Modeling Operations Costs for Human Exploration Architectures

Robert Shishko, Ph.D.*
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109

Operations and support (O&S) costs for human spaceflight have not received the same attention in the cost estimating community as have development costs. This is unfortunate as O&S costs typically comprise a majority of life-cycle costs (LCC) in such programs as the International Space Station (ISS) and the now-cancelled Constellation Program. Recognizing this, the Constellation Program and NASA HQs supported the development of an O&S cost model specifically for human spaceflight. This model, known as the Exploration Architectures Operations Cost Model (ExAOCM), provided the operations cost estimates for a variety of alternative human missions to the moon, Mars, and Near-Earth Objects (NEOs) in architectural studies. ExAOCM is philosophically based on the DoD Architecture Framework (DoDAF) concepts of operational nodes, systems, operational functions, and milestones. This paper presents some of the historical background surrounding the development of the model, and discusses the underlying structure, its unusual user interface, and lastly, previous examples of its use in the aforementioned architectural studies.

Nomenclature

DC	=	duty cycle
MTBF	=	mean time between failure
K	=	k-factor
e	=	element index
p	=	ORU application index
QPA	=	quantity per application
RIO	=	repair-in-orbit rate
P(i,e)	=	set of ORU applications of ORU type *i* in Station element *e*
i	=	ORU index

I. Introduction

OPERATIONS and support (O&S) costs for human spaceflight have not received the same attention in the cost estimating community as have development costs. This is unfortunate as O&S costs typically comprise a majority of life-cycle costs (LCC) in such programs as the International Space Station (ISS) and the now-cancelled Constellation Program. It is ironic too because program decisions made early can often have profound effects, both positive and negative, on the level of O&S costs required for the remainder of the program.

Recognizing this, the Constellation Program and NASA HQs (2005-2010) supported the development of an O&S cost model specifically for human spaceflight. This model, known as the Exploration Architectures Operations Cost Model (ExAOCM), provided the operations cost estimates for a variety of alternative human missions to the moon, Mars, and Near-Earth Objects (NEOs) in architectural studies. ExAOCM is philosophically based on the DoD Architecture Framework (DoDAF) concepts of operational nodes, systems, operational functions, and milestones. As a consequence of DoDAF's inherent flexibility, ExAOCM permits any architecture to be accurately represented, and allows the user to estimate operations costs for alternative operations concepts and architecture evolution plans. The model produces fiscal year operations cost estimates against a user-specified operations WBS.†

---

* Principal Systems Engineer/Economist, Mission Systems Concepts Section, M/S 301-160, Senior Member

† The cost information contained in this document is of a budgetary and planning nature and is intended for informational purposes only. It does not constitute a commitment on the part of JPL and/or Caltech.

This paper presents some of the historical background surrounding the development of the model, and discusses the underlying structure, its unusual user interface, and lastly, previous examples of its use in the aforementioned architectural studies.

II. How Modeling Operations Costs Supports Systems Engineering

New human spaceflight programs, in particular, face a mountain of upfront development costs and perhaps several decades of yearly operations costs that often have to compete in the budget process with other needs. With that in mind, the early systems engineering process with its focus on trade studies is intended to help make better decisions regarding the system architecture and associated concept of operations. Serious modeling of operations costs and development costs together helps to establish the Pareto surface between life-cycle costs and the attainment of human spaceflight program goals. At the least, such modeling can help decision makers determine whether a credible, worthwhile, and *affordable* architecture exists.

Beyond that, modeling operations costs permits trades between development costs and operations costs. For example, along with other engineering models and development cost models developed for Space Station Freedom (SSF) in the 1980s[1], an operations cost model called MESSOC supported subsystem trades between photovoltaic versus solar dynamic power and between hydrazine and cryogenic propulsion.[2] Other trades that required estimates of operations costs included those involving supportability and commonality. Specifically, what is the optimal repair level for Station Orbital Replacement Units (ORUs)?[3] At what level does commonality of ORUs make sense?

