NASA/TP—2007–214725

SpaceNet v1.3 User's Guide

Olivier L. de Weck, David Simchi-Levi
Massachusetts Institute of Technology, Cambridge, Massachusetts

Robert Shishko
Jet Propulsion Laboratory, Pasadena, California

Jaemyung Ahn, Erica L. Gralla, Diego Klabjan, Jason Mellein, Sarah A. Shull, Afreen Siddiqi
Massachusetts Institute of Technology, Cambridge, Massachusetts

Brian K. Bairstow, Gene Y. Lee
Jet Propulsion Laboratory, Pasadena, California

January 2007

NASA STI Program ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, and organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Phone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

Acknowledgments

The authors wish to thank Dr. Martin J. Steele, Kennedy Space Center, who served as the COTR and valued advisor for this work.

We thank Ms. Emily Unbehaun at NASA KSC for serving as NASA’s contracting officer.

Mr. Don Monell (NASA MSFC) provided valuable guidance and direction in terms of integration of this framework and software tool with NASA’s Integrated Modeling and Simulation program. Mr. Doug Craig at NASA Headquarters provided for continuity of the program during the period of performance.

The research was funded under NASA contract NNK05OA50C from May 2005-January 2007.

The following individuals have directly contributed to the development of SpaceNet 1.3 in addition to the listed authors: Nii Armar (MIT), Christine Taylor (MIT), Susan Brown (MIT), Xin Li (MIT), Hamed Mamani (MIT), Miao Song (MIT), Matt Silver (MIT), Christoph Meier (MIT/TUM), James Whiting (MIT), Babak Cohanim (JPL), Chester Borden (JPL), Erick Sturm (NASA JPL), Andy Evans (USA), Julie Finn (Administrative Assistant MIT), Peggy Bryan (Fiscal Officer MIT), Angela Olsen (Fiscal Officer MIT), Michael Corcoran (MIT Office of Sponsored Programs), Robert Kline (LMI)

Testing of SpaceNet 1.3 was conducted by: Peter Waswa (MIT), Prof. Jeffrey Hoffman (MIT), Col. (ret) John Keesee (MIT), Thomas Coffee (MIT), Ricardo Robles (MIT)

Feedback during and between review sessions was provided by: Neil Lemmons (NASA JSC), Dave Forrest (NASA JSC), Jeff George (NASA JSC), Jay Falker (NASA JSC), Michael Galluzzi (NASA KSC), Anthony Butina (NASA JSC), Kevin Watson (NASA JSC), Edgar Zapata (NASA KSC), Prof. Edward Crawley (MIT), Wilfried Hofstetter (MIT), Paul Wooster (MIT), Gergana Bounova (MIT)

Available from:

NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

This report is also available in electronic form at

http://spacelogistics.mit.edu

Requests for the SpaceNet 1.3 software program should be directed towards Dr. Martin Steele at NASA Kennedy Space Center

Version 1.3
January 31, 2007

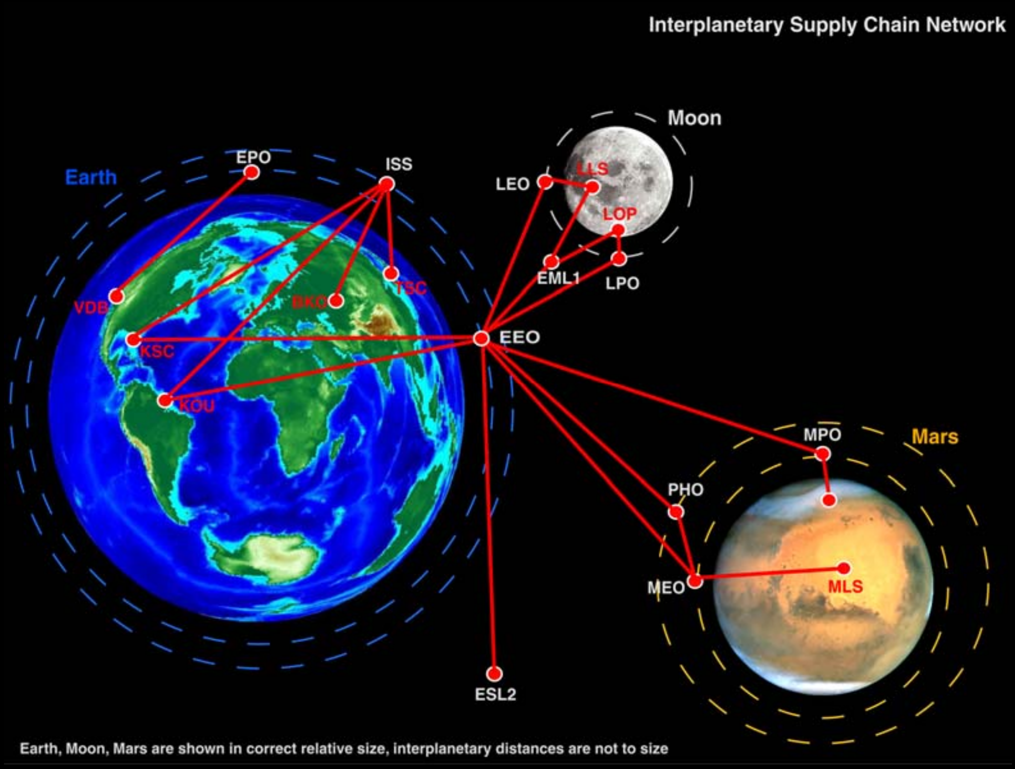
Payload Systems Inc.

Jet Propulsion Laboratory
California Institute of Technology

USA
United Space Alliance

SpaceNet v1.3 User's Guide

An Interplanetary Supply Chain Management And Logistics Planning and Simulation Software



Earth, Moon, Mars are shown in correct relative size, interplanetary distances are not to size

(this page intentionally left blank)

Executive Summary

The term “supply chain” has traditionally been used to refer to terrestrial logistics and the flow of materials and finished goods in and out of manufacturing facilities, warehouses and retail stores. Improvements in service level as well as reduced transportation and inventory holding costs have been achieved in many organizations thanks to careful supply chain design, planning and optimization. Increasingly, there is a realization that crewed space missions, such as the sustainment of the International Space Station (ISS) or the buildup and use of a lunar outpost, should not be treated as isolated missions, but rather as an integrated supply chain. This will simultaneously (i) improve exploration capability and scientific return, (ii) minimize transportation costs and (iii) reduce risks through increased system availability and resilience to failures.

SpaceNet is an integrated interplanetary supply chain management and logistics planning and simulation software tool. The goal of SpaceNet is to allow mission architects, planners, systems engineers and logisticians to focus on *what* will be needed to support future crewed exploration missions, primarily in the Earth-Moon-Mars system. Instead of helping to design the elements (vehicles) themselves in terms of propulsive and pressurized/un-pressurized cargo carrying capability, SpaceNet evaluates such vehicles in the context of a particular mission architecture and supply chain strategy. The software allows the user to specify *how* the transportation and inventory holding capacity resulting from particular mission architectures will be used in terms of various classes of supply. The most important classes of supply in space exploration are: consumables, spares and exploration items. SpaceNet allows simulating the time-varying flow of elements (vehicles), crew and supply items through the nodes and arcs (trajectories) of a supply network in space, while taking into account feasibility (ΔVs, fuel levels) as well as consumption and supply. One of the unique features of SpaceNet is that it supports not only single sortie-style missions but also detailed analysis of multi-year campaigns, where some items may be pre-positioned or re-supplied by one set of elements or crew while being used by other elements or crew. The emphasis is on ensuring logistical feasibility of a given scenario as well as a prediction of the resulting logistics measures of effectiveness (MOEs).

SpaceNet 1.3 is implemented in Matlab and exports its results directly to the user through a Graphical User Interface (GUI) and to an Excel file that can be used for post-processing analysis. Both trade study mode (the comparison of two scenarios against each other) and batch mode (automatically varying parameters within one scenario) are supported.

This User Manual explains the capabilities and limitations of SpaceNet, contains a definition of supply items, demand models and process specifications and presents the mathematics of the measures of effectiveness (MOEs). A *Quick Start* example helps to familiarize new users within about 30 minutes. A set of 5 scenarios is discussed and delivered together with the SpaceNet 1.3 release. Appendix A contains a glossary of the most important terms. Other Appendices provide useful background and reference information for advanced users.

(this page intentionally left blank)

Table of Contents

Chapter 1 Introduction

SpaceNet is an interplanetary supply chain management and logistics planning and simulation software environment. It was developed for NASA by the Massachusetts Institute of Technology[1] and the Jet Propulsion Laboratory, California Institute of Technology[2], with support from Payload Systems Inc.[3] and United Space Alliance LLC. The research described in this document was carried out under contract NNK05OA50C with the National Aeronautics and Space Administration (NASA).

Modeling Framework

One of the major challenges in the development of a space logistics model is defining the model components. Unlike terrestrial logistics, interplanetary logistics has not been previously modeled in detail, so the scope of such a model must be defined. The basic elements of the model are: transportation (shipment of crew, cargo, and vehicles), supply and demand (by supply class), information architecture, simulation and optimization. An interplanetary supply chain supporting exploration is modeled as a set of nodes (locations) and arcs (trajectories between these locations). Demand is generated over time; for example, a human mission to a lunar surface node generates demand for crew provisions, science equipment, etc. Vehicles (elements) traverse arcs carrying supplies to satisfy the demand. Users can either manually define the transportation paths through the network of nodes and arcs, or use an optimization tool to find the best solution given a particular demand scenario. One of the key enablers of SpaceNet is the *concept of time-expanded networks*. This framework provides an integrated planning and simulation tool for space logistics.

The challenge of integrating these components into a cohesive end-to-end logistics and operations model is discussed in the next sections. First, we describe the basic building blocks of the modeling framework (nodes, elements, and supplies), along with two concepts which enable us to tie these together: the time-expanded network and processes for movement through the network. Collectively, this framework allows describing and modeling both the supply/demand and the movement of all items in a logistics scenario. Finally, we describe the remaining layers that enable the effective utilization of this modeling framework: the ability to visualize, simulate, and evaluate various scenarios, as well as the application of optimization techniques.

---

[1] Principal Investigators: Prof. Olivier de Weck, Department of Aeronautics and Astronautics and the Engineering Systems Division; Prof. David Simchi-Levi, Department of Civil and Environmental Engineering and Engineering Systems Division

[2] Dr. Robert Shishko, Mission Systems Concepts, Section 312

[3] Joe Parrish, President, Payload Systems Inc.

The domain (boundary) of SpaceNet includes Earth surface points for departure (launch) and return, intermediate orbital and deep space locations as well as planetary surface nodes (Moon, Mars) for arrival, exploration and return to Earth. SpaceNet does not model the detailed operations at spaceports, ground processing and supplier management, nor does it simulate in detail the activities during surface exploration such as EVAs. The cargo demand generated by such activities, however, is modeled in detail on a node-by-node basis.

The short term vision for SpaceNet is that it will allow a more strategic approach to crewed space exploration planning by optimizing the mix of pre-deployment, carry-along and resupply flights for space exploration campaigns in the Earth, Moon and Mars system. The long term vision is that it will support campaign-level risk assessment taking into account flight delays, failures and unexpected demand as well as support of flight operations through optimized replanning and real-time asset and inventory control.

Building Blocks

This section describes the basic building blocks in the SpaceNet modeling framework: nodes, supplies, and elements. They are derived from terrestrial supply chain management and from past practices in space logistics.

Nodes: Nodes are spatial locations in the solar system. Contrary to some usages of the term, the existence of a node does not necessarily indicate that a facility exists at that location or that a node is ever used or visited. A node is simply a way to refer to locations in space. Nodes can be of three basic types: *surface nodes*, *orbital nodes*, and *Lagrangian nodes*.

Surface nodes are fairly straightforward. They exist on the surface of a central body such as the Earth, the Moon, or Mars, and they are further characterized by their latitude and longitude on that central body. Examples of surface nodes include the Kennedy Space Center (Earth 28.6N, 80.6W) or the Apollo 11 landing site at Mare Tranquilitatis (Moon 0.7N, 23.5E).

Orbital nodes are also characterized by their central body (e.g., Earth, Moon, Mars, or Sun), as well as other characteristics describing the orbit itself: *apoapsis*, *periapsis*, and *inclination*. Therefore, the ISS orbit could be an orbital node located around Earth at a circular altitude of 400 km and an inclination of 51.6 degrees. Similarly, a circular low lunar orbit (LLO) at 100 km altitude is a commonly used orbital node in lunar exploration missions.

Lagrangian nodes are located at any of the Lagrange points in the solar system. They are characterized by the two bodies and the index number of the Lagrange point. One commonly considered Lagrange point is the Earth-Moon L1 point, which lies between the Earth and the Moon (at 85% the distance towards the Moon as seen from Earth) at the point where the two bodies' gravitational pulls are balanced.

We reiterate that labeling a location as a *node* does not necessarily mean a permanent facility exists at that location. Rather, it means that some part of the logistics architecture for a space mission might make use of that location as a transit or waiting point. For example, if a spacecraft is launched from KSC to low Earth orbit (LEO), then propelled toward lunar equatorial orbit, it has passed through one surface node and two orbital nodes. The nomenclature developed around nodes allows us to build up a potential transportation network and thus to formalize descriptions of logistics architectures.

Supplies: Supplies are any items that move through the network, from node to node. Generally, supplies should include all the items needed for a sortie mission or at a planetary base, or during the journeys to and from the base. Examples include consumables, science equipment, surface vehicles, and spares. In order to track and to model the extraordinary variety of supplies that could be required, they are classified into larger categories.

We analyzed various ways to classify supplies and concluded that the best method was to develop a set of *functional classes of supply*, organized regardless of material or owner. The classes are therefore based on the essential functions for human exploration of planetary surfaces, and the associated tasks that need to be accomplished, such as research, habitation, transportation, etc. The set of ten classes of supply (COS) used in *SpaceNet* is shown in Fig.1. (NASA's Cargo Category Allocation Rate Table (CCART) classification system, presently in use for ISS logistics, was evaluated for use in this context, but it was occasionally inconsistent and was missing a number of categories required for surface exploration).

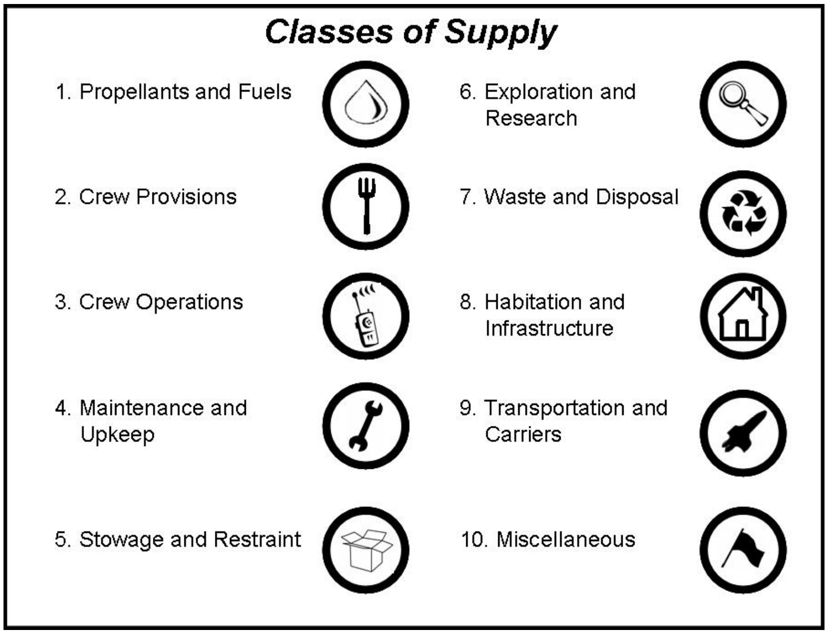


Figure 1: Functional Classes of Supply for Human Space Exploration

These classes of supply form the basis for modeling supply items. Recall that the impetus behind the development of these supply classes was the need for a manageable modeling framework for supplies moving through a space network. With these ten supply classes we can model demand for various types of items at the supply class level. In addition, we can more easily simulate and track the movement of these aggregate supply items through the transportation network, using a unified relational database for exploration. For example, a planetary base might require ten units of crew provisions, rather than certain amounts of water, dehydrated food, drink mix, eating utensils, etc. With these classes of supply, the modeling problem can be reduced to a manageable size.[4]

Elements: Elements are defined as the indivisible physical objects that travel through the network and (in general) can hold or transport supplies and may or may not have propulsive capability. Most elements are what we generally think of as "vehicles" such as the Crew Exploration Vehicle (CEV), now known as *Orion*, or various propulsion stages. Here, we also include other major end items such as surface habitats and pressurized rovers. Elements are characterized as follows; they can:

- hold supply items (e.g., fuel, crew consumables, spares, exploration items)
- be propulsive or non-propulsive
- carry crew or not carry crew
- be launched from Earth
- be reused, refueled, disposed of (staged), pre-deployed
- be "docked" with other elements to form a (temporary) stack
- not be subdivided further without loosing their integrity

In general, an element has defined capacities for three types of items: crew, cargo, and propellant. These capacities determine what types of supplies can be assigned to that element for transport, and whether the element is propulsive. Thus, elements can transport supplies and crew between the various nodes of the transportation network.

Tying it all Together

With the preceding definitions of nodes, supplies, and elements, we have defined the basic building blocks of a modeling framework for space logistics. We can now create a network of nodes, and define elements capable of traversing that network between nodes, while carrying supplies. Two remaining concepts are needed to tie these ideas together: the time-expanded network (to account for time-varying trajectories) and processes that describe how precisely elements and supplies move through the network.

Time-Expanded Network: A time-expanded network is a concept that adds the dimension of "time" to a static network. We can think of a static network based on nodes like Kennedy Space Center (KSC), Low Earth Orbit (LEO), and Low Lunar Orbit (LLO). Now suppose one where take that static network (Fig. 2 left) and expand it over time, to

---

[4] For more detailed demand forecasting, a total of 47 sub-classes of supply were developed, but these are subsequently aggregated into the ten classes shown in
Figure **1**.

account for changes in the network over time, what one would then have is a simple time-expanded network (Fig.2 right).

In Fig.2 the *static network* is made up of the three physical nodes along the left-hand side of the figure below, labeled 'KSC', 'LEO', and 'LLO'. We then use a time step Δt=1 and expand these three nodes forward in time. At time step two, therefore, we copy each of the static nodes, so that the middle column in Fig. 2 is labeled 'KSC,2', 'LEO,2', and 'LLO,2'. We copy these nodes again for time step three, creating the right-most column. The next step is to define the allowable transitions – called *arcs* – between the nodes. It is always possible to remain or wait at a given node through the next time step. Therefore we can define all of the horizontal arcs (represented by dashed arrows) in Fig.2 as "waiting" arcs. Next, we look at the allowable transitions from KSC to LEO. The vertical arrow from 'KSC,1' to 'LEO,1' is illegal because it is impossible to make an instantaneous transition from KSC to LEO. In this example, it takes one time step to make that transition, so arrows are drawn from 'KSC,1' to 'LEO,2' and 'KSC,2' to 'LEO,3'. The reverse arcs from LEO to KSC (corresponding to reentry) are also added. Finally, the transition from LEO to LLO (in this notional example) takes longer: two times steps are required, so the arcs are drawn as shown in Fig.2. This completes the definition of the time-expanded network in this simple example; we have defined time-expanded nodes, waiting arcs, and feasible transport arcs (filtered by astrodynamic constraints, i.e. actually feasible trajectories). Now, we can define paths through the network; the figure below highlights in blue a notional path through KSC,1 to LEO,2 to LEO,3 (illustrating a transfer from KSC to LEO and a subsequent wait at LEO).

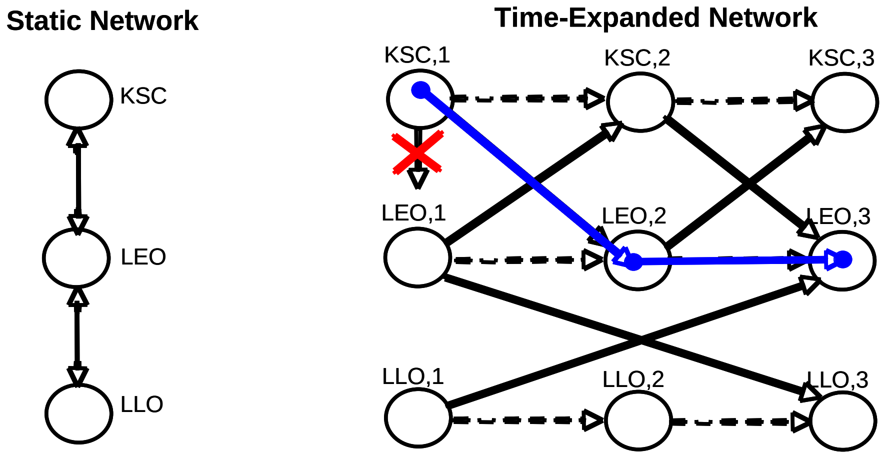


Figure 2 : left: Static Network, right: Simple Time Expanded Network

This framework is general enough to model any past, present or future crewed or uncrewed exploration mission or campaign in the solar system.

Notice that arcs are only defined in the forward direction, because it is impossible to traverse backward in time (non-causal paths are forbidden). While this network is relatively simple, the construction of such a network is nontrivial for large time horizons with small time steps, large static networks, or for physical systems like the Earth-Moon-Mars system. A realistic time expanded network with a three-year scenario (about 1000 days), a time step of one Earth day and ten nodes will have 10,000 nodes once expanded in time. The advantages of this type of network construction are that it makes time explicit and enables simulation and optimization of time-varying transportation problems, such as the launch windows to Mars. Time-expanded networks are not new per se[5], but their application to space logistics is new and is a key underpinning of SpaceNet

Processes: With the time-expanded network defined, the only remaining step is to describe how elements and supplies are allowed to move among the nodes of the time-expanded network. There are *five essential processes* that describe this movement:

Waiting: remaining at the same physical node for one time period or more
Transporting: moving an element to a new node along an allowable arc
Transferring: transferring crew and/or supply items to a different co-located element
Exploring: exploration and science operations at a node
Proximity Operations: local rendezvous/docking and undocking/separation of elements

At this point, with the building blocks, the network, and the processes defined, one can model the flow of supplies, elements, and crew through the space logistics network.

Wrapping it Up: Optimization and Simulation

The final step in building an effective modeling framework is to add wrappers that allow logistics scenarios described by the model to be created, visualized, and evaluated. We first describe the use of an optimization capability within the modeling framework described above. Second, we describe the use of a simulation capability which takes a specific scenario, created by hand, and simulates it to ensure feasibility such as that demands are met; transport processes have sufficient fuel, etc.

Optimization: In some cases, we envision that pre-defined mission architectures will be evaluated against one another, resulting in a logistics trade study. In other cases, however, the best approach would be to calculate the demand for various supplies at a lunar base for example, and ask the software to find the optimal logistics architecture to supply that mission (or series of missions). This is the goal of the optimization layer. The modeling framework was specifically built to allow for optimization.

Simulation: A specific logistics architecture (referred to as a *scenario* in SpaceNet) can be created using the modeling elements discussed above. However, in order to determine the effectiveness of the architecture, it must be simulated so that it can be evaluated in relation to others. The simulation ties together all other components of the modeling framework, taking the mission scenario as an input and producing the output information to fully describe and evaluate the mission scenario. There are two main functions that the simulation capability in SpaceNet provides:

Feasibility: can a scenario be executed feasibly (sufficient fuel, no critical undersupply)?

---

[5] John J. Jarvis, H. Donald Ratliff, "Some Equivalent Objectives for Dynamic Network Flow Problems", *Management Science*, Vol. 28, No. 1 (Jan., 1982), pp. 106-109

Effectiveness: if feasible, how effective is a particular scenario?

What is SpaceNet?

SpaceNet is a computation environment, coded in Matlab, for modeling exploration from a logistics perspective. It includes discrete event simulation at the individual mission level (e.g., sortie, pre-deploy, or resupply) or at the campaign level (i.e., set of missions). It also allows for the evaluation of manually generated exploration scenarios with respect to feasibility[6] and measures of effectiveness (MOEs), as well as the visualization of the flow of elements and supplies through the interplanetary supply chain. Finally, it includes an optimization capability and acts as a software tool to support trade studies and architecture analyses.

SpaceNet is built on the modeling framework described earlier in this chapter. SpaceNet provides a graphical user interface that allows analysts to describe complex space logistics architectures using the basic concepts of nodes, elements, and supplies.[7] Missions are modeled on a network of nodes and arcs, with elements carrying supplies through the network. Built-in *demand models* and *logistics MOEs*, a *unified database* of nodes, elements, supply item attributes and astrodynamics constraints, and an *optimization capability* assist the analyst in describing various types of supply chains and assessing various supply chain strategies through simulation. As such, SpaceNet supports architecture-level trade studies for space logistics, and can act as an integrated planning and simulation tool.

---

[6] For example, undersupply situations are explicitly flagged as error conditions.

[7] SpaceNet v1.3 focuses on the Earth-Moon system, whereas future versions will encompass Mars and other destinations. The basic interfaces for expansion beyond the Earth-Moon system are present.

(this page intentionally left blank)

SpaceNet Quick Start

The following instructions will familiarize new users with SpaceNet by walking them through the creation of a very simple scenario. The creation of this simple scenario should take a first time user less than 30 minutes.

1. Start Matlab
2. Change your Matlab current directory to the …*\spacenet_1.3\gui* folder
3. Type *SpaceNet* at the Matlab command line [The SpaceNet title GUI shown in **Figure 3** will open after about 5-10 seconds]
4. Click on *Create New Scenario*

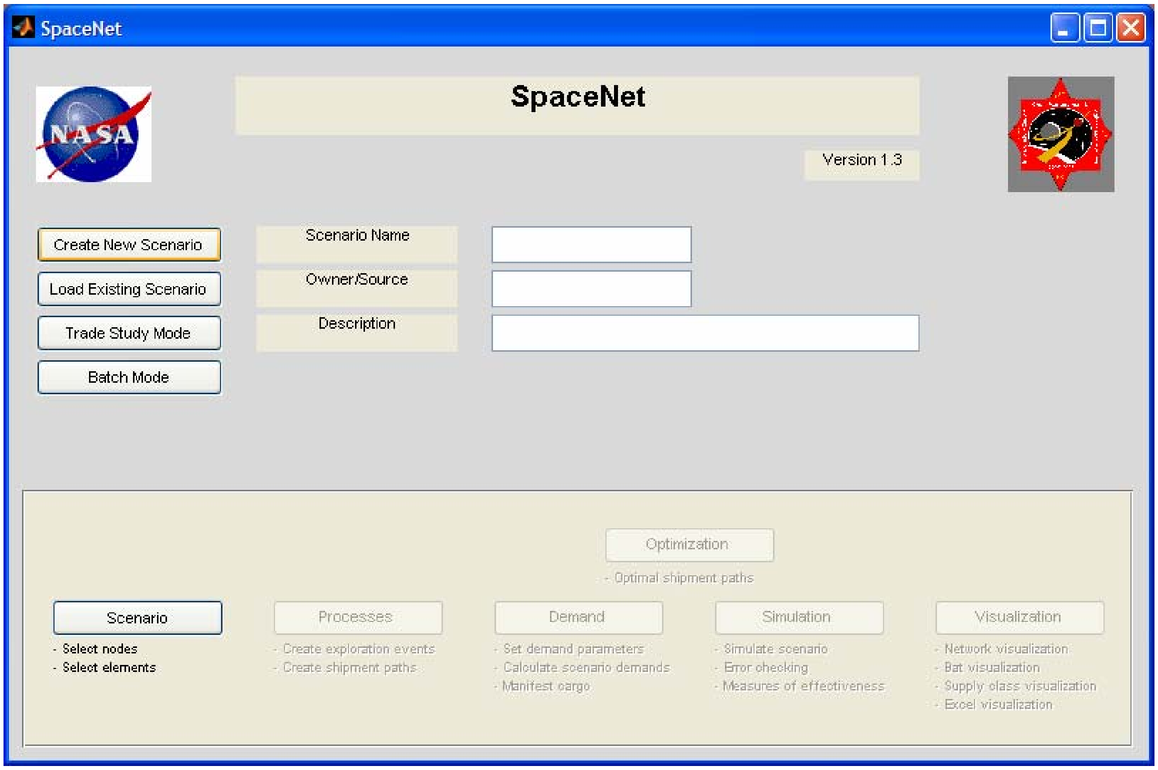


Figure 3: Title GUI

5. In the Scenario Name box type *Demo* [or any other short name you wish]
6. Leave the Owner/Source and Description boxes empty [optional]
7. Click on the *Scenario* button [The title GUI will now close and the main GUI shown in **Figure 4** will open after a few seconds]

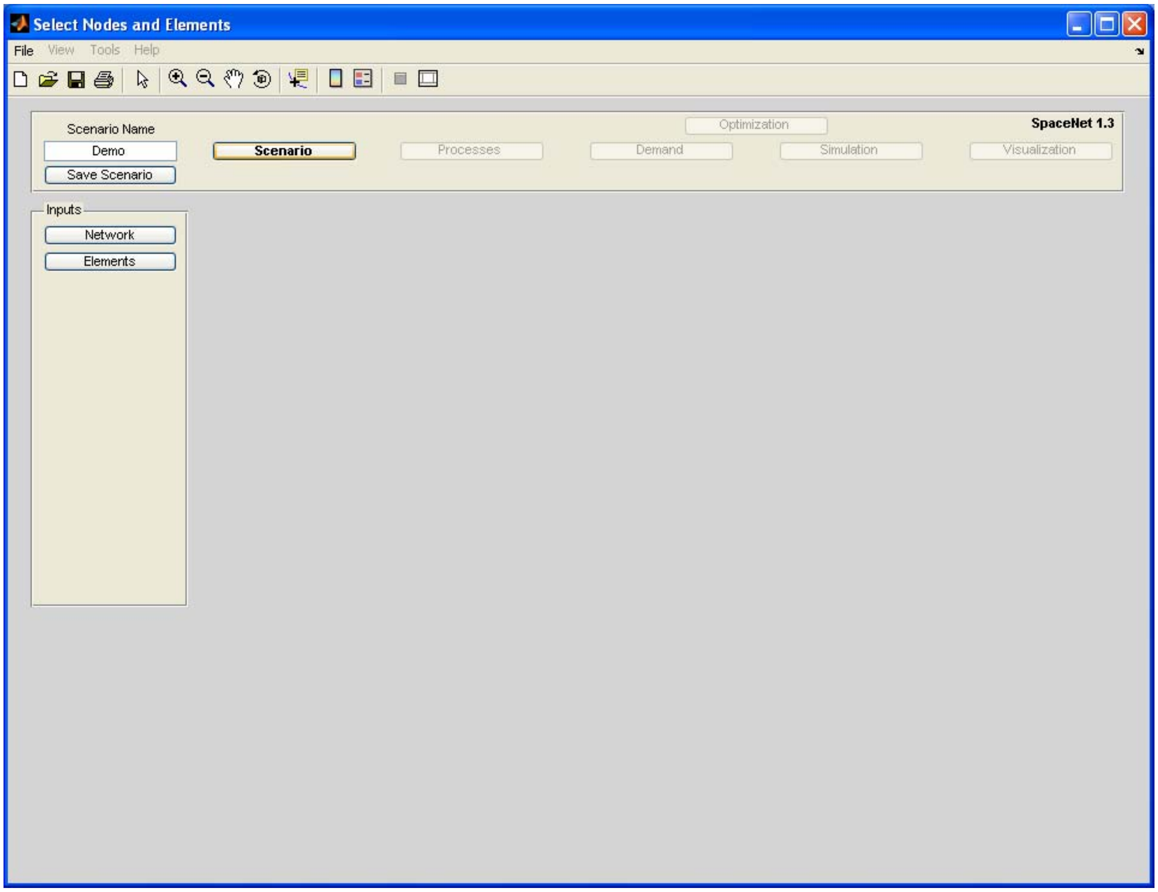


Figure 4: Main GUI – Scenario

8. Click on the *Network* button
9. Enter *01/01/10* in the start date field
10. Enter *01/04/10* in the end date field
11. Enter *0.1* in the Time Discretization box and hit *Enter* (Return) on the keyboard [A new section of the GUI will appear to allow for node selection]
12. Double click on the following surface nodes in the Node Library: *NASA KSC, Edwards AFB*
13. Change the drop down menu from Surface Nodes to *Orbit Nodes*
14. Double click on *LEO Parking Orbit*
15. Click on the *Save Scenario* button in the upper left corner of the GUI [A brief message will appear: *Demo has been saved*]
16. Click on the *Elements* button [The Element page will appear]
17. Double click on the following elements in the Element Type Library to include them in your scenario: CEV LAS, ISS CEV CM (3 Crew + Cargo) [you need to scroll down to find it], ISS CEV SM, CLV Upper Stage, CLV Booster Stage
18. Click on the *Save Scenario* button [A brief message will appear: *Demo has been saved*]
19. Click on the *Processes* button on top [The processes GUI shown in **Figure 5** will now appear]

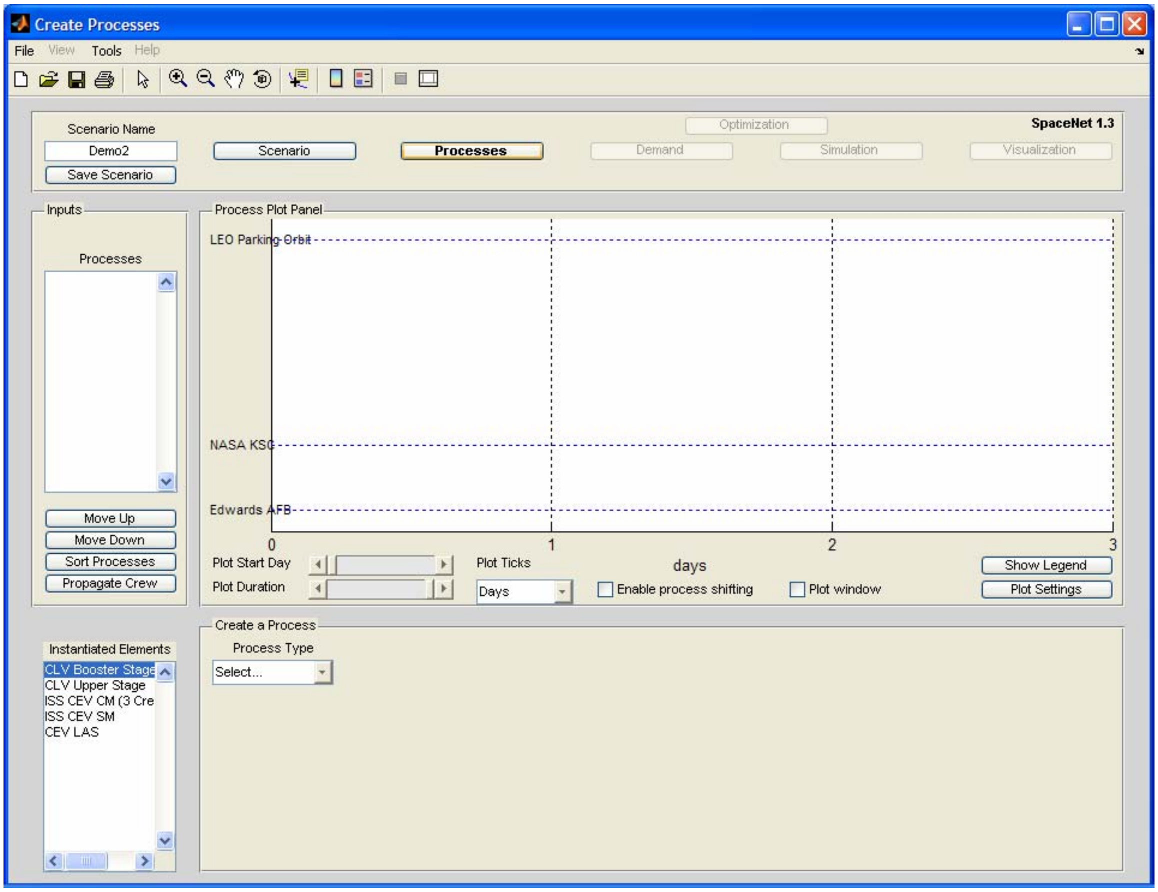
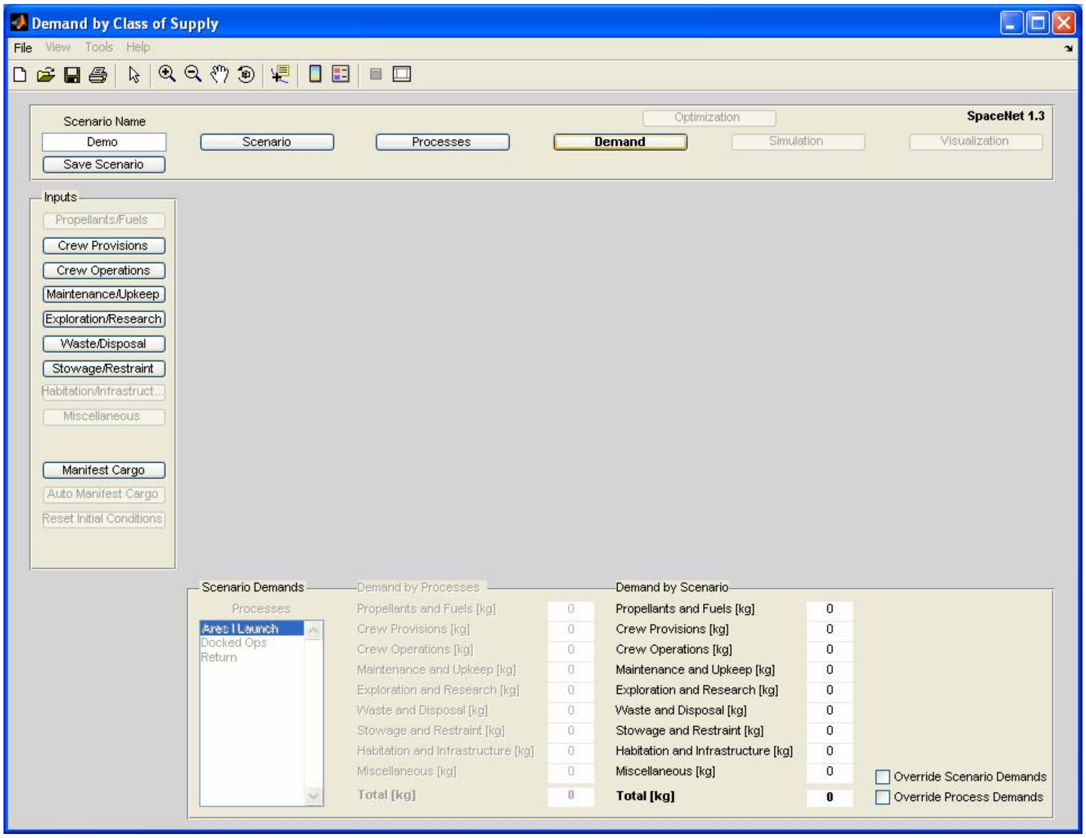


Figure 5: Main GUI – Processes

20. Select *Transport* from the Process Type drop down menu
21. Type *Ares I Launch* in the Process Name field, hit enter [additional information will appear in the GUI in the lower right]
22. Leave the *Node From* drop down menu choice as *NASA KSC*
23. Change the Node To drop down choice to *LEO Parking Orbit*
24. Double click on each element individually in the Instantiated Elements box to add them to the Elements box in the following order: CLV Booster Stage, CEV LAS, CLV Upper Stage, ISS CEV SM, ISS CEV CM (3 Crew + Cargo). (When you are finished, the elements should appear in the Elements box in the following order from bottom to top: CLV Booster Stage, CEV LAS, CLV Upper Stage, ISS CEV SM, ISS CEV CM (3 Crew + Cargo))
25. Check the *Burn 1* checkbox for the following elements: CLV Booster Stage, CLV Upper Stage
26. Check the *Stage* checkboxes for the following elements: CLV Booster Stage, CEV LAS and CLV Upper Stage
27. Check the *Crewed* checkbox for the ISS CEV CM (3 Crew + Cargo)
28. Enter *3* in the Crew Input pop-up box, press OK
29. Under the Save Process drop down, select *Add Process* [A diagonal red line should appear in the white window, indicating addition of the transport process]
30. Select *Exploration* from the Process Type drop down
31. Enter *Docked Ops* in the Process Name field, hit enter

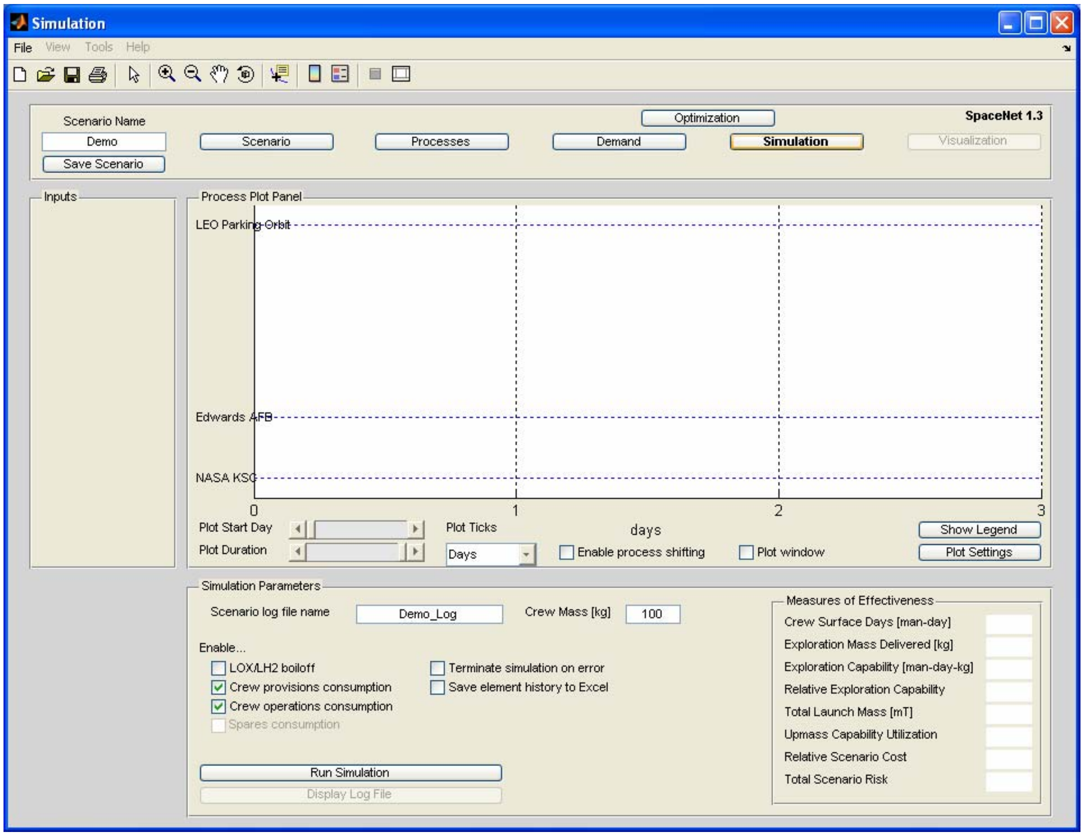
32. Verify that *LEO Parking Orbit* is selected in the Node drop down
33. Enter *2* in the Nominal Duration box
34. Enter *1* in the Contingency Duration box
35. Leave *0* in the Num. 2-Person EVAs box
36. Leave *0* in the Non-Availability % box
37. Double click on the following elements in the Instantiated Elements box: ISS CEV SM, ISS CEV CM (3 Crew + Cargo)
38. Check the *Crewed* checkbox for the ISS CEV CM (3 Crew + Cargo)
39. Enter *3* in the Crew Input pop-up box, press OK
40. Under the Save Process drop down, select *Add Process* [A horizontal blue line should appear in the white window, indicating addition of the exploration process]
41. Select *Transport* from the Process Type drop down
42. Enter *Return* in the Process Name field, hit enter
43. Verify that *LEO Parking Orbit* is selected in the Node From drop down
44. Select *Edwards AFB* from the Node To drop down
45. Double click on the following elements in the Instantiated Elements box: ISS CEV SM, ISS CEV CM (3 Crew + Cargo)
46. Check the *Crewed* checkbox for the ISS CEV CM (3 Crew + Cargo)
47. Enter *3* in the Crew Input pop-up box, press OK
48. Check the *Burn 1* and *Stage* boxes for the ISS CEV SM
49. Under the Save Process drop down, select *Add Process* [A diagonal red line should appear indicating addition of the transport process]
50. Click on the *Save Scenario* button
51. Click on the *Demand* button (on top) [The demand GUI shown in **Figure 6** will now appear]
52. Click on the *Crew Provisions* button
53. Click on the *Calculate Crew Provisions* button
54. Click on the *Crew Operations* button
55. Click on the *Calculate Crew Operations* button
56. Click on the *Exploration/Research* button
57. Double click on *Docked Ops* in the Processes box
58. Double click on the first five items in the Supply Item Type Library to add these items to your Instantiated Supply Items list
59. Click on the *Waste/Disposal* button
60. Click on the *Calculate Waste Equipment* button
61. Click on the *Stowage/Restraint* button
62. Click on *Pack Items* [A popup window appears and shows how the cargo has been packed, CTB=cargo transfer bag, CWC=Contingency Water Container]
63. Click *OK* on the Packing Summary pop-up
64. Click the *Manifest Cargo* button
65. Click the *Auto-Manifest Cargo* button [left side]
66. Click *Continue* on the Auto-Manifest Cargo pop-up



Demand by Processes		Demand by Scenario	
Propellants and Fuels [kg]	0	Propellants and Fuels [kg]	0
Crew Provisions [kg]	0	Crew Provisions [kg]	0
Crew Operations [kg]	0	Crew Operations [kg]	0
Maintenance and Upkeep [kg]	0	Maintenance and Upkeep [kg]	0
Exploration and Research [kg]	0	Exploration and Research [kg]	0
Waste and Disposal [kg]	0	Waste and Disposal [kg]	0
Stowage and Restraint [kg]	0	Stowage and Restraint [kg]	0
Habitation and Infrastructure [kg]	0	Habitation and Infrastructure [kg]	0
Miscellaneous [kg]	0	Miscellaneous [kg]	0
Total [kg]	0	**Total [kg]**	0

Figure 6: Main GUI – Demand

67. Click *OK* on the Optimization Settings pop-up
68. Click *NO* on the Reward Matrix pop-up
69. Click *OK* on the Auto-Manifest Cargo pop-up [All items manifested into …]
70. Click on the *Save Scenario* button
71. Click on the *Simulation* button [The simulation GUI shown below in **Figure 7** will now appear]
72. Click *Run Simulation* [Various waitbar popups will appear to show progress]
73. Click *OK* on the pop-up [Finished simulating …]
74. Click on the *Visualization* button in the top right of the GUI [The visualization GUI page shown below in **Figure 8** will now appear]



Scenario Name: Demo

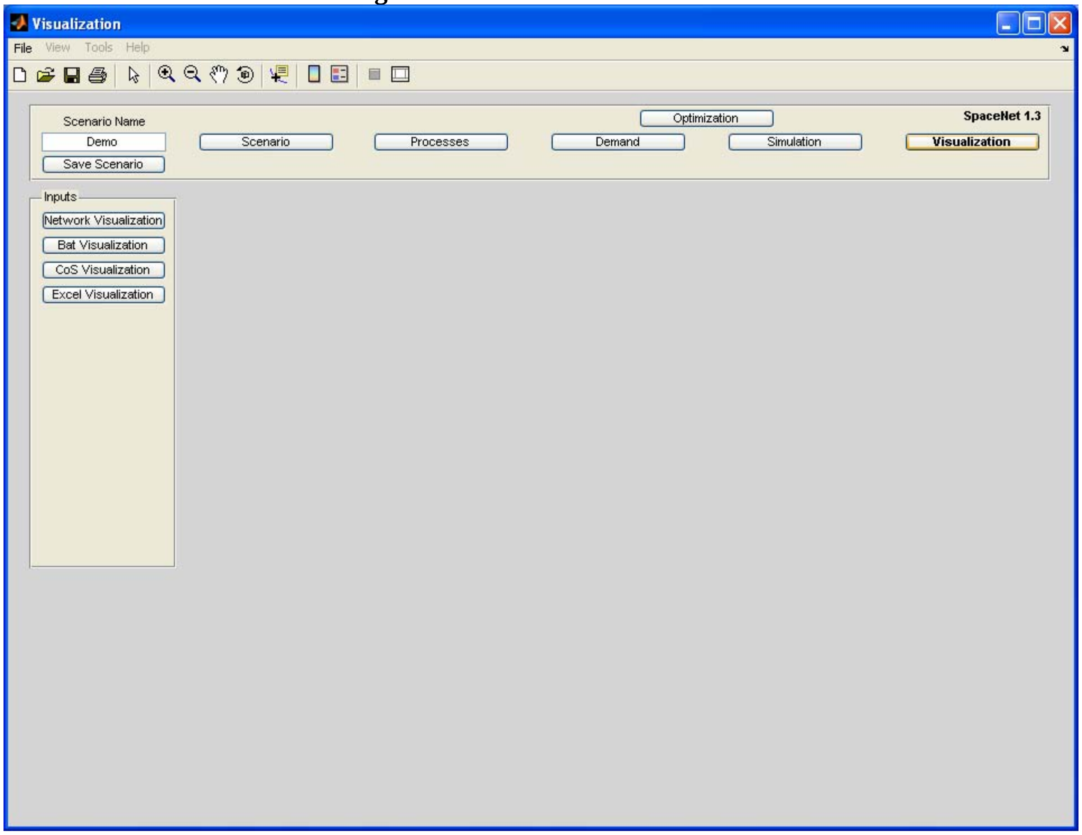
Plot Ticks: Days

☐ Enable process shifting
☐ Plot window

Scenario log file name: Demo_Log
Crew Mass [kg]: 100

Enable...
☐ LOX/LH2 boiloff
☑ Crew provisions consumption
☑ Crew operations consumption
☐ Spares consumption
☐ Terminate simulation on error
☐ Save element history to Excel

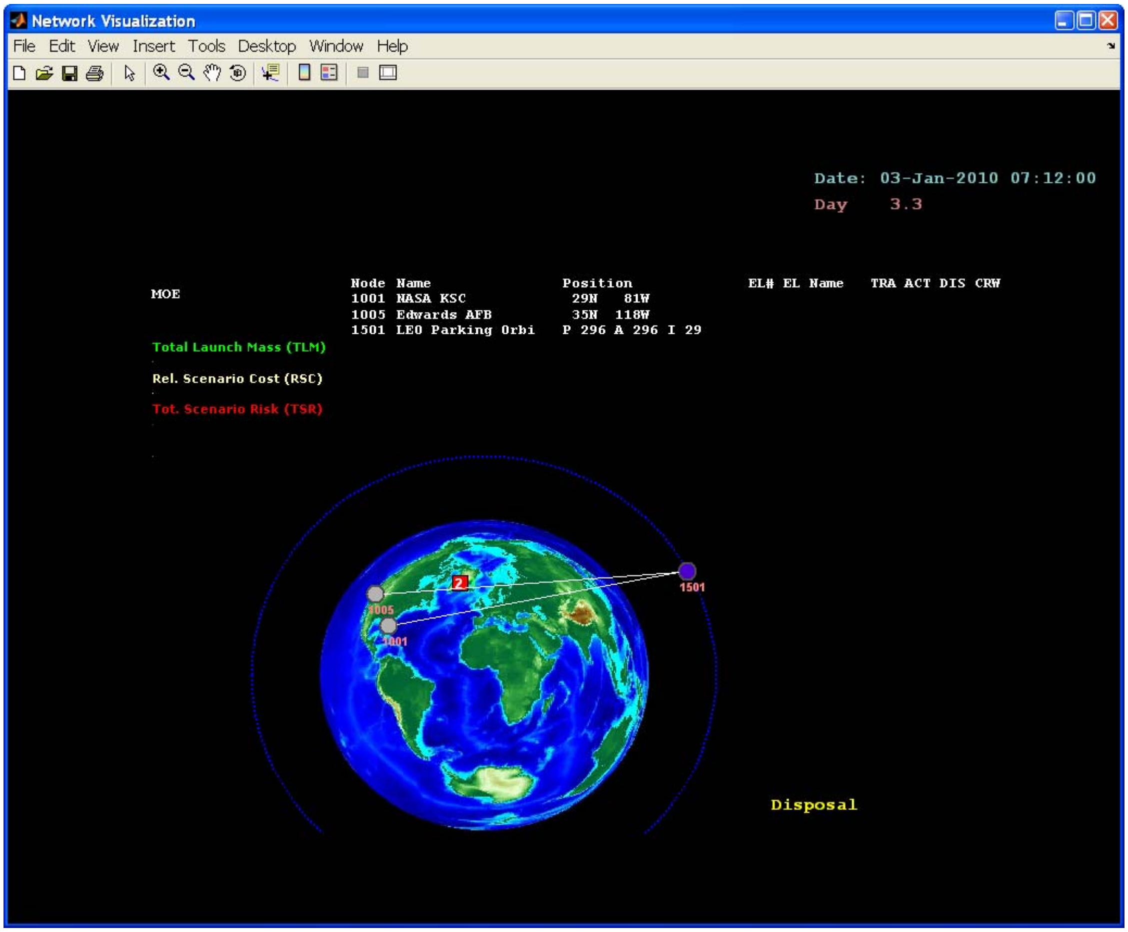
Figure 7: Main GUI – Simulation



Scenario Name: Demo

Figure 8: Main GUI – Visualization

75. Click *Network Visualization*
76. Click *Run Animation* and observe the network visualization of the scenario you just created [black background showing Earth, Earth orbit and the three nodes], see **Figure 9**.



```
Date: 03-Jan-2010 07:12:00
Day    3.3

                  Node Name              Position           EL# EL Name   TRA ACT DIS CRW
MOE               1001 NASA KSC           29N   81W
                  1005 Edwards AFB        35N  118W
                  1501 LEO Parking Orbi  P 296 A 296 I 29
Total Launch Mass (TLM)

Rel. Scenario Cost (RSC)

Tot. Scenario Risk (TSR)

                                                        Disposal
```

Figure 9 Network Visualization

77. Close the *Network Visualization* window
78. Click on the *CoS Visualization*
79. Select the ISS CEV CM (3 Crew + Cargo) from the instantiated elements list box
80. Click the *Plot* button next to Crew Provisions (2nd Plot button from the top) [This plot illustrates the consumption of food and water by the crew over the 3 day scenario]
81. Click on the *Save Scenario* button

Congratulations, you have just finished building your first scenario in SpaceNet! To exit SpaceNet select *Exit* from the File menu.

(this page intentionally left blank)

Chapter 2 Loading and Running SpaceNet

Before running SpaceNet, the user must first expand the *spacenet13.zip* file and establish a set of directories and files. The following directories must exist within a SpaceNet parent directory on the hard drive of the local computer as shown below:

- …\spacenet_1.3
    - \database
    - \demand
    - \docs
    - \gui
    - \icons
    - \optimization
    - \scenario
    - \simulation

This directory structure is already present on the SpaceNet 1.3 installation CD or *spacenet13.zip* file when downloading SpaceNet from the internet.

To run SpaceNet, open MATLAB and set the current directory to …\spacenet_1.3\gui. In the command window, type "SpaceNet". This will launch the title GUI shown below. There may be a delay of about 10 seconds when opening SpaceNet because data from the integrated database (stored in \database) is being read from Excel.

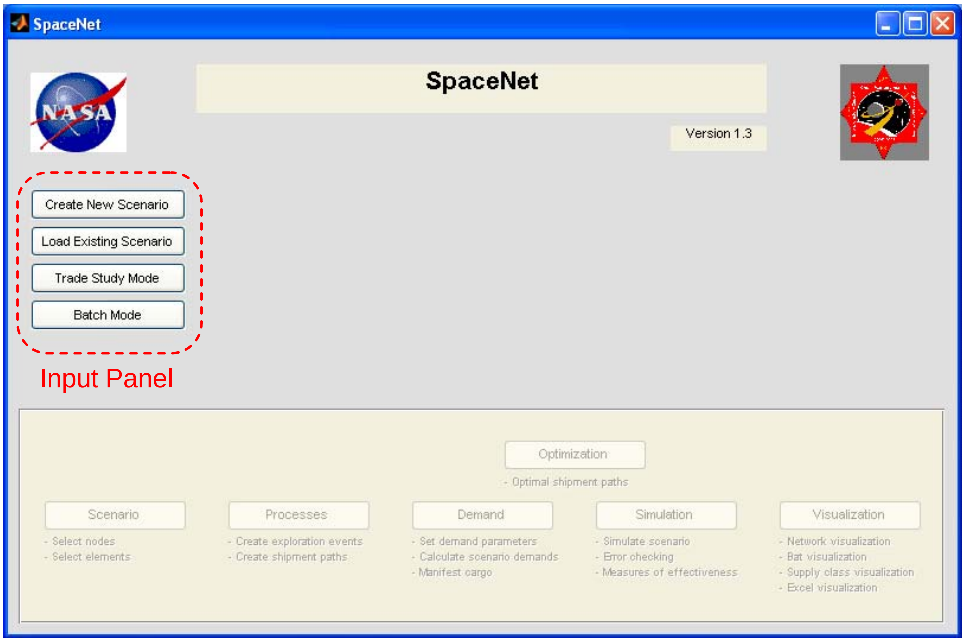


Figure 10: Title GUI

From the input panel in the title GUI (**Figure 10**) the user can create a new scenario or load an existing scenario. The user can perform analyses using trade study mode or batch mode for previously created scenarios.

The six buttons at the bottom show the sequential steps required to create a scenario, from left to right. These are initially grayed out but will be activated later. The buttons will be explained in detail later but an overview of them is given below.

- Scenario
    - *Select Nodes*: Select nodes from the integrated database and create the static network
    - *Select Elements*: Select elements (e.g. CEV) from the integrated database and instantiate them
- Processes
    - *Create Exploration Processes:* Create exploration processes at selected nodes with selected elements
    - *Create Transportation Paths*: Define how elements and supplies move in the network
- Demand
    - *Set Demand Parameters*: Save demand parameters for each class of supply to be used in the demand models and simulation
    - *Calculate Scenario Demands*: Calculate the demand for the scenario by class of supply
    - *Manifest Cargo*: Set the initial conditions at launch for each element by class of supply
- Optimization
    - *Optimal Shipment(Transportation) Paths*: Set optimization parameters and calculate the optimal shipment paths using CPLEX optimization
    - This capability will attempt to minimize the number of elements used
- Simulation
    - *Simulate Scenario*: Create time histories for each element in the scenario
    - *Error Checking*: Errors are reported and logged in a text file. This step checks for feasibility violations of a scenario
    - *Measures of Effectiveness*: Calculate supply chain measures of effectiveness (MOEs) from the element time histories
- Visualization
    - *Network Visualization*: Animate elements from a network perspective
    - *Bat Visualization*: Animate elements over time in a “bat” diagram fashion
    - *Supply Class Visualization*: Create plots of the time history of any class of supply in any element
    - *Excel Visualization:* Create plots of the element time histories in Excel using Visual Basic. This visualization capability is very powerful and can help view the interplanetary supply chain from a nodal, element or supply class perspective over time

Creating New Scenarios

Push the “Create New Scenario” button to create a new scenario.

