elements in the system state. The most common spatial errors occur when elements do not exist at an expected location or a carrier element has insufficient capacity to nest a required element. Since spatial errors are only perceived at event execution time and the system (network) state before each event depends on all previous events, the entire campaign must be simulated to determine the presence of any spatial errors. A spatial simulator quickly executes the events without demand satisfaction for this purpose.

The spatial simulator uses the information contained within the campaign to build an initial network state and an event stack sorted by execution time. Each event is sequentially executed, resulting in an updated network state and new events added to the stack for composite event execution. Any spatial errors identified during execution are logged for debugging. Figure 16 illustrates the general process of simulating a campaign using a spatial simulator.

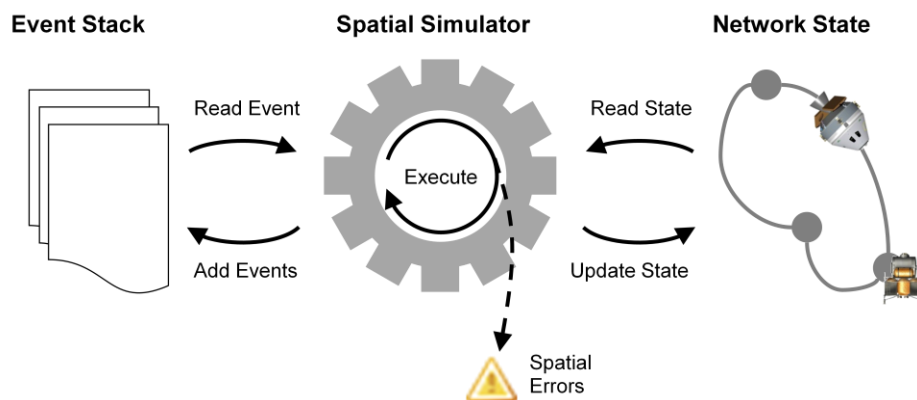


Figure 16: Spatial simulation process. *The spatial simulator sequentially executes events, updating the network state and adding events when necessary, and may generate spatial errors.*

3.3 Manifest Model

The manifest model defines a set of resource containers packed with resources and a list of cargo transfers between transport carriers to ensure resources are in place to satisfy demands during the exploration campaign. Figure 17 outlines the process used to generate a manifest, including iteration loops required to share containers and create multi-transport manifests.

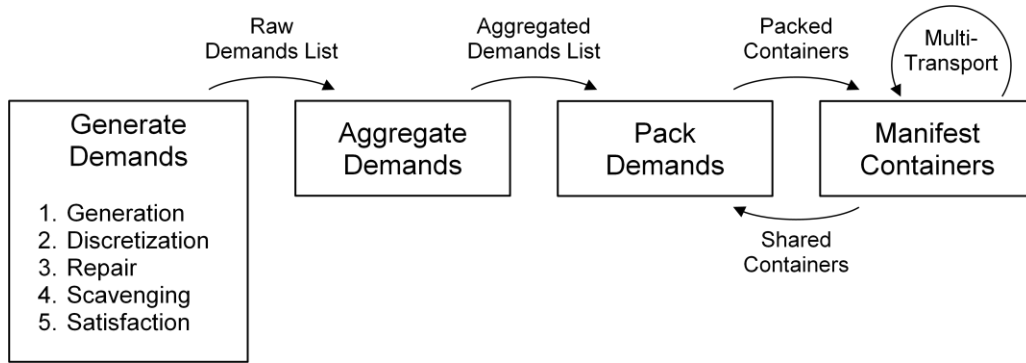


Figure 17: Manifest model creation sequence. Manifesting requires the generation of demands, aggregated to the location and time of demand. Resources are then packed into containers which subsequently are manifested onto carriers for transport.

First, demands must be generated by executing the campaign with a demand simulator which logs unsatisfied demands for resources. Second, raw demands are aggregated to simplify the delivery of resources to locations at particular times. Third, the aggregated resources are divided and packed into containers. Finally, the containers are manifested onto a sequence of carriers to bring the resources to the time and location at which it is demanded. Iteration loops exist if containers hold resources used at different times or places or if containers must be manifested onto more than one transport.

The manifesting model presented in this section only provides a representation of the manifest model within a space exploration campaign, not necessarily a method to create a manifest to meet a particular objective. The implementation of the manifesting model in SpaceNet 2.5, detailed in Section 4.3.5, provides a heuristic algorithm which attempts to find a valid, but sub-optimal, manifest. Appendix B presents a body of parallel research into the *optimal manifesting problem*, which uses a similar modeling framework to find an optimal manifest subject to a policy, strategy, or other objective function.

3.3.1 Demand Simulator

The demand simulator is a more detailed version of the spatial simulator discussed in Section 0. The main addition is a demand satisfaction sequence resulting in a list of unsatisfied demands, as shown in Figure 18. Most demands are yet unsatisfied and logged before creating the cargo manifest.

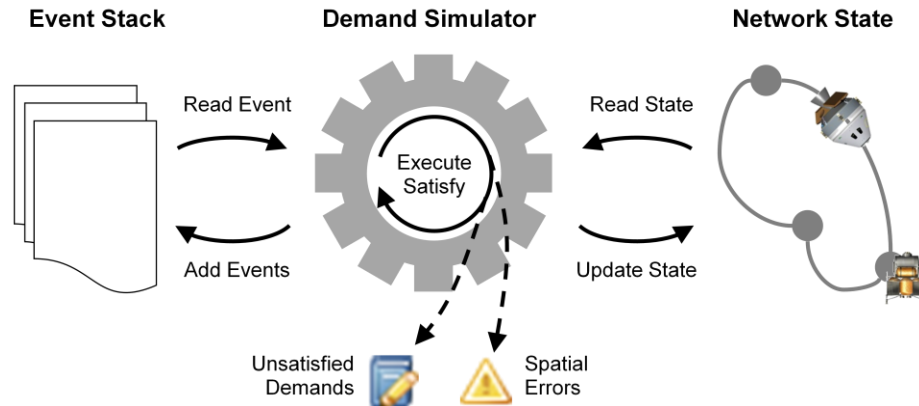


Figure 18: Demand simulation process. *The demand simulator executes events, updating the network state and adding events as necessary, may generate spatial errors, and logs unsatisfied demands for analysis and manifesting.*

A demand satisfaction sequence generates and attempts to satisfy demands during execution of the campaign events, adjusting for discretization, repair, and scavenging. The demand satisfaction sequence includes the following six steps:

1. Generate raw demands from demand models
2. Discretize demands per discretization and item aggregation policies
3. Repair demands per repair policy
4. Scavenge resources if available per scavenging policy
5. Satisfy remaining demands using co-located resources
6. Report unsatisfied demands with (optional) packing overhead mass

Demand Generation

Demands for resources originate from mission and element demand models. Since the two types of demand models operate on different levels, demand generation differs slightly. Mission demand models are queried before each mission starts for a list of resources to support the mission. Element demand models selected by the current element operational state are queried before each event execution for a list of resources demanded since the previous event execution. In some cases, there may be no elapsed time from the previous event, in which case there may be no additional demands. Co-located demands for the same resource are combined so there may be one demand for each resource from a single location.

Discretization and Item Aggregation

The time between subsequent events may often be shorter than the time required for generation of a unit for discrete resources (items). In an effort to manage the discretization process, two policies guide the

generation of whole items. The discretization policy determines how continuous demands are grouped into units and the item aggregation policy determines when the discrete units are resolved.

Figure 19 illustrates three options for the discretization policy: no discretization, by element, and by location. No discretization treats discrete resources as continuous resources and no further considerations are taken. Discretization by element aggregates discrete items separately for each element, independent of location. Discretization by location aggregates items separately for each location, allowing for spares pooling between co-located elements with common demands. The choice of discretization policy can help smooth demands for common spare parts at a single location.

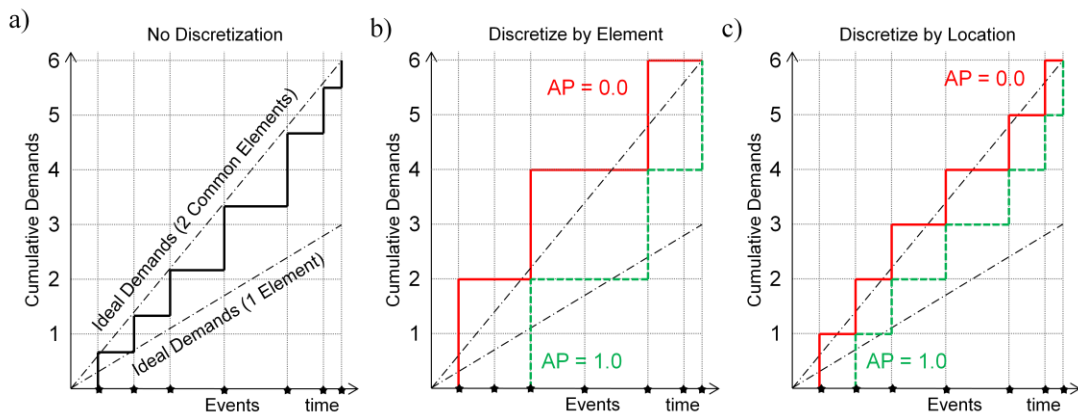


Figure 19: Impact of discretization policy. Sample demands for two elements with common discrete demands. AP is the aggregation policy value. (a) No discretization treats the demands as continuous. (b) Discretization by element aggregates items on a per-element basis. (c) Discretization by location aggregates items on a per-location basis, enabling spares pooling to locally decrease and smooth demands.

The aggregation policy, ranging from 0 to 1, sets the point at which continuous quantities are aggregated into items. The policy value determines the amount of a discrete resource to be generated before a whole unit demand is aggregated. Figure 20 illustrates several aggregation policy values. A policy level of 0.0 aggregates items at the first partial demand (equivalent to the ceiling function) while a policy level of 1.0 aggregates items after unitary demands are accumulated (equivalent to the floor function). Most analyses use an aggregation policy of 0.5 to balance the over-estimating and under-estimating behaviors.

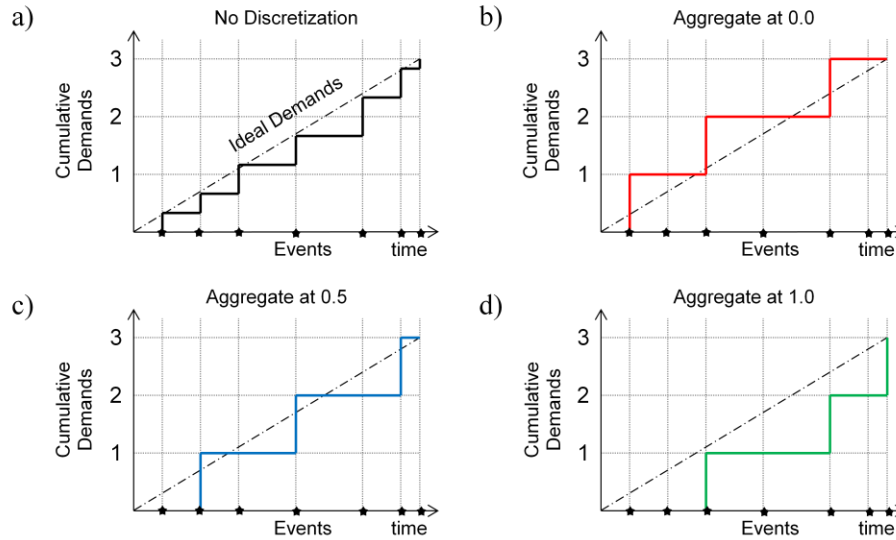


Figure 20: Impact of aggregation policy. The item aggregation policy determines at which point demands are aggregated into whole units. (a) No discretization treats demands as continuous. (b) A policy of 0.0 aggregates items at the first demand. (c) A policy of 0.5 aggregates items when half units are accumulated. (d) A policy of 1.0 aggregates items after unit demands.

Repair

Repair activities provide a trade between crew repair time and spare parts mass. Repair activities are dictated by a repair list of items targeted for repair during each crewed mission (for implementation details, see Section 4.3.4). During the repair cycle, the demand for a repaired item is replaced by any derived resources required to perform the repair.

Scavenging

If enabled, resources can be scavenged from co-located elements previously having entered a decommissioned state (see Section 2.3.3). The scavenging process reduces the amount of resources demanded by the amount of available parts contained in co-located decommissioned elements until there are no remaining resources to scavenge.

Demand Satisfaction

Any remaining demands after discretization, repair, and scavenging attempt to be satisfied by existing resources contained in elements co-located at the node or edge at which the demand was generated. If a demand originates from an element, it will be recursively inspected for any available resources including nested elements before all other co-located elements are recursively inspected for available resources. All unsatisfied demands are logged and supplied to the demand aggregation phase.

3.3.2 Demand Aggregation

Demands are aggregated into groups for further processing after generation by the demand simulator. The goal of demand aggregation is to group demands having similar spatial and temporal characteristics as to how they may be satisfied via manifesting. This is accomplished with the introduction of exploration periods, defined as the time serviced by a particular transport, i.e. the interval between subsequent arrivals of transports or the end of a campaign at a specific node. Figure 21 demonstrates the exploration period definition for a sample campaign of eight transports.

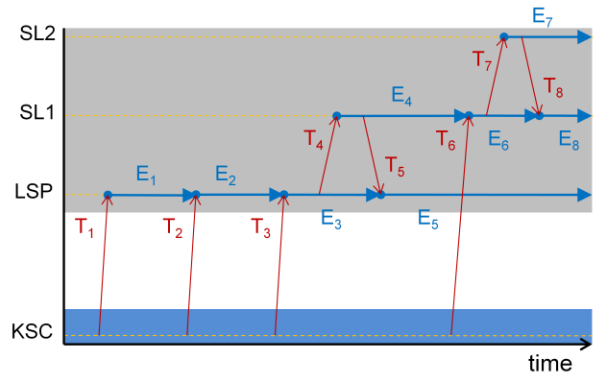


Figure 21: Campaign transports and exploration periods. Each transportation arrival marks the start of its corresponding exploration period.

Every demand in a campaign can be aggregated to either a transport (demands at edges) or an exploration period (demands at nodes). Although all demands within a transport must be self-supplied, demands within an exploration period can be supplied by any transport arriving at the same node at or before the start of the period. By definition, there are no transport arrivals in the middle of an exploration period, providing a common requirement for all inter-period demands and greatly simplifying the problem. Using these rules, resource containers can be created and packed with resources and manifested onto transports.

3.3.3 Demand Packing

The process of demand packing assigns quantities of resources to individual resource containers. Figure 22 shows the two-step process to packing. First, aggregated demands are divided into groups corresponding to packing separation. Second, resource containers are packed with cargo from one or more packed demand groups according to mass, volume, and environmental constraints and delivery constraints derived from other resources sharing the container. The result of the demand packing process is the dual map from aggregated demands to demands as packed to resource containers.

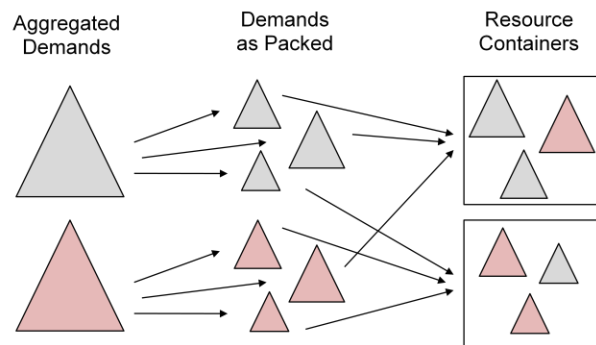


Figure 22: Demand packing. Aggregated demands are split into groups corresponding to how the demands are packed within individual resource containers.

3.3.4 Container Manifesting

The process of manifesting assigns resource containers to carriers. Manifests include a sequence of transports completing a reverse transversal from the demand location to a source node. As containers are manifested on carriers, they are subsequently demanded at the exploration period preceding the transport, requiring additional manifest steps to reach a source node. Figure 23 shows example multi-transport manifesting options for a demand from exploration period three.

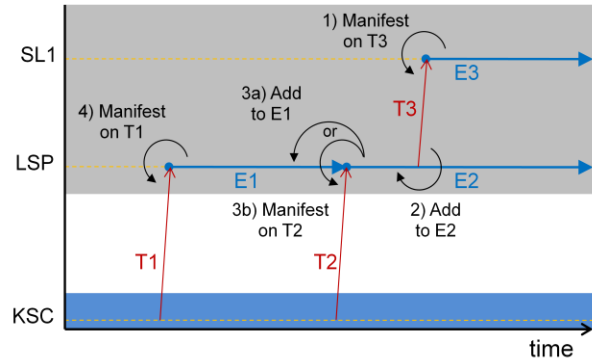


Figure 23: Multi-transport manifesting. Containers manifested on transport 3 (T3) are demanded during exploration period 2 (E2) and must be supplied by either T1 or T2.

3.4 Campaign Analysis

After the manifest has been created, the campaign is ready for the analysis phase. At this point, all demands should be satisfied by the manifested resource containers. The campaign simulator creates visualizations and computes various measures of effectiveness.

3.4.1 Campaign Simulator

The campaign simulator is an extension of the demand simulator. The completed cargo manifest is processed into scheduled events to create, fill, and transfer resource containers. Additional data logs are maintained to record the system state after each event execution and tabulate activities such as scavenging and repair. Finally, a set of measures of effectiveness are maintained to evaluate the entire campaign.

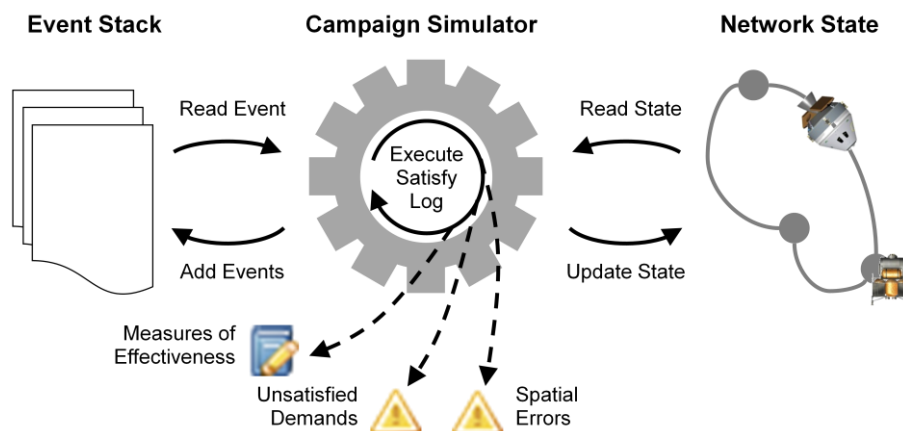


Figure 24: Campaign simulation process. The campaign simulator executes events, updating the network (system) state and adding events as necessary, and logs measures of effectiveness.

Any spatial errors or unsatisfied demands occurring during the simulation are noted for corrective action, though the simulation will still proceed. This most commonly results from a manifesting action affecting a mission event, e.g. the addition of extra cargo mass causes a propulsive infeasibility.

3.4.2 Measures of Effectiveness

The NASA Systems Engineering Handbook identifies measures of effectiveness (MOEs) as “... the ‘operational’ measures of success that are closely related to the achievement of mission or operational objectives in the intended operational environment” [28]. A summary of the MOEs used in this campaign analysis is provided in Table 17. Detailed MOE formulations are described in previous publications [10].

Many MOEs are carried over from past work, though the calculation procedure is changed from post-processing to live-logging. In a live-logging environment, a log entry is created for contributions to each of the MOEs before each event is executed in the campaign simulator. For example, if a transport event moving elements from Kennedy Space Center into low Earth orbit is to be executed, its contributions to the total launch mass and up-mass capacity utilization are logged. The resulting logs can be used to recreate the evolution of the MOEs over the course of a campaign.

Table 17: Campaign measures of effectiveness.

Measure of Effectiveness	Units	Description
Crew Surface Days	crew-days	The total number of crew-days over all non-Earth surface nodes.
Crew Corrective Maintenance Time	crew-hours	The total number of crew-hours spent on corrective maintenance (repair) activities.
Exploration Mass Delivered	kg	The total mass of exploration items (COS 6) and surface infrastructure (COS 8) delivered to all non-Earth surface nodes.
Total Launch Mass	kg	The total mass transported from Earth’s surface.
Up-mass Capacity Utilization	-	Fraction of available mass capacity utilized for cargo on transports from Earth’s surface.
Down-mass Capacity Utilization	-	Fraction of available mass capacity utilized for cargo on transports to Earth’s surface.
Exploration Capability	kg-crew-days	The dot product of crew surface days and exploration mass, resulting in a measure of the total capability for crew to perform exploration.
Relative Exploration Capability	-	The amount of productive exploration per kilogram of mass launched as compared to that of Apollo 17.

4 SpaceNet 2.5 Implementation

SpaceNet 2.5 is a Java program implementing the models discussed in Chapters 2 and 3. It was developed using user-centric design philosophies to improve its usability and efficiency. The SpaceNet graphical user interface (GUI) allows the user to build, edit, and analyze exploration campaigns without detailed knowledge of the underlying models. SpaceNet also provides visualizations and feedback to simplify the campaign creation process and quickly identify and reduce the number of simulation errors.



Figure 25: SpaceNet logo.

4.1 Development and Release

Development on SpaceNet 2.5 started in July 2008 using user-centric methodologies to iterate on concepts and design prototypes. MIT hosted a SpaceNet workshop on December 4-5 2008 which established core concepts including the element hierarchy, element operational states, and element demand models. Through January 2009, development focused on maturing concepts with prototypical user interfaces and sample simulation implementations.

From January through June 2009, the focus of development turned to design and implementation of the graphical user interface and maturing the campaign simulation to allow manifesting of demands. A review was hosted at JPL on March 24 2009 at which time a prototype was demonstrated including demand simulation. A NASA-specific version of SpaceNet 2.5 was delivered on June 30 2009 along with documentation including a User's Guide and Quick Start Tutorial. Between July and September 2009, non-publicly available components were removed from the public branch of SpaceNet, resulting in the public release on October 1, 2009.

Development after October 2009 focused on usability improvements and expanded support for data management including a data editor. SpaceNet 2.5 R2 is scheduled for release in late 2010.

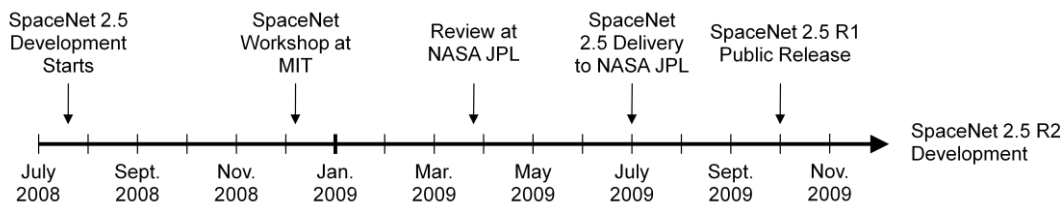


Figure 26: SpaceNet 2.5 development timeline. SpaceNet 2.5 R1 was released after initial delivery to NASA JPL. SpaceNet 2.5 R2 is currently under development.

4.1.1 Development Methodology

Improve its usability and efficiency was one of the early development goals for SpaceNet. The high-level goal of usability, however, is coupled with the low-level goal of modeling, so several prototype iterations of increasing fidelity were created and presented to representative users during the development cycle.

The first prototypes included sketches and screen mockups to display the process of building and analyzing a campaign. Figure 27 shows an example mockup from an early design cycle. Later prototypes used spreadsheets and forms to provide feedback and visualizations during the campaign design process.

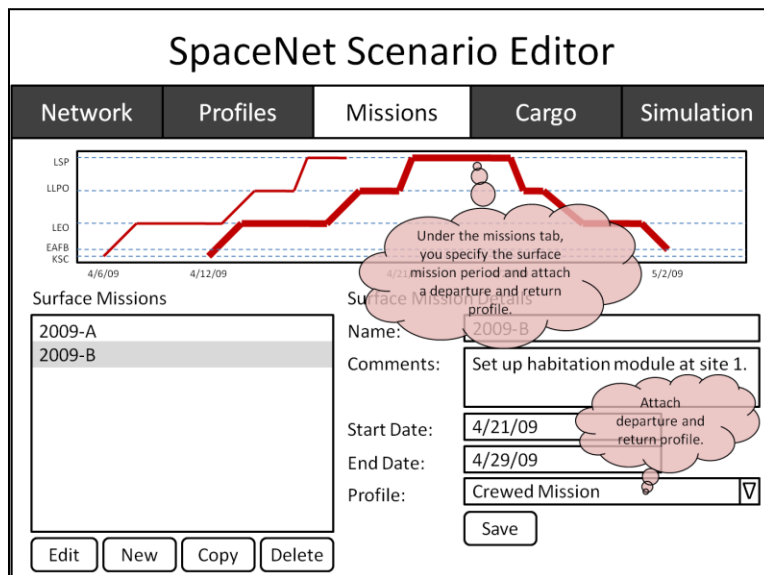


Figure 27: Early SpaceNet GUI prototype mockup. *Prototype mockups assist development iterations by walking through the campaign definition without time-intensive coding investment.*

4.1.2 Development Tools

A SpaceNet server running LAMP (Linux, Apache, MySQL, PHP) provided development tools to enable collaboration between team members. First, the server hosted a wiki using the open-source DokuWiki software [29], providing a central point of communication between all members of the development team. In addition to meeting notes and documentation, the SpaceNet wiki also provided a bug reporting and change request page summarizing and sorting feedback by level of priority (bug, high, medium, or low). The wiki logged over 100 requests throughout development of SpaceNet 2.5 R1.

Establishing a subversion repository on the server was another crucial component for development. Subversion (SVN) provides version control over a central code base [30]. In addition to the ability to roll back to previous versions, developers receive updates from others in a collaborative fashion. The subversion system logged over 1000 modifications by the release of SpaceNet 2.5 R1.

Finally, developers used the Eclipse Integrated Development Environment (IDE) for writing and debugging code [31]. Eclipse provides continuous-compilation of a project to identify errors and refactoring support for automating changes. Extensions to the Eclipse platform also provide direct integration with the SVN server.

4.1.3 Public Release

SpaceNet 2.5 R1 was released under an open source GNU General Public License (GPL) version 3 on October 1 2009 [32]. This type of license is referred to as a copyleft license, as its terms protect the content of the copyright holder by ensuring free access to the source code on the condition that any derivative works must also be distributed under a similar license. Under this philosophy, a steady-state market price of zero is achieved for the source code, though services and support may be provided for the original or derivative works for a fee.

A project website was created on the SpaceNet server to organize information about the project and provide a download point for potential users.* An online community was also established to provide a point of contact between users and developers.

4.1.4 Documentation

Documentation included with distributions of SpaceNet 2.5 R1 includes the *SpaceNet User's Guide*, *SpaceNet Quick Start Tutorial*, and Javadoc comments. Much of the technical details from the documentation has not been repeated in this thesis due to its dynamic nature with additional development and provided accessibility via the SpaceNet project website.

The *SpaceNet User's Guide* is targeted as a reference document rather than a tutorial. It provides an overview of the project motivation and implementation as well as information on every component of the SpaceNet GUI. Two appendices aid the understanding of the underlying model. The first defines the abstract object classes used in the modeling framework. The second details the data required by the data source to populate object definitions.

The *SpaceNet Quick Start Tutorial* walks through two step-by-step tutorials highlighting the analysis techniques and campaign-building methods. The first tutorial creates a lunar sortie mission to the Lunar South Pole focused on introducing the basics of mission events and establishing propulsive feasibility. The second tutorial uses abstracted flight transportations to create a multi-mission campaign to the Lunar

* The project website is available at <http://spacenet.mit.edu>

South Pole focused on introducing demand models and manifesting. All required data source files as well as completed tutorial campaign files are included with the distribution.

Javadoc files provide documentation of the detailed software implementation called an application programming interface (API) [33]. Javadoc files are automatically generated from structured comments placed within the source files and would typically only be used by developers. The source distribution of SpaceNet 2.5 includes full Javadoc files detailing the model and GUI implementation.

