Comparison of an Expendable and a Reusable Lunar Exploration Architecture Using the Lunar Architecture Stochastic Simulator and Optimizer (LASSO)

AE8900 MS Special Problems Report
Space Systems Design Lab (SSDL)
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA

Author
Kristina Alemany

Advisor
Dr. John R. Olds
Space Systems Design Lab (SSDL)

May 5, 2005

Table of Contents

List of Figures

List of Tables

Acronyms

CAIB	Columbia Accident Investigation Board
CES	Crew Escape System
CEV	Crew Exploration Vehicle
CMRG	Combined Multiple Recursive Generator
DDTE	Design Development Test & Evaluation
DES	Discrete-Event Simulation
EELV	Evolved Expendable Launch Vehicle
ET	External Tank
FIFO	First-In First-Out
GUI	Graphical User Interface
HLV	Heavy-lift Launch Vehicle
KSC	Kennedy Space Center
LASSO	Lunar Architecture Stochastic Simulator and Optimizer
LEO	Low Earth Orbit
LLO	Low Lunar Orbit
LOC	Loss of Crew
LOM	Loss of Mission
LV	Launch Vehicle
OEC	Overall Evaluation Criterion
RLV	Reusable Launch Vehicle
RNG	Random Number Generator
SRB	Solid Rocket Booster
TAT	Turn-Around Time
TEI	Trans-Earth Injection
TFU	Theoretical First Unit
TLI	Trans-Lunar Injection
VBA	Visual Basic for Applications

1.0 Introduction

In the President's Vision for Space Exploration, President Bush called for a return to the Moon no later than the year 2020[1]. In order to do so, however, it is necessary to determine the best architecture and suite of vehicles within that architecture in order to accomplish the President's Vision. Historically, many of these decisions have been made based on deterministic mass and performance-based analyses. Metrics such as cost, reliability, and the ability to meet a given campaign schedule have been considered later in the design process after many architecture and vehicle decisions have already been made, generally increasing the overall cost of the mission. Therefore, the ability to rapidly measure cost, reliability, and schedule impacts of top-level architecture and individual element decisions represents a significant improvement over the current deterministic analysis capabilities for top-level decision making. Allowing this knowledge to be brought forward in the design process will help to reduce the overall program costs down the road.

This capability is provided by LASSO, the Lunar Architecture Stochastic Simulator and Optimizer, which was developed in order to conduct this study. LASSO is a simulation-based capability, based upon discrete-event simulation (DES), that provides the ability to probabilistically simulate and optimize an overall lunar transportation approach. Discrete-event simulation is a fairly new tool to the space industry, as it has been fairly limited to the manufacturing industry. Some work has begun, however, in using DES to model aspects of space missions, although it has generally been limited to modeling only ground operations. For example, NASA Kennedy Space Center has developed GEM-FLO (A Generic Simulation Environment for Modeling Future Launch Operations) using discrete-event simulation to model the launch operations processing for space transportation systems[2]. RLVSim (Reusable Launch Vehicle Simulation) was created at Georgia Tech, which is also a discrete-event simulation model for reusable launch vehicle ground operations.[3]

LASSO, GEMFLO, and RLVSim were all created using a commercial DES program: Rockwell Software's Arena. LASSO is additionally combined with a database of vehicles in Excel and integrated into ModelCenter® to provide the capability to rapidly conduct design space exploration and optimization. In LASSO, the transportation-related aspects of lunar architectures are modeled end-to-end, from manufacturing, to integration and launch pad processes at the launch site, to all of the in-space segments and finally reentry and refurbishing of any reusable elements. Additionally, the model incorporates probabilistic simulations of cost, reliability, and processing times for each segment of the mission.

This study, then, uses LASSO to examine two competing architectures for lunar exploration. A baseline set of elements is established for each architecture, and a detailed breakdown of cost and reliability is provided for each baseline case. The main trade study is to examine a set of launch vehicle options, and determine the best choice for each architecture, based on cost, reliability, and ability to adhere to the given launch schedule. Additionally, a trade between two sizes of the in-space propulsive elements is examined. Finally, the two architectures are compared against each other, in light of the launch vehicle trade study and as a function of varying flight rates. Based on the assumed architectures in this study, conclusions are drawn as to the better architecture choice for the planned lunar exploration missions.

2.0 Discrete-Event Simulation

2.1 Overview

Discrete-event simulation (DES) as a modeling technique has existed for almost fifty years. Over the past two or three decades, with the improvement in computing power, simulation has become the most popular tool in operations research. Common examples of its use are in manufacturing, communications networks, transportation, and health-care delivery. Simulation allows for cost-effective modeling of complex, real-world systems to better understand the behavior of that system without actually having to build or prototype it. Discrete-event simulation, more specifically, refers to models where changes occur only at distinct points in time, such as parts entering or leaving a manufacturing facility at specific times. Inherent to discrete-event simulation is the ability to model stochastic systems – that is, models with random inputs, such as random manufacturing times on a part. Therefore, the true power of DES lies in its ability to model a complex system and its underlying uncertainty, and to study the behavior of that system without having to build or make changes to the real thing.

Discrete-event simulation began in the 1950s using programming languages such as FORTRAN to create simulations of complex systems. Because simulations were essentially coded from scratch, and because of the lack of computational power, discrete-event simulation was a very specialized tool used only by large corporations. By the 1970s and 1980s, as computers became faster and cheaper, specialized simulation languages were developed, such as GPSS, SIMSCRIPT, SLAM, and SIMAN. More recently, products such as Arena have emerged that have combined these programming languages with the ease-of-use of graphical user interfaces (GUIs), menus, and dialogs. Therefore, the simulation languages run in the background, while the user only sees the GUIs and a graphical animation of the simulation. The increased ease of use of these DES packages has allowed simulation to become a widespread tool in many companies, where it is being used earlier in the design process to save money, time, and effort downstream.[4,5]

The Lunar Architecture Stochastic Simulator and Optimizer (LASSO) uses Rockwell Software's Arena to model its lunar exploration architectures. Arena is a discrete-event simulation software package based on the SIMAN simulation language. It employs block-oriented programming, where at the most basic level, the user can simply string together modules to create a simulation, with little knowledge of the underlying SIMAN language. It also has the flexibility to write parts of a model in Visual Basic or C++, for example, or to read from and write to common applications such as Microsoft Excel.[5]

2.2 Fundamental DES Concepts

2.2.1 Basic Pieces of a Simulation Model

The most fundamental component of a discrete-event simulation is an *entity*. An entity represents any part in a simulation that moves around, changes status, affects and is affected by other entities, and affects the output performance measures. In general, an entity represents something physical in the simulation, such as a part entering a manufacturing facility or a customer entering a bank. Within a simulation, it is possible to have many realizations of one type of entity and many different types of entities. It is also possible to create a "fake" entity that takes care of other modeling operations, but does not represent any physical component of the actual system.

There are several basic pieces of simulation model that relate to the entities in the system. *Attributes* are used to individualize entities, with each entity having its own set of attribute values. Arena automatically assigns and keeps track of some basic attributes, but the user can define new ones depending on the needs of their simulation. For example, a user-defined attribute could be the time the entity was created or its priority in a machining queue. A *variable* is a piece of information that exists at the global level, instead of pertaining to an individual entity. Variables are accessible by all entities, and in general can be changed by any entity. Arena allows for variables to be scalars, vectors, or matrices. Again, Arena has a number of built-in variables, but also allows the user to create his or her own variables. In addition, a user can define an *expression* in Arena, which can be an equation (including numerous built-in statistical formulas) instead of a simple numerical value.

A *resource* represents something that entities compete over and that then acts on that entity, such as a drill press or a cashier in a store. As an entity moves through the system, it will seize resources as they become available and then release them when their process is completed. There can be multiple units of a particular resource, such as several cashiers in a grocery store, and an entity can seize any number of them as defined by the user. If all units of a particular resource are busy when an entity arrives – because other entities have already seized them – that entity then waits in a *queue* until a unit of that resource becomes available.[5]

In Arena, the most basic discrete-event simulation constructs are found in modules that the user can simply drag and drop into the model. These modules are packaged SIMAN code to perform the required operations in the simulation. Table 1 summarizes the modules found in the Basic Process Panel.

Table 1. Summary of modules in Arena Basic Process Panel.[6]

Arena Module	Description
Create	Used to create entities. Entities are created using a schedule or a time between arrivals. Can create any number of entities per arrival and specify a maximum number of entities to be created.
Dispose	Removes entities from the model.
Process	Main processing method in the simulation. Used for seizing and releasing resources. Process time can be defined as well as any associated cost.
Decide	Allows for decision-making in the model. Entities are branched in various directions based on a probability or a user-defined condition.
Batch	Used to group entities together. Batches can be temporary or permanent. User can specify batch size and how entities must be batched together (by having the same attribute value, e.g.).
Separate	Used to either separate temporarily batch entities or to duplicate an entity.
Assign	Used to assign variable values, entity attributes, entity types, entity pictures, or expressions.
Record	Used to collect statistics. Types of statistics are count (increases the statistic value by a specified amount each time an entity passes through), entity statistics, time interval (difference between attribute value and current simulation time), time between (time between entities entering the module), or a user-defined expression.

There are numerous other advanced modules available in other panels that are not outlined here. A simple model can be made, however, using just the eight modules explained above. In addition, Arena provides a list of Blocks that can be used in a model, which represent even smaller bits of SIMAN code.

2.2.2 Process-Oriented vs. Event-Oriented Simulation

At the surface, Arena employs process-oriented simulation (as is the case with most modern DES software). In a process-oriented simulation, an entity is tracked throughout the system, from entry to processing and finally departure. This can be looked at as taking the viewpoint of a single entity as it works its way through the model and all events that it goes through at distinct points in time. Behind the scenes, however, the discrete-event simulation is actually executed in event-orientation, which focuses on the events that occur, when they occur, and what happens to everything in the system at that time. This is required to keep track of time-persistent statistics that are extremely common in simulation.

An *event* is something that happens at a discrete instant of time that might change attributes, variables, or statistics, such as entity creations, arrivals at a queue, entities releasing resources, or entity disposals. For an actual simulation to occur, the program must keep track of all the events that are scheduled to occur. In Arena, this information is tracked within an *event calendar*. Basically, each future event that is supposed to occur in the simulation is recorded in the event calendar, with information about what entity is involved, the event time, the kind of event, etc. The events are then executed in chronological order, until the simulation is complete. In discrete-event simulation, the variables that describe the system can not change between events, since events only occur at distinct points in time and there are no other mechanisms for changing these variables. In Arena, the time in the simulation is held within the *simulation clock*.

Clearly, process-oriented simulation is much more intuitive and easier for a user to program. Process-oriented simulation actually closely resembles a common tool – flowcharting. In fact, Arena is even compatible with Visio, a drawing tool commonly used for flowcharting. This take on discrete-event simulation allows the user to easily build large, complex models without having to worry about the extreme complexity required in event-oriented simulation.

2.2.3 Basic Simulation Theory

At the heart of discrete-event simulation is a significant amount of statistical and queuing theory. The basic theory behind the most important aspects of DES, particularly those relevant to this study, is examined here.

One of the distinguishing aspects of DES is its ability to do stochastic simulations, that is, modeling random inputs and therefore random outputs. The basis of creating random inputs is the use of a random number generator (RNG), which creates an independent flow of random numbers from a continuous uniform distribution between 0 and 1. These random numbers can then be used to generate observations from other statistical distributions. Random number generation relies on a recursive algorithm that actually repeats the same sequence of random numbers over and over again. Good RNGs with high periods, however, will produce a flow of numbers that appear to be random and that pass statistical tests for uniformity and independence. Arena uses an RNG called a combined multiple recursive generator (CMRG) developed by L'Ecuyer, which takes the following form:

$$\begin{aligned}
A_n &= \left(1403580A_{n-2} - 810728A_{n-3}\right)\bmod\left(4294967087\right)\\
B_n &= \left(527612B_{n-1} - 1370589A_{n-2}\right)\bmod\left(4294944443\right)\\
Z_n &= \left(A_n - B_n\right)\bmod\left(4294967087\right)\\
U_n &= \frac{Z_n}{4294967088} \quad if\ Z_n > 0\\
U_n &= \frac{4294967087}{4294967088} \quad if\ Z_n = 0
\end{aligned} \tag{1}$$

where U_n is the random number between 0 and 1 being generated. The set of constants used above results in a cycle length of $3.1 * 10^{57}$![5]

These random numbers between 0 and 1 are then used to generate random variates from statistical distributions. Arena has thirteen different probability distributions built-in, but only the triangular and normal are of particular importance to LASSO. The triangular distribution, shown in Figure 1, is used in LASSO to model uncertainty in the input variables, such as cost and processing times. This distribution is commonly used when the exact shape of the distribution is unknown, but good estimates of the minimum, mean, and maximum values are available. Additionally, it is particularly useful for this application because it is bounded, whereas the tails of the normal distribution go to negative infinity and infinity.[5]

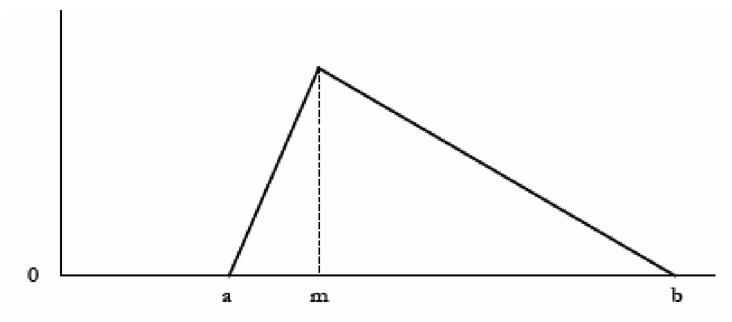

Figure 1. Probability Density Function of a Triangular Distribution.

The probability density function for a triangular distribution is as follows:

$$f(x)=\frac{2(x-a)}{(m-a)(b-a)} \quad for\ a \le x \le m$$
$$f(x)=\frac{2(b-x)}{(b-m)(b-a)} \quad for\ m \le x \le b \,, \tag{1}$$

where a = minimum, m = mean, and b = maximum.

The normal distribution is relevant when analyzing the simulation outputs. Because the inputs are random, the outputs are also random. Therefore, running the simulation once does not provide meaningful information. Several replications must be run in order to get a distribution on each of the output variables of interest. According to the Central Limit Theorem, if you have n random samples, $X_i,\ldots,X_n$, from a distribution with mean μ and variance σ^2, and if n is sufficiently large, then $\overline{X}$ is distributed normally with $\mu_{\overline{X}} = \mu$ and $\sigma_{\overline{X}} = \sigma^2 / n$. For example, if X were to represent life-cycle cost of a lunar mission, and n replications were run in Arena, then the average life-cycle cost across all the replications, $\overline{X}$, would have a normal distribution. In general, depending on the problem, n must be greater than twenty or thirty for the Central Limit Theorem to apply.[7]

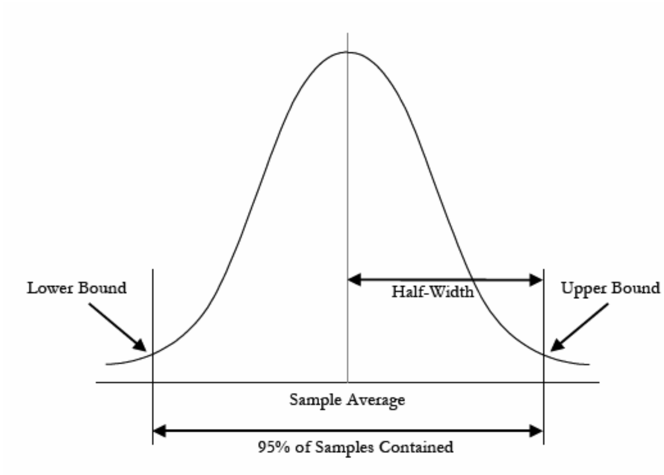

Figure 2. Probability Density Function of a Normal Distribution.

The probability density function of the normal distribution is shown in Figure 2, and can be expressed as follows:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(x-\mu)^2}{2\sigma^2}\right], \tag{2}$$

where μ is the mean and σ is the standard deviation. Because the distribution of the output statistics can be assumed to be normal, a half-width can be calculated, from which a confidence level or confidence interval can be calculated. A confidence interval represents the probability that if *m* replications are done *k* times, then 100(1-α)% of those *k* simulations will contain the true mean, where (1-α) represents the confidence level. For example, if a 95% confidence interval is desired for life-cycle cost, then 95% of the resulting *k* simulations will contain the true value of life-cycle cost. A 100(1-α)% confidence interval can be calculated as follows:

$$\left(\bar{x} - t_{\alpha/2}\frac{\sigma}{\sqrt{n}}, \bar{x} + t_{\alpha/2}\frac{\sigma}{\sqrt{n}}\right), \tag{3}$$

where t_a must be looked up in a table of critical values for *t*-distributions.

Finally, much of DES itself is based on queuing theory and the ability to calculate time-persistent statistics. Some of the important statistics that Arena automatically calculates are average queue length, average waiting time in the queue, and the average resource utilization. In queuing theory, the simplest type of queue is an M/M/1 queue, which refers to interarrival and service distributions being exponential (Markovian) and having a single server. Most basic queuing theory is based around variations of M/M/1 queues.[8] In LASSO, however, very few of the queues are M/M/1. Therefore, analytical equations can not be easily derived, and Arena must be counted on to calculate the pertinent queuing statistics. Furthermore, most queues in Arena are first-in first-out (FIFO), where the first entity to arrive in the queue is also the first to leave. In general, LASSO also uses FIFO queues.

The average queue length and average waiting times in the queue are calculated as follows:

$$\overline{Q}(s,t) = \frac{1}{t-s}\int_s^t Q(u)du \tag{4}$$

$$\overline{W}(s,t) = \frac{1}{N(s,t)}\sum_{i=N(s)+1}^{N(t)} W_i \tag{5}$$

In the above equation, *(s,t)* represents the time interval being examined. In equation 5 for the average queue length, *Q(t)* is the queue length at time t and in equation 6, W_i is the waiting time of departure entity *i*.

The average resource utilization is given by equation 7, and is a number between 0 and 1:

$$\overline{B}(s,t) = \frac{1}{t-s}\int_s^t B(u)du\ , \tag{6}$$

where *B(u)* is the number of servers busy at time *t*.

3.0 Lunar Architecture Stochastic Simulator and Optimizer

The Lunar Architecture Stochastic Simulator and Optimizer (LASSO) integrates three existing software programs to model, analyze and optimize lunar exploration architectures: Rockwell Software's Arena, Microsoft Excel, and Phoenix Integration's ModelCenter®. Arena is used to create full end-to-end models of lunar transportation architectures, including manufacturing of all the necessary elements, payload and launch vehicle integration, launch, in-space propulsive segments, Earth re-entry, and turn-around processes where applicable. Furthermore, the models include distributions on cost and time variables and a probability of failure for each launch, propulsive burn, and reentry. The Arena models are linked to Excel, which contains a database of the various elements within the architectures. These include launch vehicles, in-space propulsive stages, lunar landers, and crew exploration vehicles. For each element, Excel contains pertinent metrics such as gross mass, propellant mass, payload capacity, cost, reliability, and cycle times. Depending on the elements chosen, Arena pulls the appropriate data from Excel and stores it in variables within the model. Finally, each architecture modeled in Arena is wrapped into ModelCenter® to provide the capability for design space exploration and optimization. This allows for optimizing both the overall architecture as well as individual vehicle choices to minimize overall program cost and risk and to maximize mission throughput. Each of these components will be explained in further detail in this section.

3.1 Vehicle Database

Figure 3 provides a screen shot of the LASSO database. As can be seen, there is a row for each of the vehicle elements required in the Arena architecture models. Arena pulls data from the database by assigning an index to each element, which corresponds to a row in that element type's database. Each element type has its own Excel sheet where the data is kept (not shown in Figure 3, but can be found in Appendix A), and a lookup is then used according to the assigned index. Three launch vehicles must be chosen, one type each for crew, cargo, and propellant. Two in-space propulsive stages are chosen, one for the trans-lunar injection (TLI) stage and one for the trans-Earth injection (TEI) stage. Finally, a lander and a crewed stage are chosen.

INDICES								
LV (crew)	LV (cargo)	LV (prop.)	TLI	TEI	Lander	CEV	Architecture	
3	11	8	1	3	1	1	1	POD

	Launch Vehicle					
	Vehicle	Index	Payload *kg*	DDTE $M	Price *$M*	Reliability
'crew)	**AtlasV heavy**	3	19060	2926	321.25	0.95593
cargo)	**Colossus70 mt**	11	70000	4829	614.5	0.991
'prop)	**AtlasV heavy**	8	19060	0	257.18	0.95593

	In-Space Stages												
	Vehicle	Index	Payload *kg*	Dry Weight *kg*	Gross Weight *kg*	Prop. Weight *kg*	DDTE *$M*	TFU *$M*	Fixed Ops *$M*	Variable Ops *$M*	Propellant *$M*	Reliability	Manuf. Time *days*
(TLI)	**Manticore**	1	18000	6418	44812	38394	942.2	120	70	0.14	0.04	0.95	540
(TEI)	**PPM**	3	7700	5232	10183	4952	2335.12	434	40	1.58	0.07	0.98	540

Landers													
Vehicle	Index	Payload *kg*	Dry Weight	Gross Weight *kg*	DDTE *$M*	TFU *$M*	Fixed Ops *$M*	Variable Ops *$M*	Propellant *$M*	Reliability	Lifetime *missions*	Manuf. Time *days*	No Hab?
Eagle	1	500	4104.6	11550	1124	194	3	0.96	0.01	0.996	1	540	yes

Crewed Stages											
Vehicle	Index	Mass (launch) *kg*	Mass (in-space) *kg*	DDTE *$M*	TFU *$M*	Fixed Ops *$M*	Variable Ops *$M*	Reliability	Lifetime *missions*	Manuf. Time *days*	TAT *days*
Capsule	1	12200	7600	2500	300	50	20	0.99	1	540	0

Launches											
	Number of Launches			Element Launch Number (Cargo LVs)				Batch Size - Cargo Launches			
# TLIs	Crew	Cargo	Propellant	Lander	TLI1	TLI2	TEI	Launch 1	Launch 2	Launch 3	Launch 4
2	2	1	5	1	1	1	1	4	0	0	0

Figure 3. Screen Shot of LASSO Excel Database.

Table 2. Elements and Metrics in Database[9,10,11].

Element Type	Sub-Categories	Elements	Metrics
Launch Vehicles	Crew, Cargo, Propellant	Delta (IV Medium and IV Heavy), Atlas (V 502 and V Heavy), Centurion (C1, C2, and C3), Shuttle-derived (SRB Stick, ET Derived, and C), Vega RLV	Payload, DDTE, Reliability, and Lifetime
In-space Propulsive Stages	TLI, TEI	Manticore, PPM	Payload, Dry Mass, Gross Mass, Propellant Mass, DDTE, TFU, Operations Costs (Fixed and Variable), Propellant Cost, Reliability, Manufacturing Time
Lunar Landers		Eagle (Apollo-derived), Artemis	Payload, Dry Mass, Gross Mass, Propellant Mass, DDTE, TFU, Operations Costs (Fixed and Variable), Propellant Cost, Reliability, Lifetime, Manufacturing Time, Built-in Habitat (yes or no)
Crewed Stages		Apollo-derived capsule, Tempest	Launch Mass (with CES), In-space Mass, DDTE, TFU, Operations Costs (Fixed and Variable), Reliability, Lifetime, Manufacturing Time, and TAT

Table 2 summarizes the different element types in the database, the elements available under each type, and their associated metrics. The available architecture elements are a combination of existing vehicles and paper studies. More detailed information on each element can be found in Appendix B.

The database is also responsible for calculating the number of launches required per lunar mission based on the elements chosen. First, the number of TLIs required is calculated. If the payload of one TLI is insufficient to take all elements to lunar orbit, then the elements can be split among two TLI stages as follows: CEV/TEI and Lander. Next, the number of crew launches represents whether or not the CEV and TEI can fit on one launch vehicle. If not, the TEI stage is launched as a cargo element (only one crew launch is done carrying just the CEV as payload). Next, the number of cargo launches is calculated. If the TLI stage can not be launched fully loaded, it is launched dry and the number of dedicated propellant launches required is also calculated. All other elements must launch fully fueled. If several launch options are possible, a launch configuration with no dedicated propellant launches and the minimum number of crew launches is chosen before any configurations with dedicated propellant launches. As an example, Figure 4 shows an architecture that requires two TLI stages and one crew launch (CEV/TEI launch together).

numTLI = 2, crewLaunch = 1

Index #	Launch Configurations	Possible?	# Launches	Prop. Launches
1	Lander/TLI/TLI	0	1	0
2	Lander/TLI TLI	1	2	0
3	Lander TLI TLI	1	3	0
4	Lander/TLI/TLI(e)	1	1	3
5	Lander/TLI(e)/TLI(e)	1	1	5
6	Lander/TLI TLI(e)	1	2	3
7	Lander/TLI(e) TLI	1	2	3
8	Lander/TLI(e) TLI(e)	1	2	5
9	Lander TLI(e)/TLI(e)	1	2	5
10	Lander TLI TLI(e)	1	3	3
11	Lander TLI(e) TLI(e)	1	3	5

Figure 4. Example Launch Configuration Calculation from Database.

The order of preference for choosing the best launch configuration is top-to-bottom. For the options that are possible, it first tries to minimize the number of cargo launches subject to no propellant launches. If there are no feasible options without propellant launches, it will minimize the number of cargo launches and then the number of propellant launches. Propellant launches are undesirable because they add the complexity of on-orbit refueling to the mission. Therefore, for the case shown, launch option #2 is chosen.

Finally, a launch number is assigned to each element (the order is not important) along with the number of elements on each launch. These variables are needed in the Arena model in order to correctly launch all of the elements.

3.2 Lunar Architecture Concepts

Two different architectures are examined in this study, each modeled in Arena: an expendable Apollo-style architecture and a next-generation highly reusable architecture. Each architecture has some common mission assumptions, as outlined below:

- Orbit characteristics:
  - LEO rendezvous orbit = 400 km × 28.5°
  - LLO rendezvous orbit = 100 km × 90° (polar orbit)
- Trajectory calculation[12]:
  - Time of Flight (LEO to LLO) = 3.5 days
  - TLI Delta-V = 3100 m/s
  - LOI/TEI Delta-V = 840 m/s
- Lunar mission specifications:
  - Number of crew = 4
  - Time on lunar surface = 4 days
  - Payload to lunar surface = 500 kg
  - Payload from lunar surface = 100 kg

All of the above assumptions were required to size the individual vehicles and can only be changed by entering new elements into the database, or modifying the existing elements. For example, if more crew or a longer surface duration were desired, a larger CEV would have to be added to the database. Larger propulsive elements would then have to be included as well, in order to account for the larger CEV.

The expendable architecture, shown in Figure 5, consists of all expendable elements, as its name suggests. The baseline is shown in the figure, where all cargo elements are first launched into low Earth orbit on a cargo launch vehicle. The Centurion C2[9], with a payload capacity of 100 mt, is chosen as the baseline so that only one launch is required. The crew is then launched in the CEV, along with the TEI stage, on a man-rated Delta IV Heavy. All of these elements dock in Earth orbit, before beginning their transit to the Moon. Once in lunar orbit, the lander carries the crew to the lunar surface, while the CEV and TEI remain in lunar orbit. The lander has a built-in habitat in the ascent stage, but if another surface habitat is desired, it can be pre-deployed (not modeled in Arena). The lander descent stage remains on the lunar surface, while the ascent stage then carries the crew back to lunar orbit, where it docks with the CEV and TEI. The crew transfers to the CEV, which travels back to Earth and does a direct entry to the surface.

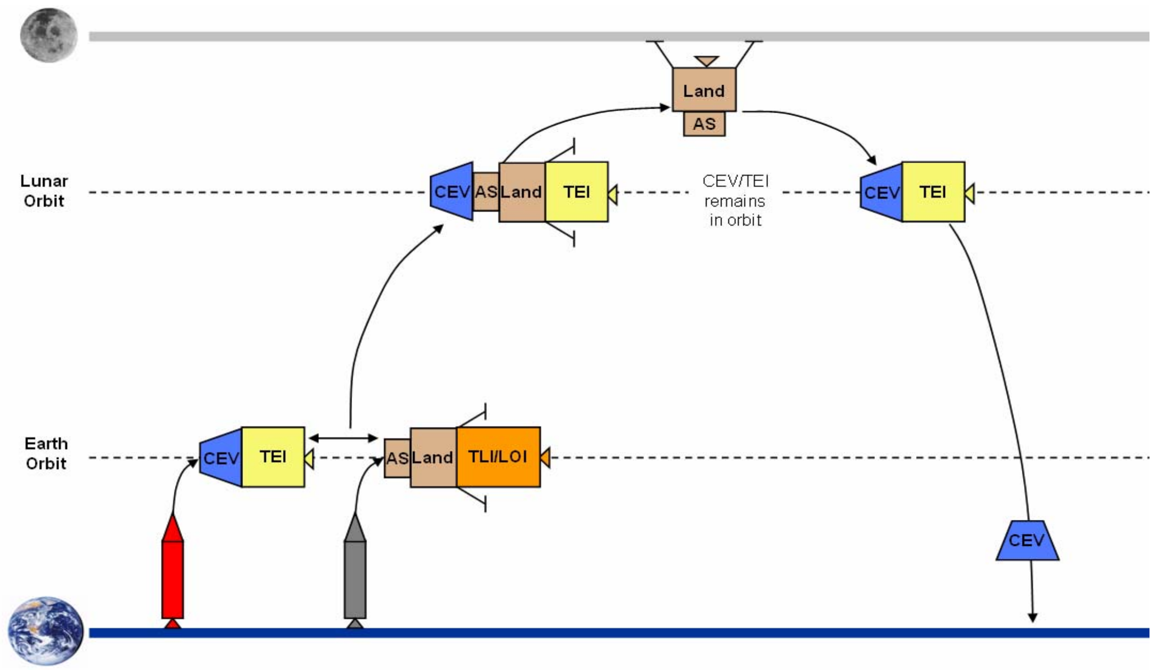

Figure 5. Diagram of Expendable Lunar Architecture.

Several variations to this architecture are possible, depending on the individual elements chosen. Depending on the payload capacity of the TLI stage, two may be needed. One TLI would carry the CEV and TEI, while the other would carry the two-stage expendable lander. If another launch vehicle were chosen, the lunar lander and TLI stage(s) may have to be launched separately. Additionally, propellant launches may be required if the TLI is launched dry. Propellant is launched into lunar orbit on a propellant resupply module (PRM), which has a propellant mass fraction of 0.84. The PRM is sized based on the payload of the launch vehicle; so for the baseline case, one launch can deliver 84 mt of propellant to Earth orbit (Centurion C2 has a payload of 100 mt). Therefore, this architecture could have up to two TLI stages and up to four total launches, not including the additional propellant launches. The number of propellant launches if of course determined by the payload of the chosen launch vehicle.