With MESSOC and the International Space Station (ISS) Probabilistic Risk Assessment (PRA), trades between operations cost and operational risks were also possible.[4] One example is the trade between resupply cycles and pre-positioned supplies versus the probability of loss of mission, known as P_{LOM}. Such a trade was possible because both models shared common failure data for Station ORUs.

III. Modeling Human Spaceflight Operations Costs: What's Different?

Modeling human spaceflight O&S costs for programs like SSF, ISS and Constellation's lunar outpost is different from modeling their development costs. This section describes some of these essential differences. First, for O&S costs, there is no historical cost database of previous programs from which to develop statistical relationships or draw statistically valid conclusions. This is hard enough for development costs using, for example, the NASA/Air Force Cost Model (NAFCOM), but for operations costs a different approach is needed.

Second, estimating development costs for a system early on usually requires a few key parameters such as mass, power, etc., but estimating O&S costs needs different system parameters, such as MTBF, *plus* a concept of operations that has sufficient clarity. The concept of operations may start with Design Reference Missions (covering how the primary systems of an architecture will be employed), but it may need to go beyond that to cover such aspects as operational timelines and phases, key operational events, and interfaces to other systems in a *system-of-systems* setting, e.g., ISS with Soyuz/Progress, Shuttle, ATV, HTV, TRDSS, etc. The full concept of operations must also spell out the end-to-end communications strategy, the integrated logistics strategy, and the operational facilities to be used.

Third, with development costs, the timing of such costs is usually estimated by applying an ad hoc "spreader function," such as a beta, to the total system development cost estimate. This spreader function recognizes that system developments costs typically have a buildup phase, peak, and then taper off. For O&S costs, when such costs are expected to be incurred must be based on an explicit *scenario*. A scenario generally consists of enduring operations activities and specific planned events, and represents a time-tagged instantiation of the concept of operations.

As with any cost model, it is important to identify the model's scope. For programs such as Shuttle, ISS, and Constellation (and its eventual successor), operations are not limited to just launch processing and launch site support. An O&S cost model must also include mission and training operations (and the maintenance of their respective facilities), post-delivery sustaining engineering (both government and "factory support"), space communications infrastructure and support, integrated logistics support (ILS) (all levels), and program management and integration (all levels). Consequently, there are numerous diverse O&S functions that must be modeled, so fourth, a complete O&S cost model's cost breakdown structure is typically more heterogenous than one sees in a development cost model like NAFCOM.

In order to estimate the cost of these heterogeneous functions in any meaningful way requires many parameters. Generally the values of these parameters are program-specific, and more so than with obtaining parameter values for development cost models, the subject matter experts are typically dispersed among many organizations. Early in a human spaceflight program, data supplied by the subject matter experts (even when they can be located) is subject to

a great deal of uncertainty, so an O&S cost model—much the same as in a development cost model—should be capable of dealing with cost risk, e.g., by having the capability to generate a cost S-curve.

IV. A Brief History of Operations Costs Modeling for SSF and ISS

This section recounts some of the work done on Space Station Freedom and the International Space Station aimed at modeling O&S costs from an engineering point-of-view.

A. Space Station Freedom (1985-1990)

In December 1984, the NASA Associate Administrator for Space Station asked the Program Manager at the Johnson Space Center (JSC) to initiate a formal task to estimate the operations costs for Space Station Freedom.[5] Then early in 1985, the Program Manager asked JPL to develop models and a process for doing design-to-cost. It became immediately clear that what was needed was full life cycle cost management, so design-to-cost quickly became design-to-life-cycle-cost (DTLCC). The subsequent effort resulted in two models: the Station Design Tradeoff Model (SDTM) and the aforementioned Model for Estimating Space Station Operations Costs (MESSOC). The DTLCC imperative meant that MESSOC would not only have to estimate O&S costs, but it had to estimate the load on the logistics system, which at the time meant Shuttle upmass, and the load on the "human system," namely the crew workload. This dictated the level of detail required by the model.