4.2 Data Sources

Campaign analysis within SpaceNet requires many objects definitions in addition to the specific campaign and mission architectures. Redefining objects for each campaign, however, is time-consuming. In particular, the objects that comprise the core of the modeling framework (nodes, edges, resources, and elements) seldom change between campaigns, prompting the usage of independent data sources that may be used between several campaigns.

SpaceNet 1.3 and 1.4 store object definitions in spreadsheets organized into several tabs, one per object class. As some objects require references to other objects (e.g. origin and destination nodes for each edge), identification numbers, or keys, are used to create data relationships between tables consistent with relational database theory. Figure 28 highlights the numerous relationships used within the interconnected SpaceNet modeling framework.

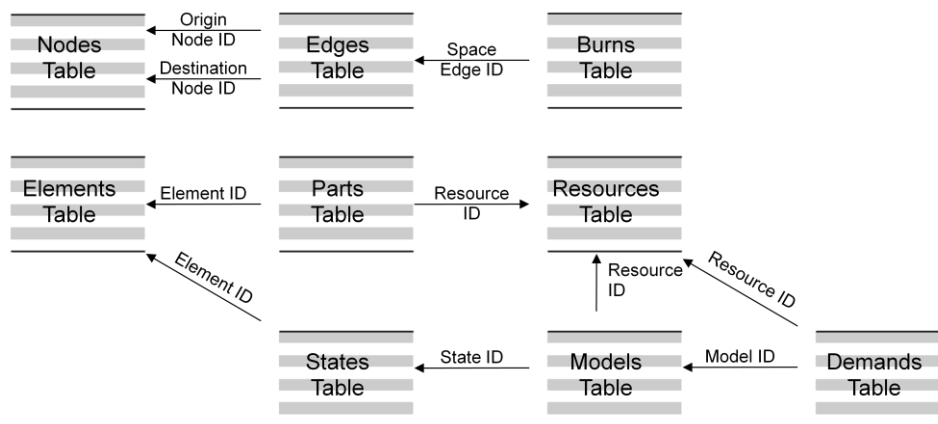


Figure 28: Data table relationships. Data is divided across several tables to represent parent-child relationships between objects.

Development in SpaceNet 2.0 investigated using relational databases as a replacement for spreadsheets. The same table structure was maintained, but validation rules were added to maintain data integrity. Database software also allows commands written in generic structured query language (SQL) and

performs operations faster than reading and writing to files. After preliminary user testing, however, it was discovered that spreadsheets provided an easy, familiar interface to the data. For this reason, a generic interface to data sources was created that does not depend on the underlying implementation. To date, data sources have been implemented for both spreadsheets and relational databases.

The movement to an abstracted data source interface also enabled a generic data editor which provides a graphical user interface for any data source.* This provides users with an easier way to edit and view the data source that enforces validation and formatting rules. In addition to the data editor, element sizing models have been introduced to aid the design of new elements for use in campaigns.

4.2.1 Data Editor

Editing data directly within a raw format such as a spreadsheet can be time-consuming and error-prone. Spreadsheet data often loses integrity due to incomplete entries as validation rules are not enforceable. Primary key-foreign key relationships between tables create a cascade of nested objects in both spreadsheets and relational databases. Capitalizing on the data source abstraction, a data editor was introduced to manage data regardless of its source implementation.

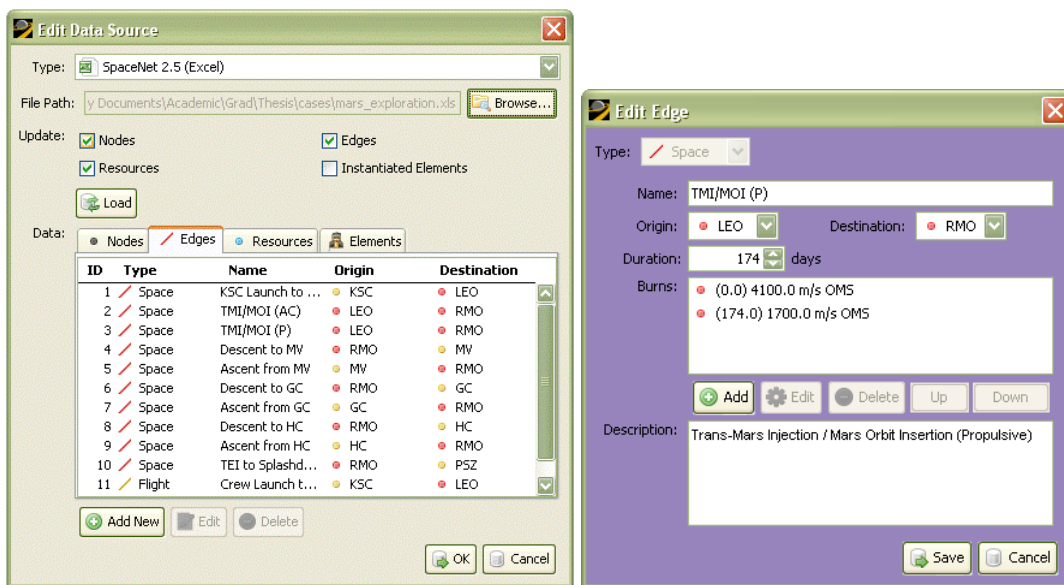


Figure 29: SpaceNet data editor user interface. The data editor provides a GUI for any data source implementation. This example edits an edge from a spreadsheet database.

The data editor is integrated with SpaceNet to provide a graphical user interface to data sources. It provides support for viewing, editing, deleting, and creating new entries for nodes, edges, resources, and

* Collaborator Ivo Ferreira led development and implementation of the data editor and element sizing tools.

elements. It helps ease the process of editing data, especially for inexperienced users who may not be familiar with the formatting or organization of underlying data sources.

4.2.2 Element Sizing Tool

Of the domain objects in the SpaceNet modeling framework, nodes, edges, and resources seldom change between campaigns but elements are susceptible to modification as designs mature or trades are considered. The ability to generate new element designs using a sizing model was introduced as a module of the data editor.

An element sizing model provides an alternative process to generate element specifications. For example, a crew member sizing model may take height as an input to estimate the element mass. Similarly, a habitat sizing model may require maximum crew occupancy and enabled technologies to generate the element mass and volume.

Current support exists for spreadsheet-based models using designated cell addresses for inputs and outputs. Future research and development will look into other methods for designing elements including leveraging existing databases of similar elements to infer the design of new elements.

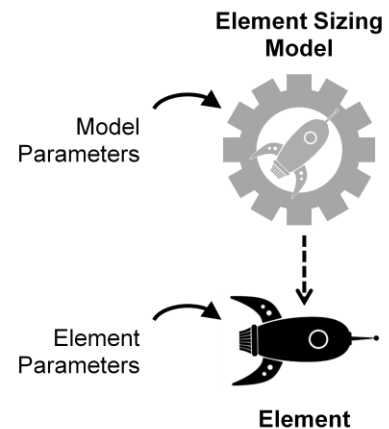


Figure 30: Element sizing model. An element sizing model creates element designs using customized model input parameters.

4.3 Graphical User Interface

The graphical user interface (GUI) is the portion of SpaceNet which users interact with to create and analyze campaigns. Figure 31 highlights some features of the SpaceNet GUI. The campaign panel and five tabs guide users through the campaign definition and analysis process, broken down into six steps.

1. Set high-level campaign parameters (*Campaign Panel*)
2. Select network nodes and edges (*Network Tab*)
3. Define exploration missions (*Missions Tab*)
 - a. Set high-level mission parameters
 - b. Define mission events
4. Analyze demands generated for resources (*Demands Tab*)
5. Create campaign cargo manifest (*Manifest Tab*)
6. Simulate and analyze campaign (*Simulation Tab*)



Figure 31: SpaceNet 2.5 user interface. *The SpaceNet GUI guides users through the process of modeling and analyzing exploration campaigns.*

4.3.1 Campaign Panel

Positioned at the top of the SpaceNet GUI, the campaign panel modifies campaign-wide options. Inputs include a name to reference the campaign, a starting date (epoch) from which to measure simulation time, a reference to the user or group creating the campaign, and a short description. Two additional dialog boxes access additional options. The ‘campaign options’ dialog sets global parameters such as precision values, constraint enforcement, demand policies, and simulation options detailed in Table 18. The ‘data source’ dialog launches the data editor to select the data source and manage the loading of data.

Table 18: Campaign options list.

Option Description	Values
Time Precision	[0.001,0.500] days
Demand Precision	[0.001,0.500] units
Mass Precision	[0.001,0.500] kg
Volume Precision	[0.1,100] cm ³
Enforce Volume Constraints	Yes/No
Item Discretization Policy	None/By Element/By Location
Item Aggregation Point	First/Half/Unit Demand {0.0, 0.5, 1.0}
Enable Scavenging	Yes/No

4.3.2 Network Tab

The network tab serves as the interface to the network model. A network visualization displays nodes and edges from a data source. Nodes and edges are colored based on type: surface nodes are yellow, orbital nodes red, and Lagrange nodes purple, surface edges green, space edges red, and flight edges yellow.

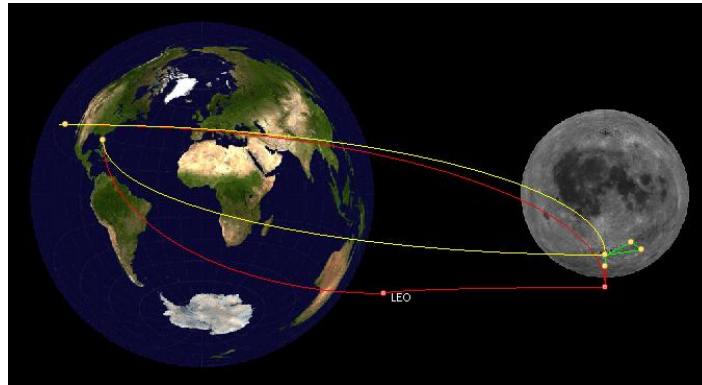


Figure 32: SpaceNet network visualization. Network for a lunar campaign including surface (yellow) and orbital (red) nodes, and surface (green), space (red), and flight (yellow) edges.

Surface nodes are mapped onto a Lambert-Azimuthal projection of planetary bodies. Orbital nodes are mapped based on altitude and inclination. Lagrange nodes are mapped based on position. The size and relative positioning of celestial bodies and the location of orbital and Lagrange nodes are scaled using an arctangent function to set a maximum and characteristic distance. Edges do not correspond to actual physical paths, but rather connect nodes with graphical arcs.

Several pre-set scenario types can be used to filter the selection of locations:

- **ISS:** Selects Earth nodes and all connected edges
- **Lunar:** Selects Earth and Moon nodes and all connected edges
- **Moon-only:** Selects Moon nodes and all connected edges
- **Martian:** Selects Earth, Moon, and Mars nodes and all connected edges
- **Mars-only:** Selects Mars nodes and all connected edges
- **Solar System:** Selects all available nodes and edges

4.3.3 Missions Tab

The missions tab defines the sequence of missions and provides top-level visualizations for the campaign shown in Figure 33. The process bat chart illustrates mission event sequences and the element bat chart illustrates the movement of individual elements through the time-expanded network.

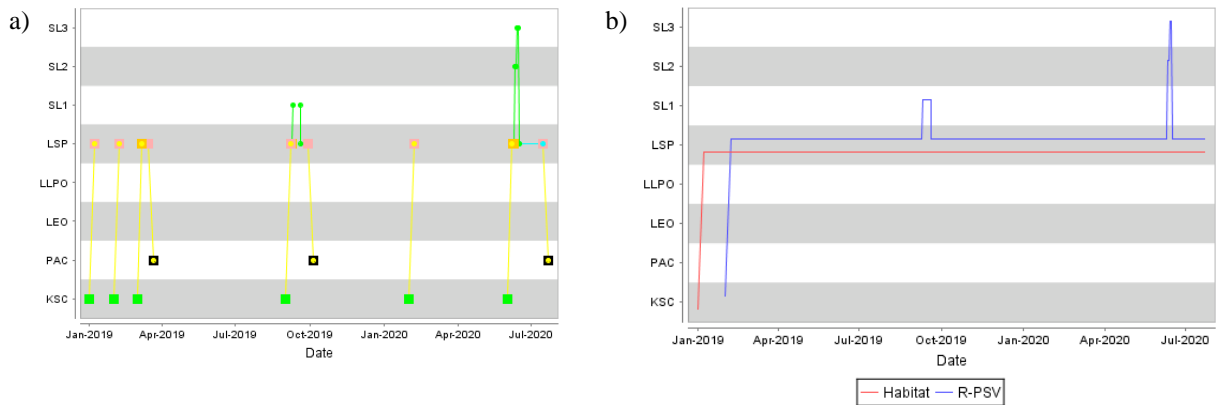


Figure 33: Campaign visualizations. (a) The process bar chart displays mission events, processes, and transports in time-expanded network. (b) The element bar chart displays individual element movement in time-expanded network.

Each mission supplies several inputs including a name, a starting date (epoch), and origin and destination nodes from which to calculate net transport and exploration durations. Also, mission demand models can be attached to calculate aggregated mission demands (see Section 3.2.1), and mission events are defined to drive the simulation (see Section 3.2.2).

Spatial simulation is continuously executed as mission events are created to determine the system state at the time of each new event, identify available elements and quickly alerting of any error conditions. Figure 34 illustrates error messages displayed in the mission event list and in the event dialog.

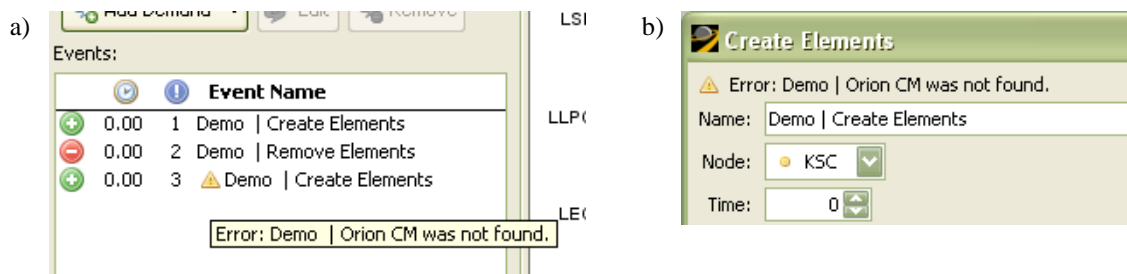


Figure 34: Spatial simulation errors. Spatial simulation errors display (a) in the mission events list and (b) in the mission event dialog to alert the user of error conditions.

4.3.4 Demands Tab

The demands tab does not correspond to a specific component of the modeling framework, but rather provides visualizations and analysis capabilities for the demands generated during a campaign. Options exist to set demand-related campaign parameters, including discretization, aggregation, and sparing policies. Other options including enabling or disabling estimates for container masses and consumption of

existing resources only change the resulting demand visualizations. There are a series of five visualizations and analysis tabs.

Scenario Feasibility Visualization

The scenario feasibility visualization plots the cumulative capacity of all source transports (transports originating from a source node) and estimates of cumulative demands for resources over the entire campaign. It is a necessary but not sufficient condition that the capacity exceeds the demand, as local infeasibilities may still exist at specific nodes in time. In practice, it serves as a first check for identifying infeasible campaigns, however, if the campaign contains only a single destination the scenario feasibility visualization is a sufficient indication of feasibility. Figure 35 illustrates scenario feasibility visualization for an example multi-destination campaign.

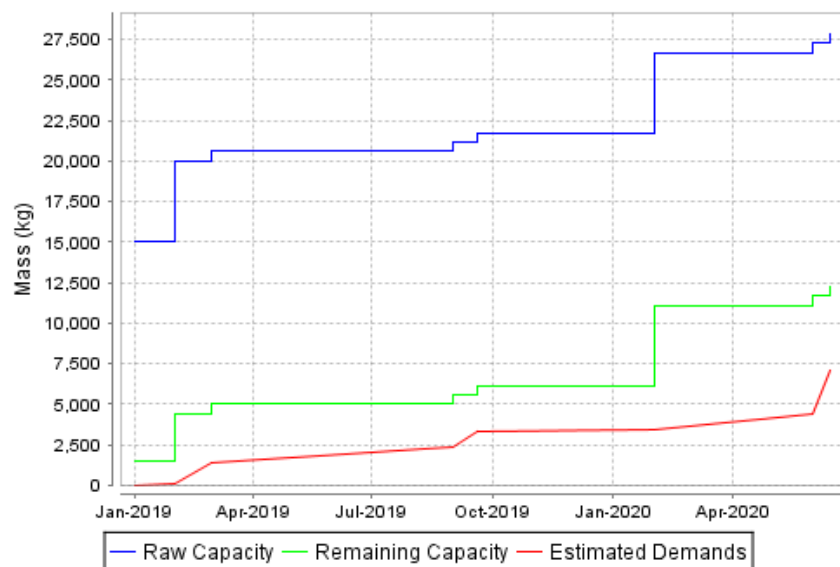


Figure 35: Scenario feasibility visualization. Cumulative raw and remaining transport capacity must always exceed estimated cumulative demands for feasible campaigns.

Supply Network Visualization

The supply network visualization displays transports and aggregated demands in a time-expanded network and is often a precursor to the manifesting process. Line width represents transport capacity, transport demands, or net transport capacity (the difference between the two). Circle diameter represents the mass of aggregated exploration period demands.

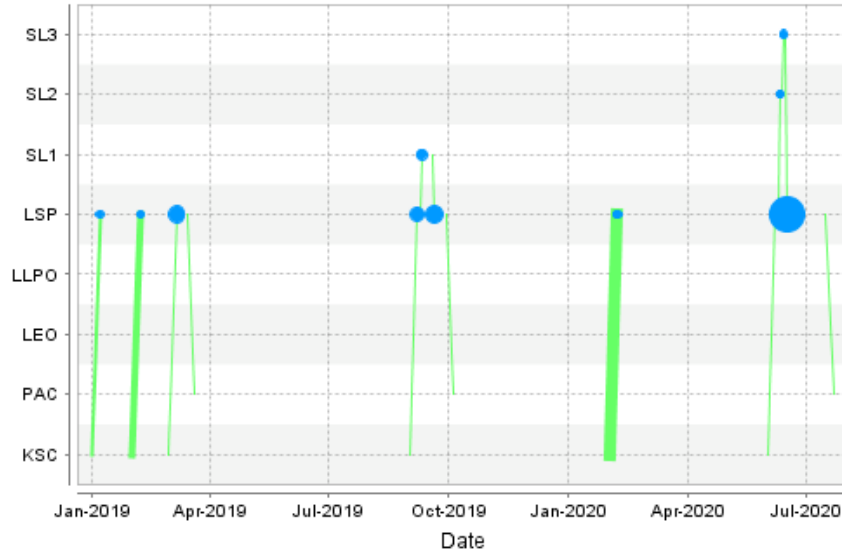


Figure 36: Supply network visualization. Line width represents net transport capacity and circle diameter represents aggregated exploration period demand mass.

Demand History Visualizations

In addition to file-format export options, charts filtered by elements, locations, or missions visualize raw demands. Demand history by element provides a time-history of the demands generated by a single element. In some cases, such as mission demand models, demands for resources are not generated by a particular element. Demand history by location provides a time-history of demands generated at a node or edge. Finally, demand history by mission provides a time-history of the demands generated by any element or location during a mission. In all three cases, demands are grouped by base class of supply.

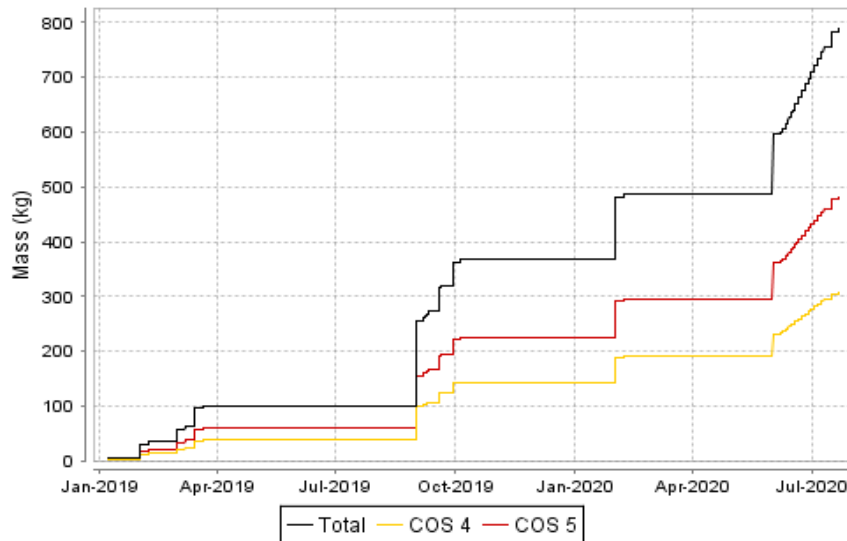


Figure 37: Element demand history visualization. Plots demands for spares (COS 4) and packing mass estimates (COS 5) generated by a habitat element. The non-linear response results from reconfigurations of element state during crewed and un-crewed periods.

Commonality Analysis

The commonality analysis tab provides details of campaign part commonality and insights to how scavenging of parts impacts demands generation. If enabled, scavenging events are logged by the demand simulator and displayed with filters by the source of scavenged parts and the type of parts scavenged.

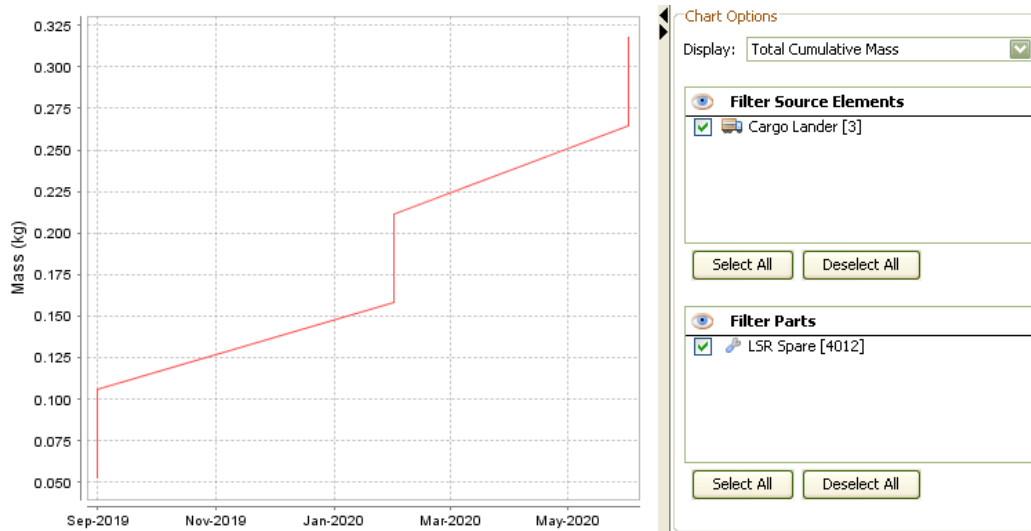


Figure 38: Commonality and scavenging analysis. Displays scavenged common parts filtered by source element and type of resource.

Repairability Analysis

The repairability analysis tab establishes a repair list for each crewed mission in an exploration campaign. Currently, repair activities are limited to crewed missions with deterministic demands; future research is required for extensions to more detailed campaigns. To efficiently use the available time, a repair policy should maximize the benefit (mass savings) for a given cost (crew time). In addition to manually-selected repair, an “Auto-repair” option chooses an optimal set of items to repair.

The repairable items demanded in a mission may be considered to be in random order. Only items demanded during a crewed mission or from non-crewed missions preceding it are considered. Each item i is associated with a mass benefit, the unit mass (M_i) less the mass to repair (MTR_i), and a time cost, the time to repair (TTR_i). If the repairable items are sorted in descending order of the benefit-cost ratio $(M_i - MTR_i)/TTR_i$ the set of items to repair to optimize the mass savings can be chosen up to a crew time availability. Figure 39 illustrates a graphical representation of this principle. This practice is repeated for each crewed mission where repair activities are allowed. Non-optimal repair lists will fall below the sorted repair list curve, indicating sub-optimal mass savings for a fixed repair time.

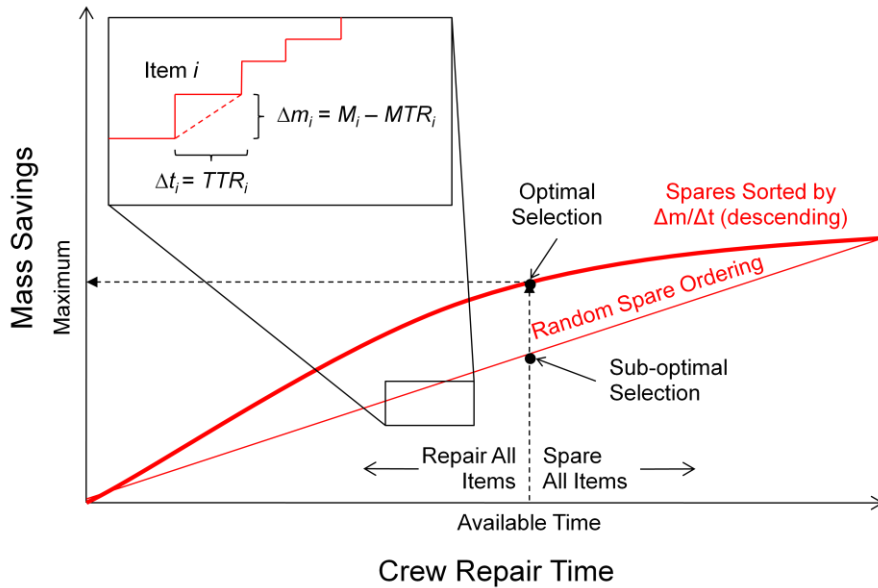


Figure 39: Repair policy for optimal mass savings. The set of items to repair to achieve optimal mass savings is found by sorting the repairable items in descending order by the benefit-cost ratio.

4.3.5 Manifest Tab

The manifest tab assists building a cargo manifest, following the steps of the manifesting model: demand generation, aggregation, packing, and manifesting (see Section 3.3). Manual manifesting consists of sequentially packing demands into resource containers and manifesting the containers onto carriers.

Supply P...	Resource	Amount	Packed
PAC (568.0)*	Generic COS 201 (...)	28.80 kg	28.80 kg
PAC (568.0)*	Generic COS 202 (F...	50.40 kg	50.40 kg
PAC (568.0)*	Generic COS 203 (G...	21.12 kg	21.12 kg
PAC (568.0)*	Generic COS 204 (H...	43.20 kg	43.20 kg
PAC (568.0)*	Generic COS 205 (Cl...	12.00 kg	12.00 kg
PAC (568.0)*	Generic COS 303 (H...	2.40 kg	2.40 kg
PAC (568.0)*	Generic COS 701 (...)	14.40 kg	14.40 kg
LSP (532.1)	Generic COS 4 (Mai...	59.73 kg	59.73 kg
LSP (532.1)	Generic COS 201 (...)	199.00 kg	199.00 kg
LSP (532.1)	Generic COS 202 (F...	251.46 kg	251.46 kg
LSP (532.1)	Generic COS 203 (G...	120.18 kg	120.18 kg
LSP (532.1)	Generic COS 204 (H...	215.46 kg	215.46 kg
LSP (532.1)	Generic COS 205 (Cl...	59.94 kg	59.94 kg
LSP (532.1)	Generic COS 303 (H...	12.06 kg	12.06 kg
LSP (532.1)	Generic COS 701 (...)	71.82 kg	71.82 kg
LSP (532.1)*	Generic COS 4 (Mai...	0.37 kg	0.37 kg
LSP (532.1)*	Generic COS 201 (...)	2.76 kg	2.76 kg
LSP (532.1)*	Generic COS 202 (F...	4.82 kg	4.82 kg
LSP (532.1)*	Generic COS 203 (G...	2.02 kg	2.02 kg
LSP (532.1)*	Generic COS 204 (H...	4.14 kg	4.14 kg
LSP (532.1)*	Generic COS 205 (Cl...	1.14 kg	1.14 kg
LSP (532.1)*	Generic COS 303 (H...	0.22 kg	0.22 kg
LSP (532.1)*	Generic COS 701 (...)	1.38 kg	1.38 kg
SL3 (529.2)	Generic COS 4 (Mai...	0.05 kg	0.05 kg
SL3 (529.2)	Generic COS 201 (...)	4.32 kg	4.32 kg
SL3 (529.2)	Generic COS 202 (F...	7.56 kg	7.56 kg

Supply P...	Resource	Amount
SL2 (526.5)	Generic COS 303...	10.74 kg
SL2 (526.5)	Generic COS 701...	16.48 kg

Container	Mass	Volume
Half CTB 112	28.22 kg	0.025 m³
Liquid Tank D...	36.39 kg	0.025 m³
Half CTB 106	28.22 kg	0.025 m³
CTB 94	47.17 kg	0.053 m³
Half CTB 81	28.22 kg	0.025 m³

Figure 40: Packing and manifesting interface. The manifesting tab helps the user to create a valid cargo manifest by guiding through the process of packing and manifesting containers.