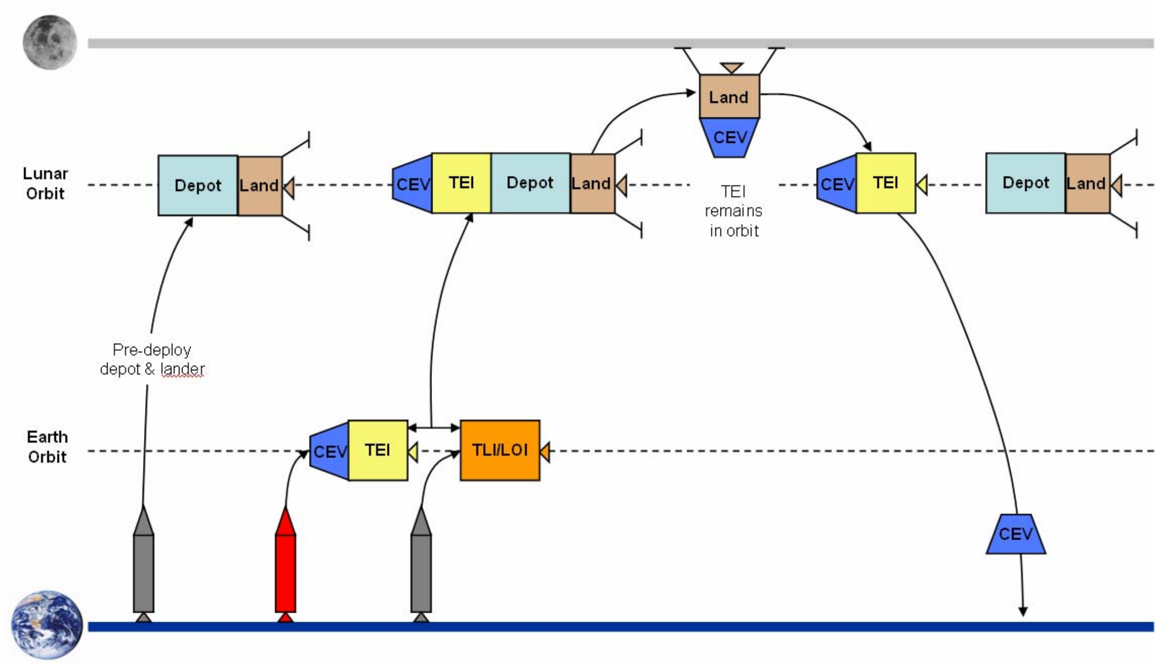

Figure 6. Diagram of Reusable Lunar Architecture.

The reusable architecture is shown in Figure 6. The major difference in this architecture is the use of a reusable lunar lander and propellant depot in lunar orbit. The baseline case again uses a man-rated Delta IV Heavy for the crew launches and the Centurion C2 for the cargo launches. The propellant depot and lander are pre-deployed along with a surface habitat if desired (the launch and costs associated with the surface habitat and fuel depot are not modeled). At the beginning of the simulation, an empty depot is assumed to already be in lunar orbit, and the lander is launched fully fueled before the first human lunar mission. Only one TLI can be used per manned mission in this architecture, since the lander is already in lunar orbit. Once the CEV/TEI and TLI have launched, they dock in Earth orbit and travel to the moon as a single stack. Upon arriving in lunar orbit, the lander carries the CEV to the surface (the reusable lander does not have a built-in habitat), while the TEI remains in lunar orbit. Upon completion of the lunar mission, the lander carries the CEV back to lunar orbit, where it docks with the TEI for Earth-return. The lander refuels from the propellant depot and remains in lunar orbit until the next lunar mission is launched. The lander must be periodically replaced when it lifetime expires.

For the baseline case, the depot has a propellant capacity of 50 mt. When the depot no longer has enough propellant to fill the lander, propellant is launched on a Centurion C2, along with a TLI stage to carry the propellant to lunar orbit. The size of the PRM is limited by the payload of the TLI stage (30 mt). Therefore, since the Centurion C2 can launch 100 mt, two launches are required: one to launch the TLI fully fueled and the other to launch the PRM. The baseline architecture therefore requires two Centurion C2 launches to ferry 25.2 mt of propellant to the lunar depot.

Although the baseline case uses the same expendable Apollo-style crew capsule, this could be substituted with a reusable CEV, to create a more highly reusable lunar architecture. Additionally, different launch vehicles can be chosen. As with the expendable architecture, the TLI can be launched dry if necessary with the addition of dedicated propellant launches. The lander can also be launched dry if necessary and fueled from the depot once it reaches lunar orbit.

3.3 Arena Lunar Architecture Models

Both of the above lunar architectures are modeled in Arena as two separate models, although each has similar data flow, logic, and assumptions. As aforementioned, the Arena models span all transportation-related aspects of the architecture, including manufacturing, integration, launch, in-space segments, and turn-around processes. Figure 7 shows a screen shot of the top-level Arena reusable lunar architecture model. All of the Arena modules are found in submodels to better organize the model, as well as to make for easy modularity for creating new architecture models. The teal colored box includes all of the submodels containing the model data flow. The plot on the top right illustrates the mission schedule in blue and when the missions actually launch in red. With this set of inputs, the mission demand is clearly not able to be met. The key inputs to the model are found in the white boxes on the left, while the key outputs are found in the yellow boxes on the right.

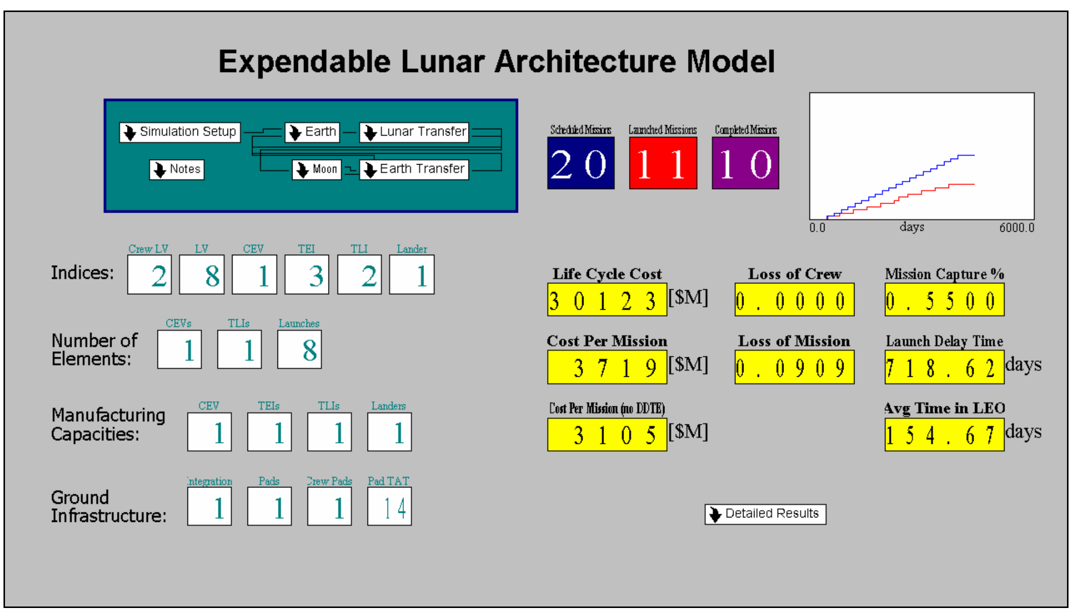

Figure 7. Screen Shot of *Arena* Reusable Lunar Architecture Model.

Figure 8 shows a top-level flowchart of the data flow in the Arena models. The initial setup contains the interface with the Excel database as well as VBA code used to simplify some calculations that must be done. Although these can be done using pre-existing Arena modules, it was significantly easier to implement in VBA. First, the VBA code calculates the expected number of reusable elements that will need to be built, with a minimum of three. Next, based on the number of missions desired per year and the manufacturing time for each element, it determines the day each element should begin manufacturing such that the mission will launch on time based on mean times for each process. It also calculates when each mission should begin integration such that it will launch on time. Finally, it checks if the mission is feasible for the given combination of elements based on the following criteria:

- Is the lander payload sufficient to carry the CEV to the lunar surface (if applicable)?
- Is the crew launch vehicle payload sufficient to launch the CEV?
- Is the cargo launch vehicle payload sufficient to launch the single largest element?
- Is the TEI payload sufficient to carry the CEV to Earth?
- Is the TLI payload sufficient to carry the lander and the CEV/TEI?

If any of these conditions are violated, the simulation is immediately terminated. Arena also uses VBA code to write the indices for each architecture element to Excel, then reads the data pertaining to each vehicle as well as the launch configuration into Arena variables. The VBA code used in the models can be found in Appendix A.

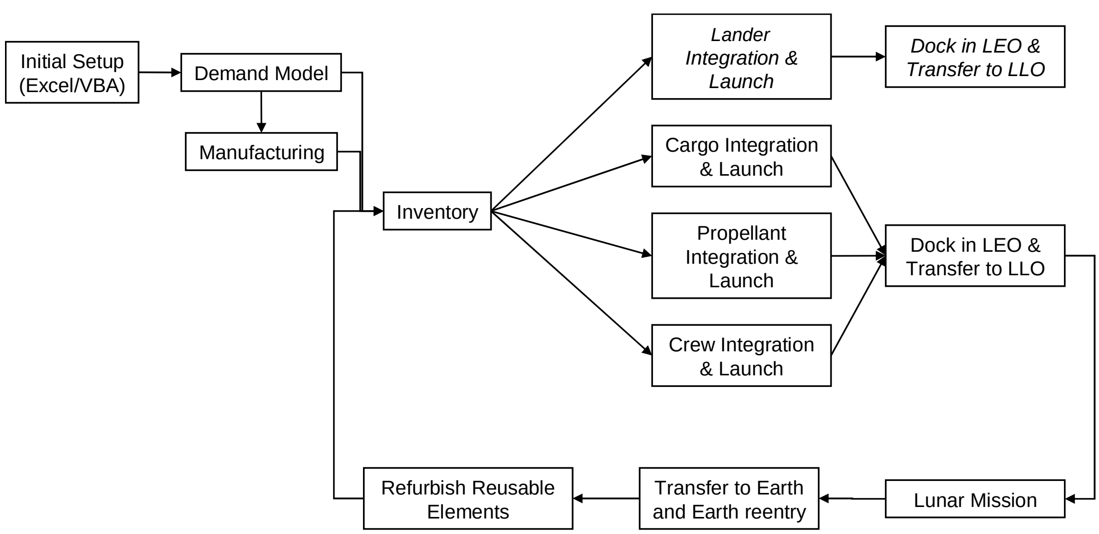

Figure 8. Top-Level Flowchart of Arena Models.

The Demand submodel in Arena is responsible for handling the scheduling of the missions. It controls when elements need to begin manufacturing, when the elements need to be removed from inventory to being integration, and for the reusable architecture, when more propellant depot resupply launches are needed. The manufacturing submodel is straightforward. Each element seizes a manufacturing resource, each with a capacity and expected manufacturing time. Once manufacturing is complete that element is stored in inventory. When a mission is scheduled to begin, the elements are sent to their correct launch facilities – between crew, cargo, and propellant – for integration and launch. Each type of launch vehicle has its own dedicated integration facility and launch pads, each with a user-defined capacity. For the reusable architecture, there is a separate submodel for launching the lander and propellant for the depot. These launches also have their own dedicated launch facilities. Although not shown in Figure 8, after a launch takes place, the launch pad goes through a turn-around process before it can be seized again by the next set of elements.

The in-space mission segments are then modeled as separate submodels. Each contains a probability of failure for each element that is used (TLI, TEI, lander, and CEV), along with times associated with each mission event (transit times, docking times, lunar surface time, etc.). Finally, upon successful Earth reentry, the expendable elements are disposed of, and any reusable elements go to refurbishing facilities before returning to inventory.

3.3.1 Assumptions

In the lunar architecture models, certain assumptions were made about the system that are outlined here. First, the LASSO architecture models model only the transportation-related elements along with their associated costs. This includes associated processes such as manufacturing, integration with the launch vehicle, and the in-spaces segments, and costs such as DDTE, production, and operations. The transportation aspect of the lunar exploration program comprises only a portion of the overall life-cycle cost of an actual program. In addition to the transportation elements, there are also costs associated with science payloads, technology development, precursor missions, etc. Therefore, the total cost reported by Arena can not be taken as the actual total cost of a lunar program.

In terms of the ground processes associated with conducting a lunar mission, LASSO model manufacturing, payload integration with the launch vehicle, time on the launch pad, and any turn-around processes required for reusable elements that return to Earth. The tool, however, does not include the costs associated with building additional launch pads or manufacturing facilities, for example. The user therefore must use some intuition to realize that, although missions may always launch on time when there are a hundred launch pads, it is not economically feasible to build an unlimited number of launch pads. The Arena models do not capture this element of ground infrastructure considerations.

In terms of manufacturing, reusable elements are scheduled to begin manufacturing such that if everything goes according to schedule, they will be ready for launch on time. This includes a user-defined time when elements are supposed to arrive in inventory before a mission, which serves as a buffer if manufacturing does not run on schedule. All reusable elements, however, are built up-front, at the beginning of the program. The number of elements required to meet the mission demand is calculated, but a minimum of three reusable elements must be built. When a reusable element fails, one is removed from inventory and then an additional one is built to replace it if necessary. The model also incorporates "pay as you go" costs, which means that only elements used in a mission are paid for. Even if there were originally twenty missions scheduled, if only ten are launched because the program fell behind schedule, then the extra elements are not manufactured or paid for.

Crew, cargo, and propellant launches each have their own launch vehicle and dedicated facilities. Even if the same vehicle is chosen for cargo and propellant launches, they still each have their own dedicated launch pads and integration facilities. Additionally, for the reusable architecture, a separate set of launch pads and integration facilities are used for launches required for depot resupply missions. This assumption had to be made for modeling purposes; otherwise the complexity of the model would have become unmanageable. For a lunar mission, the elements are always launched in the following order: cargo, propellant (if needed), crew. The crew does not launch until everything else has successfully reached orbit. If there is a launch failure of any cargo element, the mission is cancelled, and the crew, propellant, and any remaining cargo never launch. The elements that have not launched are returned to inventory. Any elements already in orbit, however, are lost. If a propellant launch fails, however, another one is simply launched to replace it. The amount of time elements must spend in lunar orbit, however, is not taken into consideration, although it is tracked as a variable. For example, if it takes a full year between when the first cargo element launches and when the crew launches, propellant boil-off or element lifetimes are not considered. Therefore, the user must examine the statistic representing the total time in low earth orbit to determine if that particular set of vehicles is feasible for a lunar mission.

There are several failures that can result in a loss of crew event. These include a crew launch vehicle failure where the abort is unsuccessful, a TLI stage failure where abort to Earth is unsuccessful, a lunar lander failure, a TEI stage failure, or a CEV reentry failure. It is important to note that smaller launch vehicles may not have sufficient payload mass to include a crew escape system, and then there is no abort option for launch. Whenever a loss of crew event occurs, a stand down time is initiated for an investigation, and all missions scheduled to launch during that time are cancelled. These cancelled missions do not count against the mission capture percentage, as will be described later. Manufacturing for these missions is also cancelled, again so unnecessary costs are not incurred.

Finally, at the end of the scheduled program, no more missions can launch, but time is added for any current mission to reach completion. For example, if a ten year program is desired, at the end of the tenth year of lunar missions, even if all the scheduled missions have not launched, the simulation is completed. If production has not begun for these missions, production costs are not incurred.

3.3.2 Inputs and Outputs

All of the important inputs and outputs can be seen in Figure 7, but are explained here in further detail. Table 3 describes each of the key inputs to the model. Where applicable, distributions are assigned to each of the inputs, as listed later in Table 5.

Table 3. Inputs to Arena Lunar Architecture Models.

Input	Description
Missions per Year	Number of scheduled lunar missions per year (can be a fractional value).
# Years	Number of years in lunar program (starting with first mission launch).
Vehicle Indices	Index number corresponding to an entry in the database for each element type (CEV, TLI, Lander, TEI, LV crew, LV cargo, and LV propellant).
Inventory Time	Time that elements should be delivered to inventory before integration is scheduled to begin.
Integration Time	Expected time for payload integration with launch vehicle.
Pad Time	Expected time for payload/launch vehicle stack to spend on launch pad prior to launch.
Pad TAT	Expected turn-around time of launch pad.
Investigation Time	Expected length of stand down time resulting from loss of crew event.
Manufacturing Capacities	Number of a particular element that can be built at one time (one variable each for CEV, TLI, TEI, and lander).
Integration Capacities	Number of launch vehicles that can be integrated with their payloads at a given time (one variable each for crew, cargo, and propellant).
Launch Pads	Number of launch pads available to that particular launch vehicle (one each for crew, cargo, and propellant).
Depot Capacity	Propellant capacity of propellant depot (for reusable model only).
Cost Lower Bound (%)	Lower bound on triangular distributions used for cost.
Cost Upper Bound (%)	Upper bound on triangular distributions used for cost.

There are numerous outputs that Arena automatically generates each time a simulation is run. For example, it generates numerous statistics on each entity, queue, resource, and record block in the model. Only a handful of these are of particular interest in evaluating an architecture simulation. Arena allows users to not only use record blocks to generate statistics but also to insert formulas as statistical expressions. Table 4 lists the key figures of merit for the lunar architectures along with a description of each.

Table 4. Key Outputs from Arena Lunar Architecture Models.

Output	Description
Life Cycle Cost	Total transportation-related program cost (includes DDTE, production, fixed and variable operations, launch, and propellant).
Cost per Mission	Life cycle cost divided by the number of launched missions.
Loss of Crew	Probability per mission that a loss of crew event occurs (number of loss of crew events divided by number of launched missions).
Loss of Mission	Probability per mission that a loss of mission event occurs (includes loss of crew events).
Mission Capture %	Percent of scheduled missions that are launched (missions cancelled due to stand down time do not count against capture %).
Launch Delay Time	Average time between scheduled launch date and actual first launch of that mission.
Time in LEO	Average time between first and last launch for a particular mission.
Bottleneck Statistics	Average waiting time in queues (manufacturing, integration, launch pads, turn-around processes).

These outputs were defined with the intention of making each as independent as possible, so as not to be affected by changes in the other metrics. Life-cycle cost includes all of the transportation-related costs modeled in LASSO for the entire program duration. Cost per mission was also chosen as a metric, because it reduces the dependence on capture percentage found in life-cycle cost. For example, if comparing two different architectures, one with 100% mission capture and one with 50% mission capture, the second will most likely have a lower life-cycle cost simply because fewer missions were launched. Cost per mission, however, divides the total life-cycle cost by the number of missions actually launched, which facilitates a more fair comparison between the two architectures. Loss of mission and loss of crew represent the probability that a particular mission will either fail or that the crew will be lost. Mission capture percentage is the percent of scheduled missions that actually launch. Even if there is a failure during the mission (including during launch), the mission still counts as having launched in calculating the capture percentage. Basically, the capture percentage is intended to measure how many missions can be achieved based on the available ground infrastructure. It is not intended to take reliability into account. Launch delay time and LEO time also pertain solely to ground infrastructure considerations, and are explained in Table 4.

The figures of merit, with the exception of the bottleneck statistics, will comprise the overall evaluation criterion (OEC) which will be used to evaluate the overall merit of each architecture combination examined. Various figures of merit can be used, each with a particular weight assigned to it, depending on what criteria is most important to that particular simulation. The bottleneck statistics are used to determine what the limiting ground infrastructure is if the mission capture percentage is less than 100%.

3.4 Running LASSO

The LASSO lunar architecture models were created using Arena Professional Version 7.01. Arena does not have good back compatibility; therefore, for the models to run correctly, an older version of Arena may not be used. In order to run LASSO in Arena, three files must be in the same folder: the Arena architecture model (*Expendable.doe* or *Reusable.doe*), the Excel database (*Database8900.xls*), and the results spreadsheet (*Results.xls*). Additionally, macros must be enabled in Microsoft Excel.

When the "Go" button is pushed in Arena, it begins running the model. A User Form will come up, as shown in Figure 9, with all of the inputs listed in Table 3. Additionally, there is an option for the number of replications and whether the model should be run in Batch Mode. Batch Mode turns off all animations, which enables the simulation to run faster. For this study, the simulations were run on a 3.4 GHz Pentium 4 processor. As an example, to run 20 replications of the expendable architecture in Batch Mode takes approximately 40 seconds; otherwise, it takes 90 seconds running at an Animation Speed Factor of 100 (the fastest animation speed option). The Database and Results spreadsheets must also be closed for the model to run.

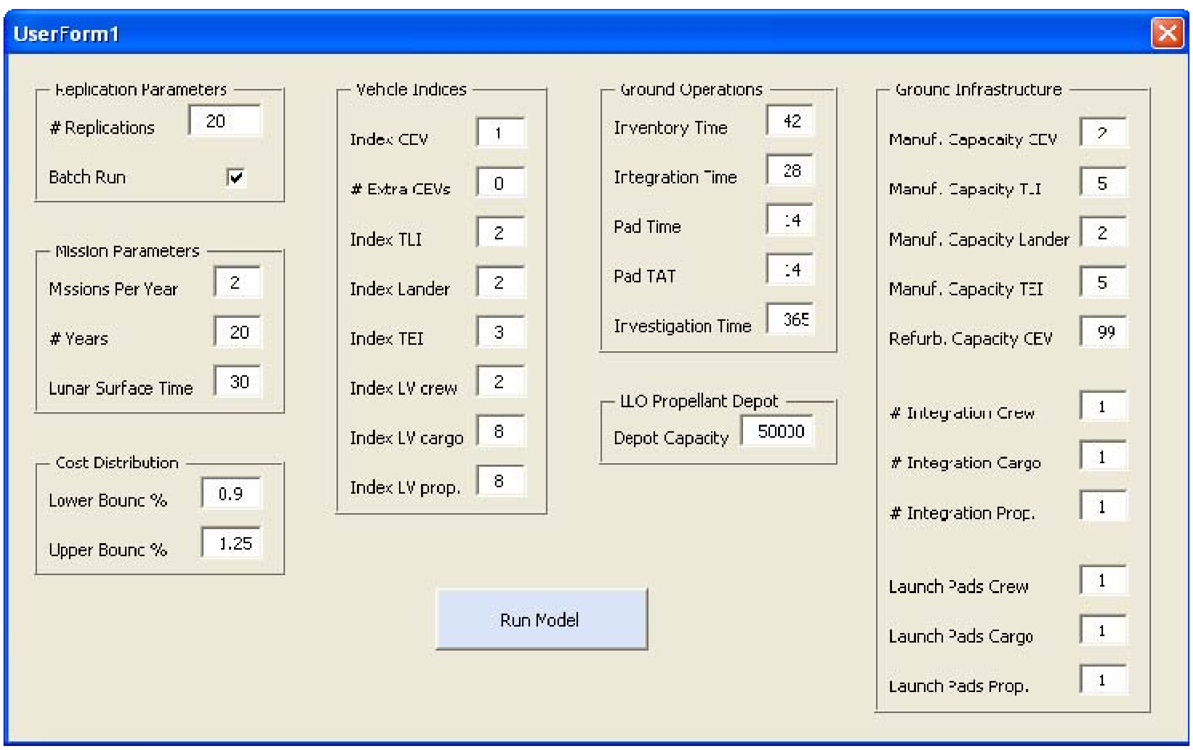

Figure 9. Screen Shot of Arena User Form.

For each replication, Arena writes the outputs listed in Table 4 to a new row in the Results spreadsheet. Although Arena has automatic report generation, it contains a large amount of extraneous information that is difficult to sift through each time a single result is desired. Additionally, Arena only provides the mean and half-width (for a 95% two-sided confidence interval) for each statistic. If a different confidence level is desired, this calculation must be done offline. The results spreadsheet is set up to calculate the mean, standard deviation, and confidence level desired (assuming a normal distribution as explained earlier). A screen shot of the Results spreadsheet is shown in Figure 10.

Number of Replications:		100						
Confidence Value:		0.95						
Mean	69648	2512	1826	0.079	0.237	0.786	522.675	235.910
Std. Dev.	4202	237	132	0.059	0.073	0.060	314.110	1.739
Confidence	$70,472	$2,559	$1,852	0.090	0.251	0.774	584.24	236.25
Rep. #	**LCC**	**Cost_Mission**	**t_Mission (No DI**	**LOC**	**LOM**	**Capture_Rate**	**Launch Delay**	**LEO Tim**
1	70818.5649	2442.019479	1827.364307	0.06896552	0.206896552	0.805555556	305.57	234.791
2	72564.54806	2267.642127	1646.798377	0.03125	0.3125	0.888888889	581.55	236.284
3	59701.0024	2985.05012	2121.70012	0.2	0.3	0.666666667	309.81	237.298
4	76122.37508	2455.560486	1766.141131	0.03225806	0.193548387	0.815789474	614.26	236.1
5	72854.19968	2601.935703	1873.57856	0.07142857	0.321428571	0.736842105	483.09	235.1
6	73654.33667	2727.938395	1987.530988	0.03703704	0.074074074	0.710526316	959.16	236.6
7	67085.76741	2580.221823	1781.87567	0.11538462	0.269230769	0.764705882	225.16	234.0
8	66945.88166	2479.477099	1818.328951	0.07407407	0.222222222	0.771428571	343.28	237.2
9	74186.05468	2393.098538	1745.034022	0.03225806	0.225806452	0.794871795	665.67	236.4
10	67304.56532	2588.637128	1881.175589	0.11538462	0.269230769	0.764705882	251.62	233.8

Figure 10. Screen Shot of Excel Results Spreadsheet.

3.5 ModelCenter® Integration

Phoenix Integration's ModelCenter® is an environment that allows various software tools to be integrated together for design space exploration and optimization. The Arena lunar architecture models have been wrapped into ModelCenter® for this express purpose. Although not currently integrated with any other tools, it does provide the flexibility for the user to expand the functionality of LASSO by adding other disciplinary tools as desired. Using ModelCenter® also allows for parametric studies to be easily conducted. Certain variables can be varied and the results will be automatically generated for each case. This increases the speed of execution, since the user does not have to run each case individually in Arena and then record the results. Although Arena does have some limited design space exploration and optimization capability, it is not sufficient for the purposes of this study. For example, as will be seen in the results, the number of missions per year can be varied across a wide spectrum and the benefit of each architecture at different flight rates can be observed. Additionally, ModelCenter® with the DOT Optimizer allows for optimization of the overall architecture choice and elements within that architecture for a user-defined OEC.

In order to wrap the Arena models into ModelCenter®, a ScriptWrapper was created using VBScript. The inputs into the Arena models and outputs generated are the same as listed in Table 3 and Table 4, with two additions. The choice of architecture is a variable within ModelCenter® (1 = Expendable, 2 = Reusable), and the OEC appears as a ModelCenter® variable that can be defined by the user. For ModelCenter® to execute properly, Arena must be set to Batch Run and the User form must be turned off (both done automatically within the ScriptWrapper). The Results spreadsheet is also wrapped in ModelCenter® to access the results for each simulation. A simple ExcelWrapper is used for this purpose.

4.0 Results

Using LASSO, several different trade studies were conducted on the expendable and reusable lunar architectures, in order to determine which architecture and which set of elements within that architecture would be best based on the various figures of merit being examined. In general, the figures of merit of interest are cost, reliability, and adherence to the program schedule, although these are each broken down further as applicable in each trade study. In order to be consistent across all trades, the same baseline mission assumptions were used, as listed in Table 5. The baseline values listed are the mean values, while some have a triangular distribution applied to them during the simulation, as indicated in the table.

Table 5. Baseline Mission Assumptions for Architecture Trade Studies

Metric	Baseline Value	Units	Lower Bound	Upper Bound
Missions Per Year	2	missions/yr.	---	---
Program Duration	10	years	---	---
Inventory Time	42	days	---	---
Integration Time	28	days	---	+10%
Pad Time	14	days	---	+30%
Pad TAT*	14	days	-5%	+10%
Investigation Time	365	Days	-25%	+100%
Learning Curve	0.85	---	---	---
Cost Metrics	---	$M	-10%	+15%

*Pad TAT set to 0 for Unlimited Ground Infrastructure Scenarios

For each result presented, 100 replications of the Arena model were run. This number was determined by qualitatively examining the output statistics to establish how many replications were necessary for the normal distribution assumption to be appropriate. All results presented are 95% confidence bounds (upper bound if the statistic is to be minimized and lower bound if statistic is to be maximized). Additionally, for the element combinations that require propellant launches, a propellant resupply module (PRM) with a mass fraction of 0.84 is assumed, as explained earlier. This element is not modeled in the database, but the assumption is consistent across all of the cases run.

It is important to remember that LASSO only models the cost related to the transportation elements within a human lunar exploration program. This includes the DDTE, production, and operations costs associated with each transportation element, and the costs associated with launch, both development (if applicable) and launch price. What is not included, however, are the other costs associated with a human exploration program, such as science payloads, technology development, and precursor missions. Therefore, the life-cycle costs presented are only for the transportation portion of the lunar exploration program, which comprises just one section of what will be required to actually land people on the Moon and conduct meaningful science and exploration.

First, the baseline results are given for each architecture. The majority of the results then comprise of a launch vehicle trade study, where five different launch vehicle families are examined. The various launch options are evaluated for each architecture, looking at several different ground infrastructure scenarios. A "best" launch option is then chosen for each architecture. Next, a trade study between using one larger TLI and two smaller TLIs is done for the expendable architecture to see if any improvements can be made to the launch vehicle trade results. Finally, once the best elements have been selected for each architecture based on the above trade studies, a study is done on the number of missions per year. It is assumed that at low flight rates, the expendable architecture will be better while at higher flight rates the reusable architecture will be better. This theory will be examined and then final comments will be made on the relative merits of each architecture for the baseline lunar mission being modeled.

4.1 Baseline Results

For each architecture, detailed results are shown for the baseline set of vehicles outlined in Section 3.2. Cost values are broken down by type (DDTE, Production, Operations, Propellant, and Launch) and by element (CEV, Lander, TLI, TEI, and Launch Vehicles). Additionally, reliability numbers are presented, which include the loss of crew probability and the loss of mission probability, both on a per mission basis. The baseline mission duration is ten years, at a flight rate of two missions per year. For these results, the mean values are shown, as opposed to the 95% confidence bounds used for the subsequent trade studies. Only the key figures of merit are output to Excel, where the 95% one-sided confidence values are calculated. The variables in the detailed results are reported in Arena, where only the mean value and half-width are given (for a 95% two-sided confidence interval, which would correspond to a 97.5% one-sided value). Therefore, for simplicity, the mean values are presented as a baseline example for each architecture.

