Space Logistics Exploration Campaign Scenario Specification for SpaceNet

Paul T. Grogan[1]
Arizona State University, Tempe, AZ, 85287

Sarah Bentley[2], George Lordos[3], Kir Latyshev[4], Ireland M. Brown[5], and Olivier L. de Weck[6]
Massachusetts Institute of Technology, Cambridge, MA, 02139

Space logistics analysis is a critical systems engineering function to ensure feasibility of space exploration campaigns starting from early design concepts and maturing to more detailed operational plans. SpaceNet is an existing simulation tool that evaluates propulsive feasibility based on impulsive maneuvers required to traverse the exploration network and logistical feasibility by comparing resource demands to sustain mission operations to the available mass and volume afforded by the exploration architecture. A SpaceNet scenario contains information about the exploration network, campaign objects (resources, demand models, and elements), and mission events that comprise a human space exploration campaign. Recent development created a SpaceNet scenario specification using JavaScript Object Notation (JSON) Schema, a declarative language that annotates and validates JSON documents, expanding options for scenario construction, sensitivity analysis and trade studies, and distributed evaluation across diverse computing platforms. A supporting open-source Python language library provides scenario object classes for scripting and automation. This paper provides a technical description of the updated SpaceNet scenario specification. An example application case models a Lunar sortie mission based on the Artemis 3 mission using SpaceNet. The exploration mission considers a crew of four who rendezvous with the human landing system in a near rectilinear halo orbit about Earth-Moon L2. Two crew members enter the lander, transfer to a low Lunar orbit, and descend to the Lunar South Pole for a seven-day surface exploration. After collecting Lunar samples, they ascend to low Lunar orbit, transfer to the halo orbit, rendezvous with the crew and service module, and return to Earth. Results verify propulsive feasibility of a baseline mission concept without considering resource demands, showing positive propellant margins in the launch vehicle, in-space stage, landing system, and service module.

Nomenclature

COS	=	class of supply
g_0	=	standard acceleration of gravity
I_{sp}	=	specific impulse
m_{burn}	=	propellant mass expended in burn
m_{stack}	=	total stack mass at start of burn
$m_{\text{available}}$	=	propellant mass available at start of burn
JSON	=	JavaScript Object Notation
ΔV	=	change in velocity
ΔV_{target}	=	target change in velocity to be achieved by a burn
$\Delta V_{\text{achieved}}$	=	change in velocity achieved by burn

[1]Associate Professor, School of Computing and Augmented Intelligence, PO Box 878809, AIAA Senior Member.
[2]Undergraduate Research Assistant, Department of Mathematics.
[3]Graduate Instructor, Department of Aeronautics and Astronautics, AIAA Student Member.
[4]Graduate Research Assistant, Department of Aeronautics and Astronautics, AIAA Student Member.
[5]Graduate Research Assistant, Department of Aeronautics and Astronautics.
[6]Apollo Program Professor, Department of Aeronautics and Astronautics, AIAA Fellow.

I. Introduction

The coming decades envision human space exploration conducted at distant locations including Earth orbit, Lunar orbit, the Lunar surface, near-Earth asteroids, and, ultimately, the Martian surface. Extended exploration campaigns will leverage logistics strategies such as pre-positioned resources, depots, in-situ resource production, and commercial (re)supply services to sustain human life and operations. Serious consideration of space logistics during early conceptual mission design phases is essential to ensure feasible architectures, especially when considering significant capital and operating cost constraints in commercial exploration ventures.

The AIAA Space Logistics Technical Committee defines space logistics, in part, as managing the flow of materiel, services, and information needed throughout a space system lifecycle. During conceptual mission design phases, space logistics analysis focuses on functions such as demands estimation, packing, and manifesting verify logistical feasibility, i.e., whether a baseline mission plan has sufficient capacity to sustain nominal operations. Demands estimation identifies the time, location, and type of resources required to support mission operations. Packing assigns bulk resources to discrete logistics containers to account for additional stowage and restraint mass and volume during transport. Finally, manifesting assigns packed containers to unique vehicles scheduled to traverse between locations.

Software tools like SpaceNet perform space logistics analysis by simulating an extended space exploration mission or campaign of missions [1]. A scenario file provides the input to specify a space exploration campaign and combines a database of object attributes (structure) with a sequence of events to be executed during a mission (behavior). SpaceNet adopts a flexible scenario structure that can model a wide variety of scenarios ranging from low-Earth orbit space stations, long-duration Lunar surface bases, asteroid missions, and Mars exploration [2].

The objective of this paper is to describe the SpaceNet scenario specification and explain how it can support systems engineering analysis during conceptual mission design stages. For example, automated and systematic variations to baseline scenarios can support trade studies, sensitivity analyses, or uncertainty quantification using Monte Carlo sampling over random variables. Within SpaceNet, a scenario specification in JSON Schema allows further interoperability across diverse computing platforms including Excel, Python, and Java.