Although demand generation and aggregation are automatic, packing resources into containers is the first task of manifesting. Since containers may be transferred across several transports before the resources are demanded, it is challenging to share containers across multiple spatial and temporal demand periods. The simplest packing scheme only allows demands from the same transport or exploration period to share a container, however there are some simple extensions that can be useful for sharing containers. In general, a partially-filled resource container may only be assigned more cargo if it has not reached its target destination before the new resources are demanded. Transport demands can be packed within a partially-filled container if it contains demands from the same transport or subsequent exploration periods at the transport destination node. Exploration demands can be packed within a partially-filled container if it contains demands from the same or later explorations at the same node.

After packing is complete, manifesting containers onto carriers must take place. Resource containers containing exploration demands may be manifested on one of the carriers of any transport arriving at the location of the exploration location at the same or earlier time as the earliest packed demand exploration period. However, if the origin of the transport is not a source node, the container must be supplied to the transport in a derivative demand.

Manual manifesting is tedious due to the recursive process of manifesting containers on subsequent transports. To help automate the process, a heuristic algorithm was implemented to create manifests. The resulting manifests are not optimal, meaning some feasible campaigns may appear infeasible and human modification may be necessary to correct logical errors.

The auto-manifesting algorithm depends on an auto-packing routine, shown below in pseudo code. The auto-packing scheme selectively uses existing partially-filled containers before creating new containers.

```
procedure auto-pack (Demand D) {
  for each existing resource container C {
    if D can be packed in C {
      pack D in C until capacity is reached
    }
  }
  while remaining D exists {
    create new resource container C
    pack D in C until capacity is reached
  }
}
```

The logic for the auto-manifesting algorithm puts emphasis on carry-along supplies, meaning demands are preferentially manifested on the transport closest in time before the point of demand, outlined below in pseudo code. Auto-packing is performed before manifesting each exploration period in attempt to

promote sharing of partially-packed containers. Also, priority is given to transport demands, as they must be manifested on the transport from which they are demanded.

```

procedure auto-manifest () {
  for each exploration period E, in reverse chronological order {
    T is the transport that supplies E
    for each demand D aggregated to T {
      auto-pack (D)
    }
    for each demand D aggregated to E {
      auto-pack (D)
    }
    for each resource container C containing a demand from T {
      for each transport S supplying E, in reverse chron. order {
        for each carrier R in S {
          if C can be manifested onto R {
            manifest C onto R
            go to next container
          }
        }
      }
      C could not be manifested: demands may be infeasible
    }
    for each container C containing a demand from E {
      for each transport S supplying E, in reverse chron. order {
        for each carrier R in S {
          if C can be manifested onto R {
            manifest C onto R
            go to next container
          }
        }
      }
      C could not be manifested: demands may be infeasible
    }
  }
}

```

The known limitations of auto-manifesting include the effects of carriers involved in multiple transports. First, carrier capacity calculations don't take into account containers manifested on previous transports not yet transferred to another carrier. This effect is partially due reverse-chronological manifesting and can cause spatial simulation errors if the auto-manifested containers violate mass or volume capacity limits. Second, containers are only transferred to a carrier immediately before a transport, potentially allowing containers to be inadvertently moved by carriers after delivery to a location. Future development efforts, in particular those of optimal manifesting methods discussed in Appendix B, should correct the deficiencies of present auto-manifesting.

4.3.6 Simulation Tab

The final simulation and analysis is performed with the simulation tab. In addition to simulation outputs including errors, it provides several visualizations for campaign analysis.

Network State History

The network state history visualization provides an animation of the movement of elements throughout the network. The animation runs forward at a set frame rate with either fixed simulation time per frame or one event per frame. Selecting specific dates enable investigation of the network state including element locations, resource amounts, and other element properties.

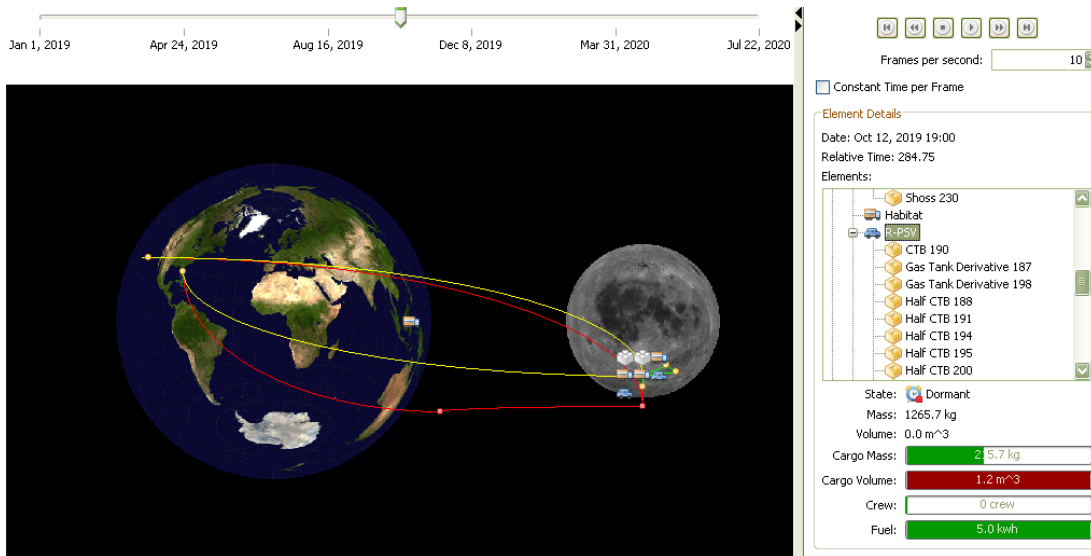


Figure 41: Network state history. The network state history shows the system state after each event execution. Elements can be inspected for contents including remaining propellant mass.

Measures of Effectiveness History

The measures of effectiveness (see Section 3.4.2) are the primary quantitative outputs of campaign simulation to assist analysis. Rather than simply providing a number to quantify the campaign, however, measure of effectiveness visualizations provide a time-history of metrics' evolution. The visualizations may help identify parts of a campaign that could be changed to provide improvement in a particular metric. Figure 42 illustrates an example exploration campaign in which the relative exploration capability (REC) does not increase monotonically as expected due to non-exploration mass launched.

Resource History Visualizations

Resource history visualizations track the supply and demand of resources during the campaign. Resources

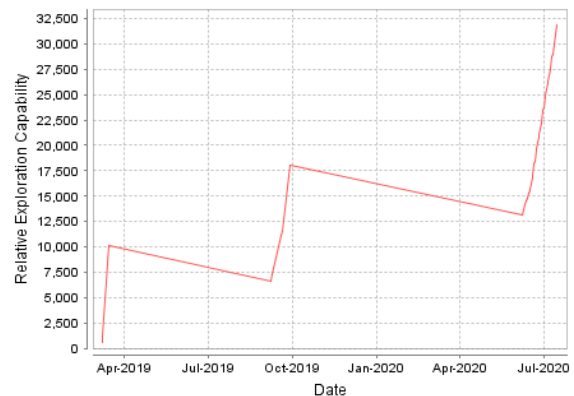


Figure 42: Relative exploration capability history. REC does not increase monotonically over a campaign due to non-exploration mass launched.

are grouped by base class of supply and can be filtered by network location or by element. Resource history by element is best used to track propellant usage during burns and resource history by location is best used to track exploration consumption.

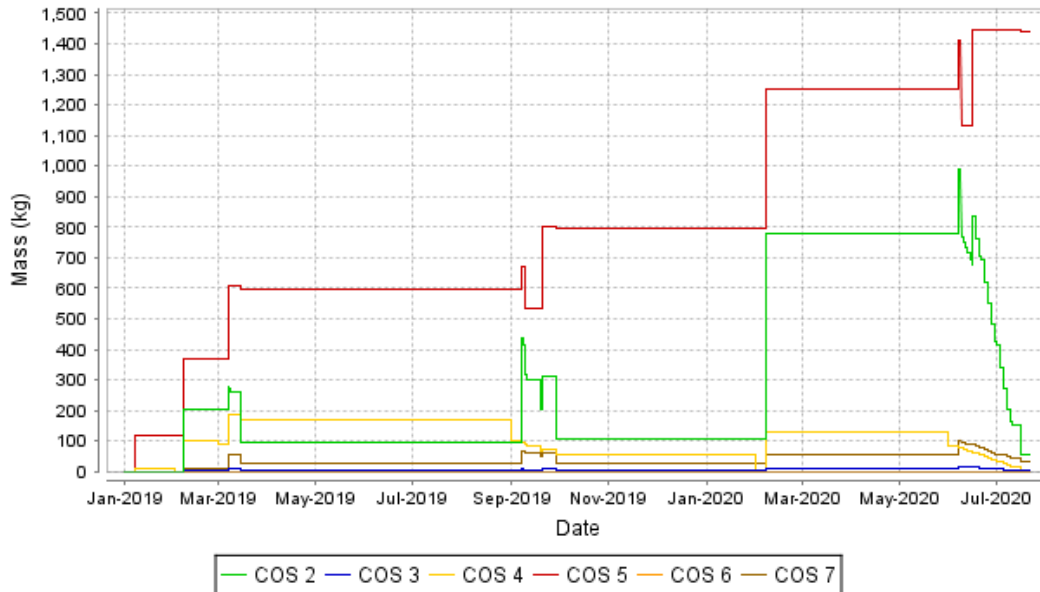


Figure 43: Location resource history. *The resource history during an example campaign shows a build-up of COS 5, representing empty resource containers.*

5 SpaceNet 2.5 Applications

This chapter presents a series of four case studies and a controlled user study to illustrate the applications of SpaceNet 2.5 in space exploration campaign analysis. The case studies are targeted towards campaigns considered for future human space exploration to demonstrate feasibility and flexibility of the SpaceNet modeling framework. The user study is focused on highlighting the usability and timescales required for performing campaign analysis using the SpaceNet 2.5 software.

5.1 Case Studies

The four case studies selected represent campaigns considered during the development of SpaceNet 2.5 and new concepts under development. First, as an example of an operational space exploration campaign with many flights, the resupply of the International Space Station is considered between 2010 and 2015. Second, a sortie exploration to a near-Earth object (NEO) demonstrates an exploration scenario not considered during initial development. Third, a lunar outpost exploration campaign similar to NASA Constellation program plans is used as an example of a long-duration planetary exploration with significant surface infrastructure. Finally, a Mars surface exploration similar to a NASA design reference mission is used as an example of an exploration having long-duration transportation segments.

As demonstrative case studies, the level of fidelity and technical correctness and completeness is limited – a more in-depth analysis for each case should be undertaken to establish validated results. In particular, the most visible inconsistencies may be present in the ISS resupply case study as it attempts to analyze an existing system using an imperfect model and limited information. As conceptual campaigns, the other three case studies model technologies and designs under development, requiring refinements as unknowns are resolved.

5.1.1 International Space Station Resupply

At the time of writing, there remain two scheduled flights of the NASA Space Transportation System (STS) while the ISS lifetime has been extended to 2020 or beyond [20]. Maintaining the crew and operations at the ISS in the coming years without the support of the shuttle is not a trivial task [34]. NASA has indicated commercial on-orbit transportation services (COTS), also known as commercial resupply services (CRS), will play a large role in supplying the ISS by issuing contracts to Orbital and SpaceX for use of Cygnus and Dragon spacecraft respectively [35]. Combined with the efforts of ESA's

automated transfer vehicle, ATV, JAXA’s H-II transfer vehicle, HTV, and RKA’s Progress and Soyuz, the ISS will become a complicated supply hub.

The goal of this case study is to model the final assembly and subsequent resupply of the ISS including all scheduled and expected flights through December 2015. Without sophisticated demand models to estimate demands for individual resources and spares, the analysis will focus on lumped mass demands by class of supply using parametric models for crew consumables and spares [36]. The case study does not consider down-mass capability, improvements to launch vehicles or spacecraft capacities, resources pre-positioned at the ISS before 2010, differences between cargo types (e.g. dry, water, or gas), or individual crew rotations.

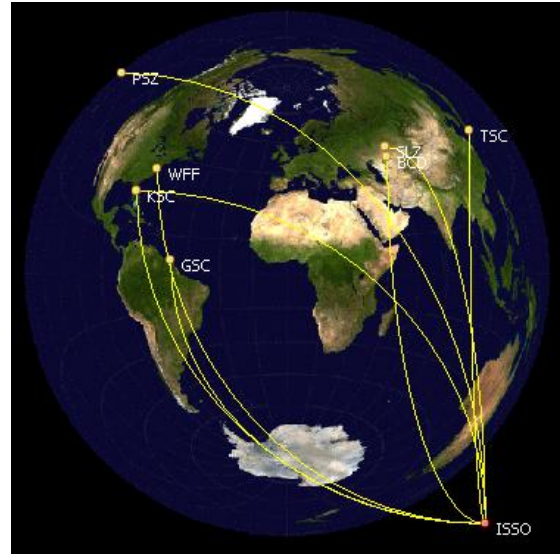


Figure 44: ISS resupply network. Visualization of Earth launch sites, landing zones, and the ISS in orbit.

Model Inputs

Most model inputs for the ISS resupply case study were derived from spacecraft datasheets where available or from publicly-available online databases. All values are approximate due to modeling simplifications and assumptions, vehicle configurations, and design evolution.

Table 19 lists the nodes considered in this case including launch and landing sites on Earth and the ISS in orbit. By modeling the entire ISS as a single node, resources are shared between all modules in orbit.

Table 19: ISS resupply nodes.

Abbrev.	Description	Parameters
ISS	International Space Station	360 km x 347 km, 51.6°
KSC	Kennedy Space Center	28.6° N, 80.6° W
CCAS	Cape Canaveral Air Station	34.9° N, 117.8° W
WFF	Wallops Flight Facility	28.4° N, 80.6° W
TSC	Tanegashima Space Center	30.4° N, 131.0° E
GSC	Guiana Space Center	5.0° N, 52.8° W
BCD	Baikonur Cosmodrome	45.9° N, 63.3° E
SLZ	Soyuz Landing Zone	50° N, 67.5° E
PSZ	Pacific Splashdown Zone	15° N, 160° W

Table 20 lists the edges corresponding to launch vehicle capabilities and landings. Using abstracted flight edges avoids the definition of launch vehicles and propellants. In all cases but the STS, launch vehicles are independent from the crew and cargo carriers, enabling for a clean definition of the flight edges. In the case of the STS, the flight edge represents the carrying capacity of the shuttle without its structural mass.

Table 20: ISS resupply edges.

Name	Origin	Destination	Max Crew	Max Cargo [kg]
Soyuz-FG Launch to ISS	BCD	ISS	3	7200
Soyuz-2 Launch to ISS	BCD	ISS	3	8500
Soyuz Landing at SLZ	ISS	SLZ	3	150
Falcon 9 Launch to ISS	CCAS	ISS	0	10450
Dragon Splashdown at PSZ	ISS	PSZ	0	3000
Taurus II Launch to ISS	WFF	ISS	0	6600
H-IIB Launch to ISS	TSC	ISS	0	16500
Ariane 5 ES Launch to ISS	GSC	ISS	0	19300
STS Launch to ISS	KSC	ISS	7	16050
STS Shuttle Landing at KSC	ISS	KSC	10	9500
Proton-M Launch to ISS	BCD	ISS	0	21600

Table 21 lists the element definitions used in the case study. Many elements correspond to the spacecraft carrying crew and cargo to the ISS, though some represent infrastructure and logistics containers. As discussed with the flight edges, the STS shuttle element does not include its infrastructure mass which is considered in the flight edge parameters.

Both the ISS and its crew produce demands with linear demand models. Annual ISS demands, including packaging mass, are estimated at 10 tons of spares and maintenance and 15 tons of science payloads. Daily crew demands are estimated at 2 kilograms of food, 3.5 kilograms of water, 1 kilogram of gases, 0.5 kilogram of hygiene items, and 0.5 kilogram of waste disposal items per crew member.

Table 21: ISS resupply elements.

Name	Dry Mass [kg]	Max Crew	Max Cargo [kg]	Description
Progress-M	4900	0	2350	RKA Progress (M Configuration)
Soyuz-TMA	6085	3	100	RKA Soyuz (TMA Configuration)
Dragon	4200	0	6000	SpaceX Dragon
Cygnus	3500	0	2000	Orbital Cygnus
Cygnus-M	3500	0	2700	Orbital Cygnus (Improved)
HTV	8100	0	6000	JAXA H-II Transfer Vehicle
ATV	11700	0	7600	ESA Automated Transfer Vehicle
STS Shuttle	0	10	16050	NASA Space Transportation System Shuttle
MLM	20300	0	0	RKA Multifunctional Laboratory Module
ELC	4400	0	2000	EXPRESS Logistics Carrier
PMM	4080	0	9070	Pressurized Multipurpose Module
AMS	6700	-	-	Atomic Magnetic Spectrometer
ISS	335000	6	35000	International Space Station

Missions

The official mission manifest provided by NASA only covers missions through the end of 2010 [36]. A complete mission manifest through 2015 was created using unofficial launch and mission manifests provided by Orbital, SpaceX, JAXA, and ESA, as well as extrapolating launch rates for Progress and Soyuz. The missions are comprised of 2 STS, 22 Progress, 22 Soyuz, 12 Dragon, 8 Cygnus, 5 HTV, and 4 ATV resupply missions and one assembly mission to replace the *Pirs* module with *Nauka*. In addition to the resupply missions, the first mission, number zero, initializes the ISS and its crew in orbit to start the demands generation.

Although it is immaterial to this analysis, it is assumed that each Soyuz spacecraft spends 180 days docked at the ISS before the subsequent return to Earth. All other spacecraft spend 60 days docked at the ISS before de-orbiting or return to Earth.

Table 22: ISS resupply missions 0-22 (2010-2011).

#	Date	Mission	Flight(s)	Element(s)
0	9/1/2010	(Initial Conditions)	-	ISS, Crew Members A-F
1	9/8/2010	Progress M-08M	Soyuz-FG Launch	Progress M “M-08M”
2	9/16/2010	STS 133	STS Launch STS Shuttle Landing	STS Shuttle “Discovery” ELC “ELC3” PMM “Leonardo”
3	9/29/2010	Soyuz TMA-01M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-01M”
4	10/27/2010	Progress M-09M	Soyuz-FG Launch	Progress M “M-09M”
5	11/10/2010	STS 134	STS Launch STS Shuttle Landing	STS Shuttle “Endeavor” ELC “ELC4” AMS “AMS-02”
6	11/18/2010	ATV-2	Ariane 5 ES Launch	ATV “Johannes Kepler”
7	11/30/2010	Soyuz TMA-20	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-20”
8	1/1/2011	HTV-2	H-IIB Launch	HTV “HTV-2”
9	2/9/2011	Progress M-10M	Soyuz-FG Launch	Progress M “M-10M”
10	3/1/2011	Soyuz TMA-21	Soyuz-FG Launch Soyuz Landing	Soyuz TMA “TMA-21”
11	3/15/2011	Progress M-11M	Soyuz-FG Launch	Progress M “M-11M”
12	4/1/2011	HTV-3	H-IIB Launch	HTV “HTV-3”
13	4/15/2011	Progress M-12M	Soyuz-FG Launch	Progress M “M-12M”
14	5/1/2011	Dragon-1	Falcon 9 Launch Dragon Splashdown	Dragon “Dragon-1”
15	5/15/2011	Progress M-13M	Soyuz-FG Launch	Progress M “M-13M”
16	5/30/2011	Soyuz TMA-02M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-02M”
17	6/15/2011	Progress M-14M	Soyuz-FG Launch	Progress M “M-14M”
18	7/15/2011	Progress M1-01M	Soyuz-2 Launch	Progress M “M1-01M”
19	8/1/2011	Cygnus-1	Taurus II Launch	Cygnus “Cygnus-1”
20	8/15/2011	Progress M-15M	Soyuz-FG Launch	Progress M “M-15M”
21	9/15/2011	Soyuz TMA-22	Soyuz-FG Launch Soyuz Landing	Soyuz TMA “TMA-22”
22	11/1/2011	Dragon-2	Falcon 9 Launch Dragon Splashdown	Dragon “Dragon-2”

Table 23: ISS resupply missions 23-45 (2011-2013).

#	Date	Mission	Flight(s)	Element(s)
23	12/1/2011	Soyuz TMA-03M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-03M”
24	12/15/2011	Progress M-16M	Soyuz-FG Launch	Progress M “M-16M”
25	12/30/2011	Nauka Assembly	Proton-M Launch	MLM “Nauka”
26	2/1/2012	Progress M-17M	Soyuz-FG Launch	Progress M “M-17M”
27	3/1/2012	Soyuz TMA-04M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-04M”
28	4/1/2012	HTV-4	H-IIB Launch	HTV “HTV-4”
29	4/15/2012	ATV-3	Ariane 5 ES Launch	ATV “Edoardo Amaldi”
30	5/1/2012	Dragon-3	Falcon 9 Launch Dragon Splashdown	Dragon “Dragon-3”
31	5/15/2012	Soyuz TMA-05M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-05M”
32	7/1/2012	Progress M-18M	Soyuz-FG Launch	Progress M “M-18M”
33	8/1/2012	Soyuz TMA-06M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-06M”
34	9/1/2012	Cygnus-2	Taurus II Launch	Cygnus “Cygnus-2”
35	10/1/2012	Dragon-4	Falcon 9 Launch Dragon Splashdown	Dragon “Dragon-4”
36	11/1/2012	Soyuz TMA-07M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-07M”
37	12/1/2012	Progress M-19M	Soyuz-FG Launch	Progress M “M-19M”
38	1/1/2013	Soyuz TMA-08M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-08M”
39	2/1/2013	Progress M-20M	Soyuz-FG Launch	Progress M “M-20M”
40	3/1/2013	Dragon-5	Falcon 9 Launch Dragon Splashdown	Dragon “Dragon-5”
41	4/1/2013	HTV-5	H-IIB Launch	HTV “HTV-5”
42	5/1/2013	Cygnus-3	Taurus II Launch	Cygnus-M “Cygnus-3”
43	6/1/2013	Soyuz TMA-09M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-09M”
44	7/1/2013	Dragon-6	Falcon 9 Launch Dragon Splashdown	Dragon “Dragon-6”
45	8/1/2013	Soyuz TMA-10M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-10M”

Table 24: 2014 ISS resupply missions 46-68 (2013-2015).

#	Date	Mission	Flight(s)	Element(s)
46	9/1/2013	Progress M-21M	Soyuz-FG Launch	Progress M “M-21M”
47	9/15/2013	ATV-4	Ariane 5 ES Launch	ATV “ATV-004”
48	10/1/2013	Cygnus-4	Taurus II Launch	Cygnus-M “Cygnus-4”
49	11/1/2013	Dragon-7	Falcon 9 Launch Dragon Splashdown	Dragon “Dragon-7”
50	11/15/2013	Soyuz TMA-11M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-11M”
51	12/15/2013	Progress M-22M	Soyuz-FG Launch	Progress M “M-22M”
52	1/15/2014	Soyuz TMA-12M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-12M”
53	2/15/2014	Progress M-23M	Soyuz-FG Launch	Progress M “M-23M”
54	3/1/2014	Dragon-8	Falcon 9 Launch Dragon Splashdown	Dragon “Dragon-8”
55	4/1/2014	HTV-6	H-IIB Launch	HTV “HTV-6”
56	4/15/2014	Cygnus-5	Taurus II Launch	Cygnus-M “Cygnus-5”
57	5/1/2014	Soyuz TMA-13M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-13M”
58	6/1/2014	Dragon-9	Falcon 9 Launch Dragon Splashdown	Dragon “Dragon-9”
59	7/1/2014	Progress M-24M	Soyuz-FG Launch	Progress M “M-24M”
60	8/1/2014	Soyuz TMA-14M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-14M”
61	9/1/2014	Dragon-10	Falcon 9 Launch Dragon Splashdown	Dragon “Dragon-10”
62	10/1/2014	Progress M-25M	Soyuz-FG Launch	Progress M “M-25M”
63	11/1/2014	Soyuz TMA-15M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-15M”
64	12/1/2014	Cygnus-6	Taurus II Launch	Cygnus-M “Cygnus-6”
65	1/1/2015	Progress M-26M	Soyuz-FG Launch	Progress M “M-26M”
66	2/1/2015	Soyuz TMA-16M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-16M”
67	2/15/2015	ATV-5	Ariane 5 ES Launch	ATV “ATV-005”
68	3/1/2015	Dragon-11	Falcon 9 Launch Dragon Splashdown	Dragon “Dragon-11”

Table 25: ISS resupply missions 69-77 (2015).

#	Date	Mission	Flight(s)	Element(s)
69	4/1/2015	HTV-7	H-IIB Launch	HTV “HTV-7”
70	5/1/2015	Soyuz TMA-17M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-17M”
71	6/1/2015	Cygnus-7	Taurus II Launch	Cygnus-M “Cygnus-7”
72	7/1/2015	Progress M-27M	Soyuz-FG Launch	Progress M “M-27M”
73	8/1/2015	Soyuz TMA-18M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-18M”
74	9/1/2015	Dragon-12	Falcon 9 Launch Dragon Splashdown	Dragon “Dragon-12”
75	10/1/2015	Cygnus-8	Taurus II Launch	Cygnus-M “Cygnus-8”
76	11/1/2015	Progress M-28M	Soyuz-FG Launch	Progress M “M-28M”
77	12/1/2015	Soyuz TMA-19M	Soyuz-FG Launch Soyuz Landing	Soyuz-TMA “TMA-19M”

Analysis and Discussion

The mission bat chart, shown in Figure 45, highlights the immense number of transports to and from ISS between August 2010 and December 2015. Docking activities at the ISS will require action for 18 arrivals per year on average, or one arrival every 20 days. For comparison, there were 14 transports to ISS in 2009, and 11 in 2008, and only 9 in 2007.