4.1.1 Expendable Architecture Baseline Results

Table 6 lists the baseline elements used in the expendable lunar architecture, as explained in Section 3.2. Using these elements, the CEV and TEI can both launch on the man-rated Delta IV Heavy, with a crew-escape system. The remaining elements can all launch on one Centurion C2, with no additional propellant launches. Additionally, only one TLI stage is needed. The baseline results presented are assuming unlimited ground infrastructure, such that the mission capture percentage is 100% and on average, all missions launch on time.

Table 6. Baseline Elements for Expendable Lunar Architecture.

Element	Name	Payload (kg)	Gross Mass (kg)	DDTE ($M)	Price/TFU ($M)	Fixed Ops ($M)	Var. Ops ($M)
Crew L.V.	Delta IV Heavy	22,850	---	3,258	321	---	---
Cargo L.V.	Centurion C2	100,000	---	6,000	785	---	---
TLI	Manticore	30,000	71,424	1,062	141	70	0.14
TEI	PPM	7,700	10,183	2,335	434	40	1.58
Lander	Eagle	500	11,550	1,124	194	3	0.96
CEV	Capsule	500	7,600*	2,500	300	50	20

*Mass does not include the launch escape system (additional 4600 kg)

Figure 11 shows the cost breakdown, by DDTE, Production, Operations, Launch, and Propellant costs. Launch costs make up the largest majority of the life-cycle cost, contributing to 38% of the program cost. DDTE is next, followed by Production. Operations and propellant costs, however, are a negligible part of the life-cycle cost (under 5%). This is expected for an expendable architecture, since none of the elements have to be refurbished or stored for long periods between launches. Table 7 then lists the actual cost values corresponding to Figure 11.

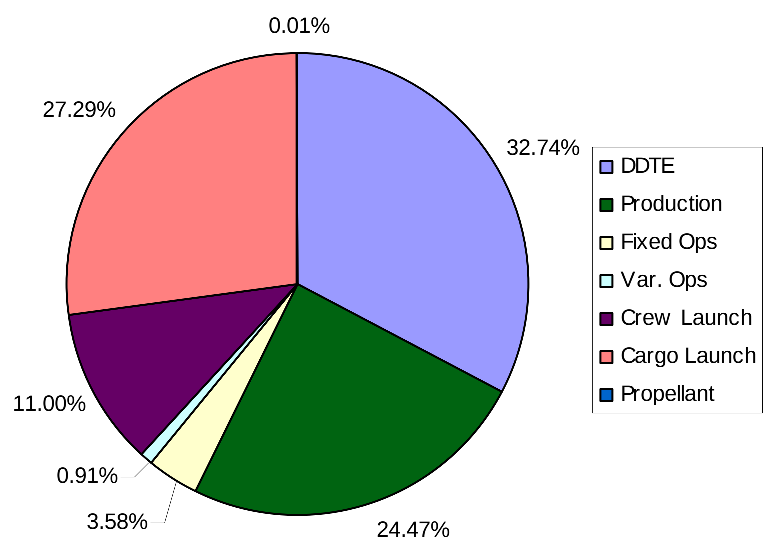

Figure 11. Cost Breakdown by Percentage for Baseline Expendable Lunar Architecture.

Table 7. Cost Breakdown for Baseline Expendable Lunar Architecture.

Breakdown	Cost ($M)
DDTE	17,110
Production	12,790
Fixed Operations	1,870
Variable Operations	480
Crew Launches	5,750
Cargo Launches	14,260
Propellant	3.0
TOTAL	52,260

Next, the costs are broken down by architecture element. Figure 12 plots the cost of each element, broken down by DDTE, Production, Fixed and Variable Operations, and Propellant. As can be seen from the figure, the launch costs are much higher than any of the individual element costs. Although each mission requires one of each type of launch, the cargo launch vehicle is much more expensive, both for DDTE and launch price. The costs associated with crew launches are actually comparable to the other transportation elements. After the launch vehicles, the TEI and CEV make up the largest portion. The largest percentage of each element's cost is due to production (launch costs in the case of the launch vehicles). Also as expected from Figure 11, the fixed operations costs do not contribute significantly to the cost of each element, and variable ops are such a small percentage that they do not even appear on the figure (except for CEV). Finally, propellant costs also have a negligible contribution to the cost per element.

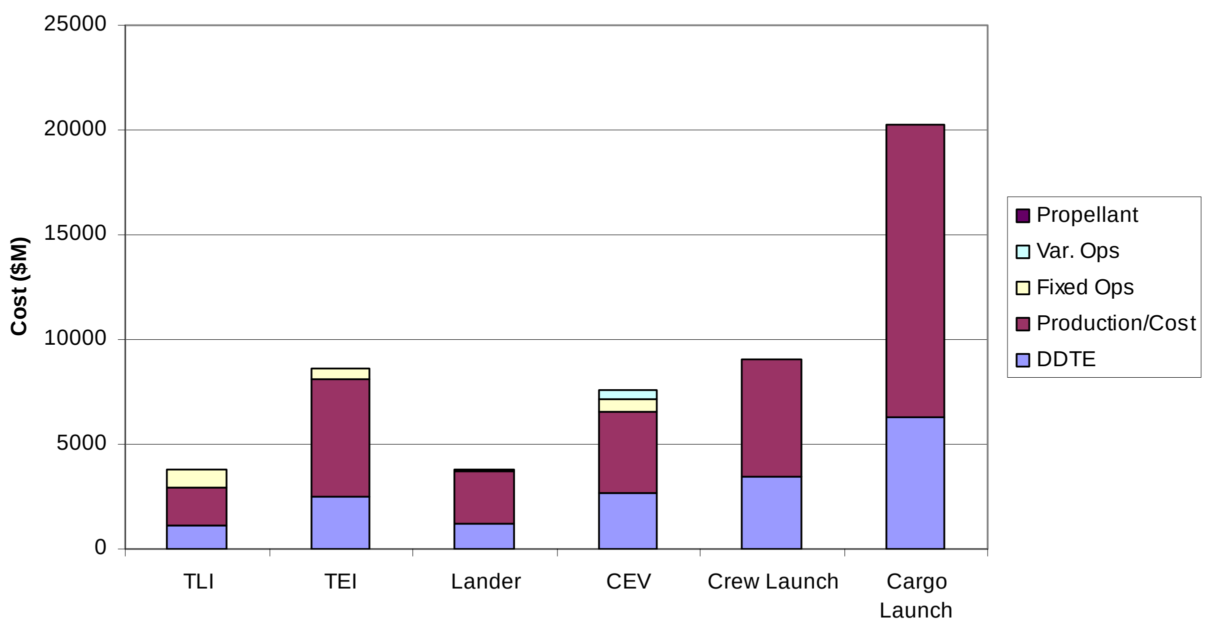

Figure 12. Cost Breakdown by Element for Baseline Expendable Lunar Architecture.

Finally, the baseline reliability numbers are as follows:

- Loss of Crew = 0.076
- Loss of Mission = 0.161

The loss of crew probability corresponds to approximately one in thirteen missions, while the loss of mission corresponds to between one in six and one in seven missions. Note that the loss of crew events are also counted as loss of mission events.

4.1.2 Reusable Architecture Baseline Results

Table 8 lists the baseline elements used in the reusable lunar architecture. A Delta IV Heavy is again used to launch the CEV/TEI, with a crew escape system. Cargo, propellant, and depot resupply launches all use the Centurion C2, with a payload capacity of 100 mt. For the baseline set of launch vehicles, one crew and one cargo launch are required per mission. As described earlier, each time a depot resupply mission must be sent, two additional launches are required, one for the TLI and one for the PRM.

Table 8. Baseline Elements for Reusable Lunar Architecture.

Element	Name	Payload (kg)	Gross Mass (kg)	DDTE ($M)	Price/TFU ($M)	Fixed Ops ($M)	Var. Ops ($M)
Crew L.V.	Delta IV Heavy	22,850	---	3,258	321	---	---
Cargo L.V.	Centurion C2	100,000	---	6,000	785	---	---
TLI	Manticore	30,000	71,424	1,062	141	70	0.14
TEI	PPM	7,700	10,183	2,335	434	40	1.58
Lander	Artemis	8,000	43,067	3,328	424	90	0.02
CEV	Capsule	---	7,600*	2,500	300	50	20
Propellant Depot±		50,000	---	---	---	---	---

*Mass does not include the launch escape system (additional 4600 kg)
±Gross mass, DDTE, and price of Prop. Depot not modeled.

The main difference in this architecture is the use of a reusable lander, which is significantly heavier and more expensive. It does not have a built-in habitat, so it must carry the entire CEV mass to and from the lunar surface. Additionally, the baseline propellant depot holds 50,000 kg. Its mass and cost are not modeled.

Figure 13 shows the cost distribution for the baseline set of elements. The "Depot-Related" cost category includes the launch and propellant costs associated with refueling the LLO propellant depot (only a negligible portion of this is propellant). It does not include the TLI-related costs, as these are included within the DDTE, production, operations, and propellant cost breakdowns. As can be seen, launch again comprises a large majority of the life-cycle cost. The depot launches alone make up over 45% of the costs, with crew and cargo launches contributing to another 21%. Most of the remaining costs consist of DDTE and Production. Operations and propellant costs make up a very small portion of the overall life-cycle cost. Table 9 lists the actual costs associated with the baseline reusable architecture. As can be seen, many of the costs are similar to the expendable architecture. DDTE is slightly higher due to using the reusable lander, while production costs are slightly lower since fewer landers need to be built (although more TLIs must be built). The major difference is the cost to launch propellant to LLO to refill the depot.

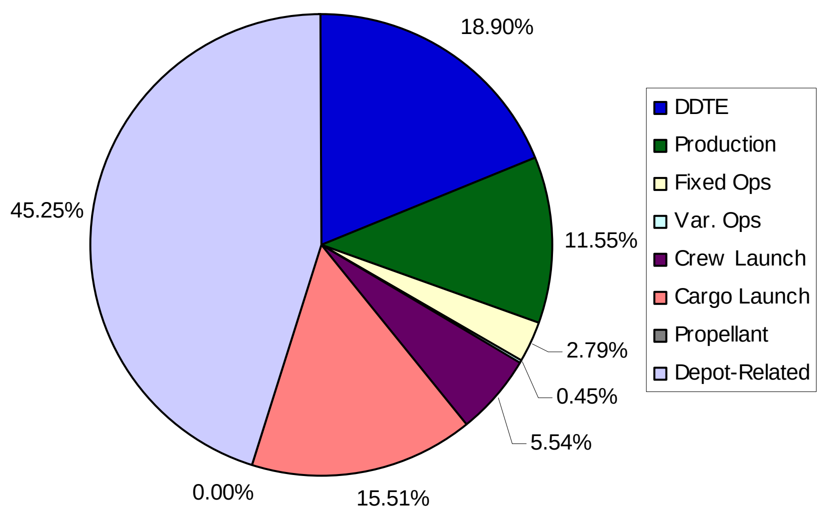

Figure 13. Cost Breakdown by Percentage for Baseline Reusable Lunar Architecture.

Table 9. Cost Breakdown for Baseline Reusable Lunar Architecture.

	Cost ($M)
DDTE	19,450
Production	11,880
Fixed Operations	2,870
Variable Operations	460
Crew Launches	5,700
Cargo Launches	15,960
Propellant	4.9
Depot-Related	46,560
TOTAL	102,880

Finally, Figure 14 breaks down the costs by architecture element, not including any launches, while Figure 15 includes the launch costs. Once again, launch costs dominate the total costs of the system. For the reusable architecture, the depot resupply launches make up almost half the life-cycle cost. Note that the depot-related launches do not incur a DDTE cost, since the same launch vehicle is used for cargo launches. In comparing these results to the expendable architecture, the TLI production costs are significantly higher than in the expendable architecture, since extra TLIs must be built to ferry propellant out to the depot. Additionally, the lander DDTE is higher, while the production cost is lower. The fixed operations costs are also higher for the reusable architecture. Propellant costs are still negligible, as was seen in Figure 12.

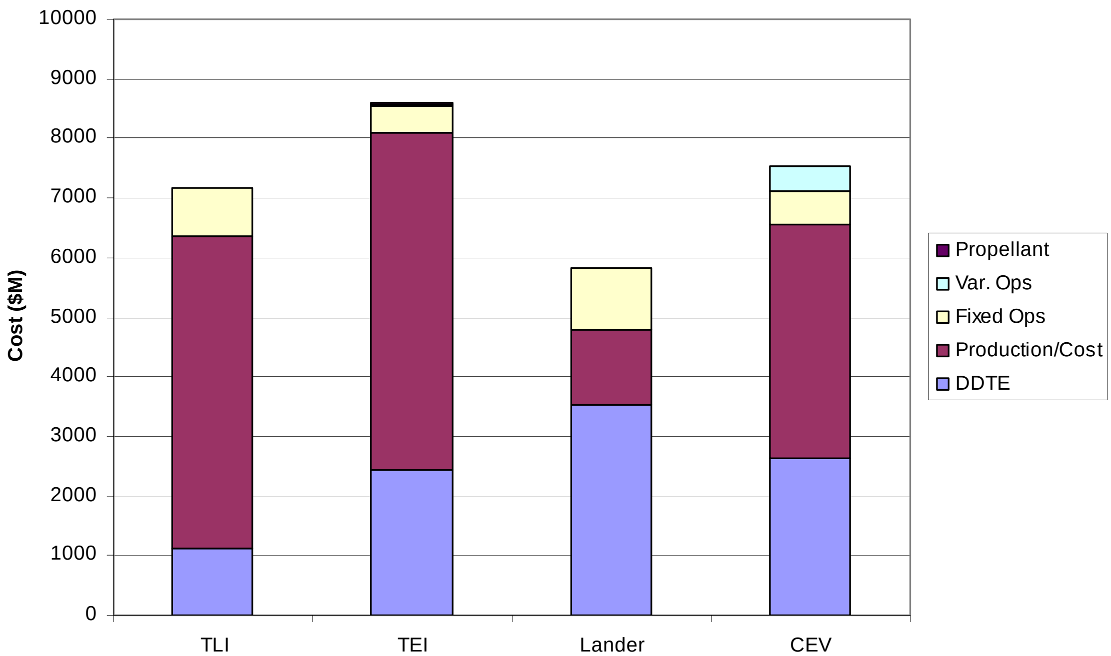

Figure 14. Cost Breakdown by Element for Baseline Reusable Lunar Architecture.

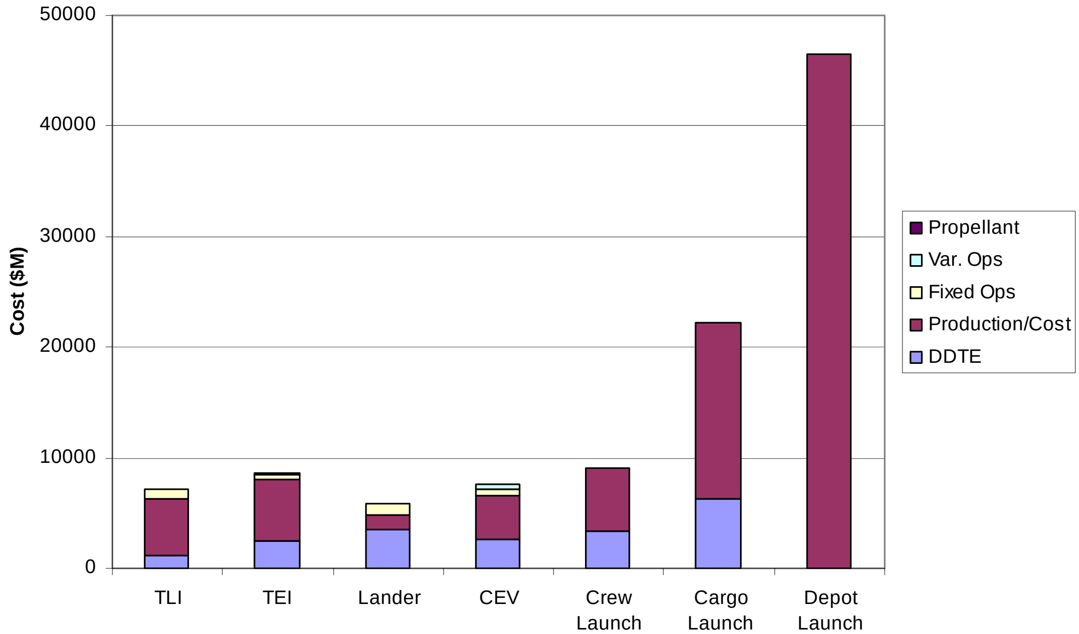

Figure 15. Cost Breakdown by Element Including Launch-Related Costs for Baseline Reusable Architecture.

Finally, the baseline reliability numbers are as follows:

- Loss of Crew = 0.072
- Loss of Mission = 0.150

The loss of crew and loss of mission probabilities are similar to the expendable architecture. This should be the case since both architectures use the same launch vehicle and both require one crew and one cargo launch per mission. The only difference in LOM or LOC would result from the lander, but the landers used in both architectures have the same reliability. Therefore, the small difference can be attributed to the probabilistic nature of the simulation.

4.2 Launch Vehicle Trade Study

In the first part of the study, four different launch scenarios (each with several subcategories) are examined for each architecture, in addition to the baseline. These were chosen to be representative of the launch options currently being examined for human lunar exploration. All of these launch scenarios assume that any dedicated propellant launches are carried out using the same launch vehicle as the cargo launches. Table 10 summarizes the various launch options examined in this study. Note that some existing launch vehicles are being used to launch crew, which requires that they be man-rated. The costs found in the database take this into account. For launch options 3a and 3b, where both launch vehicles come from the Centurion family, there is a discounted DDTE used to account for the decreased costs to develop both. While DDTE for one Centurion costs $6B, developing two in the same family will only cost $7B instead of $12B. Furthermore, the payload of the crew launch vehicles does not subtract the mass of a crew escape system – this is accounted for in the CEV launch mass. Detailed information on each launch vehicle can be found in Appendix B.

Table 10. Launch Vehicle Scenarios.

Launch Scenario		Crew L.V.		Cargo L.V.	
	Baseline	Delta IV Heavy	23 mt	Centurion C2	100 mt
1a	Existing EELVs	Delta IV Medium	8.7 mt	Delta IV Heavy	23 mt
		Atlas V 502	9.6 mt	Atlas V Heavy	19 mt
1b	Existing EELVs	Delta IV Heavy	23 mt	Delta IV Heavy	23 mt
		Atlas V Heavy	19 mt	Atlas V Heavy	19 mt
2a	Evolved EELVs	Delta IV Heavy	23 mt	Colossus	40 mt
		Atlas V Heavy	19 mt	Colossus	40 mt
2b	Evolved EELVs	Delta IV Heavy	23 mt	Colossus	70 mt
		Atlas V Heavy	19 mt	Colossus	70 mt
3a	HLVs	Centurion C1	35 mt	Centurion C2	100 mt
3b	HLVs	Centurion C1	35 mt	Centurion C3	140 mt
4a	Shuttle-Derived	SRB Stick	20 mt	Shuttle C	77 mt
4b	Shuttle-Derived	ET Derived	34 mt	Shuttle C	77 mt

In addition to the launch vehicle combinations, several ground infrastructure scenarios are examined for each case, as listed below:

- Infrastructure Scenario #1: Unlimited ground infrastructure
- Infrastructure Scenario #2: Limited launch infrastructure; unlimited manufacturing capacity
- Infrastructure Scenario #3: Limited launch infrastructure and manufacturing capacity

First, unlimited ground infrastructure will be considered such that the mission capture rate will be 100%, and the launch delay time and the time of elements in LEO will be essentially zero (assuming that launches occur at the same time). This will be modeled by setting the capacity of the manufacturing, integration, and launch vehicle resources to a very large number such that their capacity is never reached. Unlimited ground infrastructure will enable comparison between the various launch scenarios based solely on cost and reliability. Next, a limited set of launch infrastructure will be used, as shown in Table 11, with unlimited manufacturing capacities. This scenario assumes that the companies responsible for manufacturing each element will be able to upgrade their facilities as necessitated by the demand. Using this scenario, the various launch scenarios can be compared based on scheduling metrics: mission capture percentage, launch delay time, and time in LEO.

Table 11. Available Launch Infrastructure for Infrastructure Scenario #2.

	Capacity
Crew Integration	1
Cargo Integration	1
Propellant Integration	1
Crew Launch Pads	1
Cargo Launch Pads	1
Propellant Launch Pads	1

Finally, the last scenario will examine both limited launch and manufacturing capacities, also to compare launch options based on the scheduling metrics. From this scenario, the limiting capacity in the system can be determined by examining the bottleneck statistics. The manufacturing capacities used for this scenario are listed in Table 12, while the launch capacities are the same as in Table 11. The limiting case is examined for manufacturing, where all the capacities are set to one. As explained earlier, the manufacturing capacity refers to the number of a particular element that can be built at one time.

Table 12. Manufacturing Capacities for Infrastructure Scenario #3.

Manufacturing	**Capacity**
CEV	1
Lander	1
TLI	1
TEI	1

4.2.1 *Expendable Architecture Results*

In comparing the various launch options, the baseline in-space elements are used for all cases to facilitate a fair comparison, as shown in Table 6. Based on the launch vehicle combination, a different number of cargo and propellant launches are required, as listed in Table 13. Again, it was assumed that a launch configuration with no propellant launches is favorable if possible, and only the TLI stage can launch empty. Additionally, a crew escape system is added to the mass of the CEV listed in Table 6 (an additional 4600 kg[13]) if the payload capacity of the crew launch vehicle is sufficient. Therefore, launch option 1a does not include a crew escape system. Also note that with certain launch configurations, the CEV and TEI can not launch together. The model does not account for any penalty associated with the TEI launching separately from the CEV.

Table 13. Launch Breakdown for Launch Vehicle Trade Study (Expendable Architecture).

Launch Option	Crew	Cargo	Propellant	Total Launches
Baseline	CEV/TEI	1	0	2
1a (Delta)	CEV	2	4	7
1a (Atlas)	CEV	3	4	8
1b (Delta)	CEV/TEI	1	4	6
1b (Atlas)	CEV	3	4	8
2a (Delta)	CEV/TEI	1	2	4
2a (Atlas)	CEV	1	2	4
2b (Delta)	CEV/TEI	1	2	4
2b (Atlas)	CEV	1	2	4
3a	CEV/TEI	1	0	2
3b	CEV/TEI	1	0	2
4a	CEV	2	0	3
4b	CEV/TEI	2	0	3

4.2.1.1 Unlimited Ground Infrastructure

For each launch scenario, cost and reliability are the figures of merit of interest, since capture rate is 100% and launch delay and time in orbit are negligible. For cost, the primary statistic being examined is life-cycle cost. Cost per mission will also be looked at, but it exhibits the same trends as life-cycle cost. For reliability, the loss of crew and loss of mission probabilities are the statistics of interest.

For the baseline launch vehicle trade study, the crew and cargo launch vehicles listed in Table 10 are used, with propellant launching on the cargo launch vehicle. Figure 16 plots the results for life-cycle cost, and Figure 17 plots the cost per mission. The Atlas and Delta sub-cases for each launch scenario are indicated in parentheses on the figures.

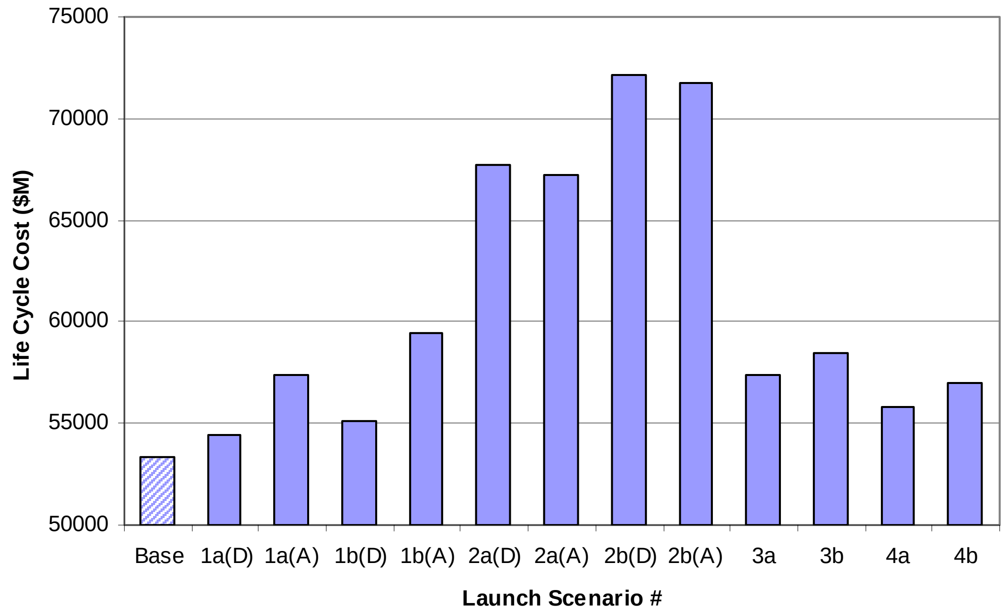

Figure 16. Life Cycle Cost for Launch Vehicle Trade Study (Expendable Architecture).

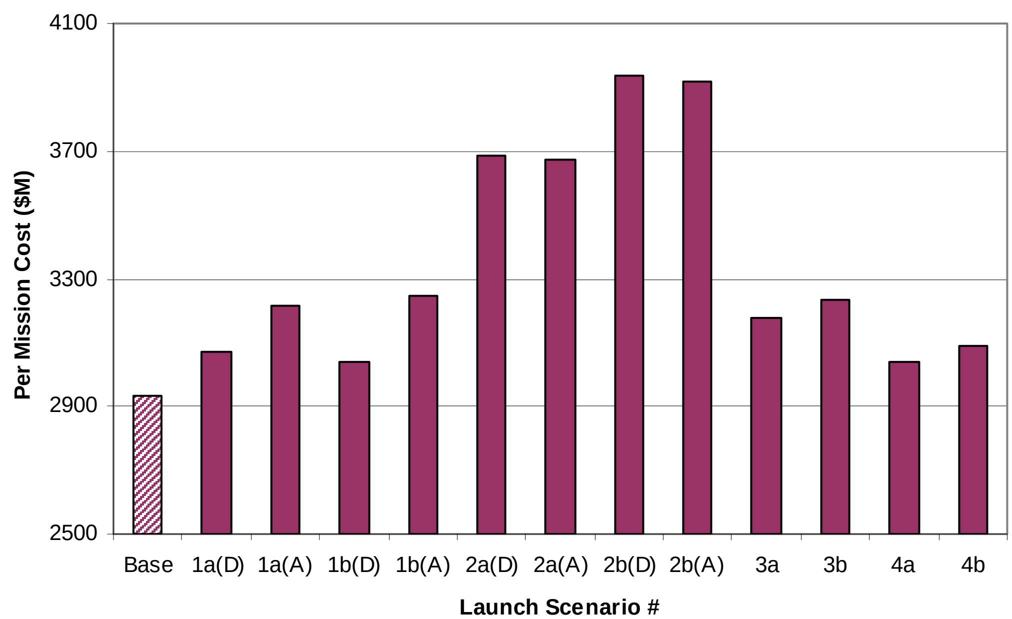

Figure 17. Cost Per Mission for Launch Vehicle Trade Study (Expendable Architecture).

As can be seen, life-cycle cost ranges from approximately $53 to $72 billion for 20 lunar missions in a ten-year period, with cost per mission ranging from $2.9 to $3.9 billion. If a launch vehicle configuration were to be chosen based only on cost, clearly the baseline option would be chosen. Of the other options examined, the Existing EELV, Shuttle-Derived, and HLV options all did very well in terms of cost. The Evolved EELVs, however, cost significantly more than all of the other options.

Several trends can be observed from Figure 16 and Figure 17. The baseline, Existing EELV, Shuttle-derived and HLV options all come out much less expensive than the Evolved EELVs for several reasons. The baseline, Shuttle-derived and HLV launch vehicles require the fewest launches per mission, as seen in Table 13. Although the DDTE and launch price are more expensive, the fewer number of launches makes up for the higher costs. Additionally, the DDTE and particularly the launch cost tend to be proportional to payload, so even if the launch price goes up, fewer launches are needed anyways. Shuttle, even though it requires two cargo launches per mission as opposed to the one cargo launch for the HLV options, has a lower DDTE because it uses existing components. Additionally, launch costs are less than for the HLVs. Of course, with all the modifications recently made to Shuttle after the Columbia failure, these launch prices could be somewhat optimistic. The HLV launch options have a considerately higher DDTE and launch price than any of the other launch vehicle options, but only require two total launches per mission (one crew and one cargo). HLV option 3b comes out more expensive because it is over-designed for this class of missions. All of the cargo elements fit on the 100 mt launch vehicle, so upgrading to the 140 mt launch vehicle simply incurs additional cost without any added benefit.

The Existing EELVs are competitive in terms of cost for the opposite reason. They require significantly more launches, but they use either existing launch vehicles (no DDTE cost) or modifications to existing vehicles. The Atlas launch vehicles result in higher life-cycle costs because their smaller payload results in more required launches. The Evolved EELVs, however, did significantly worse in terms of cost. They have a bad combination of high DDTE costs, and high per launch costs, while still requiring four launches per lunar mission. Additionally, the Evolved EELV option exhibits the same problem as the HLVs, where the 70 mt Colossus is over-designed for the mission. Upgrading from a 40 mt Colossus to a 70 mt Colossus does not reduce the required number of launches per mission, so the increased cost has no added benefit.

Reliability is then examined for each launch scenario. Figure 18 plots the loss of crew (LOC) and loss of mission (LOM) probabilities for each case. The baseline case falls somewhere in the middle of the launch options being traded.

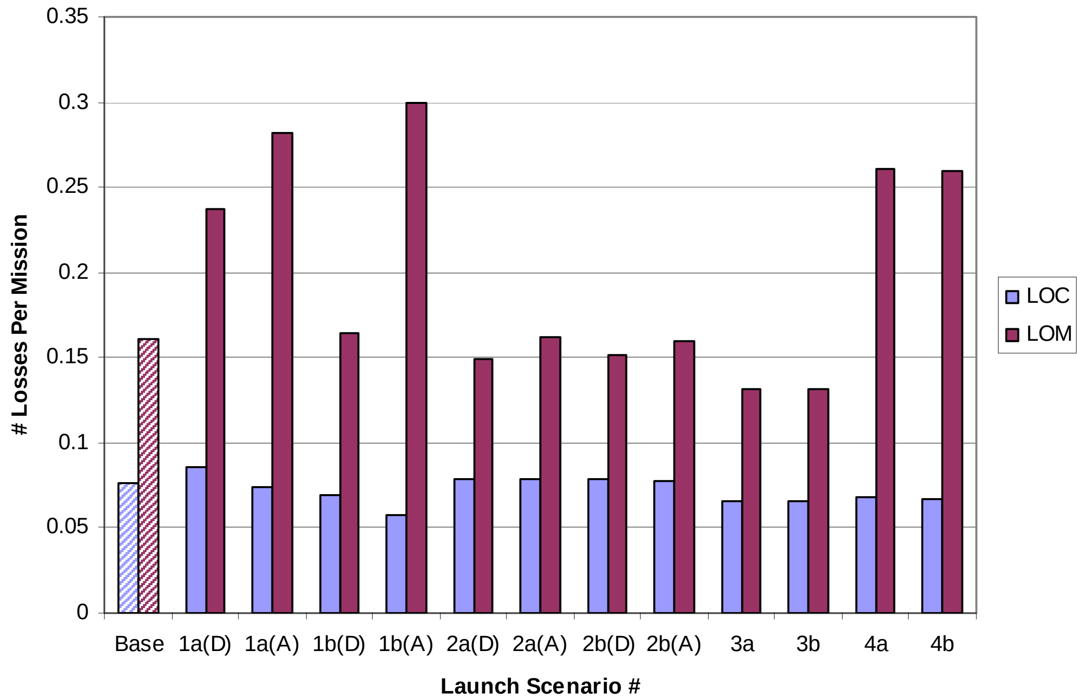

Figure 18. Reliability for Launch Vehicle Trade Study (Expendable Architecture).