The overall requirements for MESSOC were fairly simple: (1) Provide high-level insight into and *defensible* estimates for O&S costs, (2) Tie operations cost estimates to Station technical performance metrics (TPMs) within a consistent framework, (3) Recognize and accommodate year-to-year changes in the Station's configuration and operations activities, and (4) To meet the DTLCC imperative, develop the capability to estimate marginal costs and marginal logistics "cross-consumptions," e.g., the change in annual spares mass per unit change in Station power. The term defensible was interpreted as meaning that the cost relationships should be logical and the resultant estimates should be traceable to design choices, engineering data, and operations and safety policies.

The development strategy therefore was to emphasize causal (i.e., engineering-based or physics-based) relationships wherever possible. To this end, MESSOC algorithms and equations in the areas of logistics engineering, training, orbital mechanics and environments, and launch vehicle performance benefitted from legacy NASA and DoD models. Logistics engineering, in particular, was a fertile area for initial algorithm development. To estimate the cost for spares and repairs, MESSOC had to calculate how many ORUs of each type would fail and would have to be replaced in any given time frame. A starting point in the lengthy algorithm for estimating total spares and repair costs is shown as Eq. (1):

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

The inputs included design and engineering factors, such as the quantity of each ORU and its MTBF, as well as the maintenance strategy for each ORU. Early in the SSF program, this kind of input data was not available from the Station contractor design teams, so the O&S cost team called attention to this and initiated a program-wide activity to collect logistics support data much earlier in the program than had been planned.

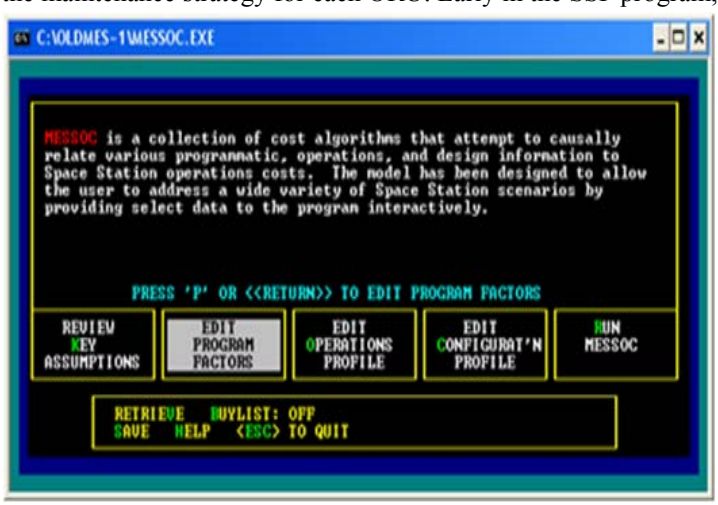

Figure 1. MESSOC User Interface for SSF c. 1987.

Over the SSF timeframe, MESSOC evolved through several versions, reflective of the rapid prototyping development strategy. These versions incorporated new or improved algorthms and improvements in the data and user interface. Most important though, these versions incorporated changes in the Station's configuration and concept of operations. For example, the April 1987 version incorporated the results of the Configuration Evaluation Task Force (CETF), which changed the basic design of the Station from the dual-keel configuration to its final configuration. The

December 1987 version incorporated changes in the concept of operations resulting from the Space Station Operations Task Force (SSOTF). By December 1989, the first configuration-managed version of MESSOC was released. This version was the result of extensive verification and validation (V&V) activities, both independent and within the program.[6,7] A year later, the final SSF version of MESSOC was released in conjunction with closeout activities for the SSF Program Office.