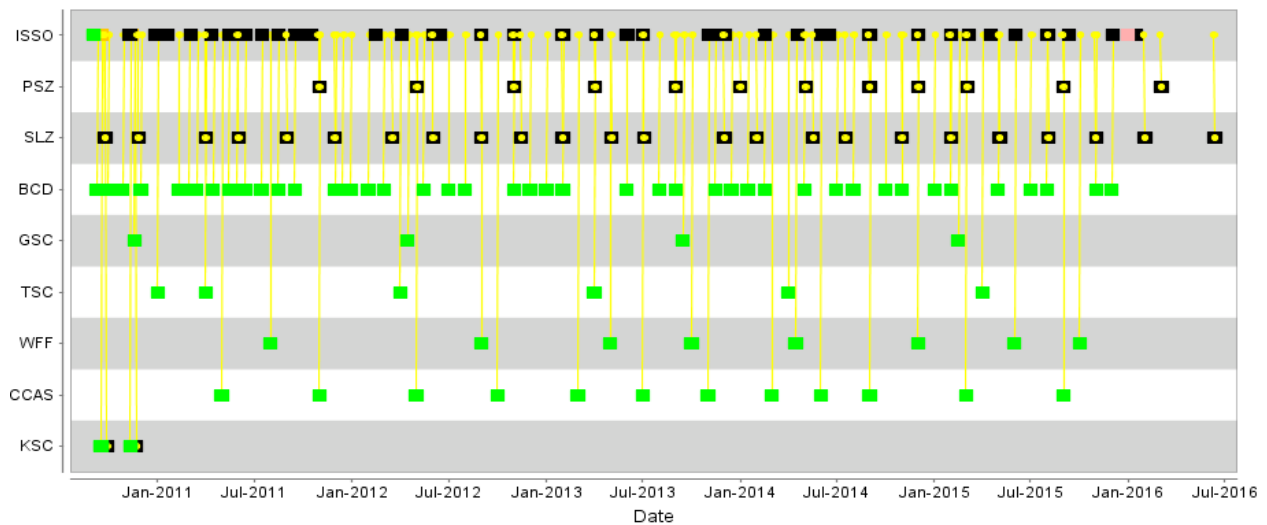


Figure 45: ISS resupply mission bat chart. Green squares are element instantiations, yellow lines are flight transports, black squares are elements removed from simulation.

Due to the simplifications in modeling demands, analysis of the ISS resupply case is limited to aggregate logistics feasibility rather than an intensive cargo manifesting analysis. The most useful visualization is the scenario feasibility plot, shown in Figure 46. Since this case only has only a single destination, it is feasible as the estimated demands remain below the remaining capacity line. The total raw capacity to ISS over the simulation is 245 tons, with 225 tons remaining after considering delivered elements on the shuttle missions (ELC, AMS, etc.). The demands over the same time period total 220 tons, of which 82 tons are for crew provisional items (COS 2), 80 tons are payloads for exploration and research (COS 6), 52 tons are for maintenance and upkeep (COS 4), and 6 tons are for waste and disposal (COS 7).

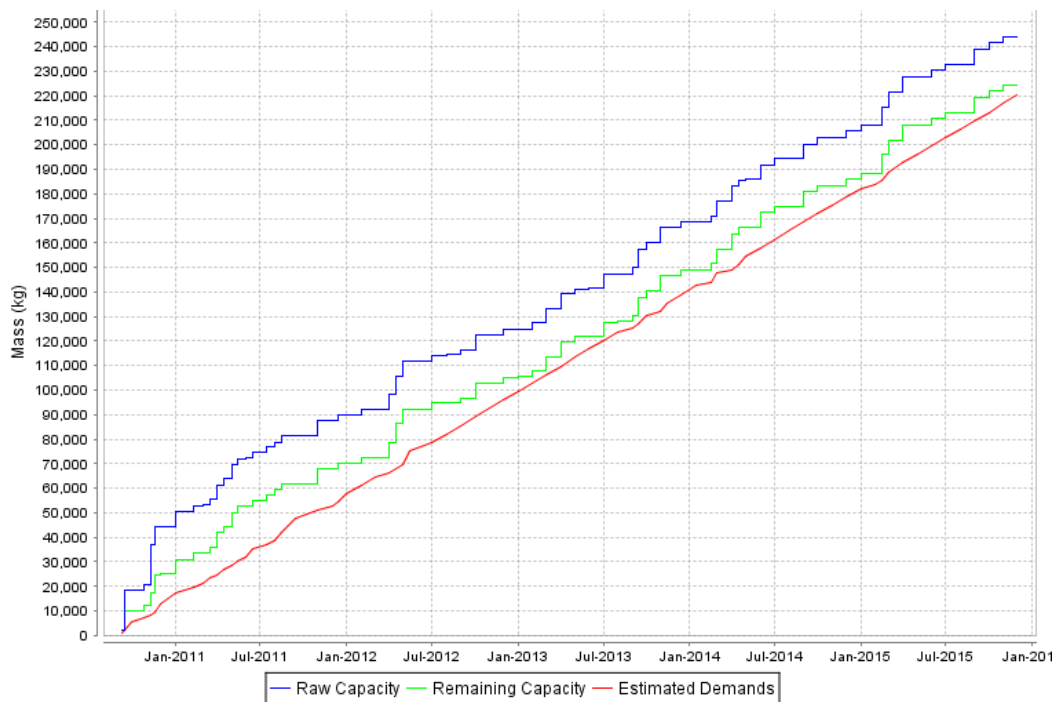


Figure 46: ISS resupply scenario feasibility. Cumulative plot of raw supply vehicle capacity, remaining capacity after pre-manifested elements, and aggregated demands indicating feasibility.

Although not modeled, any pre-positioned resources at ISS would effectively shift the estimated demands curve down by a fixed amount no more than the maximum estimated capacity of 35 tons. Analysis without considering these pre-positioned resources focuses on the steady-state supply and demand. The steady-state supply margin repeatedly falls below 5 tons, warranting additional analyses using two strategies. First, sensitivity studies for launch schedule and spacecraft availability help identify periods of high risk for supply. Second, sensitivity studies for demand rates help illustrate the effects of technology improvement on the reduction of demands.

As a hypothetical example of the impact of spacecraft availability, consider the conceivable case that one of the COTS vehicles fail to meet their contract. In Figure 47, all of the Orbital Cygnus spacecraft have

been removed from the supply capacity lines. The first signs of infeasibilities appear in early 2013 in this scenario, however depending on the amount of pre-positioned resources not modeled, there may not be any serious problems until 2014 or 2015, where the infeasible margin exceeds 10 tons.

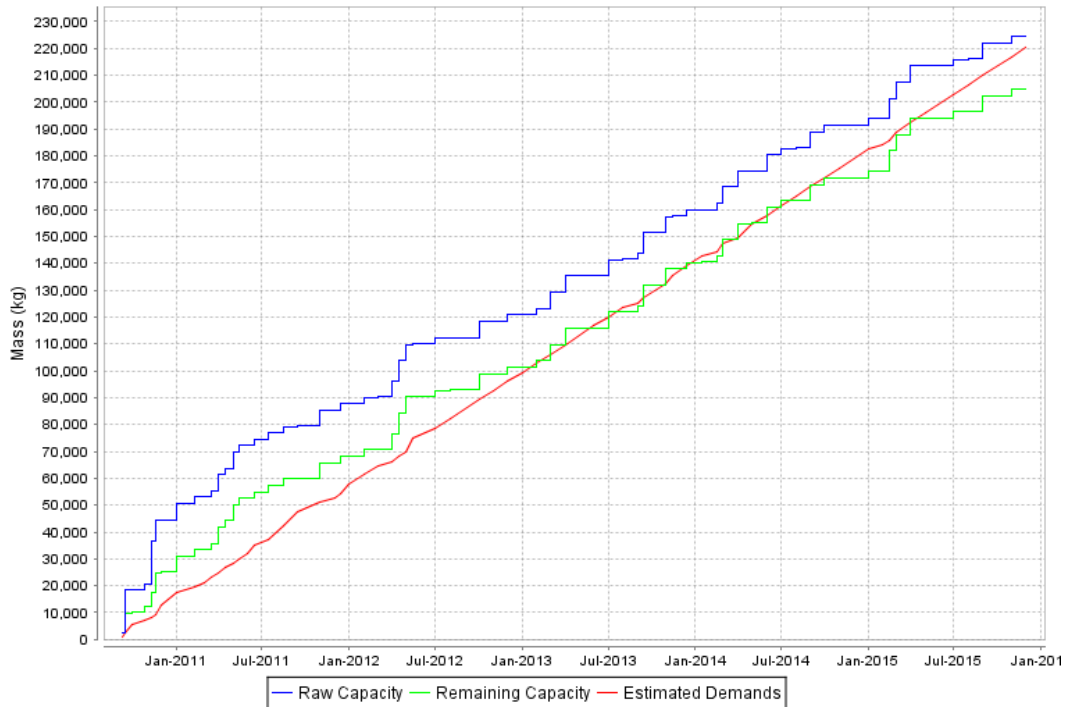


Figure 47: ISS resupply feasibility with Cygnus COTS failure. *Without the Cygnus spacecraft, steady-state infeasibilities first start to appear in early 2013 and only worsen into 2014 and 2015.*

As another hypothetical example, consider a 5-ton advanced water recovery system (AWRS) delivered by Dragon-6 in July 2013 capable of reducing the crew water demands from 3.5 to 0.5 kilograms per person per day. This trade between delivery capacity and future demands is modeled by the adding a new element, the AWRS, to the Dragon-6 before launch. After delivery to the ISS, the six crew members are reconfigured to a new state representing the lower demands for water resources.

Figure 48 shows the resulting feasibility chart highlighting the benefits that could be realized with such a decision. Although the cumulative remaining capacity is reduced by 5 tons, the mass of the AWRS, the cumulative demands decrease by a much more substantial margin of almost 40 tons by the end of 2015. Although only a notional example, similar analysis could inform technology development for advanced life support systems.

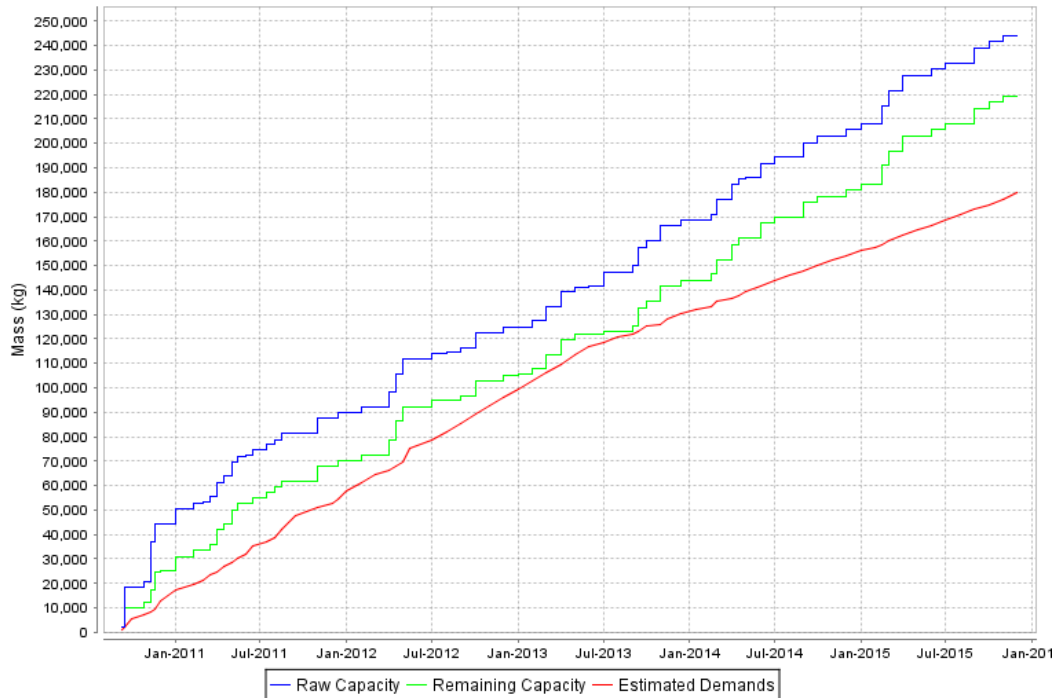


Figure 48: ISS resupply with advanced water recovery. An advanced water recovery system is delivered in July 2013 reducing demands for water from 3.5 to 0.5 kg/person/day.

Conclusions

Though only a high-level analysis without access to detailed demand models, the resupply of the ISS through 2015 warrants significant additional research. As modeled, there is limited supply capacity in steady-state, indicating undersupplies of critical resources may be a realistic concern. Steady-state infeasibilities could start to occur with the delay or cancellation of just one of the six resupply spacecraft. Also, as there is no net aggregation in resources or a high-capacity resupply vehicle, additional infeasibilities could come from a multiple failure event depleting any available spares, a conceivable possibility with the next solar maximum occurring in 2013. In order to reduce demands, efforts should also be taken to implement high-closure ECLSS systems as early as possible for maximum impact.

More detailed analysis should include additional demands for propellant required for orbital re-boost and station keeping and differentiate between pressurized, unpressurized and liquid cargo, including the multiple spacecraft configurations supporting differing capacities of each type. Provided additional information, pre-positioned resources could be modeled aboard ISS to provide a more sophisticated feasibility analysis. Finally, down-mass capacity should be quantified to inform capability to return failed equipment for repair and analysis.

5.1.2 Near-Earth Object Sortie

Concepts for human explorations to the asteroids, comets, and other objects having similar orbits to Earth (collectively called near-Earth objects, or NEOs) have existed as early as 1966 when Smith proposed a 500-plus day mission to Eros using modified Apollo spacecraft [38]. Such a mission could gain operational experience outside the Earth-Moon system without the cost required for a Martian landing.

More recently, NASA’s Advanced Projects Office performed feasibility studies for missions to NEOs [39, 40]. NEO exploration could improve technical readiness levels for space hardware, evaluate *in-situ* resource utilization systems, and provide a wealth of new information. Also, exploration would resemble docking operations and EVAs without requiring landing or ascent spacecraft. The NEOs investigated are reachable with 150-day missions including 1999 AO10, 2000 SG344, and 2006 DQ14.

The HSF review committee’s final report also mentions exploration of NEOs as one step on a flexible path to human space exploration [18]. The Flexible Path strategy calls for explorations of increasing duration and technical difficulty, starting with a lunar flyby or orbit, visiting Lagrange points, and exploring NEOs before exploring Mars.

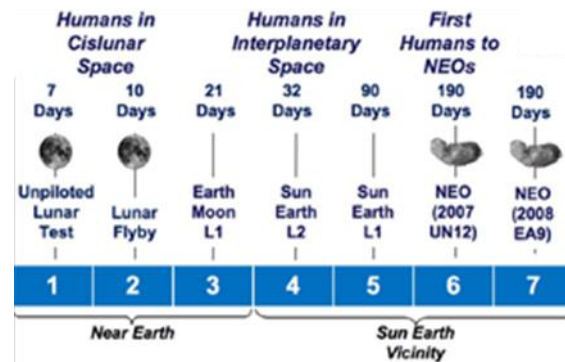


Figure 49: Flexible path strategy. Timeline of milestones, destinations and capabilities of the Flexible Path strategy, adapted from [18].

The feasibility of a 14-day exploration at NEO 1999 AO10 is considered for this case study due to a favorable launch opportunity within a conceivable timeline. Past research claims 1999 AO10 is reachable using Constellation program spacecraft with limited modifications including the Orion crew exploration vehicle and Ares V heavy lift launch vehicle [40]. The target of analysis is to evaluate the feasibility of such a mission and to propose modifications if required.



Figure 50: NEO sortie network. Visualization of space edges between launch, low-Earth orbit, the near-Earth object 1999 AO10, and splashdown in the Pacific Ocean.

Model Inputs

The nodes required for this campaign are limited to the launch and landing sites on Earth, the parking orbit in low-Earth orbit, and 1999-AO10. The NEO is roughly modeled as an orbital node about the sun.

Table 26: NEO sortie nodes.

Abbrev.	Description	Parameters
KSC	Kennedy Space Center	28.6° N, 80.6° W
LEO	Low Earth Parking Orbit	166.7 km x 166.7 km, 38.0°
AO10	Near-Earth Object 1999 AO10	1.01 AU x 0.81 AU, 2.62°
PSZ	Pacific Splashdown Zone	15° N, 160° W

The edges represent launch and in-space propulsive burn sequences based on delta-v estimates for a mission departing in September 2025 [40]. Reaction control system (RCS) and attitude control burns are not considered.

Table 27: NEO sortie edges.

Name	Origin	Destination	Duration [days]	Propulsive Burns
KSC Launch to LEO	KSC	LEO	1	t+0.0: 9.8 km/s (OMS)
LEO to AO10	LEO	AO10	111	t+0.0: 3.291 km/s (OMS) t+111: 2.193 km/s (OMS)
AO10 to Splashdown at PSZ	ISS	SLZ	31	t+0.0: 1.746 km/s (OMS)

All element designs are notional as no current plans exist for a NEO mission, but based on available data for Constellation program spacecraft. The elements used for the mission include an Ares V heavy-lift launch vehicle and an Orion crew exploration vehicle. The Orion modifications include reducing the crew module (CM) crew size to two to improve the cargo storage capacity and the inclusion of a scientific instrumentation payload (SIP) placed inside the service module (SM).

A few assumptions are included for in-space propulsion. First, zero-loss cryo-coolers are assumed to be available to prevent the boil-off of the EDS cryogenic propellant over the duration of the transit to 1999-AO10. Also, both the EDS and SM are assumed to be restartable. Additionally, as the crew exploration vehicle does not contain an airlock, all exploration will be tele-operated without EVAs.

To model demands, each crew member is assigned a linear demand model to generate 7.5 kilograms per day of generic crew consumables (COS 2). All required spares and maintenance mass for the mission duration is assumed to be included in element dry mass estimates.

Table 28: NEO sortie elements.

Name	Dry Mass [kg]	Max Crew	Max Cargo [kg]	Fuel [kg] (Type)	Isp [s]	Description
SRBs	213000	0	0	1,370,000 PBAN	269	Solid Rocket Boosters (2)
Core	173,680	0	0	1,587,000 LOX/LH2	414	Core Stage
Interstage	9190	-	-	-	-	Interstage Element
EDS	26,390	0	0	253,000 LOX/LH2	449	Earth Departure Stage
CM	9600	2	1500	-	-	Crew Module
SM	3000	0	1000	10,000 MMH/N2O4	301	Service Module
SIP	1000	-	-	-	-	Science Instrument Payload
LAS	3700	0	0	2500 HTPB	250	Launch Abort System
SA	500	-	-	-	-	Spacecraft Adapter

Missions

There is only one sortie mission to 1999-AO10. The launch from Kennedy Space Center uses a staging of the two solid rocket boosters, the Ares V core, and the Earth departure system to achieve low-Earth orbit. Once in low-Earth orbit, the Earth departure stage is used to depart from Earth orbit. Upon arrival at 1999 AO10, the Earth departure stage is fired for the last time followed by a burn of the service module. The exploration operations take place over a period of two weeks, followed by a return to Earth using the service module engine.

Table 29: NEO sortie mission events.

Date(s)	Event	Details
9/19/2025- 9/19/2025	Launch from KSC to LEO parking orbit	Burn Boosters, Stage Boosters, Stage LAS, Burn Core, Stage Core, Stage Interstage, Burn EDS
9/19/2025- 1/08/2026	Traverse in-space edge to 1999 AO10	(Departure) Burn EDS (Arrival) Burn EDS, Stage EDS, Stage SA, Burn SM
1/08/2026- 1/22/2026	Exploration at AO10	Tele-operated exploration (no EVAs)
1/22/2026- 2/22/2026	Traverse in-space edge to Pacific splashdown	Burn SM, Stage SM

Analysis and Discussion

As initially specified, the mission to 1999 AO10 is infeasible due to insufficient propellant for the return transport to Earth. Even without considering manifested cargo to satisfy the crew demands, there is approximately 525 m/s delta-v unachievable by the service module burn. In addition, the mission is logistically infeasible because there is not enough cargo space aboard the crew module to satisfy the demands generated over the course of the 150-day mission. In fact, the transit from low-Earth orbit to 1999 AO10 itself exhausts the 1500 kilogram cargo capacity without considering packing mass.

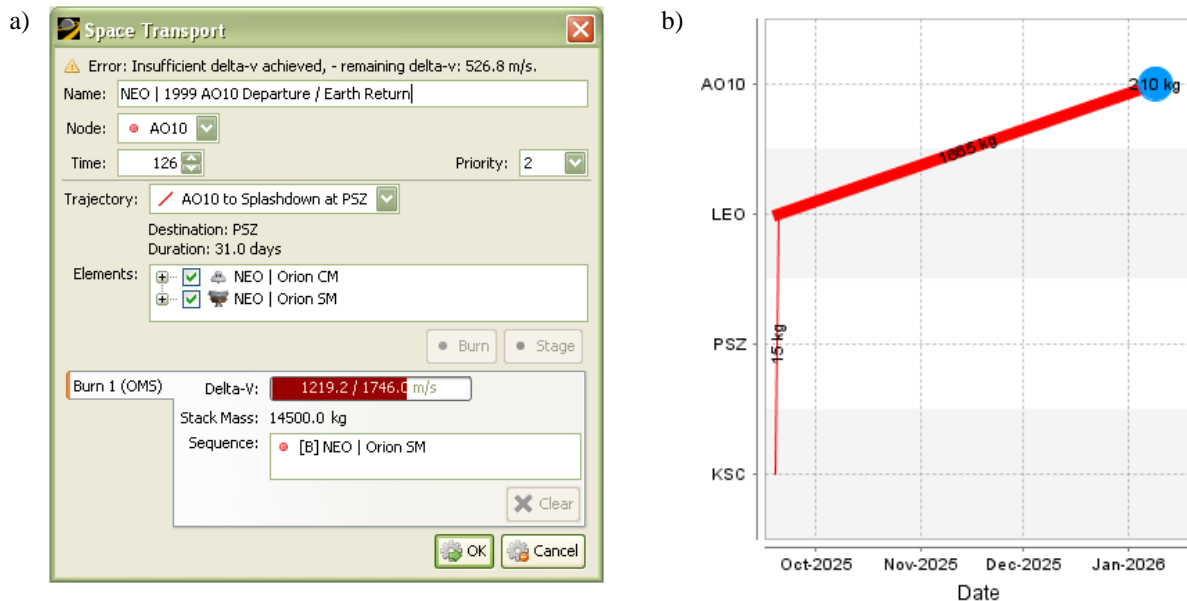


Figure 51: NEO sortie infeasibilities. (a) The service module burn to return to Earth has a 526.8 m/s delta-v deficit without considering cargo. (b) The 111-day transit to 1999-AO10 alone exhausts the crew module cargo capacity.

In attempt to correct these infeasibilities, a modified NEO sortie was constructed with several changes to improve the performance. The included changes may not be technologically possible, but are presented as a method to iterate on infeasible mission designs.

- Increase EDS fuel capacity by 20% from 253,000 to 303,600 kilograms
- Decrease EDS mass by 10% from 26,390 to 23,750 kilograms
- Increase CM cargo capacity from 1500 to 2500 kilograms
- Decrease CM mass by 10% from 9600 to 8640 kilograms
- Increase SM fuel capacity by 10% from 10,000 to 11,000 kilograms
- Science instrumentation payload is not returned to Earth

The revised NEO exploration is both propulsively and logistically feasible, though with tight propellant margins. Although the delta-v calculation does take into account the mass of manifested resources, all

resources are assumed to be discarded from the spacecraft after consumption. The service module only has 250 kilograms of excess propellant upon its return to Earth, as shown in Figure 52.

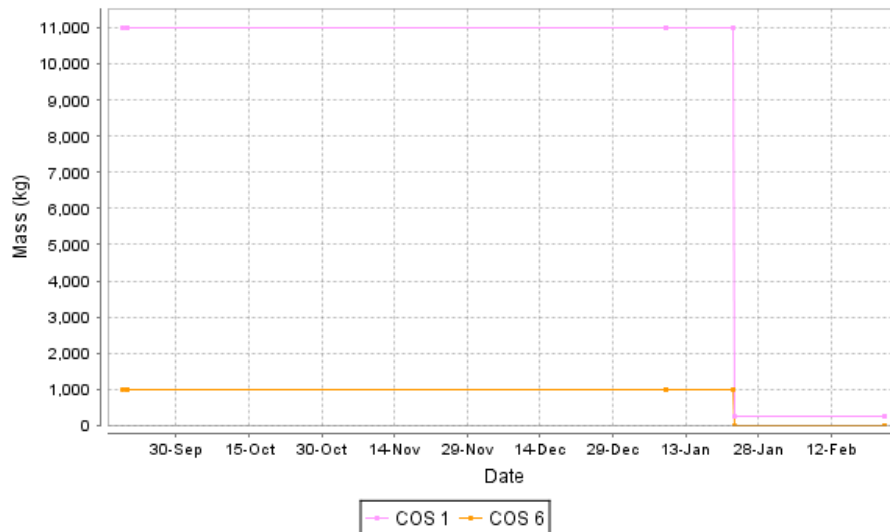


Figure 52: Modified NEO sortie service module resource history. *The science instrument payload (COS 6) is discarded. The service module propellant (COS 1) margin is 250 kg.*

A similar inspection of the crew module in Figure 53 highlights the consumption of the crew consumable resources as well as the usage of the packing materials. In this case, the automated packing method exclusively uses cargo transfer bags (CTBs) and half CTBs with nearly a 100% packing fraction for each container due to the continuous nature of the abstracted resources. Additional analysis may restrict the volume of resources in each CTB or introduce less mass-efficient containers such as gas and water tanks.

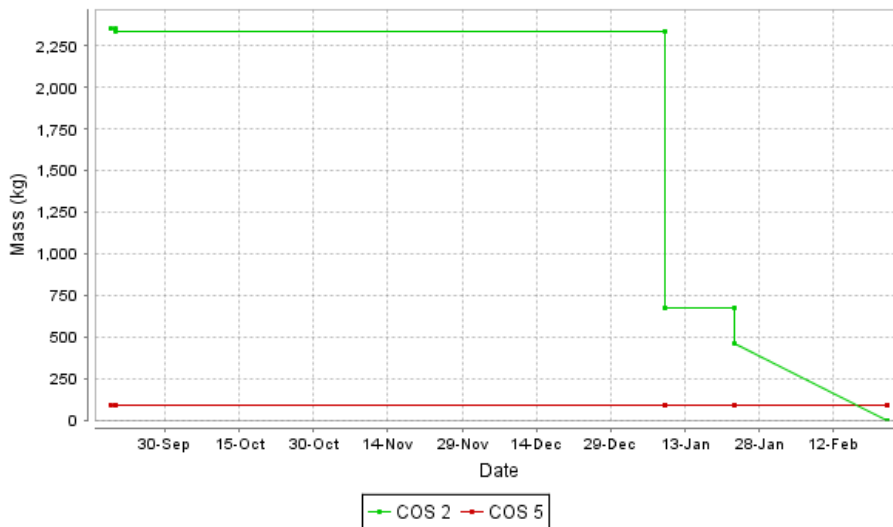


Figure 53: Modified NEO sortie crew module resource history. *All crew consumables (COS 2) are used in the crew module. Packing materials (COS 5) total about 100 kg of the cargo.*

There are two residual concerns about the feasibility of the mission as stated. First, there are no provisions for an airlock on the crew module, requiring all crewed exploration would take place from within the spacecraft. This would still provide scientific value, as crew could tele-operate rovers or other robotic elements on the surface of the asteroid, but it would not be as fulfilling as an EVA. The addition of an airlock to the spacecraft would significantly increase the mass, further constraining feasibility.

The second concern is for the boil-off of the cryogenic propellants used by the EDS. As planned for the lunar exploration, the EDS would only be used over the course of a few days to a week, limiting the time for the liquid oxygen and liquid hydrogen to boil-off. In the case of an exploration to 1999 AO10, the time between launch and arrival would surpass 100 days, significantly impacting the availability of the propellants. Further research should be undertaken to identify the boil-off rates for cryogenic propellants over long durations, including identifying enabling technologies to make this mission feasible.

Conclusions

A sortie mission to a near-Earth object such as 1999 AO10 is not feasible without significant modifications to existing Constellation program spacecraft to reduce mass and increase fuel capacity. Aside from concerns with the storage of cryogenic propellants for the 100-plus day transit to the asteroid, the existing element designs do not provide sufficient storage capacity for the additional service module propellant, the crew consumable supplies, packaging and tare mass for supplies, and the scientific equipment needed to perform studies once at the asteroid. With additions such as an expanded in-space habitat for the crew, a dedicated in-space propulsion stage, an airlock to perform EVAs, and a third crew member to assist with EVAs, the mission calls for a different set of spacecraft more in-line with a Martian rather than a lunar exploration.