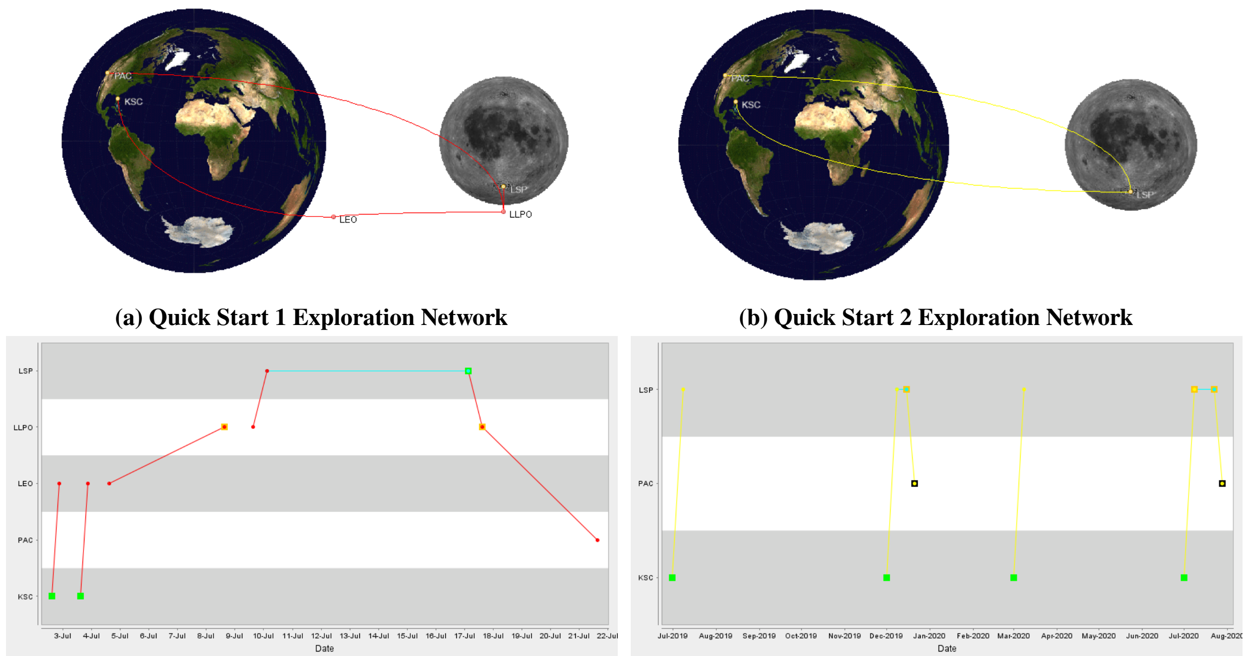
II. Background

SpaceNet was initially developed by the MIT Space Logistics Project under the NASA Constellation Program. The first public release, version 1.3, was implemented in MATLAB [3]. Efforts to generalize SpaceNet exploration scenarios beyond Lunar campaigns introduced an object-oriented data model, a Java Swing graphical user interface (GUI), and greatly expanded set of application cases [4, 5]. Application cases enabled by SpaceNet 2.5 include a 77-mission resupply campaign to the International Space Station (ISS), a 17-mission Lunar campaign to establish a permanent base with surface mobility to traverse between exploration sites, a sortie mission to a near-Earth asteroid re-using the Constellation architecture elements, and a flexible Martian exploration campaign based on the Design Reference Architecture (DRA) 5.0 augmented with sortie missions to Martian moons Phobos and Deimos [2].

Development work on SpaceNet paused after Constellation Program cancellation in 2010. In the intermediate years, space logistics campaign research explored flexible path concepts for future human space exploration [6]. In conjunction with the Space Transportation System (STS) retirement in 2011, NASA's Commercial Orbital Transportation Services (COTS) and Commercial Resupply Services (CRS) programs successfully realized commercial cargo resupply to the ISS starting in 2012. Subsequently, NASA's Commercial Crew Development (CCDev) and Commercial Crew programs successfully delivered crew to the ISS in 2020. Private astronauts on Axiom Mission 1 (Ax-1) reached the ISS in 2022.

Current NASA-solicited programs include Commercial LEO Destinations (CLD) which envisions a commercial space station in LEO, the Human Landing System (HLS) which envisions a commercial landing vehicle for Lunar surface missions, and Commercial Lunar Payload Services (CLPS) which seeks to emulate COTS/CRS for cis-Lunar resource delivery. Meanwhile, NASA and its partners develop Artemis Program components including the Space Launch System (SLS) launch vehicle, Lunar Gateway extraterrestrial space station orbiting the Earth-Moon L2 point, and Orion Multi-Purpose Crew Vehicle (MCPV). The sum of successes over the past decade demonstrate that future human spaceflight campaigns will involve parties spanning government, international, and commercial partners, significantly complicating the logistics and supply chain to support human activity at remote sites.

Over the preceding decade, the landscape for scientific computing has shifted from traditional high-level languages like C++ or Java and proprietary languages like MATLAB towards accessible languages such as Python with high-quality open-source computing libraries and distributed computing environments ranging from Jupyter Notebooks to cloud computing services. Effort starting in 2021 developed components for SpaceNet Cloud, a Python-native library to model and manage SpaceNet constructs with a prototype browser-based GUI [1]. Currently, there are two open-source



(a) Quick Start 1 Exploration Network

(b) Quick Start 2 Exploration Network

(c) Quick Start 1 Campaign Bat Chart

(d) Quick Start 2 Campaign Bat Chart

Fig. 1 Exploration networks and campaign bat charts for quick start 1 (left) and quick start 2 (right) scenarios.

versions of SpaceNet available on public repositories. SpaceNet Java maintains the version 2.5 desktop application with rich GUI and full simulation and analysis capabilities [7]. SpaceNet Cloud provides a Python-language library in progress for longer-term transition which is currently limited to scenario specification [8]. Both SpaceNet Java and SpaceNet Cloud can read and write JSON-serialized scenarios, establishing a stable interface for further development.

Both SpaceNet platforms provide two documented “quick start” scenarios to help new users learn the main constructs and analysis capabilities. Figure 1 illustrates the exploration network and campaign bat charts (showing the movement of resources and elements through a time-expanded network) for the quick start scenarios.

Quick start scenario 1 evaluates propulsive feasibility for a Lunar surface sortie mission. It considers the Constellation Program’s dual-launch architecture where an Ares-V launch vehicle delivers the Altair lander to LEO prior to the Ares-I launch vehicle delivering the crew of four in the Orion Crew Exploration Vehicle (CEV) for LEO rendezvous. The mission follows trans-Lunar injection to low Lunar polar orbit, crew transfer to the Altair, Lunar descent, a seven-day surface exploration at the Lunar South Pole with sample collection, Lunar ascent, rendezvous and crew transfer to the Orion in low Lunar polar orbit, and finally trans-Earth injection and Earth entry interface leading to Pacific ocean splashdown. Analysis verifies propulsive feasibility of the mission architecture and allows for trades on element design and operations including resource supply and sample return.

Quick start scenario 2 evaluates logistical feasibility for a campaign of four missions to the Lunar South Pole. The scenario uses flight transports that abstract detail of in-space propulsion in favor of mass and crew capacity limits. Flight edges define separate trajectories for crew and cargo supply and crew return. The campaign considers four missions over a 13-month period: (1) an automated check-out mission that pre-positions supplies at the Lunar South Pole, (2) a 7-day surface exploration with four crew members, (3) a cargo mission to supply the Lunar habitat, and (4) a 14-day surface exploration with four crew members. Demand analysis estimates resources required to sustain the crew members and exploration activities. Packing and manifesting analysis packages demands in logistics containers and assigns to flights for delivery, highlighting pre-positioning strategies and cargo constraints on mass, volume, and environment.

This paper demonstrates the updated SpaceNet scenario specification by developing a quick start scenario 3 based on notional plans for the Artemis 3 mission. Compared to quick start 1 and 2, the new scenario requires modeling of the near rectilinear halo orbit about Earth-Moon L2 point, pre-positioning of a fueled commercial Human Landing System prior to the mission, and transfers between the halo orbit and low Lunar orbit before descent and after ascent.