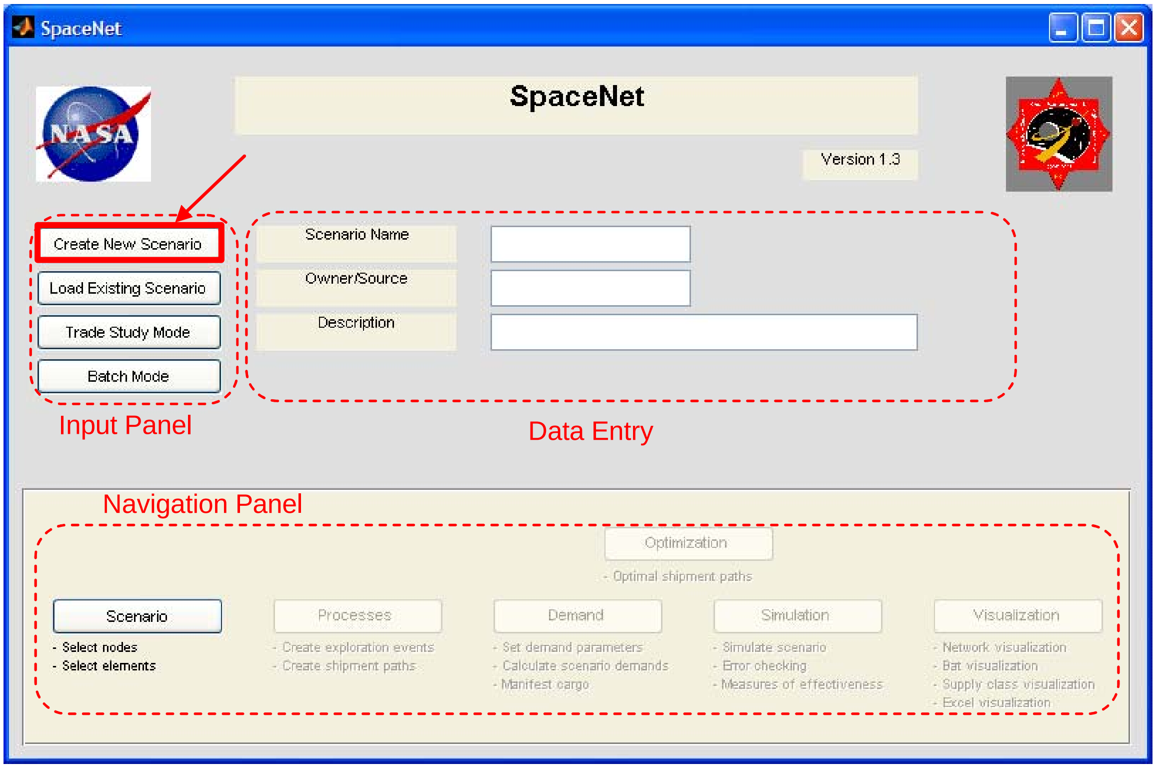


Figure 11: Title GUI - Create New Scenario

The following fields are now visible for Data Entry:

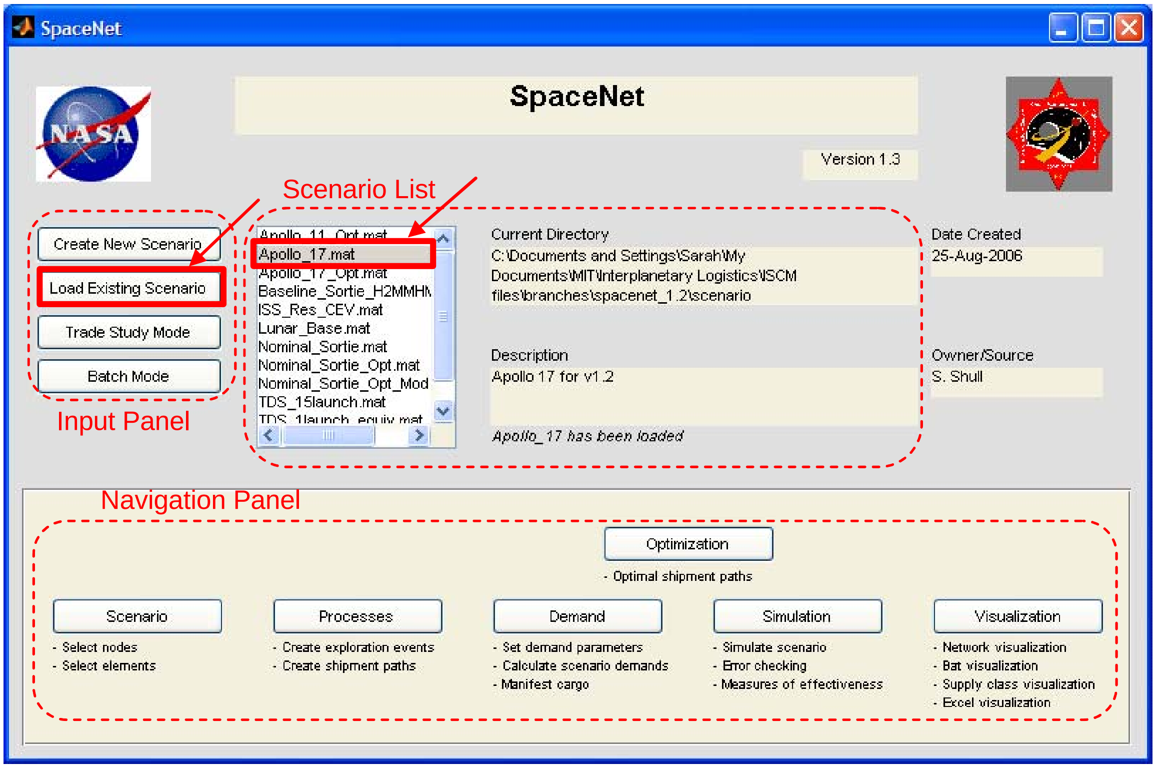
- *Scenario Name*: Type in a name for the new scenario
- *Owner/Source*: Type in the owner/source for the new scenario
- *Description*: Type in a description for the new scenario

Only the “Scenario” button is enabled in the Navigation Panel at the bottom of the screen because the rest of the buttons will be inactive until scenario information has been defined. Push the “Scenario” button to close the title GUI and launch the main GUI (refer to Chapter 3 Main GUI for a detailed explanation of each screen in the main GUI).

It might take a few seconds for the Title GUI to close and the Main GUI to appear.

Loading Saved Scenarios

Push the “Load Existing Scenario” button to load an existing scenario.



Current Directory
C:\Documents and Settings\Sarah\My Documents\MIT\Interplanetary Logistics\SCM files\branches\spacenet_1.2\scenario

Date Created
25-Aug-2006

Description
Apollo 17 for v1.2

Owner/Source
S. Shull

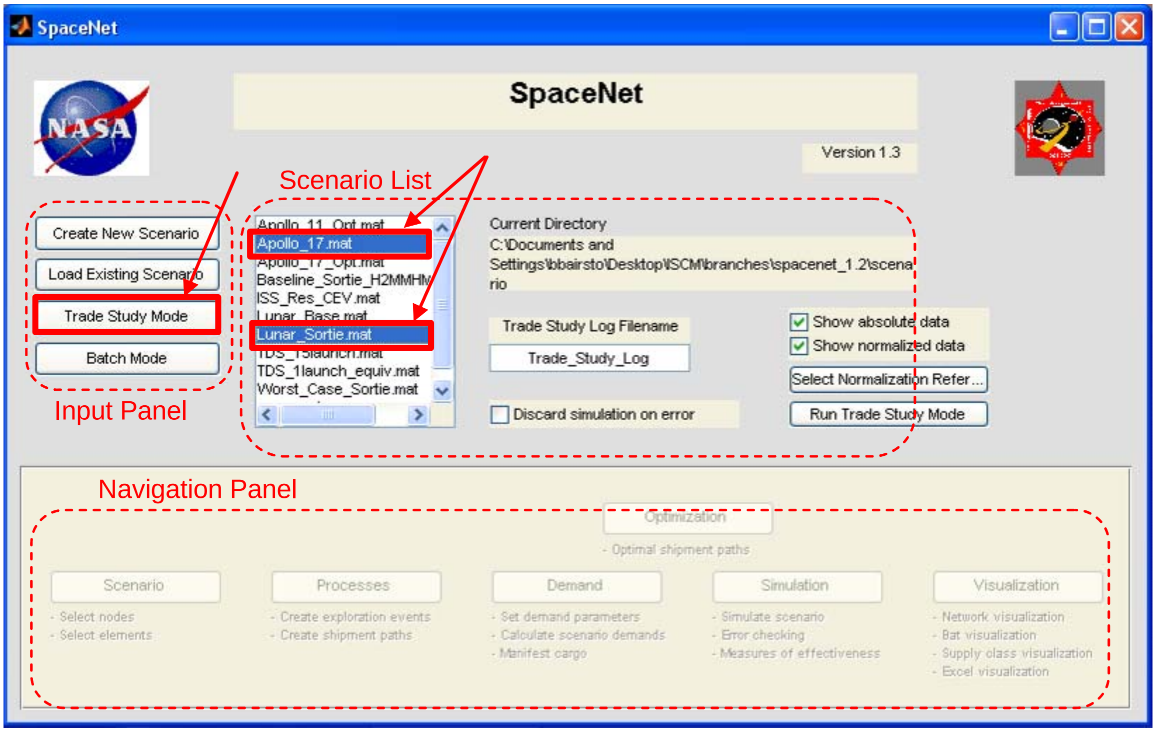
Apollo_17 has been loaded

Figure 12: Title GUI - Load Existing Scenario

The listbox displays the current scenario files in the \scenario directory. All scenarios that are delivered together with the SpaceNet 1.3 have been tested and should run without errors. Double-click on an item in the listbox to load a scenario. The original baseline scenario that was used to calibrate SpaceNet and normalize its MOEs is Apollo 17, the last crewed lunar mission in 1972. In the figure above, the italicized text “Apollo_17 has been loaded” alerts the user that the Apollo 17 scenario has been loaded. The six buttons at the bottom will be enabled depending on which step the user last saved in the scenario creation or editing process. Push any of the enabled buttons to close the title GUI and launch the main GUI (this will take between a few seconds seconds).

Trade Study Mode

Push the “Trade Study Mode” button to run trade study mode on the existing scenarios in the \scenario folder. Trade Study mode allows the user to run multiple scenarios and compare their MOEs.



Current Directory: C:\Documents and Settings\bbairsto\Desktop\ISCM\branches\spacenet_1.2\scenario

Trade Study Log Filename: Trade_Study_Log

☑ Show absolute data

☑ Show normalized data

☐ Discard simulation on error

Figure 13: Title GUI - Trade Study Mode

To run trade study mode:

- Select scenarios of interest by using control-click
- Enter the file name in which output data will be saved (*Trade_Study_Log* is the default)
- Check to discard scenarios with simulation errors
- Check to view plots displaying the MOEs
  - Show absolute data displays each MOE as an individual figure
  - Show normalized data displays all MOEs in one figure
- Select which scenario the MOE data should be normalized to with the *Select Normalization Reference* button
- Push the “Run Trade Study Mode” button to execute a trade study

In the figure above, the *Apollo_17* and *Lunar_Sortie* scenarios have been selected to be run in trade study mode.

Figure 14 shows the window in which the normalization reference is selected. The selection (in this case Apollo_17) will be used as the baseline for the comparison plots. Selecting "Maximum" will normalize the largest value as 1 for each MOE.

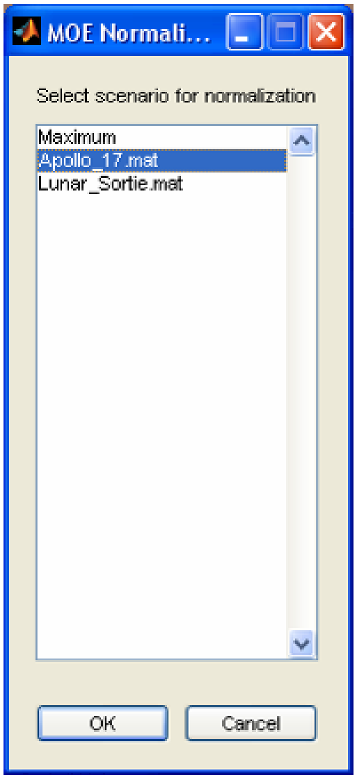
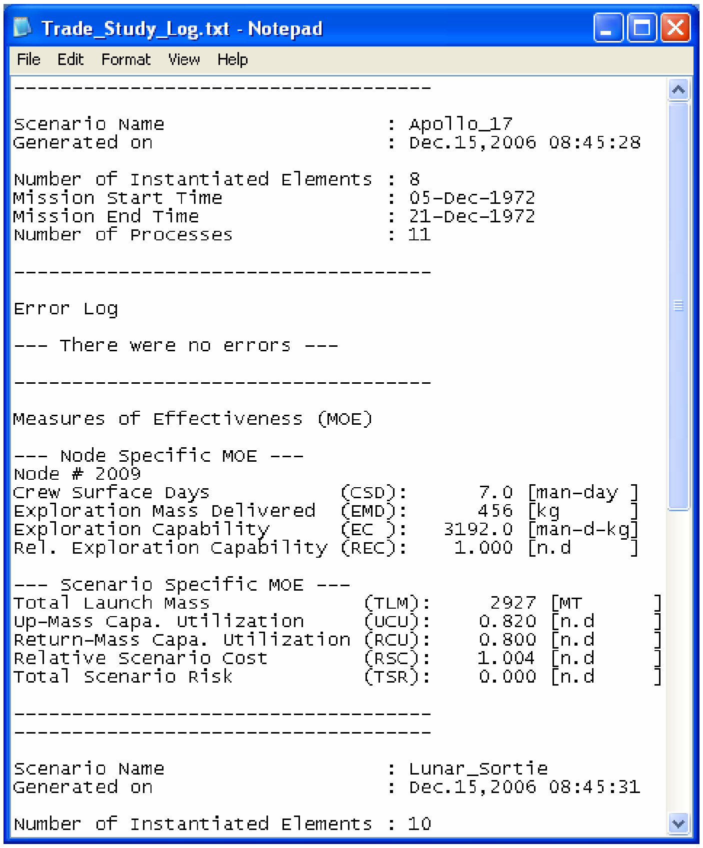


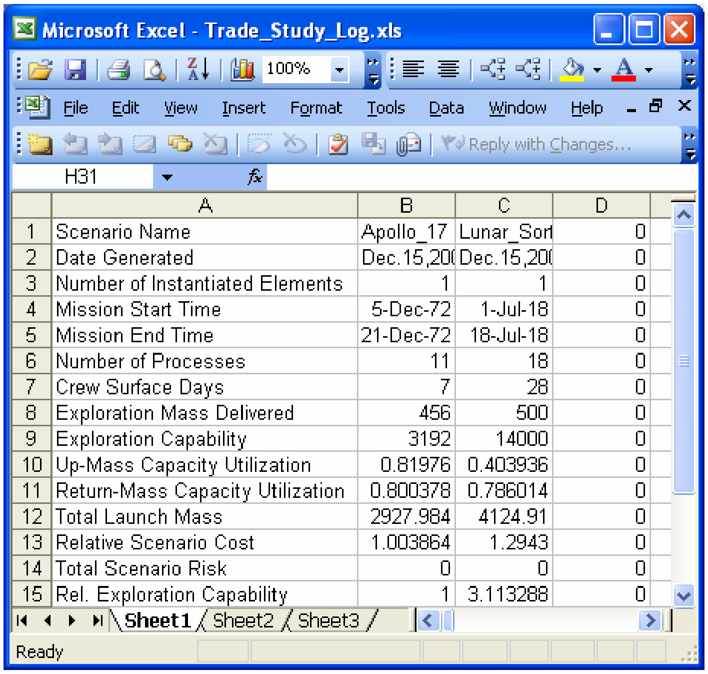
Figure 14: Trade Study Mode – Select Normalization Reference

In addition to any plots the user chooses to create, trade study mode also outputs a .txt file (see Figure 15) that logs all errors and MOEs for each scenario, and an Excel file (.xls, see Figure 16) that logs the scenario information and MOE data. This file will be placed in the \scenario folder and is entitled "*Trade_Study_Log*" by default.



```
------------------------------------

Scenario Name                     : Apollo_17
Generated on                      : Dec.15,2006 08:45:28

Number of Instantiated Elements : 8
Mission Start Time                : 05-Dec-1972
Mission End Time                  : 21-Dec-1972
Number of Processes               : 11

------------------------------------

Error Log

--- There were no errors ---

------------------------------------

Measures of Effectiveness (MOE)

--- Node Specific MOE ---
Node # 2009
Crew Surface Days            (CSD):      7.0 [man-day ]
Exploration Mass Delivered   (EMD):      456 [kg      ]
Exploration Capability       (EC ):   3192.0 [man-d-kg]
Rel. Exploration Capability (REC):    1.000 [n.d     ]

--- Scenario Specific MOE ---
Total Launch Mass              (TLM):     2927 [MT      ]
Up-Mass Capa. Utilization      (UCU):    0.820 [n.d     ]
Return-Mass Capa. Utilization (RCU):    0.800 [n.d     ]
Relative Scenario Cost         (RSC):    1.004 [n.d     ]
Total Scenario Risk            (TSR):    0.000 [n.d     ]

------------------------------------
------------------------------------

Scenario Name                     : Lunar_Sortie
Generated on                      : Dec.15,2006 08:45:31

Number of Instantiated Elements : 10
```

Figure 15: Trade Study Mode – Text Log File

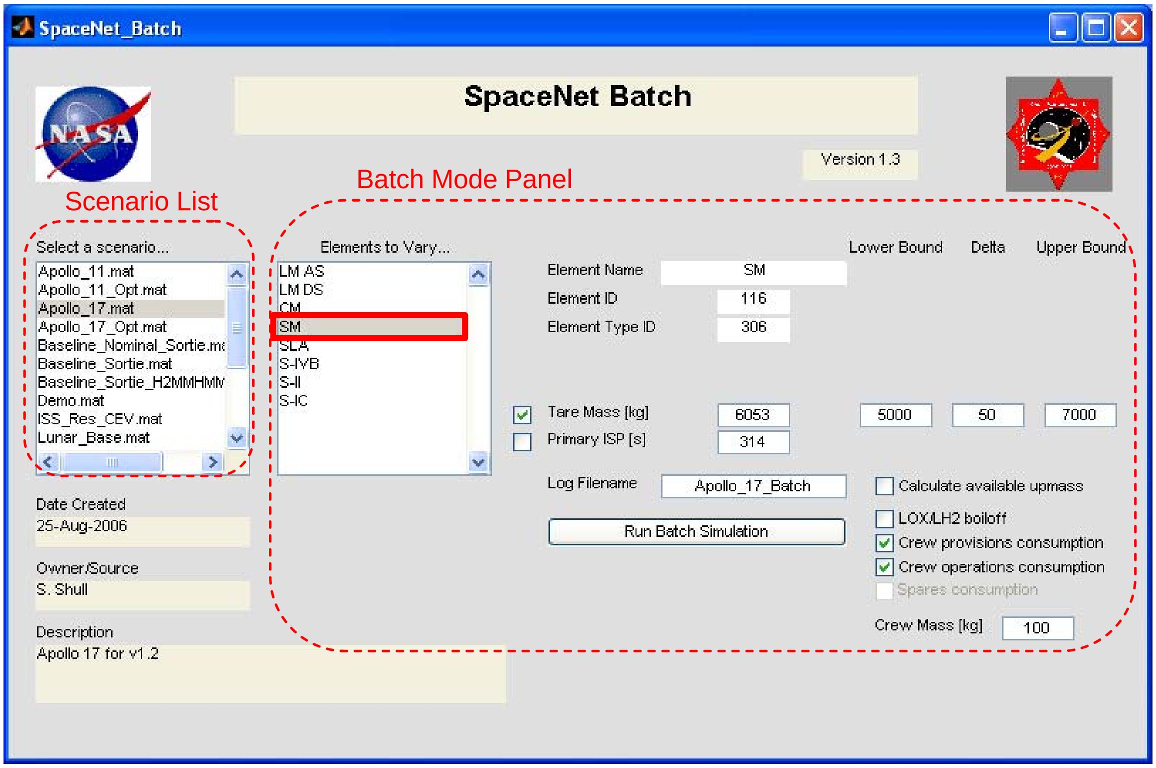


	A	B	C	D
1	Scenario Name	Apollo_17	Lunar_Sort	0
2	Date Generated	Dec.15,20(	Dec.15,20(	0
3	Number of Instantiated Elements	1	1	0
4	Mission Start Time	5-Dec-72	1-Jul-18	0
5	Mission End Time	21-Dec-72	18-Jul-18	0
6	Number of Processes	11	18	0
7	Crew Surface Days	7	28	0
8	Exploration Mass Delivered	456	500	0
9	Exploration Capability	3192	14000	0
10	Up-Mass Capacity Utilization	0.81976	0.403936	0
11	Return-Mass Capacity Utilization	0.800378	0.786014	0
12	Total Launch Mass	2927.984	4124.91	0
13	Relative Scenario Cost	1.003864	1.2943	0
14	Total Scenario Risk	0	0	0
15	Rel. Exploration Capability	1	3.113288	0

Figure 16: Trade Study Mode – Excel Log File

Batch Mode

Push the “Batch Mode” button to open the batch mode GUI (Figure 17). Batch mode allows the user to vary the mass and I_{SP} of the elements in a single scenario and then run all permutations of these parameters to analyze their impacts on the MOEs.



Select a scenario...
Apollo_11.mat
Apollo_11_Opt.mat
Apollo_17.mat
Apollo_17_Opt.mat
Baseline_Nominal_Sortie.ma
Baseline_Sortie.mat
Baseline_Sortie_H2MMHMM
Demo.mat
ISS_Res_CEV.mat
Lunar_Base.mat

Elements to Vary...
LM AS
LM DS
CM
SM
SLA
S-IVB
S-II
S-IC

Element Name SM
Element ID 116
Element Type ID 306

		Lower Bound	Delta	Upper Bound
☑ Tare Mass [kg]	6053	5000	50	7000
☐ Primary ISP [s]	314			

Log Filename Apollo_17_Batch

☐ Calculate available upmass
☐ LOX/LH2 boiloff
☑ Crew provisions consumption
☑ Crew operations consumption
☐ Spares consumption

Crew Mass [kg] 100

Date Created
25-Aug-2006

Owner/Source
S. Shull

Description
Apollo 17 for v1.2

Figure 17: Batch Mode

To run batch mode:

- Double-click on a scenario of interest to load that scenario
- Select an instantiated element
- Check whether you want to vary tare mass, I_{SP}, or both
- Enter the lower bound, delta (increment), and upper bound
- Push “Run Batch Simulation”

In the example above, Apollo_17 has been loaded and the user has chosen to vary the SM dry mass from 5000 to 7000 in increments of 50 kg in the Batch Mode panel. A message box pops-up to say that 41 scenario variants will be simulated, which will take approximately one minute to run. If one clicks “Ok” the 41 runs are executed and a red waitbar displays the progress.

To plot the batch mode results:

- Set the x y z pop-up menus to plot either 2-D plots of simulation number vs. a MOE or 3-D plots of simulation number vs. MOE1 vs. MOE2.

- Push “Plot”. If any errors occurred in a simulation, they will show up as red dots instead of blue dots.
- The plot in Figure 18 shows relative exploration capability (REC, see Appendix D) versus simulation number as an example. As the SM gets heavier relative exploration capability is reduced slightly because for the same exploration manifest, launch mass is increased.

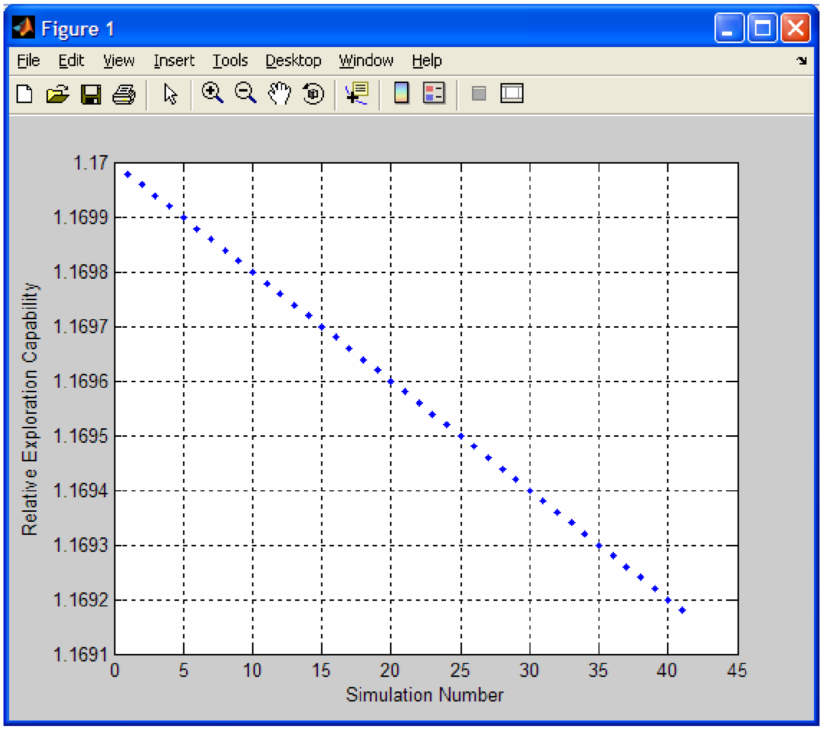
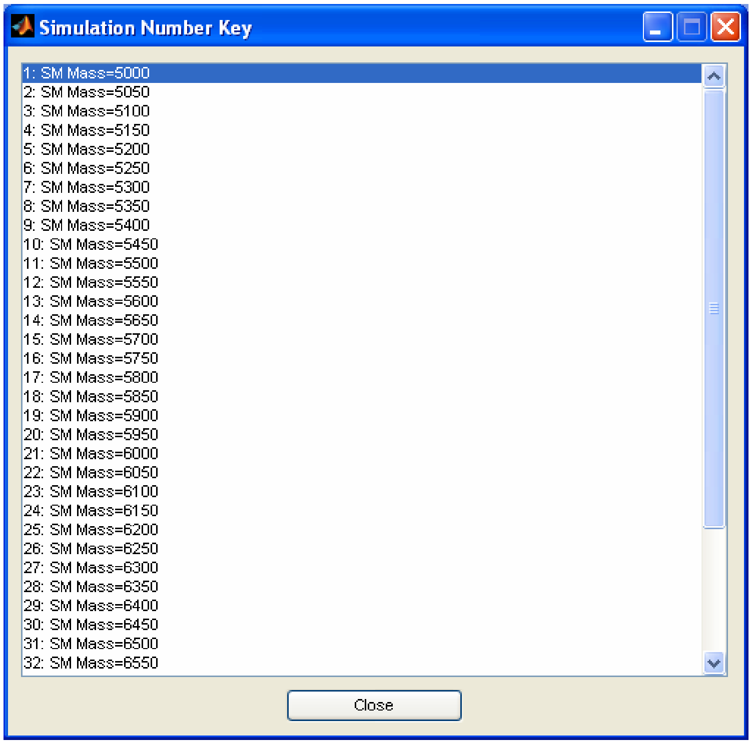


Figure 18: Batch Mode – 2-D MOE plot

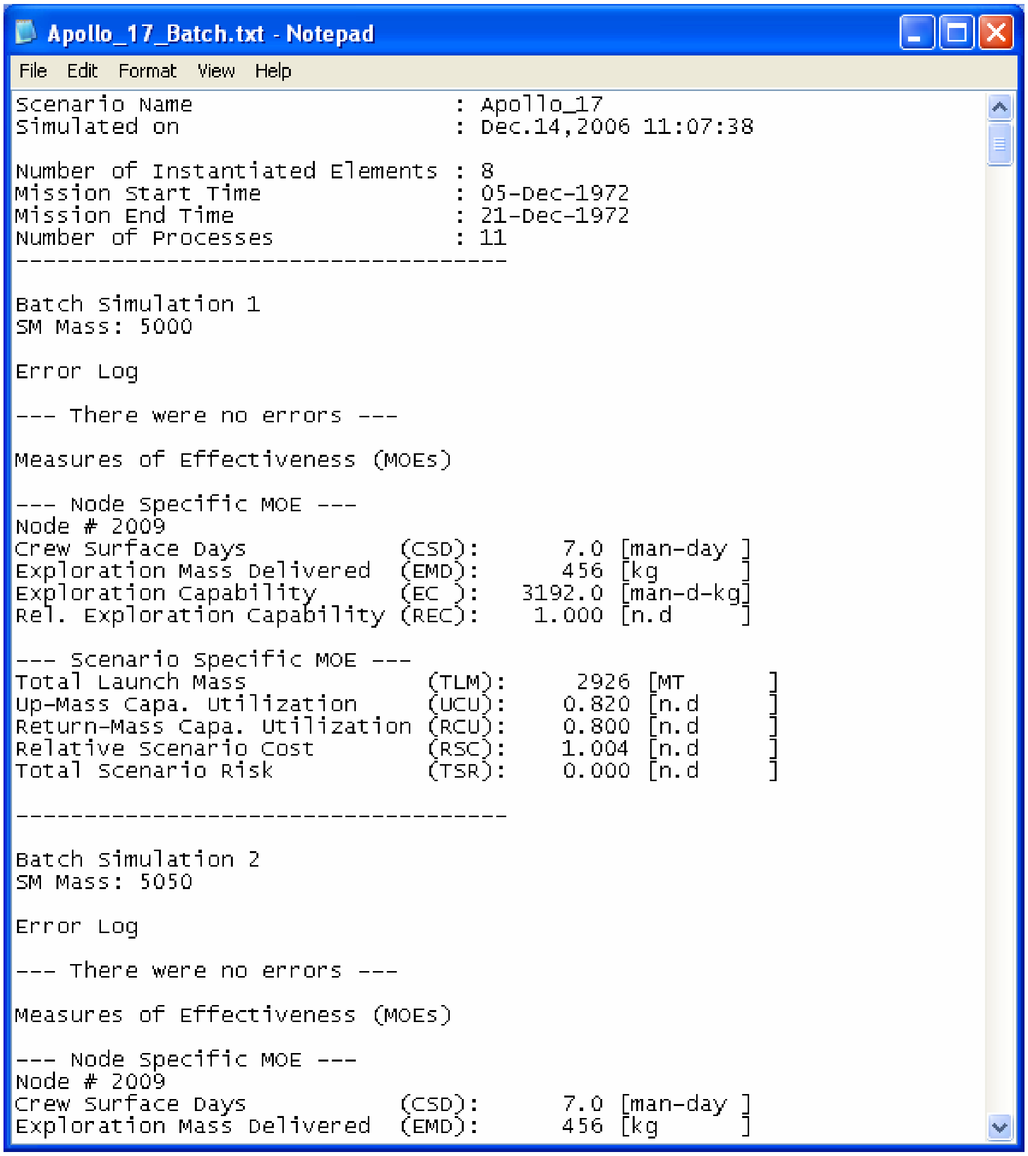
- Push the “Show Simulation Number Key” number to display the element attributes associated with each simulation number (Figure 19).



1: SM Mass=5000
2: SM Mass=5050
3: SM Mass=5100
4: SM Mass=5150
5: SM Mass=5200
6: SM Mass=5250
7: SM Mass=5300
8: SM Mass=5350
9: SM Mass=5400
10: SM Mass=5450
11: SM Mass=5500
12: SM Mass=5550
13: SM Mass=5600
14: SM Mass=5650
15: SM Mass=5700
16: SM Mass=5750
17: SM Mass=5800
18: SM Mass=5850
19: SM Mass=5900
20: SM Mass=5950
21: SM Mass=6000
22: SM Mass=6050
23: SM Mass=6100
24: SM Mass=6150
25: SM Mass=6200
26: SM Mass=6250
27: SM Mass=6300
28: SM Mass=6350
29: SM Mass=6400
30: SM Mass=6450
31: SM Mass=6500
32: SM Mass=6550

Figure 19: Batch Mode – Simulation Number Key

In addition to any plots the user chooses to create, batch mode also outputs a .txt file that logs all errors and MOEs for each scenario. This file will is placed in the scenario folder of the directory the user is working in and is entitled “*Scenario_Name_Batch*”, see Figure 20.



```
Scenario Name                   : Apollo_17
Simulated on                    : Dec.14,2006 11:07:38

Number of Instantiated Elements : 8
Mission Start Time              : 05-Dec-1972
Mission End Time                : 21-Dec-1972
Number of Processes             : 11
------------------------------------

Batch Simulation 1
SM Mass: 5000

Error Log

--- There were no errors ---

Measures of Effectiveness (MOEs)

--- Node Specific MOE ---
Node # 2009
Crew Surface Days           (CSD):      7.0 [man-day ]
Exploration Mass Delivered  (EMD):      456 [kg      ]
Exploration Capability      (EC ):   3192.0 [man-d-kg]
Rel. Exploration Capability (REC):    1.000 [n.d     ]

--- Scenario Specific MOE ---
Total Launch Mass             (TLM):     2926 [MT      ]
Up-Mass Capa. Utilization     (UCU):    0.820 [n.d     ]
Return-Mass Capa. Utilization (RCU):    0.800 [n.d     ]
Relative Scenario Cost        (RSC):    1.004 [n.d     ]
Total Scenario Risk           (TSR):    0.000 [n.d     ]

------------------------------------

Batch Simulation 2
SM Mass: 5050

Error Log

--- There were no errors ---

Measures of Effectiveness (MOEs)

--- Node Specific MOE ---
Node # 2009
Crew Surface Days           (CSD):      7.0 [man-day ]
Exploration Mass Delivered  (EMD):      456 [kg      ]
```

Figure 20: Batch Mode – Text Log File

Another feature of batch mode is to calculate the maximum cargo delivered to an exploration node such as the lunar surface for given sets of element masses and I_{SP}s. This feature currently works only for lunar sorties and is one of the “Process Tools” that is explained later in the Process section in Chapter 3 Main GUI. To calculate the maximum cargo delivered, load a scenario as before, vary the desired element masses and I_{SP}s, but now check “Calculate max. cargo delivered” before pushing “Run Batch Simulation”. In the example below (Figure 21), the “Lunar Sortie” scenario has been loaded and the user has chosen to vary the EDS and LSAM DS I_{SP}s.

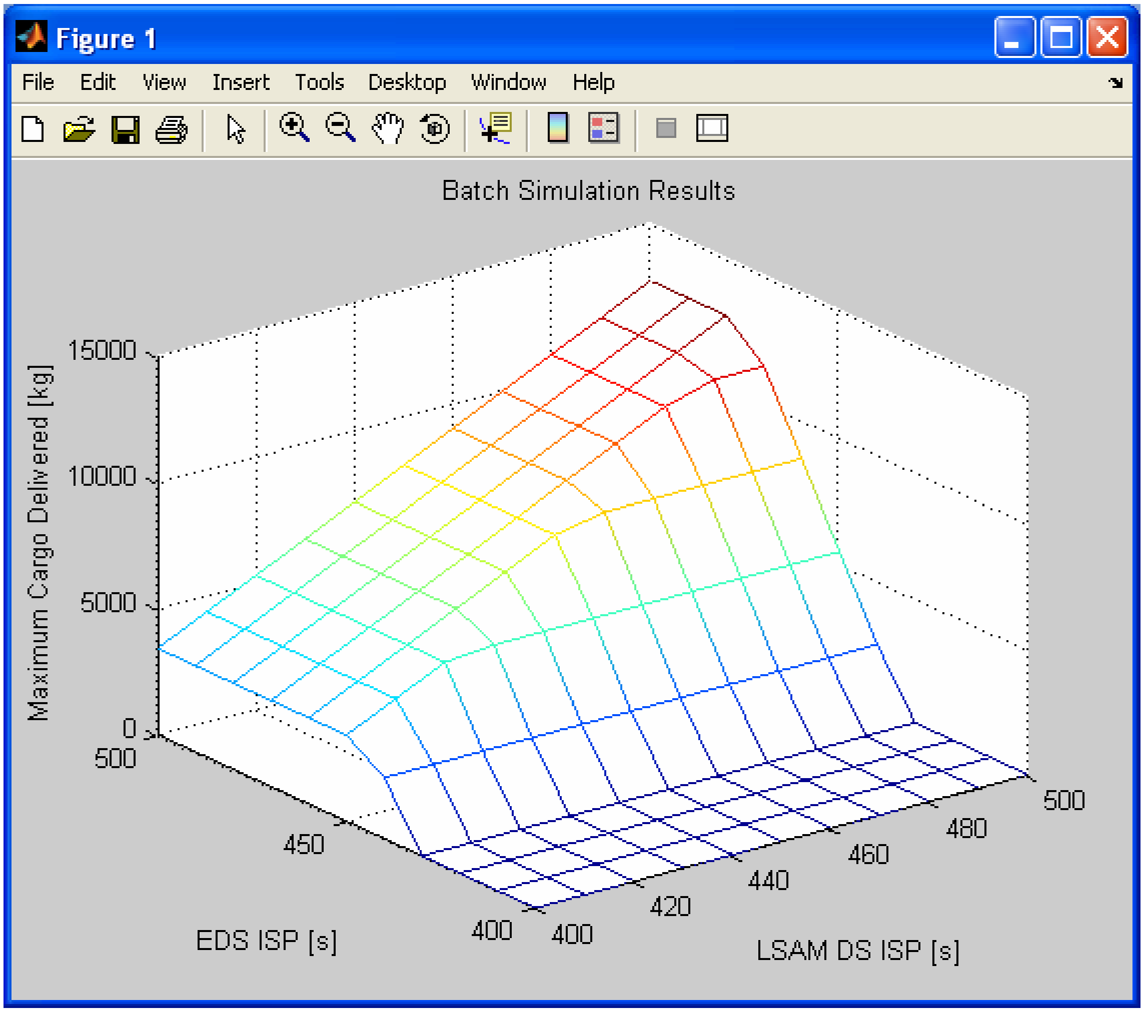


Figure 21: Batch Mode – Process Tools

The batch mode analysis in Figure 21 shows the maximum cargo delivered to the lunar surface [kg] as a function of the Earth Departure Stage (EDS) and Lunar Surface Access Module (LSAM) specific impulse I_{sp} [s]values.

Chapter 3 Main GUI

This chapter describes the Graphical User Interface (GUI) and associated functionality of SpaceNet 1.3 shown in Figure 22. The discussion proceeds screen by screen.

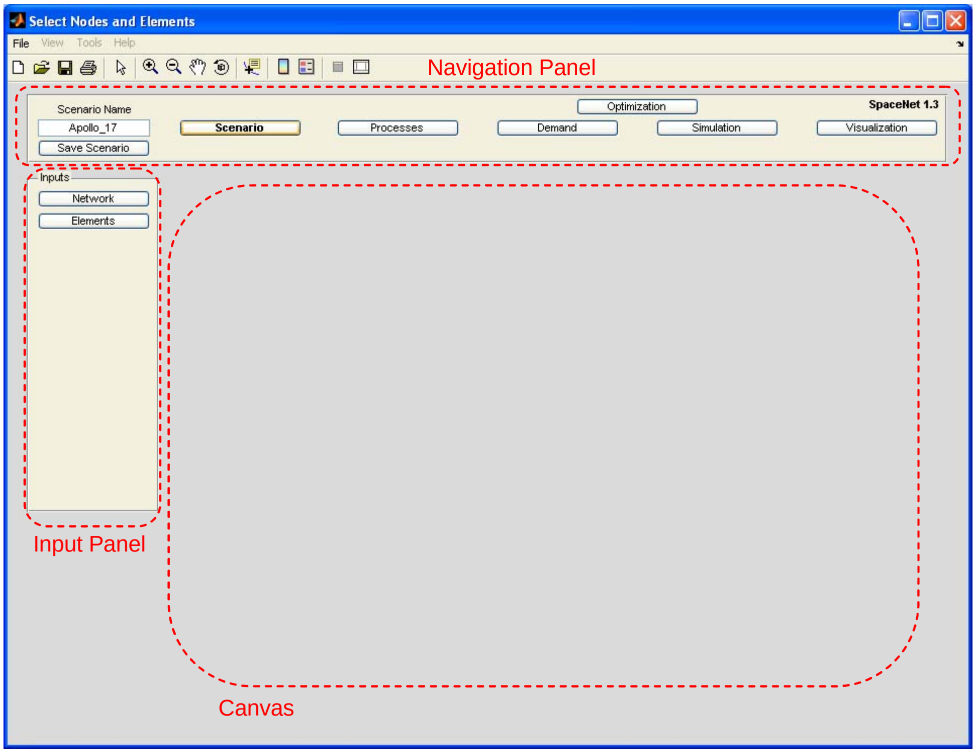


Figure 22: Main GUI

The main GUI has three distinct areas:

- *Navigation Panel* (top)
    - Displays the scenario name
    - Displays the six buttons in the scenario creation process and **bolds** the step the user is currently on
    - The user can save the scenario at any time by pushing the “Save Scenario” button. A structure in the workspace and a .mat file in the \scenario directory will be created or updated with the scenario name

- *Input Panel* (left)
    - The buttons on the input panel change according to which step the user is currently on (from the Navigation Panel)
    - In the example above, the user is on the “Scenario” step and the allowable inputs are “Network” and “Elements”.

- *Canvas* (center)
    - The canvas changes according to which step and input the user is currently on (from the Navigation and Input Panels)

The “**File**” menu at the top of the GUI has three options:

- *Load Scenario*: Closes the main GUI, opens the title GUI, and automatically pushes the “Load Existing Scenario” button
- *Save Scenario*: Saves the current scenario to file as the name entered under “Scenario Name”
- *Exit*: Closes SpaceNet (but leaves Matlab open)

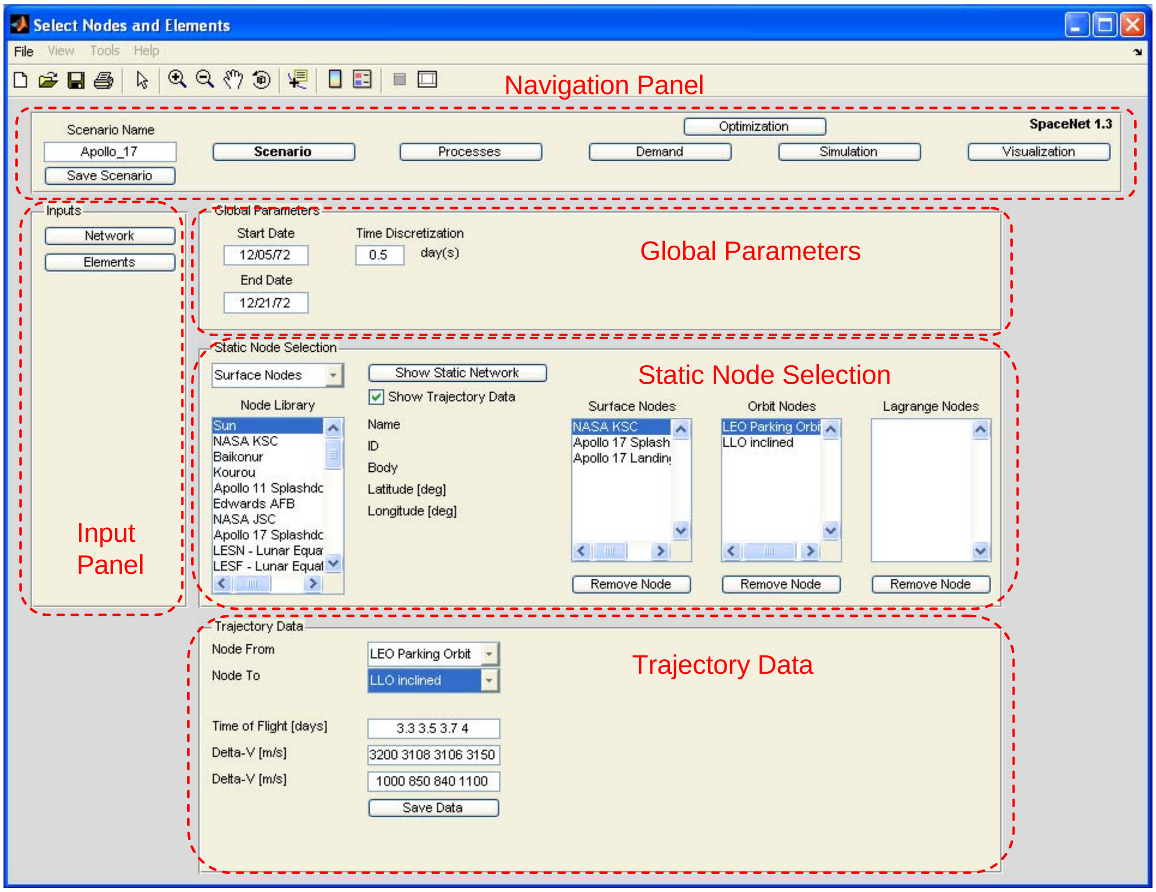
The traditional “View” and “Help” menus are not available in SpaceNet v1.3, but may be included in future releases. The “Tools” menu is enabled when the user is on the “Processes” step and is described in the respective section below.

To follow along in the following sections it is recommended that the user load the existing *Apollo_17* baseline scenario.

Scenario

Network

Push the “Network” button to display the canvas shown in Figure 23.



Scenario Name: Apollo_17

Start Date: 12/05/72

End Date: 12/21/72

Time Discretization: 0.5 day(s)

Surface Nodes

☑ Show Trajectory Data

Node Library:
Sun
NASA KSC
Baikonur
Kourou
Apollo 11 Splashdc
Edwards AFB
NASA JSC
Apollo 17 Splashdc
LESN - Lunar Equa
LESF - Lunar Equat

Surface Nodes:
NASA KSC
Apollo 17 Splash
Apollo 17 Landin

Orbit Nodes:
LEO Parking Orbi
LLO inclined

Node From: LEO Parking Orbit

Node To: LLO inclined

Time of Flight [days]: 3.3 3.5 3.7 4

Delta-V [m/s]: 3200 3108 3106 3150

Delta-V [m/s]: 1000 850 840 1100

Figure 23: Scenario – Network

The “Global Parameters” panel has the following inputs:

- *Start Date*: The scenario start date
- *End Date*: The scenario end date
- *Time Discretization*: The time step (in Earth days) the scenario will use (0.1, 0.5 or 1.0 are recommended…other values have not yet been tested)

The user will use the “Node Library” to select the nodes that will be used in the scenario. Underneath the “Static Node Selection” text the user will be able to filter the surface, orbit, and Lagrange nodes displayed in the “Node Library”. All nodes and their respective information are read from the integrated database.

Single-clicking an item in the “Node Library” will display a brief description of the node to the right including Name, ID, Body, Latitude and Longitude (for surface nodes).

Double-clicking an item in the “Node Library” will add it to the respective listbox in the panel. The node has now been selected for inclusion in the scenario. Checking “Show Trajectory Data” will bring up the “Trajectory Data” panel in the canvas, where the time of flight and delta-V associated with a selected trajectory (for transportation between nodes) can be modified.

In Figure 23, three surface nodes and two orbit nodes have been selected. The user can push the “Remove Node” button to a remove a node that has been accidentally selected.

After all of the nodes that will be used in the scenario have been selected, push “Show Static Network” to display the static network. This will pop up a figure like the one in Figure 24 that displays the selected nodes and how they are connected through feasible trajectories.

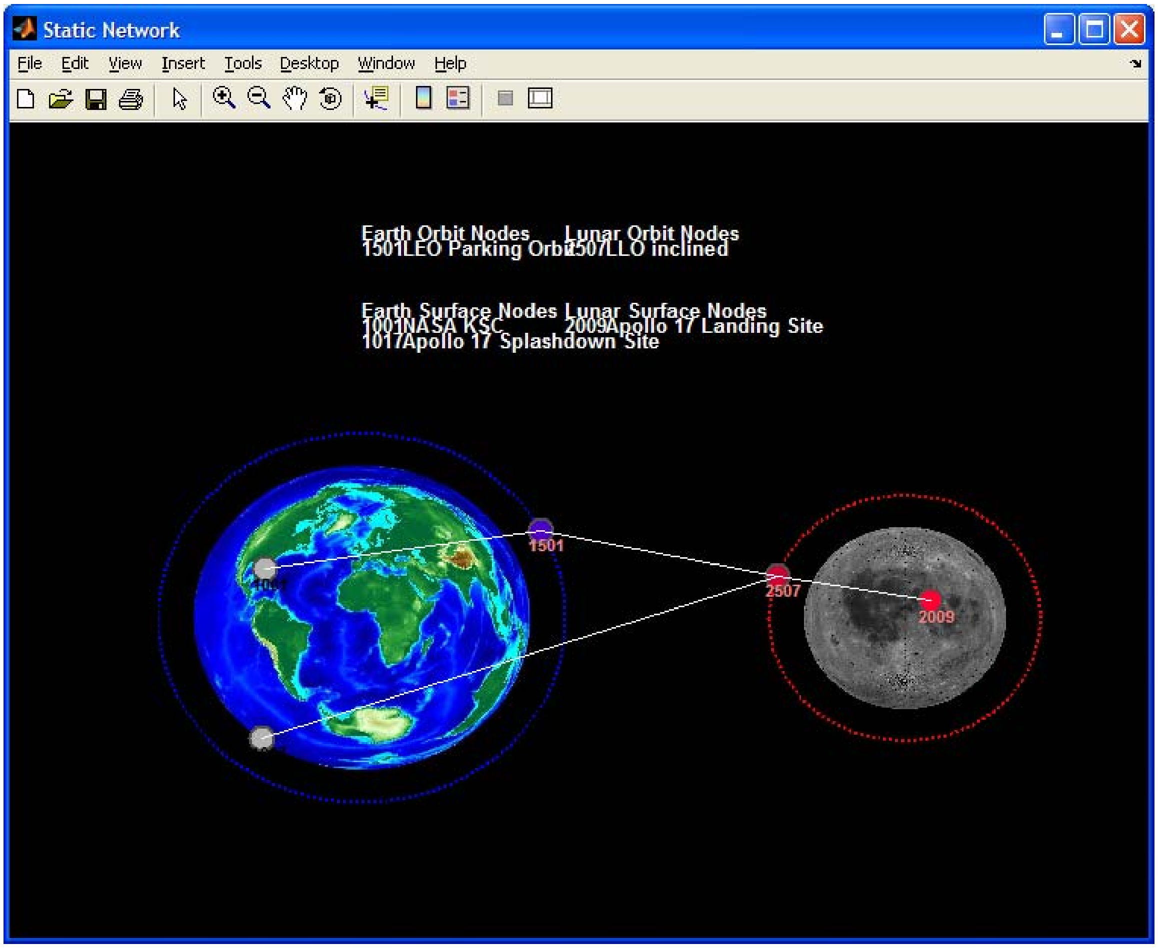


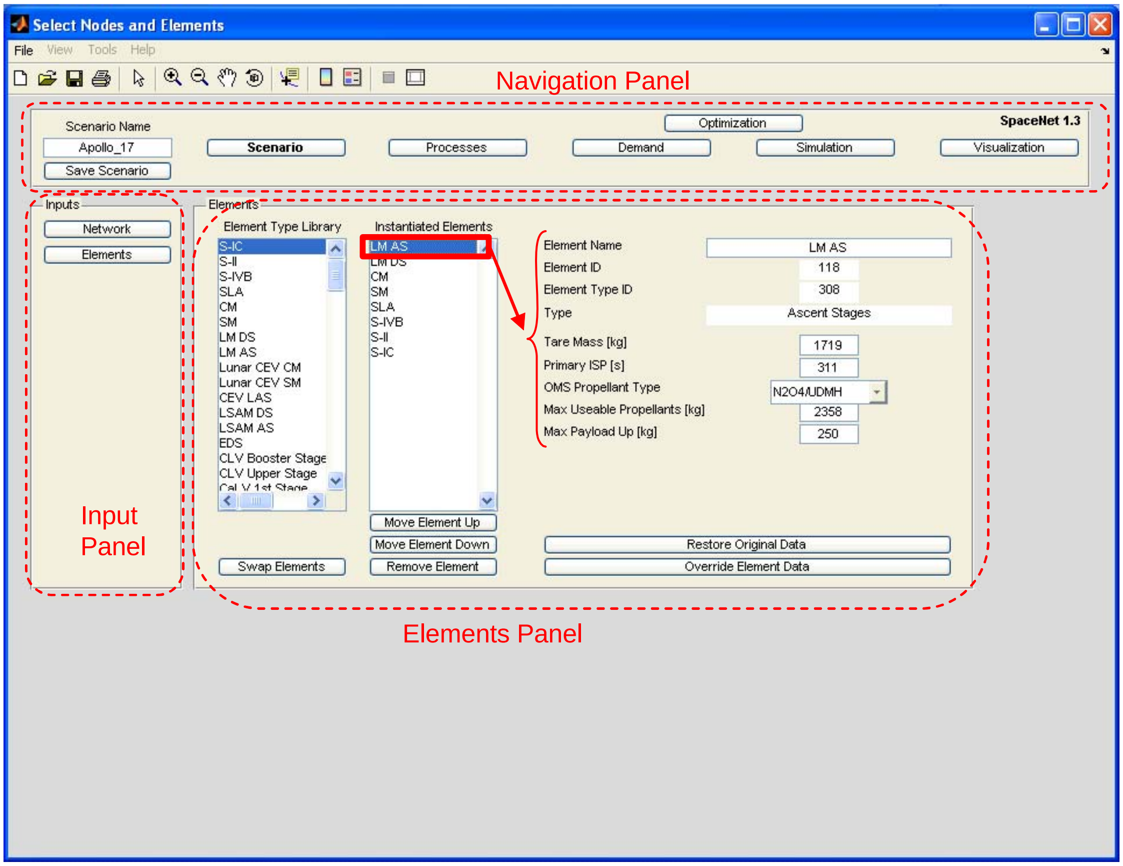
Figure 24: Static Network

If two nodes are not connected with a line, that means that a direct transport from one node to the other is infeasible, or that trajectory data in the integrated database has not yet been entered for those nodes. (Nearby nodes occasionally appear on top of one another, making it difficult to distinguish them in this view.)

The user can always push the "Save Scenario" button at any time to save any progress made. If pressed now, the network information would be saved to file. When saving a scenario a message saying "*scenario_name* has been saved" will flash up shortly.

Elements

Push the "Elements" button to display the "Elements" panel on the canvas (Figure 25).



Scenario Name: Apollo_17

Element Name: LM AS
Element ID: 118
Element Type ID: 308
Type: Ascent Stages
Tare Mass [kg]: 1719
Primary ISP [s]: 311
OMS Propellant Type: N2O4/UDMH
Max Useable Propellants [kg]: 2358
Max Payload Up [kg]: 250

Figure 25: Scenario – Elements

Elements are themselves supply class items in the integrated database that are considered to be "major end items", such as infrastructure items, carriers and propulsion modules.

The user will use the "Element Type Library" listbox to select the elements that will be used in the scenario. All elements and their respective information are read from the integrated database.

Single-clicking an element in the "Element Type Library", right-clicking, then selecting "Show Description" will pop up a brief description of the element. Double-clicking an element in the "Element Library" will create an *instantiation* of the element and add it to the "Instantiated Elements" listbox with its own unique Element ID number. The Element ID number is similar to a serial number or tail number. The user can use shift-clicking or control-clicking to instantiate multiple elements at once. Elements can also be instantiated by right-clicking a selected element in the “Element Type Library” then selecting “Instantiate Element(s)”. Multiple elements can be selected at once and instantiated in this manner. Multiple instantiations of the same element will append a (1), (2), etc… to its name. The “Move Element Up” and “Move Element Down” buttons allow the user to order the instantiated elements for ease of use later in the scenario creation process. An example of this would be to reorder the elements as they appear in the launch stack. The user can push the “Remove Element” button to a remove an element that has been accidentally instantiated.

Single-clicking on an element in the “Element Type Library” and “Instantiated Elements” listboxes will display its attributes on the right of the “Elements” panel. As mentioned before, SpaceNet will create an additional unique “Element ID” for each instantiated element that arbitrarily begins at 111. This is being done to internally keep track of instantiated elements and to save their time histories after simulation.

The attributes that are visible for every element are:

- *Element Name*: The element name
- *Element ID*: The element instantiation ID assigned by SpaceNet (unique number)
- *Element Type ID*: The element type ID from the integrated database (non-unique number). This is the number identifying the type of design of the element.
- *Type*: The element supply class or sub-supply class description
- *Tare Mass*: The element dry mass minus the element accommodation mass

Each element has a unique set of attributes dependent on its supply class or sub-supply class. These sets of attributes include:

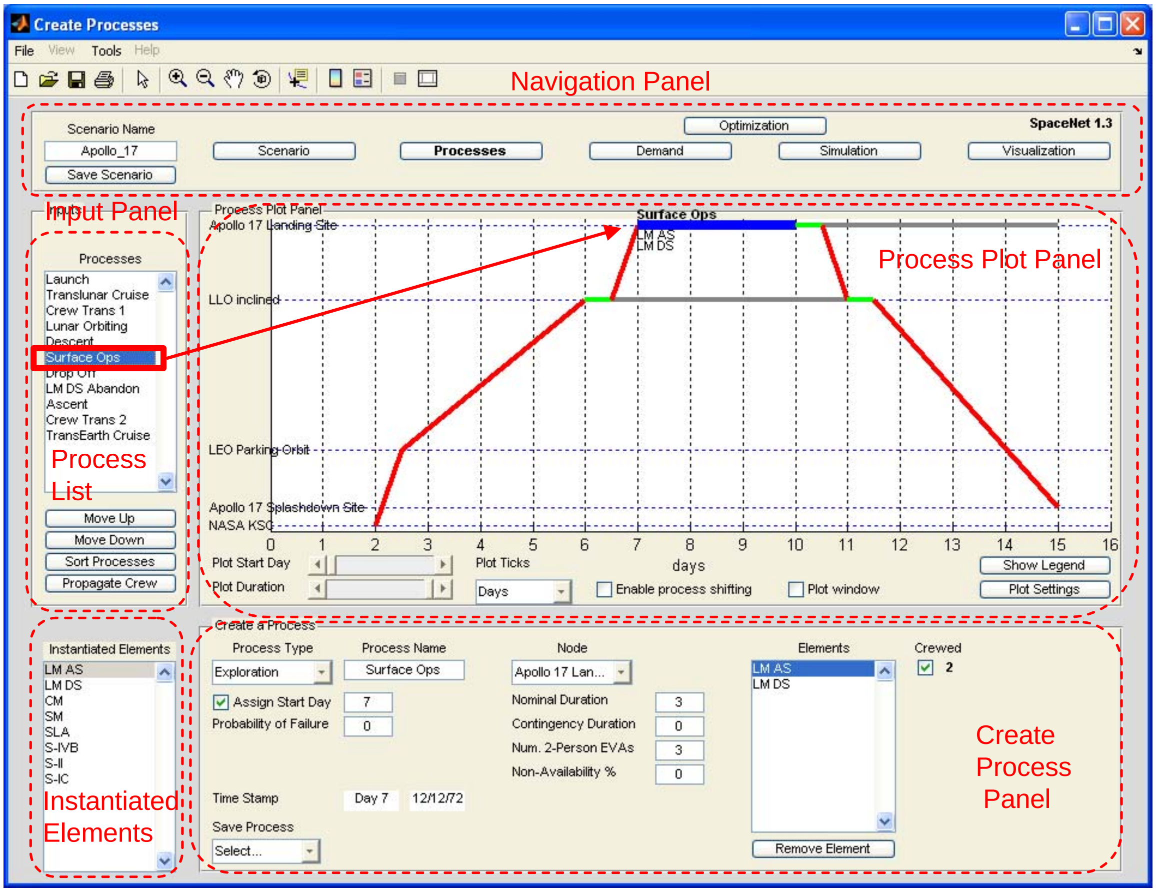
- *Accommodation Mass*: Total mass of structure, fixtures, lockers, etc…required for payload accommodation
- *Primary ISP*: Specific impulse of the primary engine
- *Max Usable Propellant*: Total propellant mass minus ullage
- *Propellant Type ID*: Propellant type
- *Max Transferable Propellant*: Maximum propellant mass that can be transferred to another element (e.g. in order for one element to potentially refuel another)
- *Crew Capacity*: Maximum number of crew the element can hold
- *Max Cargo Up*: Maximum cargo mass the element can transport from Earth
- *Max Cargo Down*: Maximum cargo mass the element can return to Earth
- *Cargo Volume*: Maximum usable cargo volume the element can hold
- *Max Pressurized Dry Cargo*: Maximum pressurized dry cargo mass the element can hold
- *Max Unpressurized Dry Cargo*: Maximum unpressurized dry cargo mass the element can hold
- *Pressurized Volume*: Maximum pressurized volume the element can hold
- *Max Gases*: Maximum mass of gases the element can hold
- *Max Fluids*: Maximum mass of fluids (excluding propellant) the element can hold

Every element attribute except for "Element ID", "Element Type ID", and "Type" can be manually overridden by the user. Push the "Override Element Data" button to save the modifications internally (this will not overwrite the database). Pushing the "Restore Original Data" button will reload the element data stored in the integrated database.