5.1.3 Lunar Outpost Buildup

An extended lunar exploration building to continuous human presence was one potential goal of NASA's Constellation program. Although the fate of the program is currently uncertain, it still serves as an adequate case study as an example of an exploration campaign with significant use of surface operations. In addition, due to the maturity of the campaign architecture, the modeled exploration benefits from detailed and realistic element models based on data developed with a reasonable amount of analysis.

As of late 2009, the most current iteration of the lunar surface architecture was Scenario 12, developed by the NASA Lunar Surface Systems Project Office (LSSPO) and the Constellation Architecture Team – Lunar (CxAT-Lunar). Scenario 12 has evolved from the confluence of three previous scenarios: Scenario 4 (Optimized Exploration), Scenario 5 (Fission Surface Power System), and Scenario 8 (Initial Extensive

Mobility) [41, 42]. In Scenario 12, successive missions at a rate of about three per year deliver infrastructure components to an outpost, building up to full capability within six years.

One of the interesting aspects of Scenario 12 from an analysis perspective is a high degree of reuse of surface mobility elements for excursions and general exploration. The lunar electric rover (LER) concept is capable of traveling up to 200 kilometers on one charge, but when not exploring, it is attached to the crew habitat to provide private sleeping quarters as well as a safe haven during galactic cosmic radiation (GCR) events. The tri-ATHLETE concept is capable of traveling alone, but when combined with a second tri-ATHLETE, it is capable of traversing difficult terrain while carrying a payload as large as a habitat module.

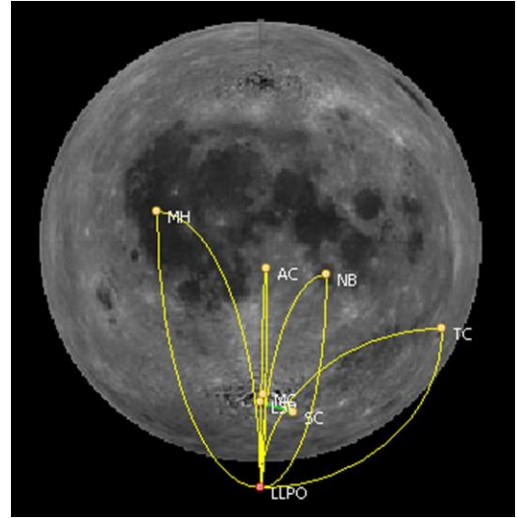


Figure 54: Lunar surface exploration network. Visualization of flight edges to various surface locations and surface edges between excursion sites near LSP.

The focus of this case is to model the surface exploration similar to what is planned under Scenario 12. As the existing scenario is well-researched, this analysis will help validate the modeling framework rather than explicitly evaluating feasibility. Both sortie missions to the surface locations of interest and the build-up of outpost elements at the Lunar South Pole will be modeled. In addition, two excursions from the outpost are modeled in detail, one short-distance excursion to the Malapert crater using two LERs over approximately one week, and one long-distance excursion to the Schrodinger Crater using two ATHLETES over approximately 60 days.

Model Inputs

As the transportation system is well-defined for the Constellation program, the analysis scope will be limited to the lunar system utilizing flight edges for all transports between the lunar surface locations and low-lunar polar orbit (LLPO). The primary landing site on the lunar surface is at the Lunar South Pole (LSP), though sortie missions access other surface sites listed in Table 30.

Table 30: Lunar outpost nodes.

Abbrev.	Description	Parameters
LLPO	Low Lunar Polar Orbit	100 km x 100 km, 90°
LSP	Lunar South Pole (Shackleton Crater)	89.9° S, 0° E
MC	Malapert Crater	85.9° S, 12.9° E
SB	Schrodinger Basin	75° S, 132.4° E
TC	Tsiolkovskiy Crater	20° S, 129° E
AC	Alphonsus Crater	13° S, 2.8° E
MH	Marius Hills	14° N, 56° W
NB	Nectaris Basin (Mare Nectaris)	15.2° S, 35.5° E

This case includes two surface edges to model excursions between LSP and Malapert Crater (MC) and Schrodinger Basin (SB). Other edges model three variations of the descent module (DM) vehicle, a “sortie mode” with a dedicated airlock, an “outpost mode” with no airlock, and a “cargo mode” which has neither an airlock nor an ascent stage.

Table 31: Lunar outpost edges.

Name	Origin	Destination	Length	Capacity
Lunar Descent (Sortie)	LLPO	LSP, TC, AC, MH, NB, OH	1 day	4 crew, 710 kg*
Lunar Descent (Outpost)	LLPO	LSP	1 day	4 crew, 1000 kg
Lunar Descent (Cargo)	LLPO	LSP	1 day	0 crew, 14600 kg
Lunar Ascent to LLPO	LSP, TC, AC, MH, NB, OH	LLPO	1 day	4 crew, 100 kg
Malapert Excursion	LSP	MC	180 km	-
Schrodinger Excursion	LSP	SB	500 km	-

* 500 kilograms baseline + 210 kilograms to support 7-day sortie exploration

Element models focus on the major surface infrastructure elements rather than the launch vehicles and transfer spacecraft because of the usage of abstracted flight edges. The elements of focus include the descent and ascent modules, surface rovers, habitats, and logistics carriers. The descent and ascent modules closely follow the respective flight edges. The sortie descent module includes 210 kilograms of integrated resources to supply a 7-day exploration, modeled as a separate zero-mass sortie consumables resource container added to its contents.

Two ISRU plants, delivered in missions 7 and 15, generate oxygen (generic COS 203) at a rate of 1000 kilograms per year. The production demands are modeled with a linear demand model using a negative

rate (-2.75 kilograms per day). The ISRU plants are activated by the next crew upon delivery and are assumed to operate continuously during crewed and un-crewed periods. All produced oxygen is stored within the ISRU plant until it is demanded by the mission-level crew consumables models.

Table 32: Lunar outpost primary elements.

Name	Dry Mass [kg]	Max Crew	Max Cargo [kg]	Description
SDM	13000	0	710*	Descent Module (Sortie Mode)
SCC	0	-	210	Sortie Consumables Container
CDM	12000	0	14600	Descent Module (Cargo Mode)
ODM	12000	0	1000	Descent Module (Outpost Mode)
AM	3000	4	100	Ascent Module
CUR	230	2	0	Capable Unpressurized Rover
LER	4000	4	1000	Lunar Electric Rover
PUP	650	-	-	Portable Utility Pallet
ISRU	275	-	-	ISRU Plant and Tools
ATH	1200	0	10000	Tri-ATHLETE
PSU	2800	-	-	Power and Support Unit
PEM	6000	4	10000	Pressurized Excursion Module
PCM	7800	4	10000	Pressurized Core Module
PLM	3400	0	17500	Pressurized Logistics Module
FSPS	9500	-	-	Fission Surface Power System

* 500 kilograms baseline + 210 kilograms to support 7-day sortie exploration

Sparing-by-mass demand models are used to model parts and maintenance demands for the major surface infrastructure elements, including the CURs, LERs, PUPs, ISRU plants, Tri-ATHLETEs, PSUs, PEM, PCM, and FSPS. A default rate of 10% mass per year is used during crewed periods, and a 5% mass per year is used during un-crewed periods. Crew members use linear demand models to generate demands for consumables. Average daily surface demands include 2 kilograms of food, 3.5 kilograms of water, 1 kilogram of gases, 0.5 kilogram of hygiene items, and 0.5 kilogram of waste disposal items per crew member. After delivery of the surface habitat in mission 13, the demands for water are decreased to 0.5 kilograms per person per day to account for greater water recovery rates. Demands during ascent, descent, and in-space periods are omitted. Linear mission demand models generate demands for science payload generally, uniquely specified for each mission. Unlike other resources, science payload demands are assumed to include any required packaging mass.

Secondary element models are simplified to only mass estimates without significant surface movement or demand models. Though they only take up supply capacity in this analysis, additional analysis could benefit from the more detailed campaign definition.

Table 33: Lunar outpost secondary elements.

Name	Dry Mass [kg]	Max Crew	Max Cargo [kg]	Description
ALC	400	0	500	Airlock-derived Logistics Carrier
SA	50	-	-	Solar Array
SOD	10	-	-	Small Offloading Device
PCT	170	-	-	Portable Communications Terminal
AAMA	270	-	-	Active-Active Mating Adapter
CB	100	-	-	Chassis Blade
CA	100	-	-	Chassis A (Small Mobile Base)
RA	110	-	-	Robotic Assistant
LSMS	190	-	-	Lunar Outpost Manipulator System
STM	50	-	-	Suit Port Transfer Module
MCT	210	-	-	Mobility Chassis Tool Kit
BT	85	-	-	Battery (9-Pack)
SSU	600	-	-	Structural Support Unit

Missions

The mission manifest is based on concepts included in Scenario 12. Most missions deliver crew and cargo to an outpost at LSP, though several sortie missions explore other sites. The first few missions provide basic exploration capability with unpressurized rovers (CURs). Additional exploration capability is delivered in missions 4 and 6, each delivering two LERs. The Tri-ATHLETE elements are delivered in missions 9 and 12 in preparation of unloading the habitation and logistics elements in later cargo missions.

Elements are reconfigured between states for many of the missions to highlight different operational conditions. While in transit and upon delivery, all elements exist in a dormant state. Upon crew arrival, all primary surface elements are reconfigured to an active state to generate increased demands. The elements are later reconfigured to a quiescent state upon the crew departure. Both CUR surface mobility elements are decommissioned after mission 5, which delivers more capable LER surface mobility elements. The two ISRU and the FSPS elements are not reconfigured to a quiescent state upon crew departure and are assumed to operate continuously.

There are two missions targeted for demonstrating surface transportation and excursions. Mission 7 includes an excursion to Malapert Crater (MC) using one logistics LER to preposition cargo and two crewed LERs. Transit spans three days on both legs of the trip, and exploration runs four days at MC.

Table 34: Lunar outpost missions 1-10.

#	Date	Flight(s)	Element(s)	Description
1	5/7/2021	Descent to LSP	SDM-1, AM-1, CUR-1, SA-1, SOD-1, PCT-1	Unmanned test flight with pre-positioning of some surface infrastructure
2	11/7/2021 11/14/2021	Descent to LSP Ascent to LLPO	SDM-2, AM-2, CUR-2, SA-2, SOD-2	7-day crewed exploration mission (4 EVAs) with 180 kg of science payload
3	4/7/2022 4/14/2022	Descent to MH Ascent to LLPO	SDM-3, AM-3	7-day crewed exploration mission (4 EVAs) with 500 kg of science payload
4	11/7/2022	Descent to LSP	CDM-1, LER-1, LER-2, PUP-1, PUP-2, AAMA-1, RA, CB, CA, LSMS-1, STM-1, BT-1, BT-2	Cargo delivery with 820 kg of science payload
5	2/7/2023 2/21/2023	Descent to LSP Ascent to LLPO	ODM-1, AM-4, MCT-1	14-day crewed exploration mission (8 EVAs) with 660 kg of science payload
6	10/7/2023	Descent to LSP	CDM-2, LER-3, LER-4, AAMA-2, AAMA-3, AAMA-4, LSMS-2, PUP-3, PUP-4, STM-2	Cargo delivery with 710 kg of science payload
7	12/7/2023 12/16/2023 12/23/2023 1/4/2024	Descent to LSP Excursion to MC Return to LSP Ascent to LLPO	ODM-2, AM-5, ISRU-1	28-day crewed exploration mission (16 EVAs) with 190 kg of science payload plus 4-day Malapert excursion
8	3/7/2024	Descent to NB Ascent to LLPO	SDM-4, AM-6	7-day crewed exploration mission (4 EVAs) with 500 kg of science payload
9	10/7/2024	Descent to LSP	CDM-3, ATH-1, ATH-2, PSU-1, AAMA-5, BT-3, BT-4, ALC-1	Cargo delivery with 1,800 kg of science payload
10	11/7/2024 12/3/2024	Descent to LSP Ascent to LLPO	ODM-3, AM-7, PCT-2	28-day crewed exploration mission (16 EVAs) with 320 kg of science payload

Continuous human presence is achieved by mission 18. Mission 20 includes an excursion to Schrodinger Basin (SB) using a “Lunabago” concept, in which two ATHLETE systems carrying the pressurized excursion module (PEM) and a pressurized logistics module (PLM) travel with the crew in two LERs. Surface transport takes 25 days to get to SB, exploration lasts for 14 days, and 45 days are provided for the return surface transport to the outpost.

Table 35: Lunar outpost missions 11-21.

#	Date	Flight(s)	Element(s)	Description
11	3/7/2025	Descent to TC Ascent to LLPO	SDM-5, AM-8	7-day crewed exploration mission (4 EVAs) with 500 kg of science payload
12	10/7/2025	Descent to LSP	CDM-4, ATH-3, PEM, PSU-2	Cargo delivery with 60 kg of science payload
13	11/7/2025 12/28/2025	Descent to LSP Ascent to LLPO	ODM-4, AM-9	50-day crewed exploration mission (4 EVAs/week) with 420 kg of science payload
14	9/7/2026	Descent to AC Ascent to LLPO	SDM-6, AM-10	7-day crewed exploration mission (4 EVAs) with 500 kg of science payload
15	10/7/2026	Descent to LSP	CDM-5, ATH-4, PCM, PSU-3, ISRU-2	Cargo delivery with 0 kg of science payload
16	12/7/2026 3/28/2027	Descent to LSP Ascent to LLPO	ODM-5, AM-11	110-day crewed exploration mission (4 EVAs/week) with 130 kg of science payload
17	2/7/2027	Descent to LSP	CDM-6, AAMA-6, PLM-1, SSU-1	Cargo delivery with 780 kg of science payload
18	7/7/2027 ¼/2028	Descent to LSP Ascent to LLPO	ODM-6, AM-12	180-day crewed exploration mission (4 EVAs/week) with 280 kg of science payload
19	10/7/2027	Descent to LSP	CDM-7, FSPS, ALC-2	Cargo delivery with 980 kg of science payload
20	1/1/2028 1/26/2028 3/5/2028 6/30/2028	Descent to LSP Excursion to SC Return to LSP Ascent to LLPO	ODM-7, AM-13	180-day crewed exploration mission (4 EVAs/week) with 70 kg of science payload
21	5/7/2028	Descent to LSP	CDM-8, PLM-2, SSU-2	Cargo delivery with 1760 kg of science payload

Although there would likely be a dozen or more additional cargo and crewed missions, it would closely resemble the ISS resupply case study and are not included in this analysis.

Analysis and Discussion

As this case study uses abstracted flight edges originating from LLPO rather than modeling the in-space transportation in detail, there are no challenges to the spatial feasibility. The process-based bat chart for the campaign is illustrated in Figure 55.

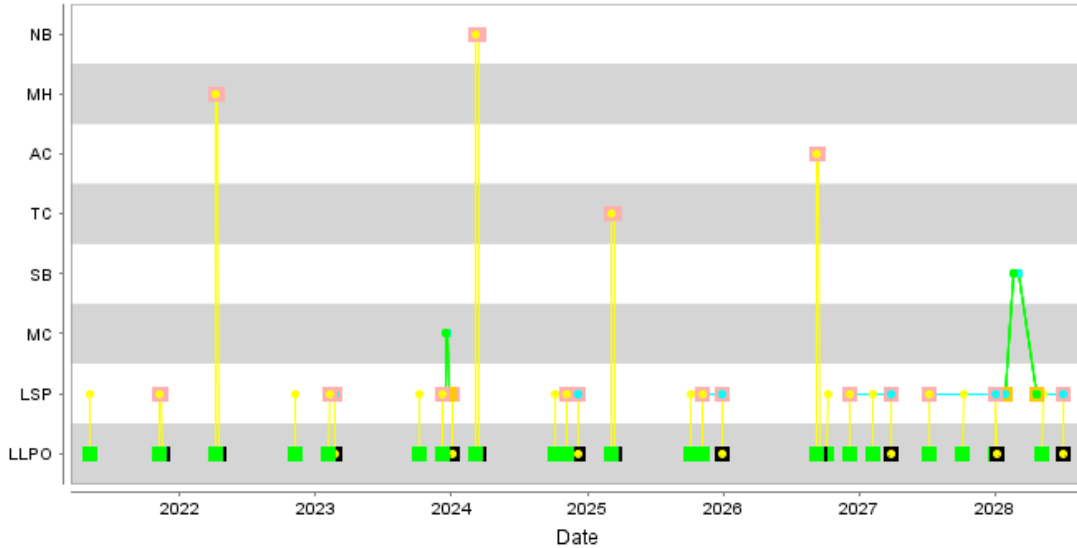


Figure 55: Lunar outpost process bat chart. Green squares are element instantiations, yellow lines are flight transports, green lines are surface transports, orange and pink squares are element movements and reconfigurations, blue lines are surface explorations, and black squares are elements removed from simulation.

Demands for each of the sortie missions (4, 8, 11, and 14) can be satisfied by the integrated resources in the SDM as the exploration duration does not exceed seven days. Logistical analysis is therefore focused at LSP starting with mission 5, the first surface exploration longer than seven days. Figure 56 shows the feasibility at LSP given the raw and remaining capacity of landers (after infrastructure elements) and demands for crew consumables, science payloads, and spare parts. Packing or logistics container masses are included for demands using packing factors of 50% for water, 100% for gases, and 120% for all other non-science pressurized items. As expected, the campaign appears logistically feasible.

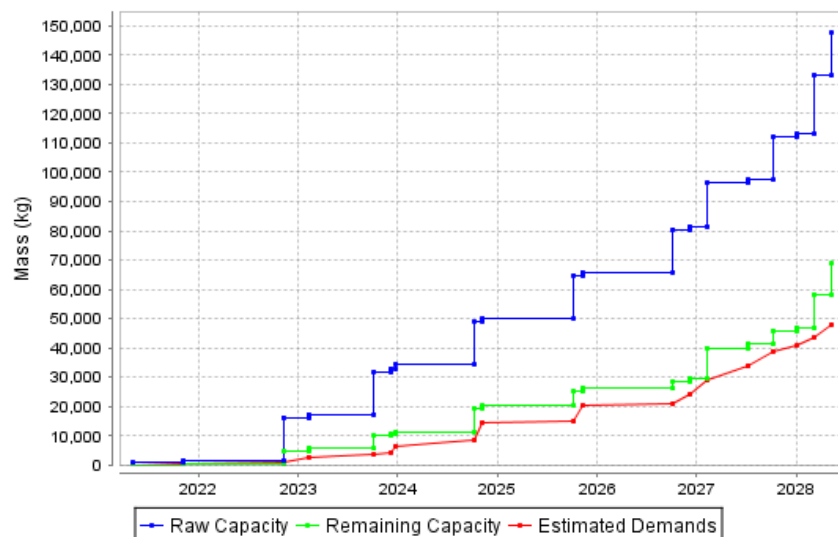


Figure 56: Lunar outpost feasibility at LSP. Demands include science payloads, crew consumables and spares with packing mass estimates included.

Using other visualizations, the breakdown of demand classification can be closely analyzed. Figure 57 highlights the breakdown of demands at LSP by class of supply. Crew consumables (COS 2) show the highest demand mass, exceeding 20 tons by 2028. The second largest class of supply, spares and maintenance (COS 4), increases rapidly as the infrastructure mass at the outpost is accumulated. Close inspection also illustrates the change in spares rates corresponding to crewed and un-crewed periods.

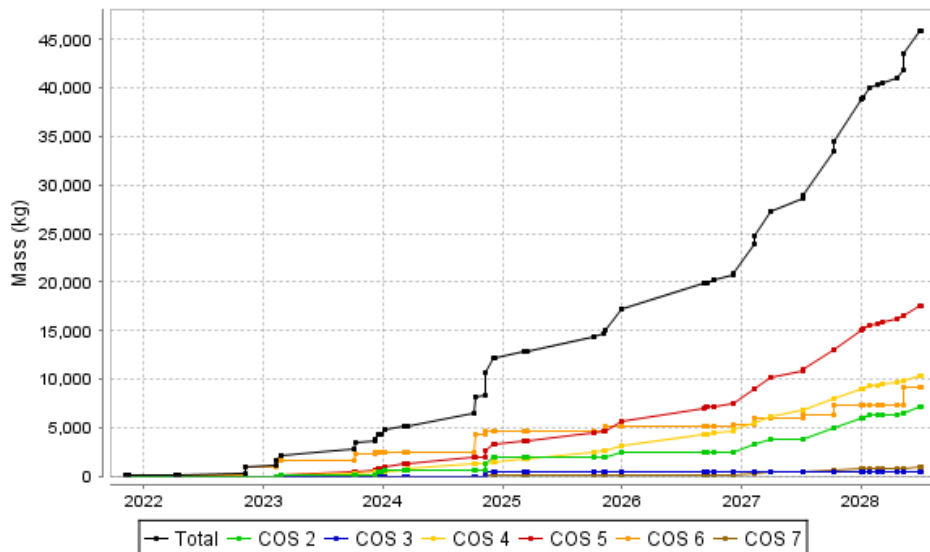


Figure 57: Lunar outpost demands at LSP. COS 2: Crew Consumables, COS 3: Operational Items, COS 4: Maintenance and Spares, COS 5: Packing Mass (Estimated), COS 6: Science Payload, COS 7: Waste Disposal.

The demands include the effects of consumption of oxygen produced by the two ISRU plants delivered in mission 7 and 15. Figure 58 shows the simulated results of oxygen produced by ISRU-1 and ISRU-2 available for consumption during the exploration.

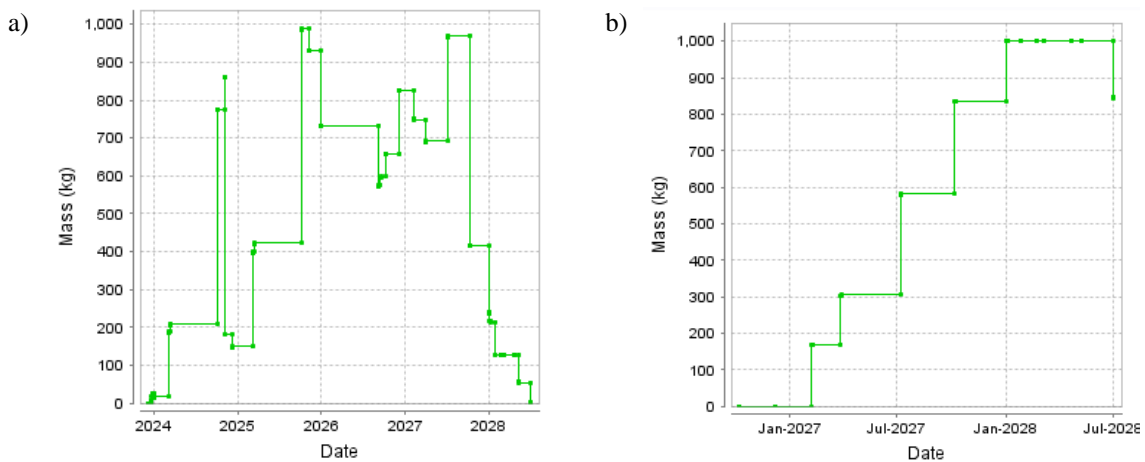


Figure 58: Lunar outpost ISRU production and consumption. Mass of oxygen available for consumption from (a) ISRU-1 and (b) ISRU-2.

The simulation results also provide several measures of effectiveness (MOEs) to quantify the exploration campaign. There are a total of 2,500 crew surface days across all sites including sorties and excursions to Malapert and Schrodinger craters. The total exploration capability, the dot product of crew and enabling infrastructure (COS 6 and COS 8) is 117.5 million crew-kg-days. The exploration capability greatly increases towards the end of the modeled campaign because of the substantial delivered infrastructure. Other MOEs including relative exploration capability and total launch mass are not computable in this case because launch and in-space portions of this campaign were not considered.

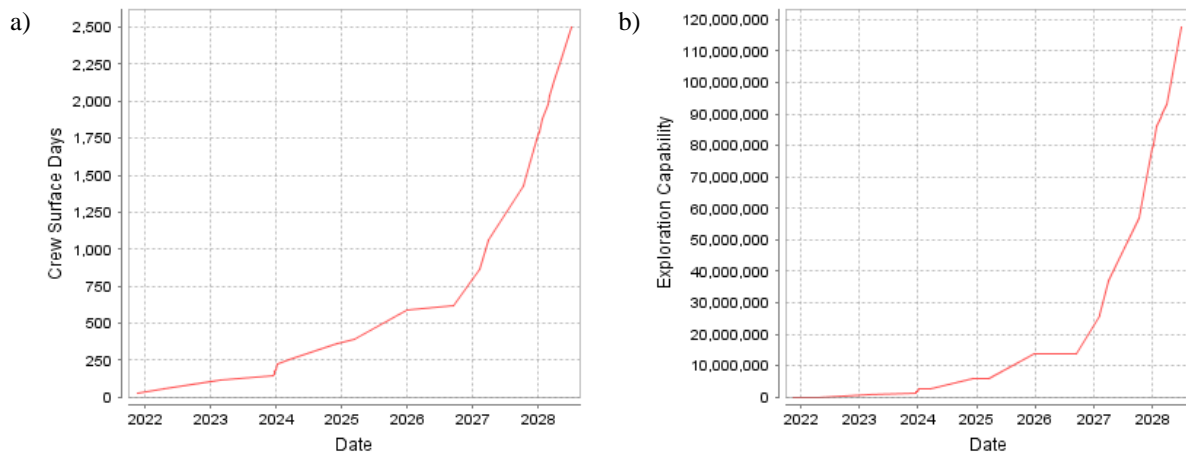


Figure 59: Lunar outpost measures of effectiveness. (a) Crew surface days across all surface sites. (b) Exploration capability (crew-kg-days) across all surface sites.

Conclusions

This case study modeled an extended lunar surface exploration campaign based on existing architectural studies. Modeling details include ISRU oxygen production, dynamic spares rates for crewed versus uncrewed periods, surface transportation for excursions, and improved water recovery rates in crew habitats. As expected with a matured design, the aggregated demands for crew consumables and spares and maintenance show it is a logistically feasible campaign.

Additional analysis for a lunar surface exploration campaign should inspect the excursions to Malapert Crater and Schrodinger Basin in more detail. In particular, the feasibility of transferring resources between the outpost and remote excursion sites may not be trivial to evaluate. Multi-transport manifesting methods are required manifest resources necessary to support the crew on long duration excursions.

5.1.4 Mars Exploration

Mars has been a highly sought-after destination for human spaceflight since the early days of rocketry. The Vision for Space Exploration announced in 2004 named Mars as one of the ultimate destinations for which precursor missions to the lunar surface would help prepare [3]. The final report of the Human Spaceflight Review Committee also states that “a human landing followed by an extended human presence on Mars stands prominently above all other opportunities for exploration” [18]. Although the Constellation Program is facing cancellation by the new US and NASA administration, human missions to Mars still hold the focus of space exploration, albeit with new technologies to provide advanced propulsion and protection against the harsh radiation environment in transit [20].

NASA has undertaken substantial effort to design reference architectures for conceptual missions to Mars, the most recent publication being Mars Design Reference Architecture 5.0 (DRA 5.0), published in July 2009 [21]. This design reference describes the spacecraft and missions which could be used for the first three excursions to the surface of Mars. The Mars exploration architecture is heavily based off of lunar concepts from the Constellation Program, including the Ares V heavy lift launch vehicle, but also includes advanced technology concepts such as nuclear thermal rockets (NTRs) for in-space propulsion, zero-loss cryogenic coolers for propellant transportation, and nuclear fission reactors for surface power.