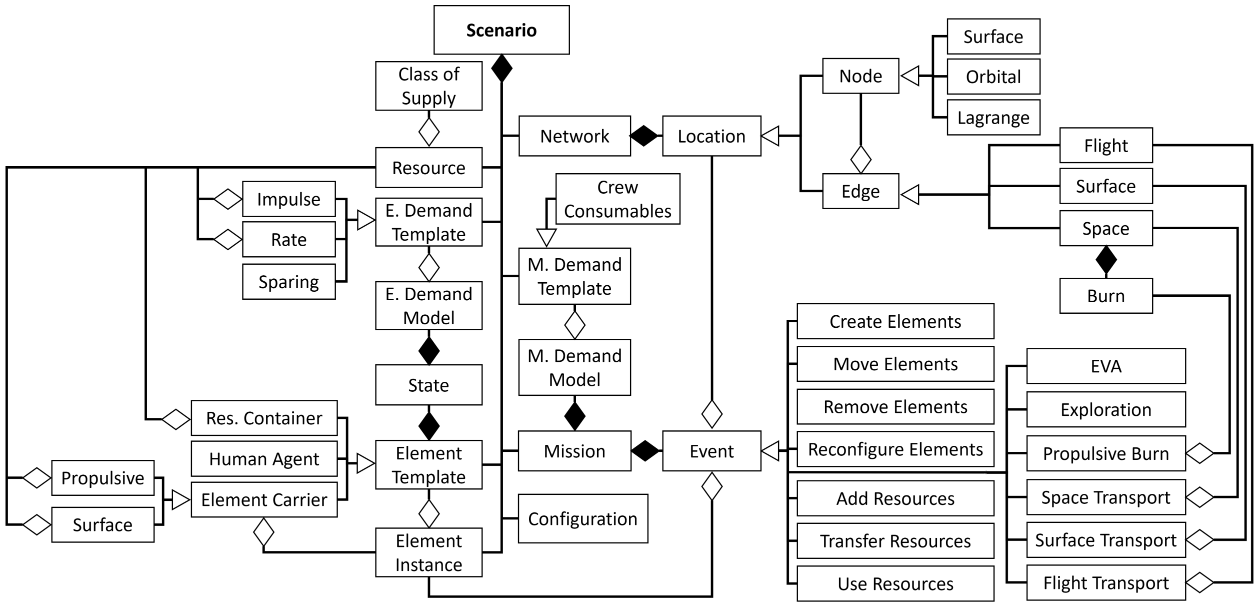


Fig. 2 Object class diagram (simplified) showing the composition of objects within a SpaceNet scenario.

III. SpaceNet Scenario Specification

A SpaceNet scenario consists of an exploration network with stable locations (nodes) and valid trajectories (edges), an internal database of campaign objects including resources, element templates, and demand model templates, a list of element instances participating in the campaign, a list of mission events that comprise the campaign behavior, and configuration settings to customize analysis behavior. Figure 2 shows a simplified object-class diagram to illustrate interrelationships among scenario objects. Black diamonds indicate composition (i.e., whole-part), white diamonds indicate aggregation (usually references), and white triangles indicate subclass specialization. The following sections describe the major components.

A. Exploration Network

The exploration network consists of time-invariant nodes and edges of various types. Nodes and edges serve as valid locations for mission elements. Nodes represent stable locations while edges represent trajectories between origin-destination node pairs. Co-located elements freely share resources. Scenarios with barriers to resource exchange (e.g., separate functional segments of a facility) must be modeled as separate locations to preserve resource independence. Each network location receives a unique identifier for reference elsewhere in the scenario.

Nodes represent stable spatial locations at which exploration activities take place. Node types include the surface of planetary bodies, in orbit around a body, or at a Lagrange point between two bodies (including in orbit about a Lagrange point). Additional properties such as surface latitude/longitude and orbit periapsis/apoapsis contribute to network visualization layout rather than physics-based calculations.

Edges represent available links between nodes. Edge types include surface paths, propulsive trajectories, and flight trajectories that abstract details of propulsive maneuvers. Surface edges specify the traversal distance between origin and destination. Space edges specify a list of timed impulsive maneuvers based on required ΔV and overall duration. Flight edges specify constraints on maximum cargo mass and crew size for established transportation architectures.

There typically exist a plurality of valid trajectories for space missions with tradeoffs between departure time, time-of-flight, and ΔV due to orbital geometry. SpaceNet does not compute or automatically select among a family of trajectories; rather, the campaign planner uses typical or reference values for space trajectories. Consequently, logistics analysis in SpaceNet is suitable for conceptual design activities rather than detailed design or operations management.

B. Campaign Objects

Campaign objects within the scenario define resources, reusable templates for demand models and elements, and element instances that participate in the exploration campaign. The internal object database maintains all necessary

Table 1 Selected Functional Classes of Supply with Default Specific Volume

COS	Description	(L/kg)	COS	Description	(L/kg)
0	None (used for Crew Members)		3	Crew Operations	5.0
1	Propellants and Fuels	1.9	301	Office Equipment and Supplies	5.0
101	Cryogens	1.9	302	EVA Equipment and Consumables	5.0
102	Hypergols	1.9	303	Health Equipment and Consumables	5.0
103	Nuclear Fuel	1.9	304	Safety Equipment	5.0
104	Petroleum Fuels	1.9	305	Communications Equipment	5.0
105	Other Fuels	1.9	306	Computers and Support Equipment	5.0
2	Crew Provisions	7.0	4	Maintenance and Upkeep	5.0
201	Water and Support Equipment	1.0	5	Stowage and Restraint	7.0
202	Food and Support Equipment	7.0	7	Waste and Disposal	5.0
203	Gases	7.0	701	Waste	5.0
204	Hygiene Items	7.0	702	Waste Management Equipment	5.0
205	Clothing	7.0	8	Habitation and Infrastructure	3.5
206	Personal Items	7.0	9	Transportation and Carriers	3.5
			10	Miscellaneous	7.0

information in the scenario, rather than relying on external spreadsheet databases as in prior versions of SpaceNet.

1. Resources

Resources define fungible substances that are demanded, produced, or supplied during a mission. Custom resources within a scenario are assigned a unique identifier, name, class of supply, unit mass, unit volume (i.e., specific volume if unit mass is 1.0 kg), stowage environment (pressurized or unpressurized), and packing factor (mass of packing material required per unit of resource). The unique identifier is referenced in demand models or events elsewhere in the scenario.

Classes of supply provide a hierarchical scheme for resource classification based on functional use [9]. Table 1 shows selected classes of supply defined in SpaceNet. Generic resources associated with a class of supply can be used in lieu of a custom resource. A generic resource has unit mass 1.0 kg, specific volume listed in Table 1, and packing factors set in the scenario configuration for gases (COS 203), water (COS 201), all other unpressurized resources, and all other pressurized resources.

2. Demand Models

Demand model templates define default properties for demand models used by elements or missions to determine campaign resource needs. Each demand model template has a unique identifier, name, description, and additional type-specific properties. When used within an element or mission, instantiated demand models reference a template's unique identifier and optionally redefine any properties.

Element demand model templates consist of impulse (one-time demand for one or more resources), rated (constant demand rate for one or more resources), and sparing-by-mass types. Sparing-by-mass sets demand rates for pressurized and unpressurized generic COS 4 resource based on a fraction of the element's (dry) mass per year.