Push the "Save Scenario" button to save the instantiated elements (and modifications) to file. Element modifications are scenario specific and will not be saved to the integrated database.

Create Processes

Push the "Processes" button in the "Navigation" panel to view or define the exploration processes and element shipment (transportation) paths as shown in Figure 26.



Scenario Name: Apollo_17

Process Type: Exploration

Process Name: Surface Ops

Node: Apollo 17 Lan...

☑ Assign Start Day: 7

Probability of Failure: 0

Nominal Duration: 3

Contingency Duration: 0

Num. 2-Person EVAs: 3

Non-Availability %: 0

Time Stamp: Day 7 12/12/72

Save Process: Select...

Crewed: ☑ 2

Figure 26: Processes – Exploration

The element shipment paths are defined in a data structure called "Processes". The processes for the Apollo 17 scenario are displayed in the figure above. At the top of the canvas is a plot of the processes in the scenario (Process List). The instantiated elements are displayed in a listbox at the lower left of the canvas beneath the "Inputs" panel. The user will use this listbox to define what the instantiated elements will do for each process.

The “Create a Process” panel is displayed at the bottom of the canvas. *To create a process, the user must first select the process type from the pull-down menu* (important!)

There are five types of processes:

- *Exploration:* This is the process that defines the main exploration and science activity of the scenario at a specified node (typically a surface node, but not always, exploration can take place at multiple nodes)
- *Proximity Ops:* Rendezvous/Docking, Undocking/Separation, or transposition of elements at the same node
- *Transfer*: Transfer crew and supply items from one element to another at the same node
- *Transport*: Transport elements from one node to another. These processes read trajectory information from the integrated database to provide allowable transport alternatives
- *Wait*: Have elements wait at a node for a specified period of time (in Earth days)

For new processes, enter a name underneath the “Process Name” text. Depending on the process type, data fields will become visible for the user to specify what the process will do. These fields are described for each process below. To add elements to a process, the user can either double-click or shift-click elements in the “Instantiated Elements” listbox, or right-click and select “Add to Process”. Multiple elements can be selected and added at once. The “Remove Element” button will clear the selected element if the user adds an element to a process by accident.

Once the user is finished the “Save Process” menu will become visible. In this menu, there are four options to select from:

- *Add Process*: Adds the process to the end of the existing processes in the “Processes” listbox, or if there are none, it will be the first one
- *Update Process*: Updates the process currently selected in the “Processes” listbox with the current information (useful for correcting mistakes)
- *Insert Process*: Inserts the process just before the process currently selected in the “Processes” listbox
- *Delete Process*: Deletes the process currently selected in the “Processes” listbox

Adding a process will plot it in the “Process Plot Panel”. Exploration processes will appear as blue horizontal bars with their names centered on top and associated elements to the bottom left. Proximity Ops processes will appear as horizontal orange bars, transport processes are red diagonal lines, transfer processes are green horizontal lines, and wait processes are gray horizontal lines. The user can push “Show Legend” to display the process color key. The user can change the appearance of the process plot by modifying the following plot options displayed beneath the axes:

- *Plot Start Day*: Change time period the plot begins (shifts processes to the right)
- *Plot Duration*: Change time period the plot ends (shifts processes to the left)

- *Plot Ticks*: Changes the tick marks at the bottom of the plot to display either days, weeks, months, or years (Earth time)
- *Enable process shifting*: This allows future or past processes to shift in time depending on the current process's duration.
- *Plot Window*: Centers the plot on the current process, with a window around it (in days) whose width is determined by the user.

Existing processes will appear in the "Processes" listbox in the upper left portion of the "Inputs" panel. Selecting an existing process will display its attributes and highlight its respective colored bar in the plot. Push "Save Scenario" to save the processes to file.

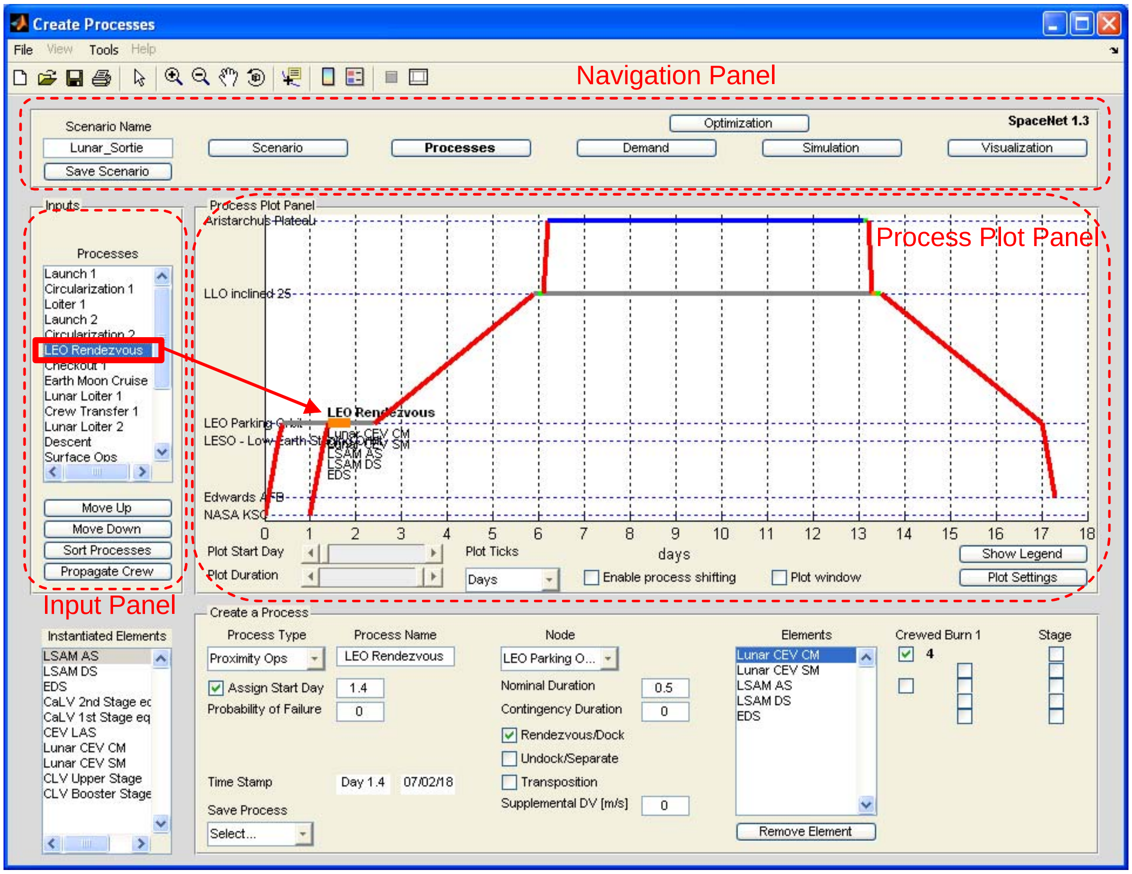
Exploration

The "Exploration" process (highlighted above in Figure 26) has several inputs:

- *Process Type*: Exploration
- *Process Name*: The process name (has to be unique)
- *Node*: The node where exploration occurs
- *Nominal Duration*: Nominal duration (in days)
- *Contingency Duration*: Contingency duration (in days)
- *Num. 2-Person EVAs*: Number of two-person EVAs during the process
- *Non-Availability %:* The percentage of time that the crew is not available to perform science (due to sleep, meals, maintenance …) during the process
- *Elements*: The instantiated elements involved in the exploration
- *Crewed:* Check box for elements that are crewed (and the number of crew)
- *Probability of Failure:* The probability that this process will fail, contributing to a Loss of Mission (LoM), Loss of Vehicle (LoV), or Loss of Crew (LoC) event.
- *Time Stamp*: The day (0 is the scenario start date) and date at which the process begins
- *Assign Start Day:* A user assigned start day for the process

Proximity Ops

The “Proximity Ops” (orange horizontal bar in Figure 27) process is used to describe rendezvous/docking, undocking/separation, and transposition events.



Scenario Name: Lunar_Sortie

Plot Ticks: Days

☐ Enable process shifting

☐ Plot window

Process Type: Proximity Ops

Process Name: LEO Rendezvous

Node: LEO Parking O...

☑ Assign Start Day: 1.4

Probability of Failure: 0

Nominal Duration: 0.5

Contingency Duration: 0

☑ Rendezvous/Dock

☐ Undock/Separate

☐ Transposition

Supplemental DV [m/s]: 0

Time Stamp: Day 1.4 07/02/18

Save Process: Select...

Elements: Lunar CEV CM, Lunar CEV SM, LSAM AS, LSAM DS, EDS

Crewed Burn 1: ☑ 4 ☐

Figure 27: Processes – Proximity Ops

Proximity Ops has the following attributes:

- *Process Type*: Proximity Ops
- *Process Name*: The process name (has to be unique)
- *Node*: The node where proximity ops occur
- *Nominal Duration*: Nominal duration (in days)
- *Contingency Duration*: Contingency duration (in days)
- *Rendezvous/Dock, Undock/Separate, Transposition:* Check which type of proximity ops process is occurring
- *Supplemental DV*: The delta-V required to perform this process
- *Elements*: The elements involved in the proximity ops
- *Crewed:* A checkbox to indicate which (if any) element is crewed

- *Probability of Failure:* The probability that this process will fail, contributing to a Loss of Mission (LoM), Loss of Vehicle (LoV), or Loss of Crew (LoC) event.
- *Time Stamp*: The day (0 is the scenario start date) and date at which the process begins
- *Assign Start Day:* A user assigned start day for the process

Transfer

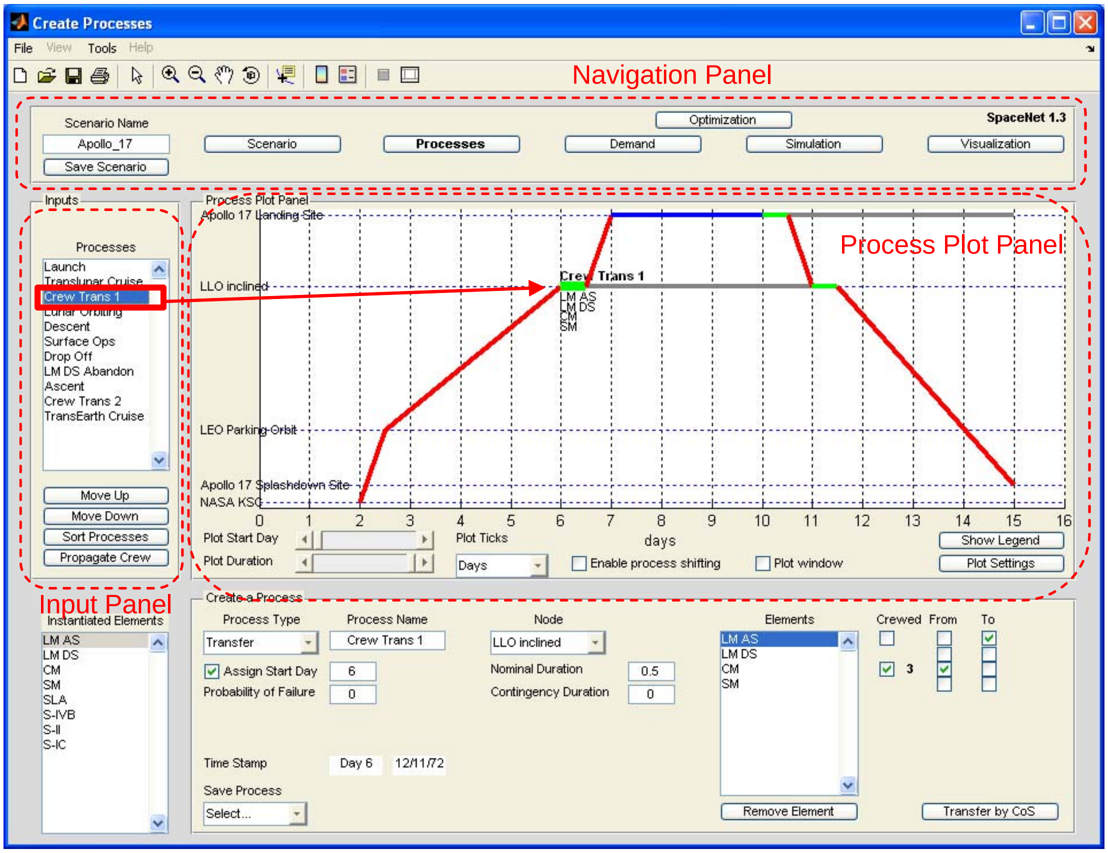

Scenario Name: Apollo_17
Process Type: Transfer
Process Name: Crew Trans 1
Node: LLO inclined
☑ Assign Start Day: 6
Probability of Failure: 0
Nominal Duration: 0.5
Contingency Duration: 0
Time Stamp: Day 6 12/11/72
Save Process: Select...

Figure 28: Processes – Transfer

The "Transfer" process (displayed in Figure 28 as a short green horizontal line) has the following attributes:

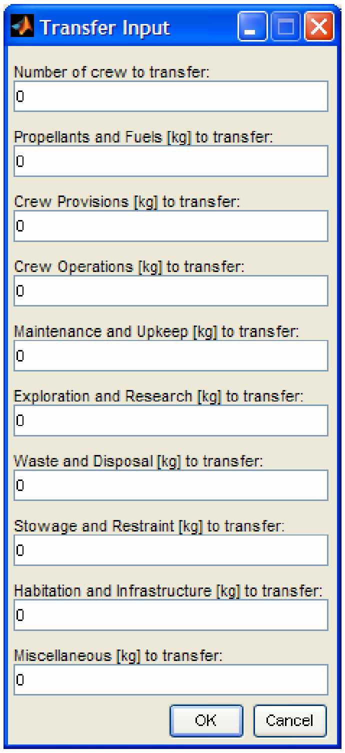
- *Process Type*: Transfer
- *Process Name*: The process name (has to be unique)
- *Node*: The node where the transfer occurs
- *Nominal Duration*: Nominal duration (in days)
- *Contingency Duration*: Contingency duration (in days)
- *Elements*: The elements involved in the transfer
- *Crewed:* Check which elements are crewed (and the number of crew)
- **From*: Check the element to transfer from.
- *To*: Check the element to transfer to.

- *Probability of Failure:* The probability that this process will fail, contributing to a Loss of Mission (LoM), Loss of Vehicle (LoV), or Loss of Crew (LoC) event.
- *Time Stamp*: The day (0 is the scenario start date) and date at which the process begins
- *Assign Start Day:* A user assigned start day for the process

*NOTE: If the *From* checkbox is not checked for any element, then SpaceNet assumes that cargo is picked up from a node and transferred to an element. If the *To* checkbox is not checked, then SpaceNet assumes that cargo is transferred from an element to a node.

When selecting the elements for transfer processes, the user must also select the inactive elements that are traveling with the elements that are involved in the transfer.

In order to specify what is to be transferred, push the “Transfer by CoS” button. This will bring up the dialog box shown in Figure 29, with the following attributes:



Number of crew to transfer: 0

Propellants and Fuels [kg] to transfer: 0

Crew Provisions [kg] to transfer: 0

Crew Operations [kg] to transfer: 0

Maintenance and Upkeep [kg] to transfer: 0

Exploration and Research [kg] to transfer: 0

Waste and Disposal [kg] to transfer: 0

Stowage and Restraint [kg] to transfer: 0

Habitation and Infrastructure [kg] to transfer: 0

Miscellaneous [kg] to transfer: 0

Figure 29: Transfer by COS

- *Crew*: The number of crew to transfer
- *Propellants and Fuels*: The mass of propellants to transfer
- *Crew Provisions*: The mass of crew provisions to transfer
- *Crew Operations*: The mass of crew operations to transfer
- *Maintenance and Upkeep*: The mass of spares to transfer
- *Exploration and Research*: The mass of exploration items to transfer
- *Waste and Disposal*: The mass of waste equipment to transfer
- *Stowage and Restraint*: The mass of stowage equipment to transfer
- *Habitation and Infrastructure*: The mass of habitation and infrastructure items to transfer
- *Miscellaneous:* The mass of miscellaneous items to transfer

(The transfer process will be updated in the future to allow the transfer of individual supply items and bags, such as CTBs, from one element to another)

Transport

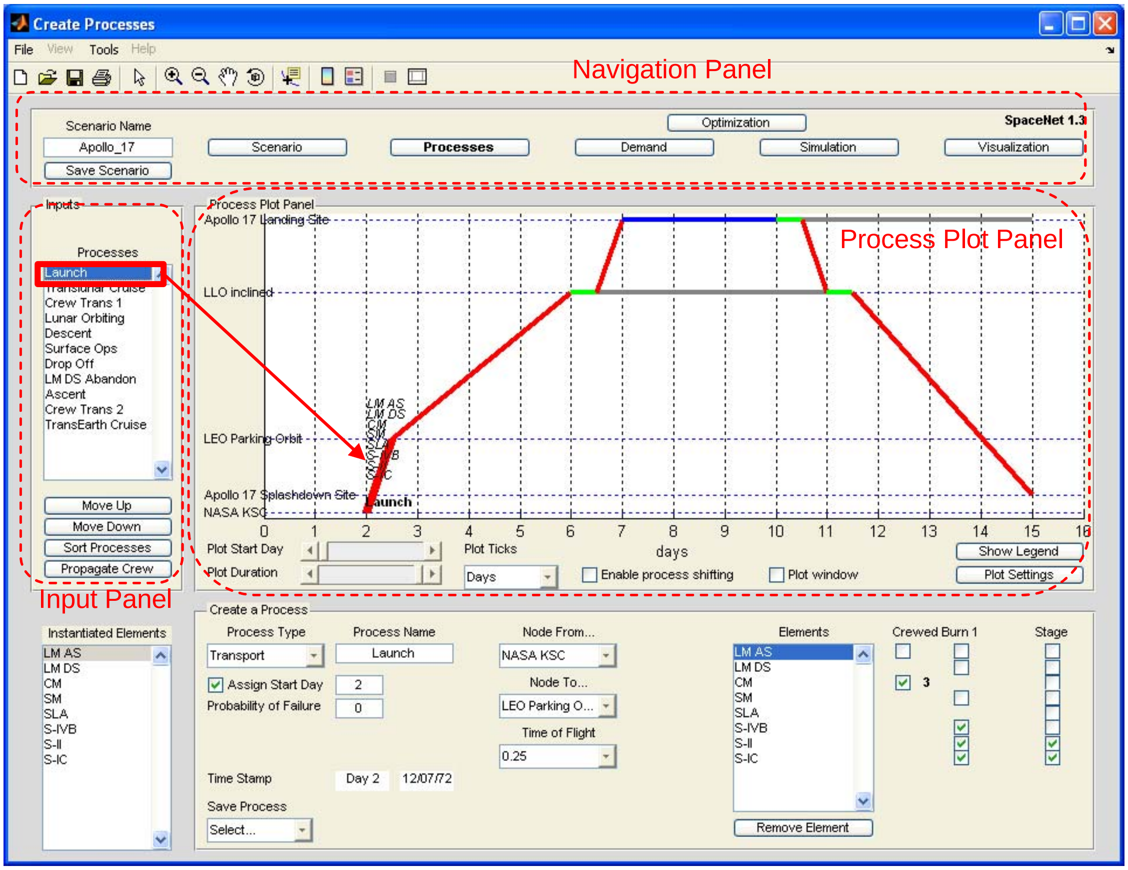

Scenario Name: Apollo_17
Process Type: Transport
Process Name: Launch
☑ Assign Start Day: 2
Probability of Failure: 0
Node From...: NASA KSC
Node To...: LEO Parking O...
Time of Flight: 0.25
Time Stamp: Day 2 12/07/72
Save Process: Select...
Plot Ticks: Days
☐ Enable process shifting
☐ Plot window

Figure 30: Processes – Transport

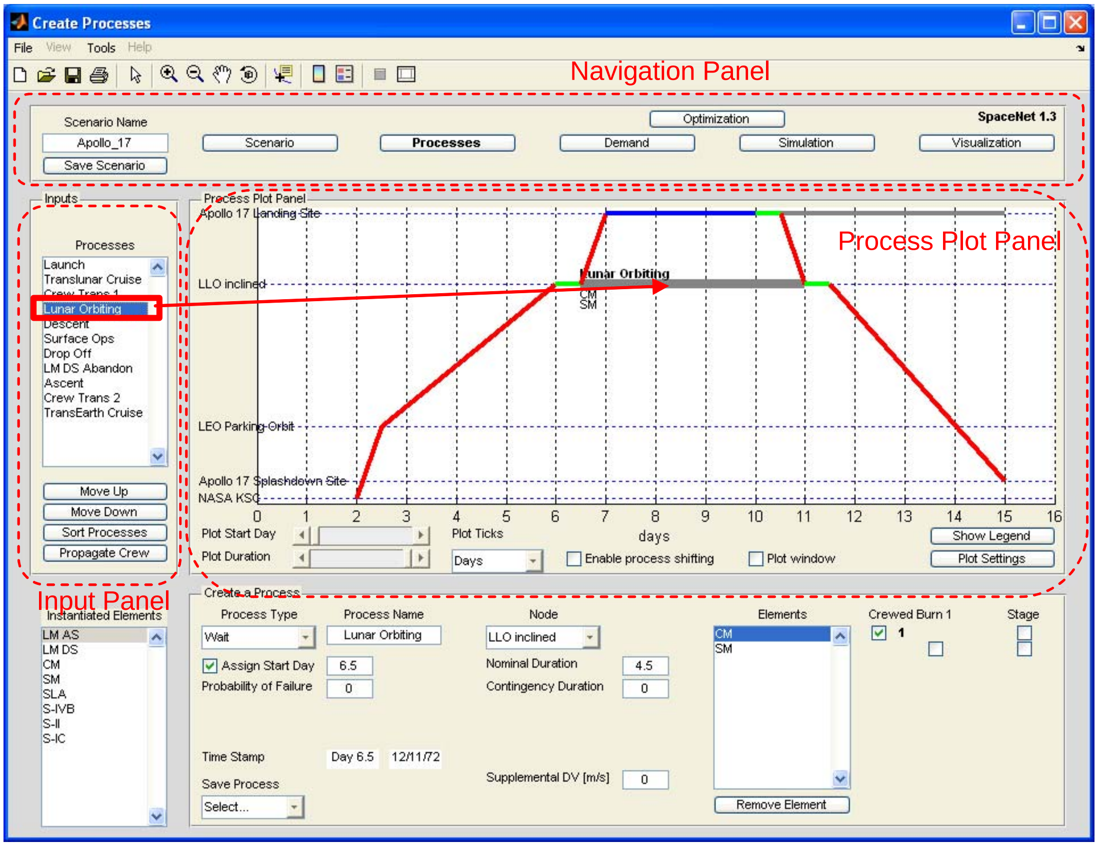
The “Transport” process (displayed in red in Figure 30) has the following attributes:

- *Process Type*: Transport
- *Process Name*: The process name (has to be unique)
- *Node From*: The node to depart from
- *Node To*: The node to arrive at
- *Time of Flight*: The time of flight for the given departure and arrival nodes (selected from values loaded from the integrated database, which stores feasible impulsive trajectories between pairs of nodes)
- *Elements*: The elements to transport together (at this point SpaceNet 1.3 does not allow flotillas, i.e. multiple but separate stacks of elements traveling together on the same arc in the time-expanded network)
- *Crewed:* Check which elements are crewed (and the number of crew)
- **Burn 1*: If the element to transport is a propulsive element, a checkbox will appear next to its name. The user checks boxes to specify what elements execute burns for a transport. Burn 1 is the departure burn.

- *Burn2*: If the element to transport is a propulsive element and the trajectory specified by the departure and arrival nodes requires two burns, a check box will appear next to its name. Burn 2 is the arrival burn where appropriate.
- *Stage*: The user can check boxes to specify what elements are discarded during a transport after a particular burn.
- *Probability of Failure:* The probability that this process will fail, contributing to a Loss of Mission (LoM), Loss of Vehicle (LoV), or Loss of Crew (LoC) event.
- *Time Stamp*: The day (0 is the scenario start date) and date at which the process begins
- *Assign Start Day:* A user assigned start day for the process

*NOTE: The order of the elements to transport is significant because the *Burn1*, *Burn2*, and *Stage* checkboxes specify a sequence of events. In the example above, the “S-IC” is first burned, discarded, then “SII” is burned, discarded, then finally the S-IVB is burned. The user will want to arrange the elements for transport processes “bottom up”.

Wait



Scenario Name: Apollo_17

Plot Ticks: Days

☐ Enable process shifting

☐ Plot window

Process Type: Wait

Process Name: Lunar Orbiting

Node: LLO inclined

☑ Assign Start Day: 6.5

Probability of Failure: 0

Nominal Duration: 4.5

Contingency Duration: 0

Time Stamp: Day 6.5 12/11/72

Supplemental DV [m/s]: 0

Save Process: Select...

Elements: CM, SM

Crewed Burn 1: ☑ 1 ☐

Stage: ☐ ☐

Figure 31: Processes – Wait

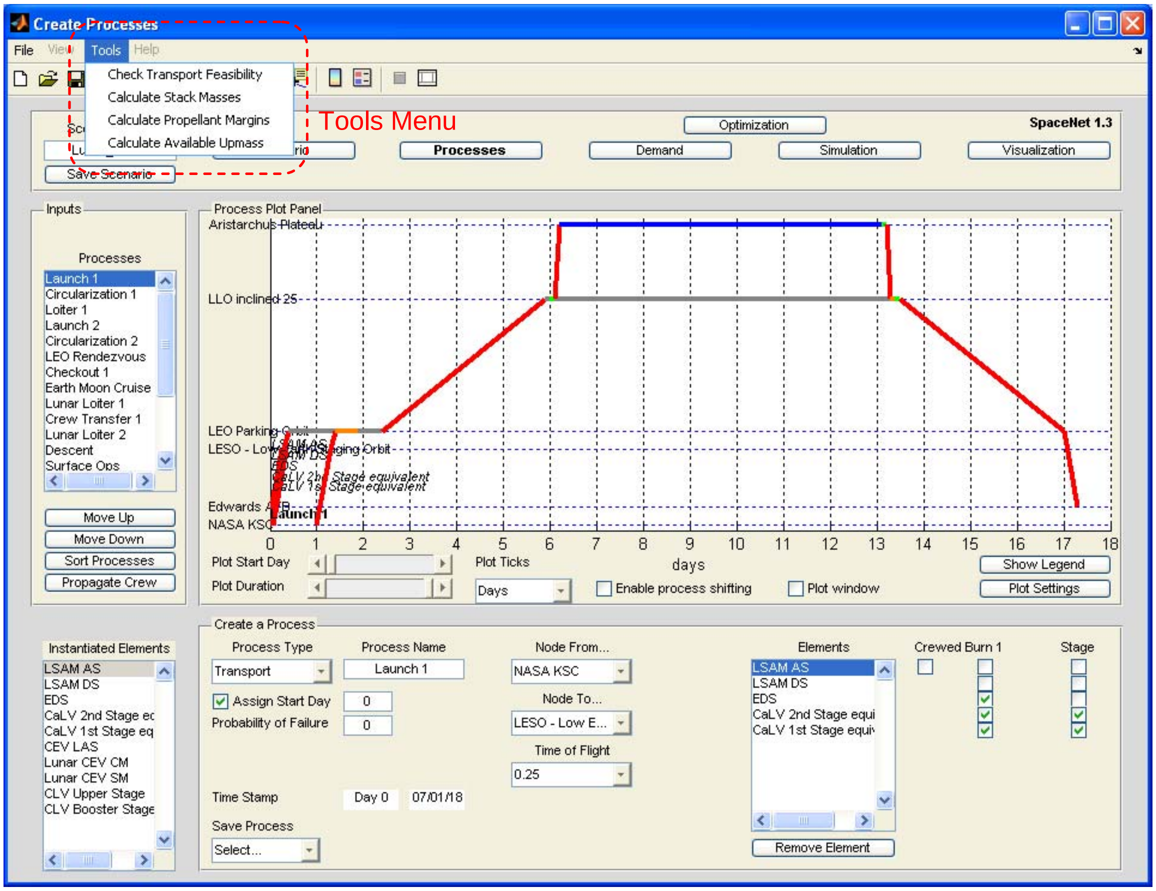
The “Wait” process (displayed above in Figure 31 as a gray horizontal bar) has the following attributes:

- *Process Type*: Wait
- *Process Name*: The process name (has to be unique)
- *Nodes*: The node to wait at
- *Elements*: The elements that are waiting
- *Nominal Duration*: Nominal duration (in days)
- *Contingency Duration*: Contingency duration (in days)
- *Supplemental DV*: The delta-V required for this process. For wait processes, this would include orbit maintenance burns.
- *Crewed:* Check which elements are crewed (and the number of crew)
- *Burn 1:* Check which elements burn for the supplemental DV
- *Stage:* Check which elements are discarded after this process
- *Probability of Failure:* The probability that this process will fail, contributing to a Loss of Mission (LoM), Loss of Vehicle (LoV), or Loss of Crew (LoC) event.
- *Time Stamp*: The day (0 is the scenario start date) and date at which the process begins
- **Assign Start Day:* A user assigned start day for the process

*NOTE: For all processes, the "Assign Start Day" checkbox and edit textbox to its right allow the user to specify what day the process begins. If this box is left un-checked, SpaceNet will automatically assign the latest date a waiting element has been used (in previous processes) as the "Time Stamp". If none of the waiting elements have been used before, the "Time Stamp" defaults to the scenario start date.

Process Tools

Upon completion of entering all of the processes in a scenario, the user has the option to select from a variety of tools to analyze the processes. These tools allow a user to analyze a scenario at a high level without demands or initial conditions. These process tools can be accessed from the "Tools" menu at the top of the GUI, see Figure 32.



Processes
Launch 1
Circularization 1
Loiter 1
Launch 2
Circularization 2
LEO Rendezvous
Checkout 1
Earth Moon Cruise
Lunar Loiter 1
Crew Transfer 1
Lunar Loiter 2
Descent
Surface Ops

Plot Ticks: Days

Instantiated Elements
LSAM AS
LSAM DS
EDS
CaLV 2nd Stage ec
CaLV 1st Stage eq
CEV LAS
Lunar CEV CM
Lunar CEV SM
CLV Upper Stage
CLV Booster Stage

Process Type: Transport
Process Name: Launch 1
Node From...: NASA KSC
Node To...: LESO - Low E...
Time of Flight: 0.25
☑ Assign Start Day: 0
Probability of Failure: 0
Time Stamp: Day 0 07/01/18
Save Process: Select...

Elements	Crewed Burn 1	Stage
LSAM AS	☐	☐
LSAM DS	☐	☐
EDS	☑	☐
CaLV 2nd Stage equi	☑	☑
CaLV 1st Stage equi	☑	☑

Figure 32: Process Tools

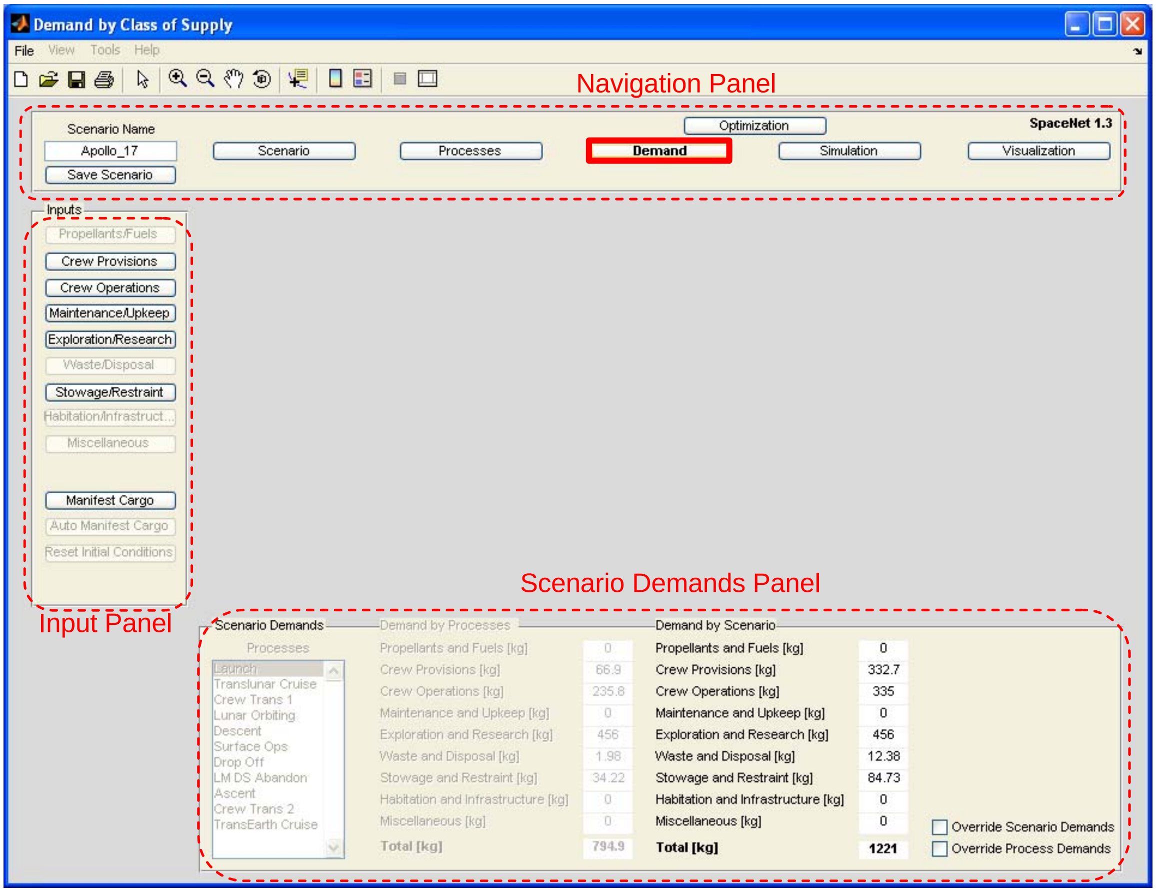
The process tools currently available in SpaceNet v1.3 are:

- *Check Transport Feasibility*: Checks if all transport processes are feasible, i.e. if they satisfy the rocket equation
- *Calculate Stack Masses*:: Calculates the aggregate mass at key points in the scenario (such as launch and trans-lunar injection)
- *Calculate Propellant Margins*: Calculates the propellant margins for the elements that are involved in transport processes. The margin is defined as the leftover propellant divided by the total propellant used for that element
- *Calculate Max Cargo Delivered*: Calculates the maximum amount of cargo, in kg, that can be delivered to the lunar surface. This feature currently works for lunar sorties only

Demand

Once the network has been created, elements have been instantiated, and processes have been created, the “Demand” button on the navigation panel will be enabled.

Push the “Demand” button to display the canvas below.



Demand by Processes		Demand by Scenario	
Propellants and Fuels [kg]	0	Propellants and Fuels [kg]	0
Crew Provisions [kg]	66.9	Crew Provisions [kg]	332.7
Crew Operations [kg]	235.8	Crew Operations [kg]	335
Maintenance and Upkeep [kg]	0	Maintenance and Upkeep [kg]	0
Exploration and Research [kg]	456	Exploration and Research [kg]	456
Waste and Disposal [kg]	1.98	Waste and Disposal [kg]	12.38
Stowage and Restraint [kg]	34.22	Stowage and Restraint [kg]	84.73
Habitation and Infrastructure [kg]	0	Habitation and Infrastructure [kg]	0
Miscellaneous [kg]	0	Miscellaneous [kg]	0
Total [kg]	794.9	**Total [kg]**	**1221**

Figure 33: Demand

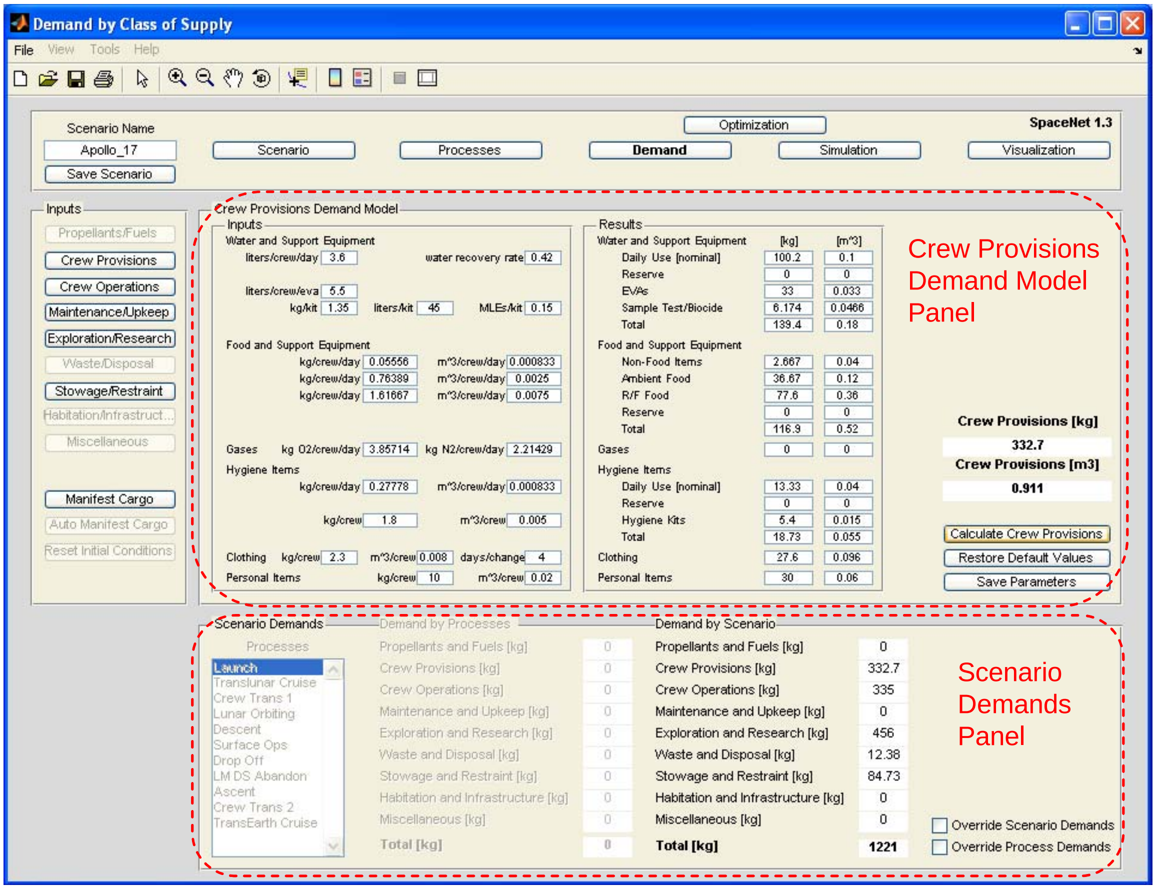
The input panel now shows the supply classes for which demand models have been defined to date:

- *Crew Provisions (COS 2)* → *water, gases, food, clothing, …*
- *Crew Operations (COS 3)* → *EVA equipment, computers, …*
- *Maintenance/Upkeep (COS 4)* → *tools, spares*
- *Exploration/Research (COS 6)* → *science tools, surface exploration items*
- *Stowage/Restraint (COS 5)* → *containers, bags,…*

Demands for each of the supply classes above are calculated for each process in the "Processes" listbox (see Figure 26, left side). Each demand model will calculate and sum the process demands to yield a total scenario demand. These demands are displayed in the "Demand by Scenario" box in the lower right of the canvas. Stowage should be calculated last because it depends on results from the other classes of supply. Push the "Save Scenario" button to save all demands and demand parameters to file. Appendix C contains the mathematical equations and parameters that are used to generate the demand in the various classes of supply.

Crew Provisions (COS 2)

Push the “Crew Provisions” button to display the “Crew Provisions Demand Model” panel, see Figure 34. See Appendix C for a detailed description of the model.



Scenario Name: Apollo_17

Crew Provisions Demand Model

Inputs

Water and Support Equipment
liters/crew/day 3.6 water recovery rate 0.42
liters/crew/eva 5.5
kg/kit 1.35 liters/kit 45 MLEs/kit 0.15

Food and Support Equipment
kg/crew/day 0.05556 m^3/crew/day 0.000833
kg/crew/day 0.76389 m^3/crew/day 0.0025
kg/crew/day 1.61667 m^3/crew/day 0.0075

Gases kg O2/crew/day 3.85714 kg N2/crew/day 2.21429

Hygiene Items
kg/crew/day 0.27778 m^3/crew/day 0.000833
kg/crew 1.8 m^3/crew 0.005

Clothing kg/crew 2.3 m^3/crew 0.008 days/change 4
Personal Items kg/crew 10 m^3/crew 0.02

Results

Water and Support Equipment	[kg]	[m^3]
Daily Use [nominal]	100.2	0.1
Reserve	0	0
EVAs	33	0.033
Sample Test/Biocide	6.174	0.0466
Total	139.4	0.18
Food and Support Equipment		
Non-Food Items	2.667	0.04
Ambient Food	36.67	0.12
R/F Food	77.6	0.36
Reserve	0	0
Total	116.9	0.52
Gases	0	0
Hygiene Items		
Daily Use [nominal]	13.33	0.04
Reserve	0	0
Hygiene Kits	5.4	0.015
Total	18.73	0.055
Clothing	27.6	0.096
Personal Items	30	0.06

Crew Provisions [kg] 332.7
Crew Provisions [m3] 0.911

Scenario Demands

Demand by Processes		Demand by Scenario	
Propellants and Fuels [kg]	0	Propellants and Fuels [kg]	0
Crew Provisions [kg]	0	Crew Provisions [kg]	332.7
Crew Operations [kg]	0	Crew Operations [kg]	335
Maintenance and Upkeep [kg]	0	Maintenance and Upkeep [kg]	0
Exploration and Research [kg]	0	Exploration and Research [kg]	456
Waste and Disposal [kg]	0	Waste and Disposal [kg]	12.38
Stowage and Restraint [kg]	0	Stowage and Restraint [kg]	84.73
Habitation and Infrastructure [kg]	0	Habitation and Infrastructure [kg]	0
Miscellaneous [kg]	0	Miscellaneous [kg]	0
Total [kg]	0	**Total [kg]**	1221

☐ Override Scenario Demands
☐ Override Process Demands

Figure 34: Demand – Crew Provisions

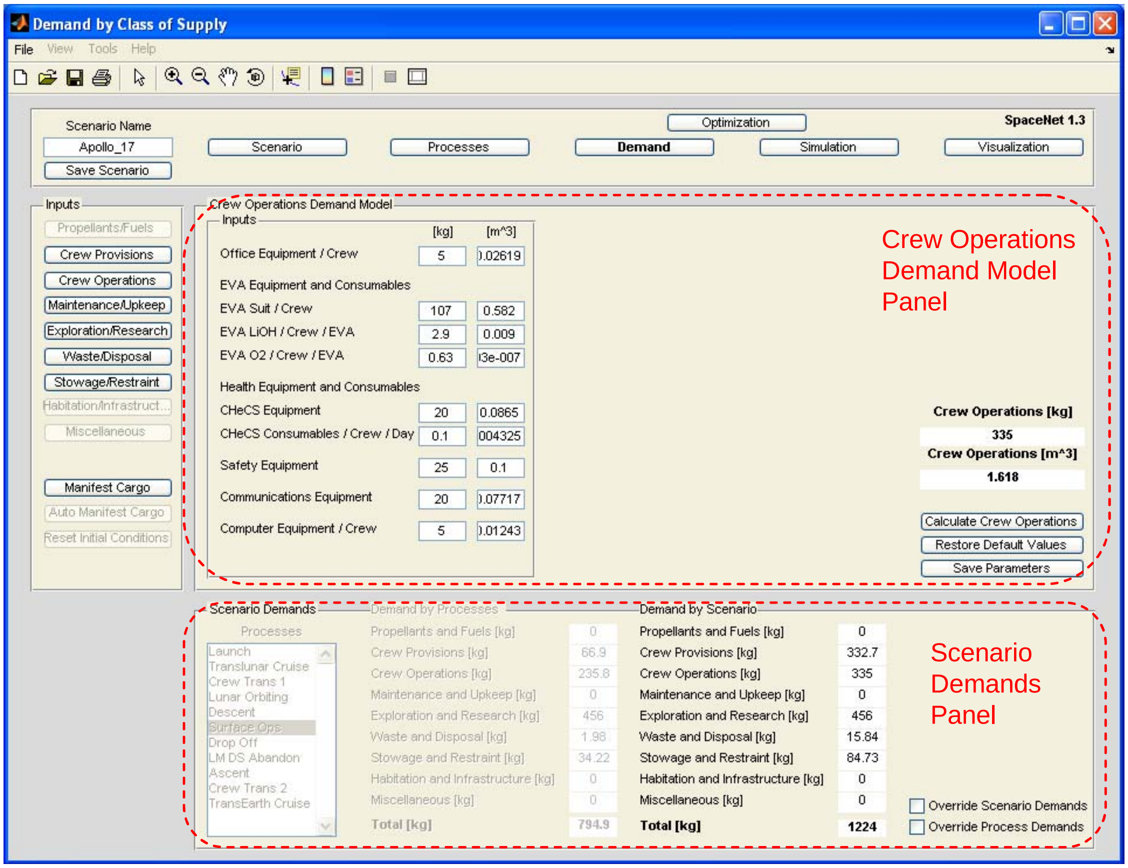
The model takes as input the process parameters displayed in the “Create Processes” panel (crew size, durations, number of EVAs, etc.) and the input parameters displayed in the “Inputs” panel in the “Crew Provisions Demand Model” panel.

Push “Calculate Crew Provisions” to run the demand model and display the results in the “Results” panel. The total crew provisions mass and volume for the entire scenario are displayed in the lower right corner of the “Crew Provisions Demand Model” panel.

The input parameters can be modified by the user and used to calculate different crew provisions masses and volumes. These scenario specific parameters can be saved to be used later in the simulation by pushing the “Save Parameters” button. These parameters can be restored to their default values by pushing the “Restore Default Values” button.

Crew Operations (COS 3)

Push “Crew Operations” to display the “Crew Operations Demand Model” panel in Figure 35. See Appendix C for a detailed description of the model.



Scenario Name: Apollo_17

Crew Operations Demand Model

Inputs

	[kg]	[m^3]
Office Equipment / Crew	5	).02619
EVA Equipment and Consumables		
EVA Suit / Crew	107	0.582
EVA LiOH / Crew / EVA	2.9	0.009
EVA O2 / Crew / EVA	0.63	l3e-007
Health Equipment and Consumables		
CHeCS Equipment	20	0.0865
CHeCS Consumables / Crew / Day	0.1	004325
Safety Equipment	25	0.1
Communications Equipment	20	).07717
Computer Equipment / Crew	5	).01243

Crew Operations [kg] 335

Crew Operations [m^3] 1.618

Scenario Demands

Demand by Processes		Demand by Scenario	
Propellants and Fuels [kg]	0	Propellants and Fuels [kg]	0
Crew Provisions [kg]	66.9	Crew Provisions [kg]	332.7
Crew Operations [kg]	235.8	Crew Operations [kg]	335
Maintenance and Upkeep [kg]	0	Maintenance and Upkeep [kg]	0
Exploration and Research [kg]	456	Exploration and Research [kg]	456
Waste and Disposal [kg]	1.98	Waste and Disposal [kg]	15.84
Stowage and Restraint [kg]	34.22	Stowage and Restraint [kg]	84.73
Habitation and Infrastructure [kg]	0	Habitation and Infrastructure [kg]	0
Miscellaneous [kg]	0	Miscellaneous [kg]	0
Total [kg]	794.9	**Total [kg]**	**1224**

☐ Override Scenario Demands
☐ Override Process Demands

Figure 35: Demand – Crew Provisions

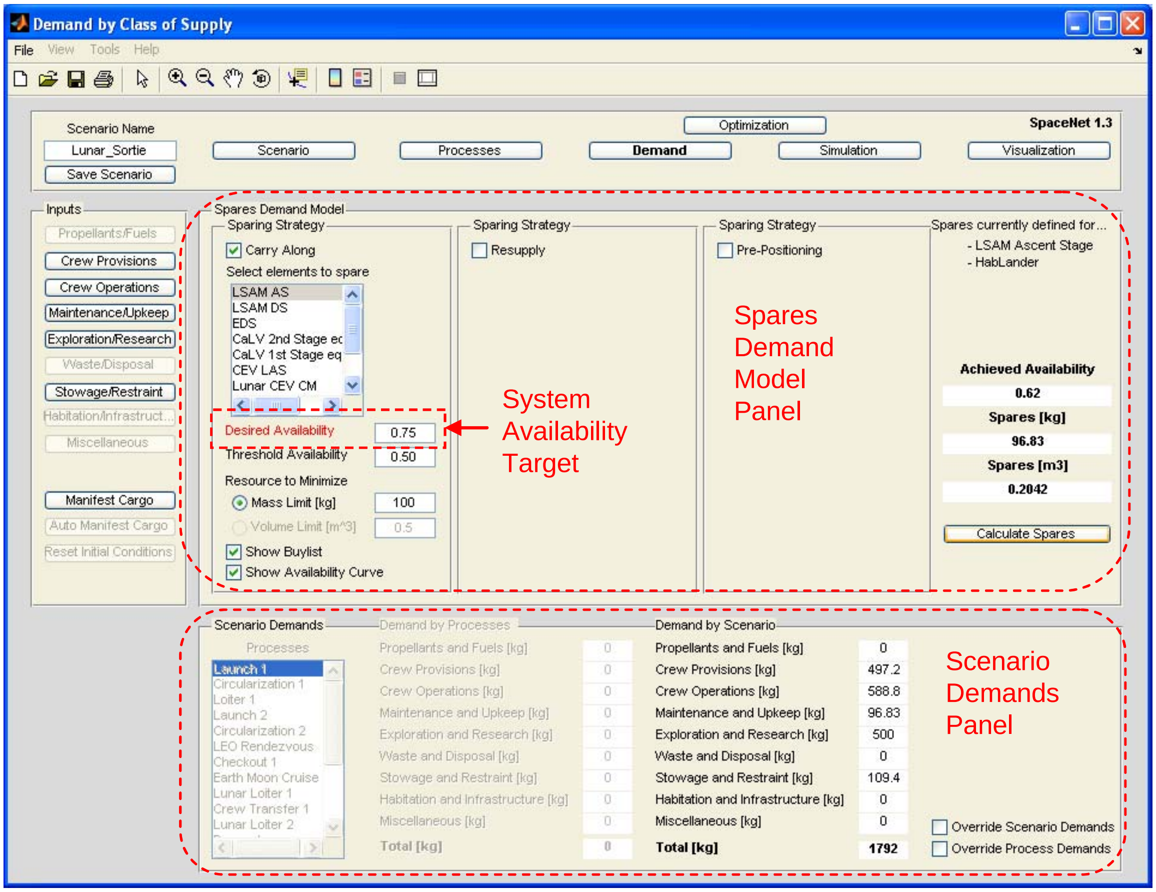
The model takes as input the process parameters displayed in the “Create Processes” panel (crew size, durations, number of EVAs, etc.) and the input parameters displayed in the “Inputs” panel within the “Crew Operations Demand Model” panel.

Push the “Calculate Crew Operations” button to call the demand model. The total crew operations mass and volume for the scenario are displayed in the lower right corner of the “Crew Operations Demand Model” panel.

The input parameters can be modified by the user and used to calculate different crew operations masses and volumes. These scenario specific parameters can be saved to be used later in the simulation by pushing the “Save Parameters” button. These parameters can be restored to their default values by pushing the “Restore Default Values” button.

Maintenance/Upkeep (COS 4)

This category focuses on spares and maintenance tools and equipment to ensure high system availability of the elements during a scenario. Push the “Maintenance/Upkeep” button to display the “Spares Demand Model” panel shown in Figure 36. See Appendix C for a detailed description of the model.



Scenario Name: Lunar_Sortie

SpaceNet 1.3

Spares Demand Model

Sparing Strategy

☑ Carry Along

Select elements to spare

LSAM AS
LSAM DS
EDS
CaLV 2nd Stage ec
CaLV 1st Stage eq
CEV LAS
Lunar CEV CM

Desired Availability 0.75

Threshold Availability 0.50

Resource to Minimize

● Mass Limit [kg] 100

○ Volume Limit [m^3] 0.5

☑ Show Buylist

☑ Show Availability Curve

Sparing Strategy

☐ Resupply

Sparing Strategy

☐ Pre-Positioning

Spares currently defined for...
- LSAM Ascent Stage
- HabLander

Achieved Availability 0.62

Spares [kg] 96.83

Spares [m3] 0.2042

System Availability Target

Spares Demand Model Panel

Scenario Demands

Processes

Launch 1
Circularization 1
Loiter 1
Launch 2
Circularization 2
LEO Rendezvous
Checkout 1
Earth Moon Cruise
Lunar Loiter 1
Crew Transfer 1
Lunar Loiter 2

Demand by Processes		Demand by Scenario	
Propellants and Fuels [kg]	0	Propellants and Fuels [kg]	0
Crew Provisions [kg]	0	Crew Provisions [kg]	497.2
Crew Operations [kg]	0	Crew Operations [kg]	588.8
Maintenance and Upkeep [kg]	0	Maintenance and Upkeep [kg]	96.83
Exploration and Research [kg]	0	Exploration and Research [kg]	500
Waste and Disposal [kg]	0	Waste and Disposal [kg]	0
Stowage and Restraint [kg]	0	Stowage and Restraint [kg]	109.4
Habitation and Infrastructure [kg]	0	Habitation and Infrastructure [kg]	0
Miscellaneous [kg]	0	Miscellaneous [kg]	0
Total [kg]	0	**Total [kg]**	1792

Scenario Demands Panel

☐ Override Scenario Demands

☐ Override Process Demands

Figure 36: Demand – Maintenance/Upkeep

The model takes as input the process parameters displayed in the “Create Processes” panel (crew size, durations, elements, etc.) and data from the integrated database. The database currently contains detailed spares data (orbital replacement units = ORU) only for the *LSAM Ascent Stage and the Habitat Lander* elements. Additional ORUs will be added in future releases.

The user has the option to select from three sparing strategies: Carry Along, Resupply, and/or Pre-Positioning. These strategies can be selected individually or in combination. Carry Along is the only strategy currently modeled in SpaceNet v1.3. Check the *Carry Along* checkbox to display the list of instantiated elements in the scenario. The user can select one or multiple instantiated elements to carry spares.

The model allows the user to specify a desired availability, threshold availability, which resource to limit (mass or volume), and its upper bound. The threshold availability is the minimum availability allowed, and the model will allocate spares as necessary to satisfy it. After the threshold availability is met, the model will try to satisfy the desired availability (system availability target) while staying within the resource constraint. Only mass is implemented as an allowed resource constraint in SpaceNet 1.3.

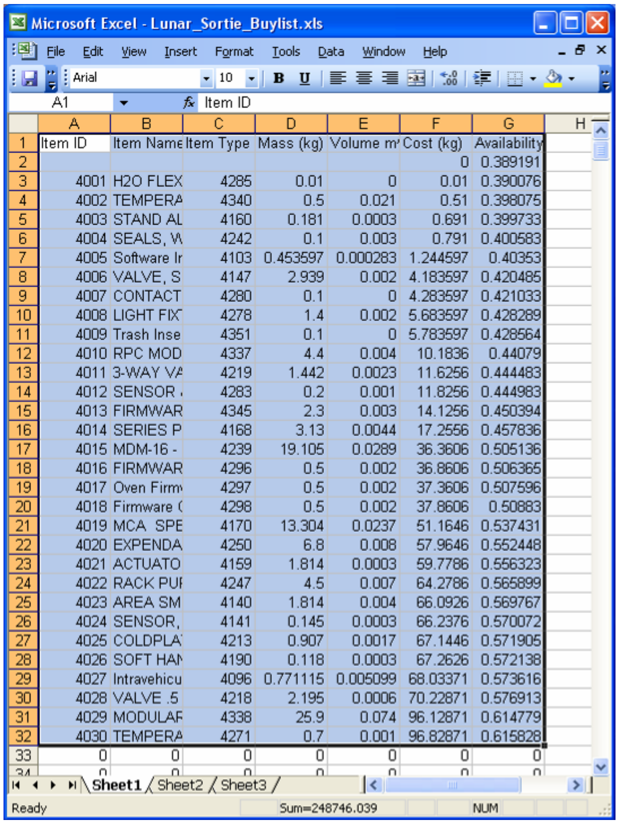
Push the "Calculate Spares" button to call the demand model. The spares model selects spare parts (Orbital Replacement Units (ORUs)) in order to meet a desired (target) availability[8], while staying within a resource (mass or volume) limit. The achieved availability and total spares mass and volume for the scenario are displayed in the lower right corner of the "Spares Demand Model" panel and next to the "Spares" text in the "Demand by Scenario" panel.

The outputs from the model are the achieved availability, mass, and volume of the selected spares. If the "Show Buylist" box is checked, an Excel file will be created and opened showing the spares chosen and their attributes (see Figure 37). If the "Show Availability Curve" box is checked, a plot will open showing the availability to resource curve for the system (see Figure 38).

In Figure 36, the final achieved availability is 0.62, meeting the threshold availability of 0.5, but not the Desired Availability of 0.75 (which is thus colored red). The total spares mass is 96.8 kg which is under the resource limit of 100 kg. The model was not able to meet the desired availability due to the mass constraint. The total spares volume is 0.204 m^3.

---

[8] Availability is defined roughly speaking as the probability of no backorders over the effective length of the scenario (mission). See Appendix C for the precise calculations and assumptions.