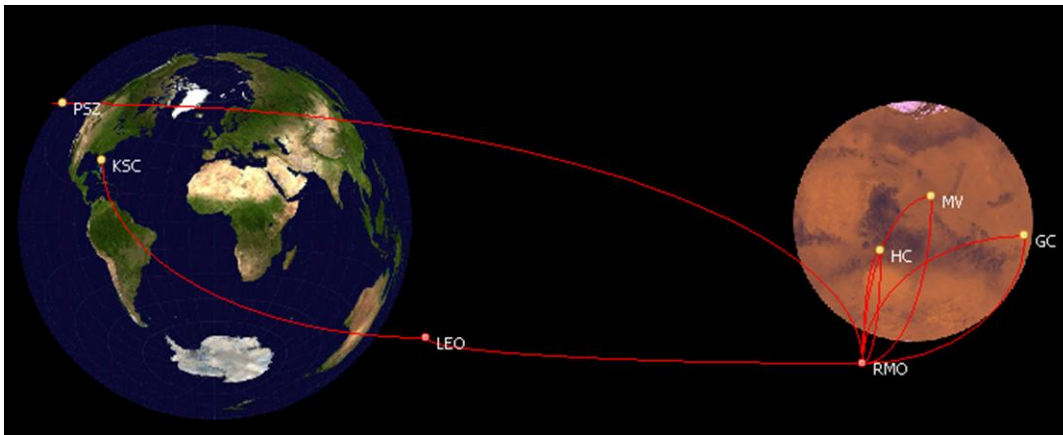


Figure 60: Mars exploration network. Visualization of space edges utilizing propulsive burns to reach potential Martian surface sites.

This case study is focused on determining the in-space propulsive feasibility to deliver surface elements required by DRA 5.0 and identifying driving factors to manage logistics feasibility for a crew of six. Surface operations are not modeled in detail, though there would be significant interest in future analysis to study two-week surface excursions using pressurized rovers.

Model Inputs

The nodes for the case study include the Kennedy Space Center for launches, a splashdown site in the Pacific Ocean, a low-Earth parking orbit for assembly of in-space vehicles, a reference Mars orbit for stationary operations on orbit, and three target surface exploration sites. Although only one exploration site, Mawrth Vallis, is used in this preliminary analysis, there would likely be at least three similar missions to justify the large investments in new spacecraft designs and propulsion technology.

Table 36: Mars exploration nodes.

Abbrev.	Description	Parameters
KSC	Kennedy Space Center	28.6° N, 80.6° W
PSZ	Pacific Splashdown Zone	15° N, 160° W
LEO	Low Earth Checkout Orbit	407 km x 407 km, 28.5°
RMO	Reference Mars Orbit	250 km x 33793 km, 70°
MV	Mawrth Vallis	24° N, 19° E
GC	Gale Crater	4.6° S, 137.2° E
HC	Holden Crater Fan	26.4° S, 34.7° W

The edges used in this case study are a mix of both space edges using propulsive burns and flight edges providing an abstracted path between nodes. Space edges model the launches of the Ares V rockets from KSC to LEO, as well as the in-space transfers to Mars orbit and descent to the Martian surface. Ascent from the Martian surface is modeled as a flight to simplify modeling efforts for the Mars Ascent Vehicle (MAV), which receives LOX propellant via ISRU production. The launch of the crew exploration vehicle (CEV) from KSC is also modeled as a flight to provide independence from any particular launch vehicle.

Table 37: Mars exploration edges

Name	Origin	Destination	Duration [days]	Propulsive Burns or Flight Capacity
KSC Launch to LEO	KSC	LEO	0.1	t+0: 9.8 km/s
CEV Launch to LEO (Flight)	KSC	LEO	0.1	6 crew, 100 kg
TMI/ MOI (Aerocapture)	LEO	RMO	202	t+0: 3.7 km/s
TMI,/MOI (All Propulsive)	LEO	RMO	174	t+0: 4.1 km/s t+174: 1.7 km/s
Martian Descent	RMO	MV	0.1	t+0.0: 0.015 km/s t+0.1: 0.595 km/s
Martian Ascent (Flight)	MV	RMO	0.1	6 crew, 250 kg
Trans-Earth Injection, Splashdown at PSZ	RMO	PSZ	201	t+0: 2.25 km/s t+200: 0.15 km/s

Elements modeled include the Ares V launch vehicle and the in-space vehicles and landers for Martian exploration. Several of the Ares V launches use a modified payload fairing which serves as an aeroshell for aerocapture and entry into the Martian atmosphere. In-space Mars transfer vehicles (MTVs) are assembled in low-Earth orbit. Two un-crewed MTVs pre-position the cargo lander on the Martian surface and the habitat lander in Martian orbit. The crewed MTV carries the six crew members and the Mars transfer habitat (MTH) to rendezvous with the habitat lander before descent and surface operations.

Table 38: Mars exploration elements

Name	Mass [kg]	Max Crew	Max Cargo [kg]	Fuel [kg] (Type)	Isp [s]	Description
Ares V SRBs	106,500	0	0	685,000 PBAN	26	Ares V Solid Rocket Boosters (2)
Ares V Core	173,680	0	0	1,587,000 LOX/LH2	414	Ares V Core
Ares V Interstage	9,190	-	-	-	-	Ares V Interstage Element
Ares V EDS	26,390	0	0	253,000 LOX/LH2	449	Ares V Earth Departure Stage
Ares V PLF	9,049	-	-	-	-	Ares V Payload Fairing
NTR	37,300	0	0	59400 LH2	950	Nuclear Thermal Rocket
Inline LH2 Tank (Cargo MTV)	10,800	0	34,100	-	-	LH2 Tank for Cargo Mars Transfer Vehicle
MDAV	25,780	0	5,500	10,600 LOX/LCH4	369	Mars Descent / Ascent Vehicle (Cargo Lander)
MAV	21,500	6	300	-	-	Mars Ascent Vehicle (Abstracted)
Aeroshell	42,900	0	0	-	-	Dual-use Aeroshell Shroud
SHAB	52,060	6	1,500	10,600 LOX/LCH4	369	Surface Habitat (Habitat Lander)
NTR-S	46,600	0	0	59,700 LH2	950	Nuclear Thermal Rocket Stage with External Radiation Shield
Inline LH2 Tank (Crewed MTV)	21,500	0	69,900	-	-	LH2 Tank for Crewed Mars Transfer Vehicle
LST	8,900	-	-	-	-	Long Saddle Truss
LH2 Drop Tank	14,000	0	73,100	-	-	LH2 Drop Tank
SST	8,900	-	-	-	-	Short Saddle Truss
CFC	1,860	-	7,940	-	-	Contingency Food Canister
DM	1,800	-	-	-	-	Second Docking Module
MTH	27,540	6	5,300	-	-	Mars Transit Habitat
CEV	14,000	6	250	1000 MMH/N2O4	301	Crew Exploration Vehicle

As additional mass for spares is included in element mass estimates from DRA 5.0, demands primarily originate from the crew using a linear demand model with two operational states. While in transit, the crew demands total 7.5 kilograms per crew member daily. The approximate breakdown is 2 kilograms for food, 3.5 kilograms for water, 1 kilogram for gases, 0.5 kilograms for hygiene items, and 0.5 kilograms for waste disposal. While on the surface, it is assumed that the ISRU plants can provide ample oxygen to eliminate the demand for additional gases resulting in demands of 6.5 kilograms per crew member daily.

Missions

Similar to the NEO sortie, this case study only inspects a single mission to Mars, though in practice there would be at least three human missions planned in succession. Modeling more than one mission would provide benefits of element reuse between sites under the assumption that the exploration sites were close enough for automated surface transportation.

A single Mars exploration mission is divided across two transit windows approximately 26 months apart. The first five launches of the heavy lift vehicle provide the elements to assemble the two cargo MTVs in low-Earth orbit. The first cargo MTV contains the cargo lander and the second the habitat lander. Once both vehicles are constructed, they depart for Mars using a trans-Mars injection burn and are aerocaptured 200 days later. The cargo lander subsequently descends to the Martian surface to commence oxygen production while the habitat lander remains in orbit for the arrival of the crew.

The second four launches of the heavy lift vehicle provide the elements to assemble the crewed MTV in low-Earth orbit. Once constructed, the crew of six is delivered using a human-rated launch vehicle and the MTV departs for Mars, jettisoning an empty drop tank after the TMI burn and arriving with a Mars orbit insertion (MOI) burn 170 days later. Once in Mars orbit, the crewed MTV docks with the habitat lander and the crew descend to the Martian surface to perform surface operations for 530 days while the crewed MTV remains in Mars orbit. In contingencies preventing the landing of the crew, such as a failure of the descent/ascent vehicle, a contingency food canister (CFC) provides necessary resources for the crew in Mars orbit until the return launch window opens.

After the exploration, the crew and surface samples use the fueled ascent vehicle to return to Mars orbit and dock with the crewed MTV. The contingency food canister is jettisoned prior to the Earth orbit injection (EOI) burn and the crew returns to Earth using a CEV for the final re-entry and splashdown into the Pacific Ocean.

Table 39: Mars exploration mission events.

Date(s)	Event	Details
1/1/2035	Cargo MTVs Launch 1	Payload: NTR #1
2/1/2035	Cargo MTVs Launch 2	Payload: NTR #2
3/1/2035	Cargo MTVs Launch 3	Payload: Cargo Inline LH2 Tanks #1 and #2
4/1/2035	Cargo MTVs Launch 4	Payload: Cargo Lander (MDAV, MAV, Aeroshell)
5/1/2035	Cargo MTVs Launch 5	Payload: Habitat Lander (SHAB, Aeroshell)
6/27/2035	Cargo Lander TMI	Aerocaptured on 1/15/2036
7/4/2035	Habitat Lander TMI	Aerocaptured on 1/22/2036
2/6/2036	Cargo Lander Descent	Landing at Mawrth Vallis
4/1/2037	Crewed MTV Launch 1	Payload: NTR-S
5/1/2037	Crewed MTV Launch 2	Payload: Inline LH2 Tank
6/1/2037	Crewed MTV Launch 3	Payload: LTS, LH2 Drop Tank
7/1/2037	Crewed MTV Launch 4	Payload: SST, CTC, DM, MTH, CEV
8/22/2037	Crew Launch (Flight)	Payload: CEV (6 Crew)
8/27/2037	Crewed MTV TMI	Propulsive MOI on 2/17/2038
2/19/2038	Habitat Lander Descent	Landing at Mawrth Vallis
8/5/2039	MAV Ascent (Flight)	Payload: MAV (6 Crew, Samples)
8/10/2039	Crewed MTV TEI	Pacific Splashdown on 2/27/2040

Analysis and Discussion

The first phase of analysis focuses on determining the propulsive feasibility of the campaign. All propulsive mission transports are modeled with the exception of the crewed transport to LEO and the Martian ascent, both abstracted with flights. In addition, notional resource containers are used to hold the maximum cargo capacity for each of the carrier elements to provide a reasonable replication of the manifesting process. For example, a mass-less resource container packed with 4,500 kilograms of consumables is created inside the MDAV.

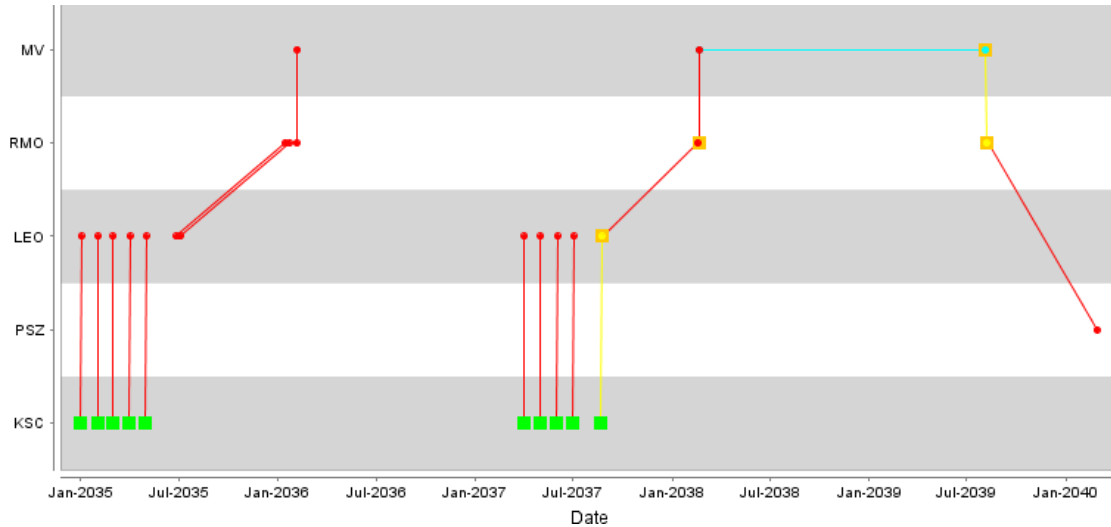


Figure 61: Mars exploration process bat chart. Green squares are element instantiations, red lines are space transports, yellow lines are flight transports, orange squares are element movements and reconfigurations, and the blue line is a surface exploration.

All space transports in the Martian exploration are found to be propulsively feasible with baseline element properties. The propellant margins for descent, however, are especially tight. The cargo lander has a propellant margin of 627 kilograms (5.9%) and the habitat lander a margin of 435 kilogram (4.1%).

Since the in-space propulsion depends on advanced nuclear thermal technology, a sensitivity analysis was performed on the specific impulse of the rocket, shown in Table 40. For a specific impulse of 850 seconds, the low end of the ranges cited in DRA 5.0, the transportation system becomes infeasible, with a residual delta-v of 160 meters per second for the in-space burn. This result indicates the performance of the NTR has a large impact on the performance of the existing architecture.

Table 40: Nuclear thermal rocket specific impulse sensitivity.

Element Group	LH2 Fuel Capacity [kg]	Remaining Fuel [kg]		
		$I_{sp} = 950 \text{ s}$	$I_{sp} = 900 \text{ s}$	$I_{sp} = 875 \text{ s}$
Cargo MTV #1 (Cargo Lander)	93,500	14,100	10,600	8,700
Cargo MTV #2 (Habitat Lander)	93,500	13,900	10,300	8,400
Crewed MTV	202,700	4,900	1,500	(infeasible)

Though the mission appears propulsively feasible, the default crew demand rates exceed the available capacity for the transportation system. Estimated demands, shown in Figure 62, are significantly larger than the 2,650 kilogram allowance for each of the transports to and from Mars and the 7,000 kilograms allowance for surface operations specified in DRA 5.0.

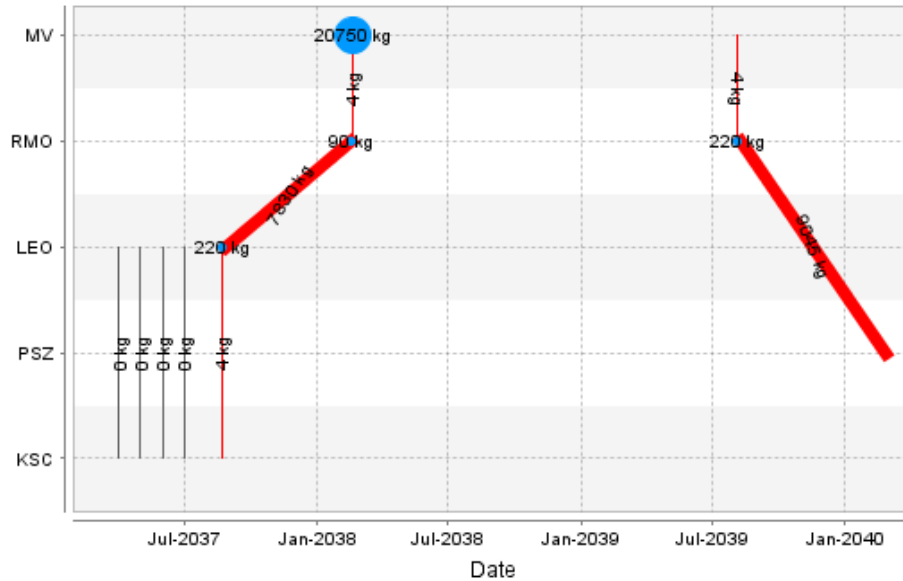


Figure 62: Baseline Mars exploration crew demands. Under baseline crew member demand rates with a crew of six, demands total 8.1 tons during transport to Mars, 20.8 tons during surface operations, and 9.3 tons during return transport.

This is, in part, because the baseline demand rates do not take into account closed-loop environmental controls and life support systems (ECLSS) which could significantly reduce the demands for water and waste disposal resources. Assuming a 95% water closure rate and increased usage of reusable hygiene and waste disposal items, crew demand rates are decreased from 7.5 to 3.375 kilograms per person per day while in transit and from 6.5 to 2.375 kilograms per person per day while on the surface.

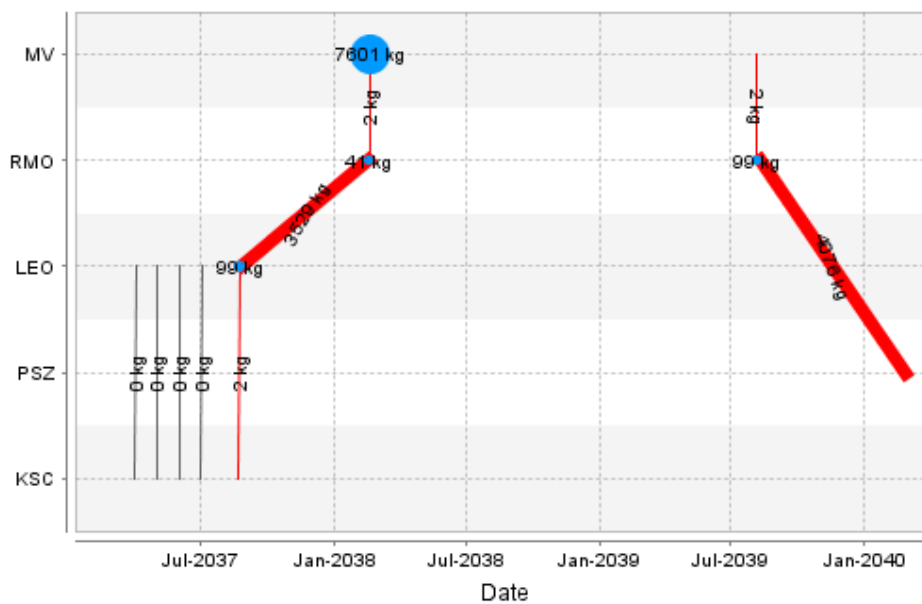


Figure 63: Modified Mars exploration crew demands. Using a 95% water closure loop and reusable hygiene and waste disposal items, demands are reduced to 3.6 tons during transport to Mars, 7.6 tons during surface operations, and 4.1 tons during return transport.

The consumables capacities in DRA 5.0 cannot support the modified demand rates either during transport or during surface operations, even without taking into account the tare mass of any resource containers that would be needed. To close the logistics loop for consumables, design changes are suggested to the MTH for transport and the SHAB and MDAV for surface demands.

Although the delivery of the MTH and its components to LEO is not closely-constrained with the Ares V launch vehicle, the crewed MTV is a bottleneck of the in-space transports. As the crewed MTV cannot support additional resources for consumption during the infeasible transport, a “creative” solution is sought using excess capacity on the cargo MTVs.

Current designs of the MTH use a contingency food canister (CFC) to supply sufficient consumables to sustain the crew until the TEI launch window opens if all or part of the Martian surface operations were aborted. If a secondary CFC were carried aboard either of the cargo MTV transports for use in the case of an emergency in Mars orbit, the first could be used to satisfy in-space transport demands. This solution assumes that consumables could safely be stored in a CFC for up to five years, a secondary CFC could be manifested on a cargo MTV launch, the primary CFC would be accessible during transit to Mars, and the secondary CFC could be accessible in Mars orbit.

A sample implementation of this solution includes a secondary CFC for the cargo MTV launch 3, currently the least mass-constrained. The secondary CFC is coupled with the habitat lander MTV for transport and docked with the crewed MTV in orbit in advance of the surface exploration. As before, all required resources are transferred to the MTH and both CFCs are jettisoned before the TEI transport.

To support the additional demands during surface exploration, either the MDAV or the SHAB must have an increased capacity. If the MDAV resource capacity is increased from 5,500 to 6,500 kilograms, the cargo lander propellant margin is reduced to 465 kilograms (4.4%). If the SHAB consumables capacity is increased from 1,500 to 2,000 kilograms, the habitat lander propellant margin is reduced to 465 kilograms (3.2%). Since the descent stages have narrow propellant margins for both landers, the additional capacity may very well come at the cost of scientific and exploration equipment.

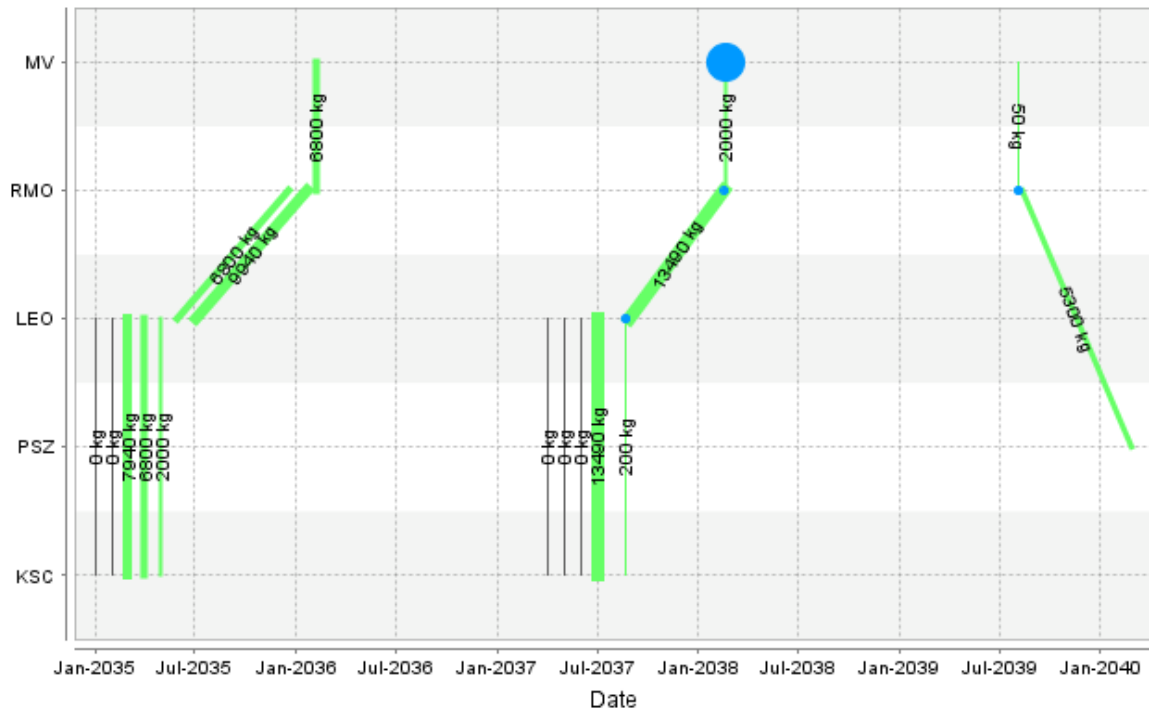


Figure 64: Modified Mars exploration transport capacity. An additional CFC is included to satisfy demands during transport and the cargo capacity for the MDAV and SHAB have been increased to satisfy demands during surface operations.

Conclusions

The Mars exploration architecture as described by DRA 5.0 is propulsively feasible as specified and logistically feasible with the addition of aggressive water recovery and some alterations to increase supply capacity. The most constrained transportation legs include the SHAB and MDAV descents to the Martian surface and the crewed MTV transit to Mars. Although this analysis focused on the feasibility of a nuclear thermal rocket propulsion option, a similar analysis could inspect a chemical propulsion option as well.

The next step of analysis could model the surface operations in much more detail. Currently, the cargo lander and habitat lander are modeled as single elements; however they are actually comprised of many components including pressurized, unpressurized, and robotic rovers, science equipment, stationary power systems, and in situ resource generation plants. Modeling these elements separately could provide much more detailed information for surface logistics, including the accumulation of ISRU resources and the option of analyzing spare parts demands on a per-element basis. Of particular interest is a two-week excursion in which two crew members use a pressurized rover to travel upwards of 100 kilometers at speeds of three kilometers per second.

5.2 Usability Experimentation

In addition to demonstrating the modeling capabilities of the SpaceNet modeling framework with case studies, a parallel task highlights its usability. Substantial effort was placed into the SpaceNet GUI to provide an intuitive and efficient interface for modeling and simulating space exploration campaigns.

A user experiment was designed to quantitatively evaluate SpaceNet 2.5 usability. The primary goal of the study is to measure the time required to model, evaluate, and resolve feasibility for a relevant space exploration scenario. It is hypothesized that SpaceNet enables faster analysis of exploration campaigns, even for users with little experience, over independent spreadsheet-based analysis techniques. To this end, a mission based on the NEO Sortie case study (see Section 5.1.2) was developed for users to implement and analyze. The scenario as specified is not feasible, so users must determine what changes are necessary to establish propulsive and logistical feasibility. The entire scenario is designed to take 30-60 minutes to solve and there is no single solution.

5.2.1 Testing Procedure

Seven volunteer subjects were selected to participate in the usability testing. All volunteers are students in aerospace engineering familiar with general concepts of space systems including specific impulse, rocket staging, and in-space transportation methods. None have previous experience using SpaceNet; though several have previously experience in modeling spacecraft and exploration using other methods. Five of the users serve as treatment subjects using SpaceNet to perform the analysis and two serve as a control subjects using independent spreadsheet analysis techniques.

Table 41: User experiment subject comparison.

	Treatment Subjects	Control Subjects
Quantity	5	2
Selected From	Astronautics students	Astronautics students with experience in spacecraft modeling and architecture
Guided Tutorial	7-day lunar sortie mission modeled in SpaceNet	7-day lunar sortie mission modeled in a spreadsheet
Task	Evaluate feasibility of a 14-day NEO exploration	Evaluate feasibility of a 14-day NEO exploration
Analysis Tools	SpaceNet	Spreadsheet

Treatment Subjects Procedure

Treatment subjects are tested one at a time using a computer using only the SpaceNet 2.5 application. Though the goal of the experiment is to measure usability, it is not designed to measure first-time learning or discovery. A 30-minute guided tutorial, derived from the “quick start” tutorials in user documentation, serves as an orientation to the SpaceNet modeling framework and application. The subject walks through a 7-day lunar surface sortie mission using Constellation-class spacecraft to learn the concepts of element instantiation, space transports and burn sequences, demand generation, and cargo manifesting.

After the tutorial session, the subjects receive with a written summary of the test scenario including mission objectives, element specifications and constraints for modification, and a spreadsheet database containing initial object definitions compatible with SpaceNet. The subject must model the exploration from scratch using the existing object definitions, evaluate feasibility, and iterate on element designs to establish feasibility. The subject may ask for help on SpaceNet functions and spacecraft design concepts during the trial but makes decisions and performs analysis independently. The test concludes when a feasible solution has been found. Finally, the subject completes a post-testing survey to provide feedback on the testing experience.

Control Subjects Procedure

Control subjects perform analysis using independent spreadsheet techniques rather than SpaceNet. As the process to evaluate feasibility without a tool requires deeper understanding technical details, control subjects were selected based on experience with similar space system architecture problems. Similar to the treatment group, control subjects receive a sample spreadsheet-based analysis of the lunar sortie tutorial used to prepare the experimental subjects as an introduction to valid techniques. Both propulsive and logistical feasibility analyses are included to guide the required analysis for the test scenario.

During the trial, control subjects receive the same written task summary of the test scenario and a spreadsheet database containing the initial element specifications. The control subject must model the exploration from scratch using any methods or tools he or she deems useful within spreadsheets, evaluate feasibility, and iterate element definitions to find a feasible design. After finding a feasible campaign, the test subjects are asked to comment on the analysis process, including methods used, difficulties incurred.

Test Scenario Description

The goal of the scenario is to evaluate the feasibility of a 14-day exploration to near-Earth asteroid 1999-AO10 in 2025 using two crew members and modified Constellation-class spacecraft. Demands for resources are generated by the crew members, each requiring 7.5 kilograms of generic crew provisions

per day. The long-duration transports required to reach 1999-AO10 require significant resources to satisfy crew demands, a new challenge not covered in the tutorial scenario.