Mission demand model templates consist of impulse, rated, and crew consumables types. Impulse and rated demand models behave similarly as element variants to generate one-time or constant rate resource demands. The crew consumables model, described in Table 2, estimates resource demands to support human exploration which are aggregated to the mission destination upon arrival. SpaceNet infers several parameters (italics in Table 2) from the scenario itself including exploration duration (elapsed time between arrival at destination and departure from destination), transit duration (elapsed time from mission start to arrival at destination plus elapsed time from destination departure to mission end), crew size (number of human agents created), and EVA crew time (total elapsed crew time of all EVA events). Other input parameters (bold in Table 2) are user-definable with default values provided in Table 3.

Table 2 Crew Consumables Model Generic Resource Demands

COS	Generic Resource Demand Calculation (*Inferred* and **Input** Parameters)
201	(*Expl. Duration* + *Transit Duration* × (1 − **Omit Transit**) + **Reserves Duration**) × **Water Rate** × *Crew Size* × (1 − **Water Recovery Rate**) + **EVA Water Rate** × *EVA Crew Time*
202	(*Expl. Duration* + *Transit Duration* × (1 − **Omit Transit**) + **Reserves Duration**) × (**Food Support Rate** + **Ambient Food Rate** + **Refrigerated Food Rate**) × *Crew Size*
203	(*Expl. Duration* + *Transit Duration* × (1 − **Omit Transit**) + **Reserves Duration**) × (**Oxygen Rate** + **Nitrogen Rate**) × *Crew Size* + **EVA Oxygen Rate** × *EVA Crew Time*
204	(**Hygiene Kit** + (*Expl. Duration* + *Transit Duration* + **Reserves Duration**) × **Hygiene Rate**) × *Crew Size*
205	(*Expl. Duration* + *Transit Duration* × (1 − **Omit Transit**) + **Reserves Duration**) × **Clothing Rate** × *Crew Size* / **Clothing Lifetime**
206	**Personal Items** × *Crew Size*
301	**Office Equipment** × *Crew Size*
302	**EVA Suit** × *Crew Size* + **EVA Lithium Hydroxide** × *EVA Crew Time*
303	**Health Equipment** + **Health Consumables** × *Crew Size*
304	**Safety Equipment**
305	**Communications Equipment**
306	**Computer Equipment** × *Crew Size*
701	(*Expl. Duration* + *Transit Duration* × (1 − **Omit Transit**)) × **Trash Bag Rate** × *Crew Size*
702	(*Expl. Duration* + *Transit Duration* × (1 − **Omit Transit**) + **Reserves Duration**) × **Waste Containment Rate** × *Crew Size*

Table 3 Crew Consumables Model Default Input Parameter Values

Input Parameter	Default Value	Input Parameter	Default Value
Reserves Duration	0 days	Hygiene Kit	1.8 kg/person
Water Recovery Rate	0.42	Clothing Rate	2.3 kg/person/day
Clothing Lifetime	4 days	Personal Items	10 kg/person
Omit Transit	0	Office Equipment	5 kg/person
Water Rate	3.6 kg/person/day	EVA Suit	107 kg/person
EVA Water Rate	0.6875 kg/person/hr	EVA Lithium Hydroxide	0.3625 kg/person/hr
Food Support Rate	0.05556 kg/person/day	Health Equipment	20 kg
Ambient Food Rate	0.76389 kg/person/day	Health Consumables	0.1 kg/person
Refrigerated Food Rate	1.61667 kg/person/day	Safety Equipment	25 kg
Oxygen Rate	3.85714 kg/person/day	Communications Equipment	20 kg
EVA Oxygen Rate	0.07875 kg/person/hr	Computer Equipment	5 kg/person
Nitrogen Rate	2.21419 kg/person/day	Trash Bag Rate	0.05 kg/person/day
Hygiene Rate	0.27778 kg/person/day	Waste Containment Rate	0.05 kg/person/day

Table 4 Generic Resource Containers with Default Property Values

Name	COS	Mass (kg)	Volume (L)	Mass Capacity (kg)	Volume Capacity (L)
Small Liquid Tank	201	11.4567	24.9	24.9333	24.9
Large Liquid Tank	201	34.37	74.8	74.8	74.8
Small Gas Tank	203	10.8	275	10	275
Large Gas Tank	203	108	2750	100	2750
Cargo Transfer Bag	all others	0.83	53	26.8	49

3. Elements

Element templates define default properties for persistent entities that participate in exploration activities. Base element properties include a unique identifier, name, description, class of supply, stowage environment (pressurized or unpressurized), accommodation mass (additional COS 5 required for stowage), mass, volume, and list of operational states. Operational states include a name, description, type (active, quiescent, dormant, or decommissioned), and list of demand model instances that drive resource consumption. As mentioned above, demand model instances reference an existing demand model template, redefining any properties to be changed for the unique element instance. A hierarchy of extended element types contribute additional properties.

Human agent elements (i.e., crew members) extend the base element definition to include an available time fraction (fraction of time available for labor, e.g., maintenance). Human agent elements enforce special cargo constraints and count towards inferred properties for crew size in the consumables model.

Resource container elements extend the base element definition to add constraints on maximum cargo mass, maximum cargo volume, and cargo environment (pressurized or unpressurized) and specify resource contents. SpaceNet uses generic resource containers listed in Table 4 during manifesting activities to package generated demands.

Element carrier elements extend the base element definition to add constraints on maximum cargo mass, maximum cargo volume, cargo environment, and number of crew members and specify element contents (by unique identifier). Two additional extensions to element carriers define vehicles to traverse edges. Propulsive vehicles have the ability to perform impulsive burns to achieve target ΔV to traverse space edges and have properties for specific impulse, constraints on maximum fuel amount, and a fuel resource amount. Surface vehicles have the ability to traverse surface edges and have properties for maximum speed, constraints on maximum fuel amount, and a fuel resource amount.

Instantiated elements can override default template properties for unique objects (e.g., serial number-specific launch vehicles). Only properties to be changed from the default value must be specified. Instantiated elements receive a unique identifier to be referenced from mission events in which they participate.

C. Mission Events

A campaign comprises a sequence of missions that capture space exploration activities within the simulation state. The initial state consists of the (empty) exploration network. Each mission consists of a start date, origin and destination nodes (to infer parameters for the crew consumables model), and sequence of events to enforce state changes. Each event is specified by an execution time (relative to the mission start), priority (between 1–5 to sequence coincident events), location (node or edge), and properties that describe the state change. SpaceNet defines seven low-level (core) events that model instantaneous state transitions and six composite processes that schedule low-level events to represent more complex behavior.

Four low-level events act on elements. The create elements event adds instantiated elements to an existing carrier or location in the exploration network. The move elements event moves existing instantiated elements to an existing carrier or a different location in the exploration network. The remove elements event permanently removes existing instantiated elements from the exploration network. Finally, the reconfigure elements event changes the current operational state to a new type (active, quiescent, decommissioned, or dormant) to trigger different demand models. Spatial errors arise at execution time if an instantiated element to be created already exists, an instantiated element's location does not match the event location, the designated carrier does not have enough capacity to store the elements, or the instantiated element does not have a designated state.