	A	B	C	D	E	F	G
1	Item ID	Item Name	Item Type	Mass (kg)	Volume m	Cost (kg)	Availability
2						0	0.389191
3	4001	H2O FLEX	4285	0.01	0	0.01	0.390076
4	4002	TEMPERA	4340	0.5	0.021	0.51	0.398075
5	4003	STAND AL	4160	0.181	0.0003	0.691	0.399733
6	4004	SEALS, W	4242	0.1	0.003	0.791	0.400583
7	4005	Software Ir	4103	0.453597	0.000283	1.244597	0.40353
8	4006	VALVE, S	4147	2.939	0.002	4.183597	0.420485
9	4007	CONTACT	4280	0.1	0	4.283597	0.421033
10	4008	LIGHT FIX	4278	1.4	0.002	5.683597	0.428289
11	4009	Trash Inse	4351	0.1	0	5.783597	0.428564
12	4010	RPC MOD	4337	4.4	0.004	10.1836	0.44079
13	4011	3-WAY VA	4219	1.442	0.0023	11.6256	0.444483
14	4012	SENSOR	4283	0.2	0.001	11.8256	0.444983
15	4013	FIRMWAR	4345	2.3	0.003	14.1256	0.450394
16	4014	SERIES P	4168	3.13	0.0044	17.2556	0.457836
17	4015	MDM-16 -	4239	19.105	0.0289	36.3606	0.505136
18	4016	FIRMWAR	4296	0.5	0.002	36.8606	0.506365
19	4017	Oven Firm	4297	0.5	0.002	37.3606	0.507596
20	4018	Firmware (	4298	0.5	0.002	37.8606	0.50883
21	4019	MCA SPE	4170	13.304	0.0237	51.1646	0.537431
22	4020	EXPENDA	4250	6.8	0.008	57.9646	0.552448
23	4021	ACTUATO	4159	1.814	0.0003	59.7786	0.556323
24	4022	RACK PUI	4247	4.5	0.007	64.2786	0.565899
25	4023	AREA SM	4140	1.814	0.004	66.0926	0.569767
26	4024	SENSOR,	4141	0.145	0.0003	66.2376	0.570072
27	4025	COLDPLA	4213	0.907	0.0017	67.1446	0.571905
28	4026	SOFT HAN	4190	0.118	0.0003	67.2626	0.572138
29	4027	Intravehicu	4096	0.771115	0.005099	68.03371	0.573616
30	4028	VALVE .5	4218	2.195	0.0006	70.22871	0.576913
31	4029	MODULAF	4338	25.9	0.074	96.12871	0.614779
32	4030	TEMPERA	4271	0.7	0.001	96.82871	0.615828
33	0	0	0	0	0	0	0

Figure 37: Example Spares Buylist

The buylist shows the selected spares. Each spare has an item ID (used internally for item manifestation), the name of the spare, the item type ID of the spare (as listed in the integrated database), and the part mass and volume. The availability curve shown below tells the user how each spare increases cost and availability of the system as it is added.

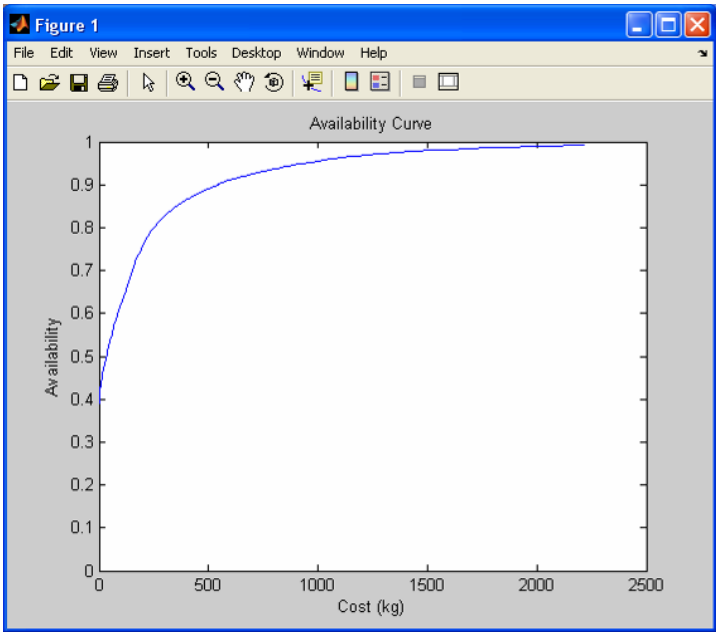
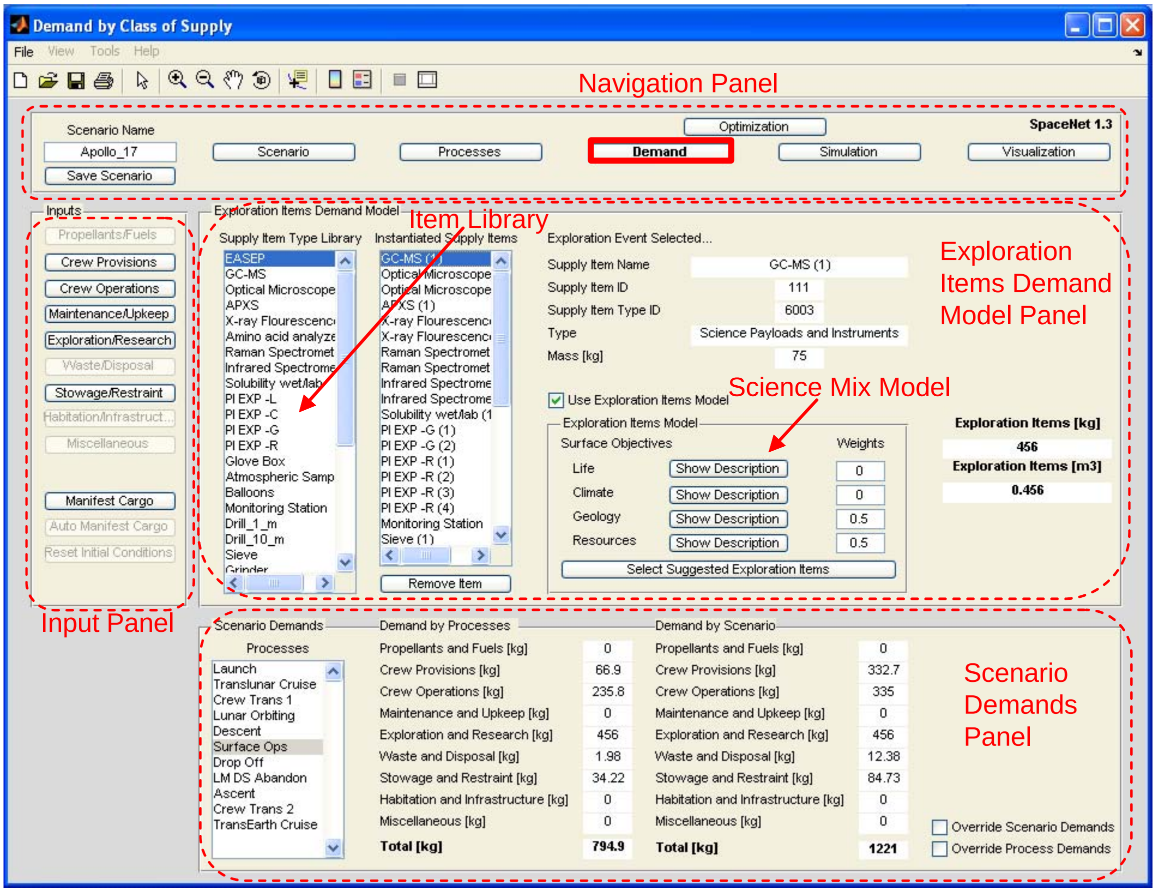


Figure 38: Availability Curve

The availability curve plots the cumulative availability and cost. The curve is calculated out to an availability of 0.99, so it may go farther than the availability given by the model or the one shown in the buylist.

Exploration Items (COS 6)

Push the "Exploration/Research" button and select an exploration process in the "Processes" listbox to display the "Exploration Items Demand Model" panel shown in Figure 39. Exploration items are different from the other demand models because they are specific for each exploration process. The user can substantially influence the type and quantity of exploration items taken along, while demands for crew consumables are more or less dictated by the number of crew and the duration of the scenario. In Figure 39 below, the "Surface Ops" process from Figure 26 has been selected. See Appendix C for a detailed description of the exploration items model.



Scenario Name: Apollo_17

Exploration Event Selected...

Supply Item Name	GC-MS (1)
Supply Item ID	111
Supply Item Type ID	6003
Type	Science Payloads and Instruments
Mass [kg]	75

☑ Use Exploration Items Model

Exploration Items Model

Surface Objectives	Weights
Life	0
Climate	0
Geology	0.5
Resources	0.5

Exploration Items [kg] 456

Exploration Items [m3] 0.456

Demand by Processes		Demand by Scenario	
Propellants and Fuels [kg]	0	Propellants and Fuels [kg]	0
Crew Provisions [kg]	66.9	Crew Provisions [kg]	332.7
Crew Operations [kg]	235.8	Crew Operations [kg]	335
Maintenance and Upkeep [kg]	0	Maintenance and Upkeep [kg]	0
Exploration and Research [kg]	456	Exploration and Research [kg]	456
Waste and Disposal [kg]	1.98	Waste and Disposal [kg]	12.38
Stowage and Restraint [kg]	34.22	Stowage and Restraint [kg]	84.73
Habitation and Infrastructure [kg]	0	Habitation and Infrastructure [kg]	0
Miscellaneous [kg]	0	Miscellaneous [kg]	0
Total [kg]	**794.9**	**Total [kg]**	**1221**

☐ Override Scenario Demands
☐ Override Process Demands

Figure 39: Demand – Exploration/Research

There are two ways to calculate the exploration items demand for an exploration process:

- User-selected exploration items (similar to instantiating elements) from library
- Exploration Items Model ("Science Mix Model")

Exploration items from the integrated database are loaded into the “Supply Item Type Library” listbox. Double-clicking on a supply item will create an instantiation of the item and add it to the “Instantiated Supply Items” listbox. The user can use shift-clicking to instantiate multiple supply items at once. Supply items can also be instantiated by right-clicking a selected supply item then selecting “Instantiate Supply Items”. Multiple supply items can be selected at once and instantiated in this manner. Multiple instantiations of the same supply item will append a (1), (2), etc… to its name. The user can push the “Remove Item” button to a remove a supply item that has been accidentally instantiated.

Single-clicking on supply items in the “Supply Item Type Library” and “Instantiated Supply Items” listboxes will display its attributes on the right of the “Exploration Items Demand Model” panel. SpaceNet will create an additional unique “Supply Item ID” for each instantiated supply item that will arbitrarily begin at 111. Supply items are not individually tracked like elements are in SpaceNet 1.3 (this might be added in future releases).

The attributes that are visible for every supply item are:

- *Supply Item Name*: The supply item name
- *Supply Item ID*: The supply item instantiation ID from SpaceNet (unique numbers)
- *Supply Item Type ID*: The supply item type ID from the integrated database (non-unique numbers)
- *Type*: The supply item class or sub class
- *Mass*: The supply item mass

Each instantiated supply item’s mass is added to the total exploration items mass displayed in the lower right corner of the “Exploration Items Demand Model” panel and next to the “Exploration Items” text in the “Demand by Scenario” panel.

Check the “Use Exploration Items Model” checkbox to enable the exploration items demand model. Push the “Show Description” button to popup a description of each of the science objective weights. The weights should sum to 1 as they reflect the relative mix of science objectives that are to be achieved during an exploration process. A suggested list of exploration items can be automatically generated by entering a weighting factor for the following four baseline planetary science programs:

- Life (look for biological agents, fossils, carbon-based molecules…)
- Climate (analyze atmospheric properties, weather phenomena…)
- Geology (obtain rock and soil samples, take images, analyze strata…)
- Resources (look for useful minerals, ISRU feed stock, water/ice,…)

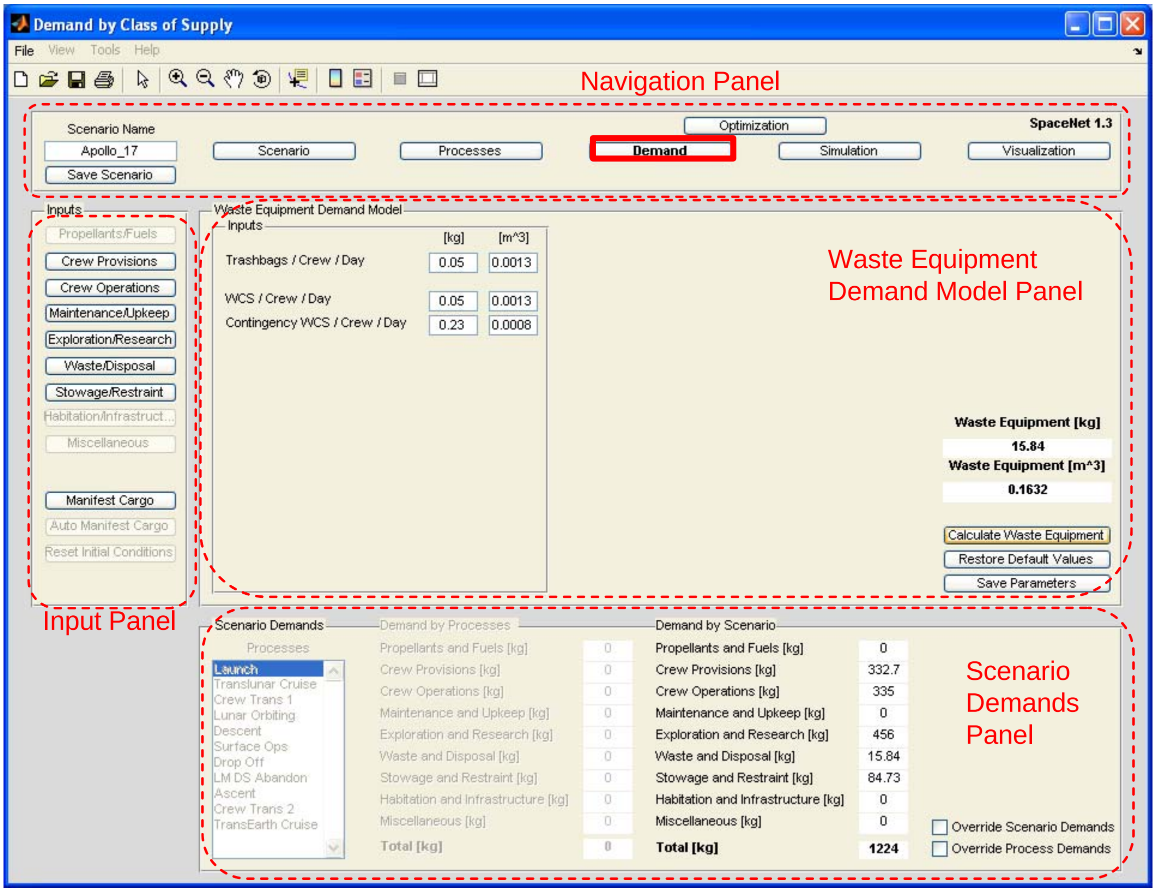
Pressing the “Show Description” button next to each science objectives gives additional details about the underlying assumptions. Push “Select Suggested Exploration Items” to instantiate a number of supply items based on the process parameters and science objective weights. The total exploration items mass and volume are displayed in their respective locations.

The user can remove and add exploration items to the list of instantiated supply items the demand model selects by pushing the “Remove Item” button and instantiating supply items.

The user can use either or both of these methods to calculate the exploration items demand for each exploration process, which are then summed for the total scenario exploration demand.

Waste Equipment (COS 7)

Push “Waste/Disposal” to display the “Waste Equipment Demand Model” panel shown in Figure 40. See Appendix C for a detailed description of the model.



Scenario Name: Apollo_17

Waste Equipment Demand Model

Inputs

	[kg]	[m^3]
Trashbags / Crew / Day	0.05	0.0013
WCS / Crew / Day	0.05	0.0013
Contingency WCS / Crew / Day	0.23	0.0008

Waste Equipment [kg] 15.84

Waste Equipment [m^3] 0.1632

Scenario Demands

Demand by Processes		Demand by Scenario	
Propellants and Fuels [kg]	0	Propellants and Fuels [kg]	0
Crew Provisions [kg]	0	Crew Provisions [kg]	332.7
Crew Operations [kg]	0	Crew Operations [kg]	335
Maintenance and Upkeep [kg]	0	Maintenance and Upkeep [kg]	0
Exploration and Research [kg]	0	Exploration and Research [kg]	456
Waste and Disposal [kg]	0	Waste and Disposal [kg]	15.84
Stowage and Restraint [kg]	0	Stowage and Restraint [kg]	84.73
Habitation and Infrastructure [kg]	0	Habitation and Infrastructure [kg]	0
Miscellaneous [kg]	0	Miscellaneous [kg]	0
Total [kg]	**0**	**Total [kg]**	**1224**

☐ Override Scenario Demands
☐ Override Process Demands

Figure 40: Demand – Waste

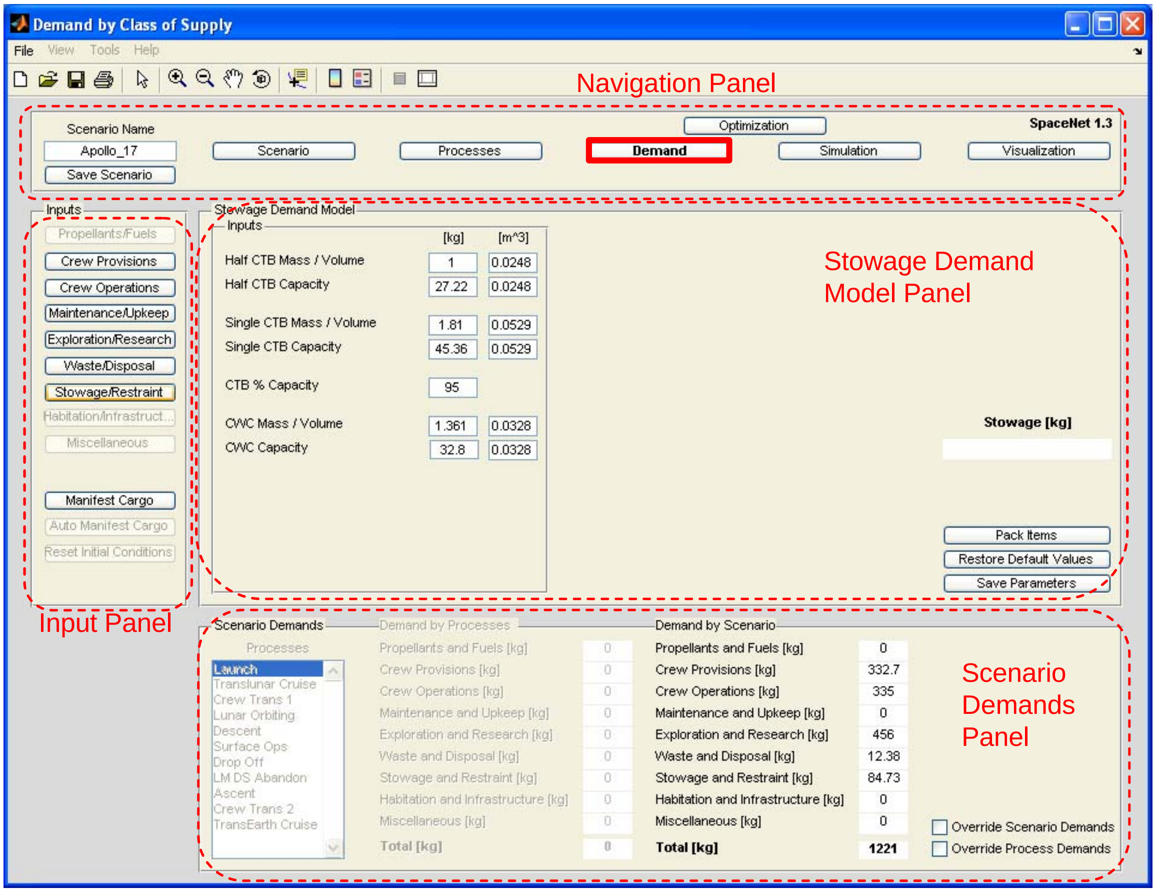
The model takes as input the process parameters displayed in the “Create Processes” panel (crew size, durations, etc.) and the input parameters displayed in the “Inputs” panel within the “Waste Equipment Demand Model” panel.

Push the “Calculate Waste Equipment” button to call the demand model. The total waste equipment mass and volume for the scenario is displayed in the lower right corner of the “Waste Equipment Demand Model” panel and next to the “Waste Equipment” text in the “Demand by Scenario” panel.

The input parameters can be modified by the user and used to calculate different waste equipment masses. These parameters can be saved to be used later in the simulation by pushing the “Save Parameters” button. These parameters can also be restored to their default values by pushing the “Restore Default Values” button.

Stowage (COS 5)

Push “Stowage/Restraint” to display the “Stowage Demand Model” panel shown in Figure 41. See Appendix C for a detailed description of the model.



Scenario Name: Apollo_17

Stowage Demand Model

Inputs

	[kg]	[m^3]
Half CTB Mass / Volume	1	0.0248
Half CTB Capacity	27.22	0.0248
Single CTB Mass / Volume	1.81	0.0529
Single CTB Capacity	45.36	0.0529
CTB % Capacity	95	
CWC Mass / Volume	1.361	0.0328
CWC Capacity	32.8	0.0328

Scenario Demands

Processes: Launch, Translunar Cruise, Crew Trans 1, Lunar Orbiting, Descent, Surface Ops, Drop Off, LM DS Abandon, Ascent, Crew Trans 2, TransEarth Cruise

Demand by Processes		Demand by Scenario	
Propellants and Fuels [kg]	0	Propellants and Fuels [kg]	0
Crew Provisions [kg]	0	Crew Provisions [kg]	332.7
Crew Operations [kg]	0	Crew Operations [kg]	335
Maintenance and Upkeep [kg]	0	Maintenance and Upkeep [kg]	0
Exploration and Research [kg]	0	Exploration and Research [kg]	456
Waste and Disposal [kg]	0	Waste and Disposal [kg]	12.38
Stowage and Restraint [kg]	0	Stowage and Restraint [kg]	84.73
Habitation and Infrastructure [kg]	0	Habitation and Infrastructure [kg]	0
Miscellaneous [kg]	0	Miscellaneous [kg]	0
Total [kg]	**0**	**Total [kg]**	**1221**

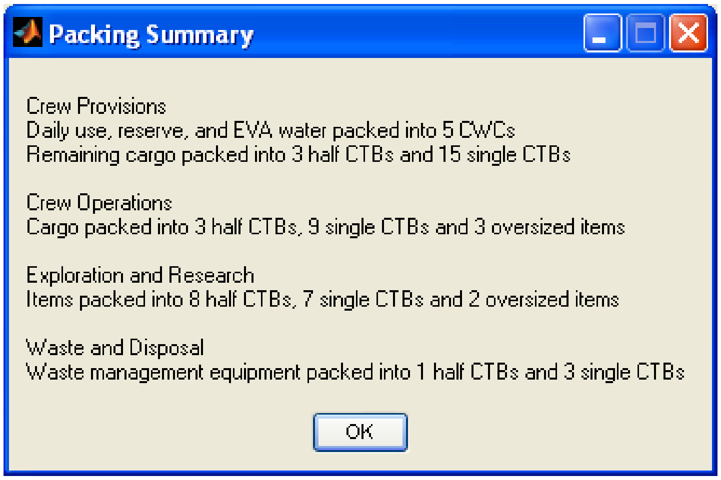
☐ Override Scenario Demands
☐ Override Process Demands

Figure 41: Demand – Stowage

The model takes as input the mass and volume of the demands calculated by the other classes of supply demand models and packs these items into either half CTBs (cargo transfer bags), single CTBs or keeps them as stand-alone items. Water is packed into contingency water containers (CWCs).

The following input parameters can be modified by the user and used to calculate different packing configurations. Most of the input parameters are self-explanatory. *CTB % Capacity* refers to the percentage (by mass and volume) full the software will pack a CTB before starting a new CTB. This can also be thought of as the packing efficiency. Any modified parameters can be saved to be used later in the simulation by pushing the "Save Parameters" button. These parameters can also be restored to their default values by pushing the "Restore Default Values" button.

Push the "Pack Items" button to run the packing model. This takes both the discrete supply items for which demand was generated (e.g. exploration items) and the continuous demand (e.g. for crew consumables) and packs them into the above-mentioned bags and containers for subsequent manifesting. When the packing algorithm has finished running the packing summary window will pop-up (see Figure 42 below). This window summarizes the results of the packing by class of supply. The packed items are saved in the scenario demand data structure.



Crew Provisions
Daily use, reserve, and EVA water packed into 5 CWCs
Remaining cargo packed into 3 half CTBs and 15 single CTBs

Crew Operations
Cargo packed into 3 half CTBs, 9 single CTBs and 3 oversized items

Exploration and Research
Items packed into 8 half CTBs, 7 single CTBs and 2 oversized items

Waste and Disposal
Waste management equipment packed into 1 half CTBs and 3 single CTBs

Figure 42: Packing Summary

Push the "Save Scenario" button to save all scenario demands and demand parameters to file.

Manifest Cargo

Push "Manifest Cargo" to display the "Manifest Cargo" panel on the canvas. Figure 43 shows the Manifest Cargo Panel. This screen is used to assign individuals bags and containers of supply items to elements with non-zero cargo capacity at the time of their first launch.

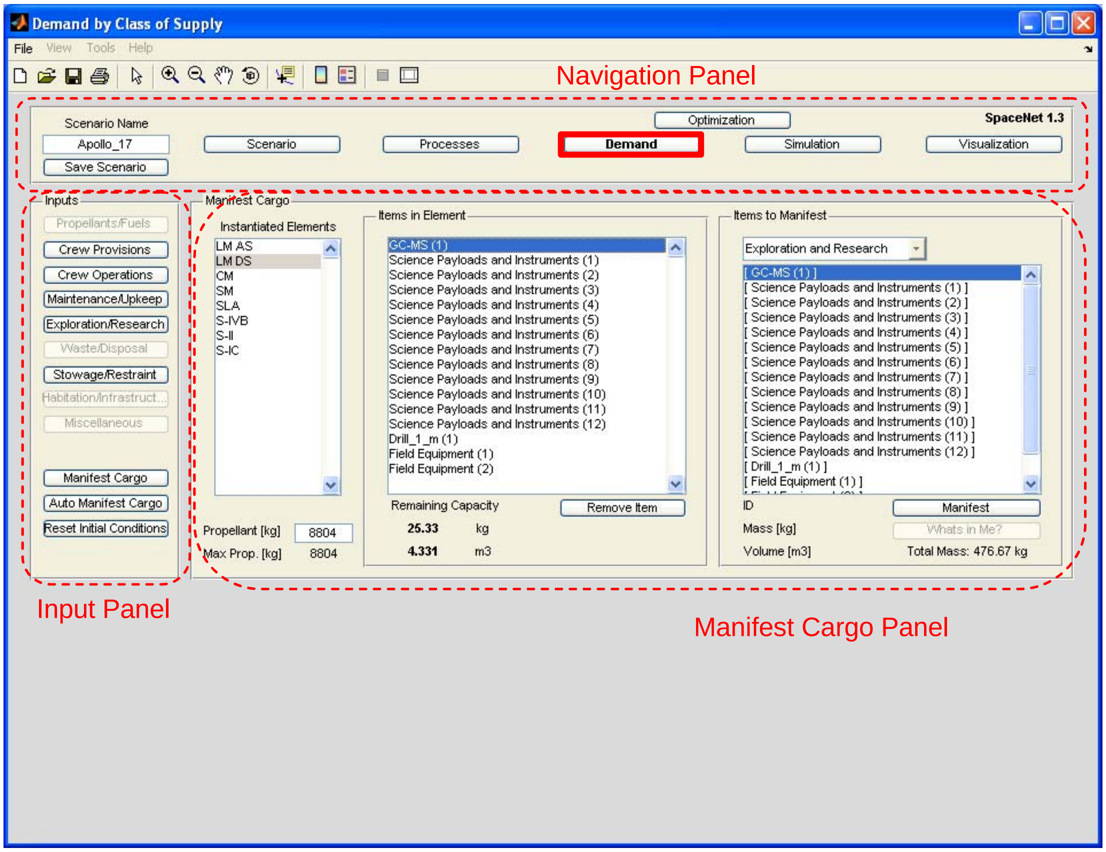


Figure 43: Manifest Cargo

The "Instantiated Elements" listbox on the left of the panel displays the instantiated elements in the scenario. Selecting an element in the listbox displays its contents (at the bag/container level) underneath the "Items in Element" section and its remaining capacity. Propellants are automatically set to the maximum capacity for each element. In order to only partially fill an element with propellant this can be overwritten.

The values on the right of the panel under the heading "Items to Manifest" display the list of bags and individual oversized items calculated by the stowage model by category. Highlighting an item and pushing "Manifest" will add that item to the element selected. If the capacity of an element has been exceeded (by mass or volume) or the item selected has already been manifested, an error message will pop up. Items that have already been manifested in the "Items to Manifest" listbox have **[square brackets]** around them.

The users can select each element and specify its initial conditions or cargo manifested at launch by class of supply until the total scenario demands have been satisfied. Pushing "Reset Initial Conditions" will clear all manifested cargo and allow the user to start the manifesting process over.

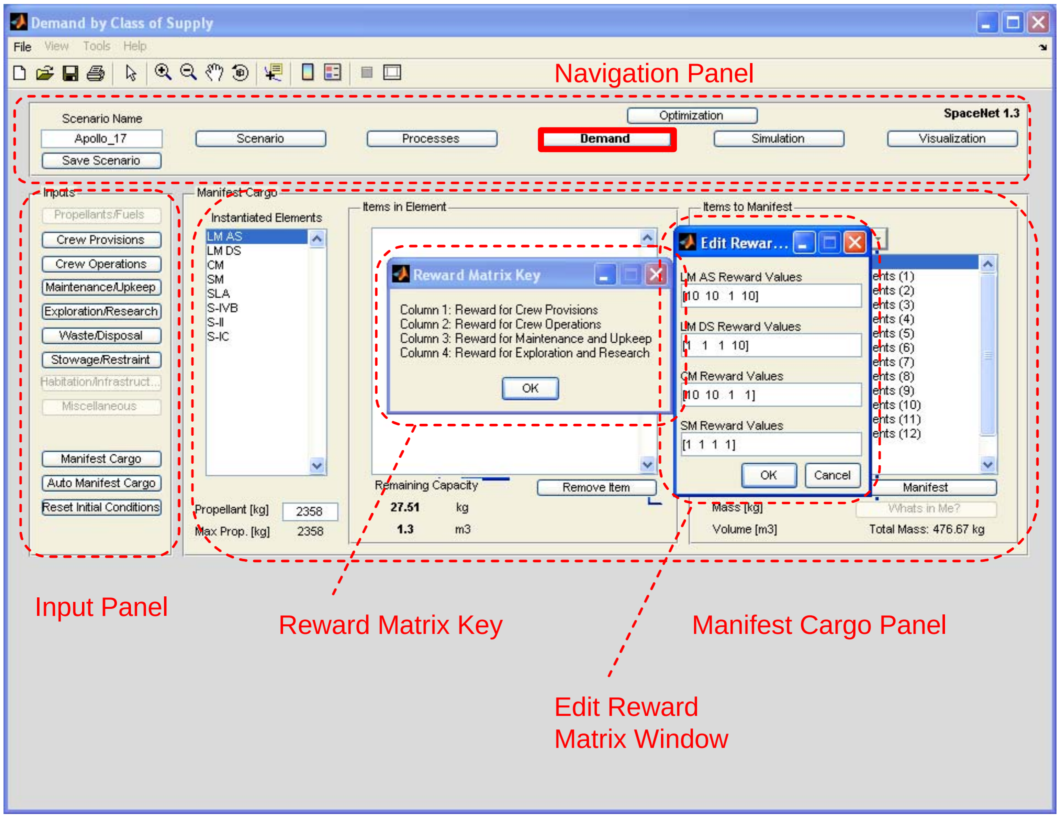
Auto-Manifest Cargo

For complex scenarios containing many instantiated elements and many supply items to-be-manifested generating a feasible manifest that will satisfy all capacity limitations and does not require many in-space transfers can be quite tedious to obtain. In order to facilitate this task an auto-manifesting optimization algorithm was embedded in SpaceNet 1.3. This algorithm automatically assigns bags/containers containing supply items to instantiated elements before launch. The objective function is to maximize the "*reward*" of the scenario manifest. The reward matrix allows the user to prioritize some supply classes over others in particular elements. For example, it is most important that those elements carrying crew always have sufficient crew provisions.

The auto-manifest cargo feature calls an optimization script to auto-manifest the items into the elements. See Appendix F for a detailed description of the auto-manifesting optimization algorithm. Pushing the "Auto-Manifest Cargo" button will first reset the initial conditions and then allow the user to input the following parameters:

- *Maximum optimization gap [%]*: the percentage the solution will deviate from the globally optimal solution
- *Maximum run time [s]*: the maximum run time the optimization script will run

It can happen that some items are left behind and must be manifested manually or that the auto-generated manifest must be manually "repaired". Nevertheless, this feature alleviates the workload significantly.



Scenario Name: Apollo_17

Propellant [kg] 2358

Max Prop. [kg] 2358

Remaining Capacity: 27.51 kg, 1.3 m3

Column 1: Reward for Crew Provisions
Column 2: Reward for Crew Operations
Column 3: Reward for Maintenance and Upkeep
Column 4: Reward for Exploration and Research

LM AS Reward Values: [10 10 1 10]

LM DS Reward Values: [1 1 1 10]

CM Reward Values: [10 10 1 1]

SM Reward Values: [1 1 1 1]

Total Mass: 476.67 kg

Figure 44: Auto-Manifest Cargo

The user will then have the option to edit the reward matrix. The reward matrix specifies the priority of manifesting a supply class to a specific element. The rows of the reward matrix correspond to the number of elements that have non-zero cargo capacities. The columns correspond to the reward for the four classes of supply that currently have demand models. The meaning of the reward matrix columns is shown in the Reward Matrix Key popup window. The entries in the matrix can be any non-negative number. In the figure above, the LM DS is only supposed to carry exploration and research items, this is why column 4 has a reward value of 10, and the other columns have a reward value of only 1. A message will pop up indicating whether auto-manifesting was successful or if it failed. In a failed case, a list of the items that could not be manifested will be displayed.

Push "Save Scenario" to save the initial conditions (manifest before launch) to file.

Network Optimization

The Process step described earlier required the user to manually define the transportation architecture by assigning instantiated elements to transportation processes. This required a substantial amount of effort. Only once the transportation architecture was defined did we generate demand for all processes and define transfers and initial manifests. In a way this is a "forward" mode of scenario definition, first defining the transportation architecture, then the crew and supply items to be carried in the elements.

The optimization capability in SpaceNet 1.3 attempts to enable a "backward" mode of scenario definition, where only the fleet of available elements and the set of exploration processes (at what nodes? how long?) is defined apriori and the optimizer acts as an automated planner to instantiate individual flights and transportation processes to fulfill the demand generated by exploration processes[9].

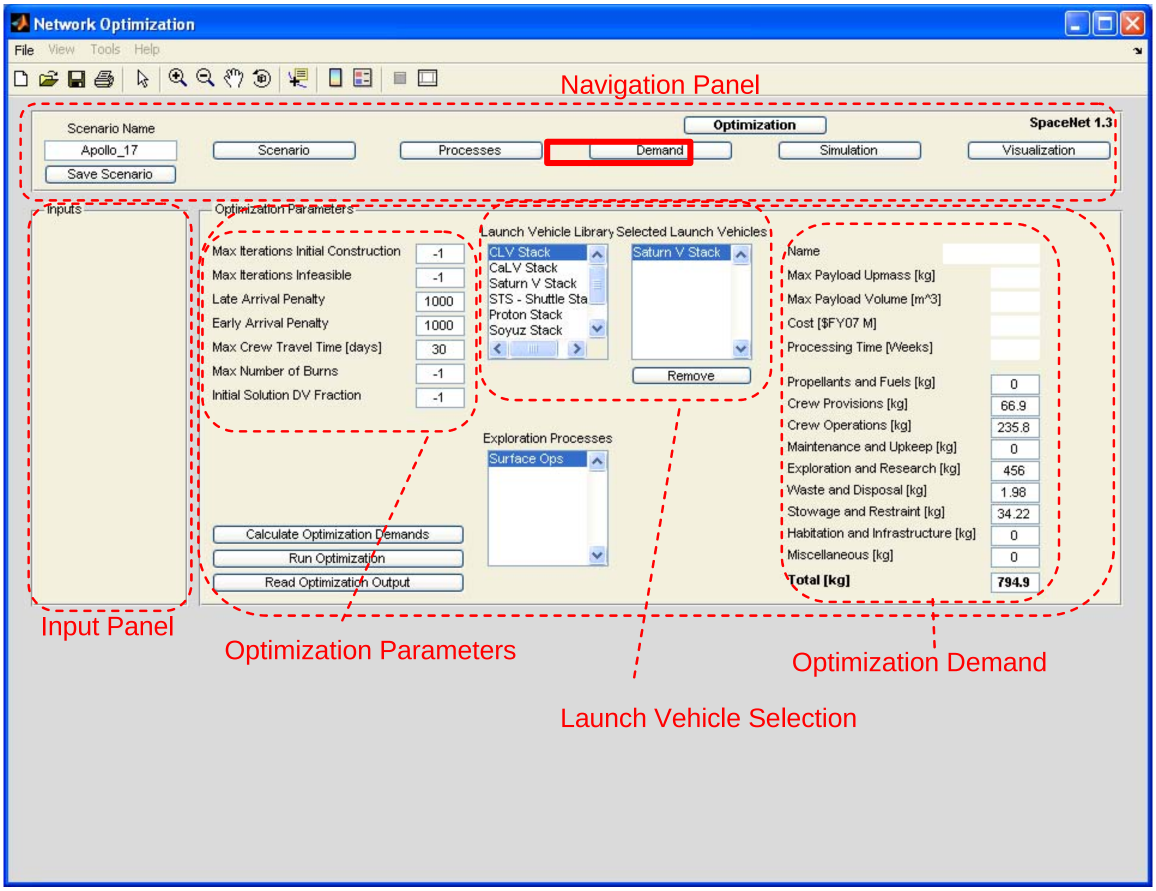
Push the "Optimization" button to display the "Optimization Parameters" panel on the canvas as shown in Figure 45. In order to be able to execute optimization, a user must have the CPLEX mixed-integer solver package installed on his or her local computer.[10]

In "Network Optimization", optimization code in CPLEX takes the scenario information as input and calculates the optimal element shipment paths and cargo manifests for each element. The scenario information has to be translated from MATLAB into text files for CPLEX, and then the output has to be translated from text files back into MATLAB for the "Simulation" and "Visualization" steps (see below) to be enabled.

---

[9] Note: The optimizer internally uses a set of simplified assumptions and constraints and often returns answers that are interesting, but sometimes counterintuitive for users. Also, there is no guarantee that a transportation architecture generated by optimization will simulate without errors. Users must sometimes be prepared to manually adjust and "repair" optimization-generated scenarios.

[10] CPLEX is a registered trademark of ILOG Inc.: http://www.ilog.com



Scenario Name: Apollo_17

Optimization Parameters	
Max Iterations Initial Construction	-1
Max Iterations Infeasible	-1
Late Arrival Penalty	1000
Early Arrival Penalty	1000
Max Crew Travel Time [days]	30
Max Number of Burns	-1
Initial Solution DV Fraction	-1

Launch Vehicle Library: CLV Stack, CaLV Stack, Saturn V Stack, STS - Shuttle Sta, Proton Stack, Soyuz Stack

Selected Launch Vehicles: Saturn V Stack

Exploration Processes: Surface Ops

Name	
Max Payload Upmass [kg]	
Max Payload Volume [m^3]	
Cost [$FY07 M]	
Processing Time [Weeks]	
Propellants and Fuels [kg]	0
Crew Provisions [kg]	66.9
Crew Operations [kg]	235.8
Maintenance and Upkeep [kg]	0
Exploration and Research [kg]	456
Waste and Disposal [kg]	1.98
Stowage and Restraint [kg]	34.22
Habitation and Infrastructure [kg]	0
Miscellaneous [kg]	0
Total [kg]	**794.9**

Figure 45: Network Optimization

The user can select launch vehicles to be used in the optimization from the “Launch Vehicle Library” selection listbox. Double-clicking on a selected vehicle will added it to the “Selected Launch Vehicles” listbox. Upon selection, the launch vehicle’s attributes are displayed on the right of the “Optimization Parameters” panel.

The launch vehicle attributes displayed are:

- *Name*: The launch vehicle name
- *ID*: The launch vehicle instantiation ID from SpaceNet (unique numbers beginning at 1)
- *Type ID*: The launch vehicle type ID from the integrated database (non-unique numbers)
- *Max Payload Upmass*: Maximum payload mass the launch vehicle can transport from Earth to a reference LEO
- *Max Payload Volume*: Maximum payload volume the launch vehicle can transport from Earth to a reference LEO
- *Cost*: The launch vehicle recurring cost in $FY07M
- *Processing Time*: The launch vehicle processing time between subsequent launches
- *Spaceport ID*: The node ID of the launch vehicle origin
- *Stack Components*: The element type IDs that make up the launch vehicle stack

The optimization parameters are specified on the left of the "Optimization Parameters" panel.

- *Max Iterations Initial Construction*: The underlying algorithm is iterative and for each iteration it tries to find a solution. This parameter defines the maximum number of iterations that the algorithm runs. The larger the value, the more opportunities the algorithm has to find a good solution. If a better solution is expected from what is obtained, then this number should be increased.
- *Max Iterations Infeasible*: If a solution is not found within the number of iterations specified by the previous parameter, then this parameter defines the number of iterations to be performed. This parameter value should be higher than the previous one. If no solution is returned, then this number should be increased.
- *Late Arrival Penalty*: The penalty per day for each element if arriving late (same units as element cost) in terms of the delivery windows specified by exploration.
- *Early Arrival Penalty*: The penalty per day for each element if arriving early (same units as element cost) in terms of the delivery windows specified by exploration..
- *Max Crew Transit Time*: Currently not used in SpaceNet 1.3
- *Max Number of Burns:* The maximum number of consecutive burns. The default value is 2 and it should not be changed.
- *Initial Solution DV Factor:* A value between 0 and 1 that specifies the attractiveness of joint commodity (supply item) paths with respect to the cost of the elements. If commodities travel on several different paths that do not overlap, then increasing this value might find overlapping paths.

The user must first push the "Calculate Optimization Demands" button to select a subset of the demands for the scenario. Because CPLEX calculates the optimal shipment paths *only the demand input for exploration processes* is necessary as input. Pushing "Run Optimization" will create the necessary input text files and then launch the optimization executable file.

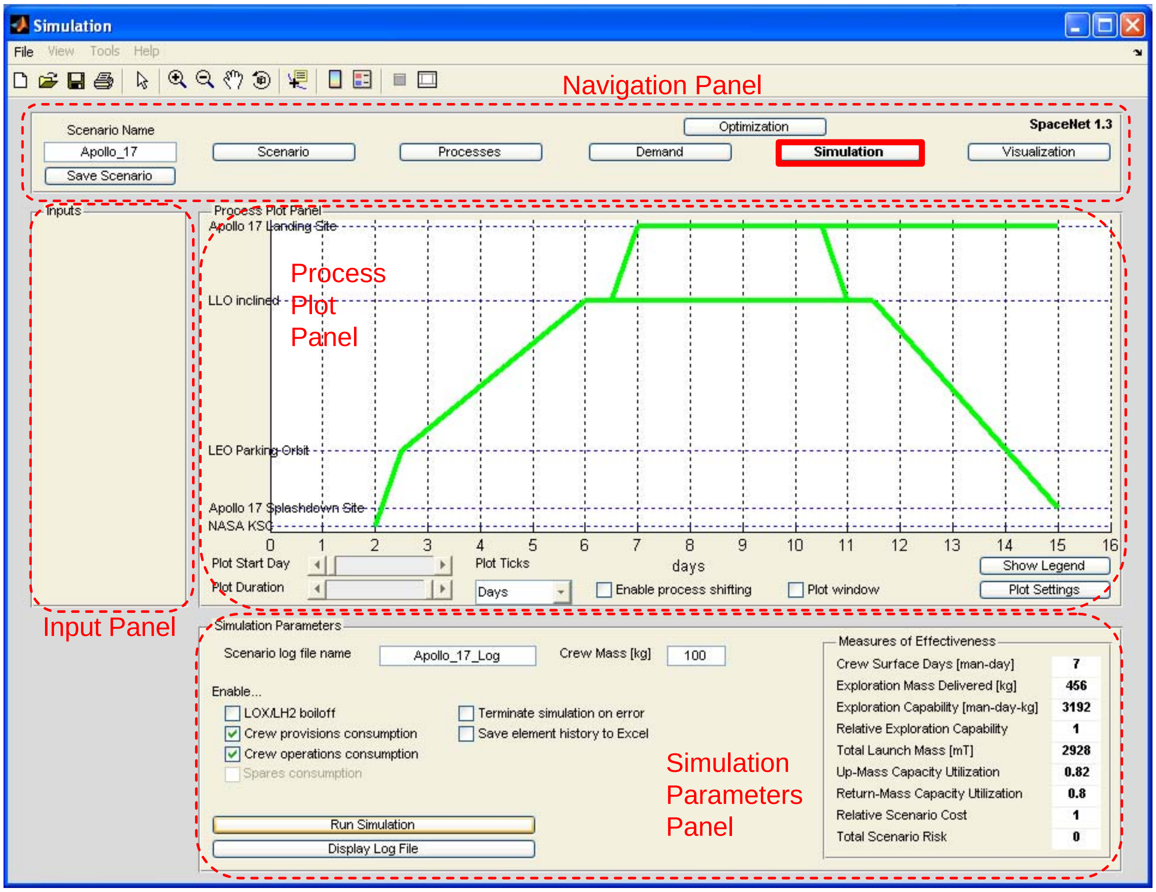
During optimization a red waitbar informs the user regarding progress ("Creating network …"). A popup window confirms successful completion of optimization. Click Ok.

Only computers with CPLEX can successfully run the optimization executable file, which will then create the output text files. Once the output text files are in place, push "Read Optimization Output" to translate the optimization output back into MATLAB. This will create a new scenario file with "_Opt" appended to it, SpaceNet will restart, and then the user will be able to load the optimized scenario when restarting SpaceNet.

Simulation

Push "Simulation" in the "Navigation" panel to display the "Simulation Parameters" panel on the canvas shown in Figure 46.

A blank white screen will appear at first (not yet showing green/red lines).



Scenario Name: Apollo_17

Plot Ticks: Days

☐ Enable process shifting

☐ Plot window

Scenario log file name: Apollo_17_Log

Crew Mass [kg]: 100

Enable...

☐ LOX/LH2 boiloff

☑ Crew provisions consumption

☑ Crew operations consumption

☐ Spares consumption

☐ Terminate simulation on error

☐ Save element history to Excel

Measures of Effectiveness	
Crew Surface Days [man-day]	7
Exploration Mass Delivered [kg]	456
Exploration Capability [man-day-kg]	3192
Relative Exploration Capability	1
Total Launch Mass [mT]	2928
Up-Mass Capacity Utilization	0.82
Return-Mass Capacity Utilization	0.8
Relative Scenario Cost	1
Total Scenario Risk	0

Figure 46: Simulation

The user has the following simulation options:

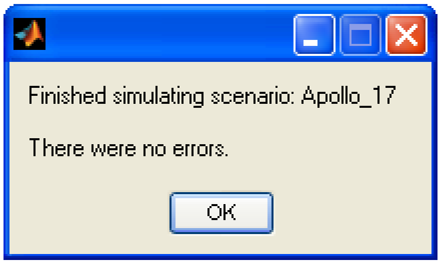
- *Scenario log file name*: The name of the text file the scenario results are saved to. The log file includes:
    - *Scenario Name*
    - *Generated On*: The date and time the scenario was first created
    - *Number of Instantiated Elements*
    - *Mission Start Time*: Scenario start date
    - *Mission End Time*: Scenario end date
    - *Number of Processes*
    - *Error Log*: Errors that cause scenario infeasibility
    - *Measures of Effectiveness (MOEs)*: See description of MOEs below
- *Crew Mass*: Assumed mass of crew in launch and entry suits; the default is 100 kg mass for each crew member
- *Enable Demand Models*: A check of the satisfaction of demand models is executed in the simulation. For example the consumption by crew members is computed for each time step, and the available inventory is decremented by the appropriate amount. The amount of waste generated is updated for each time step.

The simulation checks if any undersupply situation or backorders have occurred on a process-by-process basis.

- *Terminate simulation on error*: Terminate the simulation when an error occurs, if this is not checked the simulation will skip over errors but include them in the log file.
- *Save Element History to Excel*: Save the element histories to an Excel file. See a description of the Excel History file in the visualization section later in this chapter. This is a very useful option, but can slow down the simulation significantly for long scenarios.

Push the "Run Simulation" button to simulate the scenario by creating time histories for each element, node and supply class, checking feasibility, and calculating the MOEs. The simulation carries out the defined processes for the scenario in order. Rather than specifying the times that each element performs actions, each element has a list of actions that are carried out sequentially. This means that the simulation is event-driven rather than time-driven. The models will calculate the consumption for their respective classes of supply and update the element histories accordingly. If an error is discovered with a process, that same process will appear as a red bar on the canvas in Figure 46 and the error will be logged in the scenario log file. Processes with no errors appear as green bars on the canvas.

A waitbar will pop up and display the progress of the simulation. Calculating the MOEs and saving the element histories to Excel is computationally intensive and may take a while (between 5 seconds and several minutes depending on complexity). When the simulation is complete the MOEs will be displayed on the right side of the panel and a message box with the number of errors encountered will pop-up (see Figure 47 below).


Finished simulating scenario: Apollo_17

There were no errors.

Figure 47: Simulation Pop-Up

If errors are present the user can press the "Display Log File" button to view the ASCII text file that was written during simulation. Understanding in which element and process an error occurred is helpful in trying to resolve the issue by editing the scenario. Typical errors are:

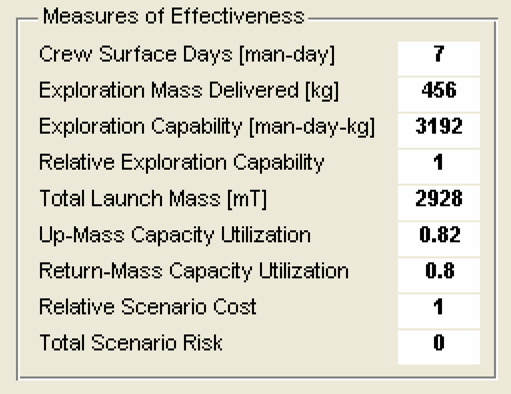
- Insufficient propellant to execute a burn
- Running out of crew provisions or crew operations in a crewed element
- Attempting to transfer supply items into an element that has insufficient capacity to accept the incoming item(s).

Generally it is up to the user to resolve errors in an iterative fashion by going through a number of simulate-edit-simulate-edit cycles in a particular scenario. SpaceNet 1.3 does not contain a detailed “wizard” that would automatically resolve errors (since there might be multiple non-unique ways in which an error can be resolved), but instead relies on the user’s experience and domain knowledge to repair infeasible scenarios.

Initially users will experience errors more frequently, but as their experience level in working with SpaceNet 1.3 increases, errors will become less and less frequent.

Measures of Effectiveness

Any supply chain is a system, and as such, its design should involve comparing alternatives by objectively measuring how well each achieves the system’s purpose. SpaceNet provides a means for doing that by calculating measures of effectiveness (MOEs). Brief descriptions for the current nine MOEs in SpaceNet 1.3 are provided here. See Appendix D for more detailed explanations of each MOE. After simulating a scenario, a typical result may be as shown in Figure 48.



Crew Surface Days [man-day]	7
Exploration Mass Delivered [kg]	456
Exploration Capability [man-day-kg]	3192
Relative Exploration Capability	1
Total Launch Mass [mT]	2928
Up-Mass Capacity Utilization	0.82
Return-Mass Capacity Utilization	0.8
Relative Scenario Cost	1
Total Scenario Risk	0

Figure 48: MOEs

Crew Surface Days (CSD)
The total cumulative number of crew surface days spent over an entire scenario over all nodes associated with exploration processes.

Exploration Mass Delivered (EMD)
The total amount of exploration mass delivered to all exploration nodes over the course of a campaign (scenarios with multiple launches). Exploration mass contains all items that are characterized as COS 6 (exploration items) and COS 8 (habitation and infrastructure).

Total Launch Mass (TLM)
The total launch mass lifted off from the surface of the Earth to accomplish a particular exploration scenario. TLM is given in metric tons [mT].

Up-Mass Capacity Utilization (UCU)

The ratio of the mass of tracked supply classes launched from Earth over all elements in a scenario, divided by the total usable cargo mass capacity (= maximum cargo up-mass capacity less accommodation mass). For example, a UCU of 0.82 corresponds to an 82% up-mass capacity utilization.

Return-Mass Capacity Utilization (RCU)

The ratio of the mass of tracked supply classes returned to Earth over all elements in a scenario, divided by the total usable cargo return-mass capacity (= maximum cargo return-mass capacity less accommodation mass). For example, a RCU of 0.80 corresponds to an 80% return-mass capacity utilization.

Exploration Capability (EC)

Exploration capability is the amount of time the crew gets to spend doing exploration and research at all exploration nodes, multiplied by the amount of total exploration mass they have to do the job at each node (associated with an exploration process) visited during the scenario. Exploration mass and crew have to be *collocated* at an exploration node for EC to be generated.

Relative Exploration Capability (REC)

The REC is a normalized measure of *exploration logistics efficiency*. It measures the amount of productive exploration that can be done for each kilogram of mass launched from the Earth's surface, relative to Apollo 17. Apollo 17 is used as the reference case because it can be argued that of all the Apollo lunar surface missions, Apollo 17 was the most productive in terms of exploration and science and also the one that came closest to approaching the constraints imposed by flight hardware elements and operational capabilities at that time. Thus, a relative exploration capability of *REC*=2.5 would indicate that 2.5 times the amount of exploration capability *per kg of mass launched* can be achieved, relative to Apollo 17.

Relative Scenario Cost (RSC)

A normalized measure of operations costs. It measures production/launch costs and mission operations costs relative to the Apollo 17 scenario. Therefore the cost is not monetized ($) but a measure of scenario complexity.[11]

Total Scenario Risk (TSR)

The combined probability of failure for all processes (Transport, Proximity Ops, etc.) in a scenario. The probability of failure of each SpaceNet process may depend in part on the particular process elements involved. The TSR risk measure in SpaceNet 1.3 is essentially a weighted sum (See Appendix D for details) of individual process risks and provides only a crude measure of risk at this point. However, the risks associated with undersupply situations, insufficient spares (backorders) and failed resupply flights can indeed be assessed in detail.

---

[11] A monetized measure of operations cost in FY$ can be obtained from other costing tools such as the ExAOCM (Exploration Architectures Operations Cost Model) developed by JPL.

Error Handling

There are several error cases that are tracked during the simulation. These errors include user logic errors in creating processes, such as a transfer process between two elements when they are not at the same node. The errors also include calculated infeasibilities, such as elements that do not have enough propellant to perform specified burns during a transport process.

Any errors encountered are logged in the output file created when running the simulation. The errors can be handled in two ways. The simulation can terminate upon running into an error and the user must go back and fix the error before running the simulation again. Alternatively, the simulation can attempt to continue so that more information can be gleaned. The simulation will handle the error as best it can before moving on, such as consuming all of a given class of supply in cases of shortage.

A list of the potential error conditions in the SpaceNet processes is given in Appendix E. The error condition and handling of the error are first described, and then the specific error and workaround texts that would be seen in the output log file are given.

Push “Save Scenario” to save the element histories and MOEs to file.

Visualization

An important part of any supply chain management and logistics planning and simulation tool is the ability to visualize both overview and detailed information about the supply chain. In commercial terrestrial supply chain software this typically involves a network view of the system, showing the locations of manufacturing plants, warehouses, distributors and retail stores along with the active transportation routes connecting them. For individual nodes in the network inventory levels are often also shown using a variety of symbols. In an analogous fashion SpaceNet 1.3 provides a number of visualizations to help users better understand the scenarios which they are creating or analyzing.

Push the "Visualization" button in the "Navigation" panel to display the canvas in Figure 49. The user can visualize the scenario simulation results in four different ways:

- *Network Visualization*: Animate elements from a network perspective
- *Bat Visualization*: Animate elements over time in a "bat" diagram
- *CoS Visualization*: Create plots of time history of any class of supply in any element
- *Excel Visualization*: Create plots of the element, node and supply class time histories in Excel using Visual Basic

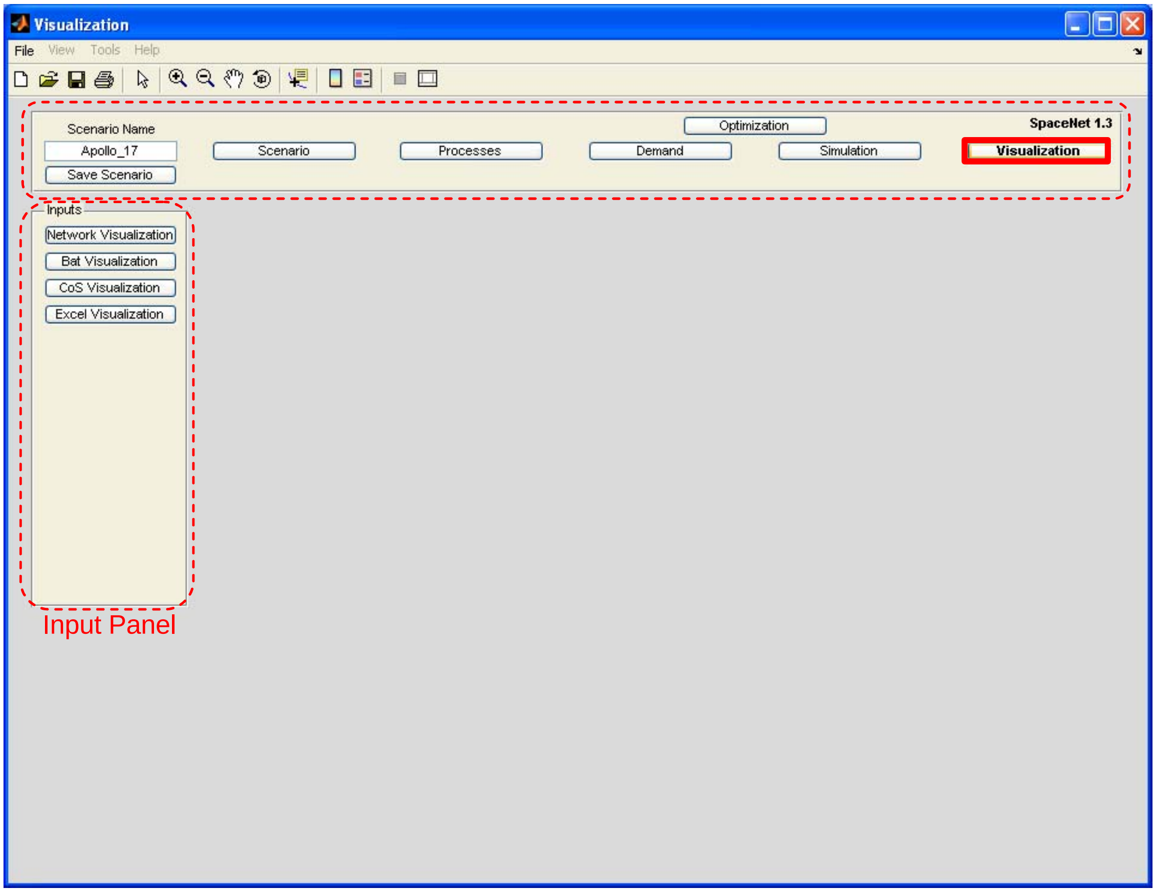
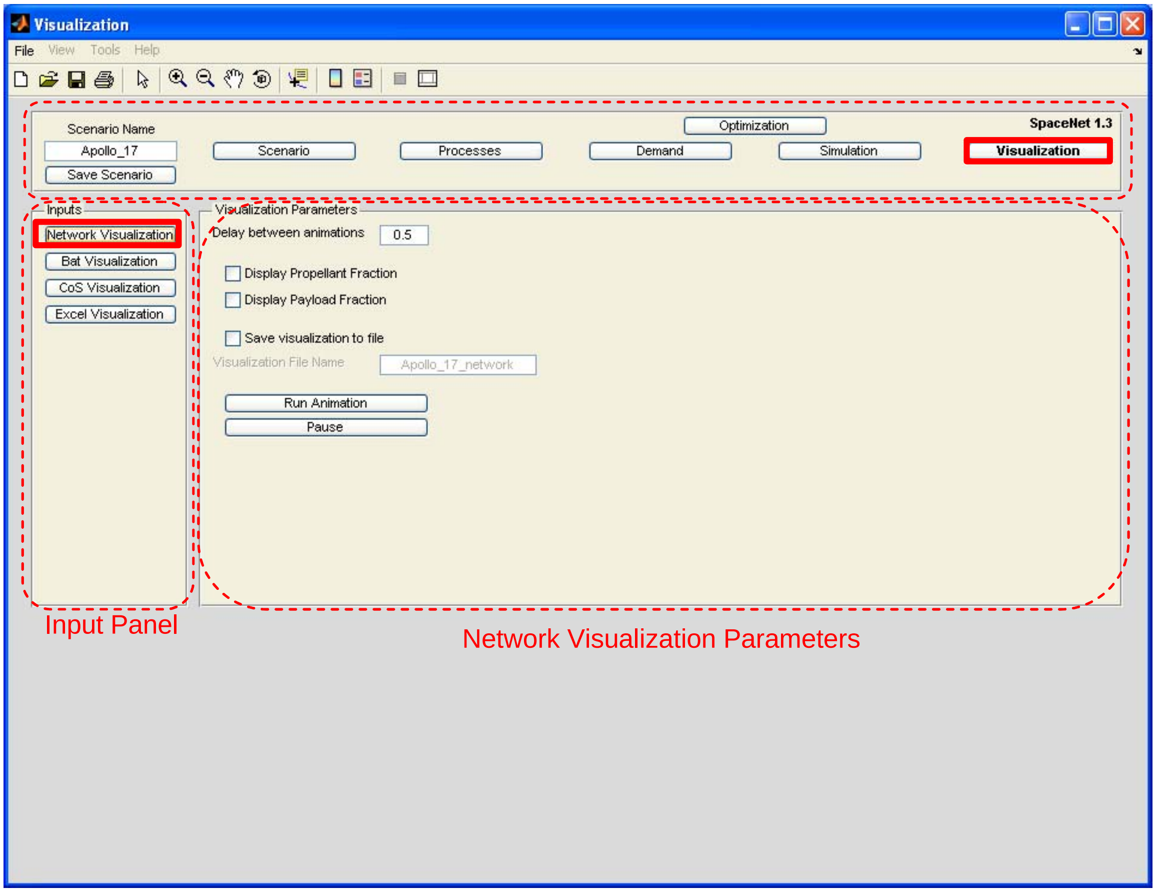


Figure 49: Visualization

Network Visualization

Push the “Network Visualization” to display the “Visualization Parameters” panel on the canvas shown in Figure 50. These parameters are specifically for the network visualization.