Another development strategy was to separate the data from the algorithms so that each could eventually be configuration-managed independently. Figure 1 shows a screen shot of MESSOC's user interface upon startup on a PC platform. By comparison with what is possible with today's development environments for PC software, MESSOC software was at best primitive. The software was coded in TurboPascal running under DOS 3.0, and the underlying databases were accessed from dBase III+ files. The user interface was neither intuitive nor friendly, so even with the last release only a small group of analysts were able to execute an estimate. Nevertheless, a user's guide and documentation of the algorithms and equations were taken very seriously during development so as to make analyst training and model maintenance less onerous.

B. International Space Station (1998-2003)

In 1998 when the first module of the ISS was launched, interest in O&S costs was renewed and development work on MESSOC restarted. This time, advances in PC software allowed for a much improved user interface and analytic engine coded in Excel © VBA. The graphical user interface used dialog boxes for user input and help files were available at every step. MESSOC had fully relational data tables, which were stored in Excel, but were also available in Access©. The algorithms and equations that made up the analytic engine were now documented as HTML files. Figure 2 shows a screen shot of MESSOC's startup tab with the multi-tab dialog box for entering high-level logistics factors. The ISS version of MESSOC was a significant improvement as most NASA cost analysts felt that with Excel they were on familiar ground. This made training in using the model significantly easier.

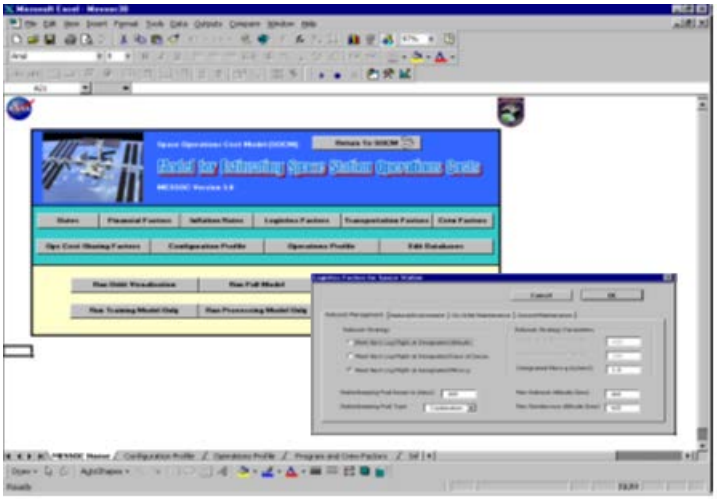

Figure 2. MESSOC User Interface for ISS c. 1999

The first ISS version debuted in December 2000 and was updated in August 2002 for use in an Independent Cost Estimate (ICE) that resulted from a report published by the ISS Management and Cost Evaluation Task Force (IMCE).[8] Using a Cost Analysis Requirements Document (CARD) prepared by the ISS Program Office, the ICE was performed by three teams, including the MESSOC team. This was significant because it provided a rare opportunity for a purely model-based cost estimate to be compared with other independent methods, tools, and cost analyst involvement. It also provided an opportunity to compare the results of MESSOC's 20 major cost categories with detailed grass-roots estimates from the Program Office. The results, which were briefed at NASA HQs in October 2002, showed a high degree of consistency with the independent and Program Office estimates.[9] The notable advantage of using MESSOC was that its estimate was more than an order-of-magnitude less costly to produce, and took far less time.

C. Lessons Learned

A number of lessons and observations can be drawn from developing and using the MESSOC versions developed for the SSF and ISS. First, MESSOC's structure made it very difficult a change or add new Station elements or potential transportation vehicles, such as the Orbital Space Plane (OSP). This usually required code changes, after which algorithms and equations had to be reverified. It was also difficult to change the WBS used to capture cost results. This also required code changes that were labor-intensive and error-prone. Further, the way dynamic cost drivers were treated was clumsy at best. Future O&S cost models for human spaceflight should not have these issues.

The second lesson learned concerns data. Operations and logistics data, in particular, if not initially specified and organized correctly, becomes a configuration management nightmare, not only because the data change rapidly

through the course of development, but because they are inherently relational. Storing the same data attribute, e.g., ORU mass, in multiple locations inevitably leads to errors and corrupted files. More importantly, it casts doubt on the resulting cost estimates.