Three low-level events act on resources. The add resources event adds resources to an existing container. The transfer resources event moves resources from an existing origin to an existing co-located destination container. The consume resources event removes resources from an existing container. All resource containers are identified by unique identifier. Spatial errors arise at execution time if a container's location does not match the specified event location, a receiving container does not have enough capacity to receive the resources, or a providing container does not have enough contents to supply the resources.

The crewed EVA event models extra-vehicular activity by crew members. It specifies the EVA duration, the crew members, the carrier from which they egress and ingress, each crew member's operational state for EVA activity, and a list of additional resource demands (e.g., makeup gases for airlock cycling). At execution time, it schedules a reconfigure elements event to adopt EVA operational states, a move elements event for egress from the carrier, a move elements event for ingress into the carrier, a reconfigure elements event to restore the original operational state, and a consume resources event to generate additional demands. Spatial errors arise at execution time if a crew member is not inside the carrier or a crew member's or carrier's location does not match the event location.

The exploration event models a series of crewed EVAs over an extended mission period at one location. In addition to the EVA properties described above, it also specifies the total exploration duration and number of EVAs to conduct per week. At execution time, it schedules a series of crewed EVAs (rounded down) during the exploration period where the first EVA concides with the start of the exploration period.

The propulsive burn event executes a sequence of impulsive burn and stage actions to achieve a target ΔV (associated with a space edge) considering the total stack mass of participating existing elements. Burn actions generate a consume resources event to remove propellant from a designated propulsive vehicle with quantity determined by the ideal rocket equation in Eq. (1). If there is insufficient propellant to achieve the target ΔV, the achieved ΔV in Eq. (2) is subtracted from the target ΔV. Stage actions generate a remove element event to discard components of the stack, thereby reducing the stack mass for subsequent burn events. Spatial errors arise at execution time if the burn and stage actions do not achieve the full target ΔV, or if an element's location does not match the event location.

$$m_{\text{burn}} = \min\left(m_{\text{available}}, m_{\text{stack}}\left(1 - \exp\left(\frac{-\Delta V_{\text{target}}}{I_{\text{sp}} g_0}\right)\right)\right) \tag{1}$$

$$\Delta V_{\text{achieved}} = I_{\text{sp}} g_0 \ln\left(\frac{m_{\text{stack}}}{m_{\text{stack}} - m_{\text{burn}}}\right) \tag{2}$$

The space transport event models space edge traversal using a series of propulsive burns. It specifies the edge to traverse, the existing elements to transport, and the list of burn-stage actions for each requisite burn. For example, a space edge with two required burns requires two sequences of burn-stage actions. At execution time, it schedules a move elements event to transfer elements from the origin node to the space edge, propulsive burn events to achieve each of the edge's required ΔV, and a move elements event to transfer the (remaining) elements to the destination node.

The flight transport event models edge traversal at a higher level of abstraction than a space edge. It specifies the flight edge to traverse and the list of existing elements to transport. At execution time, it schedules move elements events to transfer elements from the origin node to the flight edge and, later, from the flight edge to the destination node. Spatial errors arise at execution time if the total element mass or total number of crew members exceeds flight edge constraints.

The surface transport event models surface edge traversal. It specifies the edge to traverse, an existing surface vehicle to perform the transport, the transport speed, the duty cycle (fraction of time the vehicle is moving), and an operational state during transport. At execution time, it schedules a reconfigure elements event to adopt the transport state, move elements events to transfer the vehicle from the origin node to the surface edge and, later, from the edge to the destination node, and finally a reconfigure elements state to restore the original operational state.

D. Configuration Settings

Configuration settings listed in Table 5 define the mass, volume, and time precision to be enforced in the simulation, toggle volume and environmental constraints for packing and stowage operations, and set default packing factors for gas, liquid, solid pressurized, and solid unpressurized resources. Additional configuration settings set the generic resource container properties listed in Table 4. Environment constraints require pressurized elements be stored inside carriers with a pressurized cargo environment and unpressurized elements be stored inside carriers with an unpressurized cargo environment. Demands analysis uses packing factors to estimate packing factors prior to detailed manifesting.

E. Serialization

The SpaceNet scenario is serialized (i.e., written to file) using JSON, a lightweight data interchange format that is easy for both humans and machines to read and write. The JSON scenario format is specified using JSON Schema, a

Table 5 Configuration Settings

Name	Description	Default
Time Precision	Minimum time duration (days) considered to be non-zero	0.05
Demand Precision	Minimum resource amount (units) considered to be non-zero	0.01
Mass Precision	Minimum mass (kg) considered non-zero	0.01
Volume Precision	Minimum volume (L) considered non-zero	1.0
Volume Constrained	Enforce volume constraints for containers and carriers	False
Environment Constrained	Enforce environment constraints for containers and carriers	False
Liquid Packing Factor	Mass of COS 5 packing material per kg of COS 201	0.5
Gas Packing Factor	Mass of COS 5 packing material per kg of COS 203	1.0
Pressurized Packing Factor	Mass of COS 5 packing material per kg of pressurized resources	0.2
Unpressurized Packing Factor	Mass of COS 5 packing material per kg of unpressurized resources	0.6

Table 6 Quick Start Scenario 3 Nodes

Abbrev.	Description	Details
KSC	Kennedy Space Center	Earth Surface: 28.6°N, 80.6°W
LEO	Low Earth Orbit	Earth Orbit: 300 km circular
NRO	Near Rectilinear Halo Orbit	Lagrange Point: Earth-Moon L2
LLO	Low Lunar Orbit	Lunar Orbit: 100 km circular
LSP	Lunar South Pole	Lunar Surface: 89°S, 180°W
PAC	Pacific Ocean Splashdown	Earth Surface: 35.0°N, 117.9°W

declarative language that annotates and validates JSON documents [10]. SpaceNet Cloud generates a JSON Schema specification from Pydantic version 1.10.13, a Python-language validation library using type annotations [11]. Object schemas defined in Python can also serve as object classes for script-based automated scenario generation.

Scenarios can be defined programmatically using the SpaceNet Cloud Python-language library and automatically serialized to JSON format for use with SpaceNet Java. A script-based workflow allows for easier automation and verification of scenario completeness and correctness. Pre-processor tools, for example Excel macros and Visual Basic scripts can likewise export compatible JSON documents.