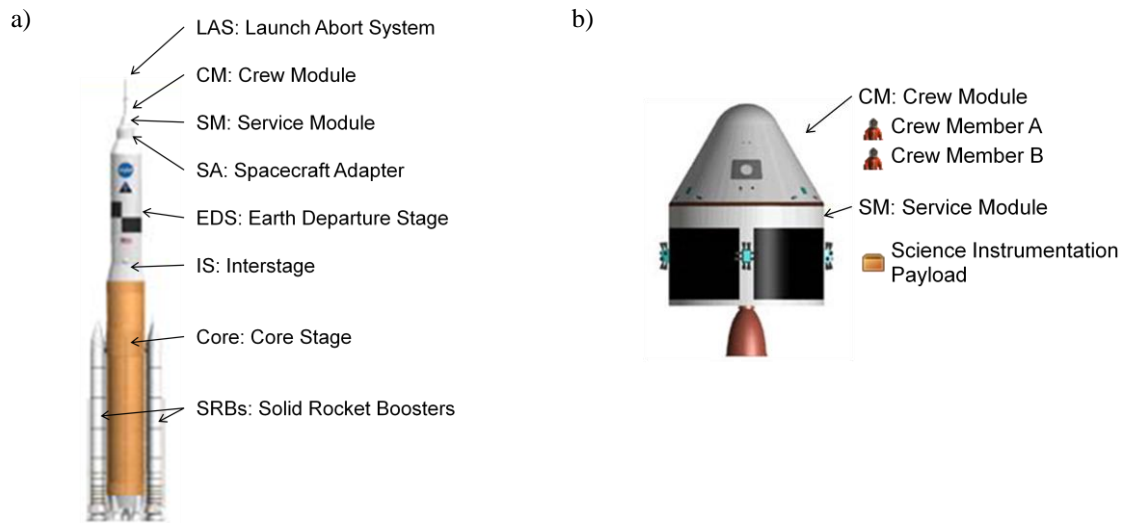


Figure 65: User experiment baseline spacecraft. (a) *Modified Ares V launch vehicle.* (b) *Modified Orion crew exploration vehicle.*

The initial scenario definition neither has sufficient cargo payload for manifested resources nor sufficient propellant to achieve the trans-Earth injection burn. Modifications must be proposed to the elements to overcome the infeasibilities. The following constraints are placed on the user's decisions:

1. No new elements may be instantiated
2. The launch vehicle architecture cannot be altered (i.e. burn sequence)
3. The fuel type and specific impulse may not be changed for propulsive elements
4. No element dry mass may be reduced by more than 10% from the baseline
5. No fuel amount may be increased by more than 20% from the baseline

Users are instructed to use engineering judgment to limit the number of modifications required for feasibility, also considering providing adequate propellant margins for contingencies and retaining science payload mass for performing exploration.

5.2.2 Results and Discussion

The results for the experiment are divided into the test subjects and the control subjects, followed by a comparison discussion.

Treatment Subject Results

All five SpaceNet subjects found a feasible campaign in less than 45 minutes. The total task times were 37, 35, 41, 32, and 32 minutes, resulting in a median task time of 35 minutes. In addition to the time required to find a feasible campaign, the time to complete all event definitions and the time of propulsive feasibility were logged, summarized with a box plot in Figure 66. In general, users first solved the propulsive feasibility problem before investigating the logistical feasibility problem. Some users iterated between the two problems due to insufficient propellant margin to accommodate required resources.

Each user had a slightly different sequence to find a feasible solution. For example, user 3 decreased all available element masses before increasing fuel capacity. User 1 did not modify the launch vehicles while all others made at least one change to the SRBs or Core elements. Two users, 2 and 5, utilized the EDS for the first burn of the trans-Earth Injection (TEI) space transport. Table 42 lists the remaining propellant margins and Figure 66 details the final feasible campaign designs developed by each user.

Table 42: Propulsive fuel margins.

User	EDS Fuel Margin [kg]	SM Fuel Margin [kg]
1	479	379
2	-	796
3	1814	504
4	1696	495
5	-	2156

In general, user feedback after the experiment was positive. Many users were enthusiastic that little prior knowledge of space exploration systems was required to perform a basic analysis. Table 43 summarizes the post-test questionnaire providing insight to the background and experiences of each user.

Table 43: Post-test questionnaire results.

<i>Ratings: 1: Strongly Disagree, 2: Disagree, 3: Neutral, 4: Agree, 5: Strongly Agree</i>	User 1	User 2	User 3	User 4	User 5	Mean
Familiar with Constellation spacecraft architecture	3	4	4	5	5	4.2
Familiar with space exploration modeling/analysis	4	4	5	5	4	4.4
Challenged to find a feasible campaign	3	4	4	4	3	3.6
SpaceNet identified propulsive and logical infeasibilities	5	5	3	5	5	4.6
SpaceNet helped construct a feasible campaign	5	5	5	4	5	4.8
Could have used another tool to perform equivalent analysis	2	2	4	2	2	2.4
Would use SpaceNet in future analyses	5	4	3	4	5	4.2

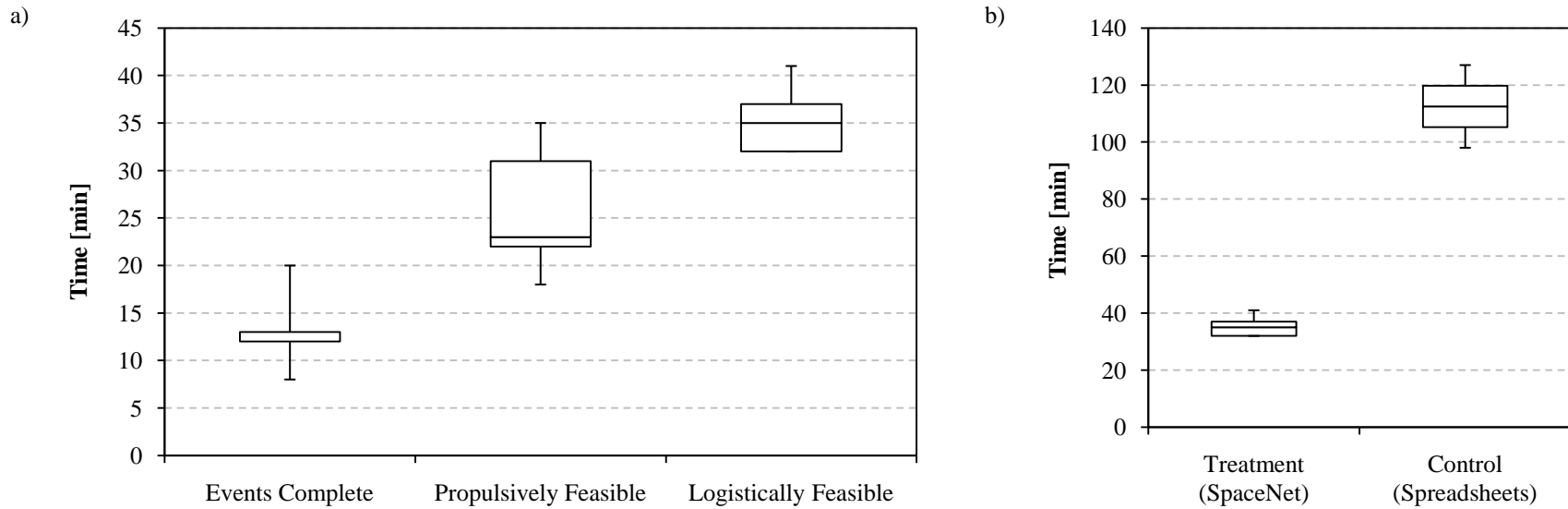


Figure 66: User experiment results. (a) *SpaceNet* subject task milestones. (b) Comparison of treatment (*SpaceNet*) and control (*spreadsheet*) subjects' total task time. Whiskers show minimum and maximum recorded values.

Table 44: User experiment feasible designs.

User	SRBs		Core		IS	EDS		SA	SM			CM		LAS	Crew	EDS TEI Burn
	Mass	Fuel	Mass	Fuel	Mass	Mass	Fuel	Mass	Mass	Fuel	Cargo	Mass	Cargo	Mass	Mass	
1	-	-	-	-	-	-10%	+20%	-	-10%	+20%	-	-10%	+67%	-	-	-
2	-	-	-	+20%	-	-	+20%	-	-	+20%	-	-	+67%	-	-	Yes
3	-10%	-	-10%	+20%	-10%	-10%	+20%	-10%	-10%	+20%	-	-10%	+156%	-10%	-10%	-
4	-10%	+20%	-	+20%	-10%	-	-	-	-10%	+20%	-	-10%	+67%	-10%	-	-
5	-10%	+20%	-	-	-	-	+20%	-	-	-	-	-10%	+68%	-	-	Yes
6*	-	-	-	+15%	-	-	+15%	-	-	+20%	+80%	-5%	+20%	-	-	-
7*	-	-	-	+20%	-	-	+19%	-	-	+14%	-	-	+63%	-	-	-

* Control Subjects

Control Subject Results

The two control subjects logged 127 and 98 minutes to complete the task. In both cases, the scenario was modeled using a sequence of events in rows with formulas to track stack mass, demands, and propellant. A substantial portion of the time (96 and 73 minutes respectively) was devoted to baseline scenario modeling before any analysis was performed. In one of the cases, there were also a few rework loops where an error was detected, requiring time to uncover it and correct the scenario.

The analysis process was very similar to that of the test subjects, first the scenario was modeled as a whole, next element definitions were modified to accommodate the resource supplies and achieve the required delta-v. The only modeling difference from the test subjects was a simplifying assumption that the logistics containers are jettisoned before each propulsive burn.

5.2.3 Discussion

Other than the time required to perform analysis, there were no significant differences between the treatment and the control subjects' feasible solutions. The treatment subjects completed the analysis and provided a feasible solution in an average of 35 minutes, several times faster than the control subjects. Perhaps more importantly, the test subjects modeled the initial scenario in an average of 12 minutes, about 15% of the average time for control subjects to reach a baseline model without including errors uncovered later. This is important for users who may not have experience with modeling exploration campaigns, as creating custom models in spreadsheets can require significant research, modeling considerations, and error tracking.

All experiment subjects enjoyed working through the scenario, both using SpaceNet and with independent analysis tools. It is an important realization that the analysis required for verifying logistical feasibility of conceptual space explorations is not seen as drudgery to be avoided. As the modeling framework is flexible to analyze a wide range of general space exploration campaigns, a tool that does not require significant background research or time may capture non-traditional users' interests, leading to more researched campaign proposals.

6 Conclusions

The refinement of the space logistics modeling framework and implementation in SpaceNet provides a novel and useful simulation and modeling tool. Care has been taken throughout the design cycle to provide flexibility to analyze a wide range of interesting and relevant space exploration scenarios while enabling the expansion for future capabilities through modular interfaces. Demonstrations of SpaceNet highlight the capability to model vastly different exploration campaigns at varying levels of fidelity while providing a user interface that is efficient and easy to use.

6.1 Significant Contributions

Many significant contributions exist in the revised modeling framework and software implementation. Modeling improvements enable new and more detailed concepts to be represented, enabling the analysis of new space exploration campaigns. Usability improvements help users model explorations efficiently and with greater feedback than before. Some of the most visible improvements are listed below.

Modeling Improvements:

- Arbitrary-Burn Space Transports
- Surface Transports and Vehicles
- Flight Transports
- Multi-destination Scenarios
- Element-centric Demand Models
- Modular Demand Model Interface
- Reconfigurable Element States
- Common Spares and Scavenging
- Repairable Parts, Auto-Repair
- Transport-level Manifesting
- Heuristic Auto-Manifesting
- Modular Data Source Interface

Usability Improvements:

- Completely Redesigned User Interface
- Element-based Bat Chart
- Scenario Feasibility Chart (Cumulative Delivery Capability and Demands)
- Time-Expanded Supply Network Visualization
- Repairability Effectiveness Chart
- Continuous Pre-Simulation for Real-time Error Feedback
- Robust Simulation Error Handling
- Integrated Database Editor
- Element Sizing Tool

6.2 Future Work

With the advancements in the modeling framework and user interface, there are several areas that could be targeted for future development and research.

First, there is a need for more detailed demand models to improve existing analyses. With the establishment of the modular demand model interface, developing and integrating new demand models

should be much easier than previously possible. More advanced models are needed to model spare parts demands, which currently depend on gross assumptions using sparing by mass models. Additional efforts should be focused on modeling electricity demand and production, one of the major portions of space exploration missing from the existing model. Electricity is different from other resources in that it is mass-less and generated by a power system and it will require refinement of production demand models, currently in limited use for ISRU plants.

Task analysis is another area for future development. Agents (both crew and robotic) should be capable of performing a task, a specialized event, which effectively makes available time another resource to manage. Tasks may include infrastructure deployment, maintenance or repair activities, or even scientific exploration. By adding tasks to the existing space exploration campaign definition, a finer level of detail is uncovered, leading to improved demand models and performance metrics.

As outlined in Appendix B, there is a distinct need for optimal multi-transport manifesting methods. For all but the most trivial campaigns or campaigns with a large margin of capacity, the heuristic auto-manifesting option only provides a starting point for manual manifesting which is a tedious and involved process. In addition to automated manifesting methods, additional options to account for varying cargo environments (pressurized and unpressurized), maximum allowable packing fractions, and required fixtures and support equipment (FSE) will help to introduce some of the finer-level challenges and inefficiencies of managing realistic cargo. There should also be options for user-defined cargo containers to help perform relevant trades.

Finally, as one of the long-term goals of space exploration campaign analysis, there is significant value in adding stochastic capabilities to the existing deterministic analysis. Though the existing domain modeling framework could represent stochastic demand generation to represent part failures or stochastic event generation to represent element failures, there is a significant body of prerequisite research as to how the campaign modeling framework should react as uncertainties are resolved. These questions are covered in detail in the Integrated Analysis Strategic Roadmap.

6.3 Integrated Analysis Strategic Roadmap

The modeling framework presented is a solid foundation on which to base future development, though there are some limitations to present analysis capabilities. Two areas have been identified as significant expansions for future development: expanding the analysis scope and the addition of stochastic analysis.

The scope of analysis within SpaceNet is currently confined to a single rigid campaign structure – any mission sequencing, operations, or element design deviations from the stated campaign require time-intensive sensitivity or trade studies. Ideally, one should be able to set the variables for complete campaign optimization, presently limited to the manifesting process for cargo.

Second, demands and events are currently considered deterministically. Some demands, such as crew provisions, have evolved from decades of research and can be consistently estimated with a reasonable degree of accuracy [43]. Demands for spares and other failure effects, however, are clearly not predictable, especially in the case of conceptual elements with unknown design parameters. Though there has been significant research into stochastic sparing models, in particular by Kline and Bachman [26], integration into existing campaign analysis is desired.

As a strategic roadmap forward, some thoughts on these two analysis expansions are listed in the following sections.

6.3.1 Analysis Scope

The analysis scope can be broken down into four levels focusing on three separate sub-problems of increasing difficulty, shown in Table 45. Analysis at level 0 simply evaluates feasibility of fully-defined campaigns. Present analysis capability is near Level 1, which considers the problem of how to manifest cargo to satisfy demands throughout a campaign. Level 2 analysis considers the problem of how to schedule missions and explorations with known architectures. Level 3 analysis considers the problem of how to architect elements and missions.

Table 45: Levels of space exploration analysis.

Analysis Level	Manifesting Problem	Scheduling Problem	Architecting Problem
0			
1*	X		
2	X	X	
3	X	X	X

* Existing level of analysis is near Level 1

Manifesting Problem

The manifesting problem considers cargo placement a variable in the campaign definition. This level of analysis is most useful for evaluating or optimizing schedule-constrained campaigns. The solution to the manifesting problem is a definition of the creation and movement of resources through the time-expanded

network using defined elements and transport mission events. An optimal cargo manifest can be found given campaign goals as an objective function.

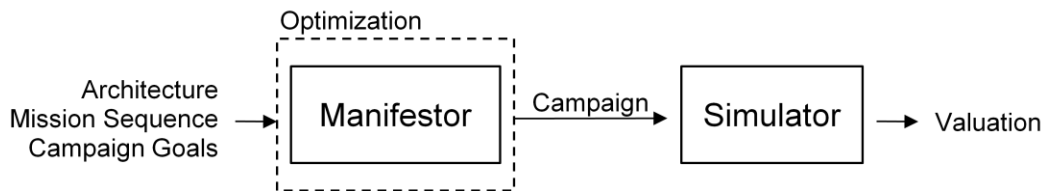


Figure 67: Manifesting problem diagram. *Optimization of the manifesting problem produces a cargo manifest to support exploration goals during simulation.*

The SpaceNet model includes a rudimentary ability to solve the manifesting problem, though optimal manifesting methods have yet to be implemented within the modeling framework (See Appendix B).

Scheduling Problem

The scheduling problem considers mission definition a variable in the campaign definition. This level of analysis is most useful for partially-constrained campaign definitions where the mission schedule is not known in detail. An example campaign that would benefit from this level of analysis is the ISS Resupply use case, in which the capabilities of the elements are known but the schedule for resupply is variable. An optimal mission schedule can be found given campaign goals as an objective function.

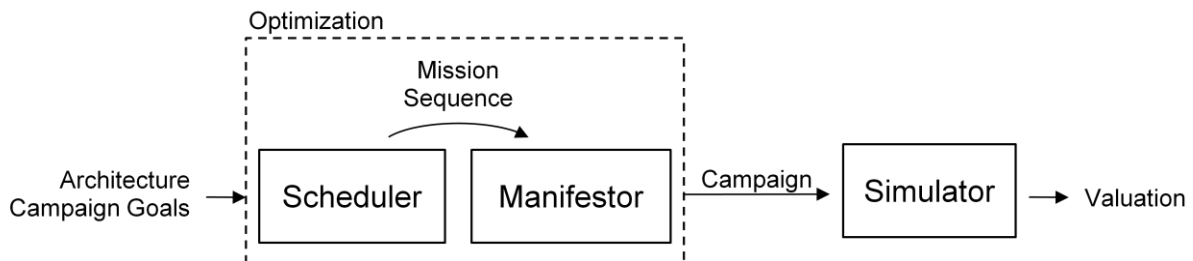


Figure 68: Scheduling problem diagram. *Optimization of the scheduling problem produces a mission sequence and cargo manifest to support exploration goals during simulation.*

The scheduling problem is the clear next step in analysis for SpaceNet. Goals for campaigns, such as the continued human habitation of ISS or the establishment of a lunar outpost, are often known far in advance of the mission sequence. By leaving the sequencing as a variable, the campaign can be quickly modified to account for changes and unexpected events.

Architecting Problem

The architecting problem considers the detailed element design a variable in the campaign definition. This level of analysis is most useful for conceptual campaign analysis used to drive element design

requirements. Examples of campaigns that would benefit from this level of analysis include long-term Lunar and Martian explorations. The solution to the architecting problem is a design description of the elements and missions to optimize the campaign value.

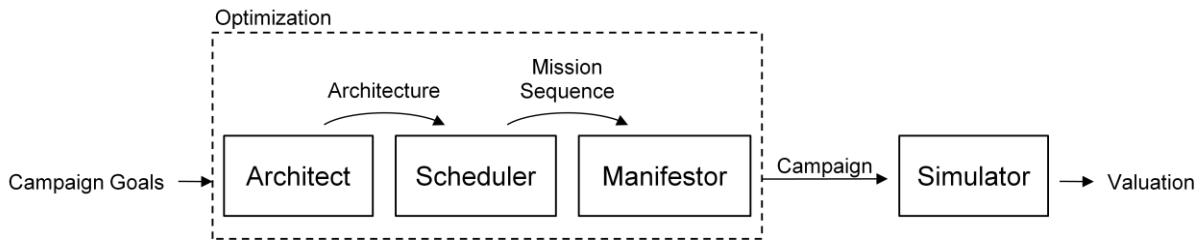


Figure 69: Architecting problem diagram. *Optimization of the architecture problem produces mission and element architectures, a mission sequence, and a cargo manifest to support exploration goals during simulation.*

Due to the immense design space for space exploration campaigns, the architecting problem may only be tractable for a handful of decisions, if at all. Traditionally, trade studies are performed to inspect ranges of design variables for elements or missions, an analysis framework tackling the architecture problem may only serve as an interface to automated trade studies. It is important to note that the solution to the scheduling and manifesting problems will likely change for each conceptual element design, further complicating the solution of the architecting problem.

6.3.2 Stochastic Analysis

The move towards stochastic analysis necessitates an important distinction between a campaign and a scenario: a scenario being an instance of a campaign taking into account a particular set of random variables driving demand generation or event execution. In other words, a scenario is one possible execution of a planned campaign. In deterministic analysis, the two terms are interchangeable.

This section only outlines the effects of stochastic analysis at level 1, that is, for manifest optimization. Two components to stochastic analysis considered include stochastic demands and stochastic events.

Stochastic Demands

From the framework of the SpaceNet model, stochastic demands could be represented by demand models that need not generate equivalent demands in two simulations. Stochastic demand models would be a simple extension from the existing demand models, though the manifesting process would be quite different. In the deterministic case, demands estimated by the demand simulator can be used to create a manifest which satisfies all demands during the campaign, resulting in a campaign valuation. With stochastic demands, however, there is no single case for which to manifest, rather, the demand simulator

must simulate many scenarios to build a distribution of demands. The demands must then be prioritized and systematically selected to build a robust manifest.

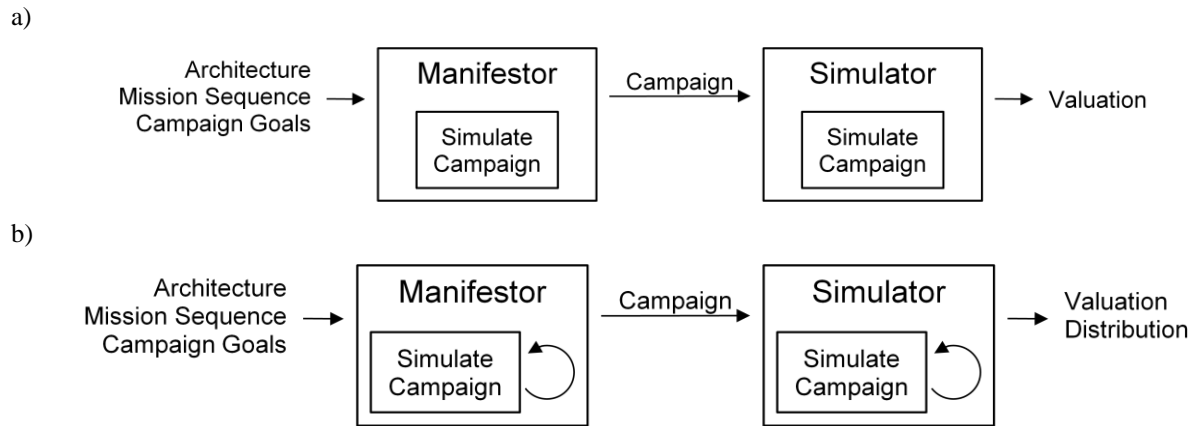


Figure 70: Deterministic versus stochastic demands. (a) Deterministic demands provide a single campaign valuation. (b) Stochastic demands require stochastic manifesting and result in a distribution of campaign valuations.

Under some scenarios, the manifested demands will likely be insufficient. The unsatisfied demands alone, however, do not have an effect on the scenario. Stochastic events provide a means to show effect from insufficient demands.

Stochastic Events

Stochastic events represent event executions that could have unexpected outcomes. For example, if an element is to enter a quiescent state if there are insufficient resources to satisfy a demand, its effect on the overall campaign is uncertain until execution time of a particular scenario. Similarly, if resupply mission failures were modeled as stochastic events, they would alter the existence and distribution of resources, clearly affecting the forward simulation.

The challenge with stochastic events is that the forward-looking scenario may change after each event execution, necessitating re-evaluation of any optimized quantities. As illustrated in Figure 71 for an example level 1 analysis, the manifest must be re-evaluated after each event execution.

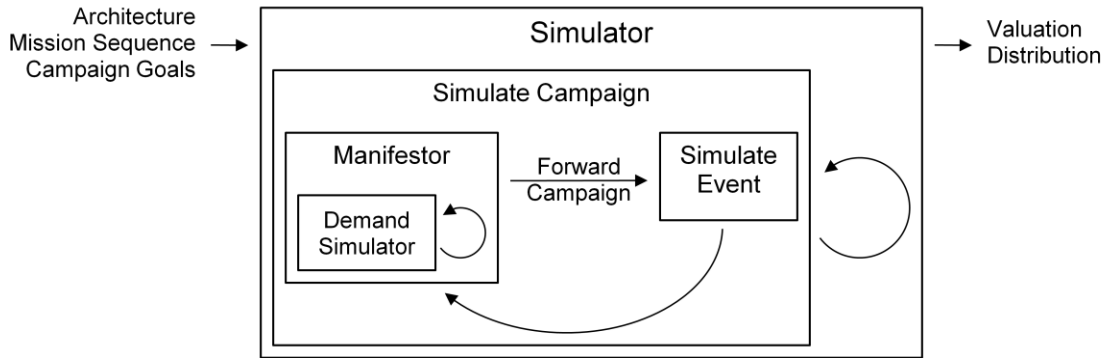


Figure 71: Stochastic event simulation. *The manifesting process must be continuously revised during simulation to react to resolved uncertainties.*

In a brute force implementation where the manifest is updated after every event, this addition would drive the simulation time from $O(n)$ to $O(n^2)$ where n is the number of events in a campaign. This increase in operational complexity is unlikely to be tractable for campaigns of interest, which may have hundreds or thousands of events and many samples of each simulation for Monte Carlo analysis. Methods should be developed to partially decouple the optimization phase (manifestor, scheduler, and architect) and simulation to get closer to the determinate case.

Appendix A Object Model Diagrams

This appendix presents object model diagrams to serve as a visual companion to the modeling framework description. Although there are several types of object model diagrams used, the ones presented here are simplified to highlight the most important aspects of the model.

Abstract object classes are displayed as boxes with an italicized font. Abstract object classes primarily provide a design interface to group concrete implementations in a way that is extensible to future development. Concrete objects classes are displayed as boxes with a standard font. Concrete object classes may implement an abstract class or may provide other functionality.

Subclass relationships, indicated with a white triangle, identify specialized object classes of the super-class that can be substituted for any generic super-class application. Aggregation relationships, indicated with a white diamond, identify access to related object classes that may change over the course of the object lifetime. This differs from composition relationships, indicated with a black diamond, which identify permanent ownership of related object classes.

In the example illustrated in Figure 72, the *Vehicle* abstract object class serves as an interface to the concrete object subclasses *Car* and *Airplane*. The *Car* object class is a composition of *Engine* using the assumption that a car's engine does not change and it is an aggregation of *Passenger*.

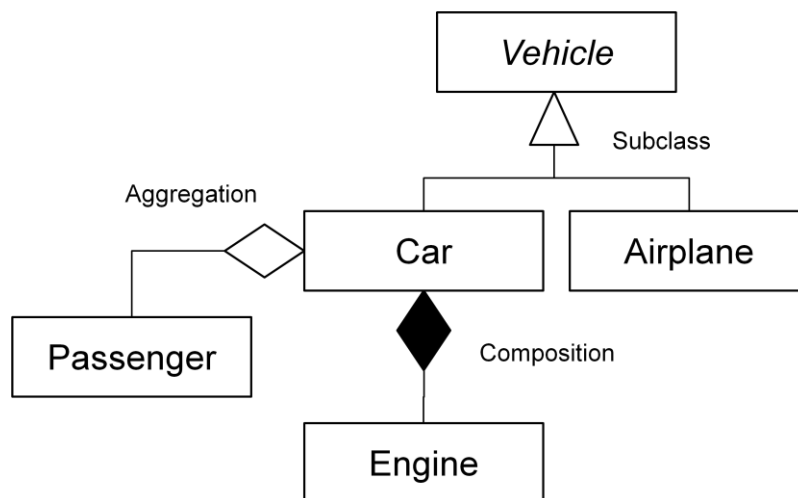


Figure 72: Example object model diagram. The *Vehicle* interface has two subclass objects, *Car* and *Airplane*. A car is comprised of an engine object and aggregates *Passenger* objects.