Third, early in a human spaceflight program, the available data are likely to be sparse and rather coarse. Therefore, it makes sense to have cost algorithms and equations that function at different levels of detail depending on program maturity. Future O&S cost models for human spaceflight should have the capability to swap out portions of the analytic engine without highly skilled programmer assistance.

Lastly, software for cost estimation easily becomes obsolete. Good documentation of the algorithms and equations, data types and structures, and definitions of inputs and outputs is essential for long-term preservation, but it is not sufficient. There is a continuing need for model maintenance, data updates, and refresher training as well.

V. Exploration Architectures Operations Cost Model (ExAOCM)

The Constellation Program (2005-2010) offered an opportunity to apply these lessons learned from previous programs, and to advance the state-of-the art in human spaceflight O&S cost models and estimation. As with the earlier models, the scope of the model for Constellation was the same—that is, recurring Phase E costs. What had changed in the interim period was the significant progress in defining and using architecture frameworks, in particular, the DoD Architecture Framework (DoDAF).[10] Further, the Constellation Program was open-ended in that it was intended to develop archtitectures, systems, and concepts of operations for ISS operations, lunar exploration, and ultimately human missions to Mars. A large number of alternative architectures, systems, and concepts of operations would have to be considered in trade studies, so it would make sense to develop a new model that could adroitly handle a wide-ranging and complex tradespace. To meet this need, the development of the Constellation O&S cost model, known as the Exploration Architectures Operations Cost Model (ExAOCM), was able to draw from the timely advances in architecture frameworks and system-of-systems thinking.[11]

DoDAF concepts, language, and formalisms actually provide a natural way of conceptualizing and organizing an human exploration O&S cost model. These concepts include: Operational Nodes, covered in the DoDAF OV-2 view; Operational Functions/Activities, covered in the DoDAF OV-5 view; Systems, covered in the DoDAF SV-1 and SV-4 views; Operational Functions/Activities mapping to Systems/System Functions, covered in the DODAF SV-5 view; and System Evolution and System Technology Milestones, covered in the DoDAF SV-8 and SV-9 views. What ExAOCM adds is the concept of a Cost Driver and the links to these other DoDAF formalisms (Nodes, Systems, Operational Functions/Activities, and Milestones).