IV. Example Use Case: Quick Start Scenario 3

This scenario models a Lunar surface mission based on notional plans for Artemis 3 (previously Exploration Mission 3, EM-3) that focuses on propulsive feasibility of launch and in-space trajectories in the cis-Lunar space. The mission considers Earth launch of a crew of four and rendezvous with a pre-positioned lander in a near rectilinear halo orbit about the Earth-Moon L2 point. Two crew members transfer to the lander while the other two remain in the crew module. The crewed lander transfers to a low-Lunar orbit before descending to the Lunar surface to conduct a 7-day exploration, collecting samples. Afterwards, the surface crew ascend to low-Lunar orbit, transfer to the halo orbit, and rendezvous and transfer to the command/service module. Finally, the crew return via a trans-Earth trajectory and splashdown in the Pacific Ocean. Each transport requires propulsive vehicles to achieve target ΔV values by burning propellant. No other crew demands are considered to simplify the early-stage mission concept.

A. Exploration Network

Figure 3 illustrates the exploration network consisting of nodes and edges in the cis-Lunar space. Table 6 lists network nodes including Kennedy Space Center (KSC) for launch, low-Earth orbit (LEO) for Earth staging, a near rectilinear halo orbit (NRO) for Lunar staging, low-Lunar orbit (LLO), the Lunar South Pole (LSP) for surface exploration, and the Pacific Ocean Splashdown (PAC) for return after Earth entry interface.

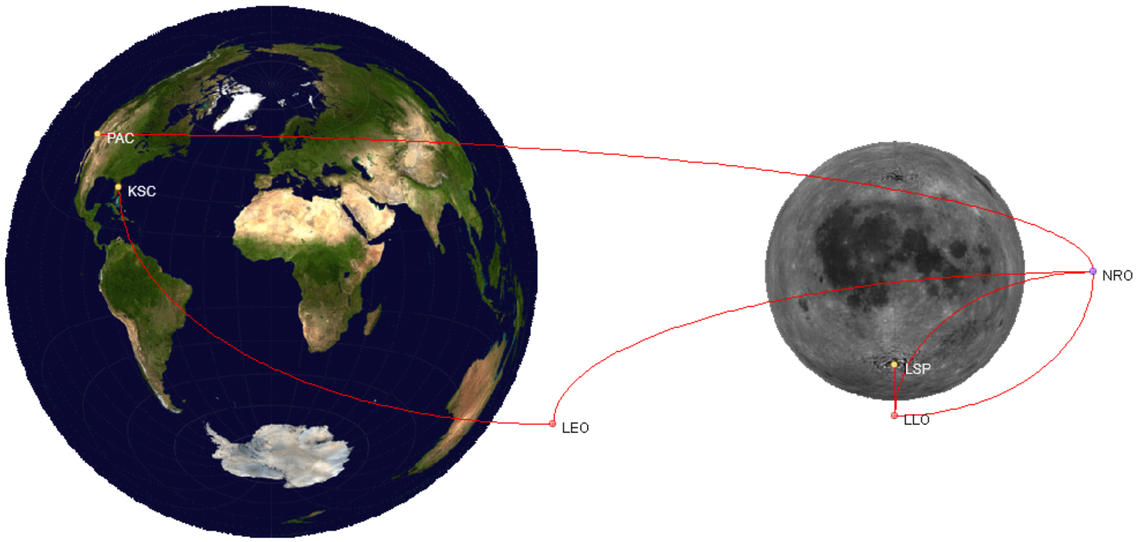


Fig. 3 Quick start scenario 3 exploration network locations in the cis-Lunar space.

Table 7 Quick Start Scenario 3 Space Edges

Origin-Destination	Description	Duration (days)	Burns: (Order), Time (days), ΔV (m/s)
KSC-LEO	Earth Ascent	0.2	(1) 0.0, 9500
LEO-NRO	Trans-Lunar Injection	7.0	(1) 0.0, 3124; (2) 0.3, 30; (3) 3.1, 242; (4) 4.4, 1.0; (5) 6.4, 126
NRO-LLO	Transfer to Low Lunar Orbit	0.5	(1) 0.0, 20; (2) 0.5, 648.6
LLO-LSP	Lunar Descent	0.2	(1) 0.0: 19.4; (2) 0.2, 1692.5
LSP-LLO	Lunar Ascent	0.2	(1) 0.0, 1692.4; (2) 0.2, 19.4
LLO-NRO	Transfer to Halo Orbit	0.5	(1) 0.0, 649.2; (2) 0.5, 125.1
NRO-PAC	Earth Return	6.4	(1) 0.0, 147; (2) 1.6, 2; (3) 2.8, 263

Table 7 shows scenario edges. Launch from KSC to LEO is modeled as a single burn of 9500 m/s. Transfer from LEO to NRO is modeled by a 3124 m/s trans-Lunar injection burn, 30 m/s hybrid burn, 242 m/s perilune burn, 1 m/s midcourse burn, and 126 m/s NRO insertion burn [12, 13]. Mission operations between NRO and the Moon are modeled based on Ref. [14]. Transfer from NRO to LLO is modeled by a 126.4 m/s departure burn and 648.6 m/s Lunar orbit insertion burn. Landing from LLO to LSP is modeled with a 19.4 m/s departure burn followed by a 1692.5 m/s landing burn. Similarly, ascent from LSP to LLO is estimated at 1692.4 m/s followed by a 19.4 m/s Lunar orbit insertion burn. Finally, transfer from LLO to NRO is modeled by a 649.2 m/s departure burn and 125.1 m/s arrival burn. Return to Earth is modeled with a 147 m/s halo orbit departure burn, 2 m/s midcourse correction burn, and 263 m/s perilune burn [13].

B. Campaign Objects

1. Resources

The scenario considers four custom propellant resources: polybutadiene acrylonitrile (PBAN, COS 105), liquid oxygen/liquid hydrogen (LOX/LH2, COS 101), liquid oxygen/liquid methane (LOX/LCH4, COS 101), and monomethyl-hyrdazine (N204/MMH, COS 102). All resources are defined with a 1.0 kg unit mass and specific volume based on inverse density (note that propellant constraints are exclusively determined by mass in SpaceNet).

Table 8 Quick Start Scenario 3 Element Templates

Abbrev.	Description	Mass (kg)	Cargo (kg)	Crew	Fuel	Fuel (kg)	I_{sp} (s)
SRB	Solid Rocket Boosters (2)	195,000	0	0	PBAN	1,256,000	269
CS	Core Stage	88,275	0	0	LOX/LH2	987,000	414
US	Upper Stage	3,490	0	0	LOX/LH2	28,576	465
SA	Spacecraft Adapter	1,900	0	0	–	–	–
CM	Crew Module	9,300	1,100	4	–	–	–
SM	Service Module	6,185	0	0	N204/MMH	9,276	316
LAS	Launch Abort System	7,250	0	0	–	–	–
HLS	Human Landing System	8,149	900	2	LOX/LCH4	32,285	363
C	Crew Member	100	–	–	–	–	–
S	Lunar Samples	100	–	–	–	–	–

2. Elements

Scenario elements consider a launch system modeled after the Space Launch System (SLS) Block 1 including two solid rocket boosters with RSRM-5 engines, a core stage with RS-25D engines, and an Interim Cryogenic Propulsion System (ICPS) upper stage with a RL10B-2 engine, crew and service modules modeled after the Orion Multi-purpose Crew Vehicle (MPCV) with European Service Module (ESM), and human landing system modeled after the single-stage LOX/LCH4 architecture in Ref. [15] with cargo capacity based on Ref. [16]. Table 8 provides key element templates properties. Data are subject to considerable uncertainty during mission design and development and should be considered as a relative baseline from which sensitivity analysis or trades can be performed rather than absolute values.