Scenario Name: Apollo_17

Delay between animations: 0.5

☐ Display Propellant Fraction

☐ Display Payload Fraction

☐ Save visualization to file

Visualization File Name: Apollo_17_network

Figure 50: Network Visualization

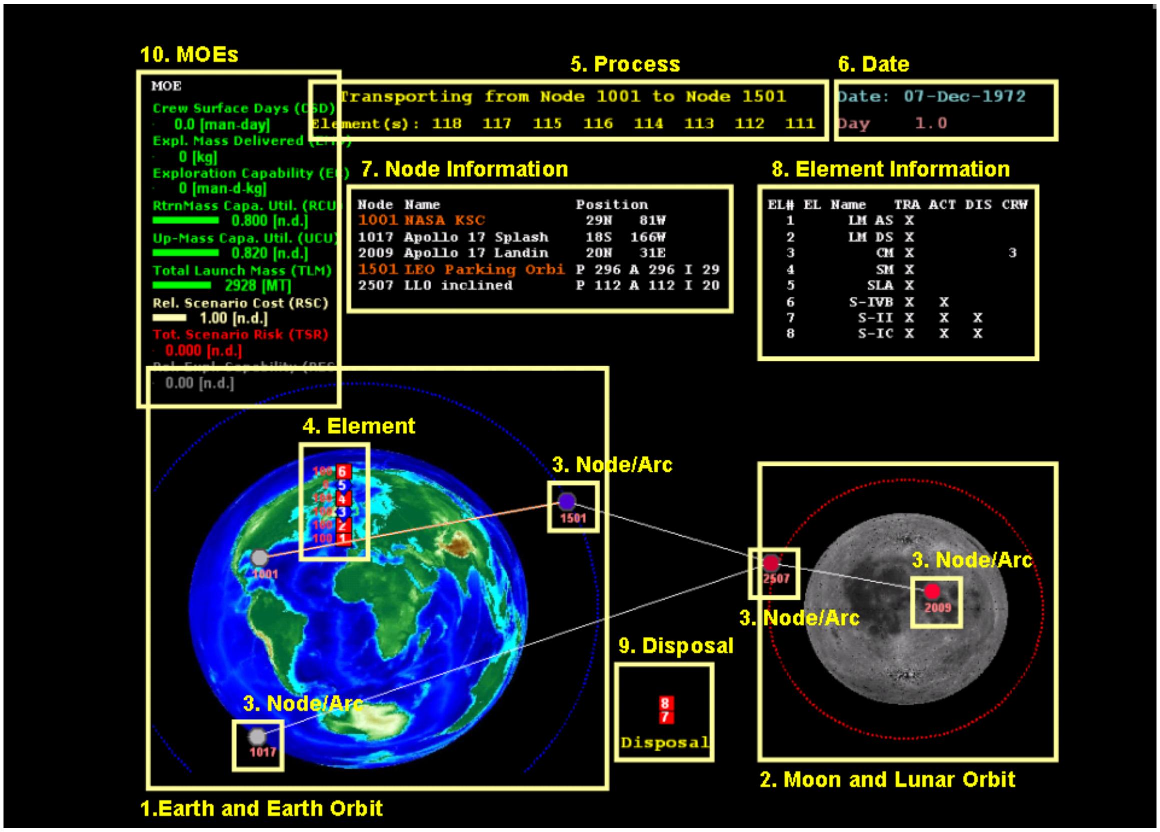
The user can set the following network visualization parameters:

- *Delay between animations*: This sets the delay between animation frames (in seconds). The default value is 0.5 seconds. If a negative value is entered, the animation is held for each frame and the user has to press a key to advance frames
- *Display Propellant Fraction*: Displays the ratio of the current propellant mass to the initial propellant mass of an element (to its left). Disables Display Payload Fraction
- *Display Payload Fraction*: Displays the ratio of the current payload to the maximum payload (cargo) capacity of an element (to its left). Disables Display Propellant Fraction

- *Save Visualization to file*: Exports the animation to an **.avi** file format for the user to play with external applications such as Windows Media Player or QuickTime
- *Visualization File Name*: The name of the .avi file. The default name is the scenario name appended with "_network.avi"

Network Visualization Figure

Push the "Run Animation" button to open the network visualization figure and run the animation. The visualization for the Apollo 17 scenario is shown below in Figure 51. The user can push the "Pause" button to pause the current frame anytime during the animation. Upon pausing the animation, the button changes from "Pause" to "Resume", which the user can push to resume the animation as before.



```
MOE
Crew Surface Days (CSD)
  0.0 [man-day]
Expl. Mass Delivered
  0 [kg]
Exploration Capability (E
  0 [man-d-kg]
RtrnMass Capa. Util. (RCU
  0.800 [n.d.]
Up-Mass Capa. Util. (UCU)
  0.820 [n.d.]
Total Launch Mass (TLM)
  2928 [MT]
Rel. Scenario Cost (RSC)
  1.00 [n.d.]
Tot. Scenario Risk (TSR)
  0.000 [n.d.]
  0.00 [n.d.]
```

```
Transporting from Node 1001 to Node 1501
Element(s): 118  117  115  116  114  113  112  111
```

```
Date:  07-Dec-1972
Day     1.0
```

```
Node Name                 Position
1001 NASA KSC              29N   81W
1017 Apollo 17 Splash      18S  166W
2009 Apollo 17 Landin      20N   31E
1501 LEO Parking Orbi   P 296 A 296 I 29
2507 LLO inclined       P 112 A 112 I 20
```

```
EL# EL Name   TRA ACT DIS CRW
  1     LM AS  X
  2     LM DS  X
  3        CM  X              3
  4        SM  X
  5       SLA  X
  6     S-IVB  X   X
  7      S-II  X   X   X
  8      S-IC  X   X   X
```

Figure 51: Network Visualization Figure

Detailed explanations for each numbered box in Figure 51 are presented below.

[1] Earth and Earth Orbit

The Earth's image was created using the MATLAB mapping toolbox. To display Earth surface nodes over the entire globe, the Lambert-Azimuthal method was selected to project the Earth onto 2-D space. A reference Earth orbit (300 km) is plotted as a dotted blue line. Dimensions and relative proportions in this view are not to scale.

[2] Moon and Lunar Orbit

The Moon image is also created using the MATLAB mapping toolbox and the Lambert-Azimuthal projection. The reference lunar orbit (100km) is plotted as a dotted red line. Dimensions and relative proportions in this view are not to scale.

[3] Node / Arc

The (physical) nodes and arcs in the scenario are displayed in the network visualization. The positions of surface nodes in the figure are determined by their respective latitude/longitude information. The positions of orbit nodes are determined by their inclination/altitude information. The 4-digit number beneath the circle representing the node is the unique node ID, corresponding to that node in the integrated database.

If there are existing arcs between two nodes, a line representing the arc is displayed. At the start of the visualization all arcs are colored white. When the animation is in progress and an arc is in use, its color turns to orange.

[4] Element

The instantiated elements in the scenario are also displayed in the network visualization as illustrated in Figure 52. There are two colors/shapes that represent the element type. Red squares generally depict propulsive elements that cannot carry crew and blue diamonds depict elements that can carry crew. An element number (for visualization purposes) is displayed in the center of each element in white.

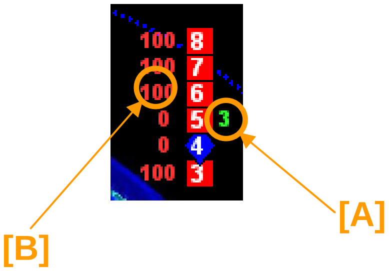
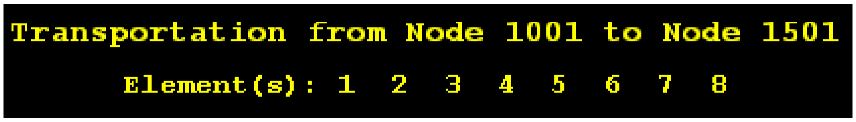


Figure 52: Network Visualization Figure – Element

A green number may appear to the right of some elements during the course of the animation (as in Figure 52 above, see [A]). It represents the number of crew present inside the element at that time step.

If the user selected either the "Display Propellant Fraction" or "Display Payload Fraction" option in the "Visualization Parameters" panel, red numbers will appear to the left of the elements (as in Figure 52 above, [B]). The number varies from 0 (0% propellant/payload fraction) to 100. (100% propellant/payload fraction).

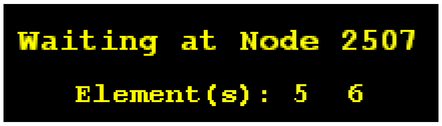
[5] Process



Transportation from Node 1001 to Node 1501

Element(s): 1 2 3 4 5 6 7 8

Transport Process



Waiting at Node 2507

Element(s): 5 6

Wait Process

Figure 53: Network Visualization Figure – Process

Out of the five processes SpaceNet models the network visualization currently animates only wait and transport processes. For transport processes, the departure/arrival nodes and the transported elements are animated. For wait processes, the node where the elements are waiting is displayed along with the elements, see Figure 53.

[6] Date

SpaceNet provides both absolute dates (DD-MM-YYYY) and relative dates (the scenario start date is 1).

[7] Node Information

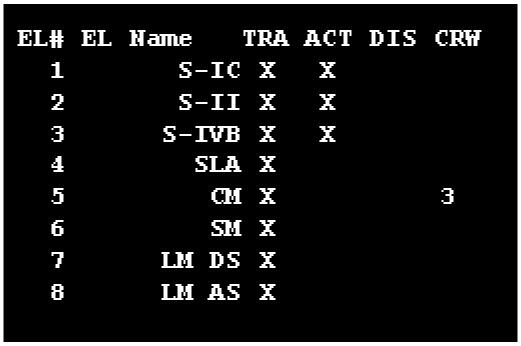
Node	Name	Position
1001	NASA KSC	29N 81W
1017	Pacific Ocean	18S 166W
2009	Apollo 17 Landin	20N 31E
1501	LEO Parking Orbi	P 296 A 296 I 29
2507	LLO inclined	P 112 A 112 I 20

Figure 54: Network Visualization Figure – Node Information

Detailed information for each node in the scenario is displayed in this section of the network visualization: node ID, node name, and node position. For surface nodes, longitude and latitude information is shown for the node position. For orbit nodes, perigee, apogee, and inclination information is shown for the node position.

The color of the node ID and node name texts turn orange when the node is associated with a currently animated process as shown in Figure 54.

[8] Element Information



```
EL# EL Name    TRA ACT DIS CRW
  1       S-IC X   X
  2       S-II X   X
  3      S-IVB X   X
  4        SLA X
  5         CM X           3
  6         SM X
  7      LM DS X
  8      LM AS X
```

Figure 55: Network Visualization Figure – Element Information

The name and current status for each element in the scenario is displayed in this section. TRA is the abbreviation for “transporting.” So for each element in transit, “X” is marked in the TRA column. The ACT (“active element”) column shows if an element is executing a burn in the current process. The DIS (“disposed”) column shows if an element is disposed after/before the current process. The CRW (“crew”) column shows the number of crew inside the element. This number is identical to the green number displayed to the right of each animated element.

Interpreting the element information shown above in Figure 55, during the current process elements 1-8 are transported together with elements 1, 2, and 3 being active. No element is/will be disposed during the process. Three crew members are present inside element number 5 (CM).

[9] Disposal

Figure 56: Network Visualization Figure – Disposal

Some elements, mainly propulsive ones, are disposed off (=staged) after they are used. The disposed elements are presented in Figure 56.

[10] Measures of Effectiveness (MOEs)

The scenario MOEs are displayed both after the simulation and continuously updated during the visualization.

Descriptions and explanations for each MOE are provided in Appendix D and in the simulation section above. The length of the bars on the left of the MOEs exhibits the relative (log-scale) magnitude of the MOE compared with a reference scenario (Apollo

17). The different color codes in Figure 57 represent the different groups of MOEs (Green: Basic Logistics Performance, Yellow: Cost, Red: Risk, Gray: Relative Exploration Capability).

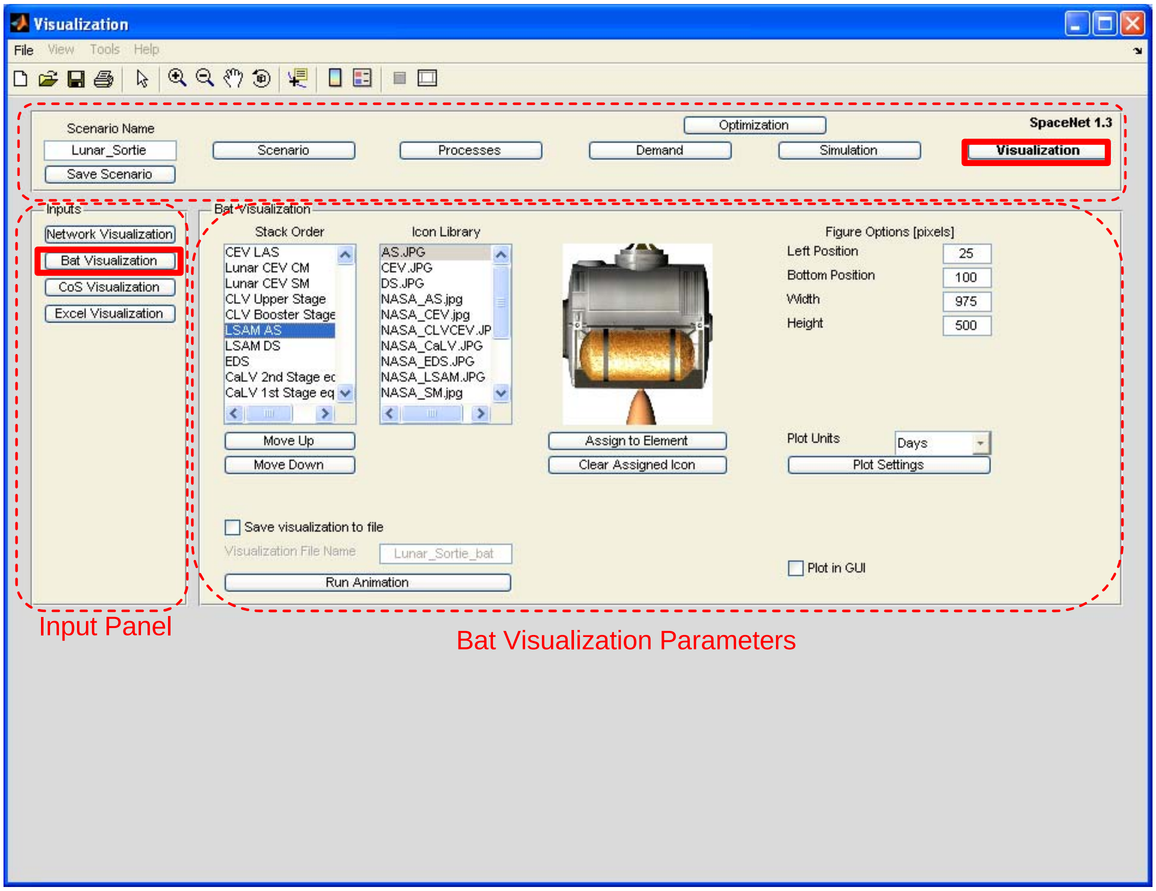
MOE
Tr
Crew Surface Days (CSD)
0.0 [man-day] Elem
Expl. Mass Delivered (EMD)
0 [kg]
Exploration Capability (EC)
0 [man-d-kg]
RtrnMass Capa. Util. (RCU)
0.800 [n.d.]
Up-Mass Capa. Util. (UCU)
0.820 [n.d.]
Total Launch Mass (TLM)
2928 [MT]
Rel. Scenario Cost (RSC)
1.00 [n.d.]
Tot. Scenario Risk (TSR)
0.000 [n.d.]
Rel. Expl. Capability (REC)
0.00 [n.d.]

Figure 57: Network Visualization Figure – MOEs

Bat Visualization

A traditional way in which mission architects at NASA (an in other organizations) depict transportation architectures is using so called bat diagrams. A bat diagram shows time on the x-axis and different nodes are represented as horizontal lines intersecting the y-axis. Earth is typically shown at the bottom, while the node(s) to be visited (ISS, Moon, Mars) is shown on top. When elements (=vehicles) arrive at their destination on top they are often shown "hanging upside down" like bats at rest, thus the name of this representation

Push "Bat Visualization" to display the "Bat Visualization" panel on the canvas as shown in Figure 58.



Scenario Name: Lunar_Sortie

Left Position: 25

Bottom Position: 100

Width: 975

Height: 500

Plot Units: Days

☐ Save visualization to file

Visualization File Name: Lunar_Sortie_bat

☐ Plot in GUI

Figure 58: Bat Visualization

(Note: the figures in this section of the manual were generated using the Constellation Lunar_Sortie scenario). The elements in the scenario are displayed in the “Stack Order” listbox to the left of the panel. By selecting elements and pushing the “Move Up” and “Move Down” buttons, the user can set the order at which elements are stacked on top of one another in the bat animation.

Each element can be represented as either text or an icon. The default representation for each element is its name. To assign an icon to an element, first select the element in the “Stack Order” listbox. Then, browse through the “Icon Library” listbox. In SpaceNet 1.3 icons are imported from the \icons directory as jpeg files. Selecting an icon will display its image to the right. Push “Assign to Element” to assign the selected icon to the selected element. Push “Clear Assigned Icon” to clear an assigned icon from a selected element.

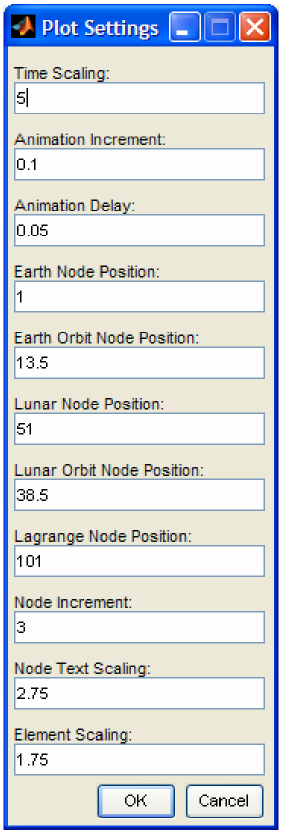
The user can set the following bat visualization parameters:

- *Save visualization to file*: Exports the animation to an **.avi** file for the user to play the bat animation with applications such as Windows Media Player
- *Visualization File Name*: The name of the .avi file. The default name is the scenario name appended with “_bat”
- *Plot in GUI*: Plots the bat visualization in the SpaceNet GUI instead of a new window

Figure Options

- *Left Position*: The position of the new figure from the left of the screen (in pixels)
- *Bottom Position*: The position of the new figure from the bottom of the screen (in pixels)
- *Width*: The width of the new figure (in pixels)
- *Height*: The height of the new figure (in pixels)
- *Plot Units*: Changes the ticks at the bottom of the plot to display either days, weeks, months, or years

Plot Settings (see Figure 59)



Time Scaling: 5
Animation Increment: 0.1
Animation Delay: 0.05
Earth Node Position: 1
Earth Orbit Node Position: 13.5
Lunar Node Position: 51
Lunar Orbit Node Position: 38.5
Lagrange Node Position: 101
Node Increment: 3
Node Text Scaling: 2.75
Element Scaling: 1.75

Figure 59: Bat Visualization Plot Settings

- *Time Scaling:* Multiplicative factor used to widen the space between animations on the x-axis
- *Animation Increment*: The fraction of the time discretization at which to animate. If the time discretization is 1 day and the time increment is 0.2, then 5 animations are created for each day. Time increment values are currently limited to 0, 0.1, 0.2, 0.5, and 1
- *Animation Delay*: The delay between animation frames (in seconds)
- *Earth Node Position*: The number at which earth surface nodes are first plotted on the y-axis
- *Earth Orbit Node Position*: The number at which earth orbit nodes are first plotted on the y-axis
- *Lunar Node Position*: The number at which lunar surface nodes are first plotted on the y-axis
- *Lunar Orbit Node Position*: The number at which lunar orbit nodes are first plotted on the y-axis
- *Lagrange Node Position*: The number at which Lagrange nodes are first plotted on the y-axis
- *Node Increment*: The number by which each subsequent node is incremented from either earth surface/orbit or lunar surface/orbit node positions
- *Node Text Scaling*: The number by which the node texts are offset to the left of the origin
- *Element Scaling*: The number by which the element texts are offset to the left of the origin

Bat Visualization Figure

Push the “Run Animation” button to open the bat visualization figure and run the animation. A snapshot of the bat visualization for the Constellation Lunar Sortie scenario is shown below in Figure 60.

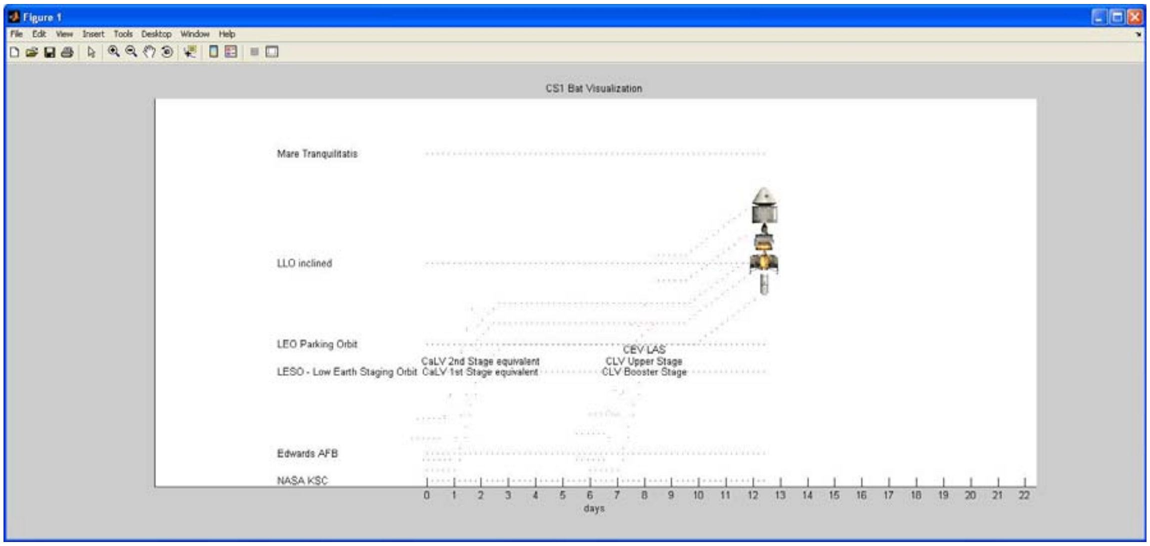


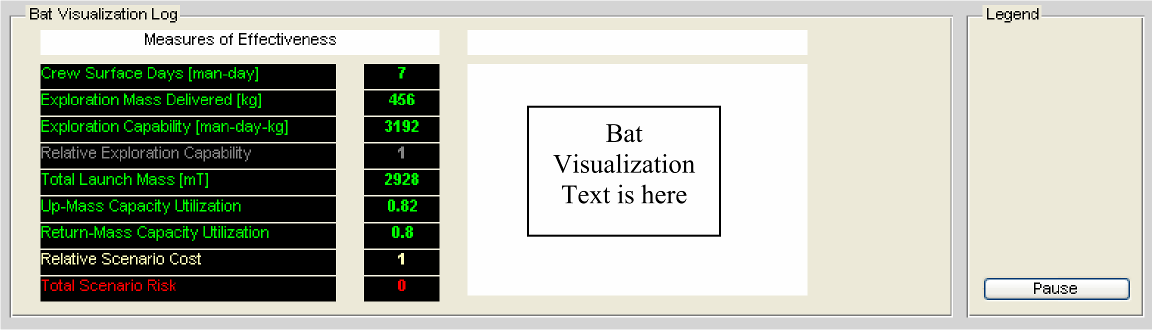
Figure 60: Bat Visualization Figure

The nodes in the scenario are plotted on the left of the origin at a distance given by the *Node Text Scaling* value. The nodes are ordered from bottom to top as follows: Earth surface nodes, Earth orbit nodes, lunar orbit nodes, and lunar surface nodes. “NASA KSC”, the first earth surface node, is plotted at the *Earth Node Position*. “Edwards AFB”, the second earth surface node, is plotted at the *Earth Node Position* plus the *Node Increment*. The other groups of nodes follow the same convention.

In the example in Figure 60, the lines of text represent elements that do not have icons assigned to them (the launch stack elements). The “CaLV 1st Stage equivalent” and “CaLV 2nd Stage equivalent” are grouped together and separated at a distance given by the *Stack Space* value. Discarded elements are represented by elements that are no longer animated. The groups of text plotted above are the discarded elements from the two launches in the scenario.

The stack of icons on the right is the group of elements traveling from the “LEO Parking Orbit” node to the “LLO inclined” node. The order in which the icons are displayed is given by the “Stack Order” listbox. The icons displayed from bottom to top are: “EDS”, “LSAM DS”, “LSAM AS”, “Lunar CEV SM” and “Lunar CEV CM”, which agrees with the order shown in the bat visualization Figure 60.

The “Run Animation” button also makes visible the “Bat Visualization Log” panel (Figure 61).



Measures of Effectiveness	
Crew Surface Days [man-day]	7
Exploration Mass Delivered [kg]	456
Exploration Capability [man-day-kg]	3192
Relative Exploration Capability	1
Total Launch Mass [mT]	2928
Up-Mass Capacity Utilization	0.82
Return-Mass Capacity Utilization	0.8
Relative Scenario Cost	1
Total Scenario Risk	0

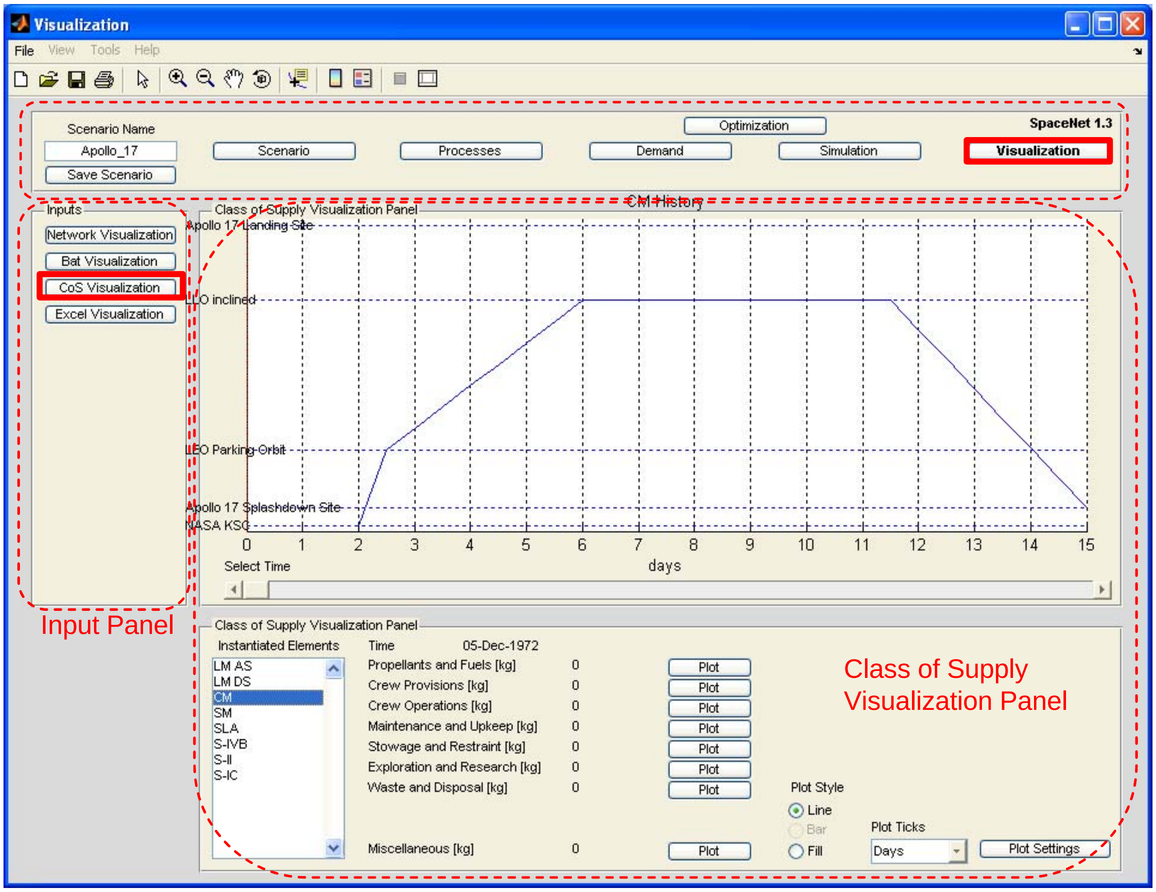
Figure 61: Bat Visualization Log

While the bat visualization is running, either in the main GUI or in a new figure, the “Bat Visualization Log” panel displays two things: MOEs and cargo transfers between elements. The MOEs are displayed in concurrence with the bat visualization, that is, if the animation is currently on day 10 then the MOEs at day 10 are displayed. When a transfer occurs in the animation, text describing the transfer is displayed to the right of the “Bat Visualization Log” panel. The user can push the “Pause” button to pause the current frame anytime during the animation. Upon pausing the animation, the button changes from “Pause” to “Resume”, which the user can push to resume the animation as before.

Push “Save Scenario” to save the bat visualization settings to file (including element-icon assignments).

COS Visualization

Push “CoS Visualization to display the “Class of Supply Visualization Panel” as shown in Figure 62.



Scenario Name Apollo_17

Time 05-Dec-1972

Propellants and Fuels [kg]	0
Crew Provisions [kg]	0
Crew Operations [kg]	0
Maintenance and Upkeep [kg]	0
Stowage and Restraint [kg]	0
Exploration and Research [kg]	0
Waste and Disposal [kg]	0
Miscellaneous [kg]	0

Plot Style ● Line

Plot Ticks Days

Figure 62: Class of Supply Visualization

The CoS Visualization displays the nodal time history of whichever element is selected in the “Instantiated Elements” listbox on the canvas. The user can also display plots of the time history of each class of supply (again for the element selected).

The following parameters can be changed by the user:

- *Plot Style*
  - *Line:* Plot a line graph
  - *Bar:* Plot a bar graph (currently disabled in SpaceNet 1.3)
  - *Fill:* Line graph with areas filled in
- *Plot Ticks:* Plot the results by month, week, day or year
- *Plot Settings:* Adjustment of several plot settings (see explanation in the Bat Visualization section)

Each plot created will open as a new figure. Figure 63 shows the time history of crew provisions in the CM during a lunar surface exploration mission. Initially the consumption rate corresponds to 3 crew members being present in the CM (during launch, LEO operations and Earth-Moon cruise). During lunar surface operations, however, only a single crew member remains in the CM and the rate of consumption is reduced accordingly. After return of the 2 crew members from the lunar surface, the rate of consumption of crew provisions increases again and now reflects the presence of three crew members until return and reentry at Earth.

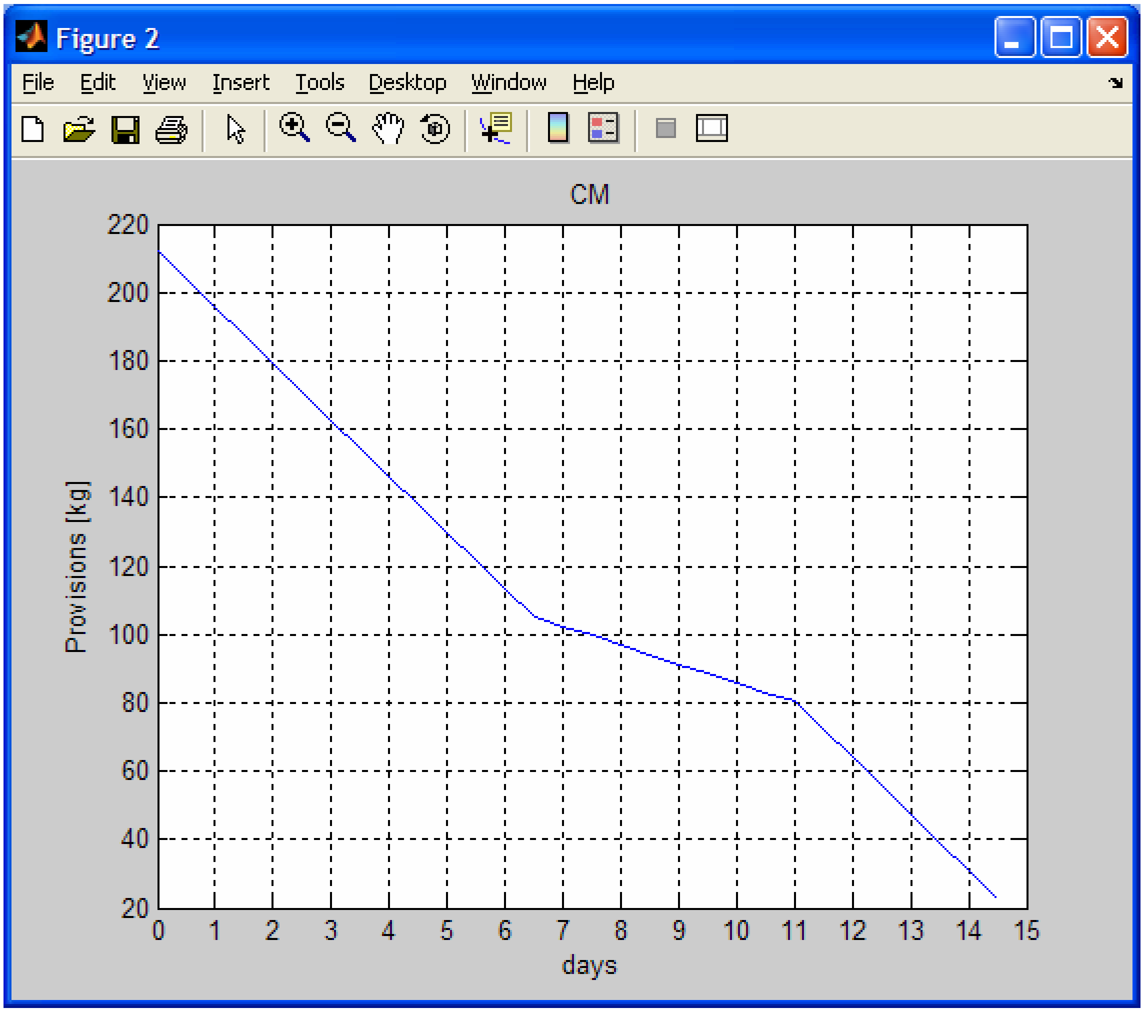
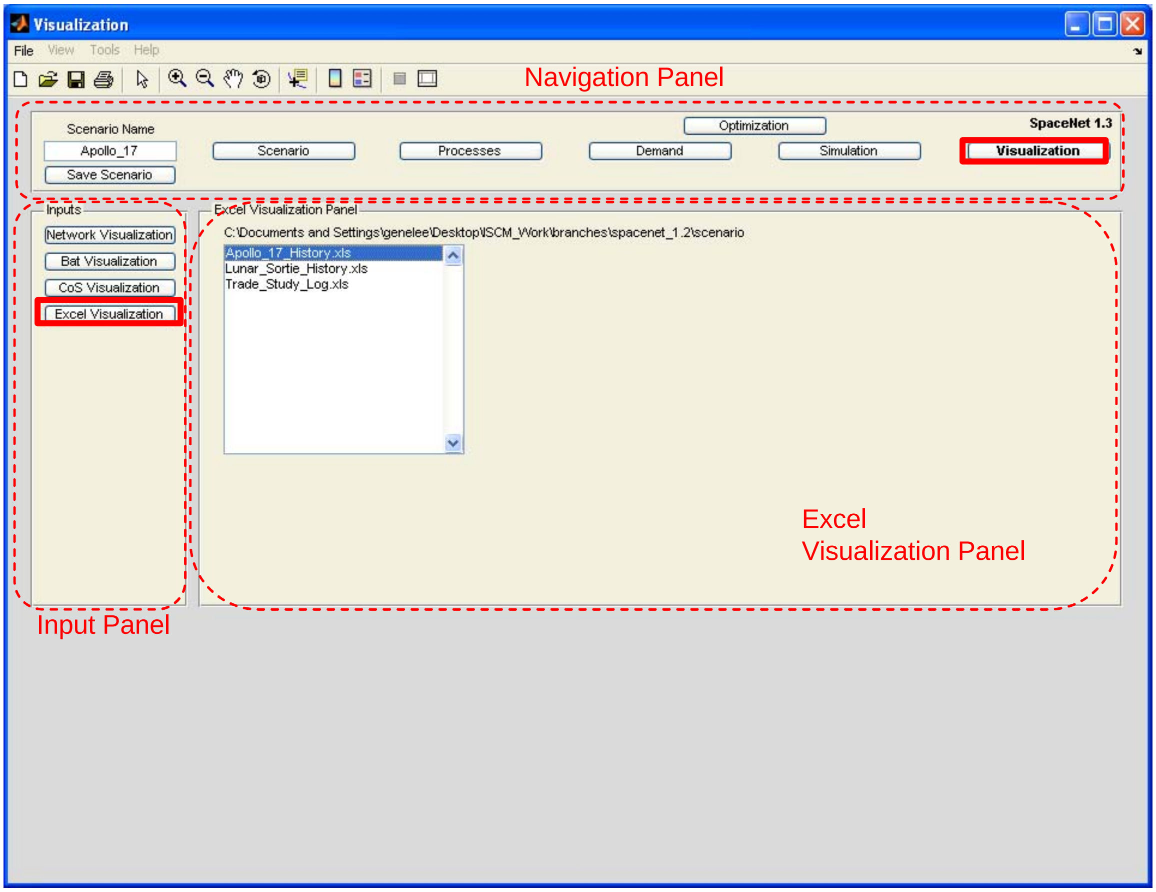


Figure 63: Class of Supply Visualization – CM Crew Provisions

Excel Visualization

Push “Excel Visualization” to display the “Excel Visualization Panel” in Figure 64.



Scenario Name: Apollo_17

C:\Documents and Settings\genelee\Desktop\SCM_Work\branches\spacenet_1.2\scenario

Apollo_17_History.xls
Lunar_Sortie_History.xls
Trade_Study_Log.xls

Figure 64: Excel Visualization

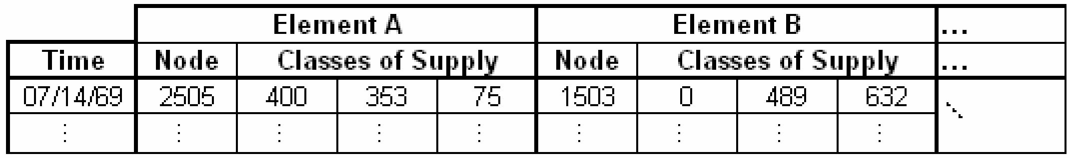
For the Excel visualization to work properly, the user must first save the element history to Excel in the simulation step. Any Excel files in the \scenario folder will appear in the listbox shown in Figure 64. The Excel visualization is activated when the user double-clicks an item in the listbox. A description of the Excel file is presented below, followed by a description of the Excel visualization (referred to as the *History Plotter*).

The Excel History File saves the scenario element history data from the simulation for easy data interpretation, post-processing and visualization. The element history data is rearranged into six different types of history tables with the data organized by element, node, or class of supply (CoS). The Excel History File has a built-in history plotter (programmed in Visual Basic) that can read the data from the history tables and generate mass tracking diagrams. Once created, Excel History Files can be opened with either SpaceNet or Excel.

The History Tables

The six different types of history tables are generated by manipulating the element history table shown in Table 1.

Table 1: Element History Table

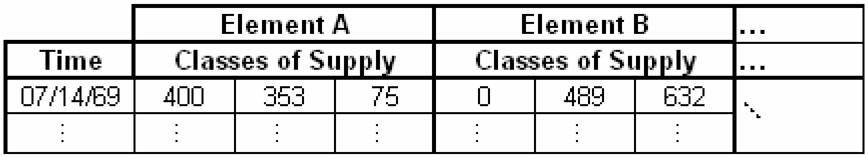


	Element A				Element B				...
Time	Node	Classes of Supply			Node	Classes of Supply			...
07/14/69	2505	400	353	75	1503	0	489	632	⋱
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

The time column specifies every time step in the scenario, the node column shows the ID of the node where the element is located at each time step, and the classes of supply columns hold the masses (in kg) of each class of supply that the element is carrying at each time step.

The first of the next two tables are created by deleting the node column from each element, and the second by deleting the classes of supply columns from each element.

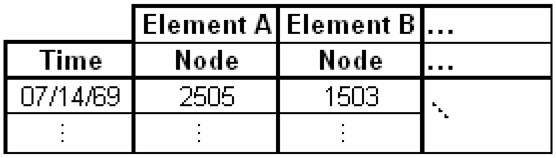
Table 2: Element Class of Supply History



	Element A			Element B			...
Time	Classes of Supply			Classes of Supply			...
07/14/69	400	353	75	0	489	632	⋱
⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Table 2 shows the mass of each class of supply in each element at each time step, grouped by element. Another table is formed by grouping Table 2 by class of supply instead of by element, as shown in Table 4.

Table 3: Element Node History

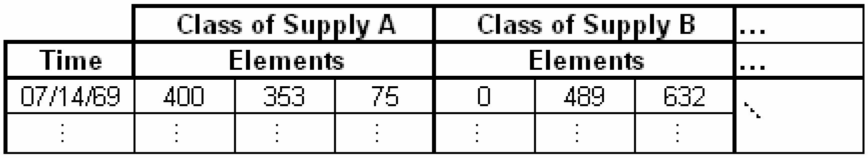


	Element A	Element B	...
Time	Node	Node	...
07/14/69	2505	1503	⋱
⋮	⋮	⋮	

Table 2 shows the mass of each class of supply in each element at each time step, grouped by element. Another table is formed by grouping Table 2 by class of supply instead of by element, as shown in Table 4.

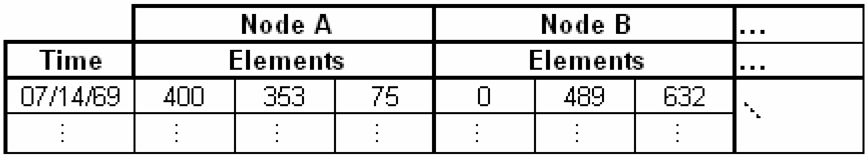
Table 3 shows the node ID where each element is at each time step. This table can also be regrouped to look like Table 5, which shows the total mass of all classes of supply carried by each element at each node at each time step.

Table 4: Class of Supply Element History



	Class of Supply A			Class of Supply B			…
Time	Elements			Elements			…
07/14/69	400	353	75	0	489	632	⋱
⋮	⋮	⋮	⋮	⋮	⋮	⋮	

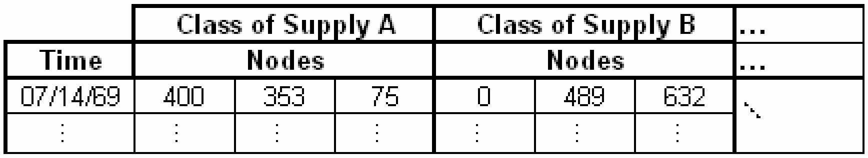
Table 5: Node Element History



	Node A			Node B			…
Time	Elements			Elements			…
07/14/69	400	353	75	0	489	632	⋱
⋮	⋮	⋮	⋮	⋮	⋮	⋮	

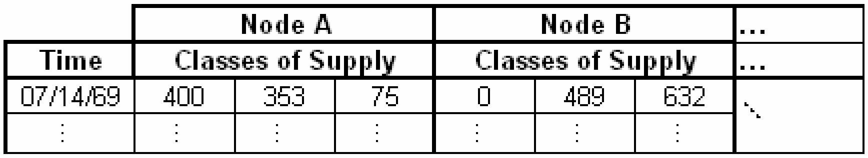
The following two tables show the mass of each class of supply at each node at each time step: one is grouped by node, the other by class of supply. Tables 6 and 7 are therefore very useful for determining the inventory level of various classes of supply at the nodes of the interplanetary supply chain. An example is for tracking the inventory levels at a lunar outpost over time.

Table 6: Class of Supply Node History



	Class of Supply A			Class of Supply B			…
Time	Nodes			Nodes			…
07/14/69	400	353	75	0	489	632	⋱
⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Table 7: Node Class of Supply History



	Node A			Node B			…
Time	Classes of Supply			Classes of Supply			…
07/14/69	400	353	75	0	489	632	⋱
⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Due to limitations in Excel, these six tables may be spread across more than six worksheets. Figure 65 shows the worksheet naming convention.

Figure 65: Worksheet Names

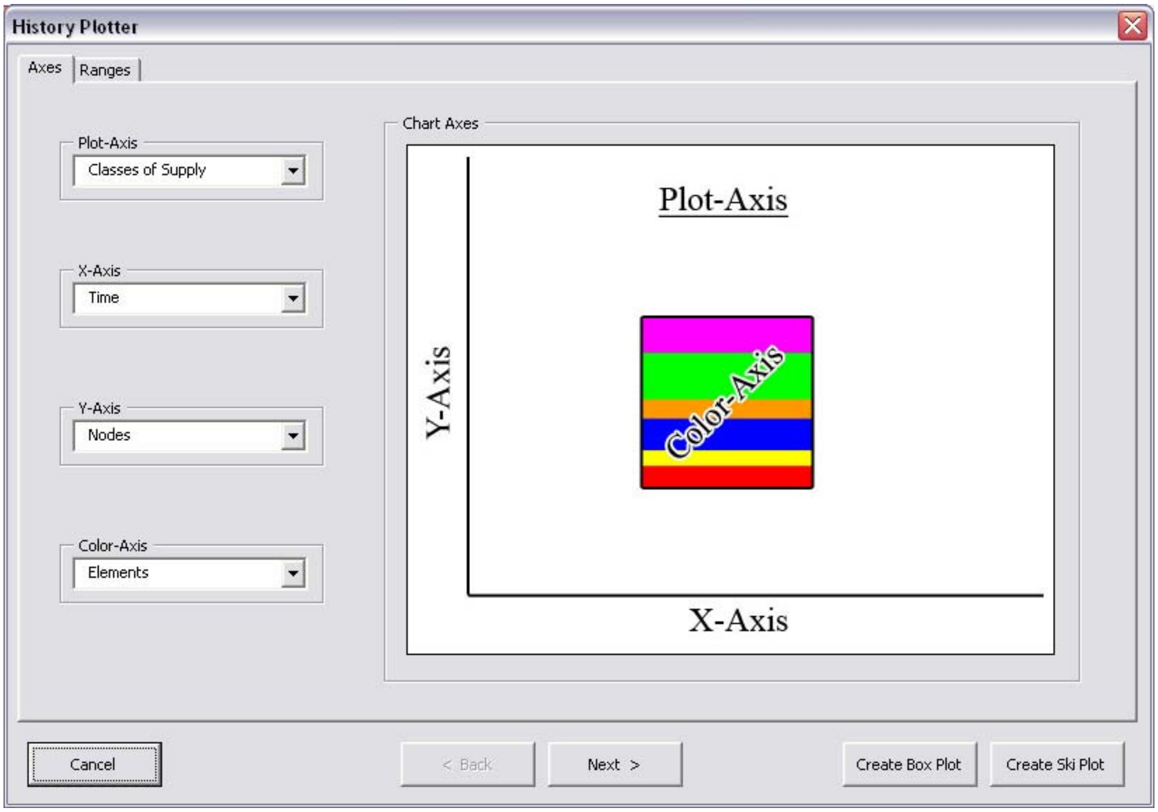
The first two letters of the worksheet name designate which of the six table types is on that sheet. For example, EC is the Element Class of Supply Table type. The “x of y” block of text designates which page of the total number of pages of that table type is on that worksheet, where x is the page number and y is the total number of pages.

The History Plotter

The History Plotter generates mass tracking diagrams of the scenario history. First open the Excel History File either in SpaceNet or Excel. Activate The History Plotter by clicking the "History Plotter" button in the Excel toolbar area, shown in Figure 66.

Figure 66: History Plotter Button

Clicking the "History Plotter" button opens the first page of History Plotter GUI, shown in Figure 67. The History Plotter GUI will also open when, from SpaceNet, the user saves the Excel History File during the simulation step.



Plot-Axis: Classes of Supply

X-Axis: Time

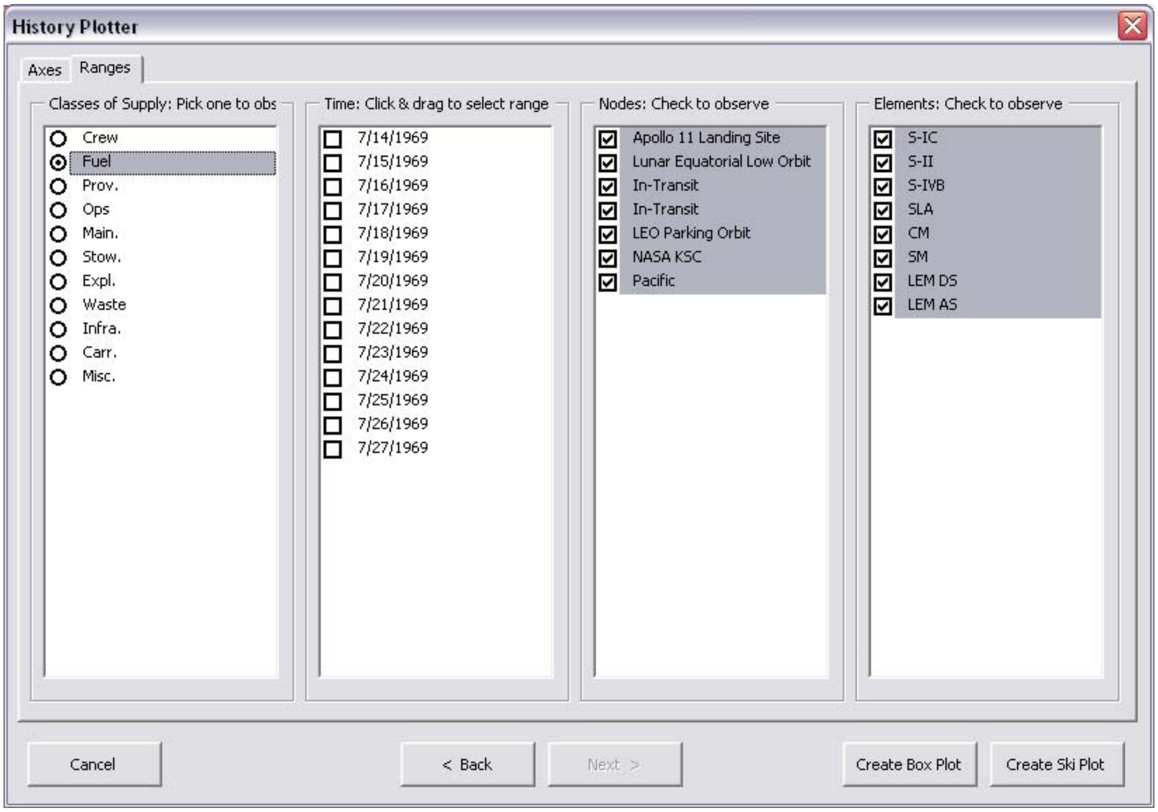
Y-Axis: Nodes

Color-Axis: Elements

Figure 67: History Plotter GUI - Axes Page

This is the first of two pages in the History Plotter GUI. On the "Axes" tab the axes of the mass tracking plot are assigned to the different history variables. The x-axis and y-axis are straight forward in their definitions. The plot-axis is the axis that will assign a history variable to the total area of the displayed squares. The color-axis is the axis that will assign a history variable to break down the plot-axis variable into smaller rectangles. This will become clearer with an example plot, which will be shown after describing the second page of the GUI.

The second page of the GUI is the Range tab, shown in Figure 68. The range page is where the different elements, nodes, classes of supply, and times are selected for observation. There are a large number of combinations to be chosen from, depending on the question being investigated. The Range page is activated by clicking the "Next" button. The "Axes" page can be returned to by clicking the "Back" button.



Classes of Supply: Pick one to obs
○ Crew
● Fuel
○ Prov.
○ Ops
○ Main.
○ Stow.
○ Expl.
○ Waste
○ Infra.
○ Carr.
○ Misc.

Time: Click & drag to select range
☐ 7/14/1969
☐ 7/15/1969
☐ 7/16/1969
☐ 7/17/1969
☐ 7/18/1969
☐ 7/19/1969
☐ 7/20/1969
☐ 7/21/1969
☐ 7/22/1969
☐ 7/23/1969
☐ 7/24/1969
☐ 7/25/1969
☐ 7/26/1969
☐ 7/27/1969

Nodes: Check to observe
☑ Apollo 11 Landing Site
☑ Lunar Equatorial Low Orbit
☑ In-Transit
☑ In-Transit
☑ LEO Parking Orbit
☑ NASA KSC
☑ Pacific

Elements: Check to observe
☑ S-IC
☑ S-II
☑ S-IVB
☑ SLA
☑ CM
☑ SM
☑ LEM DS
☑ LEM AS

Figure 68: History Plotter GUI - Range Page

Once the ranges are defined, push the "Create Box Plot" or "Create Ski Plot" buttons to generate their respective plots.

The example below will clarify the use of the History Plotter GUI. The default axes for a mass tracking plot are: time on the x-axis, nodes on the y-axis, elements on the color-axis, and classes of supply on the plot-axis. These are the default axes selections when the History Plotter is activated. Leave these alone and click next. Then select "Fuel" from the classes of supply, and click and drag to select the entire time range. Now the fuel will be broken down by element and plotted by nodes versus time. Click "Create Box Plot" to generate a plot that looks similar to the one shown in Figure 69.

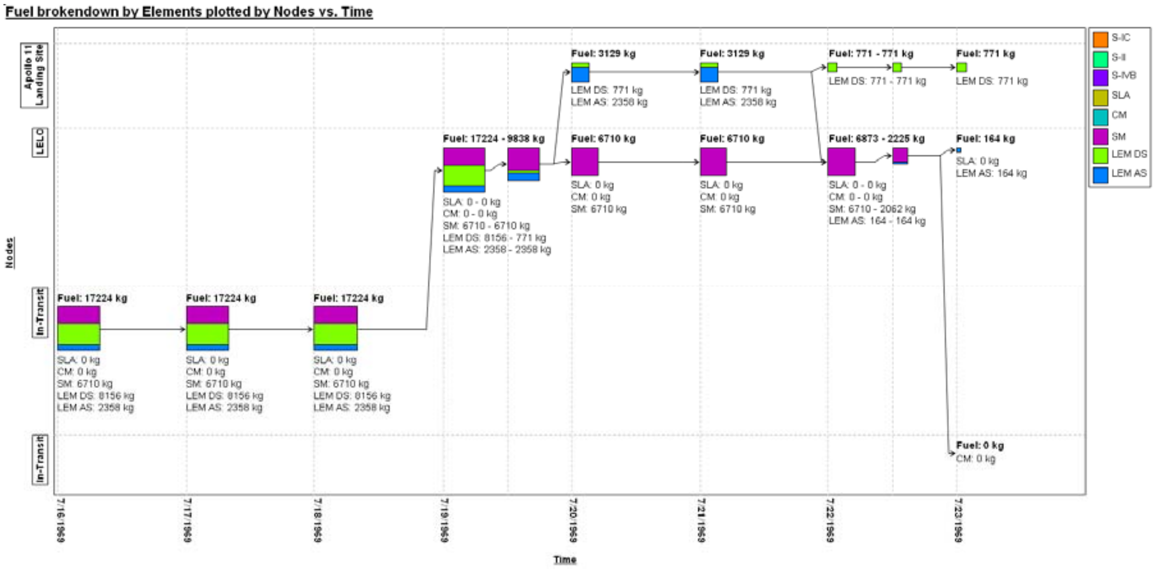


Figure 69: Box Plot

In this box plot, each box represents the total mass of fuel at a given node at a given time. The colors show how that fuel is distributed among the elements at that node at that time. The size of each square shows the relative amounts. The same plot can be generated as a ski plot, which shows the same information but also displays change in fuel from one time step to the next. A ski plot is shown in Figure 70 below.