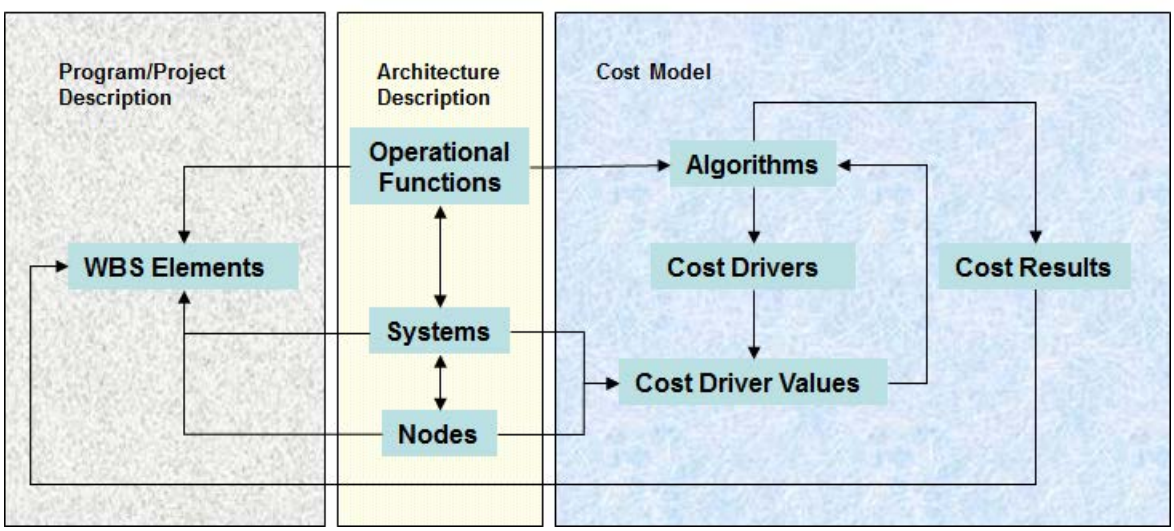

Figure 3. Conceptual Flow from an Exploration Architecture to a Costed O&S WBS

ExAOCM "works" by tying together the architecture, cost model, and program WBS perspectives. The conceptual flow is shown in Fig. 3. The concept of operations typically defines the operational functions to be performed and the operational node(s) where they may be performed. Each such operational function is represented by an algorithm in the O&S cost model. Each algorithm has arguments, which are the previously mentioned cost drivers. At a point in time, each cost driver takes on a value that is driven by a system or node attribute. Applying

the cost driver values to the algorithm results in a cost estimate, which is then placed in its proper WBS element. That element is referenced by the nexus of the operational function/activity, the system when appropriate, and the node at which the activity is performed. Since systems, their placement at various nodes, and cost driver values may change over time as the architecture matures, the milestones concept allows any of these architecture attributes to be updated and a new O&S cost for the operational function/activity to be estimated.

As an example, an operational function such as software maintenance could occur at various operational nodes and for several software-intensive systems at each node. The software maintenance algorithm might have Source Lines of Code (SLOCs), programmer productivity factors, and anticipated percent software volatility as arguments (i.e., cost drivers), in addition to programmer compensation rates. While the algorithm is the same, the cost driver values that apply to each system at each node may be quite different. Using ExAOCM, the cost analyst can easily enter the information available for each system and node, and assign the resultant estimated cost to the appropriate WBS element.

A. ExAOCM in the Constellation Program

Following the President's Vision for Space Exploration (VSE) announcement in 2005, the Exploration Systems Architecture Study (ESAS) Final Report documented the analyses and findings of a 90-day NASA-wide study.[12] This study laid out an approach for a human lunar return (HLR) that included using an Apollo-type capsule, a "1.5 launch strategy" (Ares I and Ares V), separating crew from cargo, use of Shuttle-derived technology, and a distinctly non-Apollo-type EOR-LOR (Earth Orbit Rendezvous-Lunar Orbit Rendezvous) Design Reference Mission (DRM). This study set some of the basic architectural elements for the Constellation Program along with some fundamental aspects of the concept of operations. Over the next few years, several study teams were constituted to delve further into the implementation tradespace, and these efforts led to significant refinements of the HLR architecture, systems, milestones, and DRMs. Some of the key teams and their top-level results and conclusions follow.

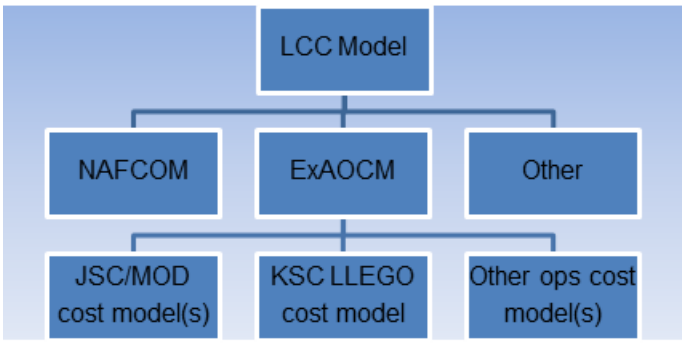

Figure 4. Life Cycle Cost Model Using ExAOCM for O&S Costs.

The Lunar Architecture Team-1 (LAT-1) reported out in December 2006. This NASA-wide team concluded that a lunar outpost would be more cost-effective than a series of lunar sortie missions and hence should be done first. It addressed the outpost architecture, systems, and build-up. It also confirmed the need for a separate lunar lander vehicle (LLV) just for cargo.