Because all of the launch options have the same in-space segments and vehicles, the variations in LOC and LOM are due entirely to the choice of launch vehicles. LOC is due mainly to the reliability of the crew launch vehicle. Therefore, launch option #1a has a slightly higher LOC, since the launch vehicle payload is not sufficient to include a crew escape system. Even though the actual launch vehicle reliability is higher than the other Atlas and Delta crew launch options, the lack of an abort options increases the LOC number.

The loss of mission numbers can be attributed to a combination of the reliability of the cargo launch vehicle and the number of cargo launches. Since a mission is cancelled if a cargo launch fails, more cargo launches results in a higher LOM if the reliabilities are similar. This is clearly exhibited in the Existing and Evolved EELV options. Although there is a variation in reliability, as the number of cargo launches decreases from three (Existing EELV – Atlas) to two (Existing EELV – Delta) to one (Existing EELV – Delta, Evolved EELVs, and HLVs), the LOM also decreases. The loss of mission is the lowest for the HLVs both because only one cargo launch is required per mission and the reliabilities are fairly high. Finally, the Shuttle-derived vehicles tend to do much worse because of their relatively poor reliability.

The HLV options come out near the top for both cost and reliability. The Shuttle-derived and Existing EELV options, on the other hand, tend to have low life-cycle costs but among the worst reliabilities. Therefore, it is useful to examine a combined cost-reliability overall evaluation criterion (OEC) to examine the trade between cost and reliability:

$$OEC = \left[\frac{1}{2} * \left(\frac{LCC}{200000}\right)\right] + \left[\frac{1}{2} * \left(\tfrac{1}{2} LOC + \tfrac{1}{2} LOM\right)\right]. \qquad (7)$$

The chosen OEC represents an even split between cost and reliability. Life-cycle cost was normalized such that the values are in the same range as the LOC and LOM results.

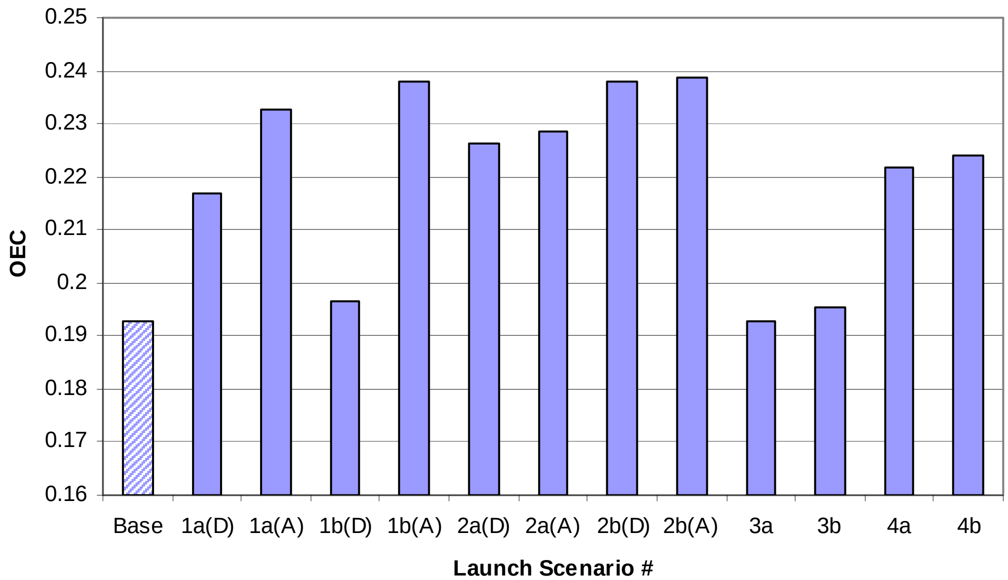

Figure 19. Overall Evaluation Criterion (Cost and Reliability) for Launch Vehicle Trade Study (Expendable Architecture).

Figure 19 plots the OEC for each launch scenario, where smaller is better. The baseline launch configuration has lowest OEC. Of the remaining options, because they did well in both cost and reliability, the HLVs and the Existing EELV #1b(D) have the next lowest OEC of any of the options being studied. The remaining Existing EELVs and Shuttle-derived options have fairly high OECs. Even though their cost was very low, the poor reliability increases their OEC. The Evolved EELV options ended up the worst, since they did poorly in both cost and reliability.

Therefore, depending on the metric of importance, a different launch vehicle choice would be made. Table 14 lists the rankings of each launch option for the three scenarios examined: cost, reliability, and a 50-50 split. As illustrated in the figures, after the baseline configuration, the HLV, Existing EELV (with Delta vehicles), and Shuttle-derived options are the best if cost is the driving factor. If reliability is the most important factor, however, HLVs do the best, but Shuttle-derived vehicles fall to near the bottom. The Evolved EELVs do much better in terms of reliability than they did for cost, although it is not sufficient to make them an attractive choice, because their life-cycle costs are appreciably more. Therefore, if cost and reliability are combined, the baseline launch option is the best, followed by the HLVs and Existing EELVs with Delta launch vehicles.

Table 14. Launch Vehicle Rankings for Expendable Architecture.

	Cost	Reliability	OEC
Base	1	6	1
1a (D)	2	9	5
1a (A)	7	12	10
1b (D)	3	5	4
1b (A)	9	13	11
2a (D)	11	3	8
2a (A)	10	8	9
2b (D)	13	4	12
2b (A)	12	7	13
3a	6	1	2
3b	8	2	3
4a	4	11	6
4b	5	10	7

The above results assumed that the same launch vehicle was used for both cargo and propellant launches. Another case to be examined is using a reusable launch vehicle for the dedicated propellant launches. The advantage of a reusable launch vehicle is its fast turn-around time and it cheaper launch costs. It is assumed that Vega[14] is used as the reusable launch vehicle, and that it is a commercial venture, such that there is no DDTE cost incurred. NASA would simply pay for each launch as needed. Figure 20 plots the life-cycle cost when the reusable launch vehicle is used for the dedicated propellant launches, at a cost of $127M per flight. For each case, using a reusable launch vehicle significantly increases the life-cycle cost, except for the cases where no propellant launches are required. Although the price per launch is significantly less, the payload is also less, requiring more propellant launches per mission. The fast turn-around time of a reusable launch vehicle is also not a factor in this case, since unlimited ground infrastructure is assumed for all cases. Furthermore, as expected, the life-cycle costs using Vega for propellant launches follow the same trend as that shown in Figure 16. Any slight variations are due to the probabilistic nature of the simulation.

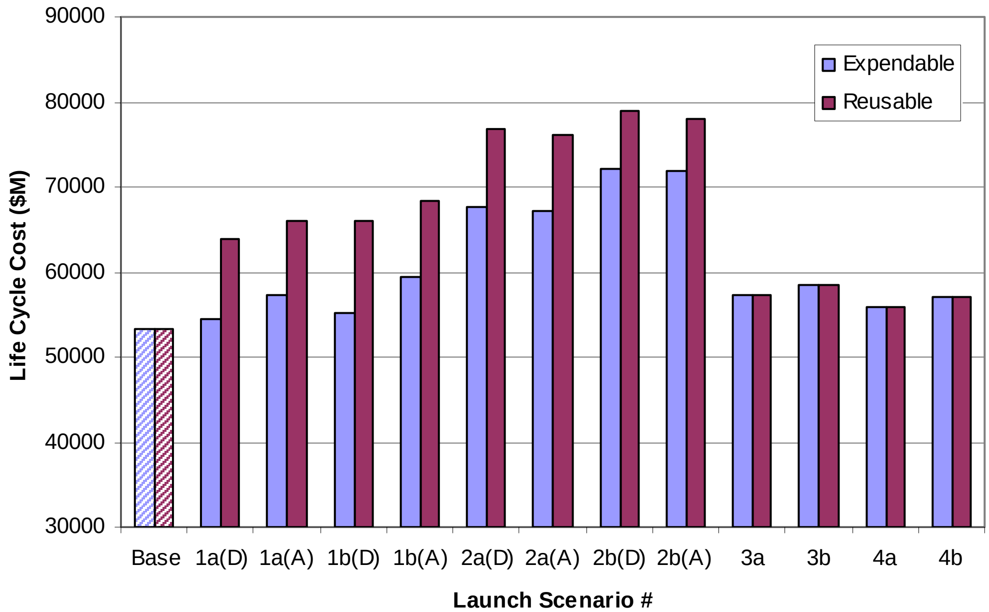

Figure 20. Expendable vs. Reusable Launch Vehicle Trade for Propellant Launches (Expendable Architecture).

Clearly, it does not make sense economically to use this reusable launch vehicle to deliver propellant to Earth orbit, at least when turn-around time is not an issue. This conclusion is based on the $127M launch price of the reusable launch vehicle. If the same launch vehicle were to be offered at a discounted price, then it may become competitive with the expendable launch vehicles. Figure 21 considers the effect of a 25% reduction in the launch price ($95M per flight), and Figure 22 plots the effect of a 50% reduction in launch price ($65M per flight). As can be seen, at 25%, the reusable launch vehicle is still not particularly competitive. Only for the Evolved EELV option with a 70 mt Colossus does using the reusable launch vehicle result in a decrease in life-cycle cost. This is among the worst launch vehicle options based on the above analysis and would most likely not be chosen anyways. At a 50% reduction in launch price, however, there is a reduction in life cycle cost across all launch options that require dedicated propellant launches. Therefore, the launch price of a reusable launch vehicle would have to be less than $100M per flight to become competitive for one launch option, and would have to be reduced even further to become beneficial across the board. Of course, reducing the launch cost by that amount may be infeasible, implying that using expendable launch vehicles for the propellant launches is the preferred option.

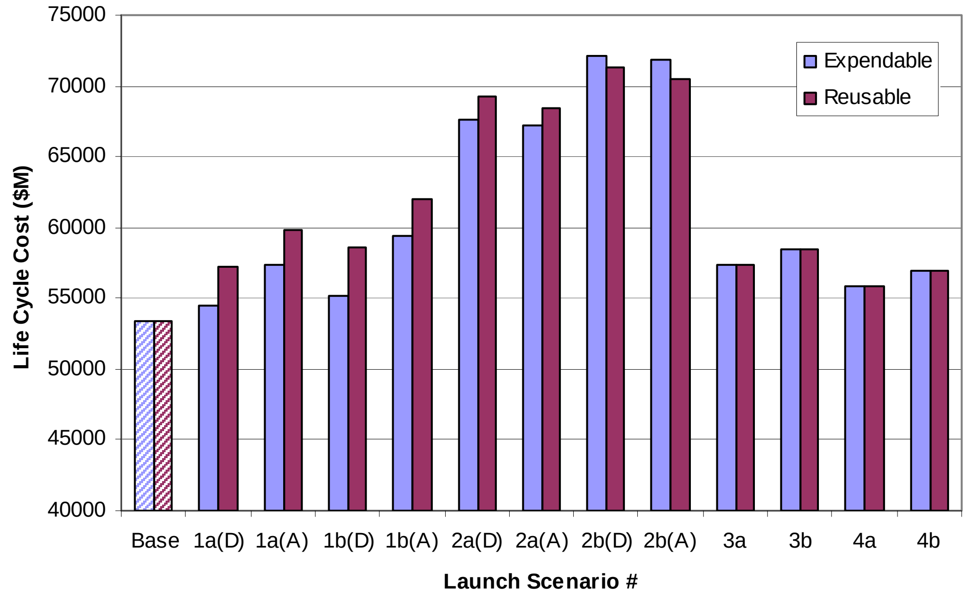

Figure 21. Expendable vs. Reusable Launch Vehicle Trade for Propellant Launches with a 25% Discount in Launch Cost (Expendable Architecture).

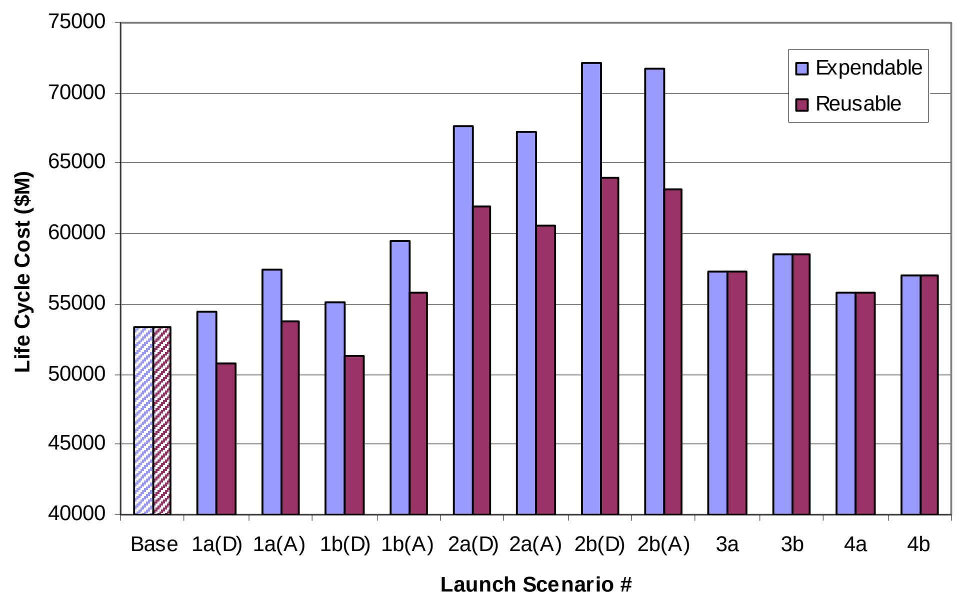

Figure 22. Expendable vs. Reusable Launch Vehicle Trade for Propellant Launches with a 50% Discount in Launch Cost (Expendable Architecture).

4.2.1.2 Minimum Launch Infrastructure

For the case of limited launch infrastructure, using the values from Table 11, the scheduling figures of merit become applicable. It is expected that the launch configurations that require more launches should have a lower mission capture percentage, a longer launch delay, and a longer time in LEO. Even using the limiting case (capacity of one for all infrastructure and launch pads), the mission capture rate is 100% for all cases except for 1a(A) and 1b(A). Even for these two launch options, the capture rate is still 98%. Therefore, at a launch rate of two missions per year, the launch infrastructure is not a limiting factor in meeting the mission demand. The launch delay is also essentially zero for all cases, since on average the first launch per mission launches on time.

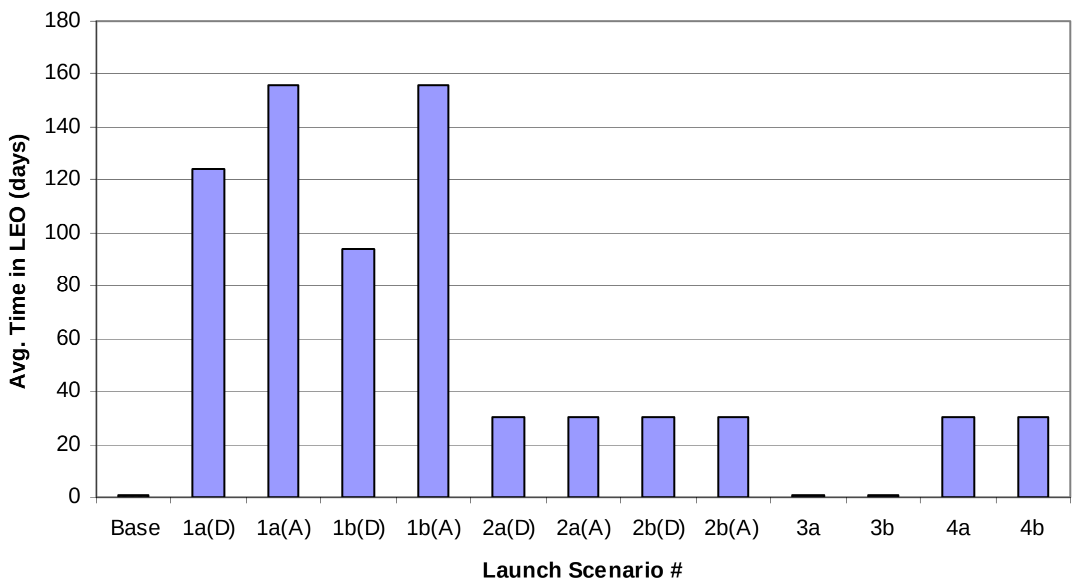

Figure 23. Average Time in LEO for Launch Vehicle Trade Study (Expendable Architecture).

The key statistic for this infrastructure scenario is the average time in LEO – that is, the average time between the first and last launch per mission. As this time increases, the mission becomes more infeasible, since elements have to remain in orbit for long periods of time before departing for the Moon. For example, engines need to be able to start after a long time in orbit and more propellant must be launched to account for boil-off or zero boil-off technology must be utilized. The more launches that are required, the longer the time in orbit should be. This is confirmed by Figure 23, which plots the average time in Earth orbit for each launch option. The trends follow the number of launches required per mission, as listed in Table 13. For the Existing EELV options, the time in LEO ranges from three months to just over five months, which begins to become prohibitively long for elements to remain in orbit. Additionally, this launch option would not be able to achieve a greater mission demand, since it already takes almost half a year to launch one mission. If more than two missions a year were desired, the mission capture rate would fall below 100%. For the Evolved EELV and Shuttle-derived options, the time in orbit is around one month, and there is almost no time in orbit for the baseline and HLV options.

Cost and reliability comparisons are not shown for this scenario, since they are the same as in Figure 17 and Figure 18. Another OEC can be created though, now including the time in LEO statistic from this infrastructure scenario:

$$OEC = \left[\frac{1}{3} * \left(\frac{LCC}{200000}\right)\right] + \left[\frac{1}{3} * \left(\tfrac{1}{2} LOC + \tfrac{1}{2} LOM\right)\right] + \left[\frac{1}{3} * \frac{LEO\,Time}{1000}\right] \quad (8)$$

Cost, reliability, and LEO time now each have a weighting of 1/3. LEO Time is normalized such that the value is in the same range as the other two metrics. Figure 24 plots the new OEC for each launch configuration.

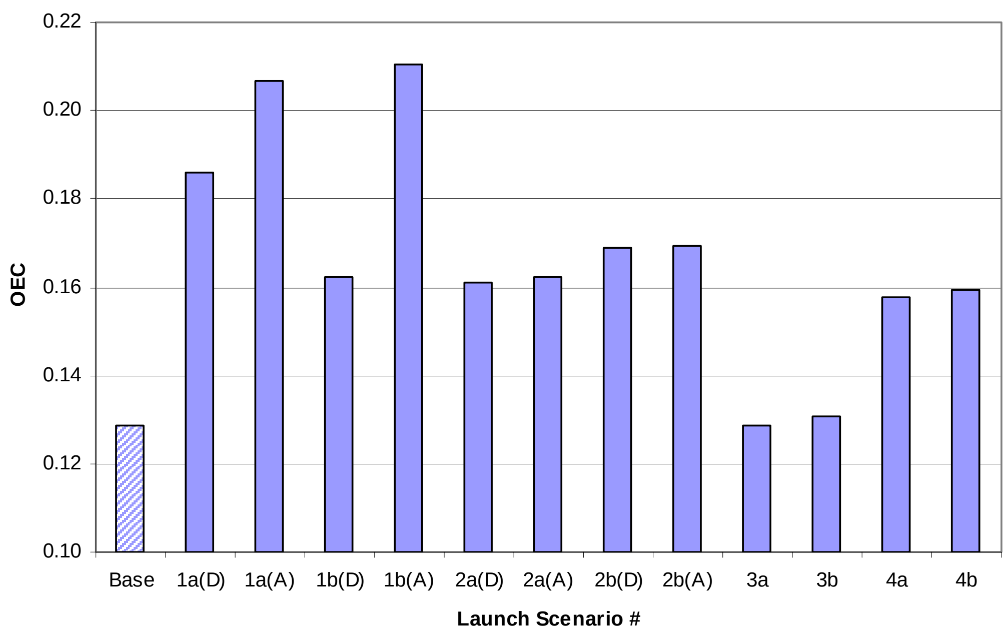

Figure 24. Overall Evaluation Criterion (Cost, Reliability, and LEO Time) for Launch Vehicle Trade Study (Expendable Architecture).

Once again, because the baseline launch configuration has the shortest time in LEO, its OEC value is the least among the available options. The HLV vehicles come in second, followed by the Shuttle-derived options. Although the Evolved EELVs are still not among the top choices, they do improve over the Existing EELVs due to their much shorter times in LEO. The Existing EELVs, which were among the best in terms of cost, now become the worst when both reliability and throughput are considered.

4.2.1.3 Minimum Launch Infrastructure and Manufacturing Capacity

As explained earlier, the final infrastructure scenario examined is limited launch infrastructure and manufacturing capacity, with the values listed in Table 11 and Table 12. While the minimum launch infrastructure did not provide a limiting case to mission throughput, the minimum manufacturing capacities do reduce the mission capture percentage below 100%. With unlimited manufacturing capacity, the launch delay was zero for all launch options. Because all elements must be retrieved from inventory before any launches can occur and because all launch options require the same set of in-space elements, the reduction in manufacturing capacity should create the same bottlenecks in the system regardless of the launch option. This in fact was the case, and Table 15 lists the average wait time in the queue for each manufacturing process. This represents the time between when an order is placed for an element and when it actually begins manufacturing. There was some small variation between each launch option, but that is due to the probabilistic nature of the simulation; therefore, the standard deviations are included for each case.

Table 15. Manufacturing Queue Wait Times for Launch Vehicle Trade Study (Expendable Architecture).

Manufacturing	Wait Time (days)	Standard Deviation
CEV	1060	3.9
Lander	1060	2.0
TLI	20	1.6
TEI	580	1.2

Clearly, the CEV and Lander manufacturing are causing the biggest bottleneck in the system, with the TEI manufacturing causing the next biggest bottleneck. The manufacturing capacity seems to be sufficient, however, for the TLI.

Because the manufacturing capacities affect each launch option equally for the baseline mission scenario, there is no launch option that would be favorable based on the least investment required in ground facilities (since the minimum launch infrastructure was sufficient for all launch options). Therefore, there is a set of minimum manufacturing capacities that all launch options would need to meet 100% mission capture. The baseline launch option was examined to determine this minimum capacity (Option 1a using the Delta launch vehicles), although any of the launch scenarios would yield the same result. Table 16 illustrates the manufacturing wait times, mission capture percentage, and launch delay time for three sets of manufacturing capacities. First, the lander and CEV manufacturing capacities are increased by one. Although this yields a significant improvement, the TEI wait time is still a problem. When the TEI capacity is also increased by one, the mission capture percentage goes to 100% and the launch delay time approaches zero. Even though there are still small wait times, the missions still launch on time. This is due to the buffer created by items needing to arrive in inventory a certain time before they are required for launch vehicle integration.

Table 16. Mission Capture Percentage and Launch Delay Time for Varying Manufacturing Capacities (Expendable Architecture, Baseline Launch Option).

CEV Capacity	1	2	2
Lander Capacity	1	2	2
TLI Capacity	1	1	1
TEI Capacity	1	1	2
CEV Wait Time	1060	20	20
Lander Wait Time	1060	20	20
TLI Wait Time	20	20	15
TEI Wait Time	580	580	0
Mission Capture %	0.52	0.80	1.00
Launch Delay	780	310	5

Therefore, it would not require a significant investment to create the ground infrastructure necessary for missions to launch on time, regardless of the launch option chosen. A manufacturing capacity of two is certainly feasible for any in-space element being produced.

Looking again at the results from Table 15, it should also be true that mission capture rate is equally affected by the reduced manufacturing capacities for all launch options since the bottlenecks are not caused by the launch infrastructure. This however was not the case when the simulation was run. The launch options with the most launches per mission actually had the highest mission capture percentage. The reason behind this was due to the fact that these launch options also tended to have the worst reliability. When missions were cancelled due to either loss of crew or a cargo launch failure, the downtime actually allowed the system to catch back up. For example, if two missions were cancelled due to stand-down time, the elements for those missions were already being built but were no longer required at that time (the reduced number of elements built is taken off at the end of the manufacturing process). Therefore, elements may actually reach inventory before they are required for launch. When fewer failures occur, there is less "catch-up" time so the capture percentage actually decreases. Therefore, to fairly compare capture percentage, all of the launch vehicles reliabilities would have to be made equal. The same is true of average launch delay time, which is inversely proportional to capture percentage. On average, the mission capture percentage for the minimum manufacturing capacities is 0.528 and the launch delay time is 752 days.

4.2.1.4 Expendable Architecture Summary

In general, when looking at cost, reliability, and throughput, the results for the launch vehicle trade tend to favor launch options that require fewer launches per mission. Cost depends on a trade between number of launches and cost per launch. Fewer launches do not necessarily translate to a lower life-cycle cost. When looking solely at cost, both the Existing EELVs (which require up to eight launches per mission) did as well as the HLVs (which required only two launches per mission). Reliability (particularly LOM), however, is strongly affected by the number of launches required, since there is not a large variation in the reliability of the actual launch vehicles being examined. Furthermore, as the launch infrastructure is limited, fewer required launches also translates into less time in LEO, which makes an architecture more feasible based on today's technology.

Additionally, for any of the launch options examined, no significant infrastructure investment needs to be made. One launch pad and an integration facility with a capacity of one for each type of launch vehicle are sufficient to launch all missions on time. Of course, to reduce the time in LEO for some of the launch options, more launch pads and integration facilities would have to be built, making those launch options more unattractive. Manufacturing capacities of two are also sufficient to meet the mission demands, which is a reasonable value to expect.

For all of the scenarios and launch options examined for the expendable Lunar Architecture, the baseline launch configuration remains the best option, in terms of a combination of cost, reliability, and scheduling. Next best are the two HLV launch options, which actually have the best reliability of all the launch options. If reliability were the primary concern, with less weighting on cost, the HLV options would be chosen. The baseline configuration does well overall though, because it uses an HLV as its cargo launch vehicle, while using a man-rated version of the existing Delta IV Heavy to launch the crew. Based on the overall consideration of cost, reliability, and scheduling, the baseline configuration is chosen as the best launch option for the expendable lunar architecture.

4.2.2 *Reusable Architecture Results*

As with the expendable architecture trade study, the baseline elements are used for all launch options for the reusable architecture, to facilitate fair comparison. These elements can be found in Table 8. Based on the launch vehicle combination, a different number of launches will be required for each lunar mission, with the possibility of having to launch elements dry and subsequently launch propellant to orbit. The launch breakdown by launch vehicle option is listed in Table 17. Additionally, each time a depot refueling mission is required, additional launches will be required, which is also shown in Table 17. For each depot resupply mission, one PRM is launched with a TLI to ferry it to lunar orbit. The TLI and PRM may be launched dry if necessary, requiring additional dedicated propellant launches. As explained earlier, the size of the PRM is determined by either the payload of the TLI stage or the payload of the launch vehicle, whichever is smaller. A depot refueling mission is called for any time the propellant in the depot drops below the amount required to fill the lander.

Table 17. Launch Breakdown for Launch Vehicle Trade Study (Reusable Architecture).

Launch Option	Crew	Lunar Mission		Depot Resupply Missions
		Cargo	Propellant	
Baseline	CEV/TEI	1	0	2
1a (Delta)	CEV	1	4	6
1a (Atlas)	CEV	2	4	7
1b (Delta)	CEV/TEI	1	4	6
1b (Atlas)	CEV	2	4	7
2a (Delta)	CEV/TEI	1	2	4
2a (Atlas)	CEV	1	2	4
2b (Delta)	CEV/TEI	1	2	4
2b (Atlas)	CEV	1	2	4
3a	CEV/TEI	1	0	2
3b	CEV/TEI	1	0	1
4a	CEV	2	0	2
4b	CEV/TEI	1	0	2

4.2.2.1 Unlimited Ground Infrastructure

The first scenario examined for the reusable architecture is that of unlimited ground infrastructure, such that the mission capture percentage is 100%, there is essentially no time spent in LEO, and on average, all missions launch on time. Figure 25 and Figure 26 plot the life-cycle cost and cost per mission (averaging the depot launches among all the lunar missions), based on the launch scenarios outlined in Table 10. Immediately apparent is the increased cost over the expendable architecture. The life-cycle costs range from $91B to $137B, with the cost per mission ranging from $4.9B to $7.4B. Although for some launch cases, there are fewer launches required per lunar mission, the added cost of launching propellant to lunar orbit significantly increases the life-cycle cost over the expendable architecture.

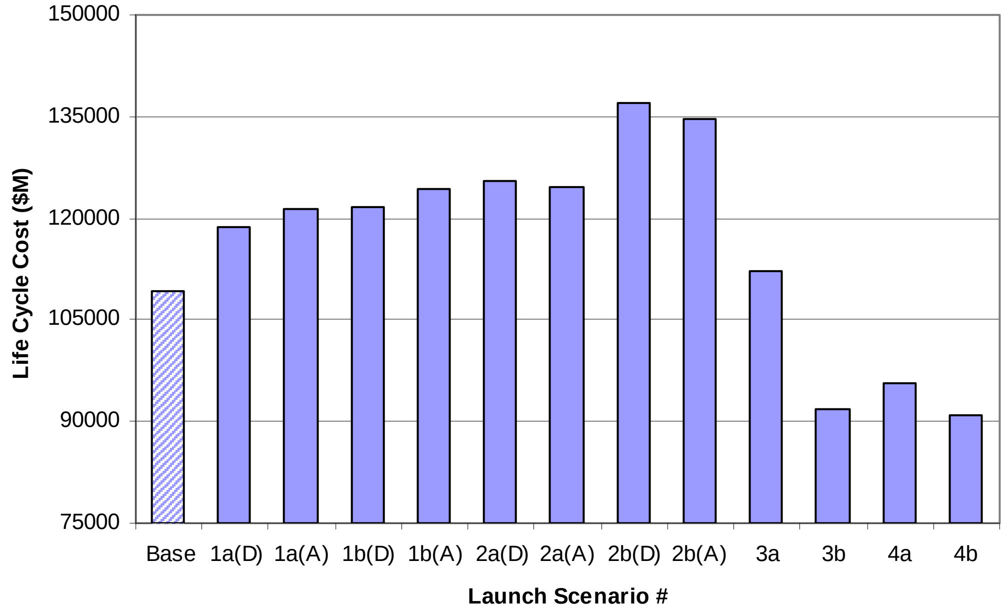

Figure 25. Life Cycle Cost for Launch Vehicle Trade Study (Reusable Architecture).