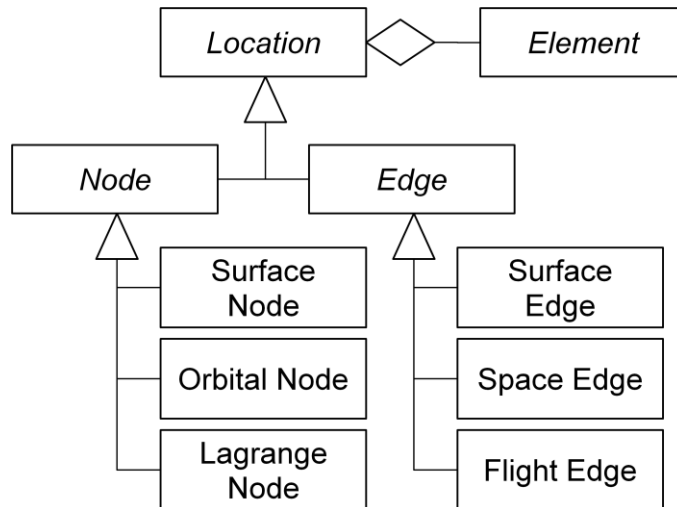


Figure 73: Location objects. Locations aggregate Elements. The Location interface is expanded by the Node and Edge interface, which both have several subclasses.

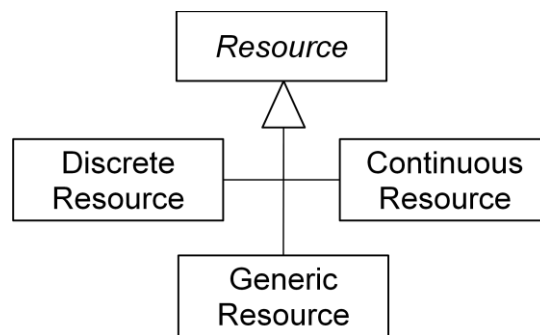


Figure 74: Resource objects. Three subclasses of the Resource interface include Discrete, Generic, and Continuous Resources.

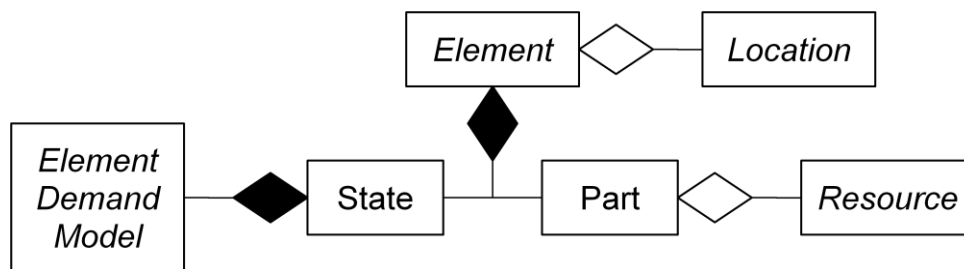


Figure 75: Element objects. The Element interface is comprised of States and Parts. States are comprised of Element Demand Models and Parts aggregate Resources.

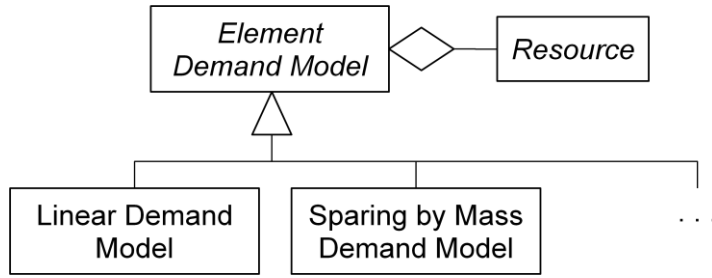


Figure 76: Element demand model objects. *Element Demand Models aggregate Resources. Subclasses include Linear and Sparing by Mass Demand Models.*

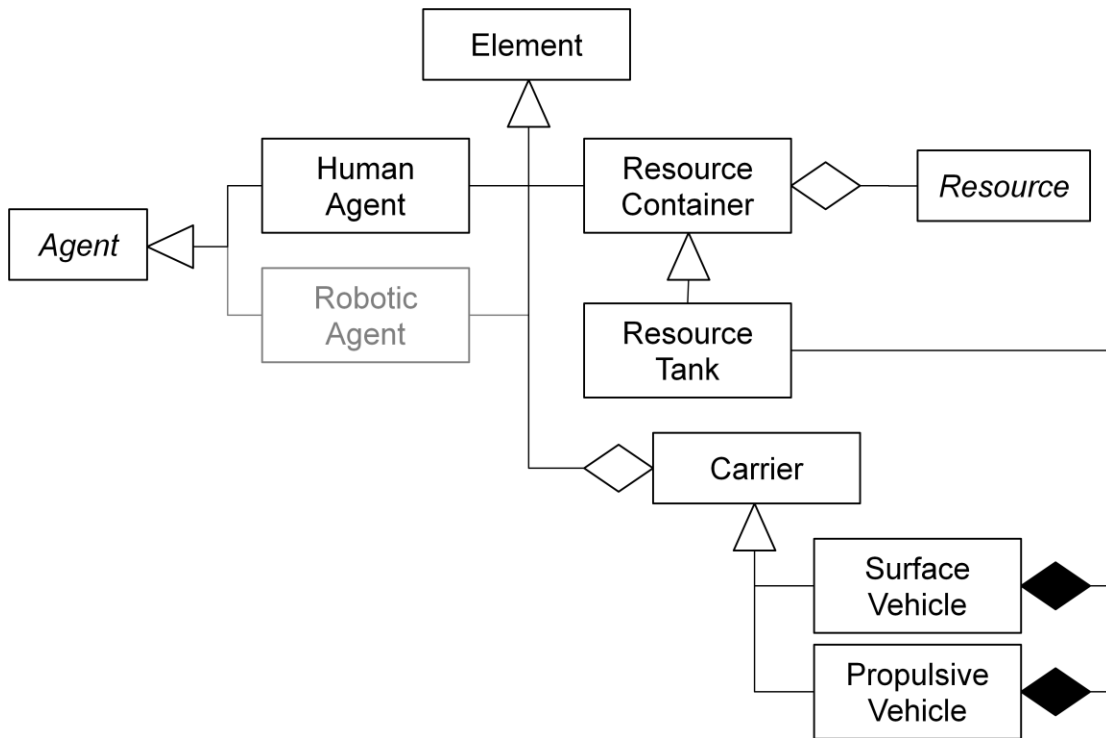


Figure 77: Element object hierarchy. *All elements inherit attributes from the base Element class. Subclasses include Human Agents, Resource Containers, and Carriers. Resource Tanks are subclasses of Resource Containers and Surface and Propulsive Vehicles are subclasses of Carriers.*

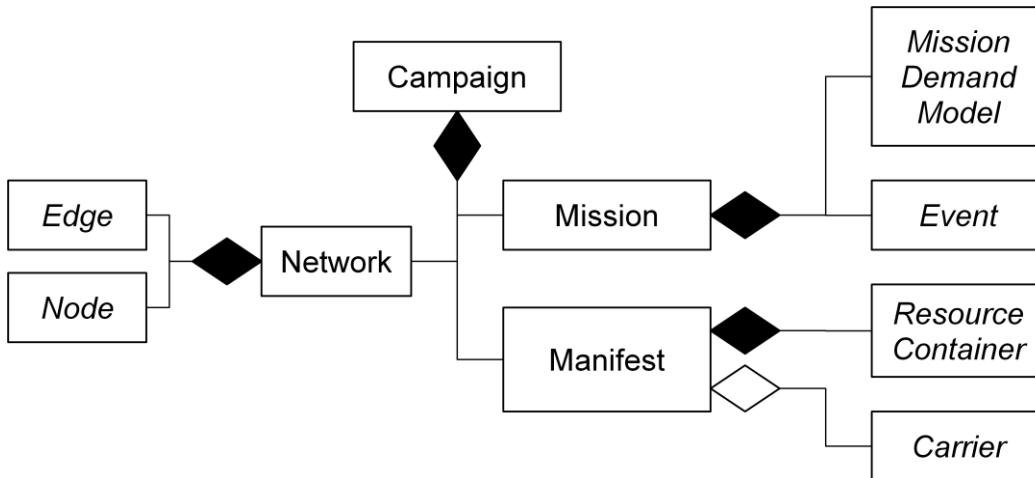


Figure 78: Campaign objects. A campaign model is comprised of a Network, Mission, and Manifest models.

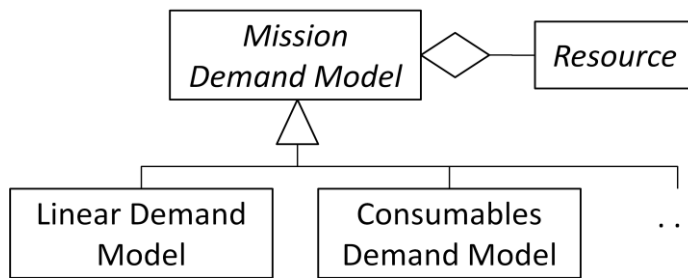


Figure 79: Mission demand model objects. Mission demand models are very similar to element demand models in that they aggregate Resources.

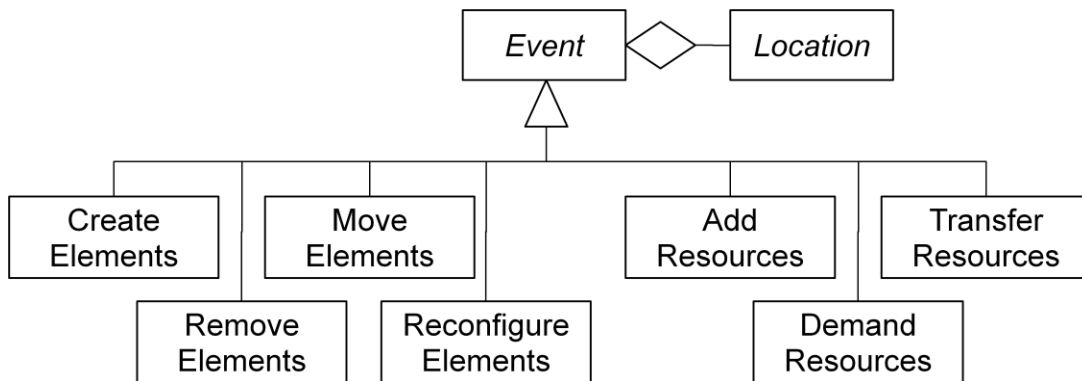


Figure 80: Core event objects. Core element-based events include creation, movement, removal, and reconfiguration. Core resource-based events include addition, demand, and transfer.

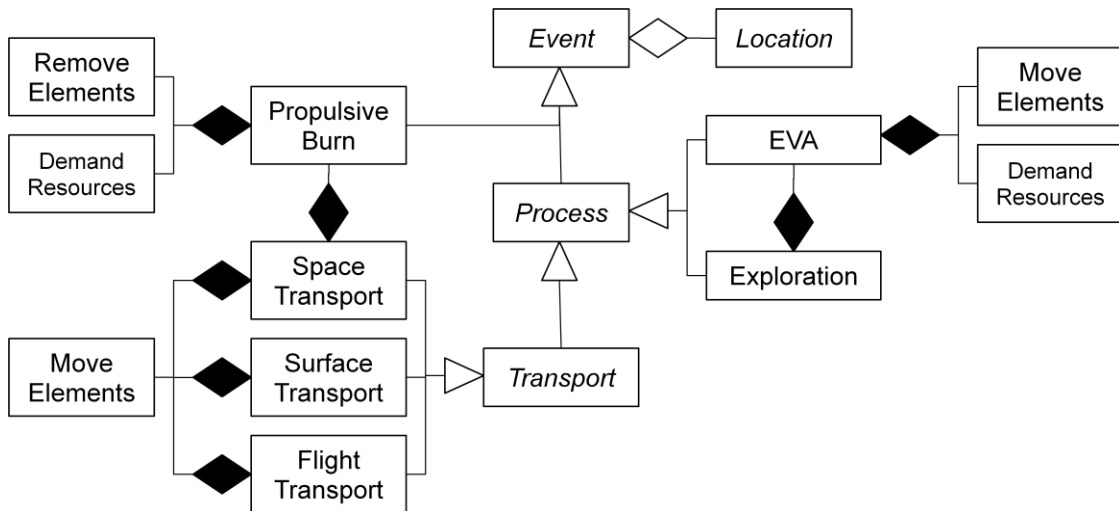


Figure 81: Composite event objects. *Composite events use sequences of core events to build up to complex functionality.*

Appendix B Optimal Manifesting Methods

The manifesting problem, of determining what cargo to place on which flight to satisfy demands for resources during exploration, is an important aspect of space exploration campaigns. A well-planned manifesting policy with an optimal mix of pre-positioned, carry-along, and re-supplied cargo is essential for balancing risks, ensuring robustness against delays and cancellations, and achieving maximum possible exploration capability and overall mission success.

The existing SpaceNet manifesting model provides a limited ability to generate cargo manifests leveraging the multi-transport capabilities of space exploration campaigns. This appendix introduces ongoing research into matrix representations of cargo manifests that can be optimized using standard techniques. The present state of research operates at a higher level than the SpaceNet modeling framework where only generic cargo mass is considered and the details of packing into resource containers and manifesting on specific carriers are abstracted to a simpler notion. The goal of future research is to combine the methods to provide a unified manifesting model for optimization.

Previous work established matrix-based methods for modeling the manifesting process which has been demonstrated with an analysis of the International Space Station resupply logistics [34]. It has been shown in detail how the cargo manifests for a multi-flight, single-destination scenarios can be represented with a square, M (manifest)-matrix. This existing model has been expanded upon using aspects of the SpaceNet modeling framework allow for multi-transport, multi-node campaigns. The expanded framework provides matrix-based methods for a wider range of scenarios including long-duration crewed missions, necessary for any beyond low-Earth orbit exploration.

Modeling Cargo Manifests

Using an abstraction of the SpaceNet modeling framework for space exploration, a campaign is represented using seven vector components, the origin node vector N_o , the destination node vector N_d , the departure time vector T_d , the arrival time vector T_a , the transport capacity vector C , the exploration demands vector D_e , and the transport demands vector D_t , shown in Eq. (A1-A7) for a campaign of n transports.

$$N_o = [n_{o,1} \quad n_{o,2} \quad \cdots \quad n_{o,n}]^T \quad (A1)$$

$$N_d = [n_{d,1} \quad n_{d,2} \quad \cdots \quad n_{d,n}]^T \quad (A2)$$

$$T_d = [t_{d,1} \quad t_{d,2} \quad \cdots \quad t_{d,n}]^T \quad (\text{A3})$$

$$T_a = [t_{a,1} \quad t_{a,2} \quad \cdots \quad t_{a,n}]^T \quad (\text{A4})$$

$$C = [c_1 \quad c_2 \quad \cdots \quad c_n]^T \quad (\text{A5})$$

$$D_e = [d_{e,1} \quad d_{e,2} \quad \cdots \quad d_{e,n}]^T \quad (\text{A6})$$

$$D_t = [d_{t,1} \quad d_{t,2} \quad \cdots \quad d_{t,n}]^T \quad (\text{A7})$$

The process of manifesting maps cargo onto transports to satisfy the demands during the exploration, rendering it feasible. A campaign's manifest is represented with three matrix components, shown in Eq. (A8-A10) for a campaign with n transports.

The exploration utilization matrix, U_e , represents the utilization of cargo during exploration periods. The element $u_{e,ij}$ is the mass of cargo brought by transport i that is consumed in exploration period j of the campaign. When cargo is utilized, it is removed from the scope of the analysis – waste, packaging, and accommodation mass are not considered. Cargo utilization between transports is only valid both transports have the same destination node.

$$U_e = \begin{bmatrix} u_{e,11} & u_{e,12} & \cdots & u_{e,1n} \\ u_{e,21} & u_{e,22} & \cdots & u_{e,2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{e,m1} & u_{e,m2} & \cdots & u_{e,mn} \end{bmatrix} \quad (\text{A8})$$

Crewed missions, especially ones with long-duration transports, are highly dependent on demands during transport. These demands must be satisfied by the transport in which they originate, further constraining the campaign. The transport utilization matrix, U_t , represents utilization of cargo during transports. The element $u_{t,ij}$ is the mass of cargo brought by transport i that is consumed for transport j .

$$U_t = \begin{bmatrix} u_{t,11} & u_{t,21} & \cdots & u_{t,1n} \\ u_{t,12} & u_{t,22} & \cdots & u_{t,2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{t,m1} & u_{t,m2} & \cdots & u_{t,mn} \end{bmatrix} \quad (\text{A9})$$

The transfer matrix, T , represents cargo is transferred from one transport to another. The element τ_{ij} is the mass of cargo brought by transport i that is transferred to transport j . Transfer of cargo between transports may not involve physical movement if the same vehicles are used in both transports. A transfer is valid

only if the destination node of transport i is the origin node of transport j and the arrival of transport i is before the departure of transport j .

$$\mathbf{T} = \begin{bmatrix} \tau_{11} & \tau_{12} & \cdots & \tau_{1n} \\ \tau_{21} & \tau_{22} & \cdots & \tau_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tau_{n1} & \tau_{n2} & \cdots & \tau_{nn} \end{bmatrix} \quad (\text{A10})$$

The manifest, μ , is the vector defined by the set of inputs that drive the movement of cargo throughout a campaign, given by the valid elements of the matrices \mathbf{U}_e , \mathbf{U}_t , and \mathbf{T} . Validity conditions enforce spatial and temporal criteria for manifest actions, as shown in Eq. (A11).

$$\mu \equiv \{\mathbf{U}_e, \mathbf{U}_t, \mathbf{T}\} \forall \begin{cases} u_{e,ij} \mid n_{d,i} = n_{d,j} \\ u_{e,ij} \mid t_{a,i} \leq t_{a,j} \\ u_{t,ij} \mid i = j \\ \tau_{ij} \mid n_{d,i} = n_{o,j} \\ \tau_{ij} \mid t_{a,i} \leq t_{d,j} \end{cases} \quad (\text{A11})$$

Campaign Feasibility and Manifest Optimization

For a manifest to be feasible, three criteria must be satisfied: capacity constraints, demand satisfaction, and mass conservation. Capacity constraints ensure the sum of exploration utilization, transport utilization, and transferred cargo does not exceed the capacity of each transport, as shown in Eq. (A12).

$$\sum_j (u_{e,ij} + u_{t,ij} + \tau_{ij}) \leq c_i \quad \forall i \quad (\text{A12})$$

Demand satisfaction ensures that cargo is utilized to satisfy all exploration and transportation demands, as shown in Eq. (A13-A14).

$$\sum_j u_{e,ji} = d_{e,i} \quad \forall i \quad (\text{A13})$$

$$\sum_j u_{t,ji} = d_{t,i} \quad \forall i \quad (\text{A14})$$

Mass conservation ensures that mass is not created or destroyed for each transport, with the exception for transports that originate at a source node, represented with the set \mathbf{S} , where resources can be created, as shown in Eq. (A15).

$$\sum_j (\tau_{ji} - u_{e,ij} - u_{t,ij} - \tau_{ij}) = 0 \forall i | n_{o,i} \notin S \quad (\text{A15})$$

The campaign is considered feasible if a solution exists to a system of linear inequalities equalities based on the above constraints, as shown in Eq. (A16). The A-matrices impose constraints on the manifest vector elements. A_c imposes capacity constraints from Eq. (A12), A_{de} imposes exploration demand constraints from Eq. (A13), A_{dt} imposes transport demand constraints from Eq. (A14), and A_m imposes mass conservation constraints from Eq. (A15).

$$\text{find } \mu \text{ such that } \begin{cases} A_c \cdot \mu \leq C \\ A_{de} \cdot \mu = D_e \\ A_{dt} \cdot \mu = D_t \\ A_m \cdot \mu = 0 \\ \mu \geq 0 \end{cases} \quad (\text{A16})$$

Once a campaign is deemed feasible, its manifest may be optimized subject to objectives based on some desired policy or strategy. Potential objective functions may seek to distribute risk, improve robustness to schedule delays or cancellations, or maximize pre-positioned resources.

References

1. AIAA Space Logistics Technical Committee. Retrieved 13 May 2010 from <http://info.aiaa.org/tac/SMG/SLTC>
2. "A Renewed Spirit of Discovery: The President's Vision for U.S. Space Exploration," NASA, February 2004.
3. "The Vision for Space Exploration," NASA-NP-2004-01-334-HQ, February 2004.
4. "NASA's Exploration Systems Architecture Study," NASA-TM-2005-214062, November 2005.
5. "NASA Unveils Global Exploration Strategy and Lunar Architecture", Press Release 06-361, Houston, Texas, December 4, 2006.
6. Shull S., Gralla E., de Weck O., "A Modeling Framework for Interplanetary Supply Chains", AIAA-2006-7229, AIAA Space 2006 Conference and Exposition, San Jose, California, September 19-21, 2006.
7. Taylor C., Song M., Klabjan D., de Weck O., Simchi-Levi D., "A Mathematical Model for Interplanetary Logistics", Logistics Spectrum, Vol. 41, No. 1, 2007, pp 23-33.
8. Armar, N., de Weck, O.L., Cohanin, B., "1 versus 1.5 versus 2 Launch Architecture", CxP IDAC-2, NASA, August 2, 2006.
9. Whiting, J., Sturm, E., Armar, N., de Weck, O.L., "Impact of Element Propellant Types on Lunar Payload Delivery", CxP IDAC-2, NASA, August 15, 2006.
10. de Weck O.L., Simchi-Levi D., Shishko R., Ahn J., Gralla E., Klabjan D., Mellein J., Shull A., Siddiqi A., Bairstow B, Lee G., "SpaceNet v1.3 User's Guide", NASA/TP-2007-214725, January 2007.
11. Shull S., de Weck O., "Modeling and Simulation of Lunar Campaign Logistics", AIAA-2007-6244, AIAA Space 2007 Conference and Exposition, Long Beach, California, September 18-20, 2007.
12. Shull, S. "Integrated Modeling and Simulation of Lunar Exploration Campaign Logistics", M.S. Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts. 2007.
13. Lee G., de Weck O.L., Armar N., Jordan E., Shishko R., Siddiqi A., and Whiting J., "SpaceNet: Modeling and Simulating Space Logistics", AIAA-2008-7747, AIAA Space 2008 Conference and Exposition, San Diego, California, September 9-11, 2008.
14. Lee G., Benzin K., de Weck O., Armar N., Siddiqi A., Jordan E., Whiting J., "Integrated Cx Mission Modeling," ATA-01-1006, 2007.
15. Armar N., de Weck O., "Cargo Revenue Management for Space Logistics", AIAA-2009-6723, AIAA Space 2009 Conference and Exposition, Pasadena, California, September 14-17, 2009.
16. Holdren, J., Letter to Christopher Scolese, NASA, May 7, 2009.
17. "U.S. Announces Review of Human Space Flight Plans," Press Release, Washington D.C., May 7, 2009.
18. "Seeking a Human Space Flight Program Worthy of a Great Nation," NASA, October, 2009.
19. "Fiscal Year 2011 Budget Estimates", NASA, February 2010.
20. "Statement by Charlie Bolden", NASA Budget Press Conference, February 1 2010.
21. "Human Exploration of Mars Design Reference Architecture 5.0," NASA, NASA-SP-2009-566, July 2009.
22. Cirillo W., Earle K., Goodliff K., Reeves J.D., Stromgren C., Andraschko M., Merrill R.G., "Strategic Analysis Overview," AIAA-2008-7778, AIAA Space 2008 Conference and Exposition, San Diego, California, September 9-11, 2008.
23. Stromgren C., Galan R., Cirillo W., "Micro-Logistics for Human Space Exploration," AIAA-2008-7780, AIAA Space 2008 Conference and Exhibition, San Diego, California, September 9-11 2008.

24. Andraschko M., Merrill R.G., Earle K., "Logistics Modeling for Lunar Exploration Systems," AIAA-2008-7746, AIAA Space 2008 Conference and Exposition, San Diego, California, September 9-11, 2008.
25. Siddiqi A., de Weck O. L., "Spare Parts Requirements for Space Missions with Reconfigurability and Commonality," *Journal of Spacecraft and Rockets*, Vol. 44, No. 1, 2007.
26. Kline R. C., Bachman T. C., "Estimating Spare Parts Requirements with Commonality and Redundancy," *Journal of Spacecraft and Rockets*, Vol. 44, No. 4, 2007.
27. Shull S., Gralla E., de Weck O., Shishko R., "Future of Asset Management for Human Space Exploration: Supply Classification and an Integrated Database", AIAA-2006-7232, AIAA Space 2006 Conference and Exposition, San Jose, California, September 19-21, 2006.
28. "NASA Systems Engineering Handbook," Revision 1, NASA/SP-2007-6105, NASA, 2007.
29. Gohr A., "DokuWiki", URL: <http://www.dokuwiki.org> [cited 2 February 2010].
30. The Apache Software Foundation, "Apache Subversion", URL: <http://subversion.apache.org> [cited 2 February 2010].
31. Eclipse.org, "Eclipse Home", URL: <http://www.eclipse.org> [cited 11 June 2010].
32. Free Software Foundation, "GNU General Public License", Version 3, 29 June 2007, URL: <http://www.gnu.org/licenses/gpl.txt> [cited 1 February 2010].
33. Oracle Corporation, "Javadoc Tool Home Page", URL: <http://java.sun.com/j2se/javadoc/> [cited 2 February 2010].
34. Siddiqi, A., de Weck, O., Lee, G., & Shull, S. "Matrix Modeling Methods for Spaceflight Campaign Logistics Analysis", *Journal of Spacecraft and Rockets*, Vol. 46, No. 5, 2009, pp. 1037-1048.
35. "NASA Awards Space Station Commercial Resupply Services Contracts," Contract Release C08-069, NASA, December 2008.
36. Macias B., "Cargo Resupply Analysis for the International Space Station, Post Assembly Complete Timeframe," AIAA-2000-0007, 38th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 10-13, 2000.
37. "Consolidated Launch Manifest," NASA, URL: http://www.nasa.gov/mission_pages/station/structure/iss_manifest.html [cited 24 June 2010].
38. Smith E. A., "A Manned Flyby Mission to Eros," *Proceedings of the Third Space Congress*, "The Challenge of Space," Cocoa Beach, Florida, March 7-10, 1966, pp 137-155.
39. Landis R., Kormsmeier D., Abell P., Adamo D., "A Piloted Orion Flight Mission to a Near-Earth Object: A Feasibility Study," AIAA-2007-6168, AIAA Space 2007 Conference and Exposition, Long Beach, California, September 18-20, 2007.
40. Abell P. A., Kormsmeier D. J., Landis R. R., Jones T. D., Adamo D. R., Morrison D. D., Lemke L. G., Gonzales A. A., Gershman R., Sweetser T. H., Johnson L. L., Lu E., "Scientific exploration of near-Earth objects via the Orion Crew Exploration Vehicle," *Meteoritics & Planetary Science*, Vol. 44, No. 12, 2009, pp. 1825-1836.
41. Mazanek D. D., Troutman P. A., Culbert C. J., Leonard M. J., Spexarth G., "Surface Buildup Scenarios and Outpost Architectures for Lunar Exploration," IEEE-1093, 2009 IEEE Aerospace Conference, Big Sky, Montana, March 7-14, 2009.
42. Kennedy K. J., Toups L. D., Rudisill M., "Constellation Architecture Team – Lunar Scenario 12.0 Habitation Overview," JSC-CN-19362, Earth and Space 2010 Conference, Honolulu, Hawaii, March 15-17, 2010.
43. Larson W., Pranke L., "Human Spaceflight: Mission Analysis and Design", McGraw-Hill, 1999.