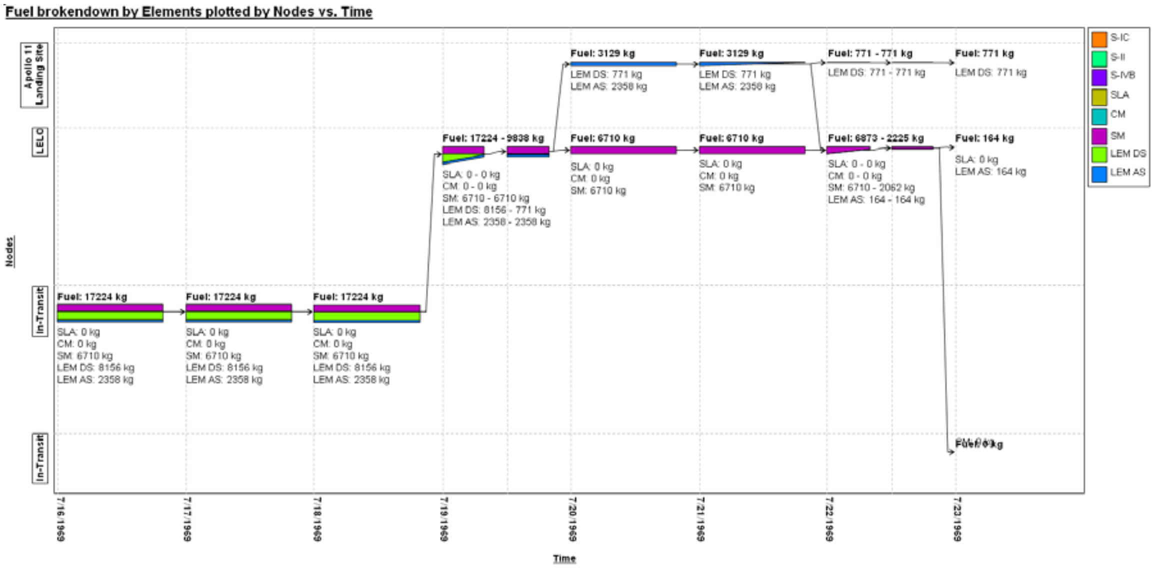


Figure 70: Ski Plot

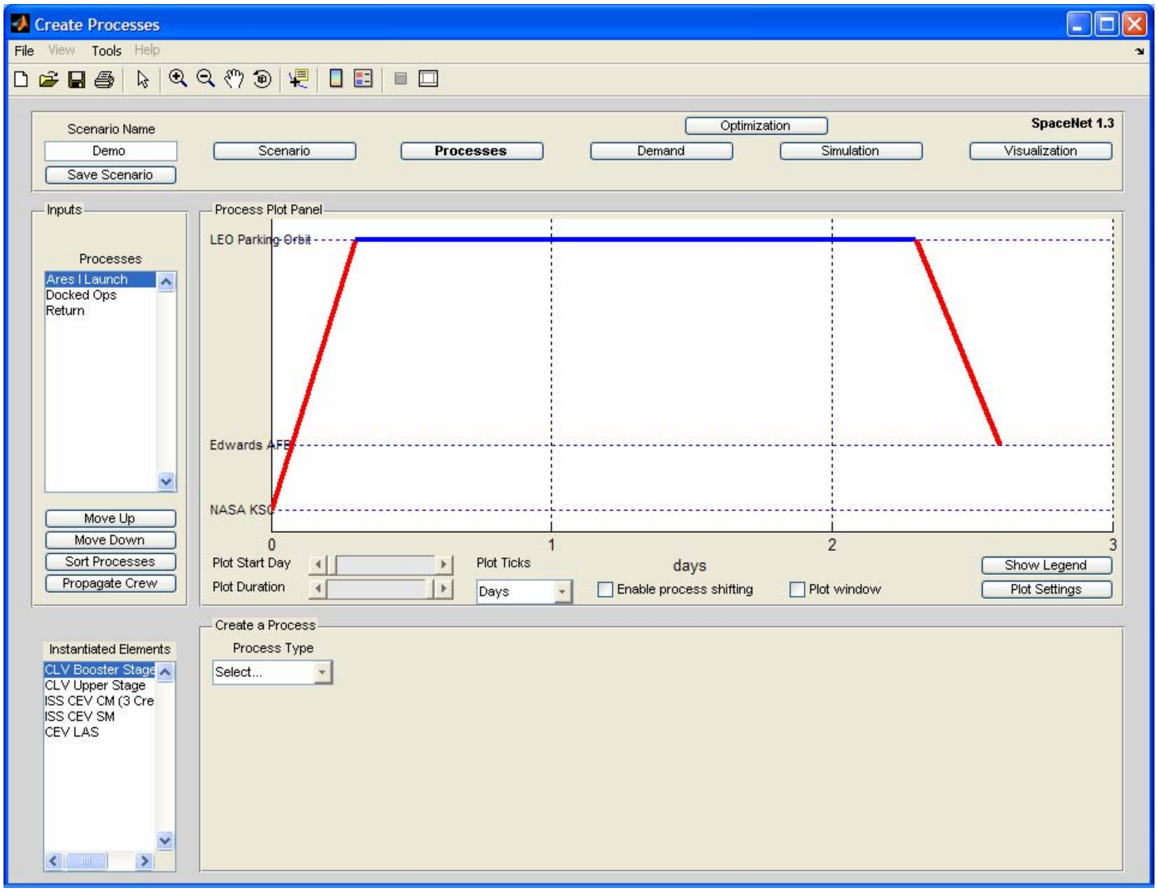
Any combination of history variable axes assignments and observed history variables can be plotted with the History Plotter. We encourage users to explore this very useful (and enjoyable) feature.

Chapter 4 Scenarios

Included with SpaceNet 1.3 are five completed scenarios meant to demonstrate the capabilities of SpaceNet. These scenarios are: *Demo_Ref, Apollo_17, ISS_Res_CEV, Lunar_Sortie* and *Lunar_Base*. Each scenario will be described in the following sections. This is followed by a critical discussion and comparison of the MOE results obtained for Apollo 17, the lunar sortie and the lunar base scenario.

Demo_Ref

The *Demo_Ref* scenario is the scenario that corresponds to the SpaceNet Quick Start (See Chapter 1). This scenario is included for comparison purposes after the user performs the SpaceNet Quick Start. This simple scenario portrays an Ares I launch from KSC to low Earth orbit (LEO), a two day stay in LEO and a landing at Edwards Air Force Base (AFB). The processes for the *Demo_Ref* scenario can be seen in Figure 71.



Scenario Name: Demo

Processes:
- Ares I Launch
- Docked Ops
- Return

Plot Ticks: Days

☐ Enable process shifting

☐ Plot window

Instantiated Elements:
- CLV Booster Stage
- CLV Upper Stage
- ISS CEV CM (3 Cre
- ISS CEV SM
- CEV LAS

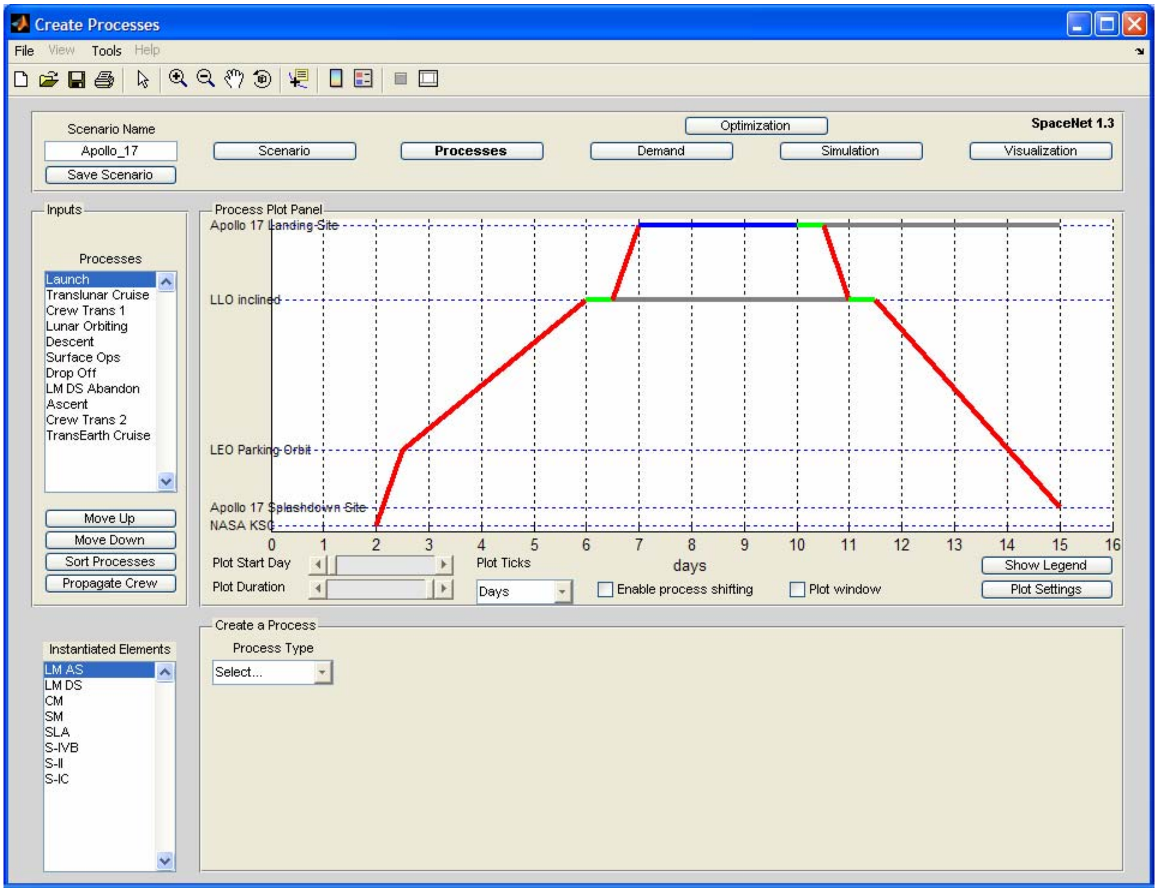
Process Type: Select...

Figure 71: Demo_Ref Processes Screen

Apollo_17

The *Apollo_17* scenario models the Apollo 17 mission from 12/07/72 to 12/20/72. This final lunar Apollo surface mission was chosen because it is the Apollo surface mission that achieved the longest lunar stay and carried the most cargo. From a logistics perspective it essentially reached the constraints of the capabilities of the Saturn V launch vehicle and of the Apollo LOR architecture and its associated vehicles (CM, SM, LM). Detailed reference data for this scenario is given in Appendix H.

This scenario consists of a Saturn V launch of three crewmembers to low lunar orbit (LLO) through LEO, a separation of the lunar module (LM) from the command and service module (CSM) with crew transfer, one crew member remaining in LLO while the other two descent to the lunar surface for a three day exploration stay, a transfer into the LM ascent stage, an ascent and rendezvous in LLO with the CSM and a return to Earth of all three crewmembers in the command module. The processes for the *Apollo_17* scenario are shown in Figure 72.



Scenario Name: Apollo_17

Plot Ticks: Days

☐ Enable process shifting

☐ Plot window

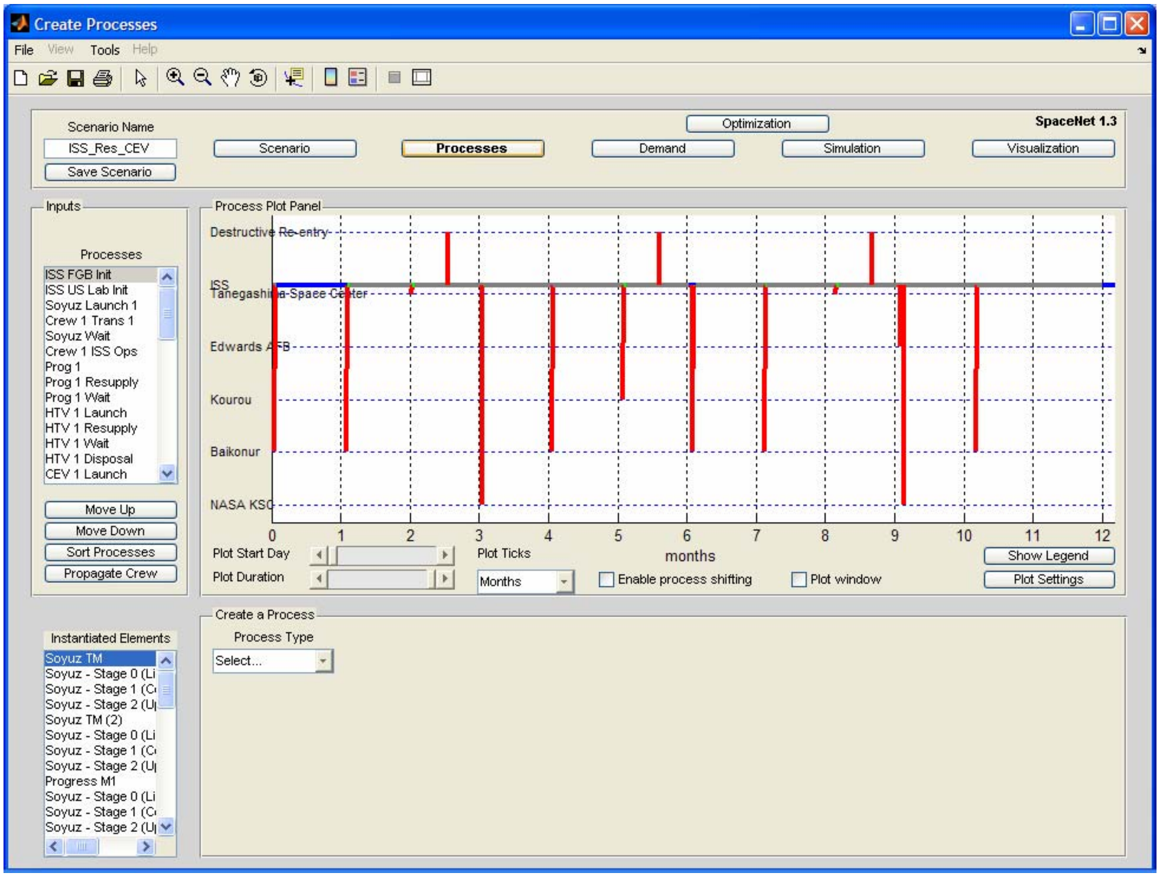
Process Type: Select...

Figure 72: Apollo_17 Processes Screen

ISS_Res_CEV

The *ISS_Res_CEV* scenario models a full calendar year of ISS traffic post shuttle retirement in 2010. For this scenario crew rotation and ISS resupply for a crew of six is achieved with a total of 11 flights consisting of two Soyuz flights, four Progress flights, one Ariane Transfer Vehicle (ATV) flight, two H-II Transfer Vehicle (HTV) flights and two Crew Exploration Vehicle (CEV) flights. For simplicity in this scenario each flight is spaced ~30 days apart with the first launch occurring on day one of the scenario. The processes for the *ISS_Res_CEV* scenario are shown in Figure 73. Due to the extended timeline each flight (transport process) appears as a nearly vertical red line.

This scenario demonstrates the capability of SpaceNet to model not just individual flights but entire campaigns over extended periods of time.


Scenario Name: ISS_Res_CEV

Plot Ticks: Months

☐ Enable process shifting

☐ Plot window

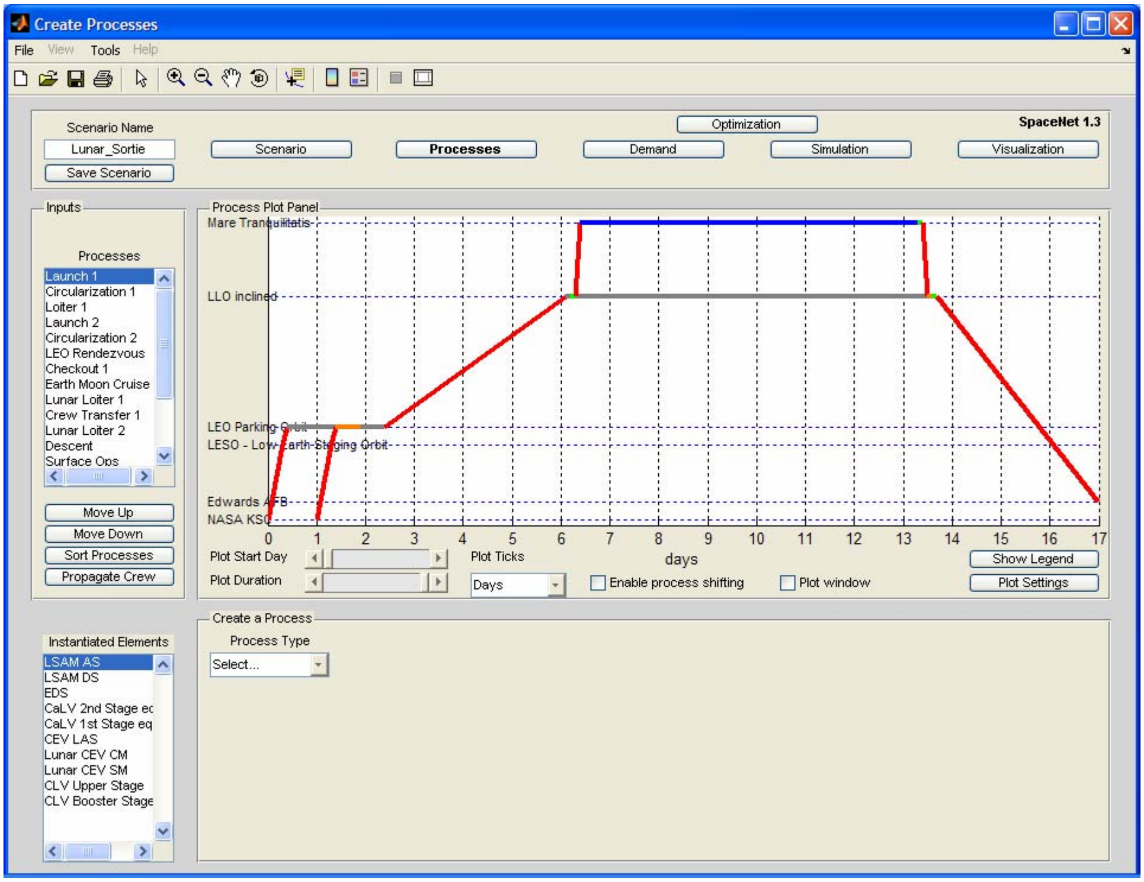
Process Type: Select...

Figure 73: ISS_Res_CEV Processes Screen

Lunar_Sortie

The *Lunar_Sortie* scenario (Figure 74) models a single lunar sortie mission as described in the Exploration Systems Architecture Study (ESAS) final report. This scenario models the 1.5 Earth orbit rendezvous-lunar orbit rendezvous (EOR-LOR) launch architecture with the unmanned launch of an Ares V cargo launch vehicle occurring on day zero followed by the manned Ares I launch on day one. The duration of the on-orbit loiter (gray horizontal bar separating the two launches = waiting process) can be varied.

After proximity operations in LEO on day 1 and 2, the Earth Departure Stage (EDS) propels the stack consisting of the CEV CM and SM as well as the LSAM through TLI into Earth-Moon Cruise. After LOI arrival into LLO all four crew transfer into the LSAM which descends to the lunar surface. Four crewmembers are involved in this scenario, all of whom spend seven days exploring the Moon at Mare Tranquilitatis. After their surface stay, all four crewmembers return to Earth in the LSAM-AS and then the CEV command module for a landing at Edwards Air Force Base. The process for the *Lunar_Sortie* scenario can be seen in Figure 74. There are three major differences between the Lunar_Sortie scenario and the Apollo_17 scenario: (i) two launches versus one, requiring an EOR, (ii) the LOI burn is performed by the LSAM-DS, not the SM and (iii) the CEV CM is uncrewed while waiting in LLO.



Scenario Name: Lunar_Sortie

Processes:
- Launch 1
- Circularization 1
- Loiter 1
- Launch 2
- Circularization 2
- LEO Rendezvous
- Checkout 1
- Earth Moon Cruise
- Lunar Loiter 1
- Crew Transfer 1
- Lunar Loiter 2
- Descent
- Surface Ops

Plot Ticks: Days

Instantiated Elements:
- LSAM AS
- LSAM DS
- EDS
- CaLV 2nd Stage eq
- CaLV 1st Stage eq
- CEV LAS
- Lunar CEV CM
- Lunar CEV SM
- CLV Upper Stage
- CLV Booster Stage

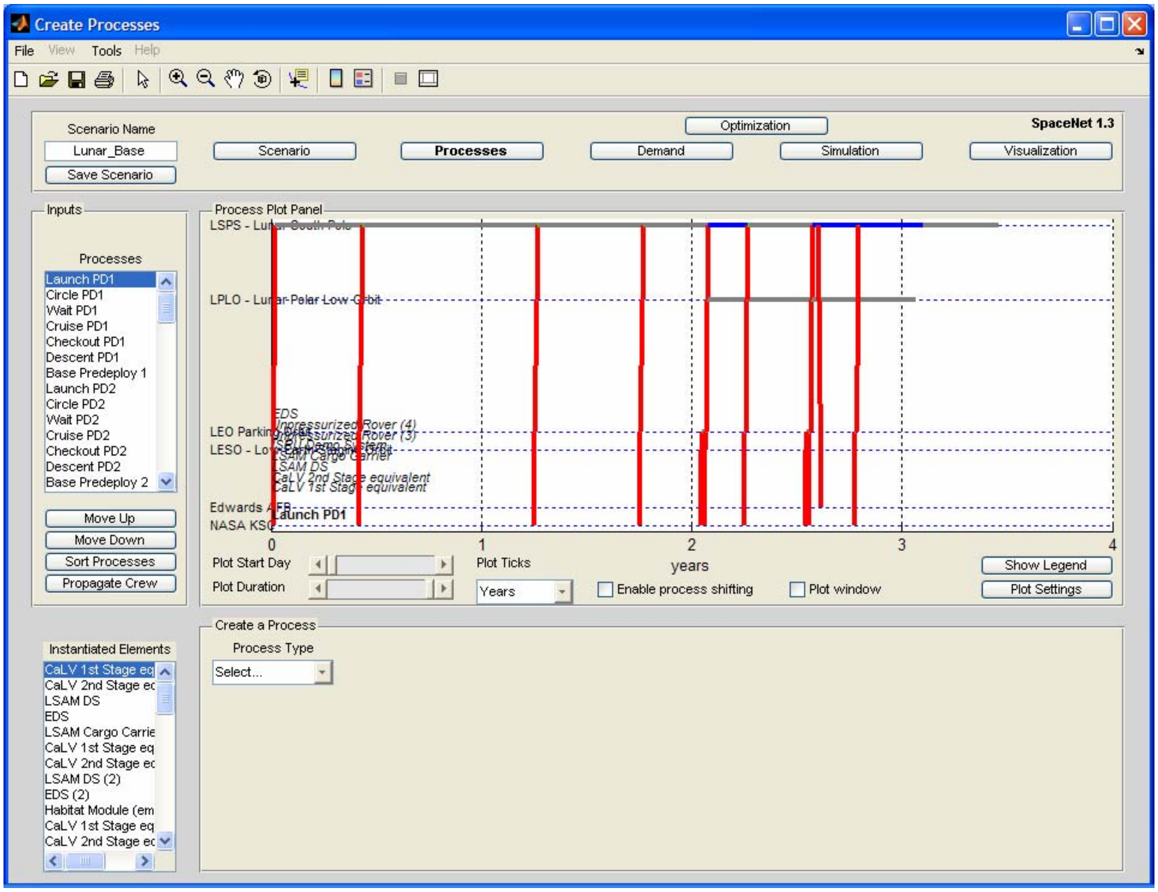
Process Type: Select...

Figure 71: Lunar_Sortie Processes Screen

Lunar_Base

The *Lunar_Base* scenario models the campaign consisting of the four year build-up of a lunar base as described in the Exploration Systems Architecture Study (ESAS) final report. This scenario consists of eight flights including four pre-deploy missions, two lunar base crew rotation missions and two logistics resupply missions. Each pre-deploy and logistic resupply mission consist of an unmanned launch of the Ares V cargo launch vehicle to deliver supplies to the Lunar South Pole Station (LSPS). Each crew rotation mission utilizes the 1.5 launch architecture with an Ares V launch followed by an Ares I launch and EOR. Each lunar base crew consists of four crewmembers living at the LSPS for six months. The scenario computes and simulates all demands for crew provisions, crew operations etc… during this period.

The processes for the *Lunar_Base* scenario are shown in Figure 75. Individual flights reaching the LSPS are shown as nearly vertical read lines due to the time compression of the x-axis. This scenario is the most complex and demonstrates the ability to model the logistics involved in a multi-year campaign to the lunar surface.



Scenario Name: Lunar_Base

Plot Ticks: Years

Process Type: Select...

Figure 75: Lunar_Base Processes Screen

Comparison of Scenarios

Interesting insights can be gained by comparing MOEs across scenarios. Table 8 shows a tabular comparison of MOE results obtained from the *Apollo_17* (baseline), *Lunar_Sortie* and *Lunar_Base* scenarios as described above. Note that the lunar sortie and base scenarios are inspired by the ESAS in terms of the transportation architecture but are not a direct reflection of the same underlying assumptions. The results are shown for illustration purposes only. The three scenarios shown in Table 8 simulate without errors, which means that they are feasible from a transportation (propulsive) as well as supply/demand perspective.

Table 8: Comparison of MOEs for SpaceNet Scenarios

MOE	**Apollo_17**	**Lunar_Sortie**	**Lunar_Base**
Crew Surface Days (CSD)	7	28	2294
Exploration Mass Delivered (EMD)	456	500	49,500
Exploration Capability (EC)	3,192	13,999	109,711,000
Relative Exploration Capability (REC)	1.0 (baseline)	3.113	3,565
Total Launch Mass (TLM)	2,927	4,124	28,229
Up-Mass Capacity Utilization (UCU)	0.82	0.40	0.435
Return-Mass Cap. Utilization (RCU)	0.80	0.786	1.0
Relative Scenario Cost (RSC)	1.0 (baseline)	1.29	106

The results show that the lunar sortie scenario achieves a factor of 4.4 improvement in terms of lunar exploration capability (EC) over Apollo 17 and once launch mass is taken into account a factor of 3.1 improvement in relative exploration capability (REC). This means that for every kg of mass launched from Earth's surface the *Lunar_Sortie* scenario could (theoretically) achieve 3.1 times more exploration than was accomplished in the *Apollo_17* scenario. Inspection of Table 8 shows us that this is primarily accomplished through landing a crew of 4 (rather than 2) and staying for 7 days (rather than 3.5). Whether or not this level of improvement will actually be achieved in the future once dry mass growth of elements and other real world effects have been taken into account remains to be seen. Table 8 also suggests that *Lunar_Sortie* has the potential for doing even better by increasing its up-mass capacity utilization (UCU) to a value closer to one.

The other important takeaway from Table 8 is that there is a large difference in terms of MOEs between a lunar sortie and the lunar base scenario. The lunar base scenario extends over 3 years (9 flights) rather than only 2 weeks (1 flight) and represents more than an order of magnitude increase in terms of most MOEs. Specifically the total exploration capability is a factor of 7837 larger for the lunar base scenario compared to the lunar sortie scenario while the total launch mass (TLM) is only greater by a factor of 6.84. How can this be?

The reason for this is that in the lunar base case a supply chain is space is being established and any exploration mass delivered in previous missions (flights) at the lunar outpost (lunar base) accumulates over time and serves to increase the exploration capability of subsequent crews. Thus, the benefit of returning to previously visited nodes to reuse exploration equipment is expressed by the sharp increase in both EC and REC[12] shown in Table 8 and more clearly in Figure 76.

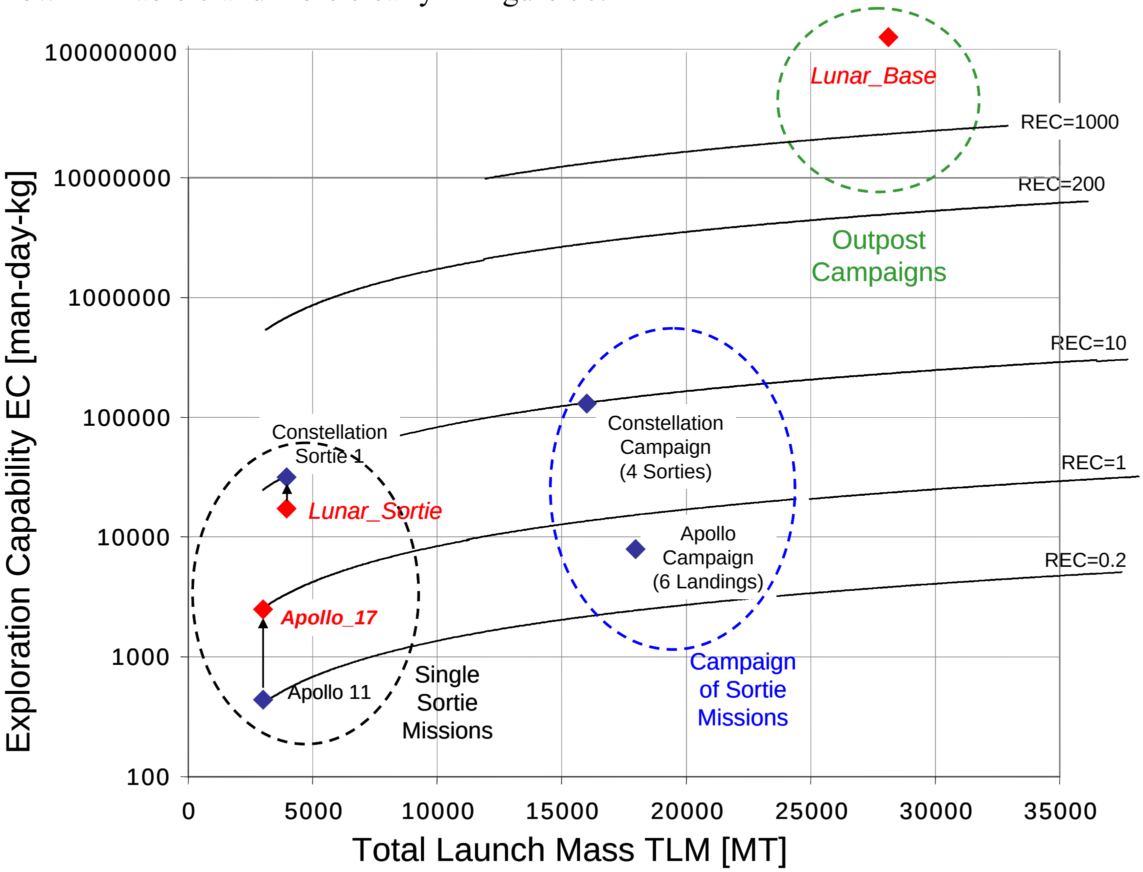


Figure 76: Space Logistics Tradespace for lunar campaigns

---

[12] One may argue that over extended periods of time the EC metric looses its value as a metric, since there would be diminishing scientific returns to be obtained by returning to the same exploration node over and over again. While this may be true, there are examples on Earth, e.g. the Haughton-Mars analogue base on Devon Island (see de Weck O.L., Simchi-Levi D., “Haughton-Mars Project Expedition 2005”, Final Report, NASA/TP-2006-214196, January 2006), where even after nearly a decade of exploration at the same node new scientific discoveries do occur. This, however, requires that the location of a lunar base be very carefully chosen in the first place.

Figure 76 shows the position of the three scenarios, *Apollo_17*, *Lunar_Sortie* and *Lunar_Base* (red) in the space logistics tradespace of lunar campaigns. By plotting exploration capability (EC) versus total launch mass (TLM) we can find the contours of constant relative exploration capability (REC). Any point along the same REC contour has the same space exploration logistics efficiency.

By definition the Apollo 17 scenario achieves an REC=1. It is interesting to note that Apollo 11 achieved an REC=0.2. Between Apollo 11 and 17 the cargo mass was gradually increased until the feasibility limit of the architecture was reached. This can be seen by the fact in Fig. 76 that their launch mass is nearly identical (see also Appendix H), but that their EC differs by a factor of 5. The Constellation lunar sortie represents an REC improvement between a factor of 3.1 (as in the *Lunar_Sortie* scenario) and a factor of 10 (Constellation Sortie 1), when fully loaded (UCU=1).

When conducting a campaign that is a set of individual sortie missions to different exploration nodes, the best one can hope for is to remain on the same constant REC contour associated with a single sortie mission. In the case of Apollo all seven lunar landing attempts (Apollo 11-17) where at different landing locations and because of the failure of Apollo 13 to reach its intended target and the fact that not all missions maximized UCU, the Apollo campaign level REC was somewhat less than 1.0. However, once the campaign level strategy includes returns to the same node along with uncrewed pre-positioning and resupply flights, significantly larger REC values can be achieved as in the case of the *Lunar_Base* scenario (upper right corner of Fig.76). It is in these cases that SpaceNet is especially useful in evaluating and comparing various supply chain strategies in space.

Additional Uses of SpaceNet

This section briefly highlights sample results for the types of analyses that can be supported by SpaceNet. The results are for illustrative purposes only.

Figure 77 shows a comparison of MOEs for four different launch strategies: the 1.5 launch baseline (a single launch of an Ares-V followed by a single launch of an Ares-I followed by EOR), a single launch strategy with unmodified CEV and LSAM elements, a single launch strategy where a performance equivalent to the 1.5 launch architecture is enforced and finally a 2 launch solution which would involve two launches of an Ares-V class launch vehicle with subsequent EOR. The results show that the single launch strategy with unmodified elements is highly undesirable because it significantly degrades REC. The 1-launch equivalency launch strategy appears to be interesting but would require a single Saturn-V class launch vehicle that is larger than the baseline Ares V design. Finally, the two launch architecture yields higher MOEs but is also the most expensive one in terms of launch mass and relative scenario cost. SpaceNet can be used to compare the exploration capability impact of different strategies, while ensuring their basic feasibility.

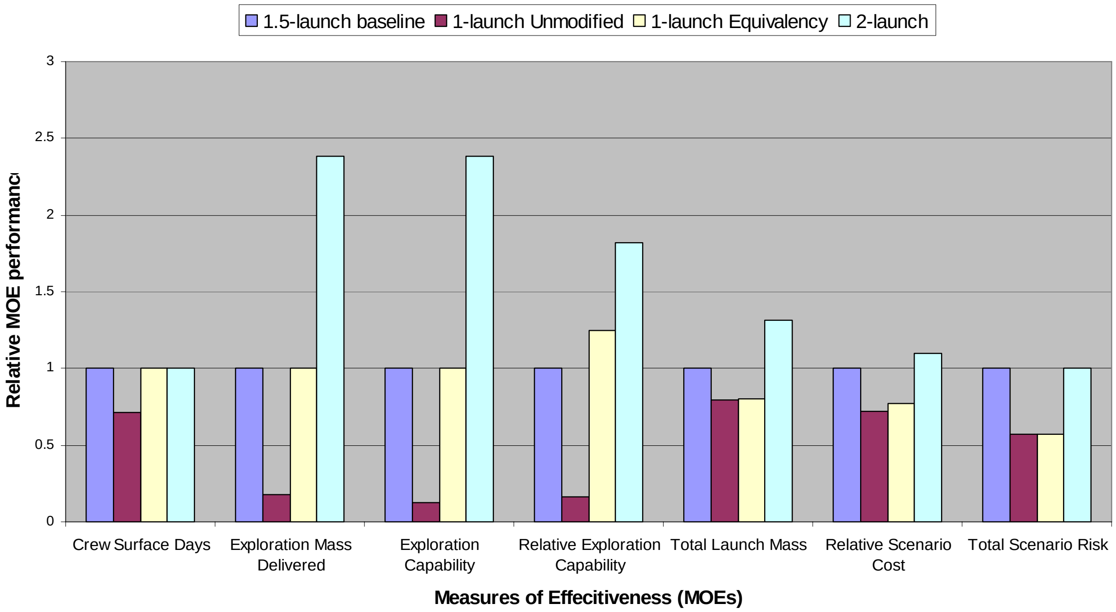


Figure 77: Comparison of different launch strategies

One of the uses of SpaceNet is to determine and fine-tune the "optimal" manifest for crewed space exploration missions. Figure 78 shows the breakdown of a notional lunar surface cargo manifest in terms of the SpaceNet functional classes of supply (CoS).

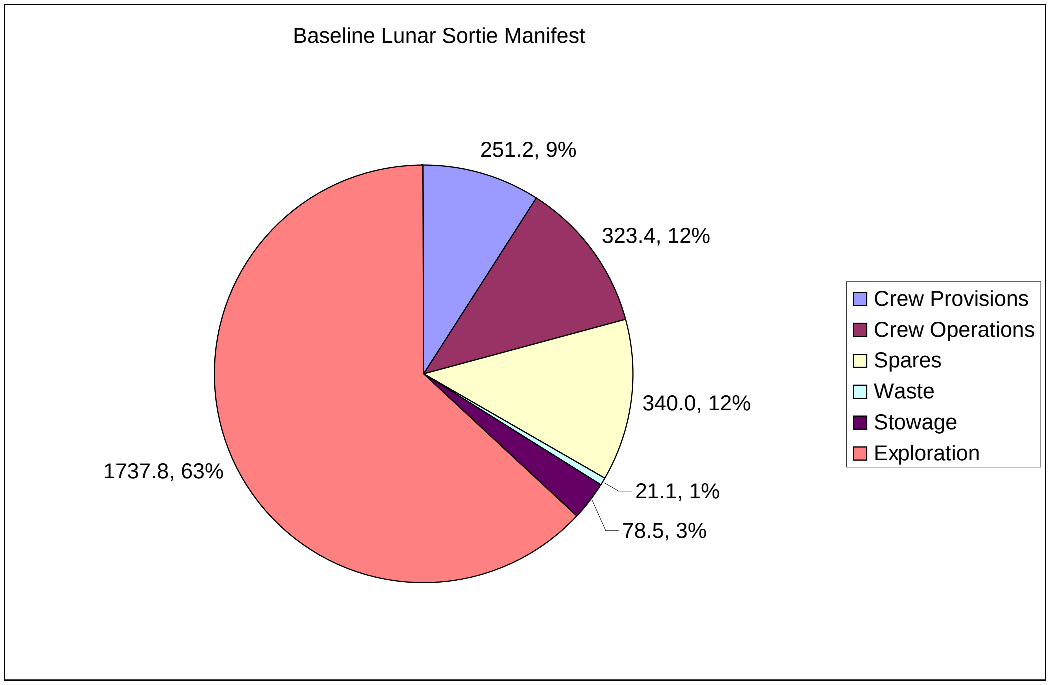


Figure 78: Lunar Cargo Surface manifest: 2,752 kg, crew =4, 7 surface days

The SpaceNet lunar sortie scenario corresponding to Fig. 78 is based on a total lunar surface cargo mass of 2,752 kg, a crew of 4 and surface duration of 7 days. The nominal process for arriving at such a manifest would be to first ensure survival of the crew and appropriate margins which determine the amount of crew provisions (252 kg) and crew operations equipment (323 kg) that would need to be brought along. Next, a target system availability would be chosen (e.g. 95% for LSAM system availability), which would determine the amount of spare parts and tools to be taken along (340 kg in this case). Finally the remainder of the available cargo capacity could be allocated for exploration items (COS 6, 1738 kg) to maximize the capacity utilization of the system[13].

Another important and frequent activity in space logistics (e.g. on the ISS program) is to trade one type of supply class against another, given a fixed transportation capacity. Figure 79 illustrates the effect of extending the lunar surface duration (effectively lengthening the lunar exploration process shown in Figure 26). As surface duration is extended more consumables are needed (for a given ECLSS loop closure rate or consumables recovery rate). For a fixed lunar surface cargo capacity this extra consumables mass (magenta colored wedge in Fig.79) has to be taken away from exploration mass (yellow colored wedge in Fig. 79). There is a theoretical lunar surface duration (blue parabola) that will maximize exploration capability, while maintaining feasibility. Fig. 79 shows that this optimum occurs at 21 days with a crew of 4.[14]

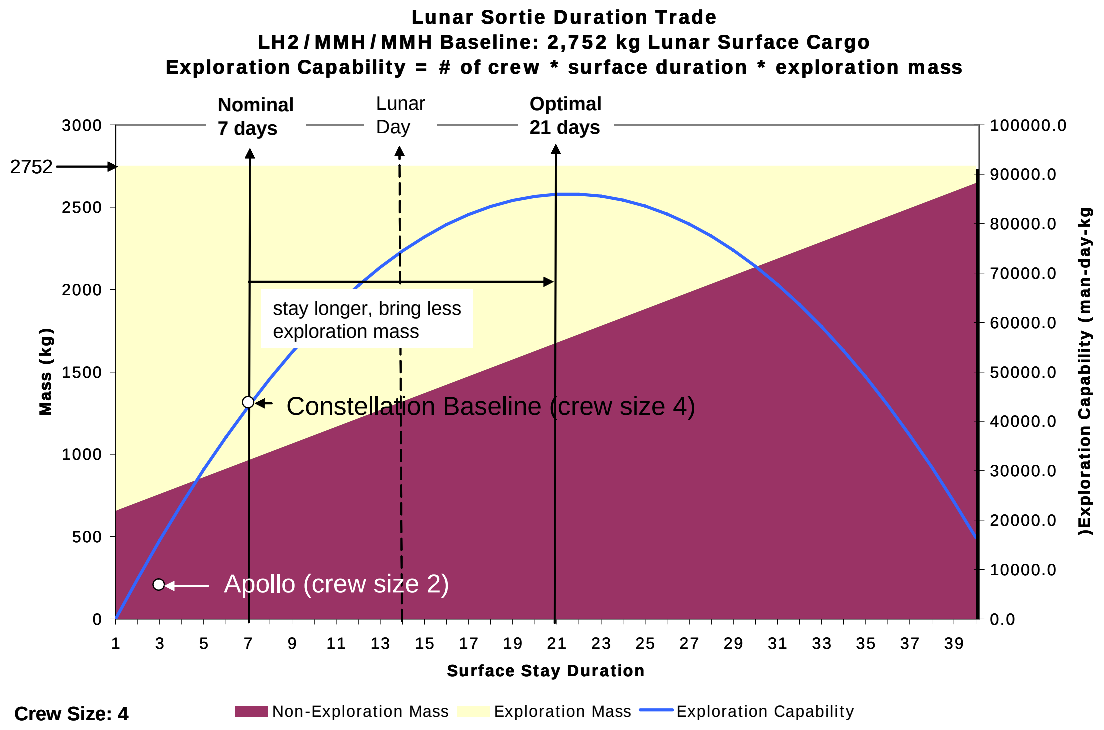


Figure 79: Effect of lunar surface duration on Lunar Cargo Surface manifest: 2,752 kg

Another useful feature of SpaceNet in conjunction with the Excel Visualization capability is the ability to track the flow of elements and supply classes across nodes and time. Figure 80 shows the location, consumption and flow of consumables throughout a

---

[13] Waste and stowage are minor contributors to the manifest.

[14] Other factors such as propulsive anytime return may be the limiting factor, depending on lunar landing location and time.

scenario. The x-axis corresponds to time, the y-axis indicates nodal location and the colors correspond to the elements in which consumables are contained for the Apollo 17 scenario. In this scenario crew consumables are only held in the CM and the LM AS.

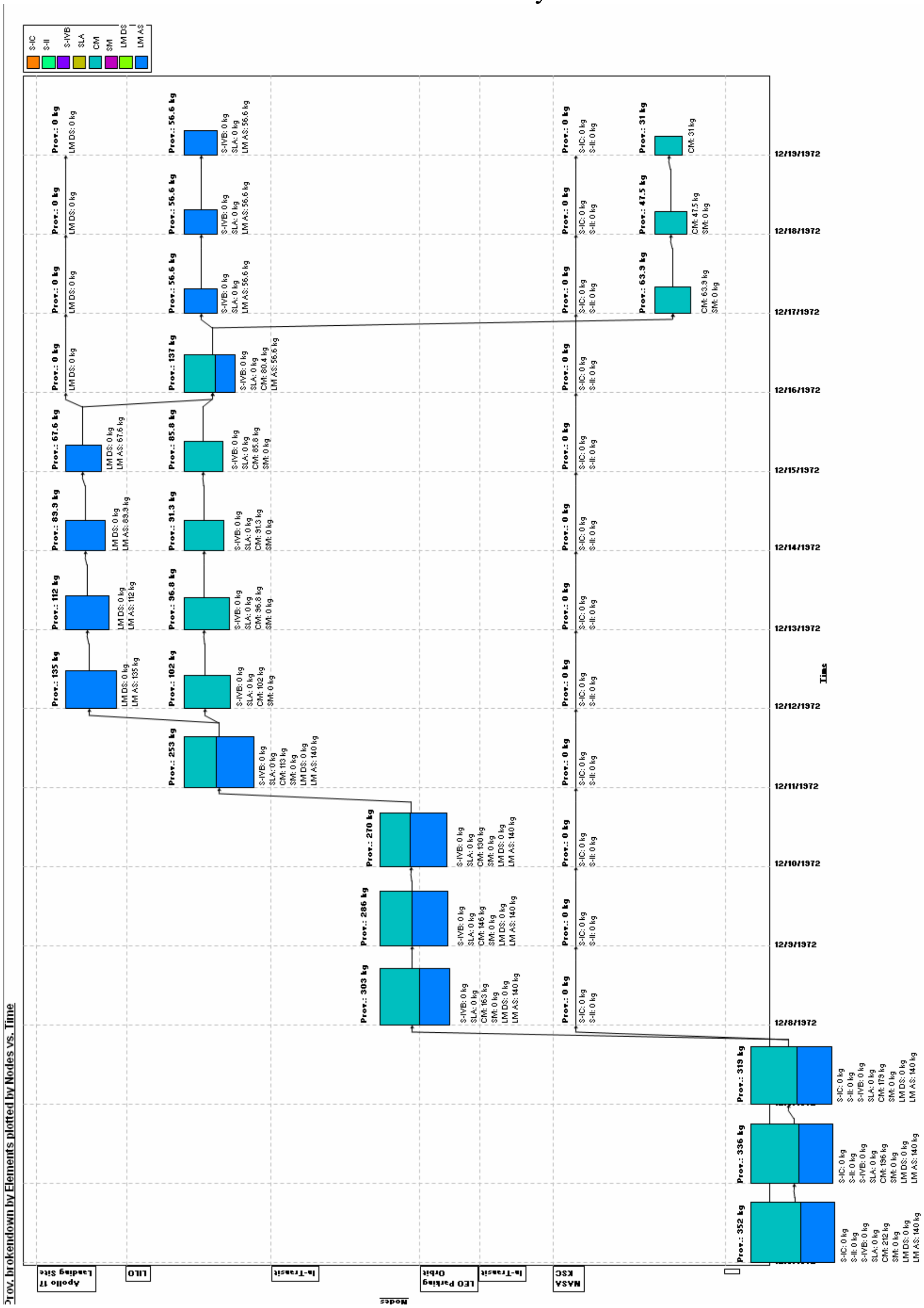


Figure 80: Tracking of crew consumables in the Apollo 17 scenario

SpaceNet 1.3 supports such types of analysis by simulating scenarios in an integrated fashion with a view to maximizing exploration capability, while ensuring feasibility.

(this page intentionally left blank)

Appendix A: SpaceNet Glossary

Acronyms

AFB	Air Force Base
AS	Ascent Stage
BKO	Baikonur
CCART	Cargo Category Allocation Rate Table
CEV	Crew Exploration Vehicle
CM	Command Module
COS	Class of Supply
CPLEX	Large-scale Mathematical Programming (Optimization) Software
CSD	Crew Surface Days
CTB	Cargo Transfer Bag
CWC	Contingency Water Container
DB	Database
DV	Delta Velocity
EC	Exploration Capability
EDS	Earth Departure Stage
EEO	Earth Equatorial Orbit
EMD	Exploration Mass Delivered
EML1	Earth-Moon Lagrange Point 1
EPO	Earth Polar Orbit
ESL2	Earth-Sun Lagrange Point 2
EVA	Extra-Vehicular Activity
GUI	Graphical User Interface
ID	Identification
ISP	Specific Impulse
ISRU	In-Situ Resource Utilization
ISS	International Space Station
JSC	Johnson Space Center
KOU	Kourou
KSC	Kennedy Space Center
LAS	Launch Abort System
LEO	Low Earth Orbit
LLO	Low Lunar Orbit
LOI	Lunar Orbit Insertion
LOC	Loss of Crew
LOM	Loss of Mission
LOV	Loss of Vehicle
LPO	Lunar Polar Orbit
MEO	Mars Equatorial Orbit
MOE	Measures of Effectiveness
MPO	Mars Polar Orbit

NASA	National Aeronautics and Space Administration
ORU	Orbital Replacement Unit
PHO	Phobos
REC	Relative Exploration Capability
RCU	Return-Mass Capacity Utilization
RSC	Relative Scenario Cost
TSC	Tanegashima
SM	Service Module
TLI	Trans-Lunar Injection
TLM	Total Launch Mass
TSR	Total Scenario Risk
UCU	Up-Mass Capacity Utilization
USA	United Space Alliance LLC
VDB	Vandenberg

Glossary

(Note: Capitalized nouns typically refer to terms that are defined in the glossary)

Arc - The equivalent of a line or an edge in graph theory. Merely indicates the presence of a connection between any two nodes but not the nature of that connection (directionality, delta-V, etc.). See Trajectory.

Bat Visualization – A particular type of visualization for space missions over time where the x-axis is time, the y-axis shows the position of nodes and vehicles (elements) are shown at different points in time. The word “bat” is used because typically the destination node (Moon, Mars,…) is at the top of the figure and vehicles that are shown to rest on the surface appear to be hanging upside down, similar to bats at rest.

Building Block - A term representing a Campaign or Sortie. Building Blocks facilitate the build-up of a Scenario by nesting Missions within Campaigns or Processes within Missions. The grouping of processes into building blocks is not implemented in SpaceNet 1.3 but will be implemented in SpaceNet 2.

Buy-List – An itemized list of supply items which has to be procured on Earth and provided to the spaceport for storage (inventory) or manifesting on a set of launched elements. Specific examples of buy-lists are sets of orbital replacement units (ORU) that have to be provided as spares.

Campaign - A set of multiple missions spread out over time. Generally includes situations in which missions (such as sorties, pre-deployment flights, resupply flights, crew rotation flights) are launched at greatly separated points in time and can follow different paths and/or trajectories to arrive at the same location for some shared purpose. Campaigns are used when the destination demand for an exploration node is greater than what can be provided by a single mission. Campaigns typically last more than one Earth year, typically on the order of 5-20 years. An example of a campaign was the Apollo lunar campaign (Apollo 11-17), see Figure 76.

Cargo Manifest – An itemized list of supply items that are placed in an instantiated Element for launch.

Classes of Supply - This represents categories of items moving through the supply chain. SpaceNet uses *function-based* classes of supply. These functions are essential to supporting crewed and uncrewed spaceflight operations. Classes of supply are decomposed into sub-classes of supply. There are 10 classes of supply:

1. Propellants and Fuels
2. Crew Provisions
3. Crew Operations
4. Maintenance and Upkeep
5. Stowage and Restraint
6. Exploration and Research
7. Waste and Disposal
8. Habitation and Infrastructure
9. Transportation and Carriers (including all space vehicle Elements)
10. Miscellaneous

An iconic representation of the SpaceNet classes of supply is shown in Figure A-1 below.

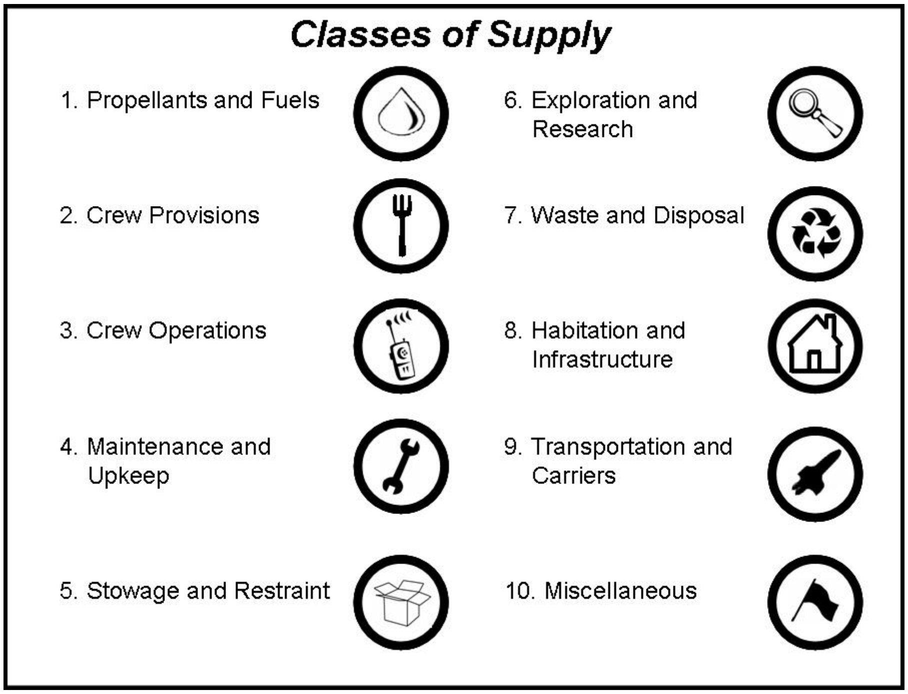


Figure A-1: Classes of Supply Icons

A more detailed view, including sub-classes of supply is shown in Table A-1 below.

Table A-1: COS and sub-classes of supply

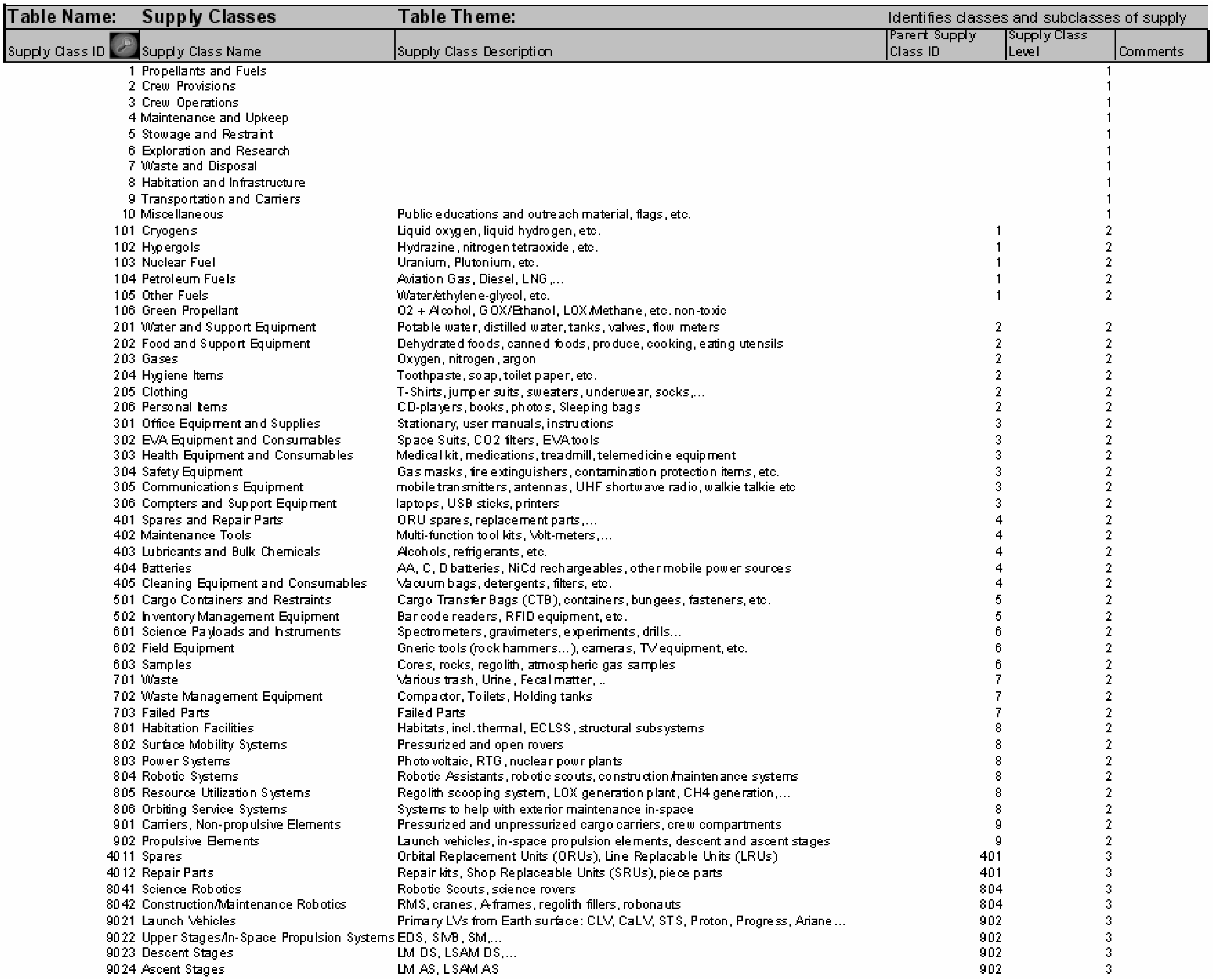


Table Name:	Supply Classes	Table Theme:	Identifies classes and subclasses of supply		
Supply Class ID	Supply Class Name	Supply Class Description	Parent Supply Class ID	Supply Class Level	Comments
1	Propellants and Fuels			1	
2	Crew Provisions			1	
3	Crew Operations			1	
4	Maintenance and Upkeep			1	
5	Stowage and Restraint			1	
6	Exploration and Research			1	
7	Waste and Disposal			1	
8	Habitation and Infrastructure			1	
9	Transportation and Carriers			1	
10	Miscellaneous	Public educations and outreach material, flags, etc.		1	
101	Cryogens	Liquid oxygen, liquid hydrogen, etc.	1	2	
102	Hypergols	Hydrazine, nitrogen tetraoxide, etc.	1	2	
103	Nuclear Fuel	Uranium, Plutonium, etc.	1	2	
104	Petroleum Fuels	Aviation Gas, Diesel, LNG,...	1	2	
105	Other Fuels	Water/ethylene-glycol, etc.	1	2	
106	Green Propellant	O2 + Alcohol, GOX/Ethanol, LOX/Methane, etc. non-toxic			
201	Water and Support Equipment	Potable water, distilled water, tanks, valves, flow meters	2	2	
202	Food and Support Equipment	Dehydrated foods, canned foods, produce, cooking, eating utensils	2	2	
203	Gases	Oxygen, nitrogen, argon	2	2	
204	Hygiene Items	Toothpaste, soap, toilet paper, etc.	2	2	
205	Clothing	T-Shirts, jumper suits, sweaters, underwear, socks,...	2	2	
206	Personal Items	CD-players, books, photos, Sleeping bags	2	2	
301	Office Equipment and Supplies	Stationary, user manuals, instructions	3	2	
302	EVA Equipment and Consumables	Space Suits, CO2 filters, EVA tools	3	2	
303	Health Equipment and Consumables	Medical kit, medications, treadmill, telemedicine equipment	3	2	
304	Safety Equipment	Gas masks, fire extinguishers, contamination protection items, etc.	3	2	
305	Communications Equipment	mobile transmitters, antennas, UHF shortwave radio, walkie talkie etc	3	2	
306	Compters and Support Equipment	laptops, USB sticks, printers	3	2	
401	Spares and Repair Parts	ORU spares, replacement parts,...	4	2	
402	Maintenance Tools	Multi-function tool kits, Volt-meters,...	4	2	
403	Lubricants and Bulk Chemicals	Alcohols, refrigerants, etc.	4	2	
404	Batteries	AA, C, D batteries, NiCd rechargeables, other mobile power sources	4	2	
405	Cleaning Equipment and Consumables	Vacuum bags, detergents, filters, etc.	4	2	
501	Cargo Containers and Restraints	Cargo Transfer Bags (CTB), containers, bungees, fasteners, etc.	5	2	
502	Inventory Management Equipment	Bar code readers, RFID equipment, etc.	5	2	
601	Science Payloads and Instruments	Spectrometers, gravimeters, experiments, drills...	6	2	
602	Field Equipment	Gneric tools (rock hammers...), cameras, TV equipment, etc.	6	2	
603	Samples	Cores, rocks, regolith, atmospheric gas samples	6	2	
701	Waste	Various trash, Urine, Fecal matter, ..	7	2	
702	Waste Management Equipment	Compactor, Toilets, Holding tanks	7	2	
703	Failed Parts	Failed Parts	7	2	
801	Habitation Facilities	Habitats, incl. thermal, ECLSS, structural subsystems	8	2	
802	Surface Mobility Systems	Pressurized and open rovers	8	2	
803	Power Systems	Photovoltaic, RTG, nuclear powr plants	8	2	
804	Robotic Systems	Robotic Assistants, robotic scouts, construction/maintenance systems	8	2	
805	Resource Utilization Systems	Regolith scooping system, LOX generation plant, CH4 generation,...	8	2	
806	Orbiting Service Systems	Systems to help with exterior maintenance in-space	8	2	
901	Carriers, Non-propulsive Elements	Pressurized and unpressurized cargo carriers, crew compartments	9	2	
902	Propulsive Elements	Launch vehicles, in-space propulsion elements, descent and ascent stages	9	2	
4011	Spares	Orbital Replacement Units (ORUs), Line Replacable Units (LRUs)	401	3	
4012	Repair Parts	Repair kits, Shop Replaceable Units (SRUs), piece parts	401	3	
8041	Science Robotics	Robotic Scouts, science rovers	804	3	
8042	Construction/Maintenance Robotics	RMS, cranes, A-frames, regolith fillers, robonauts	804	3	
9021	Launch Vehicles	Primary LVs from Earth surface: CLV, CaLV, STS, Proton, Progress, Ariane...	902	3	
9022	Upper Stages/In-Space Propulsion Systems	EDS, SMB, SM,...	902	3	
9023	Descent Stages	LM DS, LSAM DS,...	902	3	
9024	Ascent Stages	LM AS, LSAM AS	902	3	

Demand - mass, and/or volume, and/or number of units of a given supply item required for a given entity (crew, class of supply) over some segment of a scenario (process, mission, campaign).