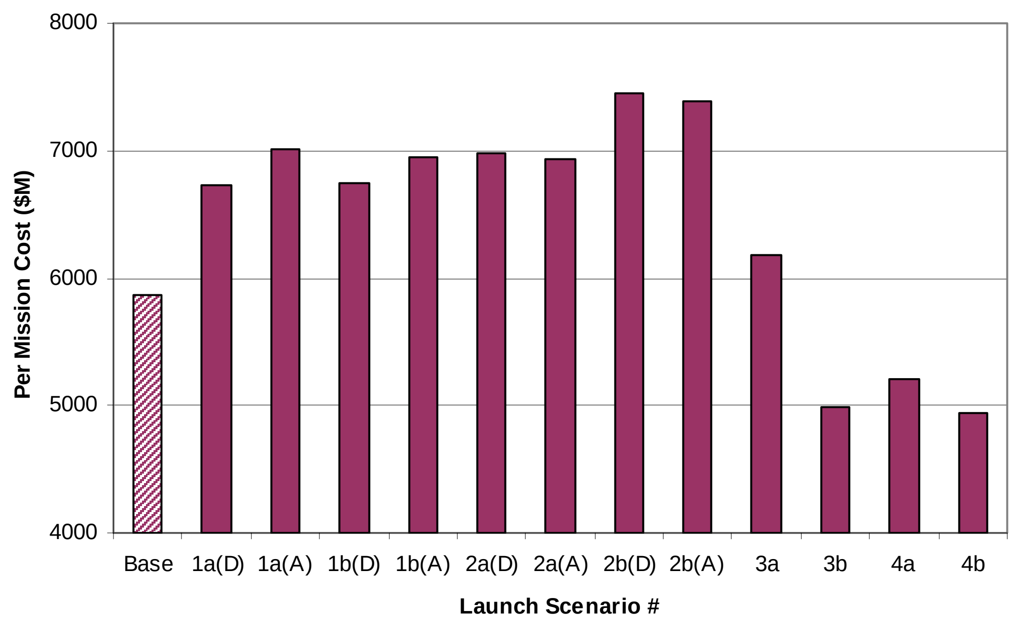

Figure 26. Cost Per Mission for Launch Vehicle Trade Study (Reusable Architecture).

For the reusable architecture, the baseline launch vehicle combination actually does not come out the least expensive. Both Shuttle-derived options and the HLV option with a 140 mt Centurion are the least expensive. Overall, these options require the fewest launches, both for the lunar missions and the depot refueling missions. The Existing and Evolved EELVs are significantly more expensive, partly because they require the most launches overall. Additionally, each depot resupply mission delivers less propellant, since the payload of the launch vehicles is the constraining factor, not the TLI payload.

Figure 27 plots the reliability for all of the launch options. As expected, the launch options with the least number of cargo launches per lunar mission have the lowest loss of mission probability. The propellant and depot launches will not affect reliability, since those missions are simply relaunched if there is a failure. Of the launch options that require one cargo launch per mission, option #4b has a slightly higher LOM than the other options due to the lower reliability of the Shuttle-derived vehicle. Loss of crew is again a function of the reliability of the crew launch vehicle. The highest values of LOC occur for options #1a, where a crew escape system does not fit in the launch vehicle.

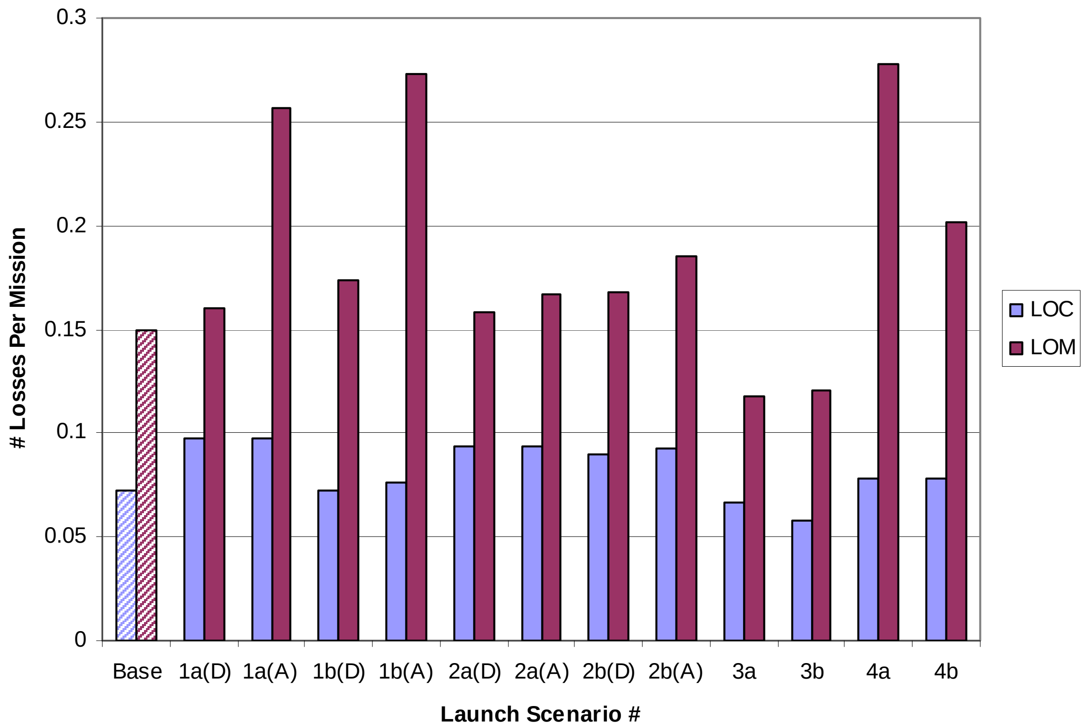

Figure 27. Reliability for Launch Vehicle Trade Study (Reusable Architecture).

When compared to the expendable architecture, the LOM numbers tend to be consistently lower for the reusable architecture. Again, this can be attributed to the fewer cargo launches required per lunar mission, since the lander does not have to be launched each time.

In general, the trends seen in cost are similar to those seen in reliability, with the exception of the Shuttle-derived vehicles that are low in cost but high in LOM probability. When looking at the OEC, the same launch options should be consistently the best across all weightings (again, with the exception of the Shuttle-derived vehicles). Figure 28 plots the cost-reliability OEC. As expected, most of the launch options are consistently at the top or the bottom, regardless of the weighting of cost and reliability. The HLVs and Shuttle-derived vehicles come out as the best options when looking at cost and reliability evenly weighted, followed by the baseline configuration. Table 18 lists the rankings of each launch option for the three weightings looked at: cost, reliability, and a 50-50 split. The baseline and HLV options are consistently among the best options. The OEC of the Shuttle-derived vehicles are among the best due to the significantly lower life-cycle cost. If reliability is very important, however, Shuttle-derived vehicles are among one of the worst choices. Therefore, based solely on cost and reliability, the best overall choice would be either one of the HLVs or the baseline.

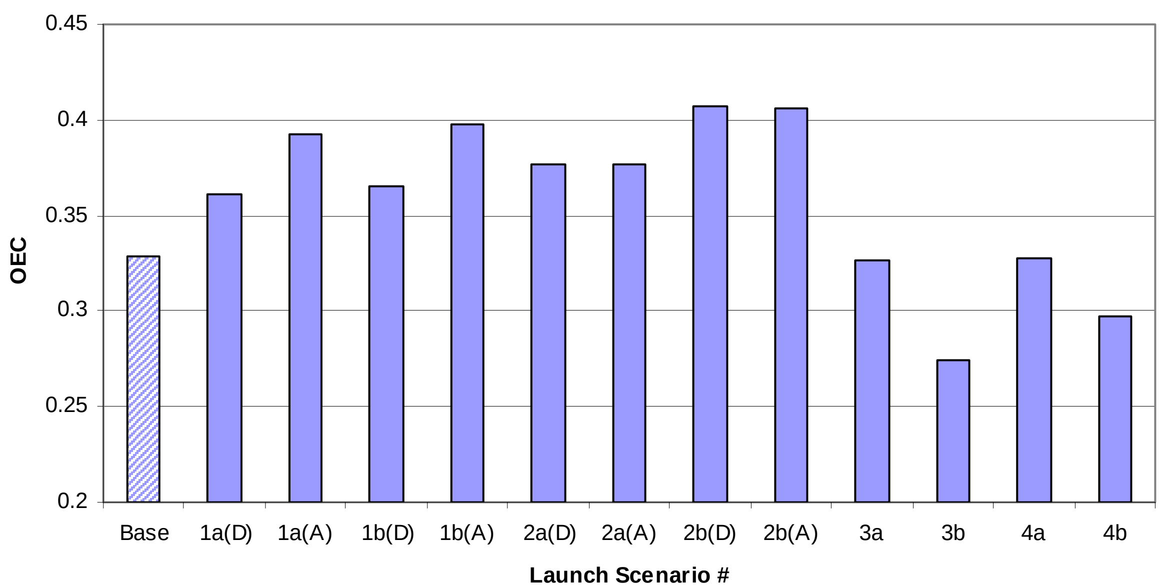

Figure 28. Overall Evaluation Criterion (Cost and Reliability) for Launch Vehicle Trade Study (Reusable Architecture).

Table 18. Launch Vehicle Rankings for Reusable Architecture.

	Cost	Reliability	OEC
Base	4	3	5
1a (D)	6	6	6
1a (A)	7	12	10
1b (D)	8	4	7
1b (A)	9	11	11
2a (D)	11	5	9
2a (A)	10	8	8
2b (D)	13	7	13
2b (A)	12	9	12
3a	5	2	3
3b	2	1	1
4a	3	13	4
4b	1	10	2

4.2.2.2 Minimum Launch Infrastructure

Limiting the launch infrastructure should have a significant effect on the reusable architecture, because of the additional launches required for refilling the propellant depot. When all of the integration capacities and number of launch pads are set to 1, as shown in Table 11, some of the launch options result in a mission capture percentage below 100%. Figure 29 plots the capture percentage for the minimum launch infrastructure case.

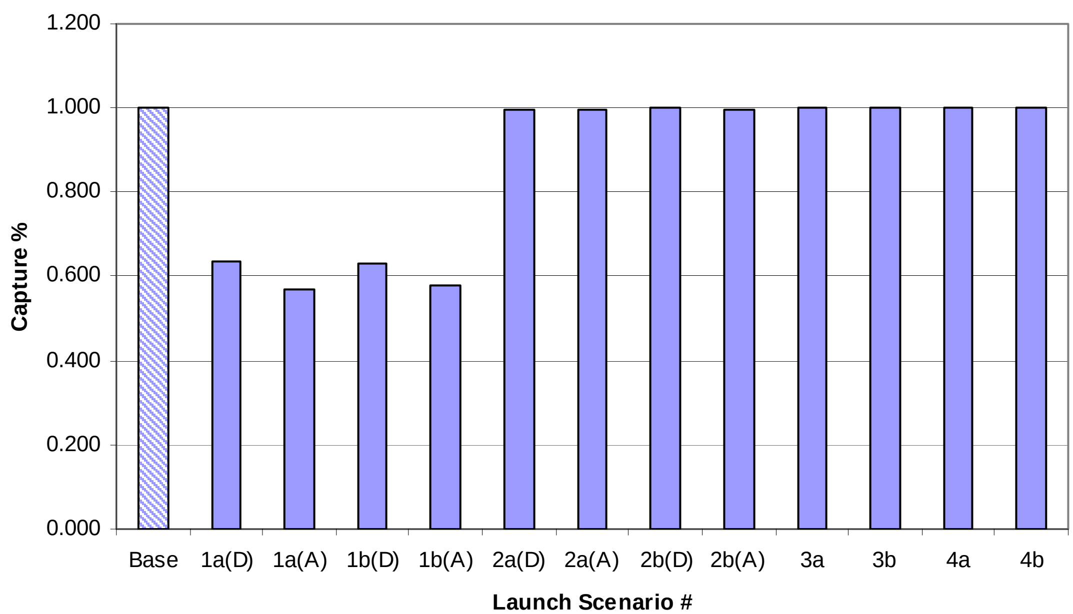

Figure 29. Mission Capture Percentage for Minimum Launch Infrastructure (Reusable Architecture).

Because of the smaller payload capacities of the cargo launch vehicles used in options 1 and 2, many more launches are required, as previously shown. Therefore, the Existing EELV options required greater launch infrastructure just to meet the baseline mission demand.

The average launch delay time is a mirror image of the above plot. For the options with 100% capture, all of the missions launch close to on time. As the capture rate decreases, however, the launch delay time also increases. For the Existing EELVs, the launch delay ranges from 662 days to 789 days. The longest average time in LEO is also the highest for the Existing EELVs, as expected, since it requires the most cargo and propellant launches for each lunar mission. Figure 30 plots the average time in LEO for each of the launch options.

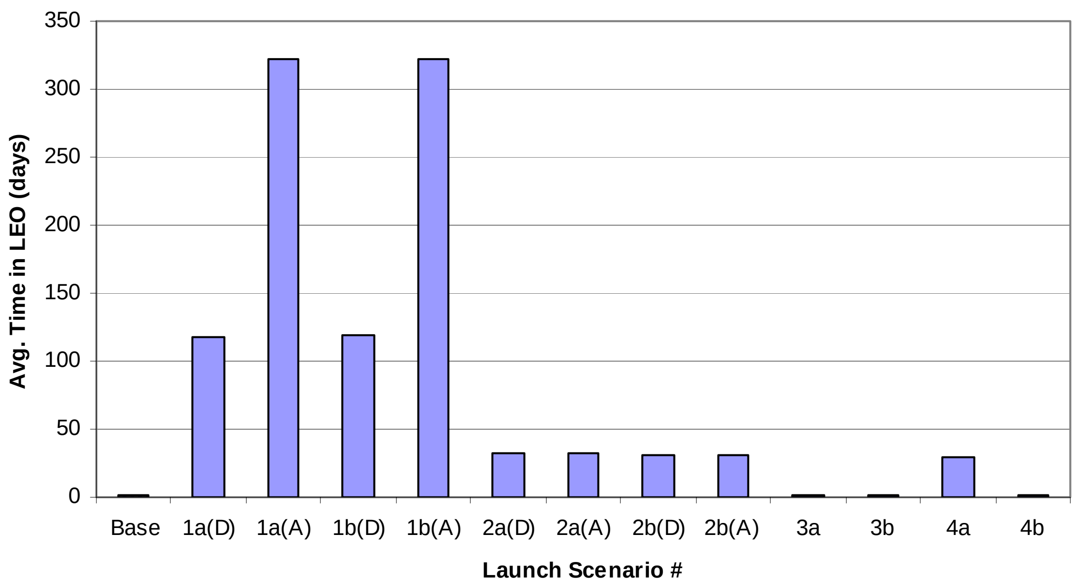

Figure 30. Average Time in LEO for Launch Vehicle Trade Study (Reusable Architecture).

A new OEC can now be created, with the addition of the scheduling metrics, which will be a combination of capture percentage and average time in LEO. Both will be included since each addresses a different issue. Capture percentage relates to the ability to actually meet the mission demand, while time in LEO relates to the feasibility of the architecture. As was seen with the expendable architecture, an architecture with a mission capture rate of 100% can still have a significant time in LEO penalty. The new OEC can therefore be written as follows:

$$OEC = \left[\frac{1}{3}*\left(\frac{LCC}{200000}\right)\right] + \left[\frac{1}{3}*\left(\tfrac{1}{2}LOC + \tfrac{1}{2}LOM\right)\right] + \left[\frac{1}{3}*\left(\frac{1}{2}*\frac{LEO\,Time}{1000}\right) + \left(\tfrac{1}{2}*\left(1 - Capture\%\right)\right)\right] \quad (9)$$

Mission capture rate is written as *(1-Capture%)*, since OEC is a quantity to be minimized and capture percentage should be maximized. Figure 31 plots this new OEC for each of the launch options. As expected, the same launch options come out the best (HLVs, Shuttle-derived, and baseline), since they also had among the highest capture percentages and lowest times in LEO. Adding the scheduling statistics does create a larger disparity between the best and worst launch vehicle configurations. Therefore, when all three components of the OEC are included, the Existing EELV launch options become an even worse choice for this architecture.

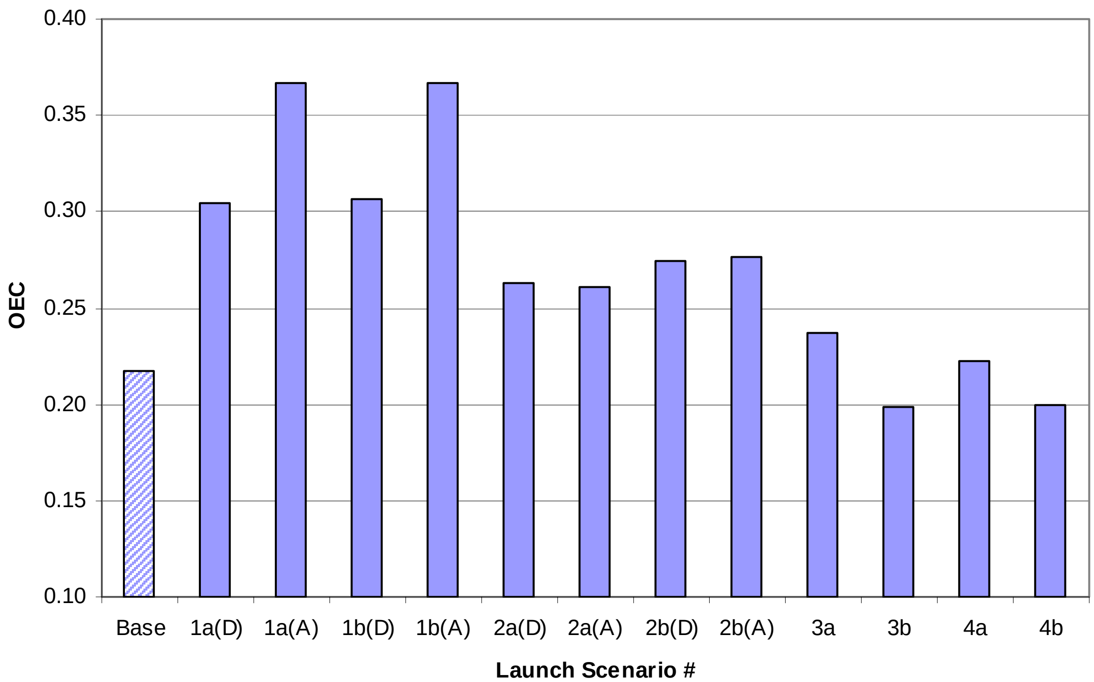

Figure 31. Overall Evaluation Criterion (Cost, Reliability, and Scheduling) for Launch Vehicle Trade Study (Reusable Architecture).

4.2.2.3 Minimum Launch Infrastructure and Manufacturing Capacity

Finally, the launch options are examined with the minimum possible launch infrastructure and manufacturing capacities. As in the expendable architecture trade study, manufacturing queue wait times are used as the bottleneck statistics for the CEV and TEI. The TLI and Lander manufacturing, however, is modeled by sending the total amount necessary for the program to the manufacturing facility all at the beginning of the simulation. Therefore, the manufacturing queue wait times will not be representative of the actual bottlenecks in the system. For the TLI and Lander, the bottlenecks will be represented by the average queue times while waiting to retrieve the element once it is needed. This represents the delay time between when a TLI or Lander is actually needed and when it is available in inventory. This is equally representative of the manufacturing bottlenecks, since the delay times are due to insufficient manufacturing throughput.

Table 19 lists the bottleneck statistics described above, as an average across all launch options, since each launch option uses the same number of elements for the lunar missions. As can be seen, the standard deviation of the TLI wait time is a higher percentage of the mean than the other values because the number of TLIs used depends slightly on the launch configuration. For the case where the propellant launched to the depot is constrained by the payload of the launch vehicles, slightly more depot refueling missions must be launched since the propellant mass per launch is less. When the propellant launched to the depot is constrained by the payload of the TLI, slightly fewer depot refueling launches must be sent. The effect, however, is not significant so providing the average value for the TLI wait time is still appropriate.

Table 19. Bottleneck Statistics for Launch Vehicle Trade Study (Reusable Architecture).

Element	Wait Time (days)	Standard Deviation
CEV	1110	2.7
Lander	490	2.1
TLI	450	22
TEI	620	0.74

The CEV and TEI wait times are approximately equal to those for the expendable architecture, as expected, since the lunar mission schedule is the same. Whereas TLI manufacturing did not cause a bottleneck in the expendable architecture, it does in the reusable architecture because more TLI stages are required for the depot launches.

Figure 32 plots the mission capture percentage for each launch option. The values are significantly less than the previous scenario where only launch infrastructure was limited. The Existing EELV options are lower than the others since they are also affected by the limited launch infrastructure. The average capture percentage of the remaining launch options is approximately 37%. Additionally, the time in LEO and the launch delay time both follow the same trend observed in the limited launch infrastructure case, just with higher values of launch delay time.

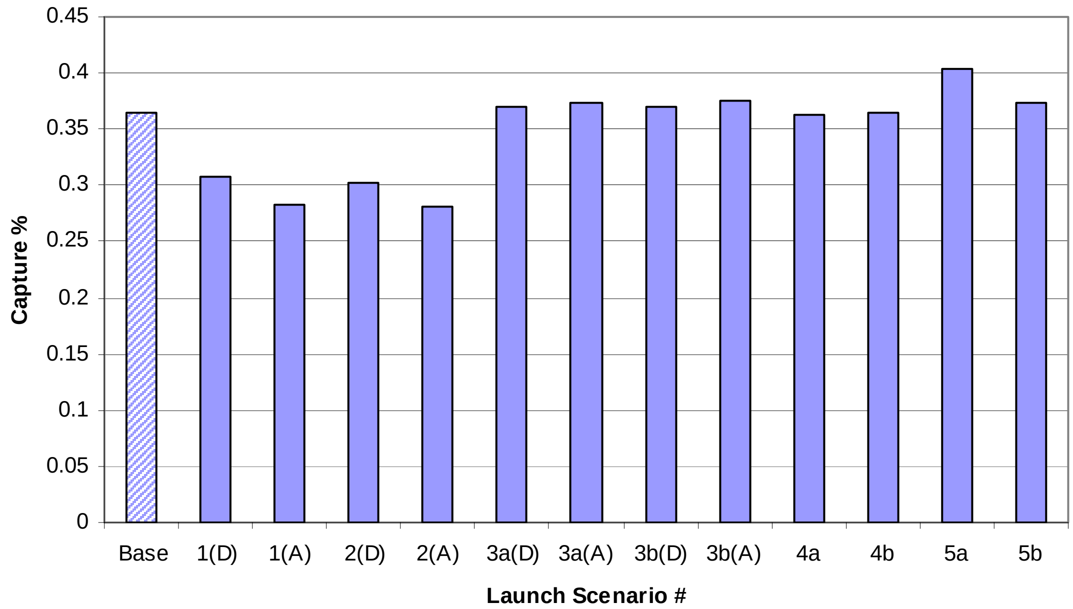

Figure 32. Mission Capture Percentage for Minimum Launch Infrastructure and Manufacturing Capacities (Reusable Architecture).

Once again, the baseline option was chosen as a representative launch configuration with which to determine the minimum manufacturing capacities necessary to reach a capture rate of 100%. Table 20 lists the manufacturing bottleneck statistics for varying manufacturing capacities, until the maximum capture rate is achieved.

Table 20. Mission Capture Percentage and Launch Delay Time for Varying Manufacturing Capacities (Reusable Architecture, Baseline Launch Option).

CEV Capacity	**1**	**2**	**2**
Lander Capacity	1	1	1
TLI Capacity	1	1	3
TEI Capacity	1	2	2
CEV Wait Time	1100	18	16
Lander Wait Time	490	500	490
TLI Wait Time	420	410	60
TEI Wait Time	620	0	0
Mission Capture %	0.37	0.37	0.996
Launch Delay	1490	1480	20

First the CEV and TEI manufacturing capacities were increased to two, as was done with the expendable architecture. This of course, has no affect on the capture percentage or delay time since the TLI bottleneck still exists. In order to eliminate the TLI bottleneck, the manufacturing capacity had to be increased to three. Although the lander and TLI wait times are still non-zero, the mission capture percentage is maximized (it never quite reached one when there was unlimited manufacturing capacities for this launch option) and the launch delay is small. In fact, increasing the lander and TLI manufacturing capacities beyond their current final values has no effect. The wait times that appear in the Arena output are due to wait times at the very beginning of the simulation before any lunar missions have launched. The model wants to launch the lander (which also requires a TLI) at the very beginning of the simulation, but it has to wait until one has been manufactured, which also corresponds to the time of the first lunar mission. Therefore, this wait time does not represent a bottleneck in the system, but instead a side effect of how the architecture was modeled. The final values in Table 20 represent the minimum manufacturing capacities necessary to meet the baseline mission demand for all launch options except the Existing EELVs. Clearly, this is more than was required for the expendable architecture, but is still within the realm of feasibility.

The Existing EELVs require a significant amount more infrastructure to meet the baseline mission demand, as can be seen in Table 21.

Table 21. Mission Capture Percentage and Launch Delay Time for Varying Manufacturing Capacities (Reusable Architecture, Launch Option 1b(A)).

CEV Capacity	2	2	2
Lander Capacity	1	1	1
TLI Capacity	3	4	4
TEI Capacity	2	2	2
Integration Crew	1	1	2
Integration Cargo	1	1	2
Integration Prop.	1	1	2
Launch Pads Crew	1	1	2
Launch Pads Cargo	1	1	2
Launch Pads Prop.	1	1	2
CEV Wait Time	18	17	20
Lander Wait Time	490	490	490
TLI Wait Time	90	50	50
TEI Wait Time	0	0	0
Mission Capture %	0.55	0.58	0.91
Launch Delay	870	780	170

First, the TLI manufacturing capacity had to be increased to four. The launch infrastructure also requires expansion. As can be seen from the table, even increasing all of the integration capacities and launch pads to two yields only a 91% mission capture percentage and a launch delay of 170 days. Increasing the capture percentage to 100% and eliminating the launch delay would require further increase in the launch infrastructure, which would be prohibitive in terms of cost and space.

4.2.2.4 Reusable Architecture Summary

As with the expendable architecture, the best launch option for the reusable lunar architecture when considering cost, reliability, and throughput is heavily dependent on the number of launches. This architecture is particularly sensitive to the payload capacity of the launch vehicles, since extra launches are required to continuously refuel the LLO propellant depot. Fewer overall launches tend towards a lower cost solution, while fewer cargo launches per lunar mission tend towards a lower loss of mission probability. Additionally, as the launch infrastructure and manufacturing capacities are limited, the launch options requiring more launches take a particularly big hit in terms of mission capture percentage and launch delay time.

The architecture investment required for the reusable architecture (with the exception of the Existing EELV options) is slightly more than for the expendable architecture, although it is still not enough to make the architecture infeasible. All of the manufacturing capacities are three or less, and the launch infrastructure is at a minimum. The Existing EELVs would require a more significant investment, particularly to the launch infrastructure. Most likely, however, these launch options would not be chosen because they also performed so poorly in cost and reliability.

Of the launch options examined, the ones that performed the best were the baseline, the HLVs and the Shuttle-derived vehicles. Because the Shuttle-derived vehicles did so poorly in reliability, however, neither of these will be chosen as the preferred launch option. The best overall of the remaining choice in terms of cost, reliability, and scheduling is the second HLV option, using the 140 mt Centurion. This launch configuration does so well because it only requires one cargo launch per lunar mission and one launch per depot refueling mission. Looking at the launch vehicles, available, however, another option was considered. While the Centurion C3 is a good choice for the cargo and propellant launches, the Centurion C1 is over-designed for the crew launches. Only 22 mt must be launched and its payload is 35 mt. The Delta IV Heavy is sufficient to launch the CEV with a crew escape system and the TEI at a lower per launch cost.

Therefore a "baseline improved" option was considered using the man-rated Delta IV Heavy to launch crew and the Centurion C3 to launch cargo and propellant. Table 22 summarizes the relevant figures of merit for the baseline improved case, along with its relative ranking for each. The reliability number listed is again a combination of LOC and LOM, and the overall OEC includes cost, reliability, and scheduling metrics, calculated using Equation 9. Clearly, the baseline improved case is preferable to all of the launch options already examined. While it ranks third in reliability, it is not significantly worse than the HLV options.

Table 22. "Baseline Improved" Launch Option Results (Reusable Architecture).

Metric	Units	Value	Ranking
LCC	$M	85,545	1
Reliability	---	0.108	3
Capture %	---	1.00	1
LEO Time	days	0.70	1
Overall OEC		0.176	1

Even this baseline improved case, however, is still significantly more expensive than even the most expensive expendable architecture option. Several improvements could be made to the current reusable architecture that could make it more competitive. The depot refueling launches are currently modeled as launching one mission at a time. Therefore, if the TLI and PRM do not fit on one launch vehicle, they are launched separately on that same type of launch vehicle, even if there is unused payload capacity on the launch vehicle. Better packaging would decrease the number of depot refueling launches required. The unused payload capacity could be used to launch another PRM or elements needed for a lunar mission, if their schedules coincided. Conversely, the PRM and TLI could be sized to best fit on the chosen launch vehicle (there would be two different TLI stages then – one for the lunar missions and one for the depot resupply missions). Additionally, more flexibility in the choice of launch vehicle could reduce the costs to refuel the depot. Several different expendable launch vehicles could be used throughout the lunar program, depending on the needs of that particular mission, or a different launch vehicle could be used for each different element being launched. Furthermore, the launch vehicle costs are fixed in the database. In reality, as the flight rate increases, the cost per launch would decrease, since the annual fixed costs would be spread over more launches. Because the reusable architecture requires more cargo and propellant launches per year, the costs per launch would actually be less than for the expendable architecture using the same launch vehicle. This could also help to improve the economic performance of the reusable architecture.

4.3 Expendable Architecture TLI Trade Study

For the launch vehicle trade study, the TLI stage was sized such that only one was needed to carry the Lander, CEV, and TEI to lunar orbit. Another option is to use two smaller TLI stages, such that one carries the Lander and the other carries the CEV/TEI stack. This would have the advantage of launching the TLI stages fully loaded instead of having to launch them empty and add propellant launches, depending on the payload capacity of the launch vehicle. Table 23 compares the larger TLI used earlier with the smaller TLI used for this trade:

Table 23. TLI Stage Sizes for TLI Trade Study.

	Payload (kg)	Dry Mass (kg)	Gross Mass (kg)
1 TLI	30,000	9,428	71,424
2 TLIs	18,000	6,418	44,812

Because the TLI is smaller and two are needed instead of one, the number of launches required per mission for each launch option may change. Table 24 lists the cargo and propellant launches needed for both TLI cases, as a point of comparison.