The scenario uses one instance of each element template except for four crew member instances. Element template abbreviations with a dot notation indicate element instances. For example, C.1 represents the first crew member instance.

C. Mission Events

The scenario defines one mission with 14 events. The mission nominally starts in July 2025 with an origin at KSC, destination at LSP, return origin at LSP, and return destination at PAC. Notably, this mission assumes the human landing system is pre-positioned in NRO which is accomplished in the first event. Table 9 describes the mission events based on relative time, location, type, and details. Figure 4 illustrates a time-expanded network diagram (described as a "bat" chart) visualizing the time and location of events and transportation processes.

D. Analysis and Discussion

This quick start scenario models a seven-day Lunar surface mission that spans nearly four weeks (27.6 days) of simulated time. The modeling focus only considers propulsive feasibility: no crew member demands or supplies are considered; however, the mission retrieves and returns to Earth 100 kg of lunar samples from the exploration site.

Scenario analysis is limited to propulsive feasibility using SpaceNet Java to simulate the single-mission campaign. The quick start scenario does not produce any simulation errors during execution, indicating closure of propulsive logistics. Results in Table 10 show residual fuel amounts for propulsive vehicles.

Analysis of the residual fuel shows that, under baseline assumptions, the launch vehicle is close to its total lift capability, delivering approximately 57.2 t to LEO with a 2% margin. Similarly, the human landing system is close to its total round-trip Lunar surface capability, landing approximately 32.5 t on the Lunar surface with a 5% propellant margin. The service module retains a large margin (37%) indicating the transfer to and from NRO has could carry additional resources, subject to the initial Earth ascent launch vehicle bottleneck (n.b., a more detailed scenario could investigate having the service module contribute a larger portion of the trans-Lunar injection ΔV).

Open questions for further analysis could broaden the initial campaign boundary to also consider mission operations required to pre-position the human landing system at NRO including Earth ascent, propellant stockpiling, and propellant transfer from on-orbit depots. Cryogenic propellant boiloff presents a major concern that should be considered. Additionally, this analysis did not consider crew provisions or other operational demands that would introduce additional mass to be transported along the interplanetary supply chain.

Table 9 Quick Start Scenario 3 Mission Events

Time	Location	Type	Name	Details
0.0	NRO	Create Elements	Pre-position Lander	Create HLS.1 in NRO
0.0	KSC	Create Elements	Create Launch Stack	Create SRB.1, CS.1, SA.1, US.1, CM.1, SM.1, LAS.1 in KSC
0.0	KSC	Create Elements	Create Crew	Create C.1, C.2, C.3, C.4 in CM.1
0.0	KSC	Space Transport	Earth Ascent	Transport SRB.1, CS.1, SA.1, US.1, CM.1, SM.1, LAS.1 on KSC-LEO with burn sequence: burn SRB.1, stage SRB.1, stage LAS.1, burn CS.1, stage CS.1, stage SA.1
1.0	LEO	Space Transport	Trans-Lunar Injection	Transport US.1, CM.1, SM.1 on LEO-NRO with burn sequences: (1) burn US.1, stage US.1; (2) burn SM.1; (3) burn SM.1; (4) burn SM.1; (5) burn SM.1
8.0	NRO	Move Elements	Rendezvous, Crew Transfer	Move C.1, C.2 to HLS.1
9.0	NRO	Space Transport	Transfer to Low Lunar Orbit	Transport HLS.1 on NRO-LLO with burn sequences: (1) burn HLS.1; (2) burn HLS.1
10.0	LLO	Space Transport	Lunar Descent	Transport HLS.1 on LLO-LSP with burn sequence: (1) burn HLS.1; (2) burn HLS.1
11.0	LSP	Exploration	Surface Exploration	Explore with C.1 and C.2 for 7.0 days with 5 EVAs/week (8 hr duration)
18.0	LSP	Create Elements	Create Samples	Create S.1 in HLS.1
18.0	LSP	Space Transport	Lunar Ascent	Transport HLS.1 on LSP-LLO with burn sequence: (1) burn HLS.1; (2) burn HLS.1
19.0	LLO	Space Transport	Transfer to Halo Orbit	Transport HLS.1 on LLO-NRO with burn sequence: (1) burn HLS.1; (2) burn HLS.1
20.0	NRO	Move Elements	Rendezvous, Crew Transfer	Move C.1, C.2, S.1 to CM.1
21.0	NRO	Space Transport	Trans-Earth Injection	Transport CM.1 on NRO-PAC with burn sequence: (1) burn SM.1; (2) burn SM.1; (3) burn SM.1, stage SM.1

Table 10 Quick Start Scenario 3 Propellant Margin

Element	Initial Fuel (kg)	Final Fuel (kg)	Margin (%)
SRB.1	1256000	0.0	0%
CS.1	987000	20806.6	2.1%
US.1	28576	201.2	0.7%
HLS.1	32285	1656	5.1%
SM.1	9276	3471.7	37.4%