The Lunar Architecture Team-2 (LAT-2) built on the work of LAT-1, and reported out in August 2007. This team addressed aspects of the lunar outpost tradespace, such as a mini-habitation module versus a larger inflatable module, and solar versus nuclear power. It explored the attributes of the lunar surface systems in more detail and defined what the LLV might look like. From the latter, it was determined that the Heavy-Lift Launch Vehicle (HLLV), later to become the Ares V, would need to have a 10-m fairing to accommodate the LLV. The team also defined the campaign metrics that would used to evaluate the tradespace and produced a reasonably detailed LCC estimate. This LCC estimate was produced by an ensemble of cost models, shown in Fig. 4, that included NAFCOM for development costs and ExAOCM for O&S costs.

The Constellation Architecture Team (CAT)-Lunar continued to refine the lunar outpost tradespace in terms of improved habitability, power, and surface mobility. It also refined the campaign metrics and produced detailed LCC estimates for a variety of outpost build-up scenarios. The team's work culminated in June-July 2008 when the Constellation Program passed a significant programmatic milestone, called the Lunar Capability Concept Review (LCCR).[13]

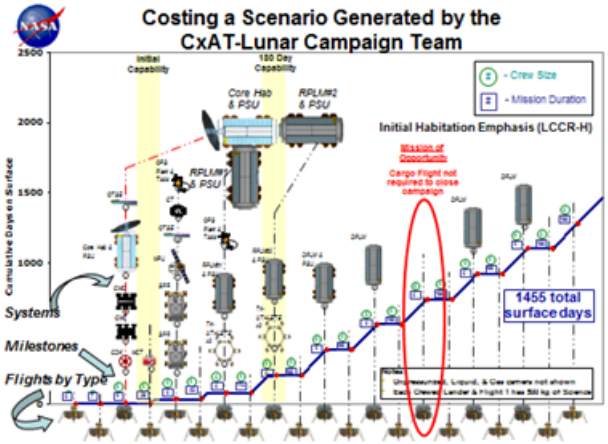

Figure 5. Lunar Outpost Build-Up Scenarios with Metrics

Over the previous three years, these various teams demonstrated that the Constellation transportation systems closed from a performance and mass margins point-of-view, and that a lunar outpost was technically feasible and consistent with the transportation systems. The CAT-Lunar was able to generate and cost on the order of 20 scenarios, with O&S costs estimated using ExAOCM. A typical CAT-Lunar scenario was summarized in a diagram like the one in Fig. 5. This is significant because much of the high-level architecture and flight rate information needed by ExAOCM could be obtained using this single graphic alone.

Follow-on work was performed by the NASA Strategic Analysis Team. This team performed further trades on the LLV and in October 2009 reported out their results on a crew-optimized LLV versus a cargo-optimized LLV. ExAOCM was used again to estimate the O&S costs for various lunar scenarios. Lastly in 2010, ExAOCM provided estimated O&S costs for the Mars Design Reference Architecture[14] (DRA) 5.0 to the Lunar-Mars Integration Team.

B. ExAOCM's Graphical User Interface

A unique aspect of ExAOCM is its Graphical User Interface (GUI) that resembles a DoDAF OV-2 diagram. When an Excel-based ExAOCM file is opened, the analyst is presented with the main screen, which displays whatever architecture the analyst has already been created overlaid on the Earth, moon, Mars and its moons, and a Near-Earth Object (NEO). The main screen shows a network of operations "nodes" of various colors, connected by arcs called "needlines," also in various colors. Figure 6 is an example of the main screen for an architecture created for a human exploration mission to a NEO. The same architecture is recreated in Fig. 7 from a different perspective, and the analyst can toggle between these displays. The GUI's purpose is to provide visual feedback to the analyst about the architecture being estimated, so even the colors are intended to convey useful information about the operational nodes and needlines.