Table 24. Launch Breakdown for TLI Trade Study.

Launch Option	1 TLI Crew	1 TLI Cargo	1 TLI Propellant	2 TLIs Crew	2 TLIs Cargo	2 TLIs Propellant
Baseline	CEV/TEI	1	0	CEV	2	0
1a (Delta)	CEV	2	4	CEV	2	5
1a (Atlas)	CEV	3	4	CEV	2	5
1b (Delta)	CEV/TEI	1	4	CEV/TEI	2	5
1b (Atlas)	CEV	3	4	CEV	2	5
2a (Delta)	CEV/TEI	1	2	CEV/TEI	1	3
2a (Atlas)	CEV	1	2	CEV	1	3
2b (Delta)	CEV/TEI	1	2	CEV/TEI	2	0
2b (Atlas)	CEV	1	2	CEV	2	0
3a	CEV/TEI	1	0	CEV/TEI	2	0
3b	CEV/TEI	1	0	CEV/TEI	1	0
4a	CEV	2	0	CEV	2	0
4b	CEV/TEI	2	0	CEV/TEI	2	0

4.3.1 Unlimited Ground Infrastructure

The TLI trade is first conducted based on unlimited ground infrastructure, as was done in Section 4.2.1.1. As can be seen from Table 24, the only launch option that will benefit from using two TLIs is #2b, which goes from one cargo launch and two propellant launches to two cargo launches and no propellant launches, saving one launch per mission. Therefore, launch option #2b should be the only one to have a lower life-cycle cost when going to a two-TLI mission configuration. Additionally, the life-cycle cost of launch options 1a (Atlas) and 1b (Atlas) should be unaffected, since one cargo launch is simply being replaced by a propellant launch. HLV option 3a and both Shuttle-derived options should also remain unchanged, since their launch breakdown remains unchanged. This is confirmed by Figure 33, which plots the life-cycle cost for both the 1-TLI and 2-TLI mission options. The costs of launch options #2b are significantly reduced by adding a second TLI, since it saves one Colossus launch per mission. Even though building two smaller TLIs is more expensive than building one larger TLI, the savings gained in the launch costs far outweigh the added production cost. For some of the other launch options, however, there is a significant penalty in going to 2 TLIs because of the added cost of both extra launches and the extra TLIs built.

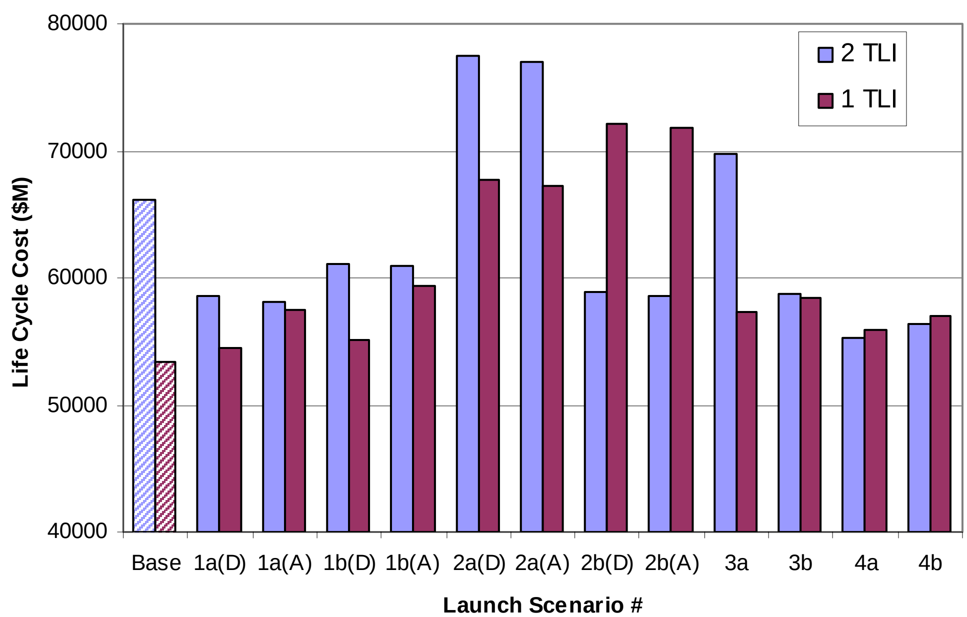

Figure 33. Life Cycle Cost for TLI Trade Study (Expendable Architecture).

Although the overall life-cycle cost can be significantly improved in a couple of cases by using two TLIs, the effect on reliability must also be examined. The loss of mission should be affected the most, for two reasons. First, adding a TLI adds another possibility of failure to the mission in the form of another set of TLI propulsive burns. Second, for many of the launch options, using two TLIs increases the number of cargo launches, which also introduces another possibility of failure. Figure 34 plots the LOM probability for both TLI scenarios. As expected, the

LOM probability increases with two TLIs for many of the launch options. The largest increase occurs for launch options that require an extra cargo launch. Even for options 1a(A) and 1b(A), where the number of cargo launches decreases by one, the LOM probability still increases slightly. The improved reliability of reducing the cargo launches is counteracted by the added failure possibility of another TLI burn. For all launch options, the loss of crew probability remains relatively unchanged. The added cargo launches have no effect on LOC, and only one TLI is used to carry the CEV to lunar orbit anyways.

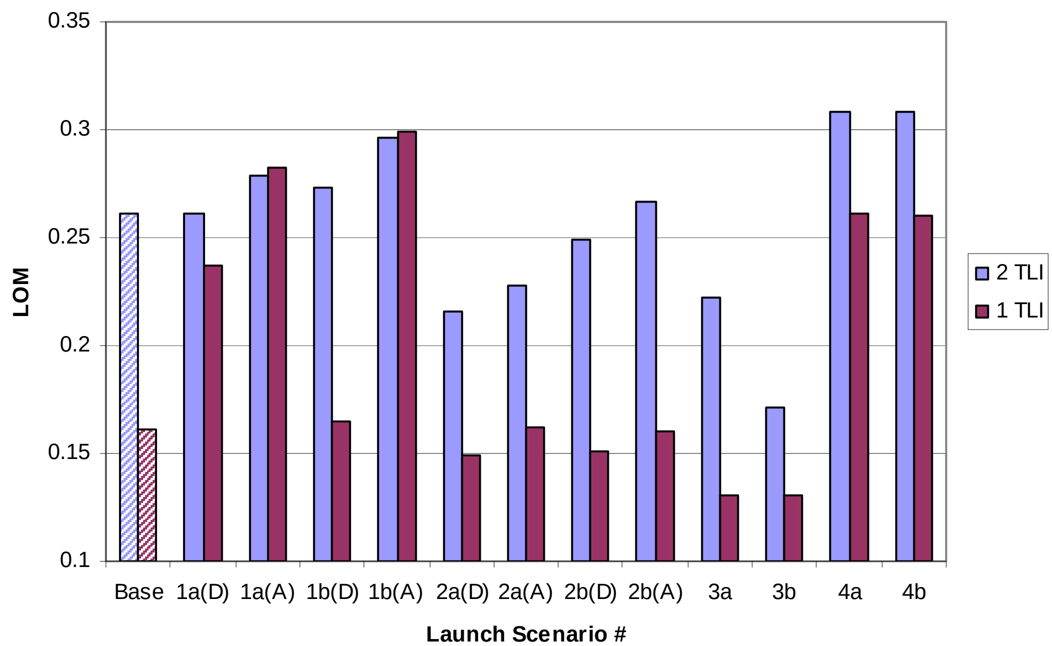

Figure 34. Loss of Mission Probability for TLI Trade Study (Expendable Architecture).

Therefore, the cost-reliability overall evaluation criteria is again examined, this time for the TLI trade study, to determine if there is any benefit to using two TLIs. It appears that for a majority of the launch options examined, there is no benefit to using two small TLIs as opposed to one large TLI. For launch options 1a and 1b using Atlas vehicles, there is clearly no benefit or detriment in terms of cost and reliability, since both remain relatively unchanged. There is a potential for improvement for launch option 2b, since two TLIs significantly reduces the cost, even though it increases the loss of mission probability.

Figure 35 plots the OEC from Equation 8, which includes life-cycle cost, LOC and LOM. Again, lower OEC values indicated a better launch choice. As can be seen, the only option that shows any benefit is #2b, and the improvement is small (less than a five percent reduction in OEC).

Although the LOM probability is worse for two TLIs in this case, the life-cycle cost is significantly better and the LOC probability is unchanged, resulting in a slightly lower value of the OEC.

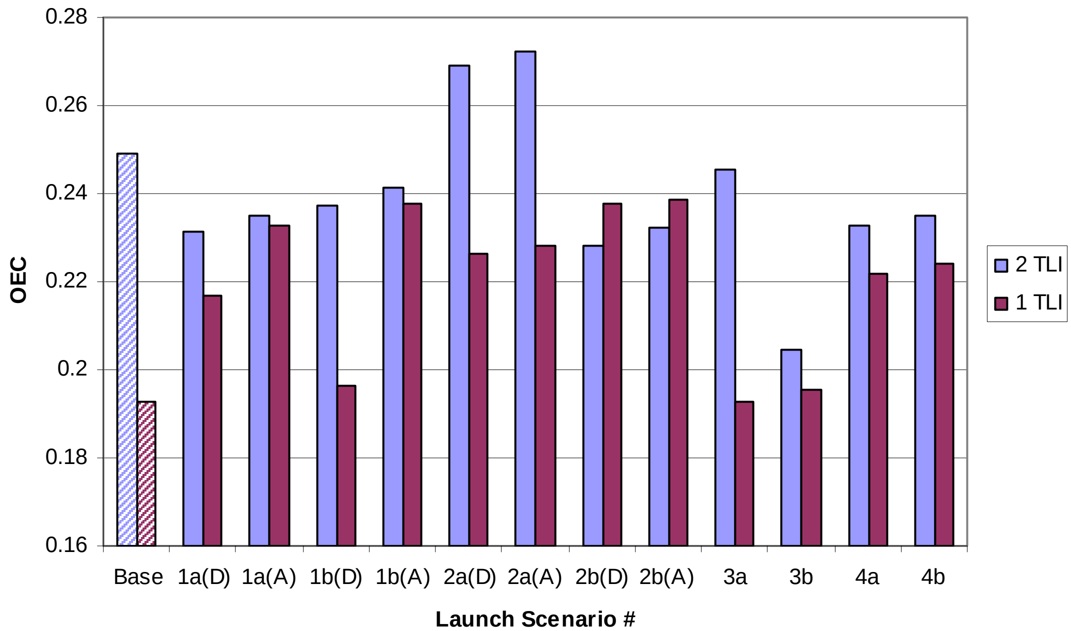

Figure 35. OEC Based on Cost and Reliability for TLI Trade Study (Expendable Architecture).

4.3.2 Minimum Launch Infrastructure and Manufacturing Capacity

The previous TLI trade results were based on an assumption of unlimited launch infrastructure and manufacturing capacities. It is important to again compare this case with the minimum set required to achieve the mission demand, particularly because two TLIs must now be built per mission, and in some cases, more launches are also required. Therefore, the two TLI case may result in even worse values of the OEC when the mission capture and scheduling statistics are included.

The first scenario examined is minimum launch infrastructure (launch pads and infrastructure capacities set to one) with unlimited manufacturing capacities. As with one TLI, the launch infrastructure will not be the limiting case for mission capture percentage. The capture percentages for two TLIs are all 100%, with the exception of launch options 1a and 1b, which all still have a capture percentage of 98%. The limited launch infrastructure, however, does affect the time in LEO, as shown in Figure 36. For the launch options that require more cargo and/or propellant launches for 2 TLIs, the time in LEO also increases. The time in LEO for launch options 1a and 1b using Atlas launch vehicles, 2b, 3b, 4a, and 4b are all unaffected by using a smaller TLI. Obviously, none of the launch options improve in terms of the scheduling figures of merit.

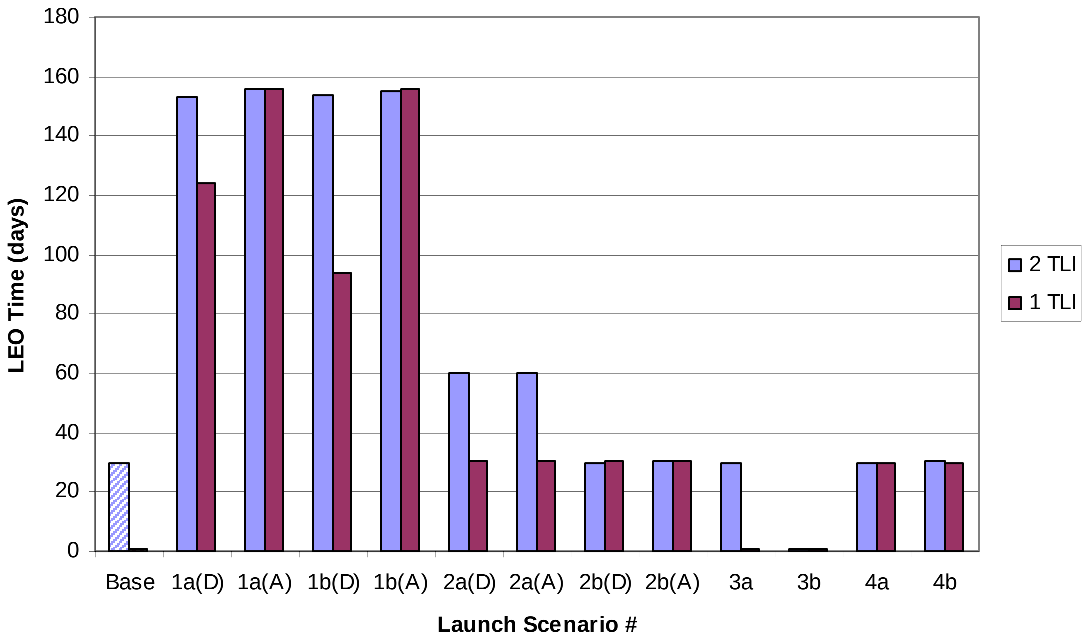

Figure 36. Average Time in LEO for TLI Trade Study (Expendable Architecture).

Therefore, when the time in LEO is added to the OEC (using Equation 8), as shown in Figure 37, one of the HLV options (using the 140 mt Centurion) becomes the best choice. The main difference is that the Evolved EELVs, which previously were among the worst launch choices, now are second best in terms of the overall OEC. They are still significantly though worse than option #3b, and showed little improvement over the 1 TLI OEC. The baseline case, which was chosen as best overall for the expendable architecture, now falls out of the top five best launch options. It is important to point out that while using a smaller TLI generally degraded the OEC for most launch options, it did not improve any of the launch options significantly either. The best OEC using 1 TLI is still better than the best OEC using 2 TLIs. Furthermore, there are more competitive launch options to choose from when using 1 TLI, as opposed to only one good choice (#3b) when using 2 TLIs.

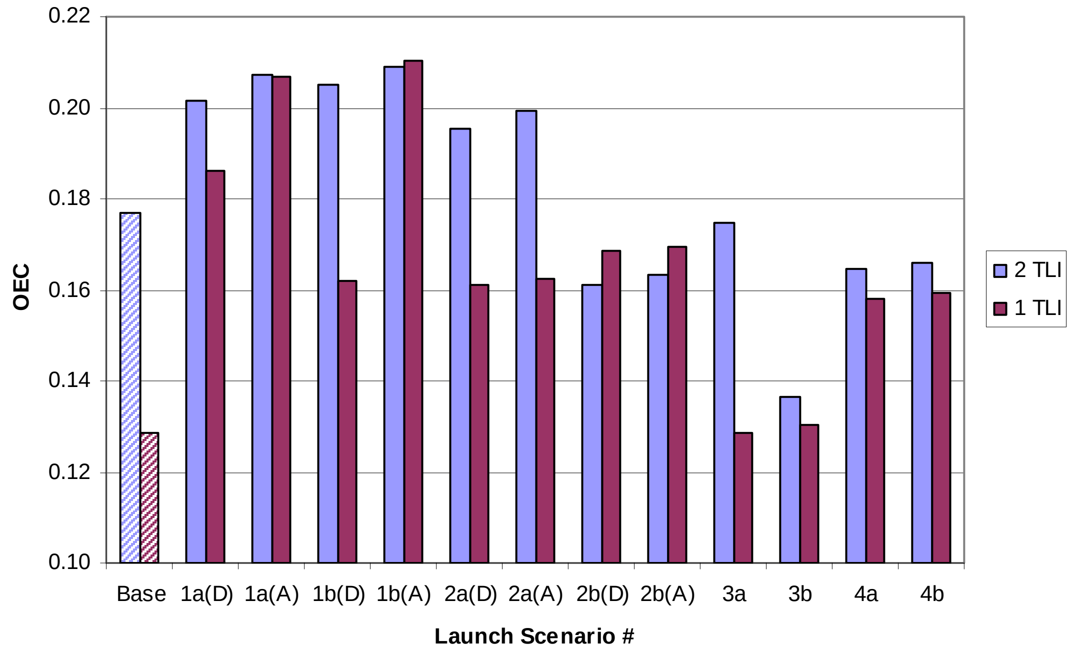

Figure 37. Overall Evaluation Criterion (Cost, Reliability, and LEO Time) for TLI Trade Study (Expendable Architecture).

When two TLIs are required per mission, the TLI manufacturing capacity should have a larger effect on the mission capture percentages. As explained earlier, since all launch options require the same in-space elements, and all elements must be available before any launches can take place, changing the manufacturing capacities should have the same effect on all the launch options. To be consistent with the results using one TLI, the baseline launch option is examined. Table 25 lists the queue wait times for each of the manufacturing facilities, the capture percentage and the launch delay time for several different manufacturing capacities. As expected, manufacturing capacities of one are not sufficient to reach a capture percentage of 100%. Additionally, the average wait times for the Lander, CEV, and TEI are the same as for the 1-TLI case. With a manufacturing capacity of one, however, the TLI manufacturing wait time increases to over 1100 days. Therefore, the capacity of the TLI manufacturing facility must also be increased to two in order to launch all missions using two TLIs.

Table 25. Mission Capture Percentage and Launch Delay Time for Varying Manufacturing Capacities (Expendable Architecture, Baseline Launch Option).

CEV Capacity	1	2
Lander Capacity	1	2
TLI Capacity	1	2
TEI Capacity	1	2
CEV Wait Time	1056.28	17.84
Lander Wait Time	1054.02	17.28
TLI Wait Time	1121.30	14.85
TEI Wait Time	580.51	0.00
Mission Capture %	0.512	1.00
Launch Delay	771.21	4.93

4.3.3 TLI Trade Study Summary

The purpose of using two smaller TLIs would be for better packaging on certain launch vehicles, such that they can be launched full instead of having to launch numerous propellant launches. Only two of the launch options saw any benefit (Evolved EELV with 70 mt Colossus), and the improvement was very small. Although the cost decreased significantly for these two options, the loss of mission probability increased. Even if cost were the only metric of interest, many of the other launch options still have a lower life-cycle cost if using one TLI. The penalty incurred in reliability, however, makes two TLIs unattractive as a valid architecture option. Additionally, the TLI manufacturing capacity would have to be increased, although this should not warrant any significant penalty, since it is still a reasonable number. Therefore, based on the assumptions in this study, the best solution for the expendable architecture remains the baseline presented in Section 3.2. There are benefits, however, to using two smaller TLIs that are not addressed using LASSO. Primarily, on-orbit propellant transfer is a technology that must be further developed before it could actually be used for a mission. The time and costs associated with this technology maturation program are not accounted for. If using two TLIs can enable them to be launched fully fueled, which is the case for some of the launch options, propellant transfer would be eliminated as a show-stopping technology for the lunar missions.

4.4 Annual Funding

In the previous launch vehicle trade studies, cost was evaluated based on undiscounted life-cycle cost and cost per mission. Another important factor, however, is the maximum peak annual funding required. Generally a program is allocated a given budget on a yearly basis, not a lump sum that can be spent each year as needed. Therefore, while a chosen architecture may fit within a ten or twenty year budget, each year's spending must also fit within that given year's budget. The two architectures will be thus compared across all of the launch options in terms of yearly spending. It is assumed that DDTE costs are spread evenly across the first five years of the program. Production will then start the following year, with costs assigned to the year in which production of a particular element starts. Over the ten years where missions are flown, costs are broken down into production, launch, and operations costs. Finally, there are no launches in the last year of the program, but fixed operations costs are still incurred. As an example, Figure 38 plots the annual funding required for the expendable architecture using the baseline launch configuration. The costs are broken down into DDTE, launch, production, and operations. For the baseline, there is a significant amount of DDTE costs up-front, and then the annual costs decline once lunar missions begin. Production costs decrease each year as the learning curve decreases the cost of producing each element. All of the costs shown are in 2005 dollars.

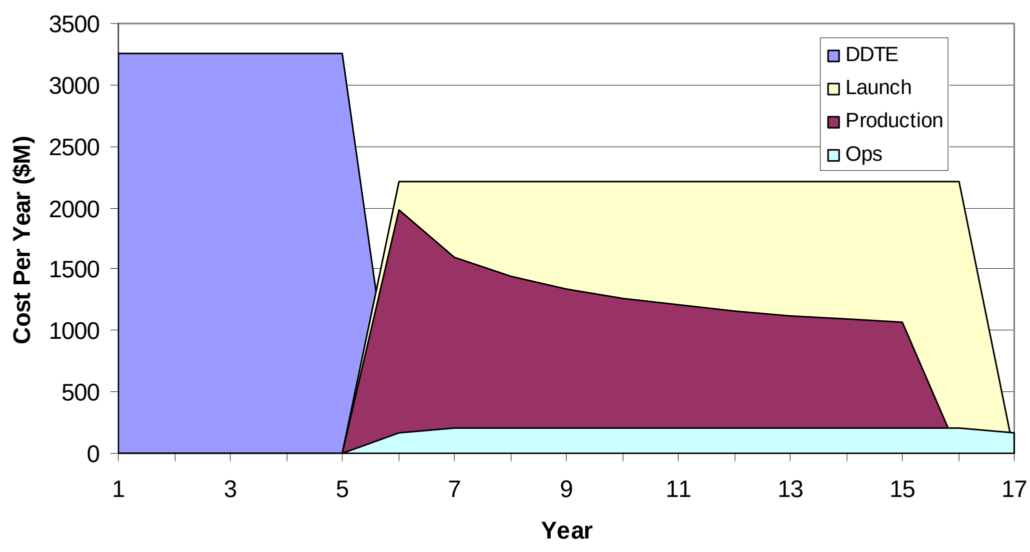

Figure 38. Annual Costs for Expendable Architecture using Baseline Launch Configuration.

Depending on the choice of architecture and architecture elements, the cost distribution will be different than for the baseline case shown in Figure 38. For example, the baseline expendable architecture can be compared to the Existing EELV launch option #1a(D), as shown in Figure 39. Both of these launch options did well overall in terms of cost ($53B and $55B for the baseline and #1a(D), respectively). How their costs are spread out over the entire program, however, could have an impact on which is chosen depending on budget constraints. The baseline case has a higher DDTE (over $1B per year), but lower annual costs once lunar missions start. This results from the baseline launch vehicles having higher DDTE costs, but lower launch costs overall. Therefore, if more money were available up-front, the baseline case would make more sense financially, particularly if the program duration were increased. The lower annual costs will pay off even more the more missions that are flown. If the program duration were reduced, however, the baseline would become more expensive, since more money was invested up-front. If less money were available up-front for DDTE, then the Existing EELV option would have to be chosen. This would then result in higher costs down the road.

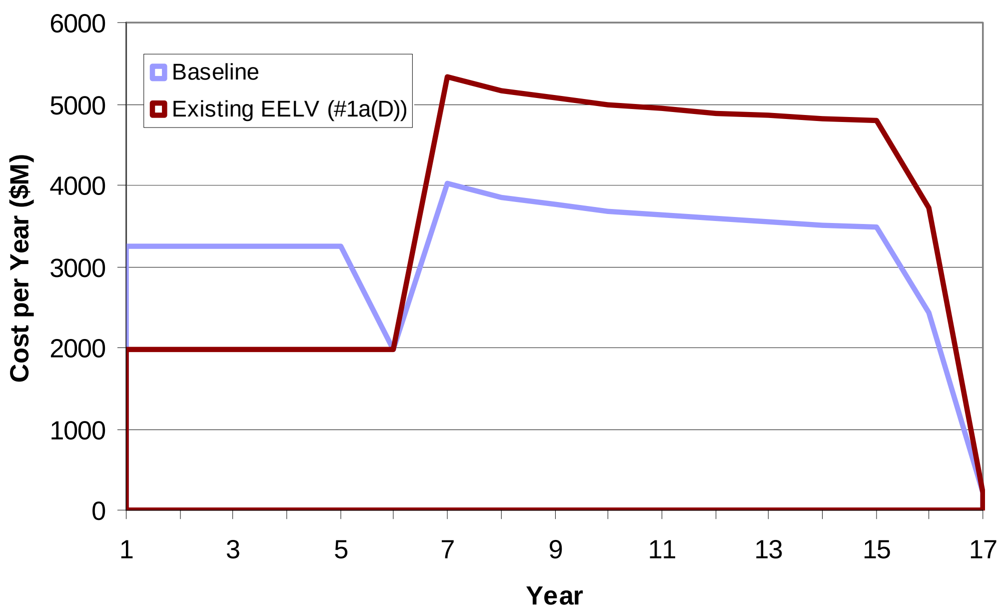

Figure 39. Annual Costs for Expendable Architecture Comparing Baseline Launch Option and #1a(D).

The same charts can be created for the reusable architecture. Figure 40 shows the annual costs broken down by DDTE, depot resupply launches, lunar mission launches, production costs, and operations costs. For the reusable architecture, manufacturing takes place for two years before the first lunar mission, as opposed to one year for the expendable architecture. This is due to a longer manufacturing time for the reusable lander as well as the need to build three landers up front. The costs between the two architectures are actually fairly similar on a yearly basis, with the exception of the added cost of the propellant depot resupply launches. Although more TLIs need to be built per year, no landers are built on a yearly basis, which evens out the production costs. Therefore, the added cost in the reusable architecture can be accounted for in the cost of propellant launches to the lunar depot.

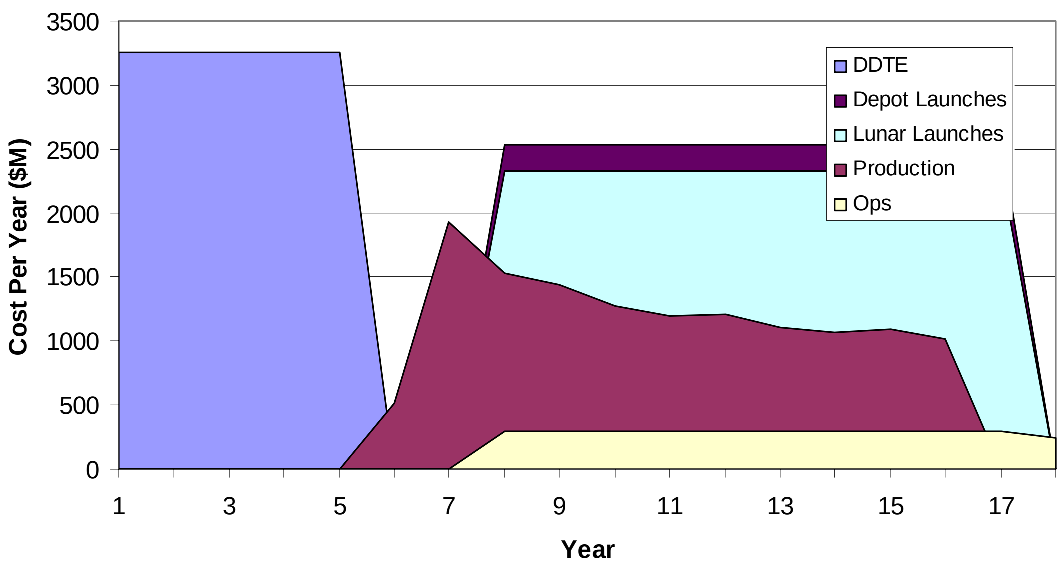

Figure 40. Annual Costs for Reusable Architecture using Baseline Improved Launch Configuration.

The baseline improved launch option can also be compared to the Existing EELV launch option #1a(D) for the reusable architecture, as shown in Figure 41. The same trend is seen as for the expendable architecture. While the baseline improved launch configuration requires a higher up-front cost, its annual costs once the lunar missions begin are significantly lower.

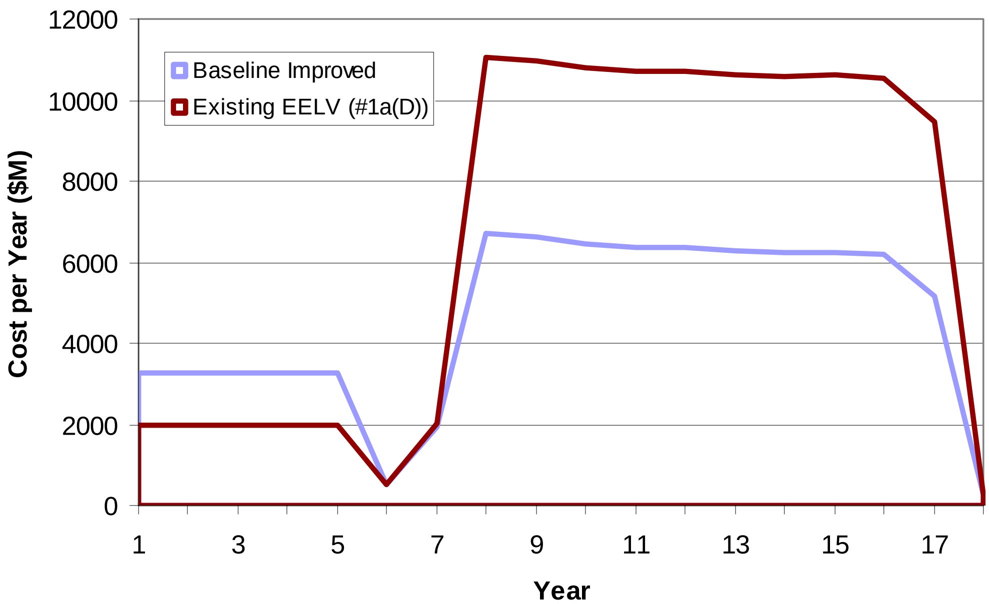

Figure 41. Annual Costs for Reusable Architecture Comparing Baseline Improved Launch Option and #1a(D).

Finally, the peak annual funding required must be considered for each launch option for both architectures. While life-cycle cost and average cost per mission are important metrics, it is also important to consider the maximum annual cost in order to stay within a given budget. Obviously, lower is better. Figure 42 plots the peak annual cost for each launch option for both architectures. The reusable architecture, which has higher life-cycle costs, also has significantly higher peak annual costs. For the expendable and reusable architectures, the best launch options chosen based on the previous studies are the baseline and baseline improved, respectively. These also have the lowest peak annual cost. Therefore, the funding profile further confirms the selection of these launch configurations as the best choices for each architecture.