Delivery Demand - Total amount of all items to be delivered to a node for a mission or campaign.

Demand Model - a system of equations and data characterizing the supply requirements for a specific entity (crew or class of supply) or process, per unit time or per event. Demand Models are aggregated for an entire Scenario --> Processes --> Elements/Crew.

They include the following data where applicable.

1. Mean Time to Repair rates, where applicable, for sparing.
2. Mean Time between Failure rates, where applicable, for sparing.
3. Demand Rates for supply items (not including ISRU, Reclamation, or Production)
4. ISRU Rates for supply items that may be extracted or processed from the environment at the exploration site (e.g., fuel, gasses)
5. Local Production Rates for supply items that may be produced at the exploration site (e.g., food)
6. Reclamation Rates for supply items that may be reclaimed/recycled during the mission
7. Service Levels for each supply item.

(Note that ISRU (local production outside Earth) and reclamation are not implemented in SpaceNet 1.3. The Demand Models in SpaceNet 1.3 are explained in Appendix C.

Depot – A node or element(s) dedicated to holding inventory of supply items for future use.

Element – An indivisible functional unit that has either propulsive capability, payload holding capability (for agents and/or cargo) or some combination of the two. Elements can temporarily dock to form stacks. Most space vehicles and launch stages are modeled as elements, but the concept is broader in that elements can also comprise larger end items such as habitat modules, pressurized rovers, power stations and so forth.

Exploration Plan - An exploration plan is a system of data characterizing the locations (nodes), durations, agents and method of exploration for a scenario. The exploration plan cannot be automatically generated by SpaceNet, it has to be entered by the user. It comprises the following data:

1. Exploration Nodes (can be surface nodes, orbit nodes or Lagrangian nodes)
2. Exploration start date
3. Duration
4. Presence Profile (number of crew present over time)
5. Site Development Plan (number and type of elements to be delivered)

In SpaceNet 1.3 the exploration plan is essentially defined as the sum of all the manually defined exploration processes.

Fleet - The set of elements (both propulsive and non-propulsive) instantiated for a given scenario.

Launch Schedule - A system of data characterizing what element stacks will be launched, when, from what spaceport (Earth surface node) and with what cargo manifests to meet exploration mission or campaign objectives. Includes the following data:

1. Elements launched, incl. stack configurations
2. Launch schedule
3. Cargo manifest for each element

Measures of Effectiveness – Quantifiable criteria by which a scenario can be judged. SpaceNet 1.3 computes a total of 9 measures of effectiveness (see Appendix D for details). MOEs are similar to Figures of Merit (FOM) or Key Performance Indicators (KPI).

Mission - Set of related processes that achieve a defined mission objective. Typical instances of missions are sorties, pre-deployment flights, resupply flights, crew rotation flights and so forth.

Network - A set of nodes and related arcs representing the origin nodes, destination nodes, return nodes and all allowed way-point (intermediate) nodes. Origin(s), destination(s), and return(s) must be specified by the user. Way-point(s) may be specified by the user or suggested by optimization.

Node - A stable point in space to include Orbits, Lagrange Points, and points on the surface of a physical body in space. The presence of a node in a scenario does not automatically imply that it will be used.

Path - A directional sequence of arcs from an origin node to a destination node or a destination node to a return node. A path may or may not include waypoint nodes. A path can be expressed in the static network, as an ordered sequence of node ID numbers (1001, 1501, 2502,…) or as a set of doublets expressing the path in the Time-Expanded Network ((1001,1), (1001,2), (1501,3), (2502,8)….), whereby the first number refers to the node ID and the second number refers to the time period.

Presence Profile - This is a profile that describes the exploration presence at an exploration node over time. It includes the following data:

1. Frequency of visit - With what frequency do agents (crews/robots) arrive at the site
2. Size of Crew - How many agents arrive on each crewed sortie
3. Nominal Duration of Stay

SpaceNet 1.3 does not provide a separate way of defining the presence profile for a scenario. Crew presence is defined directly as individual exploration processes are specified.

Process - The fundamental state changes defining a scenario. SpaceNet contains five basic process types from which all scenarios can be constructed:

1. Transporting – changing the location of an instantiated element from one node to another
2. Waiting – remaining at a node for one time period or longer
3. Transferring – moving crew or a supply item from one element to another co-located element
4. Proximity Operating – rendezvous/docking, undocking/separate, or transposition of elements
5. Exploring – visiting a node for a certain number of time periods to gain new knowledge about that node

Reorder Point - That point at which time a stock replenishment requisition would be submitted to maintain the predetermined or calculated inventory objective to within predetermined service-levels. The sum of the safety level of supply plus the level for order and shipping time equals the reorder point. SpaceNet 1.3 does not calculate the Reorder point for various supplies but future versions may.

Return Delivery Demand - This specifies what entities will be delivered to a return node. This includes the crew, elements, and cargo such as science return (e.g. rock and soil samples).

Scenario - A system of data describing an exploration plan (set of exploration processes), the transportation architecture to support of that plan, the sequence of processes necessary to support the plan, and the demand profiles for classes of supply over time. Scenarios will contain ownership and time of creation information, simulation history, and resulting metrics (MOEs). A scenario is subject to the following constraints, based on SpaceNet 1.3 limitations:

1. The time horizon may be from 0.1 Earth days to 30 Earth years
2. The network may have 2 to 50 Nodes.
3. The fleet may have 1 to 100 Elements (stacks will be counted as the number of elements in the stack)

Site Development Plan - An optional system of data characterizing how exploration presence at a given node will change over time. In effect, will nodal presence increase or decrease, at what rates and involving what supply items/elements/crew? It includes the following data:

1. Initial delivery manifest (what shall be delivered to the exploration site)
2. Rates of change of crewed presence
3. Rates of change of element presence (e.g., number of rovers, habitats)
4. Rates of change of equipment presence (e.g., non-crew driven site development)
5. Items not delivered on first visit but planned for later visits.

The definition of a site development plan is not explicitly supported in SpaceNet 1.3. It is possible, however, to manually define the delivery and drop-off of major end items as elements (habitats, pressurized rovers, power plants…) in conjunction with exploration processes.

Sortie – A special type of mission which typically involves the short term exploration of a single node, typically in orbit or on the surface of a body other than Earth, followed by a return to Earth. A sortie mission does not typically involve the buildup of infrastructure at a node.

Space Travel Lead Time - Generally, the amount of time between the placing of an order (for resupply) and the receipt of the supply items ordered at the node where the resupply order was placed. Specific to the SpaceNet context, however, Lead Time is the amount of time between the start of launch countdown and final delivery of the supply items to the intended destination (i.e.: net travel time).

Stack – An ordered set of elements (propulsive and non-propulsive) associated with a particular process. From this it is possible to determine (to a first order approximation) the burn sequence and staging order for transporting processes and feasibility of rendezvous and docking for proximity operating processes.

Surface Node - A node on the surface of some physical body in space (e.g. Earth, Moon, Mars, asteroids, etc.). A surface node is uniquely defined by the latitude and longitude in the local coordinate system of the physical body.

Time-Expanded Network – An acyclic network of nodes and arcs, whereby arcs connecting the same physical (static) nodes at different times are referred to as waiting arcs and arcs connecting different physical nodes at different times are referred to as transport arcs. Arcs connecting different physical nodes at the same time are illegal. Time-expanded networks are the key to handling time-varying (type 3) trajectories in SpaceNet.

Trade-Study Mode - A simulation of multiple instances of different scenarios in a single simulation session to compare suitability to meet some user defined objective. Output is arranged to facilitate easy comparison between different scenarios.

Trajectory - The path in space followed by an element or stack of elements to travel from one node to another. It is described mathematically by the geometry of the path or as the position of the object over time in a defined orbit space. The geometric description of this path allows computation of the delta-V (change in potential and kinetic energy) of the element or stack of elements and the required time of flight. For any given transport arc in the time-expanded network, there may be one or many trajectories. Trajectories using any known or theoretically viable form of propulsion may be used. SpaceNet 1.3 distinguishes between three types of trajectories:

Type 1: time-invariant, modeled as a single delta-V burn (launch, reentry)
Type 2: time-invariant, modeled as two delta-V burns, burn 1 = departure burn, burn 2 = arrival burn, can have tradeoff between time-of-flight (TOF) and delta-V,

typically used to model orbital transfers around a single body or low-fidelity modeling of Earth-Moon transport

Type 3: time-varying trajectories, depend on absolute time (e.g. for Mars)

Transportation Architecture - The system consisting of Fleet (elements and stacks), Network (nodes and arcs) and Building Blocks (processes) that determines both transportation and inventory holding capacity of a space based supply chain system (Scenario).

(this page intentionally left blank)

Appendix B: Data Tables

This section summarizes the key data tables that are used in the SpaceNet 1.3 database.

Using the Database

The database for SpaceNet 1.3 resides in Excel with each table on a separate sheet. The Excel file is located in the …\spacenet_1.3\database directory. While the user may add password protection to this file, this causes SpaceNet 1.3 to request the password for each table that it attempts to read. Consequently, adding password protection is not recommended.

All additions and/or changes to the data in the Excel file must be performed before SpaceNet is opened, since SpaceNet reads all the data tables into memory at the outset. The database is structured in relational form, even though Excel does not perform relational database management functions.[15] The table below shows the principal data tables in SpaceNet 1.3.

Table B-1: Principal Data Tables

Table Name	**Table Theme**	**Primary Use**
Supply Classes	Identifies classes and subclasses of supply	Establishes classes of supply item types for modeling/simulation
Physical Nodes	Identifies a pre-defined set of static nodes from which a scenario can be constructed	Establishes feasible network
Astro	Identifies a set of feasible arcs (trajectories) and associated attributes based on astrodynamics	Establishes feasible network
Supply Item Type	Identifies common attributes of a supply item type	Manifesting
Element Type	Identifies common attributes of an element type	Establishes capabilities for processes
Stacks	Identifies composition of stack types	Optimization
Crew Provisions	Captures predicted usage rates	Crew provisions demand
Crew Operations	Captures predicted usage rates	Crew operations demand
Parts: Common	Identifies common attributes for ORUs/SRUs	Spares demand
Parts: Application-Specific	Identifies attributes for ORUs dependent on a particular application or installed location within an element type	Spares demand
Maintenance Tasks	Identifies all maintenance tasks and frequency of scheduled maintenance tasks for each ORU application	Maintenance demand
Maintenance	Identifies resources types, quantities, and times needed	Maintenance demand

---

[15] The tables have been made human-readable and are compatible with existing ISS data-tables in MESSOC.

Table Name	Table Theme	Primary Use
Resources	for each maintenance task	
Task Mode Type	Identifies attributes of ORU/SRU maintenance task modes	Maintenance demand
Part Type	Identifies attributes of ORU/SRU reliability classes, including K-factors	Spares demand
Partners	Identifies specific international partners and responsible governmental organizations	General information
Suppliers	Identifies specific supply item suppliers, including OEMs	General information
Unit Type	Identifies types of units used	General information

A data dictionary for each principal data table is provided in Appendix G. Other tables are used in conjunction with the principal ones to show status, types, levels, etc. that are not intended to be changed or supplemented with new rows. These data tables typically have a simple structure consisting of the tuple: ID, Name, Description.

Sample Entity-Relationship Diagrams

The following figures show examples of entity relationships in the SpaceNet 1.3 integrated database. Figure B-1 shows the relationships between nodes and allowed astrodynamics arcs (trajectories).

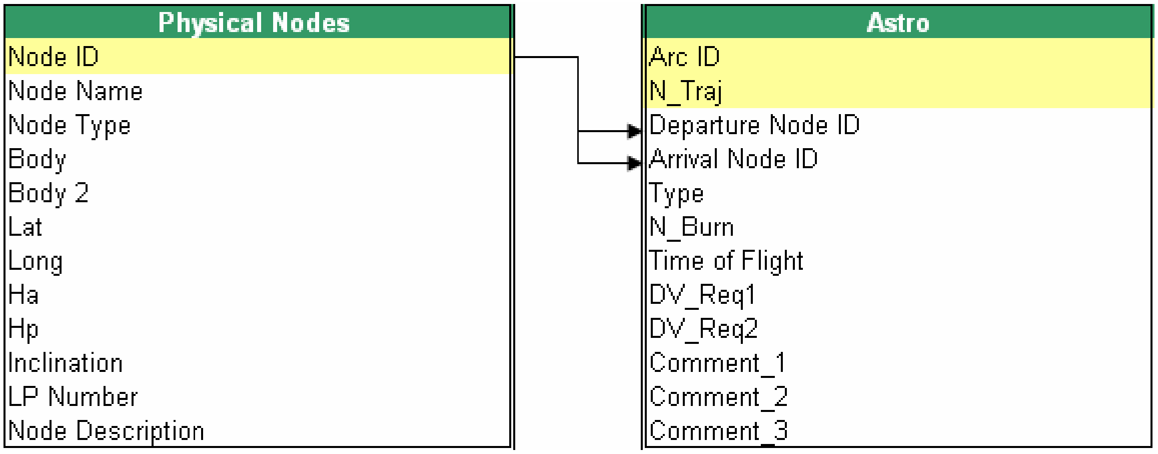
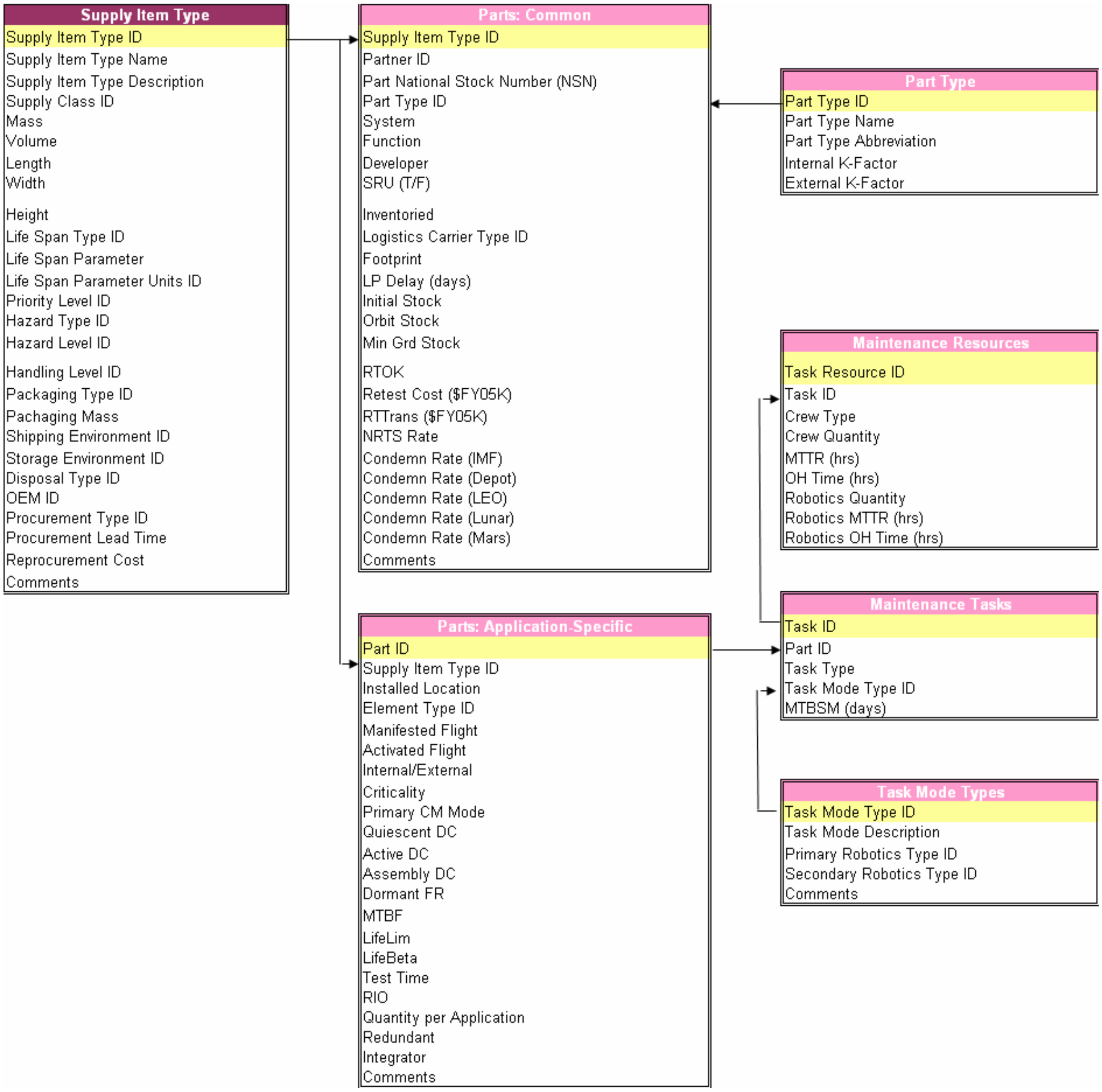


Figure B-1: E-R Diagram for Nodes and Arcs

Figure B-2 shows the relationships among the tables that are used to compute spares and maintenance demand.



Supply Item Type
Supply Item Type ID
Supply Item Type Name
Supply Item Type Description
Supply Class ID
Mass
Volume
Length
Width
Height
Life Span Type ID
Life Span Parameter
Life Span Parameter Units ID
Priority Level ID
Hazard Type ID
Hazard Level ID
Handling Level ID
Packaging Type ID
Pachaging Mass
Shipping Environment ID
Storage Environment ID
Disposal Type ID
OEM ID
Procurement Type ID
Procurement Lead Time
Reprocurement Cost
Comments

Parts: Common
Supply Item Type ID
Partner ID
Part National Stock Number (NSN)
Part Type ID
System
Function
Developer
SRU (T/F)
Inventoried
Logistics Carrier Type ID
Footprint
LP Delay (days)
Initial Stock
Orbit Stock
Min Grd Stock
RTOK
Retest Cost ($FY05K)
RTTrans ($FY05K)
NRTS Rate
Condemn Rate (IMF)
Condemn Rate (Depot)
Condemn Rate (LEO)
Condemn Rate (Lunar)
Condemn Rate (Mars)
Comments

Part Type
Part Type ID
Part Type Name
Part Type Abbreviation
Internal K-Factor
External K-Factor

Maintenance Resources
Task Resource ID
Task ID
Crew Type
Crew Quantity
MTTR (hrs)
OH Time (hrs)
Robotics Quantity
Robotics MTTR (hrs)
Robotics OH Time (hrs)

Parts: Application-Specific
Part ID
Supply Item Type ID
Installed Location
Element Type ID
Manifested Flight
Activated Flight
Internal/External
Criticality
Primary CM Mode
Quiescent DC
Active DC
Assembly DC
Dormant FR
MTBF
LifeLim
LifeBeta
Test Time
RIO
Quantity per Application
Redundant
Integrator
Comments

Maintenance Tasks
Task ID
Part ID
Task Type
Task Mode Type ID
MTBSM (days)

Task Mode Types
Task Mode Type ID
Task Mode Description
Primary Robotics Type ID
Secondary Robotics Type ID
Comments

Figure 72: B-2 Diagram for Spares and Maintenance

(this page intentionally left blank)

Appendix C: Demand Models

Boiloff (COS 1)

CLASS OF SUPPLY: BOILOFF

NAME OF CALCULATED VARIABLE: BOILOFF H_2, O_2, H_2O, and Propellant

COGNIZANT ENGINEER: B.COHANIM

Equations

$$boiloffH_2 = propmass * LH_2\%ofprop * boiloffrateH_2 * arcduration$$
$$boiloffO_2 = propmass * LOX\%ofprop * boiloffrateO_2 * arcduration$$
$$boilofffuelcellwater = \min(9/8 * boiloffO_2, 9 * boiloffH_2)$$
$$boiloffexcessH_2 = boiloffH_2 - boilofffuelcellwater / 9$$
$$boiloffbreathableO_2 = boiloffO_2 - boilofffuelcellwater * 8/9$$

Definitions/Data Sources

$LH_2\%ofprop = \text{propellant mass fraction of } \mathrm{LH_2}$

$LOX\%ofprop = \text{propellant mass fraction of LOX}$

$boiloffrateH_2 = \text{nominal boiloff rate of } \mathrm{H_2}$

$boiloffrateO_2 = \text{nominal boiloff rate of } \mathrm{O_2}$

$propmass = \text{useable propellant mass}$

$arcduration = \text{lenth of time spent on an arc}$

Narrative

The boiloff model calculates how much excess hydrogen, oxygen, and water are produced from propellant boiloff (currently LOX/LH_2) and fuel cell byproducts. The boiloff model is called for each element with propellant and if there are boiloff values, the amounts of H_2, O_2, and H_2O are calculated to be used against the amount needed for crew provisions. The boiloff masses, which are given by day, are multiplied by the duration of the arc on which the elements are traveling to calculate the total boiloff by arc.

The priority is to use as much hydrogen and oxygen to produce water first, and then use oxygen for breathable air if any is left over. Excess hydrogen is dumped. The equation for fuel cell water calculates the amount of fuel cell water based on the molecular composition of water, with a 16:2 ratio of oxygen to hydrogen mass per water molecule.

The boiloff mass is subtracted from each element's current propellant mass. The water and oxygen mass produced are subtracted from the amount needed for crew provisions. The assumption is the equipment required to extract and transfer the boiloff is massless and volumeless.

Crew Provisions (COS 2)

CLASS OF SUPPLY: CREW PROVISIONS

NAME OF CALCULATED VARIABLE: CREW PROVISIONS MASS AND VOLUME

COGNIZANT ENGINEER: R. SHISHKO, B. COHANIM

Equations

$$
\begin{aligned}
DailyWater_{kg} &= liters/crew/day*(1-WaterRecoveryRate)*crewsize*arcduration - boilofffuelcellwater \\
ReserveWater_{kg} &= DailyWater_{kg} * contingencyduration / arcduration \\
EVAWater_{kg} &= 2*CrewEVA_{liters}*num\_evas \\
SampleKits_{kg} &= (liters/crew/day*(arcduration + contingencyduration)*crewsize+EVAWater)*Kit_{kg}/Kit_{liters} \\
Nonfood_{kg} &= nfikg*crewsize*arcduration \\
Ambientfood_{kg} &= afkg*crewsize*arcduration \\
Rffood_{kg} &= rffkg*crewsize*arcduration \\
Reservefood_{kg} &= (Nonfood_{kg} + Ambientfood_{kg} + Rffood_{kg}) * contingencyduration / arcduration \\
breathableoxygen_{kg} &= oxygencrewday * crewsize * arcduration - boiloffbreathableO_2 \\
reserveoxygen_{kg} &= breathableoxygen_{kg} * contingencyduration / arcduration \\
atmosphericnitrogen_{kg} &= nitrogencrewday * crewsize * arcduration \\
reservenitrogen_{kg} &= atmosphericnitrogen_{kg} * contingencyduration / arcduration \\
DailyHygieneUse_{kg} &= dailyhygienekg * crewsize * arcduration \\
ReserveHygieneUse_{kg} &= DailyHygieneUse_{kg} * contingencyduration / arcduration \\
HygieneKits_{kg} &= hygienekitkg * crewsize \\
Clothing_{kg} &= clotheskg * arcduration / dayschange * crewsize \\
PersonalItems_{kg} &= personalkg * crewsize
\end{aligned}
$$

$$
\begin{aligned}
DailyWater_{m^3} &= DailyWater_{kg} / 1000 \\
\mathrm{Re}\,serveWater_{m^3} &= \mathrm{Re}\,serveWater_{kg} / 1000 \\
EVAWater_{m^3} &= EVAWater_{kg} / 1000 \\
SampleKits_{m^3} &= (liters / crew / day * (arcduration + contingencyduration) * crewsize + EVAWater_{kg}) * volumekitMLEs / MLEvolume \\
Nonfood_{m^3} &= nfim3 * crewsize * arcduration \\
Ambientfood_{m^3} &= afm3 * crewsize * arcduration \\
Rffood_{m^3} &= rffm3 * crewsize * arcduration \\
\mathrm{Re}\,servefood_{m^3} &= (Nonfood_{m^3} + Ambientfood_{m^3} + Rffood_{m^3}) * contingencyduration / arcduration \\
breathableoxygen_{m^3} &= breathableoxygen_{kg} / LOXdensity \\
reserveoxygen_{m3} &= reserveoxygen_{kg} / LOXdensity \\
atmosphericnitroge_{m^3} &= atmosphericnitrogen_{kg} / N_2 density \\
reservenitrogen_{m^3} &= reservenitrogen_{kg} / N_2 density \\
DailyHygieneUse_{m^3} &= dailyhygienem3 * crewsize * arcduration \\
\mathrm{Re}\,serveHygieneUse_{m^3} &= DailyHygieneUse_{m^3} * contingencyduration / arcduration \\
HygieneKits_{m^3} &= hygienekitm3 * crewsize \\
Clothing_{m^3} &= clothesm3 * arcduration / dayschange * crewsize \\
PersonalItems_{m^3} &= personalm3 * crewsize
\end{aligned}
$$

Definitions/Data Sources

$DailyWater$ = daily water usage

$\mathrm{ReserveWater}$ = reserve water usage

$EVAWater$ = water usage during EVAs

$SampleKits$ = sample kits

$Nonfood$ = nonfood items

$Ambientfood$ = ambient food items

$Rffood$ = rffood items

$\mathrm{Re}\,servefod$ = reserve food items

$breathableoxygen$ = oxygen used by crew for breathing

$r\mathrm{eserveoxygen}$ = reserve oxygen used by crew for breathing

$atmosphericnitrogen$ = nitrogen used for pressurized atmosphere

$\mathrm{re}servenitrogen$ = reserve nitrogen used for pressurized atmosphere

$DailyHygieneUse$ = daily hygiene items

$\mathrm{Re}\,serveHygieneUse$ = reserve hygiene items

$HygieneKits$ = hygiene kits

$Clothing$ = crew clothing

$Personal$ = crew personal items

$crewsize =$ number of crew on an arc
$arcduration =$ length of time spent on an arc
$contingencyduration =$ length of contingency time
$liters/crew/day =$ liters of water required per crew per day
$Water\mathrm{Re}\,\mathrm{cov}\,eryRate =$ ECLSS water recovery rate for water reuse
$boilofffuelcellwater =$ excess water produced by fuel cells for human consumption
$CrewEVA_{liters} =$ liters required per team of astronauts for EVAs
$num\_evas =$ number of EVAs
$Kit_{kg} =$ mass of kits
$Kit_{liters} =$ volume of kits
$nfikg =$ non food mass
$afkg =$ ambient food mass
$rffkg =$ rf food mass
$oxygencrewday =$ amount of oxygen consumed by a crewperson per day
$boiloffbreathableO_2 =$ oxygen produced by fuel cells used for breathing
$nitrogencrewday =$ amount of nitrogen needed by a crewperson per day
$dailyhygienekg =$ daily hygiene mass
$hygienekitkg =$ hygiene kit mass
$clotheskg =$ clothes mass
$dayschange =$ number of times crew changes clothes per day
$personalkg =$ personal items mass

Narrative

Crew Provisions are split up into a “consumed” and “demand” model. The “demand” model computes all demands, mass and volume, for every subclass of crew provision: water, food and support equipment, gasses, hygiene items, clothing, and personal items based on the parameters in the Crew Provisions table (defined above) along with crew size, arc duration, contingency duration, and number of EVAs. The “demands” are computed before the simulation is run. The “consumed” crew provisions are computed only during the simulation, such as the mass and volume of water consumed on an arc. The mass and volume data come from MESSOC.

2.1 Water and Support Equipment
- Daily water usage is calculated by multiplying the amount of water the crew needs per day (a constant) by the recovery rate of water the ECLSS system provides, crew size, arc duration and subtracted by the amount of water produced from the fuel cells as a byproduct (from the boiloff model).
- Reserve water is calculated as an additional fraction by the ratio of contingency duration to arc duration.
- EVA water usage is calculated by multiplying the number of crew per EVA (2) by the number of liters the crew needs during an EVA by the number of EVAs
- Volumes for water are calculated by dividing the density of water into the mass of water. Other volumes are calculated by using ratios of values located in the Crew Provisions table of the database.

2.2 Food and Support Equipment
- Food values are calculated by multiplying a rate per day (constant from database) by the crew size by the arc duration.
- Reserves are calculated in the same way as water; by adding an additional fraction due to contingency duration.
- Volumes are calculated using constants from the database

2.3 Gases
- Oxygen and nitrogen masses are calculated by multiplying daily crew usage by crew size and arc duration.
- Oxygen from boiloff is subtracted from the demand for oxygen.
- Volumes are calculated using density.

2.4 Hygiene Items
- Hygiene masses are calculated by using daily masses per crew (from the database) by crew size and arc duration.
- Reserves are calculated in the same way as water; by adding an additional fraction due to contingency duration.
- Volumes are calculated using constants from the database

2.5 Clothing
- Clothing mass and volume are calculated by multiplying constants for daily usage (from the database) by crew size and arc duration.

2.6 Personal Items
Personal items mass and volume are calculated by multiplying constants for daily usage (from the database) by crew size.

Crew Operations (COS 3)

CLASS OF SUPPLY: CREW OPERATIONS

NAME OF CALCULATED VARIABLE: CREW OPERATIONS MASS AND VOLUME

COGNIZANT ENGINEER: S.SHULL, G. LEE

Equations

$$OfficeEquipment = OfficeEquip * crewsize$$

$$EVASuit = EVASuitCrew * crewsize$$

$$EVALiOH = LiOHCrewEVA * 2 * num\_evas$$

$$EVAO_2 = O_2CrewEVA * 2 * num\_evas$$

$$CHeCSEquipment = CHeCSEquip$$

$$CHeCSConsumables = CHeCSCons * crewsize * arcduration$$

$$SafetyEquipment = SafetyEquip$$

$$CommunicationsEquipment = CommEquip$$

$$ComputerEquipment = CompEquip * crewsize$$

Definitions/Data Sources

$OfficeEquip$ = mass and volume of office equipment per crew

$EVASuitCrew$ = mass and volume of an EVA Suit per crew

$LiOHCrewEVA$ = mass and volume of LiOH per crew per EVA

$O_2CrewEVA$ = mass and volume of $\mathrm{O_2}$ per crew per EVA

$CHeCSEquip$ = mass and volume of CHeCS equipment

$CHeCSCons$ = mass and volume of CHeCS consumables per crew per day

$SafetyEquip$ = mass and volume of safety equipment

$CommEquip$ = mass and volume of communications equipment

$CompEquip$ = mass and volume of computer equipment per crew

Narrative

The mass data for the Crew Operations demand model comes from HSMAD, the ESAS final report, and engineering judgement. Using a sub-set of Space Station data, average densities were calculated for each of the Crew Operations sub-classes. The volume data was then calculated by dividing the mass data by these densities.

Maintenance and Upkeep (COS 4)

CLASS OF SUPPLY: MAINTENANCE AND UPKEEP

NAME OF CALCULATED VARIABLE: PIPELINE FOR ORU$_i$

COGNIZANT ENGINEER: R. SHISHKO

Equations

For pre-positioned spares at a node (and for carryalong sorties):

$$Truereplace/day_{i,e} = \sum_{p \in P(i,e)} \left(\frac{24DC_p K_p}{MTBF_p} \right) QPA_p (1 - RIO_p) = \sum_{p \in P(i,e)} \psi_p$$

$$Truereplace/day_{i,n,t} = \sum_{e \in E(n,t)} (Truereplace/day_{i,e})$$

$$\begin{aligned} Truereplace/resupplycycle_{i,n,t} &= (Truereplace/day_{i,t})(days/resupplycycle_{n,t}) \\ &= \lambda_i \end{aligned}$$

Definitions/Data Sources

$DC_p =$ **Duty cycle for application p**

$K_p =$ **K - factor for application p**

$MTBF_p =$ **Mean time between failure for application p**

$RIO_p =$ **Repair - in - orbit rate for application p**

$QPA_p =$ **Quantity per application p**

$P(i,e) =$ **Set of all applications of ORU i in instantiated element e**

$E(n,t) =$ **Set of all instantiated elements at node n at time t**

$\lambda_i =$ **Pipeline for ORU i**

Narrative

For carryalong sorties, *days/resupplycycle* is replaced by effective mission length, and *E(n,t)* is replaced by the set of all elements occupied by the crew at any time, *t*, during the mission. MTBF is measured in hours, which accounts for the 24.

CLASS OF SUPPLY: MAINTENANCE AND UPKEEP

NAME OF CALCULATED VARIABLE: CONSTRAINED AVAILABILITY MAXIMIZATION AND OPTIMAL SPARES

COGNIZANT ENGINEER: R. SHISHKO

Equations

For pre-positioned spares at a node (and for carryalong sorties):

$$max\ A(s_1, s_2, ..., s_m) = \prod_{i=1}^{m} A_i = \prod_{i=1}^{m} \left(1 - \frac{EBO_i(s_i, \lambda_i)}{q_i}\right)^{q_i}$$

subject to $R(s_1, s_2, ..., s_m) \le \overline{R}$ **and** $s_i \ge 0$

where $EBO_i(s_i, \lambda_i) = \sum_{x > s_i}^{\infty} (x - s_i) p(x, \lambda_i)$

$$= \lambda_i (1 - \sum_{x=0}^{s_i - 1} p(x, \lambda_i)) - s_i (1 - \sum_{x=0}^{s_i} p(x, \lambda_i))$$

$$q_i = \sum_{e \in E(n,t)} \sum_{p \in P(i,e)} QPA_p$$

Select optimal spares set by ordering $\frac{\partial \ln A / \partial s_i}{\partial R / \partial s_i} \cong \frac{\ln A_i(s_i + 1) - \ln A_i(s_i)}{R_i}$ **from highest to lowest values; cutoff point when target availability is achieved**

Definitions/Data Sources

λ_i = **Pipeline for ORU i**

$EBO_i(s_i,\lambda_i)$ = **Expected backorders for ORU i with** s_i **prepositioned spares**

$p(x,\lambda_i)$ = **Poisson probability of x events with parameter** λ_i

q_i = **Total quantity of ORU i at node n at time t**

Narrative

For carryalong sorties, q_i is the total quantity of ORU i in elements occupied by the crew at any time, *t*, during the mission.

The resource constraint can be mass or volume as selected by the user. The user also selects a target availability, e.g., 95%. The *SpaceNet* algorithm returns an optimal set of spares, called the "buylist", that meets the target availability and that does not violate the resource constraint. The mass and volume of the buylist can then be calculated. (See below.)

If the target availability is set too high, then the *SpaceNet* algorithm returns a buylist that meets the resource constraint, and it reports the achieved availability.

The user can also select a threshold availability, e.g., 65%. The *SpaceNet* algorithm always returns a buylist that meets the threshold availability, even if it means violating the resource constraint. This feature is intended for use in auto-manifesting.

CLASS OF SUPPLY: MAINTENANCE AND UPKEEP

NAME OF CALCULATED VARIABLE: SPARES MASS AND VOLUME

COGNIZANT ENGINEER: R. SHISHKO

Equations

For pre-positioned spares at a node (and for carryalong sorties):

$$Buylistmass_{n,t} = \sum_i s_i^* (mass_i)$$

$$Buylistvol_{n,t} = \sum_i s_i^* (volume_i)$$

Definitions/Data Sources

$s_i^* =$ **Quantity of ORU i from the constrained availability maximization problem**

Narrative

The total mass and volume of spares to be pre-positioned (or manifested for a carryalong sortie) is obtained by multiplying the quantity of ORU i in the buylist by its mass or volume respectively.

CLASS OF SUPPLY: MAINTENANCE AND UPKEEP

NAME OF CALCULATED VARIABLE: EXPECTED ORU REPLACEMENTS AND RETURNS

COGNIZANT ENGINEER: R. SHISHKO

Equations

For resupply and return at a node:

$$\xi_p \equiv MTBPMRR_p / MTBF_p$$

$$Allreplace/day_{i,e} = \sum_{p \in P(i,e)} \psi_p /(1 - RTOK_i) + \sum_{p \in P(i,e)} \psi_p \, exp(-\xi_p)/(1 - exp(-\xi_p)) \textbf{ for } \xi > 0$$

$$Allreplace/day_{i,n,t} = \sum_{e \in E(n,t)} (Allreplace/day_{i,e})$$

$$Allreplace/resupplycycle_{i,n,t} = (Allreplace/day_{i,n,t})(days/resupplycycle_{n,t})$$

$$Allreturns/returncycle_{i,n,t} = (Allreplace/resupplycycle_{i,n,t})(1 - CIO_i)\left(\frac{days/returncycle_{n,t}}{days/resupplycycle_{n,t}}\right)$$

Definitions/Data Sources

n = **node**
t = **time**
i = **ORU type**
e = **instantiated element**

Narrative

All replacements per day include true replacements inflated to account to false removals using RTOK, plus additional preventative removals, represented by the second term. When RTOK is included in the K-factor, the calculation of all replacements per day simplifies. All replacements per resupply cycle depend on the node's instantiated elements during the previous resupply cycle. All returns also depend on the condemnation policies for each ORU type.

CLASS OF SUPPLY: MAINTENANCE AND UPKEEP

NAME OF CALCULATED VARIABLE: EXPECTED ORU REPLACEMENT AND RETURN MASS (AND VOLUME)

COGNIZANT ENGINEER: R. SHISHKO

Equations

For resupply at a node:

$$Allreplacemass/resupplycycle_{i,n,t} = (Allreplace/day_{i,n,t})(days/resupplycycle_{n,t})m_i$$

$$Allreplacemass/resupplycycle_{n,t} = \sum_i (Allreplacemass/resupplycycle_{i,n,t})$$

$$AllreplacemassSigma/resupplycycle_{n,t} = \sqrt{\sum_i (Allreplacemass/resupplycycle_{i,n,t})m_i^2}$$

$$Allreturnsmass/returncycle_{i,n,t} = (Allreturns/returncycle_{i,n,t})m_i$$

$$Allreturnsmass/returncycle_{n,t} = \sum_i (Allreturnsmass/returncycle_{i,n,t})$$

$$AllreturnsmassSigma/returncycle_{n,t} = \sqrt{\sum_i (Allreturnsmass/returncycle_{i,n,t})m_i^2}$$

Definitions/Data Sources

m_i = **mass of ORU i**

Narrative

The expected total mass (volume) of spares to be resupplied is obtained by multiplying the expected resupply quantity of ORU i by its mass (volume) respectively. The expected total mass (volume) of spares to be returned is obtained by multiplying the expected return quantity of ORU i by its mass (volume) respectively.

CLASS OF SUPPLY: MAINTENANCE AND UPKEEP

NAME OF CALCULATED VARIABLE: CORRECTIVE MAINTENANCE CREWHOURS

COGNIZANT ENGINEER: R. SHISHKO

Equations

Corrective maintenance at a node:

$$TIMMH_p = \sum_{s \in M_{SRU}(p)} \left(MTTR_s + MTTROH_s \right) T_s$$

$$TRRMH_p = \sum_{s \in M_{CMRR}(p)} \left(MTTR_s + MTTROH_s \right) T_s$$

$$CorrCrewHrs / day_{i,e} = \sum_{p \in P(i,e)} \left(\frac{24 DC_p K_p}{MTBF_p} \right) QPA_p \left(\frac{(1 - RIO_p) TRRMH_p}{(1 - RTOK_p)} + RIO_p TIMMH_p \right)$$

$$CorrCrewHrs / resupplycycle_{i,n,t} = \sum_{e \in E(n,t)} (CorrCrewHrs / day_{i,e})(days / resupplycycle_{n,t})$$

Definitions/Data Sources

$MTTR_s$ = **Mean Time To Repair for subtask s**

$MTTROH_s$ = **Mean Time To Repair overhead for subtask s**

T_s **= Number of crewpersons involved in subtask s**

$M_{CMRR}(p)$ = **Set of subtasks involved in corrective maintenance by remove-and-replace for application p**

$M_{SRU}(p)$ = **Set of subtasks involved in corrective maintenance by SRU replacement for application p**

Narrative

The expected corrective maintenance crewhours per resupply cycle is obtained by multiplying the expected number of corrective actions by crewhours per action. The equations allows for corrective maintenance by either remove-and-replace, by SRU replacement, or a mix. When RTOK is included in the K-factor, the calculation is simplified. Overhead time (preparation, travel to worksite, cleanup) is counted separate from actual worksite time. Typically, the number of crewpersons involved in each subtask is 1, though for larger ORUs or more complex subtasks, 2 crewpersons may be needed.

Stowage and Restraint (COS 5)

CLASS OF SUPPLY: STOWAGE AND RESTRAINT

NAME OF CALCULATED VARIABLE: ITEMS TO MANIFEST, STOWAGE MASS AND VOLUME

COGNIZANT ENGINEER: G. LEE

Equations

Definitions/Data Sources

$\mathit{HalfCTB}$ = mass and volume of a half cargo transfer bag (CTB)

$\mathit{HalfCTBCapacity}$ = mass and volume capacity of a half CTB

$\mathit{SingleCTB}$ = mass and volume single CTB

$\mathit{SingleCTBCapacity}$ = mass and volume capcity of a single CTB

$\mathit{CTB\%Capacity}$ = mass and volume packing efficiency

CWC = mass and volume of a contingency water container (CWC)

$\mathit{CWCCapacity}$ = mass and volume capacity of a CWC

Narrative

The stowage model packs the mass and volume demands from the other demand models into half CTBs, single CTBs, CWCs, or oversized items. Bags are functionally divided by supply class and sub-supply class. In the cases of crew provisions and crew operations where lump masses and volumes are calculated, the stowage model uses the limiting constraint (usually volume) to calculate the number of bags. In the cases of maintenance and upkeep and exploration and research where the individual supply item types are calculated, the stowage model uses a packing efficiency parameter to specify when to start a new bag. Items that exceed single CTB mass and volume constraints are manifested individually as oversized items.

Exploration and Research (COS 6)

CLASS OF SUPPLY: EXPLORATION AND RESEARCH

NAME OF CALCULATED VARIABLE: EXPLORATION ITEMS QUANTITY

COGNIZANT ENGINEER: A. SIDDIQI

Equations

$$EquipmentNames = [A, B, C....]$$

$$W_{COS}\mathrm{Ref} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ \vdots & & & \\ x_{n1} & x_{n2} & x_{n3} & x_{n4} \end{bmatrix}$$

$$Q_{COS}\mathrm{Ref} = [y_1, y_2, ... y_n]$$

$$W_{norm} = \left[W_{COS}\mathrm{Ref} \times \mathrm{S}_{\mathrm{mix}}\right] / F$$

$$Scale_{crwe} = ceil(nTeams / nTeamsref)$$

$$Quantity_{COS} = ceil((W_{norm} .* \ Scale_{crew}).* (Scale_{time}))$$

Definitions/Data Sources

$EquipmentNames$ = vector in which each element is a name (string) of an exploration item
$W_{COS}\mathrm{Ref}$ = matrix in which each row corresponds to a particular exploration item
in a class of supply (CoS), and the four columns correspond to weights for life, climate,
geology, resources
$Q_{COS}\mathrm{Ref}$ = vector of reference quantity for each exploration item in a CoS
$\mathrm{S}_{\mathrm{mix}}$ = vector of 4 elements that are weights (between 0 and 1) for type of
exploration (life, climate, geology, resources) to be carried out
F = normalizing factor based on reference study mission exploration items
W_{norm} = normalized weight for each exploration item based on reference
weight, and given $\mathrm{S}_{\mathrm{mix}}$
$Scale_{crew}$ = scaling factor for crew
$\mathrm{Scale}_{\mathrm{time}}$ = scaling factor for time (mission duration)
$Quantity_{COS}$ = vector of quantities of each exploration item in a CoS

Narrative

The quantity of exploration items is determined from a baseline Mars exploration mission with a defined set of equipment required to satisfy some defined science exploration objectives. The *EquipmentNames* vector contains a set of equipment names from which the set of exploration items is selected. The $W_{COS}Ref$ matrix contains data from the baseline mission and provides weights for each item for each type of exploration (Life, Climate, Geology, Resources). The $Q_{COS}Ref$ contains the reference quantities for each item. Based on the baseline case, and using the given weights for the four exploration types (in S_{mix}), the normalized weights for each item are calculated then used for determining the quantity for each item to be used in the mission defined by the user. Depending on the class of supply (CoS) the weight may also be scaled with the number of teams (for concurrent EVAs) the and mission duration. The scaling for teams is performed for CoS 602 which is field equipment and each team must have its own set. The time scaling is performed for monitoring stations (an exploration item) and samples (which are CoS 603). The assumption is that for longer missions larger areas will be explored and therefore more monitoring stations will be necessary, and similarly due to greater exploration more samples will be collected. Also, depending on the planet, certain items are not included such as balloons and atmospheric samplers for Lunar missions.

Waste and Waste Management (COS 7)

CLASS OF SUPPLY: WASTE AND WASTE MANAGEMENT EQUIPMENT

NAME OF CALCULATED VARIABLE: WASTE MANAGEMENT EQUIPMENT MASS AND VOLUME

COGNIZANT ENGINEER: S. SHULL

Equations

*total trashbags = trashbags*crewsize*missionduration*
*wcs supplies = wcs*crewsize*missionduration*
*contingency wcs supplies= contingencywcs*crewsize*missionduration*

Definitions/Data Sources

trashbags = mass and volume of trashbags needed per crew per day
wcs = mass and volume of other waste containment system (wcs) equipment needed per crew per day
contingencywcs = mass and volume of contingency wcs supplies needed per crew per day

Narrative

The mass and volume data for the waste management equipment demand model comes from Chapter 18 of HSMAD.

(this page intentionally left blank)

Appendix D: Measures of Effectiveness

The value of planetary space exploration comes primarily from healthy and qualified explorers and scientists (=crew) being able to spend a certain amount of time at one or more planetary surface locations ("nodes"). To first order, the exploration benefit should scale linearly with both the number of people (crew size) as well as the duration of their stay. In order to do productive research, however, the crew needs to have specific exploration equipment and scientific instruments, such as cameras, rock hammers and so forth with them. Surface infrastructure items (habitation modules, power systems, surface mobility systems, etc.) act as further enablers and multipliers for exploration productivity. The concept of "exploration mass" in *SpaceNet* includes both exploration/science equipment and infrastructure mass. More precisely in *SpaceNet*, "exploration mass" consists of items in Class-of-Supply (COS) 6 (Exploration and Research) and COS 8 (Habitation and Infrastructure). See Appendix A for a list of all classes-of-supply and subclasses.

Crew Surface Days (CSD)

$$CSD_{tot} = \Delta T \sum_{i=1}^{T} \sum_{j=1}^{S} N_{crew,i,j} \tag{1}$$

ΔT = Earth days per *SpaceNet* time period
T = total number of *SpaceNet* time periods in a scenario[16]
S = number of planetary exploration nodes visited in a scenario
$N_{crew,i,j}$ = number of crew present at exploration node j during time period i

CSD_{tot} is the total cumulative number of crew surface days spent over an entire scenario over all S nodes explored. We can also compute the total number of crew surface days accumulated up to a certain time in a campaign as:

$$CSD(k) = \Delta T \sum_{i=1}^{k} \sum_{j=1}^{S} N_{crew,i,j} \quad \text{whereby } k < T \tag{2}$$

Similarly, *SpaceNet* 1.3 defines the cumulative number of crew surface days spent over the entire scenario at a particular surface node as:

$$CSD_{tot}(j) = \Delta T \sum_{i=1}^{T} N_{crew,i,j} \tag{3}$$

---

[16] A scenario can include a single "sortie" style mission or an entire campaign, containing multiple missions.

The units of these metrics are [crew-days]. An example of such a calculation can be made based on the hypothetical data given in Table C-1.

Table C-1: Hypothetical Lunar Campaign Crew Data by Node, ΔT=2days

Node *j*	*i*=1	*i*=2	*i*=3	*i*=4	*i*=5	*i*=6	*i*=7
2001	0	3	3	0	0	0	0
2009	0	0	2	2	2	2	0
2312	0	0	0	0	4	4	4

Here, three surface nodes are visited, first three crew ($N_{crew}=3$) explore Node 2001 during time periods 2 and 3, then a crew of two explores Node 2009 during time periods 3 through 6. Finally, Node 2312 is visited by a crew of four during the last three periods.

The resulting CSD metrics for this scenario (campaign) are:

$$CSD_{tot}=2\left(0+3+5+2+6+6+4\right)=52$$
$$CSD(k)=[0\ \ 6\ \ 16\ \ 20\ \ 32\ \ 44\ \ 52]$$
$$CSD_{tot}\left(j\right)=\left[12\ \ 16\ \ 24\right]$$

Exploration Mass Delivered (EMD)

SpaceNet 1.3 computes the total amount of exploration mass delivered to all surface nodes over the course of a campaign (scenario). Exploration mass contains all items that are characterized as COS 6 and COS 8.[17]

$$EMD_{tot}=\sum_{i=1}^{T}\sum_{j=1}^{S}[\Delta m_{COS6,i,j}+\Delta m_{COS8,i,j}] \tag{4}$$

T = total number of time periods in a scenario
S = number of planetary surface nodes visited in a scenario[18]
$\Delta m_{COS6,i,j}$ = additional (new) exploration items mass delivered at time period *i* to node *j*
$\Delta m_{COS8,i,j}$ = additional (new) infrastructure mass delivered at time period *i* to node *j*

In Eq. (4), only mass additions are counted, i.e., removal of exploration mass from a surface node is ignored even though such an action causes a Δm_{COS} from one period to another. As with crew surface days, the cumulative *EMD* up to a certain time in the campaign can be defined as:

---

[17] Sub-class-of-supply 806 (orbiting service systems) is not factored into exploration mass, because it is not delivered to a surface node. An example of an orbiting service system is a communications relay spacecraft in lunar orbit.

[18] For programming purposes, planetary surface nodes can be identified based on the node number in *SpaceNet*. Lunar surface nodes are all nodes from 2001 to 2500.

$$EMD(k) = \sum_{i=1}^{k} \sum_{j=1}^{S} [\Delta m_{COS6,i,j} + \Delta m_{COS8,i,j}] \qquad \text{where } k < T \qquad (5)$$

Similarly, *SpaceNet* defines the cumulative EMD over the entire scenario at a particular surface node as:

$$EMD_{tot}(j) = \sum_{i=1}^{T} [\Delta m_{COS6,i,j} + \Delta m_{COS8,i,j}] \qquad (6)$$

The units of these metrics are [kg]. An illustration of such a calculation can be made based on the hypothetical data given in Tables C-1 and C-2.

Table C-2: Hypothetical Lunar Campaign Exploration Mass Data by Node [kg]

Node *j*		*i*=1	*i*=2	*i*=3	*i*=4	*i*=5	*i*=6	*i*=7
2001	COS 6	0	200	200	200	200	200	200
	COS 8	0	500	500	500	500	500	500
2009	COS 6	0	0	400	400	400	400	0
	COS 8	0	0	800	800	1200	1200	0
2312	COS 6	0	600	600	600	600	600	600
	COS 8	0	1000	1000	1000	1000	1000	1000

Here, exploration mass (COS 6 + COS 8) is delivered to all three surface nodes. The cells in Table C-2 show the amount of COS 6 and 8 mass present at each node, at each time period. New deliveries can be inferred by taking the difference with the previous time period. In some cases the exploration mass arrives concurrently with the crew (carry-along) and remains on the surface after the crew leaves (e.g., at Node 2001). In other cases there can be a re-supply while the crew is on the surface and/or the exploration mass is removed by the crew upon completion of the surface mission (see Node 2009). Also, exploration mass can be pre-positioned before arrival of the crew (Node 2312). The resulting *EMD* calculations are as follows:

$$EMD_{tot} = 0 + 2300 + 1200 + 400 + 0 + 0 + 0 = 3900$$

$$EMD(k) = \begin{bmatrix} 0 & 2300 & 3500 & 3900 & 3900 & 3900 & 3900 \end{bmatrix}$$

$$EMD_{tot}(j) = \begin{bmatrix} 700 & 1600 & 1600 \end{bmatrix}$$

SpaceNet 1.3 counts COS 6 and 8 as exploration mass because these *directly* (explicitly) contribute to exploration capability and can therefore be counted as a benefit. The other COSs are all *indirect* enablers (e.g., propellants, crew consumables, spares, etc.).

Total Launch Mass (TLM)

TLM is the total launch mass lifted off from the surface of the Earth to accomplish a particular exploration scenario. *SpaceNet* calculates this metric as:

$$TLM = \sum_{v=1}^{N_f}\sum_{e=1}^{E} m_{empty,e} N_{e,v} + \sum_{k \neq 0,8,9} m_{COSk} + \sum_{v=1}^{N_f}\sum_{e=1}^{E} m_{crew} N_{crew,e} N_{e,v} \qquad (7)$$

where

N_f = total number of launches in a scenario
E = number of element types in a scenario[19]
$m_{empty,e}$ = empty mass (= dry (tare) mass + accommodation mass) of element type e
$N_{e,v}$ = number of type e elements on launch v
$m_{COS,k}$ = total mass launched for class of supply k (includes k= COS 1 (propellant) and excludes COS 0, 8, and 9)
$N_{crew,e}$ = number of crew present on type e elements
m_{crew} = nominal mass of a crew member (usually taken to be ~220 lbs = 100 kg)

This calculation is important because each component is needed to develop the mass fractions used in Eq. (13). The first term in Eq. (7) divided by *TLM* contains the mass fractions for COS 8 and COS 9, the individual masses in the second term divided by *TLM* are the mass fractions for COS 1 through 7 and 10, and the last term divided by *TLM* is the mass fraction for the crew (COS 0).

The units of this metric are [kg]. The total launch mass for Apollo 17 was approximately 2,930 metric tons, about 3 million kilograms.

Up-Mass Capacity Utilization (UCU)

A key metric to understand exploration logistics efficiency is that of cargo *capacity utilization*. Each propulsive and non-propulsive element (COS 9) has a cargo capacity (in terms of mass and volume) associated with it. In some cases for propulsive elements such as the Earth Departure Stage[20] (EDS), this capacity is zero, but for non-propulsive elements, the cargo capacity is usually non-zero.[21]

SpaceNet 1.3 quantifies the fraction of available up-mass (launched from Earth) cargo capacity that is actually used for all classes-of-supply other than infrastructure (COS 8), transportation and carriers (COS 9), propellant (COS 1), and crew (COS 0). An efficient transportation system has a capacity utilization of *unity*. For a variety of reasons, this number is commonly lower in actuality. One such reason is that a volume constraint may come into play before the cargo mass capacity is reached.

$$UCU = \frac{\sum_{e=1}^{EI} \sum_{k=2,3,4,5,6,7,10} m_{COSk,e}}{\sum_{e=1}^{EI} (m_{cap,e} - m_{accom,e})} \qquad (8)$$

---

[19] Note that this is not the same as the number of instantiated elements in the scenario. Elements in COS 9 include both propulsive and non-propulsive (carrier) elements.
[20] The EDS, as presented in NASA's ESAS architecture, fulfills a role similar to the S-IVB in Apollo.
[21] Fuel capacity (COS 1) is treated separately in *SpaceNet 1.3*.

$m_{COS,k,e}$ = total mass of class of supply k carried in instantiated element e at Earth launch
EI = number of instantiated elements in a scenario
$m_{cap,e}$ = up-mass cargo capacity of instantiated element e (from element type)
$m_{accom,e}$ = accommodation mass of instantiated element e (from element type)

Eq. (8) states that the upmass capacity utilization in a scenario is the ratio of the mass of tracked supply classes launched over all elements, divided by the total usable cargo mass capacity (= maximum cargo up-mass capacity less accommodation mass). *SpaceNet 1.3* performs a check to ensure that only elements, *e*, that are launched from Earth are counted in the summation. *UCU* will always be a number between zero and one.[22] A *UCU* of zero means that all elements are being launched empty, while a *UCU* of 1.0 indicates that the cargo up-mass capacity of the system is being fully utilized.

Computing the *UCU* for a scenario as a whole is useful, especially during iterative scenario definition. If *UCU* is less than 1.0, additional supply items such as spares, consumables or exploration items can be loaded (if volume constraints are not exceeded) with only a small impact on *TLM*. These additional items will eitherimprove *EC,* improve *system availability* or decrease the probability of under-supply situations due to a lack of consumables or spares. Another potential source of up-mass capacity utilization loss occurs when the launch vehicle capability is the binding constraint, not the carrier's cargo capacity. This is handled in *SpaceNet* as follows: if the analyst has manifested cargo up to the carrier's cargo capacity, but the stack is too massive to launch, an error message is issued. The analyst may try to overcome this by offloading some cargo until a feasible launch mass is achieved. An alternative would be to create another launch to carry the excess cargo.

Return-Mass Capacity Utilization (RCU)

Another important metric to quantify is the fraction of available non-destructive return-mass (returned to Earth) cargo capacity that is actually used for all classes of supply other than infrastructure (COS 8), transportation and carriers (COS 9), propellant (COS 1), and crew (COS 0). *SpaceNet 1.3* calculates this fraction using Eq. (9).

$$RCU = \frac{\sum_{e=1}^{EI} \sum_{k=2,3,4,5,6,7,10} m^{r}_{COSk,e}}{\sum_{e=1}^{EI} (m_{returncap,e} - m_{accom,e})} \tag{9}$$

$m^{r}_{COS,k,e}$ = total mass of class of supply k carried in instantiated element e at Earth return
EI = number of instantiated elements in a scenario
$m_{returncap,e}$ = non-destructive return-mass cargo capacity of instantiated element e (from element type)

---

[22] The reason that UCU cannot exceed 1 is that if the analyst tries to allocate more than an element can hold, an error message is issued in *SpaceNet 1.3*.