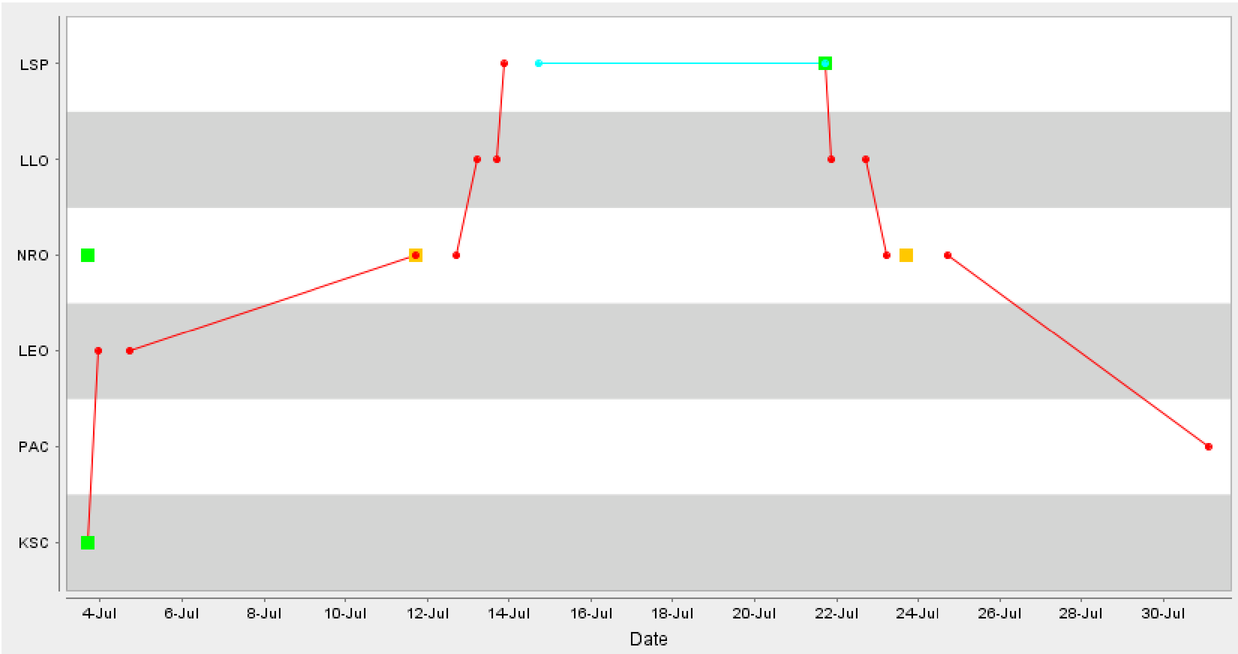


Fig. 4 Quick start scenario 3 bat chart showing transport of elements across a time-expanded network.

V. Conclusion

Human space exploration campaigns in the cis-Lunar space and beyond exhibit widespread activities across remote sites operated by multiple agencies including government, international, and commercial organizations. Space logistics campaign planning is essential to develop feasible plans for propulsive maneuvers and resource management and, in particular, coordinate plans across organizational boundaries.

This paper describes the SpaceNet scenario specification consisting of an exploration network, campaign objects (resources, demand mdoels, and elements), and mission events. A SpaceNet scenario can be serialized to JSON format for interoperability between computing platforms including modeling and scripting in SpaceNet Cloud and analysis and visualization in SpaceNet Java. Quick start scenario 3 models a lunar sortie mission following the Artemis mission architecture, verifying initial propulsive feasibility.

Future applications of SpaceNet scenario development to other Artemis program elements include build-up and steady-state operation of remote facilities such as the Lunar Gateway and Artemis Base Camp. Campaign logistics analysis can address significant planning and coordination challenges to deliver and integrate new modules and infrastructure at remote sites, establish an inter-planetary supply chain of goods and services, and manage scarce shared resources such as docking ports among multiple participating partners.

References

[1] Capra, L., Hilton, J., Bentley, S., Sherman, T., Alfaro, A., Savin, R., de Weck, O. L., and Grogan, P. T., "SpaceNet Cloud: Web-based Modeling and Simulation Analysis for Space Exploration Logistics," *ASCEND*, AIAA, Las Vegas, NV, 2021. https://doi.org/10.2514/6.2021-4068.

[2] Grogan, P. T., Yue, H. K., and de Weck, O. L., "Space Logistics Modeling and Simulation Analysis using SpaceNet: Four Application Cases," *AIAA SPACE 2011 Conference & Exposition*, AIAA, Long Beach, CA, 2011. https://doi.org/10.2514/6.2011-7346.

[3] de Weck, O. L., Simchi-Levi, D., Shishko, R., Ahn, J., Gralla, E. L., Klabjan, D., Mellein, J., Shull, S. A., Siddiqi, A., Bairstow, B. K., and Lee, G. Y., "SpaceNet v1.3 User's Guide," Tech. Rep. NASA/TP-2007-214725, Jan 2007.

[4] de Weck, O., Armar, N., Grogan, P., Siddiqi, A., Lee, G., Jordan, E., and Shishko, R., "A flexible architecture and object-oriented model for space logistics simulation," *AIAA Space 2009 Conference & Exposition*, AIAA, Pasadena, CA, 2009. https://doi.org/10.2514/6.2009-6548.

[5] Grogan, P. T., "A Flexible, Modular Approach to Integrated Space Exploration Campaign Logistics Modeling, Simulation, and Analysis," Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 9 2010. URL https://dspace.mit.edu/handle/1721.1/62317.

[6] Ho, K., de Weck, O. L., Hoffman, J. A., and Shishko, R., "Campaign-level dynamic network modelling for spaceflight logistics for the flexible path concept," *Acta Astronautica*, Vol. 123, 2016, pp. 51–61. https://doi.org/10.1016/j.actaastro.2016.03.006.

[7] Massachusetts Institute of Technology, "SpaceNet Java v.2.5.1469," , 2023. URL https://github.com/space-logistics-org/spacenet-java.

[8] Massachusetts Institute of Technology, "SpaceNet Cloud," , 2023. URL https://github.com/space-logistics-org/spacenet.

[9] Gralla, E. L., Shull, S., and de Weck, O., "A Modeling Framework for Interplanetary Supply Chains," *Space 2006*, San Jose, CA, 2006. https://doi.org/10.2514/6.2006-7229.

[10] Internet Engineering Task Force, "JSON Schema: A Media Type for Describing JSON Documents," , 2022. URL https://json-schema.org/draft/2020-12/json-schema-core.

[11] Colvin, S., "Pydantic v1.10.13," , 2023. URL https://github.com/pydantic/pydantic.

[12] Zimovan, E. M., Howell, K. C., and Davis, D. C., "Near Rectilinear Halo Orbits and their Application in Cis-Lunar Space," *3rd IAA Conference on Dynamics and Control of Space Systems*, Moscow, Russia, 2017. https://doi.org/10.2514/6.2011-7346.

[13] Pratt, W., Buxton, C., Hall, S., Hopkins, J., and Scott, A., "Trajectory Design Consideration for Human Missions to Explore the Lunar Farside from the Earth-Moon Lagrange Point EM-L2," *AIAA SPACE Forum*, San Diego, CA, 2013. https://doi.org/10.2514/6.2013-5478.

[14] Condon, G. L., Esty, C. C., Berry, C. F., Downs, S. P., Ocampo, C., Mahajan, B., and Burke, L. M., "Mission and Trajectory Design Considerations for a Human Lunar Mission Originating from a Near Rectilinear Halo Orbit," *AIAA SciTech Forum*, Orlando, FL, 2020. https://doi.org/10.2514/6.2020-1921.

[15] Latyshev, K., Garzaniti, N., Crawley, E., and Golkar, A., "Lunar human landing system architecture tradespace modeling," *Acta Astronautica*, Vol. 181, 2021, pp. 352–361. https://doi.org/10.1016/j.actaastro.2021.01.015.

[16] Watson-Morgan, L., Chavers, G., Connolly, J., Crowe, K., Krupp, D., Means, L., Percy, T., Polsgrove, T., and Turpin, J., "NASA's Initial and Sustained Artemis Human Landing System," *2021 IEEE Aerospace Conference*, Big Sky, MT, 2021. https://doi.org/10.1109/AERO50100.2021.9438179.