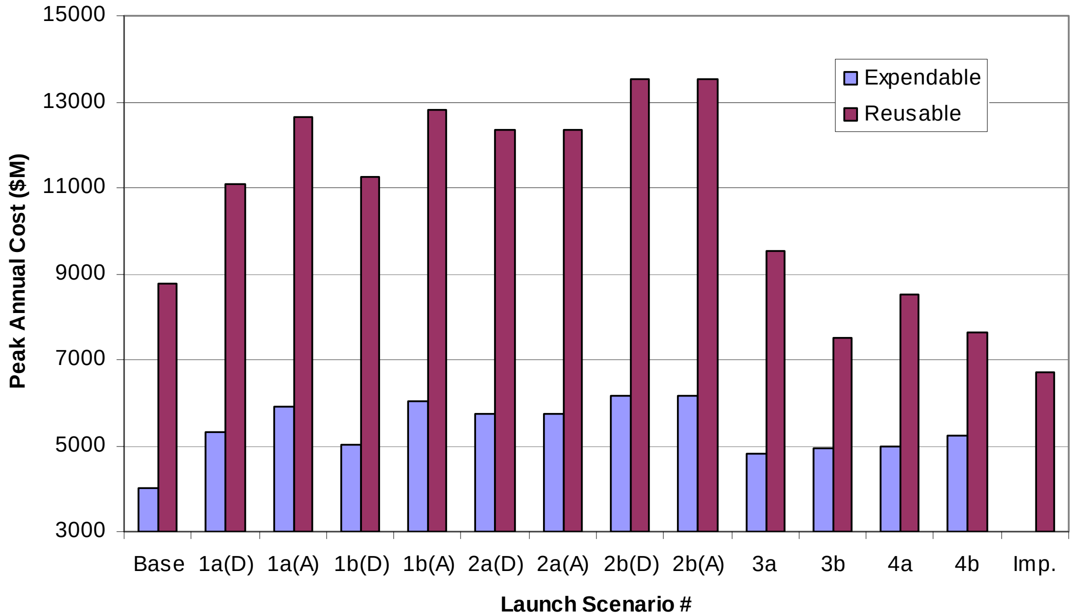

Figure 42. Peak Annual Cost for Expendable and Reusable Architectures.

4.5 Expendable vs. Reusable Architecture Flight Rate Trade Study

When comparing a reusable architecture against an expendable architecture, it is expected that at some flight rate, there will be a crossover between which architecture is more affordable. In general, reusable architectures should require more infrastructure investment up-front, but as the number of missions flown increases, money should be saved since the variable costs per mission are much less. Expendable architectures, however, employ more of a "pay as you go" approach. There is less up-front investment, so you are basically paying for each mission as it is flown. Expendable architectures therefore tend to be favorable for low flight rates, while reusable architectures are favored for high-flight rates.

This theory was thus tested for the reusable and expendable architecture modeled for this study, by varying the flight rate and plotting the costs at each flight rate. Figure 43 and Figure 44 plot the life-cycle cost and the cost per mission for each architecture, respectively, assuming unlimited ground infrastructure. The launch vehicles used are the best options resulting from the launch vehicle study, as explained earlier. For the expendable architecture, the crew launch on a Delta IV Heavy and the cargo launches on a Centurion C2 (baseline option). For the reusable architecture, the crew launch on a Delta IV Heavy and the cargo and propellant launch on a Centurion C3 (baseline improved option).

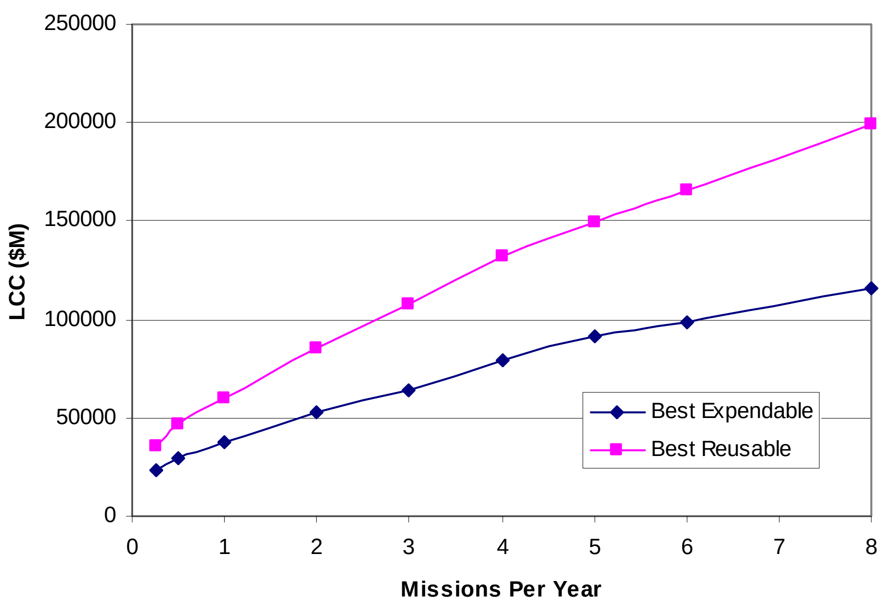

Figure 43. Life-Cycle Cost as a Function of Annual Flight Rate.

As can be seen, there is no flight rate where a crossover in cost will occur. In fact, the two curves are diverging. The intended advantage of the reusable architecture is a reduction in the number of launches required per lunar mission, since a lander does not need to be launched for each mission. Additionally, fewer landers need to be produced, since a single reusable lander is used for multiple missions. The reusable lander, however, requires the addition of a propellant depot in lunar orbit that must be periodically refilled. Because the lander requires 35 mt of propellant per lunar mission, on average this much propellant must be launched to lunar orbit for every manned mission. Additionally, a TLI stage must also be launched to ferry the PRM out to lunar orbit. For all launch vehicles except the Centurion C3, more than one launch is required each time a depot resupply mission is launched. Therefore, anything gained by pre-deploying a reusable lander is counteracted by the addition of propellant launches to the depot and the production cost of more TLI stages.

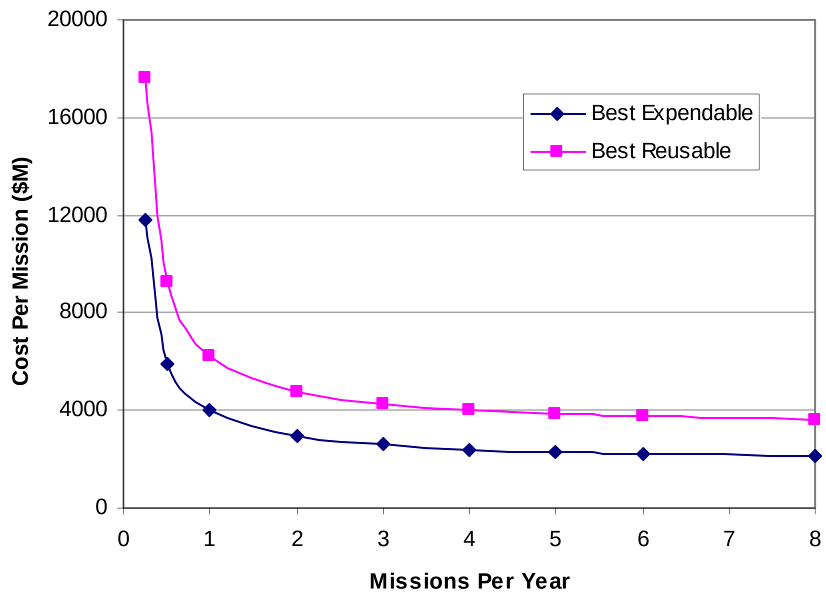

Figure 44. Cost per Mission as a Function of Annual Flight Rate.

The same result can be seen from the cost per mission plot. Because there are fixed costs associated with each architecture (DDTE and fixed operations costs), it is advantageous to fly over a certain number of missions per year. As the flight rate increases, these fixed costs are spread more thinly between the various missions, causing the cost per mission to approach an asymptote. For these two architectures, more than two missions per year appears to be the best flight rate.

The two architectures can also be compared in terms of capture percentage for a given ground infrastructure as the flight rate is varied. The values used for the ground infrastructure are listed in Table 26. Figure 45 plots the mission capture percentage as a function of number of missions per year. As can be seen, for the ground infrastructure available, the expendable architecture achieves a higher capture percentage across all flight rates, since it requires fewer launches and fewer TLI stages (even though it requires more landers, but this does not produce a significant bottleneck).

Table 26. Ground Infrastructure Values for Mission Capture Rate Trade Study

	Capacity
CEV Manufacturing	2
Lander Manufacturing	2
TLI Manufacturing	2
TEI Manufacturing	2
Crew Integration	1
Cargo Integration	1
Prop. Integration	1
Crew Launch Pads	1
Cargo Launch Pads	1
Prop. Launch Pads	1

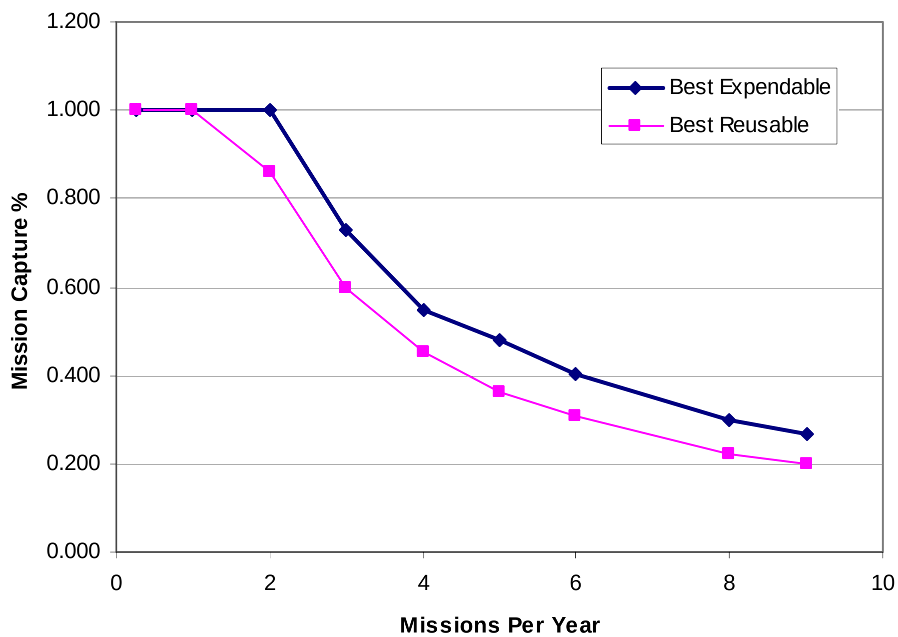

Figure 45. Mission Capture Rate as a Function of Annual Flight Rate.

For a given ground infrastructure, there is a maximum number of missions per year that can be achieved for each architecture. Therefore, as the desired number of missions per year increases, the actual number of missions flown will remain constant once the maximum is reached. This result is presented in Figure 46, which plots the actual number of mission achieved per year by each architecture. As expected from the above results, the expendable architecture is able to fly more missions. It plateaus around 2.4 missions per year, while the reusable architecture has a maximum of 1.8 missions per year for the given ground infrastructure.

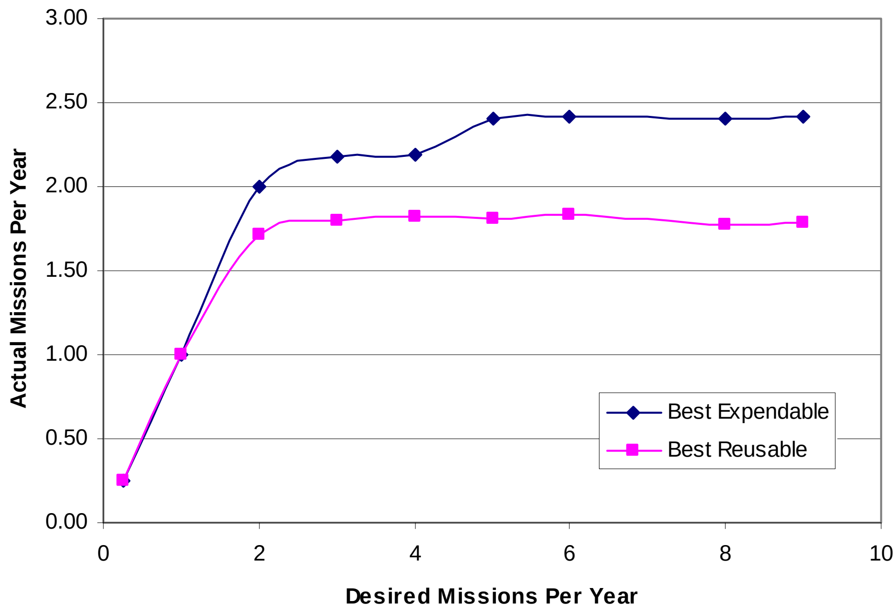

Figure 46. Actual Number of Missions Per Year as a Function of Desired Annual Flight Rate.

5.0 Conclusions

The purpose of this study was to evaluate, using LASSO, two representative lunar exploration architectures in terms of cost, reliability, and mission throughput. The primary trade study was to examine various combinations of launch vehicles for each architecture, based on current leading candidates for the planned lunar missions. The best option for the expendable architecture, based on an overall evaluation criterion including cost, reliability, and scheduling, was to use a Delta IV Heavy for crew launches and a 100 mt Centurion C2 for cargo and propellant launches. The best option for the reusable architecture was to use a Delta IV Heavy for crew and a 140 mt Centurion C3 for cargo and propellant. In general, the trade study showed that heavy-lift launch vehicles were favored, particularly for the cargo and propellant missions, since they required fewer launches per mission. The Shuttle-derived vehicles also did fairly well in terms of cost (not so for reliability), but these costs were based on pre-CAIB estimates. Therefore, if the actually launch costs were to increase, the Shuttle-derived vehicles would become significantly less attractive. The Existing Evolved Expendable Launch Vehicles also did very well in terms of cost, but because of the increased number of launches required, their loss of mission probability and scheduling metrics suffered. Unless the mass of the mission elements could be reduced, the only way to improve the scheduling metrics for the Existing EELVs would be to increase the number of launch pads available or to decrease the turn-around time of the pads. The loss of mission probability for the EELVs could be improved, however, by changing some of the mission assumptions. Instead of canceling a mission when a cargo launch fails, that element could simply be re-launched if an extra were always kept in inventory for such a situation. As a result, launch configurations requiring more cargo launches per lunar mission would not be penalized in terms of mission reliability simply because more launches are required. This change would make the Existing EELVs more competitive, albeit still worse than the HLVs for a given ground infrastructure due to their throughput capabilities. Therefore, it appears that investing in a heavy-lift launch vehicle would be beneficial. This of course assumes that the money is available up-front for the development of such a launch vehicle since their DDTE costs are much higher. If this were the case, it would save significant money down the road in terms of the lunar mission transportation costs.

Regardless of the launch option chosen, however, the expendable architecture appears to be favorable. Although the loss of mission probability decreases slightly in some cases for the reusable architecture, the life-cycle cost and cost per mission increase significantly. The transportation-related life-cycle and per mission costs range from \$53B to \$72B and from \$2.9B to \$3.9B for the expendable architecture. For the reusable architecture, they range from \$86B to \$137B and from \$4.7B to \$7.4B. As a point of comparison, Apollo cost \$19.4B[15] over twelve years, which included seven attempted lunar landings (six of which were successful), while the baseline mission includes twenty attempted lunar landings over a 17-18 year program. Furthermore, the Apollo program cost not only includes the development, production, and operations costs associated with the transportation elements, but also includes costs associated with other program aspects that are not modeled in LASSO, such as science payloads and advanced technology development studies. Translated into 2005 dollars, Apollo would cost approximately $68B. Even the cheapest option for either launch architecture had a life-cycle cost of $53.3B for just the transportation elements. When the remaining programs are included, this cheapest option will begin to exceed the Apollo program budget, albeit for more lunar missions. Even though more missions could be flown for around the same cost as the Apollo program, the budget is more restricted today than it was in the 1960's and 1970's. At its peak, Apollo spent $2.9B ($13.3B is 2005 dollars) in 1967, which comprised 70% of NASA's total budget.[15] The latest NASA budget request (FY 2006) allocates $3.16B per year to the Exploration Systems directorate, which is responsible for the human lunar exploration. This is only 19% of NASA's total budget, and yearly increases through FY 2010 are only planned to account for inflation.[16] Even considering only the transportation-related costs modeled in LASSO, the cheapest architecture option does not fit within NASA's current budget request, when the annual program costs are considered as shown in Section 4.4. Reducing program costs, therefore, is of utmost importance in selecting a lunar transportation architecture.

In addition to transportation-related costs, the expendable architecture also has the added advantage of requiring less infrastructure investment for a given flight rate and launch option. Manufacturing capacities are not an important factor, since for the capacities required (less than five in all cases examined), it is assumed that the production facilities would be expanded to meet the required demand. Building more launch pads and launch vehicle integration facilities, however, would be prohibitive to a successful human lunar program. Currently, the Shuttle has two available launch pads at KSC, and the Atlas V and Delta IV Heavies each have one at Cape Canaveral.[17] Expanding beyond this current infrastructure would be prohibitive in terms of cost and available real estate. If the flight rate were to be increased beyond the baseline of two missions per year, the reusable would be the first of the two architectures to require increased launch infrastructure beyond what is currently available.

The reliability of both architectures was also examined. In general, the loss of crew probability ranged anywhere from 0.058 to 0.113 per mission. This probability represents the entire mission, so it encapsulates the reliability of the launch vehicles, in-space propulsive stages, the lunar lander, and the CEV reentry. The low end of this range may be marginally acceptable for a limited human lunar program, but any LOC number approaching or exceeding 10% might prevent the program from proceeding. Statistically, a 5% LOC will result in one loss of crew event every twenty missions. Therefore, for the baseline, one loss of crew event would occur in the ten-year program duration. The LOM numbers, of course, are higher, and range from 0.128 to 0.316. These numbers are also higher than desired, since at best, more than one in ten missions will not reach completion. If a sustained human presence is desired, this number is unacceptably high. Without improving the reliability of the individual elements, however, these numbers cannot be improved.

From this study and the assumptions it contains, it appears that the reusable architecture should not be chosen over the expendable architecture, regardless of the launch configuration. This is not necessarily the case for general reusable vs. expendable architectures. It is still expected that a reusable architecture could be designed that would perform better in terms of cost for higher flight rates, even if it still requires more infrastructure investment up-front. Several improvements could be made to the current architecture that should make it more competitive, as explained in Section 4.2.2.4. In particular, there could be more flexibility in the choice of launch vehicle for the depot resupply launches or in the packaging of all the elements for those missions. Although this study did not do an extensive launch vehicle optimization, this could improve the attractiveness of the reusable architecture. For the specific architectures studied, however, the expendable architecture is superior, although reusable architectures should not be definitively eliminated as a viable option for sustained human lunar exploration. Further study into other reusable architectures would be required to ascertain their overall effectiveness.

Based on the LASSO results, several key conclusions can be drawn in regards to the architecture choices that must be made for human lunar exploration, as outlined below. It is important to remember that these conclusions are based on the architectures modeled and the mission and vehicle assumptions contained in this study.

1. An expendable architecture is favored over a reusable architecture, based on cost, reliability, and scheduling figures of merit.
2. Man-rated versions of Existing EELVs are most cost-effective for crewed launches.
3. Expendable launch vehicles should be used for dedicated propellant launches (an RLV tanker was shown to not be cost effective).
4. Heavy-lift launch vehicles are preferred for cargo and propellant launches, because of the fewer launches required per lunar mission.

These considerations are critical to establishing a cost-effective and sustainable human lunar exploration program. If only performance metrics are used during the conceptual stage of the design process, a program can run into budget and schedule problems down the road, when they will be more difficult and expensive to correct. Therefore the capability to evaluate space exploration architectures based on cost, reliability, and scheduling figures of merit will be essential to successfully implementing the President's Vision for Space Exploration through the next several decades.

References

1. *The Vision for Space Exploration*, National Aeronautics and Space Administration, Washington, D.C., Feb. 2004.
2. Zapata, Edgar, *GEM-FLO*, National Aeronautics and Space Administration, 2003. [http://science.ksc.nasa.gov/shuttle/nexgen/GEM-FLO_main.htm. Accessed 4/7/05.]
3. Reeves Jr., John Daniel, *Discrete Event Simulation of Reusable Launch Vehicle Ground Operations (RLVSim)*, Space Systems Design Lab, Georgia Institute of Technology, Atlanta, GA, July 2004.
4. Fishman, George S., *Discrete-Event Simulation*, Springer-Verlag, New York, NY, 2001.
5. Kelton, David W., Sadowski, Randall P., Sadowski, Deborah A., *Simulation with Arena, 2nd Edition*, McGraw Hill, New York, NY, 2002.
6. *Arena Standard*, User's Guide, Rockwell Software Inc., 2001.
7. Devore, Jay L., *Probability and Statistics, 5th Edition*, Duxbury Thomas Learning, Pacific Grove, CA, 2000.
8. Ross, Sheldon M., *Introduction to Probability Models, 8th Edition*, Academic Press, 2003.
9. Young, D., Olds, J.R., Hutchinson, V., Krevor, Z., Pimentel, J., Reeves, J.D., Sakai, T., Young, J., *Centurion: A Heavy-Lift Launch Vehicle Family for Cis-Lunar Exploration*, AIAA 2004-3735, 40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference And Exhibit, Fort Lauderdale, Florida, July 11 - 14, 2004.
10. Young, D., Olds, J.R., Hutchinson, V., Krevor, Z., Young, J., *Artemis: A Reusable Excursion Vehicle Concept for Lunar Exploration*, AIAA 2005-4010, 40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference And Exhibit, Fort Lauderdale, Florida, July 11 - 14, 2004.
11. Hutchinson, V., Olds, J.R., *Tempest: A Lunar Transfer Crew Exploration Vehicle*, AIAA 2005-4190, 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Tuscon, Arizona, July 10-13, 2005.
12. Hong, P.E., Kent, P.D., Olson, D.W., Vallado, C.A., *Interplanetary Program to Optimize Simulated Trajectories (IPOST)*, User's Guide, Martin Marietta Astronautics, Denver, CO, Oct. 1992.
13. Powell, R.W., Striepe, S.A., Desai, P.N., Bruan, R.D., *Program to Optimize Simulated Trajectories (POST)*, Utilization Manual, NASA Langley Research Center, Hampton, VA, Oct. 1997.
14. Kokan, T., Olds, J.R., *Vega, URETI RLV*, Internal Conceptual Design Study, Space Systems Design Lab, Georgia Institute of Technology, Atlanta, GA, Dec. 2002.
15. *Apollo Program Budget Appropriations*. [http://history.nasa.gov/SP-4029/Apollo_18-16_Apollo_Program_Budget_Appropriations.htm. Accessed 4/7/05.]
16. *Fiscal Year 2006: Budget Estimates*, D.C., NP-2005-02-392, National Aeronautics and Space Administration, Washington, 2005.

[http://www.nasa.gov/pdf/107486main_FY06_high.pdf Accessed April 2005.]

17. Wade, M., *Cape Canaveral*, Encyclopedia Astronautica, 2004. [http://www.astronautix.com/sites/capveral.htm. Accessed April 2005.]
18. Haddox, E., *Launch Services Program's Vehicle Performance Website,* National Aeronautics and Space Administration. [http://elvperf.ksc.nasa.gov/elvMap/index.html Accesssed March 2005.]
19. *Design Reliability Comparison for SpaceX Falcon Vehicles*, Futron Corporation, Bethesda, MD, Nov. 2004.
20. Isakowitz, S.J., Hopkins, J.B., Hopkins, Jr., J.P., *International Reference Guide to Space Launch Systems, 4th Edition*, AIAA, Reston, VA, 2004.
21. Cyr, K., *Gross Domestic Product Deflator Inflation Calculator*, National Aeronautics and Space Administration, Jan. 2005. [http://www1.jsc.nasa.gov/bu2/inflateGDP.html Accessed March 2005.]
22. Wade, M., *Orbital Launch Vehicle Index*, Encyclopedia Astronautica, 2004. [http://www.astronautix.com. Accessed March 2005.]
23. Koelle, D.E., *Handbook of Cost Engineering for Space Transportation Systems with Transcost 7.0*, TransCostSystems, Germany, Nov. 2000.
24. Jenkins, D. R., *Space Shuttle: The History of the National Space Transportation System*, World Print Ltd, Hong Kong, 2001.

Appendix A: Arena VBA Code

(Note: Only the code for the Expendable.doe architecture is included, since the Reusable.doe code is almost identical.)

A.1 Expendable.doe – User Form

```
'This VBA code is used to set the variable values in Arena to those typed into the User Form.

Private Sub CommandButton1_Click()

Dim oSIMAN As Arena.SIMAN
Dim oModel As Arena.Model

Set oSIMAN = ThisDocument.Model.SIMAN
Set oModel = ThisDocument.Model

'Replication Parameters
oModel.Modules(oModel.Modules.Find(smFindTag,"Replications")).Data("initialvalue") = PODInputs.Replications
If (PODInputs.BatchRun.value = True) Then
    oModel.BatchMode = True
Else
    oModel.BatchMode = False
End If

'Mission Parameters
oModel.Modules(oModel.Modules.Find(smFindTag,"MissionsPerYear")).Data("initialvalue")=
PODInputs.MissionPerYear
oModel.Modules(oModel.Modules.Find(smFindTag, "Num_Years")).Data("initial value") = PODInputs.NumYears
oModel.Modules(oModel.Modules.Find(smFindTag, "Lunar_Time")).Data("initial value") = PODInputs.LunarTime

'Cost Distribution
oModel.Modules(oModel.Modules.Find(smFindTag, "lowerCost")).Data("initial value") = PODInputs.LowerCost
oModel.Modules(oModel.Modules.Find(smFindTag, "upperCost")).Data("initial value") = PODInputs.UpperCost

'Vehicle Indices
oModel.Modules(oModel.Modules.Find(smFindTag, "indexCEV")).Data("initial value") = PODInputs.indexCEV
oModel.Modules(oModel.Modules.Find(smFindTag, "extraCEVs")).Data("initial value") = PODInputs.ExtraCEVs
oModel.Modules(oModel.Modules.Find(smFindTag, "indexTLI")).Data("initial value") = PODInputs.indexTLI
oModel.Modules(oModel.Modules.Find(smFindTag,"indexLander")).Data("initialvalue") = PODInputs.indexLander
oModel.Modules(oModel.Modules.Find(smFindTag, "indexTEI")).Data("initial value") = PODInputs.indexTEI
oModel.Modules(oModel.Modules.Find(smFindTag,"indexLVcrew")).Data("initialvalue")= PODInputs.indexLVcrew
oModel.Modules(oModel.Modules.Find(smFindTag, "indexLV")).Data("initial value") = PODInputs.indexLVcargo
oModel.Modules(oModel.Modules.Find(smFindTag,"indexLVprop")).Data("initialvalue")= PODInputs.indexLVprop

'Ground Operations
oModel.Modules(oModel.Modules.Find(smFindTag,"Inventory_Time")).Data("initialvalue")=
PODInputs.InventoryTime
oModel.Modules(oModel.Modules.Find(smFindTag,"Integration_Time")).Data("initialvalue")=
PODInputs.IntegrationTime
oModel.Modules(oModel.Modules.Find(smFindTag, "Pad_Time")).Data("initial value") = PODInputs.PadTime
oModel.Modules(oModel.Modules.Find(smFindTag, "padTAT")).Data("initial value") = PODInputs.PadTAT
oModel.Modules(oModel.Modules.Find(smFindTag,"Investigation_Time")).Data("initialvalue")=
PODInputs.InvestigationTime

'Ground Infrastructure
oModel.Modules(oModel.Modules.Find(smFindTag,"CEVManufacture")).Data("capacity") = PODInputs.buildCEV
```

```
oModel.Modules(oModel.Modules.Find(smFindTag, "TLI Manufacture")).Data("capacity") = PODInputs.buildTLI
oModel.Modules(oModel.Modules.Find(smFindTag,"LanderManufacture")).Data("capacity")= PODInputs.buildLander
oModel.Modules(oModel.Modules.Find(smFindTag, "TEI Manufacture")).Data("capacity") = PODInputs.buildTEI
oModel.Modules(oModel.Modules.Find(smFindTag, "CEV Refurbish")).Data("capacity") = PODInputs.refurbCEV
oModel.Modules(oModel.Modules.Find(smFindTag,"IntegrationCrew")).Data("capacity")=
PODInputs.IntegrationCrew
oModel.Modules(oModel.Modules.Find(smFindTag,"Integration")).Data("capacity") = PODInputs.IntegrationCargo
oModel.Modules(oModel.Modules.Find(smFindTag,"IntegrationPropellant")).Data("capacity")=
PODInputs.IntegrationProp
oModel.Modules(oModel.Modules.Find(smFindTag, "Launch Pad Crew")).Data("capacity") = PODInputs.LPCrew
oModel.Modules(oModel.Modules.Find(smFindTag, "Launch Pad")).Data("capacity") = PODInputs.LPCargo
oModel.Modules(oModel.Modules.Find(smFindTag, "Launch Pad Prop")).Data("capacity") = PODInputs.LPProp

PODInputs.Hide

End Sub
```