$m_{accom,e}$ = accommodation mass of instantiated element e (from element type)

Eq. (9) states that the non-destructive return-mass capacity utilization in a scenario is the ratio of the mass of tracked supply classes returned to Earth over all elements, divided by the total usable cargo return-mass capacity (= maximum cargo return-mass capacity less accommodation mass). *SpaceNet* performs a check to ensure that only elements, *e*, that are launched from Earth and then return to Earth are counted in the summation. *RCU* will always be a number between zero and one.[23] An *RCU* of zero means that all elements launch from and returning to the Earth are empty upon return, while a *RCU* of 1.0 indicates that the non-destructive return-mass cargo capacity of the system is being fully utilized.

Computing *RCU* for a scenario as a whole is important, since it reflects that benefits obtained by being able to return samples (COS 603) or failed equipment back (COS 703) for inspection on Earth.

Exploration Capability (EC)

To first order, the exploration capability is the amount of time the crew gets to spend doing exploration and research at a surface node, multiplied by the amount of total exploration mass they have to do the job at each node visited during the scenario. The amount of time the crew can spend doing exploration and research while at a surface node is limited by a number of factors. These sources of crew non-availability include: housekeeping activities, maintenance and repair, in-situ crew activity planning/scheduling, medical, EVA preparation, and physiological (exercise, sleep/rest, eating). In general, the fraction of non-available crew-hours may vary with the size of the crew at the surface node and the length of the surface stay. *SpaceNet 1.3* defines exploration capability as:

$$EC_{tot} = \Delta T \sum_{i=1}^{T} \sum_{j=1}^{S} f_{i,j}(N_{crew,i,j})[m_{COS6,i,j} + m_{COS8,i,j}] \tag{10}$$

ΔT = Earth days per *SpaceNet* time period
T = total number of time periods in a scenario
S = number of exploration nodes visited in a scenario
$N_{crew,i,j}$ = number of crew present at node j during time period i
$m_{COS6,i,j}$ = total exploration items mass present during time period i at node j
$m_{COS8,i,j}$ = total infrastructure mass present during time period i at node j

To account for crew non-available time in the simplest way, rewrite Eq. (10) as:

---

[23] The reason that RCU cannot exceed 1 is that if the analyst tries to allocate more than an element can hold, an error message is issued in *SpaceNet*.

$$EC_{tot} = \Delta T \sum_{i=1}^{T} \sum_{j=1}^{S} (1-\alpha_{ij}) N_{crew,i,j} [m_{COS6,i,j} + m_{COS8,i,j}] \qquad (10')$$

where α_{ij} = fraction of non-available crew time at node *j* during period *i*. The units of this metric are [kg • crew-days].

Note that, even when α_{ij} = 0, the exploration capability is generally *not* the same as simply the product of total exploration mass delivered and total crew surface days in the entire scenario.

$$EC_{tot} \neq EMD_{tot} \times CSD_{tot} \qquad (11)$$

In order to provide exploration capability, crew and exploration equipment have to be *co-located* at the same node concurrently. Crew at a surface node without exploration equipment does not produce benefit (*SpaceNet 1.3* does not account for the productivity of robotic agents), and conversely exploration equipment at a node without crew to operate that equipment does not provide any benefit either. However, when both crew and exploration mass always travel together, as during the Apollo missions, then Eq. (11) holds with equality.

Sample calculations for *EC*, using the data in Tables C-1 and C-2 and assuming that α_{ij} = 0 yield:

$$EC_{tot} = \Delta T \sum_{i=1}^{T} \sum_{j=1}^{S} (1-\alpha_{ij}) N_{crew,i,j} [m_{COS6,i,j} + m_{COS8,i,j}] = 2\Big[\big(0+0+0\big)+\big(3\cdot 700+0+0\big)+ \\ \big(3\cdot 700+2\cdot 1200+0\big)+\big(0+2\cdot 1200+0\big)+\big(0+2\cdot 1600+4\cdot 1600\big)+\big(0+2\cdot 1600+4\cdot 1600\big)+ \\ \big(0+0+4\cdot 1600\big)\Big] = 69,200$$

Clearly, this is not the same as,

$$EC_{tot} = 69,200 \neq EMD_{tot} \times CSD_{tot} = 52 \times 3900 = 202,800 .$$

This is so, because EC_{tot} requires concurrency and co-location of crew and exploration mass. In other words, the total exploration capability is the *dot product* of the exploration mass and the number of crew present, summed up for each (surface) node explored in a scenario.

Relative Exploration Capability (REC)

The relative exploration capability is a normalized measure of *exploration logistics efficiency*. It measures the amount of productive exploration that can be done for each kilogram of mass launched from the Earth's surface, relative to Apollo 17. Apollo 17 is used as the reference case because it can be argued that of all the Apollo lunar surface missions, Apollo 17 was the most productive in terms of exploration and science and also the one that came closest to approaching the constraints imposed by flight hardware elements and operational capabilities at that time.

The relative exploration capability metric is based on well-established theory of "linked" index numbers, used to measure changes in economic productivity due to technology improvements.[24] The exploration capability metric, Eq. (10'), is *explicitly* a function of the number of crew and the exploration mass present at each node. It is also implicitly a function of all classes of supply since crew provisions, crew operations equipment, propellants, carriers, spares etc. are needed to "produce" crew at the surface node. A relative exploration capability metric needs to account for the necessary use and consumption of these resources as well in establishing overall logistics efficiency. *SpaceNet* uses Eq. (12), a commonly computed linked index number, as the relative exploration capability metric using Apollo 17 as the basis for normalization (i.e., *REC* =1 for Apollo 17).

$$REC_b = \frac{EC_{tot}^{b} / EC_{tot}^{a17}}{\prod_k \left(\frac{m_{COSk}^{b}}{m_{COSk}^{a17}} \right)^{\beta_k}} \tag{12}$$

where

$$\beta_k = \frac{1}{2}\left(\omega_k^{a17} + \omega_k^{b}\right) \tag{13}$$

EC^{b}_{tot} = exploration capability metric for scenario (campaign) *b* according to Eq. (10').
m^{b}_{COSk} = total class of supply *k* mass delivered in scenario (campaign) *b*.[25]
ω^{b}_{k} = mass fraction for class of supply *k* in scenario (campaign) *b*.

The *REC* metric is dimensionless. Specific numbers for Apollo 17 were derived from data at http://history.nasa.gov/SP-4029/Apollo_17a_Summary.htm

An interplanetary supply chain with a *REC* >1 would indicate a more efficient supply chain technology than Apollo 17 because more exploration capability is being provided for each unit of mass launched from Earth. Conversely a *REC* < 1 would indicate a less efficient supply chain technology than Apollo 17. Clearly, the value of the *REC* metric will be influenced by a number of factors such as:

- the chosen mission/transportation architecture
- the use of various propulsion (and other) technologies
- various supply chain strategies implemented (e.g., on orbit depots)
- the application of various ISRU technologies.

---

[24] Also called Divisia index numbers. See Hulten, Charles, "Divisia Index Numbers", *Econometrica*, Vol. 41, No. 6, (November 1973)

[25] Class of Supply 0 (the crew itself) must be included in Eq. (11). The nominal mass of a crew member is usually taken to be ~220 lbs = 100 kg without a space suit, but with indoor clothing.

For example, *REC* would be able to capture the effect of using in-situ resources (ISRU) as a supply chain strategy. If no ISRU is applied, a certain amount of consumables must be carried along to supply the missions in a particular scenario. These consumables directly contribute to the denominator of Eq. (12), making the value of *REC* less. If ISRU is used[26], the consumables mass over an entire scenario might be reduced, but the upfront mass penalty for transporting the ISRU (and related power) equipment to a node in the first place would be automatically captured as well. Whether or not an investment in ISRU would be worthwhile for a particular scenario, could then be assessed by comparing the *REC* of both alternatives with and without ISRU.

We can carry out sample calculations for *REC*, assuming that the total launch mass, not explicitly shown in Table C-2, is 3,000 metric tons for each surface mission.

Table C-3: Hypothetical Calculation of REC for a Lunar Sortie (normalized against Apollo 17=A17)

	m_{COSk} A17	$\omega^{a17}{}_k$	m_{COSk} b	$\omega^b{}_k$	β_k (Eq. 10)
k=0 crew	3(100)=300	$1.024\ 10^{-4}$	9(100)=900	$1.0\ 10^{-4}$	$1.012\ 10^{-4}$
k=6 exploration	207	$7.06\ 10^{-5}$	1200	$1.33\ 10^{-4}$	$1.018\ 10^{-4}$
k=8 infrastructure	208	$7.10\ 10^{-5}$	2700	$3.0\ 10^{-4}$	$1.855\ 10^{-4}$
k=other	2,929,987	0.9997560	8,995,200	0.999467	0.9996115
Total [kg]	2,930,702		9,000,000		

Clearly, supply classes other than COS 0, 6, and 8 dominate the total mass launched. Specifically, these are COS 9 (propulsive and non-propulsive elements) and especially COS 1 (propellants and fuels). Even without breaking these other classes-of-supply out in greater detail, we can substitute numerical values in Eq. (12):

$$REC_b = \frac{EC_{tot}^{b} / EC_{tot}^{a17}}{\prod_k \left(\frac{m_{COSk}^{b}}{m_{COSk}^{a17}} \right)^{\beta_k}} = \frac{69,200/2,594}{(1.0001 \cdot 1.00018 \cdot 1.00048 \cdot 3.069)} \cong 8.69$$

Thus, the hypothetical scenario *b* and data in Tables C-1 and C-2) has a combined exploration logistics efficiency that is approximately 9 times that of Apollo 17. In reality a variety of factors will influence this number--as described above-- with propulsive efficiency (I_{sp}) likely being the key driver. We can approximate the Divisia index number in Eq. (12) by simply dividing by the total launch mass for each scenario. For scenario *b*, Eq. (12) and (12') yield identical values to the 4th significant digit.

$$REC_b \approx \frac{EC_{tot}^{b} / EC_{tot}^{a17}}{TLM^{b} / TLM^{a17}} \qquad (12')$$

---

[26] SpaceNet 1.3 does not have the ability to explicitly model ISRU (left for future releases).

Relative Scenario Cost (RSC)

The cost of a space mission or scenario typically involves three main components: DDT&E (Design, Development, Test, and Evaluation), launch cost (production and pre-launch processing), and mission operations costs (planning, training, and flight control). *SpaceNet 1.3* assumes that all instantiated elements are available for use and therefore ignores DDT&E[27]. *SpaceNet 1.3* is not a spacecraft/vehicle design tool, but analyzes the implications of using vehicles (elements) with certain specified capabilities. In other words, estimating DDT&E costs of flight hardware elements is outside the scope of *SpaceNet 1.3.*

Launch costs, however, are included and are assumed to be a direct function of *TLM*. The main driver of mission operations costs are labor-hours spent on mission planning and during a mission supervising or actively operating the instantiated elements, which reflects both the total mission time and the mission complexity. For each vehicle (= element) *SpaceNet 1.3* assumes an operating profile, whereby $\Gamma_{e,i} = 1$ if instantiated element *e* is operating during time period *i* and $\Gamma_{e,i} = 0$ otherwise. An element is non-active if it is *not* associated with at least one of the five *SpaceNet* processes (waiting, transporting, proximity operations, transferring crew/cargo, exploring) during a time period. An instantiated element can be active, i.e., $\Gamma_{e,i} = 1$, even when no crew is in that element or when crew is co-located in another element at the same node. The assumption for *RSC* is that an element will still have to be monitored and will therefore cause operations costs, even when it is waiting or pre-positioned. Lacking element specifications or another adequate measure of complexity, *SpaceNet 1.3* normalizes these two major cost components using weighting "prices" γ and δ. The relative scenario cost (*RSC*) equation is then a linear function of *TLM* and the temporal element operating profiles, Γ, as follows:

$$RSC = \gamma TLM + \delta \Delta T \sum_{e=1}^{EI} \sum_{i=1}^{T} \Gamma_{e,i} \tag{13}$$

TLM = total launch mass [kg] for the entire scenario from Eq. (11)
ΔT = Earth days per *SpaceNet* time period
EI = number of instantiated elements in a scenario
T = total number of time periods in the scenario
$\Gamma_{e,i}$ = 1 if the instantiated element *e* is "active" during time period *i*
γ = normalizing multiplier for *TLM*
δ = normalizing multiplier for operations complexity and time

The estimation of total scenario costs in terms of current monetary units [$] is outside the scope of *SpaceNet 1.3*. *RSC* can be used for a relative comparison of supply chain costs between exploration scenarios, but it is not suitable for budgetary planning purposes.

---

[27] Other tools are available for design and manufacturing cost estimation, see for example NASA JSC's cost estimation website: http://cost.jsc.nasa.gov/index.htm

When used in this manner, *RSC* is actually a *Laspeyres* quantity index with γ and δ serving as base period "prices." We again chose to normalize to Apollo 17—that is, specific values for γ and δ were established such that the *RSC* for Apollo 17 equals 1. These multiplier values are calculated assuming that launch costs and operations cost each contribute about half of the scenario costs. The resulting values are:

$$\gamma = 1.706 \cdot 10^{-7}$$
$$\delta = 0.0077$$

Total Scenario Risk (TSR)

Risk is a critical, if difficult to quantify, metric for evaluation of exploration scenarios. Detailed risk calculations should include probabilities of failure for different mission operations processes and elements, resulting in a mission risk cumulative distribution function (cdf), expected mission risk, and a time-phased risk profile. As a first iteration, however, *SpaceNet 1.3* uses known quantities readily computable using *SpaceNet* outputs to get a first-order measure of total scenario risk (*TSR*). This metric uses the probabilities of failure associated with each *SpaceNet* process.

SpaceNet produces a rough measure of risk for scenario *S* using Eq. (14).

$$TSR = 1 - \prod_{i \in S} (1 - P_i) \qquad (14)$$

P_i = Probability of failure during *SpaceNet* process *i*

The probabilities of failure of each mission event will depend in part on the particular elements used in the scenario. Transport processes (e.g., launch, and descent and landing) and proximity operations processes (e.g., rendezvous and docking, undocking and separation, and transposition) contribute significantly to overall campaign (scenario) risk.

It is intended that these probabilities eventually come from formal PRAs (Probabilistic Risk Assessments) such as those performed for Shuttle and ISS[28], but initially the *SpaceNet 1.3* user must provide such values for each instantiated process through the user interface. Refinements of *TSR* can be made by distinguishing among the probabilities of failure involving the loss of mission (LOM) and/or loss of crew (LOC).

---

[28] See, for example, O'Connor, Bryan, and Clayton Smith, "Probabilistic Risk Assessment of the International Space Station—Phase II", Futron Corporation, December 2000.

(this page intentionally left blank)

Appendix E: SpaceNet Error Conditions

Following is a list of error conditions checked during the simulation of a scenario in SpaceNet. Each error has a description of the error condition, the way the error is handled, and an example of the way the error would be logged. When running a simulation, the user can choose to end the simulation if any error occurs instead of handling (and logging) the error and continuing the simulation.

Errors common to all processes

The following error cases are checked for each simulation process.

Checking input logic

Check that the elements used in the process have not been discarded by being previously staged during a burn. If they have, an error is flagged and the process does not take place.

```
'Using Staged Elements <X>'
      'No [transport] was taken.'
```

Check that the elements used in the process are accounted for up to the time the process is scheduled to take place. If not, an error is flagged and the element's time history is updated.

```
'Element <X> is at time <T1> instead of time <T2>.'
      '[Transport] performed from time <T2>.'
```

Problems in Simulation

After mass of crew provisions consumed during the duration of the process is calculated, check if there is enough inventory of provisions mass. If not, an error is flagged and all available crew provisions mass is consumed. The same check is carried out for the crew operations equipment and spare parts.

```
'Crew Provisions not sufficient for element <X>.  Need an additional
<C> kg.'
      'All available provisions consumed.'

'Crew Operations not sufficient for element <X>.  Need an additional
<C> kg.'
      'All available crew operations used.'

'Spare parts not sufficient for element <X>.  Need an additional <C>
kg.'
      'All available spares parts used.'
```

Transport function (sn_transport)

Checking input logic

Check that there is an arc connecting the departure and arrival nodes. If not, an error is flagged and the transport does not take place.

```
'There is no arc connecting nodes <N1> and <N2>.'
      'No transport was performed.'
```

Check that the initial locations for all the elements in the stack are actually at the departure node. If any are not, an error is flagged and the transport does not take place for any element.

```
'Element <X> is not currently at departure node <N>.'
      'No transport was performed.'
```

Problems in simulation

When calculating fuel consumption, if there is not enough fuel in the burns specified, then an error is flagged and the transport takes place while consuming all available fuel.

```
'Transport from <N1> to <N2> by element <X> is currently infeasible in
burn <B>!  Insufficient propellant to execute this transport; add more
propellant or burn additional elements to provide a Delta V of <DV>
m/s.'
      'Transport completed with all available fuel consumed.'
```

Transfer function (sn_transfer)

Checking input logic

Check that all involved elements are at the same node. If not, an error is flagged and no transfer takes place.

```
'Element <X> is not currently at node <N>.'
      'No transfer was performed.'
```

Problems in simulation

Before allowing a transfer of crew or cargo, check that the first element has enough supply, and that the second element has enough capacity. If any of these checks find a problem, then log the appropriate errors, and carry out the transfer to the degree that is possible.

```
'Number of crew to transfer exceeds number currently in element <X> by
<C>. Edit transfer process and re-run simulation.'
```

```
      'All available crew transferred.'

'Mass of class of supply <S> to transfer exceeds mass currently in
element <X> by <C> kg. Edit transfer process and re-run simulation.'
      'All available cargo transferred.'

'Number of crew to transfer exceeds element <X> maximum crew capacity
by <C>. Edit transfer process and re-run simulation.'
      'As many crew transferred as possible.'

'Mass of cargo to transfer exceeds element <X> maximum fuel capacity by
<C> kg. Edit transfer process and re-run simulation.'
      'As much propellant transferred as possible.'

'Mass of cargo to transfer exceeds element <X> maximum cargo capacity
by <C> kg. Edit transfer process and re-run simulation.'
      'As much cargo transferred as possible.'
```

Wait function (sn_wait)

Checking input logic

Check that the locations for all the elements in the stack are actually at the wait node. If any are not, an error is flagged and the wait step does not take place for any element.

```
'Element <X> is not currently at node <N>.'
      'No wait step was performed.'
```

Exploration function (sn_exploration)

Checking input logic

Check that the locations for all the elements in the stack are actually at the exploration node. If any are not, an error is flagged and the exploration process does not take place for any element.

```
'Element <X> is not currently at node <N>.'
      'No exploration was performed.'
```

Proximity Operations function (sn_proximityops)

Checking input logic

Check that the locations for all the elements in the stack are actually at the node. If any are not, an error is flagged and the proximity operations process does not take place for any element.

```
'Element <X> is not currently at node <N>.'
      'No docking/rendezvous was performed.'
```

(this page intentionally left blank)

Appendix F: Automatic Manifesting

This Appendix describes the mathematics underlying automatic manifesting in SpaceNet 1.3 and the limitations of the current implementation:

1. Problem Setup

Nomenclature

$i = 1,...,n_E$: Element ID (n_E : # of elements)

$j = 1,...,n_C$: Item ID (n_C : # of supply items)

r_{ij} : Reward of assigning item *j* to element *i*

Can be calculated as followings:

$r_{ij} = R_{iJ} \cdot w_j$

R_{iJ} : Reward of assigning supply type J to element i,

where item *j* belongs to class of supply *J (Note: Only a subset of four classes of supply are considered in automatic manifesting; these are numbered here as J = 1: Crew Provisions, 2: Crew Operations, 3: Maintenance and Upkeep, 4: Exploration and Research)*

w_j : Weight (importance) of item *j* (currently all set to 1)

x_{ij} : 1, if item *j* is assigned to element *i*, 0 otherwise

m_i^C, v_i^C : Mass and volume capacity of element *i*

m_j, v_j : Mass and volume of item *j*

Integer Programming (IP) Formulation

$$\max \sum_{i=1}^{n_E} \sum_{j=1}^{n_C} x_{ij} r_{ij} \qquad (1)$$

subject to

$$\sum_{i=1}^{n_E} x_{ij} \le 1 \quad (\forall j = 1,\ldots,n_C) \qquad \text{(2) (Assigns an item to at most one element)}$$

$$\sum_{j=1}^{n_C} m_j x_{ij} \le m_i^C \quad (\forall i = 1,\ldots,n_E) \qquad \text{(3) (Element mass capacity constraints)}$$

$$\sum_{j=1}^{n_C} v_j x_{ij} \le v_i^C \quad (\forall i = 1,\ldots,n_E) \qquad \text{(4) (Element volume capacity constraints)}$$

$$x_{ij} \in \{0, 1\} \qquad \text{(5) (Non-negativity constraints)}$$

(Note: The actual formulation for the internal code is a little bit different from this to gain more efficiency. But the functionality is identical to the one described here.)

2. SYMPHONY

The problem formulated using equations (1) to (5) is an "Integer Program" and the nature of "integer solutions" makes the problems difficult to solve. There are several commercial software packages which can handle (Mixed) Integer (Linear) Programs (MILP), but these are usually very expensive to purchase. (CPLEX is one of them.)

SYMPHONY on the other hand is an *open-source software package which can be used to solve MILPs. It is being developed and updated by Dr. Ted Ralph, who is currently a professor in the Department of Industrial and System Engineering, Lehigh University, Bethlehem, PA.

*NOTE: Because SYMPHONY is open-source, the automatic manifesting code does not require the installation of any additional software.

Several complex integer programming problems have been successfully implemented in SYMPHONY including the Traveling Salesman Problem, Vehicle Routing Problem, Set Partitioning Problem, etc. More detailed explanations are available at the SYMPHONY homepage. (http://www.branchandcut.org)

3. Optimality Gap

First, keep in mind that the solution of an Integer Program can be achieved by adding more and more constraints to the relaxed version of a Linear Program. So in case of a minimization problem, a relaxed LP optimal value (allowing the decision variables to be continuous rather than integer) is always smaller than the true integer optimal value.

SYMPHONY continues branching until it finds a true optimum – that is, until the optimal LP value becomes the same as optimal IP value. During this branching process, we can get a "Best Feasible (Integer) Solution So Far", which is a "Lower Bound" of the optimal integer program objective function, (that is, the "True" optimum) and an "Upper Bound" of the optimal LP objective function, which is generally not feasible because the variables are not guaranteed to be integers.

When the two values coincide, we get a true optimum. But when we can allow a solution which is good enough, we can stop calculation when the difference of two values lies within a certain value and get the best feasible solution found so far. In this case, we can guarantee the accuracy of the solution using the upper bound of the optimal real-valued solution.

For example, assume that after a certain number of iterations, we get an upper bound of the optimal "Real-Value LP solution" whose objective function is 100.0, and "Best Feasible Solution found so far" whose objective function is 104.0. In this case, we can know that the "TRUE OPTIMAL SOLUTION" (which ensures that those decision variables that need to be integers are indeed integers) is no less than 100.0. If we ask

SYMPHONY to stop calculation when the optimality gap is within 5 %, it will stop calculation in this situation. And we would obtain the best feasible integer-variable objective value of 104.0, with a guarantee that worst case error between this solution (104.0) and true optimum is less than 4 %. (4% = (104-100)/100.

(this page intentionally left blank)

Appendix G: Database Dictionary

This Appendix contains a dictionary for the SpaceNet 1.3 integrated database.

Note: [K] indicates table key.

Table Name: Supply Classes
Table Theme: Identifies classes and subclasses of supply

Attribute Name	**Definition [units]**	**Nullable (T/F)**	**Valid Range**
Supply Class ID **[K]**	Unique identifier	F	Integer ≥1
Supply Class Name	Proper name of the supply class or subclass	T	Text
Supply Class Description	Brief description of the supply class or subclass	T	Text
Parent Supply Class ID	Supply Class ID of the parent supply class if this item is a subclass	T	Integer ≥1
Supply Class Level	Level in supply class hierarchy (e.g., level 1 for the ten basic classes of supply, level 2 for the immediate subclasses)	T	Integer ≥1
Comments		T	Text

Table Name: Physical Nodes
Table Theme: Identifies a maximal set of static nodes from which a scenario can be constructed

Attribute Name	**Definition [units]**	**Nullable (T/F)**	**Valid Range**
Node ID **[K]**	Unique identifier. First digit specifies central body (0 Sun, 1 Earth, 2 Moon, 3 Mars, 9 Lagrange pt). Last three digits in [500,999] indicates orbit node; in [001-499] indicates surface node.	F	Integer 0 – 9999
Node Name		T	Text
Node Type	Specifies that node is a surface location, orbit, or Lagrange point	F	‘Surface’, ‘Orbit’, or ‘Lagrange’
Body	Node central body (or one of the bodies if Lagrange point)	F	Text
Body 2	If node is Lagrange point, the second planetary body is specified here.	T	Text
Lat	[deg] Latitude (surface nodes only)	T	[-90,90]
Long	[deg] Longitude (surface nodes only)	T	[-180,180]
Ha	[km] Apoapsis altitude (orbit nodes only)	T	≥0
Hp	[km] Periapsis altitude (orbit nodes only)	T	≥0
Inclination	[deg] (orbit nodes only)	T	[0,180]
LP Number	Lagrange point number. ‘1’ refers to the point located between the two bodies, ‘2’ to the point directly opposite ‘1’ on the far side of the second body.	T	1,2,3,4,5
Node Description		T	Text

Table Name: Astro
Table Theme: Identifies a set of feasible arcs (trajectories) and associated attributes based on astrodynamics

Attribute Name	Definition [units]	Nullable (T/F)	Valid Range
Arc ID **[K]**	Unique identifier for each pair of nodes. Combine with N_Traj for unique table key.	F	Integer ≥1
N_Traj **[K]**	Unique identifier for each trajectory between same pair of nodes. Combine with Arc ID for unique table key.	F	Integer ≥1
Departure Node ID	FK reference to Table Name: Physical Nodes	F	Integer ≥0
Arrival Node ID	FK reference to Table Name: Physical Nodes	F	Integer ≥0
Type	Type of arc: (1) Arc represented by single values of DV and TOF. (2) Arc with tradeoff between TOF and DV, represented by a 1-d table. (3) Arc with tradeoff between TOF, DV, and departure time, represented by a matrix.	F	1,2,3
N_Burn	Number of burns in trajectory	F	1,2
Time of Flight	[days]	F	≥0
DV_Req1	[m/s] Delta-V requirement for first burn	F	≥0
DV_Req2	[m/s] Delta-V requirement for second burn	T	≥0
Comment_1		T	Text
Comment_2		T	Text
Comment_3		T	Text

Table Name: Supply Item Type
Table Theme: Identifies common attributes of a supply item type

Attribute Name	Definition [units]	Nullable (T/F)	Valid Range
Supply Item Type ID **[K]**	Unique Identifier	F	Integer ≥1
Supply Item Type Name	Name of the supply item	F	Text
Supply Item Type Description	Description of the supply item	T	Text
Supply Class ID	FK reference to Table Name: Supply Classes	F	Integer ≥1
Mass	[kg] Mass of the item at launch	F	≥0
Volume	[cm^3] External volume of the item at launch	F	≥0
Length	[cm] Length of the item in its launch configuration	T	≥0
Width	[cm] Width of the item in its launch configuration	T	≥0
Height	[cm] Height of the item in its launch configuration	T	≥0
Life Span Type ID	FK reference to Table Name: Life Span Type	T	Integer ≥0
Life Span Parameter	The number of life span parameter units that define this items life span	T	≥0
Life Span Parameter Units ID	FK reference to Table Name: Unit Type	T	Integer ≥1

Attribute Name	Definition [units]	Nullable (T/F)	Valid Range
Priority Level ID	FK reference to Table Name: Priority Level	T	Integer ≥0
Hazard Type ID	FK reference to Table Name: Hazard Type	T	Integer ≥0
Hazard Level ID	FK reference to Table Name: Hazard Level	T	Integer ≥0
Handling Level ID	FK reference to Table Name: Handling Level	T	Integer ≥0
Packaging Type ID	FK reference to Table Name: Packaging Type	T	Integer ≥0
Packaging Mass	[kg] Mass of packaging not accounted for in supply item type mass	T	≥0
Shipping Environment ID	FK reference to Table Name: Shipping Environment	T	Integer ≥0
Storage Environment ID	FK reference to Table Name: Storage Environment	T	Integer ≥0
Disposal Type ID	FK reference to Table Name: Disposal Type	T	Integer ≥0
OEM ID	FK reference to Table Name: Suppliers	T	Integer ≥0
Procurement Type ID	FK reference to Table Name: Procurement Type	T	Integer ≥0
Procurement Lead Time	[days] The lead time required to procure this item	T	≥0
Reprocurement Cost	[$FY07K] The cost of reprocuring this item	T	≥0
Comments		T	Text

Table Name: Element Type
Table Theme: Identifies common attributes of an element type

Attribute Name	Definition [units]	Nullable (T/F)	Valid Range
Element Type ID **[K]**	Unique Identifier	F	Integer ≥1
Element Type Name	Name of the element type	F	Text
Supply Class ID	FK reference to Table Name: Supply Classes	F	Integer ≥1
Partner ID	FK reference to Table Name: Partners	T	Integer ≥0
Length	[m] Length of the element at launch	T	≥0
Width	[m] Width of the element at launch	T	≥0
Height	[m] Height of the element at launch	T	≥0
Diameter	[m] Diameter of the element at launch	T	≥0
Dry Mass	[kg] Dry Mass of the element at launch	F	≥0
Primary Isp	[s] Primary Isp of the element	T	≥0
Secondary Isp	[s] Secondary Isp of the element	T	≥0
Thrust	[N] Maximum thrust of the element	T	≥0
Burn Time	[s] Maximum burn time of the element	T	≥0
Max Cargo Up	[kg] Cargo capacity of the element at launch	F	≥0
Max Cargo Ret	[kg] Return to Earth cargo capacity of the element	F	≥0

Attribute Name	Definition [units]	Nullable (T/F)	Valid Range
Payload Volume	[m^3] Volume available for payload storage at launch	F	≥0
Pressure	[Pa] Pressure inside the element	T	≥0
Pressurized Volume	[m^3] Volume of the pressurized portion of the element	T	≥0
Crew Capacity	Maximum number of crew that the element was designed to accommodate	F	Integer ≥0
Accommodation Mass	[kg] Mass of accommodation hardware in the element	T	≥0
FSE Overhead	[kg] Mass of FSE overhead in the element	T	≥0
Attach Fittings / Adapters	[kg] Mass of attachment fitting and adapters in the element	T	≥0
Propellant Type ID	FK reference to Table Name: Supply Item Type	F	Integer ≥1
Max Propellant	[kg] Maximum amount of propellant the element can hold	F	≥0
Max Usable Propellant	[kg] Maximum propellant minus ullage	F	≥0
Max Transferable Propellant	[kg] Maximum mass of transferable propellant the element can hold	T	≥0
Max Fluids	[kg] Maximum fluid capacity of the element	T	≥0
Max Gases	[kg] Maximum gas capacity of the element	T	≥0
Max Press Dry Cargo	[kg] Maximum mass of pressurized dry cargo the element can hold	F	≥0
Max Unpress Dry Cargo	[kg] Maximum mass of unpressurized dry cargo the element can hold	T	≥0
In-Space Life	[years] In-space life span of the element	T	≥0
Number Available	The number of this element available	T	≥0
Recoverable	True, if this element is recoverable	T	T/F
Element Non-Recurring Cost	[$FY07M] Non-recurring costs of launching this element	T	≥0
Element Recurring Cost	[$FY07M] Recurring costs of launching this element	T	≥0
First Element Processing	[workyears] Ground processing time to launch this element the first time	T	≥0
Learning Rate	[%] Unit learning curve parameter for ground processing workyears (i.e., ratio of second unit ground processing workyears to first unit ground processing workyears	T	[0,1]
Min Processing Time	[days] Minimum processing turn around time for this element	T	≥0
OEM ID	FK reference to Table Name: Suppliers	T	Integer ≥0
Spaceport Node ID	FK reference to Table Name: Physical Nodes	F	Integer ≥0
Launch Stack	True, if this “element” is an allowable launch stack (i.e., a composite of several elements) (for optimization use)	F	T/F
Comment_1		T	Text
Comment_2		T	Text

Table Name: Stacks
Table Theme: Identifies composition of stack types (all key table for optimization use)

Attribute Name	**Definition [units]**	**Nullable (T/F)**	**Valid Range**
Stack Type ID **[K]**	FK reference to Table Name: Element Type	F	Integer ≥1
Element Type ID **[K]**	FK reference to Table Name: Element Type	F	Integer ≥1

Table Name: Crew Provisions
Table Theme: Captures predicted usage rates

Attribute Name	**Definition [units]**	**Nullable (T/F)**	**Valid Range**
Supply Item Type ID **[K]**	FK reference to Table Name: Supply Item Type	F	Integer ≥1
Partner ID **[K]**	FK reference to Table Name: Partners	F	Integer ≥0
Usage Rate Predict (LEO)	Predicted/estimated usage rate value	F	≥ 0
Usage Rate Predict (Lunar)	Predicted/estimated usage rate value	F	≥ 0
Usage Rate Predict (Mars)	Predicted/estimated usage rate value	F	≥ 0
Usage Rate Units ID	FK reference to Table Name: Unit Type	F	Integer ≥1
Application	Male, Female, or Unisex	T	M,F,U
Inventoried	True, if carried in inventory and managed by partner	T	T/F
Comments		T	Text

Table Name: Crew Operations
Table Theme: Captures predicted usage rates

Attribute Name	**Definition [units]**	**Nullable (T/F)**	**Valid Range**
Supply Item Type ID **[K]**	FK reference to Table Name: Supply Item Type	F	Integer ≥1
Partner ID **[K]**	FK reference to Table Name: Partners	F	Integer ≥0
Usage Rate Predict (LEO)	Predicted/estimated usage rate value	F	≥ 0
Usage Rate Predict (Lunar)	Predicted/estimated usage rate value	F	≥ 0
Usage Rate Predict (Mars)	Predicted/estimated usage rate value	F	≥ 0
Usage Rate Units ID	FK reference to Table Name: Unit Type	F	Integer ≥1
ORU	True, if an ORU	T	T/F
Inventoried	True, if carried in inventory and managed by partner	T	T/F
Comments		T	Text

Table Name: Parts: Common

Table Theme: Identifies common attributes for ORUs/SRUs

Attribute Name	**Definition [units]**	**Nullable (T/F)**	**Valid Range**
Supply Item Type ID **[K]**	Unique identifier	F	Integer ≥1
Partner ID	FK reference to Table Name: Partners	T	Integer ≥0
Part National Stock Number (NSN)	National Stock Number, or equivalent	T	Text
Part Type ID	FK reference to Table Name: Part Type	F	Integer ≥1
System	System or subsystem, if applicable, e.g., EPS	T	Text
Function	Subsystem or functional description, e.g., PMAD	T	Text
Developer ID	FK reference to Table Name: Suppliers	T	Integer ≥0
SRU	True, if SRU	F	T/F
Inventoried	True, if carried in inventory and managed by partner	T	T/F
Logistics Carrier Type ID	FK reference to Table Name: Element Type	T	Integer ≥1
Footprint	[m^2] Area for external in-space transportation/storage	T	≥0
LP Delay	[days] Launch pad delay	T	≥0
Initial Stock	Quantity held in inventory (all nodes) at start of simulation	T	≥0
Orbit Stock	Quantity held in inventory (in-space nodes) at start of simulation	T	≥0
Min Grd Stock	Minimum desired quantity held in ground inventory	T	≥0
RTOK	Retest-OK rate; the proportion of returned and retested ORUs that indicate no failure	T	[0,1]
Retest Cost	[$FY07K] Retest cost	T	≥0
RTTrans	[$FY07K] Round-trip ground transportation cost from inventory site to depot site	T	≥0
NRTS Rate	Not-reparable-this-station rate	F	[0,1]
Condemn Rate (IMF)	Condemnation rate at the intermediate maintenance level	F	[0,1]
Condemn Rate (Depot)	Condemnation rate at the depot maintenance level	F	[0,1]
Condemn Rate (LEO)	Condemnation rate at the organizational maintenance level in low Earth orbit	F	[0,1]
Condemn Rate (Lunar)	Condemnation rate at the organizational maintenance level on the lunar surface	F	[0,1]
Condemn Rate (Mars)	Condemnation rate at the organizational maintenance level on Mars surface	F	[0,1]
Comments		T	Text

Table Name: Parts: Application-Specific
Table Theme: Identifies attributes for ORUs dependent on a particular application or installed location within an element type

Attribute Name	Definition [units]	Nullable (T/F)	Valid Range
Part ID **[K]**	Unique identifier	F	Integer ≥1
Supply Item Type ID	FK reference to Table Name: Supply Item Type	F	Integer ≥1
Installed Location	ORU location description (e.g., airlock, internal) in human-readable form (primarily for ISS use)	T	Text
Element Type ID	FK reference to Table Name: Element Type	F	Integer ≥1
Manifested Flight	Assembly flight designation (primarily for ISS use)	T	Text
Activated Flight	Activation flight designation (primarily for ISS use)	T	Text
Internal/External	Internal = 0; External = 1 (used to determine k-factor)	F	0,1
Criticality	Short text for ORU application criticality	F	1,1R,1S,1SR, 1P,2,2R,3
Primary CM Task Type	Short text describing most often used nature of corrective maintenance (e.g., by remove-and-replace or by SRU replacement); correlates with RIO	T	CM/PM_RR, CM_SRU
Quiescent DC	Duty cycle during quiescent or uncrewed state	F	[0,1]
Active DC	Duty cycle during active or crewed state	F	[0,1]
Assembly DC	Duty cycle during assembly state	T	[0,1]
Dormant FR	Dormant failure rate; proportion of ORUs that fail when switched on or activated	T	[0,1]
MTBF	[hours] Inherent ORU mean time between failures	F	≥0
LifeLim	[years] Weibull distribution scale (characteristic life) parameter (primarily for ISS use)	T	≥0
LifeBeta	Weibull distribution shape parameter (primarily for ISS use)	T	≥0
Test Time	[hours] Testing time for an ORU pulled from inventory and manifested	T	≥0
RIO	Repair-in-orbit rate; proportion of units of an ORU that are repaired by SRU replacement at the organizational maintenance level	F	[0,1]
Quantity Per Application (QPA)	ORU quantity per application	F	>0
Redundant	True, if ORU application has built-in (stand-by) redundancy	T	T/F
Integrator	Organization responsible for integration (primarily for ISS use)	T	Text
Comments		T	Text

Table Name: Maintenance Tasks
Table Theme: Identifies all maintenance tasks and frequency of scheduled maintenance tasks for each ORU application

Attribute Name	**Definition [units]**	**Nullable (T/F)**	**Valid Range**
Task ID **[K]**	Unique identifier	F	Integer ≥1
Part ID	FK reference to Table Name: Parts: Application-Specific	F	Integer ≥1
Task Type	Short text describing nature of task (e.g., corrective maintenance by remove-and-replace)	F	CM/PM_RR, CM_SRU, PM_SI
Task Mode Type ID	FK reference to Table Name: Task Mode Type	F	Integer ≥0
MTBSM	[days] For a schedule maintenance task, the mean time between performance of the task	T	≥0

Table Name: Maintenance Resources
Table Theme: Identifies resources types, quantities, and times needed for each maintenance task

Attribute Name	**Definition [units]**	**Nullable (T/F)**	**Valid Range**
Task Resource ID **[K]**	Unique identifier	F	Integer ≥1
Task ID	FK reference to Table Name: Maintenance Tasks	F	Integer ≥1
Crew Type ID	FK reference to Table Name: Crew Type	F	Integer ≥1
Crew Quantity	Number of crewpersons for task	F	>0
MTTR	[hrs] Mean time to repair (i.e., time at worksite for task)	F	>0
OH Time	[hrs] Overhead time for task (i.e., preparation, translation to/from worksite, and cleanup)	T	≥0
Robotics Quantity	Number of robotic devices for task	T	≥0
Robotics MTTR	[hrs] Robotic mean time to repair (i.e., time at worksite for task)	T	≥0
Robotics OH Time	[hrs] Robotic overhead time for task (i.e., preparation, translation to/from worksite, and cleanup)	T	≥0

Table Name: Task Mode Type
Table Theme: Identifies attributes of ORU/SRU maintenance task modes

Attribute Name	**Definition [units]**	**Nullable (T/F)**	**Valid Range**
Task Mode Type ID **[K]**	Unique identifier	F	Integer ≥0
Task Mode Description	Short text for primary corrective maintenance approach (e.g., IVA, EVA, etc.) in human-readable form	T	Text
Primary Robotics Type ID	FK reference to Table Name: Element Type	T	Integer ≥1
Secondary Robotics Type ID	FK reference to Table Name: Element Type	T	Integer ≥1
Comments		T	Text

Table Name: Part Type
Table Theme: Identifies attributes of ORU/SRU reliability classes, including K-factors

Attribute Name	Definition [units]	Nullable (T/F)	Valid Range
Part Type ID	Unique identifier	F	Integer ≥1
Part Type Name	Part type description	T	Text
Part Type Abbreviation	Short text for part type description (primarily for ISS use)	T	ET,EE,EM ME,SW,SO
Internal K-Factor	Multiplier to account for increased maintenance actions not included in MTBF estimates for internal ORUs/ SRUs (e.g., human-induced damage, environment-induced damage, equipment-induced damage, false or incorrect maintenance)	F	≥1
External K-Factor	Multiplier to account for increased maintenance actions not included in MTBF estimates for external ORUs/ SRUs (e.g., human-induced damage, environment-induced damage, equipment-induced damage, false or incorrect maintenance)	F	≥1

Table Name: Partners
Table Theme: Identifies specific international partners and responsible governmental organizations

Attribute Name	Definition [units]	Nullable (T/F)	Valid Range
Partner ID **[K]**	Unique identifier	F	Integer ≥0
Partner Name	Short name for partner or governmental organization	T	Text
Partner Location	Short partner location description, e.g., address	T	Text
Partner Status ID	FK reference to Table Name: Org Status	T	Integer ≥0
Comments		T	Text

Table Name: Suppliers
Table Theme: Identifies specific supply item suppliers, including OEMs

Attribute Name	Definition [units]	Nullable (T/F)	Valid Range
Supplier ID **[K]**	Unique identifier	F	Integer ≥0
Supplier Name	Short name for supplier or source organization	T	Text
Supplier Location	Short supplier location description, e.g., address	T	Text
Supplier Status ID	FK reference to Table Name: Org Status	T	Integer ≥0
Comments		T	Text

Table Name: Unit Type
Table Theme: Identifies types of units used

Attribute Name	Definition [units]	Nullable (T/F)	Valid Range
Unit Type ID **[K]**	Unique identifier	F	Integer ≥1
Unit Type Name	Units name (use ‘/’ instead of ‘per’)	F	Text

(this page intentionally left blank)

Appendix H: Apollo 17 Reference Data

This Appendix summarizes reference data for the Apollo 17 mission[29] because it is used as a reference scenario in *SpaceNet,* against which any other scenario can be compared. In other words, Apollo 17 serves as the normalization baseline for SpaceNet 1.3. The MOE's listed in this Appendix are defined in Appendix D.

CSD_{tot}^{a17} = 2 crew members for 74 hrs 59 min 40 sec on the lunar surface = **6.25** [crew-days]

EMD_{tot}^{a17} = mass of LRV[30] + mass of ALSEP[31] + mass of field geology tools[32] = 462 lbs + 360 lbs + 100 lbs (est.) = 922 lbs = **415** [kg]

Note that the lunar roving vehicle (LRV) is in COS 802, while the Apollo Lunar Surface Exploration Package (ALSEP) is in COS 604 and the geology tools are in COS 602.

$EC_{tot}^{A17} = CSD_{tot}^{a17}$ x EMD_{tot}^{a17} =~ **2594** [kg crew-days]

This captures at a high level the capability of the exploration supply chain for Apollo 17 (benefit). This must be balanced against the mass that had to be launched from Earth's surface to achieve this capability (i.e., *TLM* for Apollo 17)[33]:

TLM_{A17} = 6,529,784 [lbs] ~ **2,930,702 [**kg]

An approximation to Eq. (12) in Appendix D can be made applying Eqs. (H1) and (H2).

$$\eta_{a17} = \frac{EC_{tot}^{a17}}{TLM_{a17}} = 8.85 \cdot 10^{-4} \qquad \text{(H1)}$$

$$REC_b \cong \frac{EC_{tot}^{b} / TLM_b}{\eta_{a17}} \qquad \text{(H2)}$$

In order to do the more detailed calculations required by the *Divisia index* in Appendix D, Eq. (12), we require a detailed mass breakdown by class of supply. This breakdown has to be estimated since the Apollo program did not use the COS system shown in Appendix

---

[29] Source: http://history.nasa.gov/SP-4029/Apollo_17i_Timeline.htm

[30] LRV = Lunar Roving Vehicle: http://www.hq.nasa.gov/alsj/lrvhand.html

[31] ALSEP = Apollo Lunar Surface Exploration Package: http://www.astronautix.com/craft/apoalsep.htm

[32] Geology and Soil Sampling Tools: http://history.nasa.gov/SP-4029/Apollo_18-35_Geology_and_Soil_Tools.htm

[33] Total Launch Mass at the pad Apollo 17, December 5, 1972 (ground ignition): http://history.nasa.gov/SP-4029/Apollo_18-19_Ground_Ignition_Weights.htm

A. The approximate Total Launch Mass (TLM) breakdown for Apollo 17 is as in Table H-1:

Table H-1: Apollo 17 Launch mass breakdown [lbs]

Weights In Pounds Mass	**Apollo 17**
Ground Ignition Weights December 5, 1972	
S-IC stage, dry	287,356
S-IC stage, fuel	1,431,921
S-IC stage, oxidizer	3,314,388
S-IC stage, other	5,395
S-IC stage, total	**5,039,060**
S-IC/S-II interstage, dry	**9,975**
S-II stage, dry	80,423
S-II stage, fuel	844,094
S-II stage, oxidizer	160,451
S-II stage, other	934
S-II stage, total	**1,085,902**
S-II/S-IVB interstage, dry	**8,019**
S-IVB stage, dry	25,040
S-IVB stage, fuel	43,752
S-IVB stage, oxidizer	195,636
S-IVB stage, other	1,658
S-IVB stage, total	**266,086**
Total Instrument Unit	**4,470**
Spacecraft/Lunar Module Adapter	3,961
LM	36,262
Command and Service Module	66,942
Total Launch Escape System	9,104
Total Spacecraft	**116,269**
Total Vehicle	**6,529,784**

This can be roughly translated into the COS system used in *SpaceNet 1.3* (using some estimated COS assignments) in order to approximate the total mass by COS and by element at the time of the Apollo 17 launch, see Table H-2.

Table H-2: Approximate initial COS distribution for Apollo 17, units of [kg]

	e=1	*e=2*	*e=3*	*e=4*	*e=5*	*e=6*	*e=7*	*e=8*	**Total**
COS	S-IC	S-II	S-IVB	LAS	CM	SM	LM DS	LM AS	
k=0	0	0	0	0	300	0	0	0[34]	**300**
k=1	2,135,839	452,045	107,725	0	75	18,413	8,804	2,358	**2,725,259**
k=2	0	0	0	0	350	57	0	100	**507**
k=3	0	0	0	0	144	0	0	96[35]	**240**
k=4	0	0	0	0	10	0	0	10	**20**[36]
k=5	0	0	0	0	10	0	20	10	**40**[25]
k=6	0	0	0	0	0	0	207	0	**207**
k=7	0	0	0	0	10	0	0	10	**20**[25]
k=8	0	0	0	0	0	0	208	0	**208**
k=9	133,982	38,415	12,014	4,097	4,841	6,053	2,770	1,719	**203,891**
k=10	0	0	0	0	0	0	0	10	**10**[25]
Total	**2,269,821**	**490,460**	**119,739**	**4,097**	**5,740**	**24,523**	**12,009**	**4,313**	**2,930,702**

Some of the data in Table H-2 had to be approximated, but the total masses by vehicle are accurate to within a few % from actual values. The mass breakdown by COS, m_{COSk}^{a17} , is shown in the rightmost column of Table H-2. This allows calculation of the mass fraction by class of supply. These are required for Eq. (12) in Appendix D:

Table H-3. COS mass fractions for Apollo 17, ω_k^{a17}

k=0	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
1.024 10^{-4}	0.930	1.730 10^{-4}	0.819 10^{-4}	0.068 10^{-4}	0.136 10^{-4}	0.706 10^{-4}	0.068 10^{-4}	0.710 10^{-4}	0.069	0.034 10^{-4}

The result of these calculations is very clear. Nearly 93% of the total mass launched by Apollo 17 consisted of propellant (fuel + oxidizer), while about 7% represented vehicle dry mass. The actual supply items carried along (including the crew) represent only about 0.1% of the total launch mass.

Figure H-1 shows the main elements of the Apollo 17 mission (Saturn V launch vehicle and its stages S-IC, S-II and S-IVB are not shown):

---

[34] During lunar descent two crew members transfer into the LM AS. We conservatively use 2 x 100 kg = 200 kg for the crew mass, while the actual crew mass in Apollo 17 without suits was approximately 144 kg.

[35] Each Apollo astronaut had three custom fitted A7L suits - one for flight, one for training, and one for flight back-up. The Apollo suit weighed 22 kg and its PLSS Portable Life Support System, 26 kg. The A7LB modification was used for Apollo J series lunar landing missions. Apollo 17 was the last of the J missions.

[36] Estimate for modeling purposes.

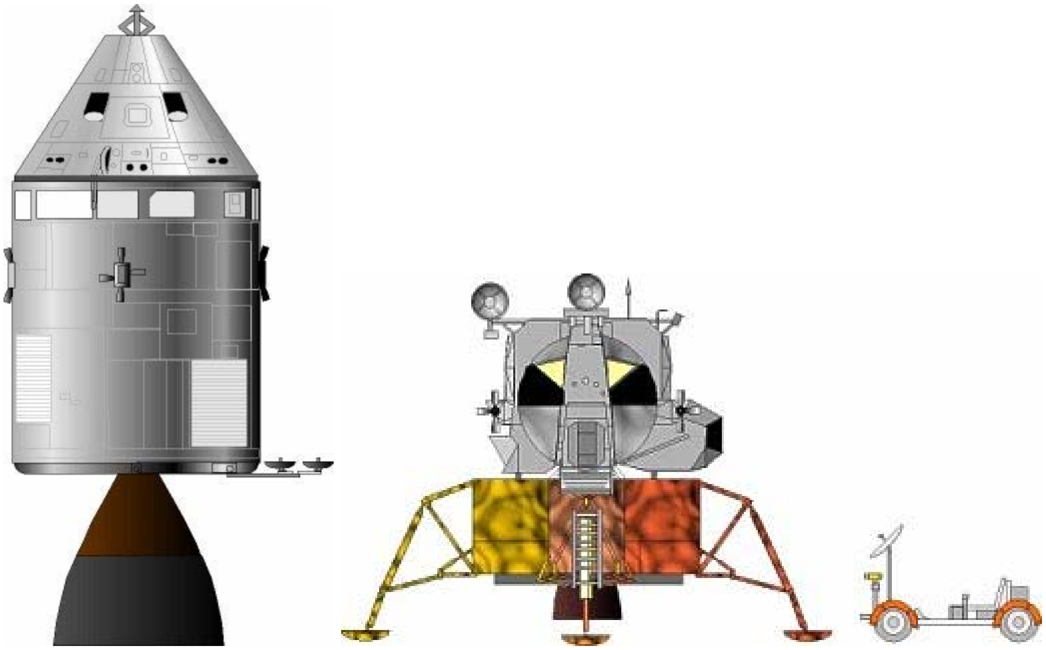


Figure H-1: (left) Command and Service Module (CSM), (middle) Lunar Module with ascent and descent stages shown, (right) Lunar Roving Vehicle (LRV). Figure credit: Mark Wade ©, www.astronautix.com

Appendix I: Known Limitations

This Appendix contains a known list of limitations and bugs in *SpaceNet 1.3*. Some of these are based on simplified assumptions and abstractions that were made to enable the general interplanetary modeling framework, others would require additional time and effort to clear up. It is the intention to remove these limitations in *SpaceNet 2.0* (web-based version) and any future releases of SpaceNet 1.3, as applicable.

1. Running SpaceNet

- SpaceNet 1.3 startup in Matlab (after typing "SpaceNet" at the command prompt, with the current directory in …\spacenet_1.3\gui) is relatively slow and takes about 10 seconds. This is mainly due to the fact that the Excel integrated database is being read at that time.

- Currently SpaceNet 1.3 is a local application on a single computer, and it has no password protection or user authentication

- User guidance and help features in SpaceNet 1.3 are rather limited. The best way to learn SpaceNet 1.3 (beyond doing the simple Demo example described in Chapter 1) is to read the User Manual and study the existing scenarios. Additional help may be available at: http://spacelogistics.mit.edu

- SpaceNet 1.3 requires Matlab Version 7.0.1.24704 (R14) Service Pack 1 or higher. The software has been tested on Windows XP, but not on other operating systems (MAC OS, Linux).

- Trade Study Mode and Batch mode have been implemented, but have undergone only limited testing.

2. Database

- The existing SpaceNet 1.3 database exists in Excel and is saved at the following location: …\spacenet_1.3\database\IP_SCM_Database.xls. The database is not password protected.

- The ISCMLA team has undertaken a significant effort to ensure that the database is accurate and up to date. However, the data in this database has not been officially validated by NASA, and some data, such as dry masses of the Constellation elements (CEV CM, SM, LSAM), are likely to continually evolve as the design of these elements matures.

- The database (and SpaceNet 1.3 in general) contains the capability (provisions for) modeling Mars missions and campaigns. While Mars surface and orbit nodes are contained in the database, the Mars trajectory information (type 3 trajectories) has not been populated in the database. A Mars scenario has not yet been created and tested in SpaceNet 1.3.

3. Scenario Building/Processes

- The current limitations allow the building of Earth-only and Earth-Moon scenarios.

- SpaceNet 1.3 has no “stack compatibility” constraints and does not check whether two particular elements can actually be used in the same process together or operated in a particular stack sequence based on engineering constraints.

- During manifesting, cargo can be packed and manifested in individual bags and containers for initial launch manifesting (manually or through auto-manifesting). In the remainder of SpaceNet 1.3., however, individual instances of supply items are not tracked. Rather, SpaceNet 1.3 aggregates the total mass [kg] of a particular class of supply and treats it as a continuous quantity for consumption simulation and processes like transfers.

- Push “Save Scenario” frequently when building or editing a scenario

4. Demand Models and Manifesting

- The demand models for spares (COS 4) in SpaceNet 1.3 are limited to the ORU data that is stored and available in the database. Currently only ORU data for the LSAM AS and Hab Lander is available. This set will be expanded in the future.

- Currently no demand model exists for COS 1 (fuels and propellants) for fuels that are *not* directly used as fuels for propulsion in propulsive elements. In some cases fuel might be needed for propelling rovers, heating habitats etc….COS 1 does contain a basic boiloff model. Also there is no demand model for COS 10 (miscellaneous). The non-existent demand models are grayed out on the demand GUI.

- Auto-Manifesting works quite well for scenarios of small and intermediate size (< 20 elements, <200 items to manifest). For larger problem instances either the optimality gap has to be increased (>2%) or longer running times (>10 sec) have to be allowed. Even when converged, in some cases auto-manifesting does not manifest items that would still fit on an element after manual inspection.

- Auto-Manifesting only takes into account the initial conditions at launch and does not attempt to minimize the subsequent number of transfers or considerations of downstream consumption in space. The purpose of auto-manifesting is to lighten the workload of the user. It does not always build a perfect manifest.

- Sometimes shortages can occur even if the total demand is satisfied for the scenario as a whole. This can occur when there are local mismatches between supply and demand in one or more processes.

- SpaceNet 1.3 does not have an explicit capability to model ISRU (In Situ resource Utilization)

5. Simulation

- The current MOEs computed by SpaceNet 1.3 (see Appendix D) assume that exploration processes will be associated with surface nodes (e.g. on the Moon). Therefore in order to compute MOEs such as CSD and EMD, SpaceNet 1.3 looks for mass delivered and crew time spent at surface nodes. MOE calculations for exploration processes at orbital nodes or Lagrange nodes are not currently supported.

- The value delivered by robotic agents (non-human crew) is not currently captured.

- The risk measure, TSR, is very simple and should not – without caution – be applied to compare the absolute risk of different scenarios. Rather TSR is meant as a relative metric capturing scenario complexity.

- The list of typical errors that SpaceNet 1.3 will detect is given in Appendix E. These errors can be seen in the _Log.txt file. SpaceNet 1.3 does not automatically repair such errors, nor does it provide extensive guidance to the user on how to do so. Experience using SpaceNet 1.3 will help avoid errors and remove the ones that do exist in particular scenarios.

- SpaceNet 1.3 does not model probabilistic events.

6. Visualization

- The network visualization may be a bit jumpy and may produce somewhat unnatural stacking sequences for very large campaign scenarios, e.g. buildup of a lunar outpost.

- The .avi files produced by the network visualization can get quite large, sometimes exceeding 100 MB. More intelligent compression and other visualization technologies (e.g. Adobe Flash) will be investigated in the future.

- When using a screen or projector with limited resolution (e.g. 600 x 800 pixels), only a portion of the network visualization may be visible. The network visualization cannot easily be rescaled in this release.

- Bat visualization may plot some element labels on top of each other

- When an Excel time history file is produced by simulation (a checkbox option on the Simulation GUI screen), the resulting .xls file should always be saved to the local …\spacenet_1.3\scenario folder so that the .xls file may later be accessed by SpaceNet 1.3 (currently no ability to browse other folders)

7. Optimization

- In order to run optimization a local copy of CPLEX has to be installed; the environment path variable has to be set such that CPLEX may be launched from the command line

- In order to run optimization at a minimum one exploration process has to be defined, and the "Calculate Optimization Demands" button has to be pushed before running optimization

- After optimization has finished (acknowledge the popup window), SpaceNet 1.3 must be prompted to read the optimization output files. Once this has occurred a new file … with the designation "_Opt" will be generated and SpaceNet 1.3 will shut down (this is planned) so that the optimization-generated scenario may be loaded.

- The results produced by optimization should meet all propulsive requirements, but may in some cases produce errors, particularly related to crew provisions and crew operations. An example is given below:

```
Scenario Name                   : Apollo_17_Opt
Simulated on                    : Jan.17,2007 01:31:32
Number of Instantiated Elements : 9
Mission Start Time              : 05-Dec-1972
Mission End Time                : 21-Dec-1972
Number of Processes             : 22
------------------------------------

Error Log


Crew provisions not sufficient for element LM AS (2). Need an additional
5.4849 kg.

  All available crew provisions consumed.
```

```
Crew operations not sufficient for element LM AS (2). Need an additional
0.1 kg.
  All available crew operations consumed.
```

- In such cases an optimization scenario must be manually “repaired” by increasing cargo, or modifying transfer processes and initial manifests, as the situation demands.
- Optimization may generate some partially disjointed and “orphan” processes
- Some optimization-generated processes may be shifted by one or more time periods

Please submit any SpaceNet 1.3 bugs, problems, errors and suggestions with the online submission form at http://spacelogistics.mit.edu ; click on **SpaceNet Feedback**.

(this page intentionally left blank)

User Notes