A.2 Expendable.doe – VBA Code

```
Option Explicit

Dim currentRep, Num_Years, Missions() As Integer
Dim Year_Demand(), Num_Missions, MissionsPerYear As Double
Dim oExcel As Excel.Application, oWorkbook As Excel.Workbook, oWorksheet As Excel.Worksheet


Private Sub ModelLogic_RunBegin()

'Initially set large arrays sizes
Model.Modules(Model.Modules.Find(smFindTag, "Launch_Dates")).Data("Rows") = 100
Model.Modules(Model.Modules.Find(smFindTag, "Launch_Dates_Actual")).Data("Rows") = 100
Model.Modules(Model.Modules.Find(smFindTag, "Begin_Launch")).Data("Rows") = 100
Model.Modules(Model.Modules.Find(smFindTag, "buildTLI")).Data("Rows") = 100
Model.Modules(Model.Modules.Find(smFindTag, "buildTEI")).Data("Rows") = 100
Model.Modules(Model.Modules.Find(smFindTag, "buildLander")).Data("Rows") = 100
Model.Modules(Model.Modules.Find(smFindTag, "buildCEV")).Data("Rows") = 100
Model.Modules(Model.Modules.Find(smFindTag, "cancelledMission")).Data("Rows") = 100

'Set Values in User Form to Values from Previous Arena Run
PODInputs.Replications = Model.Modules(Model.Modules.Find(smFindTag, "Replications")).Data("Initial Value")
PODInputs.BatchRun.value = True
PODInputs.MissionPerYear = Model.Modules(Model.Modules.Find(smFindTag, "MissionsPerYear")).Data("Initial Value")
PODInputs.NumYears = Model.Modules(Model.Modules.Find(smFindTag, "Num_Years")).Data("Initial Value")
PODInputs.LunarTime = Model.Modules(Model.Modules.Find(smFindTag, "Lunar_Time")).Data("Initial Value")
PODInputs.LowerCost = Model.Modules(Model.Modules.Find(smFindTag, "lowerCost")).Data("Initial Value")
PODInputs.UpperCost = Model.Modules(Model.Modules.Find(smFindTag, "upperCost")).Data("Initial Value")
PODInputs.indexCEV = Model.Modules(Model.Modules.Find(smFindTag, "indexCEV")).Data("Initial Value")
PODInputs.ExtraCEVs = Model.Modules(Model.Modules.Find(smFindTag, "extraCEVs")).Data("Initial Value")
PODInputs.indexTLI = Model.Modules(Model.Modules.Find(smFindTag, "indexTLI")).Data("Initial Value")
PODInputs.indexLander = Model.Modules(Model.Modules.Find(smFindTag, "indexLander")).Data("Initial Value")
PODInputs.indexTEI = Model.Modules(Model.Modules.Find(smFindTag, "indexTEI")).Data("Initial Value")
PODInputs.indexLVcrew = Model.Modules(Model.Modules.Find(smFindTag, "indexLVcrew")).Data("Initial Value")
PODInputs.indexLVcargo = Model.Modules(Model.Modules.Find(smFindTag, "indexLV")).Data("Initial Value")
PODInputs.indexLVprop = Model.Modules(Model.Modules.Find(smFindTag, "indexLVprop")).Data("Initial Value")
PODInputs.InventoryTime = Model.Modules(Model.Modules.Find(smFindTag, "Inventory_Time")).Data("Initial Value")
PODInputs.IntegrationTime = Model.Modules(Model.Modules.Find(smFindTag, "Integration_Time")).Data("Initial Value")
PODInputs.PadTime = Model.Modules(Model.Modules.Find(smFindTag, "Pad_Time")).Data("Initial Value")
PODInputs.PadTAT = Model.Modules(Model.Modules.Find(smFindTag, "padTAT")).Data("Initial Value")
PODInputs.InvestigationTime = Model.Modules(Model.Modules.Find(smFindTag, "Investigation_Time")).Data("Initial Value")
PODInputs.buildCEV = Model.Modules(Model.Modules.Find(smFindTag, "CEV Manufacture")).Data("Capacity")
PODInputs.buildTLI = Model.Modules(Model.Modules.Find(smFindTag, "TLI Manufacture")).Data("Capacity")
PODInputs.buildLander = Model.Modules(Model.Modules.Find(smFindTag, "Lander Manufacture")).Data("Capacity")
PODInputs.buildTEI = Model.Modules(Model.Modules.Find(smFindTag, "TEI Manufacture")).Data("Capacity")
PODInputs.refurbCEV = Model.Modules(Model.Modules.Find(smFindTag, "CEV Refurbish")).Data("Capacity")
PODInputs.IntegrationCrew = Model.Modules(Model.Modules.Find(smFindTag, "Integration Crew")).Data("Capacity")
PODInputs.IntegrationCargo = Model.Modules(Model.Modules.Find(smFindTag, "Integration")).Data("Capacity")
```

```
PODInputs.IntegrationProp = Model.Modules(Model.Modules.Find(smFindTag, "Integration Propellant")).Data("Capacity")
PODInputs.LPCrew = Model.Modules(Model.Modules.Find(smFindTag, "Launch Pad Crew")).Data("Capacity")
PODInputs.LPCargo = Model.Modules(Model.Modules.Find(smFindTag, "Launch Pad")).Data("Capacity")
PODInputs.LPProp = Model.Modules(Model.Modules.Find(smFindTag, "Launch Pad Prop")).Data("Capacity")

PODInputs.Show

End Sub

Private Sub ModelLogic_RunBeginSimulation()

currentRep = 1

End Sub

Private Sub ModelLogic_RunBeginReplication()

Dim oSIMAN As Arena.SIMAN
Set oSIMAN = ThisDocument.Model.SIMAN

Set oExcel = CreateObject("Excel.Application")
oExcel.Visible = False
Set oWorkbook = oExcel.Workbooks.Open(ThisDocument.Model.Path & "Database8900.xls")
Set oWorksheet = oWorkbook.ActiveSheet

With oWorksheet
    .Cells(5, 2) = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("indexLVcrew"))
    .Cells(5, 3) = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("indexLV"))
    .Cells(5, 4) = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("indexLVprop"))
    .Cells(5, 5) = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("indexTLI"))
    .Cells(5, 6) = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("indexTEI"))
    .Cells(5, 7) = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("indexLander"))
    .Cells(5, 8) = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("indexCEV"))
    .Cells(5, 9) = 1
End With

With oWorksheet

    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("payloadLV", 1)) = .Cells(12, 4)
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("ddteLV", 1)) = .Cells(12, 5)
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("priceLV", 1)) = .Cells(12, 6)
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("reliabilityLV", 1)) = .Cells(12, 7)
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("payloadLV", 2)) = .Cells(13, 4)
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("ddteLV", 2)) = .Cells(13, 5)
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("priceLV", 2)) = .Cells(13, 6)
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("reliabilityLV", 2)) = .Cells(13, 7)
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("payloadLV", 3)) = .Cells(14, 4)
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("ddteLV", 3)) = .Cells(14, 5)
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("priceLV", 3)) = .Cells(14, 6)
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("reliabilityLV", 3)) = .Cells(14, 7)

    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("payloadTLI")) = .Cells(19, 4)
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("drymassTLI")) = .Cells(19, 5)
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("massTLI")) = .Cells(19, 6)
```

```
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("propmassTLI")) = .Cells(19, 7)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costTLI", 1)) = .Cells(19, 8)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costTLI", 2)) = .Cells(19, 9)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costTLI", 3)) = .Cells(19, 10)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costTLI", 4)) = .Cells(19, 11)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("propcostTLI")) = .Cells(19, 12)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("reliabilityTLI")) = .Cells(19, 13)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildtimeTLI")) = .Cells(19, 14)

oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("payloadTEI")) = .Cells(20, 4)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("drymassTEI")) = .Cells(20, 5)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("massTEI")) = .Cells(20, 6)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("propmassTEI")) = .Cells(20, 7)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costTEI", 1)) = .Cells(20, 8)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costTEI", 2)) = .Cells(20, 9)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costTEI", 3)) = .Cells(20, 10)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costTEI", 4)) = .Cells(20, 11)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("propcostTEI")) = .Cells(20, 12)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("reliabilityTEI")) = .Cells(20, 13)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildtimeTEI")) = .Cells(20, 14)

oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("payloadLander")) = .Cells(25, 4)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("drymassLander")) = .Cells(25, 5)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("massLander")) = .Cells(25, 6)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costLander", 1)) = .Cells(25, 7)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costLander", 2)) = .Cells(25, 8)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costLander", 3)) = .Cells(25, 9)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costLander", 4)) = .Cells(25, 10)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("propcostLander")) = .Cells(25, 11)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("reliabilityLander")) = .Cells(25, 12)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("lifetimeLander")) = .Cells(25, 13)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildtimeLander")) = .Cells(25, 14)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("surfaceCEV")) = .Cells(25, 15)

oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("massoutCEV")) = .Cells(30, 4)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("massinCEV")) = .Cells(30, 5)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costCEV", 1)) = .Cells(30, 6)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costCEV", 2)) = .Cells(30, 7)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costCEV", 3)) = .Cells(30, 8)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costCEV", 4)) = .Cells(30, 9)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("reliabilityCEV")) = .Cells(30, 10)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("lifetimeCEV")) = .Cells(30, 11)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildtimeCEV")) = .Cells(30, 12)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("refurbtimeCEV")) = .Cells(30, 13)

oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("numTLI")) = .Cells(37, 2)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Num_Launches_Crew")) = .Cells(37, 3)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Num_Launches")) = .Cells(37, 4)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Num_Launches_Prop")) = .Cells(37, 5)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("numLaunchLander")) = .Cells(37, 6)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("numLaunchTLI1")) = .Cells(37, 7)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("numLaunchTLI2")) = .Cells(37, 8)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("numLaunchTEI")) = .Cells(37, 9)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("batchlaunches", 1)) = .Cells(37, 10)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("batchlaunches", 2)) = .Cells(37, 11)
```

```
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("batchlaunches", 3)) = .Cells(37, 12)
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("batchlaunches", 4)) = .Cells(37, 13)

End With

oExcel.DisplayAlerts = False
oWorkbook.SaveAs ThisDocument.Model.Path & "Database8900.xls"
oWorkbook.Close
oExcel.Quit


End Sub

Private Sub VBA_Block_1_Fire()

Dim oSIMAN As Arena.SIMAN
Set oSIMAN = ThisDocument.Model.SIMAN

'Convert array containing number of missions per year (Yearly_Demand) to array
'containing scheduled launch date for each mission (Launch_Dates) assuming constant
'spacing between missions per year, and first mission begins at time = 0

Dim i As Integer
Dim j As Integer

MissionsPerYear = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("MissionsPerYear"))
If MissionsPerYear = 0 Then
    'Number of years in program
    Num_Years = Model.Modules(Model.Modules.Find(smFindTag, "Yearly_Demand")).Data("Rows")
    ReDim Year_Demand(Num_Years)

    'Read in array containing number of missions per year
    'Set total number of missions
    Num_Missions = 0
    For i = 1 To Num_Years
        Year_Demand(i - 1) = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Yearly_Demand", i))
        Num_Missions = Num_Missions + Year_Demand(i - 1)
    Next i
Else
    Num_Years = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Num_Years"))
    ReDim Year_Demand(Num_Years)
    Num_Missions = MissionsPerYear * Num_Years
    For i = 1 To Num_Years
        Year_Demand(i - 1) = MissionsPerYear
    Next i
End If

Num_Missions = Round(Num_Missions)
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Num_Missions")) = Num_Missions
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Num_Missions_Scheduled")) = Num_Missions

ReDim Missions(Num_Missions)
Dim current As Integer

'Set array containing values for the schedule launch date for each mission
```

```
'based on first mission at t = 0
current = 0
If MissionsPerYear = 0 Then
   For i = 1 To Num_Years
      For j = 1 To Year_Demand(i - 1)
         Missions(current) = Round(365 * (i - 1) + (365 / Year_Demand(i - 1)) * (j - 1))
         current = current + 1
      Next j
   Next i
Else
   For i = 1 To Num_Missions
      Missions(current) = (i - 1) * (365 / MissionsPerYear)
      current = current + 1
   Next i
End If


'Determine expected minimum # of CEV's necessary to meet mission demand
Dim CEV_TAT, CEV_Refurb, Mission_Time, Pad, Integration, Inventory As Integer
Dim minTAT, temp, numCEV_temp, ExtraCEVs, numCEV, lifetimeCEV As Integer
Dim Mission_TAT() As Integer

CEV_Refurb = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("refurbtimeCEV"))
Mission_Time = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Mission_Time"))
Pad = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Pad_Time"))
Integration = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Integration_Time"))
Inventory = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Inventory_Time"))
lifetimeCEV = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("lifetimeCEV"))

CEV_TAT = Integration + Pad + Mission_Time + CEV_Refurb

oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("numCEV")) = 1
If lifetimeCEV > 1 Then
   j = 0
   minTAT = CEV_TAT - 1
   Do While CEV_TAT > minTAT
      j = j + 1
      temp = 1000000
      ReDim Mission_TAT(Num_Missions - j)
      For i = 1 To Num_Missions - j
         Mission_TAT(i - 1) = Missions(i - 1 + j) - Missions(i - 1)
         If Mission_TAT(i - 1) < temp Then
            temp = Mission_TAT(i - 1)
         End If
      Next i
      minTAT = temp
   Loop

   numCEV_temp = j
   ExtraCEVs = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("extraCEVs"))
   If numCEV_temp > 3 Then
      numCEV = numCEV_temp + ExtraCEVs
   Else
      numCEV = 3 + ExtraCEVs
   End If
```

```
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("numCEV")) = numCEV
End If


'If CEV goes to lunar surface, check if lander payload is sufficient
If oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("surfaceCEV")) = 1 Then
    If oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("massinCEV")) > oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("payloadLander")) Then
        oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("EndSim_Flag")) = 1
    End If
End If

End Sub

Private Sub VBA_Block_2_Fire()

Dim oSIMAN As Arena.SIMAN
Set oSIMAN = ThisDocument.Model.SIMAN


'Determine maximum launch mass:
'CEV/TEI launch together on Crew LV (single TEI, other can launch separately)
'All other stages (including extra TEIs) can launch seperately on non-crew LV

Dim payloadLV, payloadLVcrew, massLander, massTLI, drymassTLI, massTEI, launchmassCEV As Double

payloadLVcrew = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("payloadLV", 1))
payloadLV = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("payloadLV", 2))
massLander = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("massLander"))
massTLI = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("massTLI"))
drymassTLI = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("drymassTLI"))
massTEI = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("massTEI"))
launchmassCEV = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("massoutCEV"))

'Crew launch vehicle check:
If launchmassCEV > payloadLVcrew Then
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("EndSim_Flag")) = 2
End If

'Cargo launch vehicle check:
If oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Num_Launches_Prop")) = 0 Then
    If massTLI > payloadLV Then
        oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("EndSim_Flag")) = 3
    ElseIf massLander > payloadLV Then
        oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("EndSim_Flag")) = 3
    End If
Else
    If drymassTLI > payloadLV Then
        oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("EndSim_Flag")) = 3
    ElseIf massLander > payloadLV Then
        oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("EndSim_Flag")) = 3
    End If
End If
```

```
'TEI payload check
If oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("payloadTEI")) <
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("massinCEV")) Then
   oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("EndSim_Flag")) = 4
End If

'TLI payload check
If oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("numTLI")) = 0 Then
   oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("EndSim_Flag")) = 5
End If


End Sub

Private Sub VBA_Block_3_Fire()

Dim oSIMAN As Arena.SIMAN
Set oSIMAN = ThisDocument.Model.SIMAN


'Determine program start day:
'first launch - (pad time + integration time + inventory time + max. manufacturing time)

Dim TLI, TEI, Lander, CEV As Integer
Dim capTLI, capTEI, capLander, capCEV As Integer
Dim numTLI, numTEI, numCEV As Integer
Dim max_manuf, temp_manuf, Advance_Time As Integer

TLI = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildtimeTLI"))
TEI = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildtimeTEI"))
Lander = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildtimeLander"))
CEV = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildtimeCEV"))

capTLI = Model.Modules(Model.Modules.Find(smFindTag, "TLI Manufacture")).Data("Capacity")
capTEI = Model.Modules(Model.Modules.Find(smFindTag, "TEI Manufacture")).Data("Capacity")
capCEV = Model.Modules(Model.Modules.Find(smFindTag, "CEV Manufacture")).Data("Capacity")

numTLI = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("numTLI"))
numTEI = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("numTEI"))
numCEV = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("numCEV"))

'Determine maximum manufacturing lead time
max_manuf = 0

If Int(numTLI / capTLI) <> (numTLI / capTLI) Then
   TLI = TLI * Int(numTLI / capTLI + 1)
Else
   TLI = TLI * (numTLI / capTLI)
End If
If TLI > max_manuf Then
   max_manuf = TEI
End If

If TEI > max_manuf Then
   max_manuf = TEI
```

```
End If

If Lander > max_manuf Then
    max_manuf = Lander
End If

If Int(numCEV / capCEV) <> numCEV / capCEV Then
    CEV = CEV * Int(numCEV / capCEV + 1)
Else
    CEV = CEV * (numCEV / capCEV)
End If
If CEV > max_manuf Then
    max_manuf = CEV
End If


'Days before first schedule launch when program must begin:
Dim Pad, Integration, Inventory As Integer
Pad = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Pad_Time"))
Integration = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Integration_Time"))
Inventory = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Inventory_Time"))

Advance_Time = Pad + Integration + Inventory + max_manuf


'Adjust array containing scheduled start day for each mission such that it no longer starts at t=0
'(based on required advance time for manufacturing)

Dim i As Integer
For i = 1 To Num_Missions
    Missions(i - 1) = Missions(i - 1) + Advance_Time
Next i

Model.Modules(Model.Modules.Find(smFindTag, "Launch_Dates")).Data("Rows") = Num_Missions + 1
Model.Modules(Model.Modules.Find(smFindTag, "Launch_Dates_Actual")).Data("Rows") = Num_Missions + 1
Model.Modules(Model.Modules.Find(smFindTag, "cancelledMission")).Data("Rows") = Num_Missions + 1
For i = 1 To Num_Missions
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Launch_Dates", i)) = Missions(i - 1)
Next i

oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Launch_Dates", Num_Missions + 1)) = 9999

'Convert array with launch dates to array with start date for integration process
'for each mission

Model.Modules(Model.Modules.Find(smFindTag, "Begin_Launch")).Data("Rows") = Num_Missions + 1
For i = 1 To Num_Missions
    oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Begin_Launch", i)) = Missions(i - 1) - Pad - Integration
Next i
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Begin_Launch", Num_Missions + 1)) = 99999


'Convert array with launch dates to array with start date for manufacturing for
'each mission element
```

```
'CEV:
If oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("lifetimeCEV")) > 1 Then
   Model.Modules(Model.Modules.Find(smFindTag, "buildCEV")).Data("Rows") = numCEV + 1
   For i = 1 To numCEV
      oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildCEV", i)) = Missions(0) - (Pad + Integration + Inventory) - CEV + (i - 1)
   Next i
   oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildCEV", numCEV + 1)) = 99999
Else
   Model.Modules(Model.Modules.Find(smFindTag, "buildCEV")).Data("Rows") = Num_Missions + 1
   For i = 1 To Num_Missions
      oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildCEV", i)) = Missions(i - 1) - (Pad + Integration + Inventory) - CEV
   Next i
   oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildCEV", Num_Missions + 1)) = 99999
End If

Model.Modules(Model.Modules.Find(smFindTag, "buildTLI")).Data("Rows") = Num_Missions + 1
Model.Modules(Model.Modules.Find(smFindTag, "buildTEI")).Data("Rows") = Num_Missions + 1
Model.Modules(Model.Modules.Find(smFindTag, "buildLander")).Data("Rows") = Num_Missions + 1

For i = 1 To Num_Missions
   oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildTLI", i)) = Missions(i - 1) - (Pad + Integration + Inventory) - TLI
   oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildTEI", i)) = Missions(i - 1) - (Pad + Integration + Inventory) - TEI
   oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildLander", i)) = Missions(i - 1) - (Pad + Integration + Inventory) - Lander
Next i

oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildTLI", Num_Missions + 1)) = 99999
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildTEI", Num_Missions + 1)) = 99999
oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("buildLander", Num_Missions + 1)) = 99999

End Sub

Private Sub VBA_Block_5_Fire()

Dim oSIMAN As Arena.SIMAN
Set oSIMAN = ThisDocument.Model.SIMAN

Dim i As Integer
Dim builtTLI, builtTEI, builtLander, builtCEV As Integer
Dim extraTLI, extraTEI, extraLander, extraCEV As Integer
Dim costTLI, costTEI, costLander, costCEV As Double
Dim LC, extraProdCost As Double

builtTLI = oSIMAN.CounterValue(oSIMAN.SymbolNumber("TLIs Built"))
builtTEI = oSIMAN.CounterValue(oSIMAN.SymbolNumber("TEIs Built"))
builtLander = oSIMAN.CounterValue(oSIMAN.SymbolNumber("Landers Built"))
builtCEV = oSIMAN.CounterValue(oSIMAN.SymbolNumber("CEVs Built"))

extraTEI = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("unusedElements", 1))
extraTLI = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("unusedElements", 2))
extraLander = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("unusedElements", 3))
```

```
extraCEV = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("unusedElements", 4))

costTLI = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costTLI", 2))
costTEI = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costTEI", 2))
costLander = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costLander", 2))
costCEV = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("costCEV", 2))

extraProdCost = 0
LC = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("learncurve"))
For i = (builtTLI - extraTLI + 1) To builtTLI
    extraProdCost = extraProdCost + costTLI * i ^ (Log(LC) / Log(2))
Next i
For i = (builtTEI - extraTEI + 1) To builtTEI
    extraProdCost = extraProdCost + costTEI * i ^ (Log(LC) / Log(2))
Next i
For i = (builtLander - extraLander + 1) To builtLander
    extraProdCost = extraProdCost + costLander * i ^ (Log(LC) / Log(2))
Next i
If oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("lifetimeCEV")) <= 1 Then
    For i = (builtCEV - extraCEV + 1) To builtCEV
        extraProdCost = extraProdCost + costCEV * i ^ (Log(LC) / Log(2))
    Next i
End If

oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("extraProdCost")) = extraProdCost


End Sub
Private Sub VBA_Block_6_Fire()

End Sub

Private Sub ModelLogic_RunEndReplication()

Dim oSIMAN As Arena.SIMAN
Set oSIMAN = ThisDocument.Model.SIMAN

Dim extraRows As Integer
extraRows = 9

Set oExcel = CreateObject("Excel.Application")
oExcel.Visible = False
Set oWorkbook = oExcel.Workbooks.Open(ThisDocument.Model.Path & "Results.xls")
Set oWorksheet = oWorkbook.ActiveSheet

With oWorksheet
    .Cells(2, 4) = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Replications"))
    .Cells(3, 4) = oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Confidence"))
    .Cells(currentRep + extraRows, 2) = currentRep
    .Cells(currentRep + extraRows, 3) = oSIMAN.OutputStatisticValue(oSIMAN.SymbolNumber("Life Cycle Cost"))
    .Cells(currentRep + extraRows, 4) = oSIMAN.OutputStatisticValue(oSIMAN.SymbolNumber("Cost Per Mission"))
    .Cells(currentRep + extraRows, 5) = oSIMAN.OutputStatisticValue(oSIMAN.SymbolNumber("Cost Per Mission No
DDTE"))
    .Cells(currentRep + extraRows, 6) = oSIMAN.OutputStatisticValue(oSIMAN.SymbolNumber("LOC"))
    .Cells(currentRep + extraRows, 7) = oSIMAN.OutputStatisticValue(oSIMAN.SymbolNumber("LOM"))
```

```
    .Cells(currentRep + extraRows, 8) = oSIMAN.OutputStatisticValue(oSIMAN.SymbolNumber("Mission Capture Rate"))
    .Cells(currentRep + extraRows, 10) = oSIMAN.OutputStatisticValue(oSIMAN.SymbolNumber("Avg Launch Delay"))
    .Cells(currentRep + extraRows, 11) = oSIMAN.OutputStatisticValue(oSIMAN.SymbolNumber("Avg LEO Time"))
    .Cells(currentRep + extraRows, 9) = oSIMAN.OutputStatisticValue(oSIMAN.SymbolNumber("Cancelled Missions"))
    .Cells(currentRep + extraRows, 12) = oSIMAN.OutputStatisticValue(oSIMAN.SymbolNumber("CEV Queue Time"))
    .Cells(currentRep + extraRows, 13) = oSIMAN.OutputStatisticValue(oSIMAN.SymbolNumber("Lander Queue Time"))
    .Cells(currentRep + extraRows, 14) = oSIMAN.OutputStatisticValue(oSIMAN.SymbolNumber("TLI Queue Time"))
    .Cells(currentRep + extraRows, 15) = oSIMAN.OutputStatisticValue(oSIMAN.SymbolNumber("TEI Queue Time"))
End With

currentRep = currentRep + 1

If currentRep > oSIMAN.VariableArrayValue(oSIMAN.SymbolNumber("Replications")) Then
    oWorksheet.Cells(1, 1) = 0
End If

oExcel.DisplayAlerts = False
oWorkbook.SaveAs ThisDocument.Model.Path & "Results.xls"
oExcel.Visible = False
oWorkbook.Close
oExcel.Quit

End Sub

Private Sub ModelLogic_RunEndSimulation()

Set oExcel = CreateObject("Excel.Application")
oExcel.Visible = True
Set oWorkbook = oExcel.Workbooks.Open(ThisDocument.Model.Path & "Results.xls")
Set oWorksheet = oWorkbook.ActiveSheet

End Sub
```

Appendix B: Vehicle Database

B.1 Launch Vehicles

L.V.	Payload Type	Payload (kg)	DDTE ($M)	Launch Price ($M)	Reliability
Delta IV Medium	Crew	8,710	2,885	219	0.972
Atlas V 502	Crew	9,605	2926	245	0.972
Atlas V Heavy	Crew	19,060	2,926	321	0.956
Delta IV Heavy	Crew	22,850	3,258	321	0.967
Centurion C1	Crew	35,000	6,000	718	0.986
Shuttle SRB Stick	Crew	20,000	2,600	412	0.963
Shuttle ET Derived	Crew	34,000	1,750	546	0.965
Atlas V Heavy	Cargo	19,060	0	257	0.956
Delta IV Heavy	Cargo	22,850	0	257	0.967
Colossus (40mt)	Cargo	40,000	4,009	550	0.993
Colossus (70mt)	Cargo	70,000	4,829	615	0.991
Centurion C2	Cargo	100,000	6,000	785	0.984
Centurion C3	Cargo	140,000	6,000	845	0.984
Shuttle C	Cargo	77,000	2,495	588	0.961
Vega RLV	Cargo	6,036	0	127	0.999

The payload and reliability for the Delta and Atlas launch vehicles are based on existing data.[18,19,20] The DDTE and launch price were determined as follows, all converted to 2005 dollars[21]:

- Delta IV Medium (crew):
  - DDTE: $1B to human-rate core; $1250M to human-rate engines (1 RS-68, 4 RL-10s); $635 for new upper stage (based on 2850kg dry weight)[22,23]
  - Launch Price: $100 M for cargo version (based on 1999 estimate) was increased by 75% to $175 in 2004 dollars (same increase as Atlas V 502) + 25% for manned considerations[22]
- Atlas V 502 (crew):
  - DDTE: $1B to human-rate core; $400M to Americanize production of RD-180; $1B to human-rate 4 RL-10 Engines; $526 for new upper stage (based on 2026 kg dry weight)[22,23]
  - Launch Price: $196M for cargo version + 25% for manned considerations[22]
- Delta IV Heavy (crew):
  - DDTE: $1B to human-rate core; $1500M to human-rate engines (2 RS-68s, 4 RL-10s); $758 for new upper stage (based on 3940kg dry weight)[22,23]

  - Launch Price: $257M for cargo version + 25% for manned considerations[22]
- Atlas V Heavy (crew):
  - DDTE: $1B to human-rate core; $400M to Americanize production of RD-180; $1B to human-rate 4 RL-10 Engines; $526 for new upper stage (based on 2026 kg dry weight)[22,23]
  - Launch Price: $257M for cargo + 25% for manned considerations[22]

Data for the Shuttle-derived vehicles came from various sources. Shuttle C is based on an existing study on the possible expanded functionality of the Shuttle program[24]. The Shuttle SRB Stick is a single stick 4-segment expendable SRB core, with a new upper stage using 2 J-2S engines and new Al-Li tanks and structure. The DDTE assumes $600M for the new upper stage and $2000 for the J-2S engines. The launch price is based on $262M in fixed costs[22], plus $49.5M for the SRB, $66.5M for the J-2S, and $34.4M for the upper stage.[23] The Shuttle ET Derived is an in-line vehicle, using the ET with four expendable SSME engines on the first stage (with engine out capability). It also has two strap-on 3-segment SRB motors. It has a new upper stage with 4 RL-10A engines, AL-Li tanks, and engine out capability. The DDTE assumes $750M for the new upper stage and $1000M to human-rate the RL-10 engines. The launch prices is based on $262M in fixed costs[22], $100M for the SSMEs, $40M for the RL-10 engines, $45M for the upper stage, and $99 for the SRBs. The reliability is based on an aggregation of current reliability number[19].

Colossus, Centurion, and Vega are all based on Georgia Tech Space Systems Design Lab conceptual design studies.

B.2 In-Space Elements

Element	Pay-load (kg)	Dry Weight (kg)	Gross Weight (kg)	DDTE ($M)	TFU ($M)	Fixed Ops ($M)	Var Ops ($M)	Prop Cost ($M)	Relia-bility	Manuf. Time (days)
Manticore (TLI/LOI)	18,000	6,418	44,812	942	120	70	0.14	0.04	0.95	180
Manticore (TLI/LOI)	30,000	9,428	71,424	1,062	141	70	0.14	0.06	0.95	180
PPM (TEI)	7,700	5,232	10,183	2,335	434	40	1.58	0.07	0.98	250

Manticore and PPM are based on Georgia Tech Space Systems Design Lab conceptual design studies. Manticore uses LOx/LH2 as its propellant, while the PPM uses NTO/MMH. The variable operations costs and propellant costs are both given on a per mission basis. The reliability represents the probability of failure. Whether that results in loss of crew or loss of mission is determined in the Arena models by assigning a probability of catastrophic failure and a probability of a successful abort if a failure occurs for one of these in-space elements.

B.3 Landers

Element	Pay-load (kg)	Dry Weight (kg)	Gross Weight (kg)	DDTE ($M)	TFU ($M)	Fixed Ops ($M)	Var Ops ($M)	Prop Cost ($M)	Relia-bility	Life-time	Manuf. Time (days)
Eagle	500	4,104	11,549	1,124	194	3.4	0.96	0.01	0.996	1	360
Artemis	8,000	7,975	43,067	3,328	424	90	0.02	0.04	0.966	10	480

Eagle and Artemis are based on Georgia Tech Space Systems Design Lab conceptual design studies. The payload listed represents the round-trip payload capacity, to and from the lunar surface. As before, the variable operations costs and propellant costs are given on a per mission basis. Reliability essentially refers to loss of crew, since a lander failure will results in a loss of crew event. The lifetime represents the maximum number of missions before the lander must be retired. Eagle is an expendable lander, so is only good for one mission. Artemis can be used for up to ten missions or five years, whichever comes first.

B.4 CEVs

Element	Launch Mass (kg)	In-Space Mass (kg)	DDTE ($M)	TFU ($M)	Fixed Ops ($M)	Var Ops ($M)	Relia-bility	Life-time	Manuf. Time (days)	TAT (days)
Capsule	12,200	7,600	2,500	300	50	20	0.99	1	360	0
Tempest-1	11,749	9,566	2,272	339	382	24	0.99	10	540	22
Tempest-2	15,756	12,830	2,843	451	382	24	0.99	10	540	22
Tempest-3	17,653	14,373	3,073	498	382	24	0.99	10	540	22

The Capsule in an Apollo CSM-derived vehicle. Tempest is based on a Georgia Tech Space Systems Design Lab conceptual design study. Although it was not used in this study, it is provided as a reference as an available element in LASSO. The three different Tempests (1, 2, and 3) represent three internal volume categories: cramped, comfortable, and spacious.