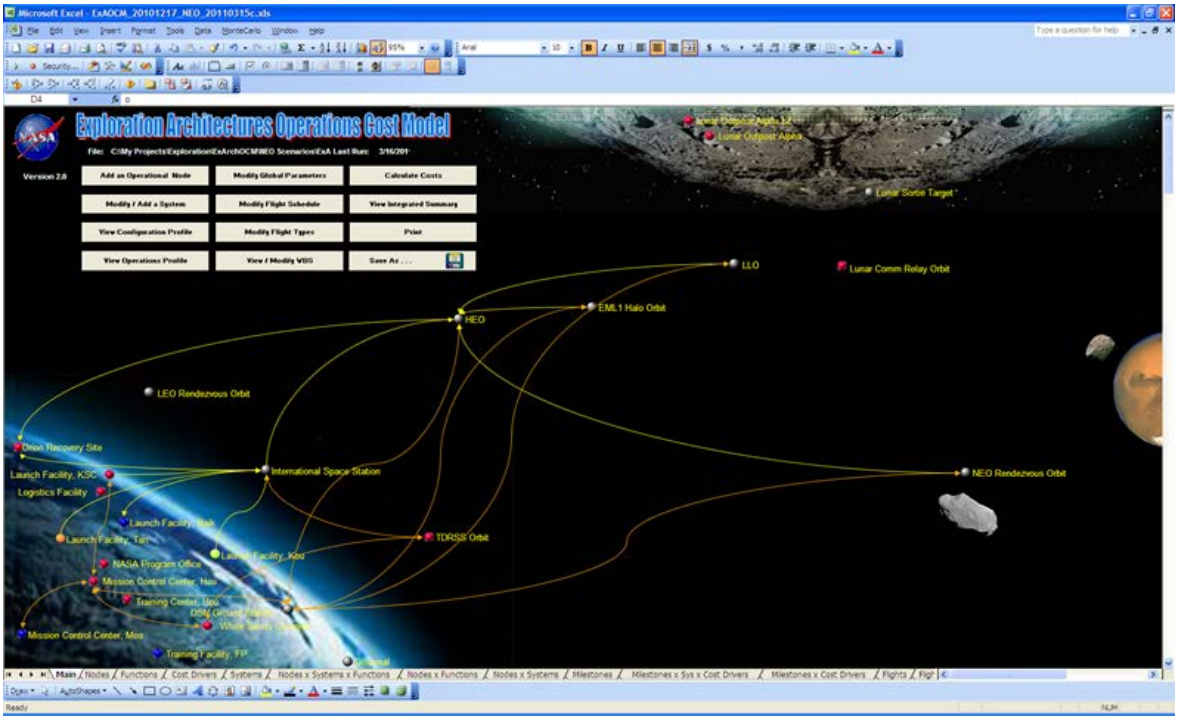

Figure 6. ExAOCM Main Screen Showing a Previously Created NEO Exploration Architecture

The GUI's second purpose is to provide a systematic way of helping the analyst modify the architecture and enter (or modify) a large set of data. (The actual work of storing all this information and making the data available to the algorithms is done by the ExAOCM relational database. The full data model is described in the ExAOCM User's Guide.[15]) For example, the analyst can add or modify operational nodes, systems, operational functions, cost drivers, and global parameters, such as the number of years in a scenario. Adding or changing these is done through individual dialog boxes.

Adding a new operational node to the architecture may be needed, for example, if the architecture is expanded to include future destinations such as Mars, or if there is a desire to add operational functions to the WBS. The choice

of when to add an operational node is up to the analyst based on his/her understanding of the architecture's concept of operations and what is a sensible allocation of operational functions to different nodes. To add an operational node, the analyst clicks the GUI's upper left button to invoke the appropriate dialog box. The dialog box asks the analyst to provide basic information such as a name for the node, where the analyst would like the name to be displayed, where the node should initially be placed on the main screen, and other descriptive information. The analyst associates the node with a specific partner (or joint) from a pull-down list. This choice determines with what color the node will appear on the main screen.

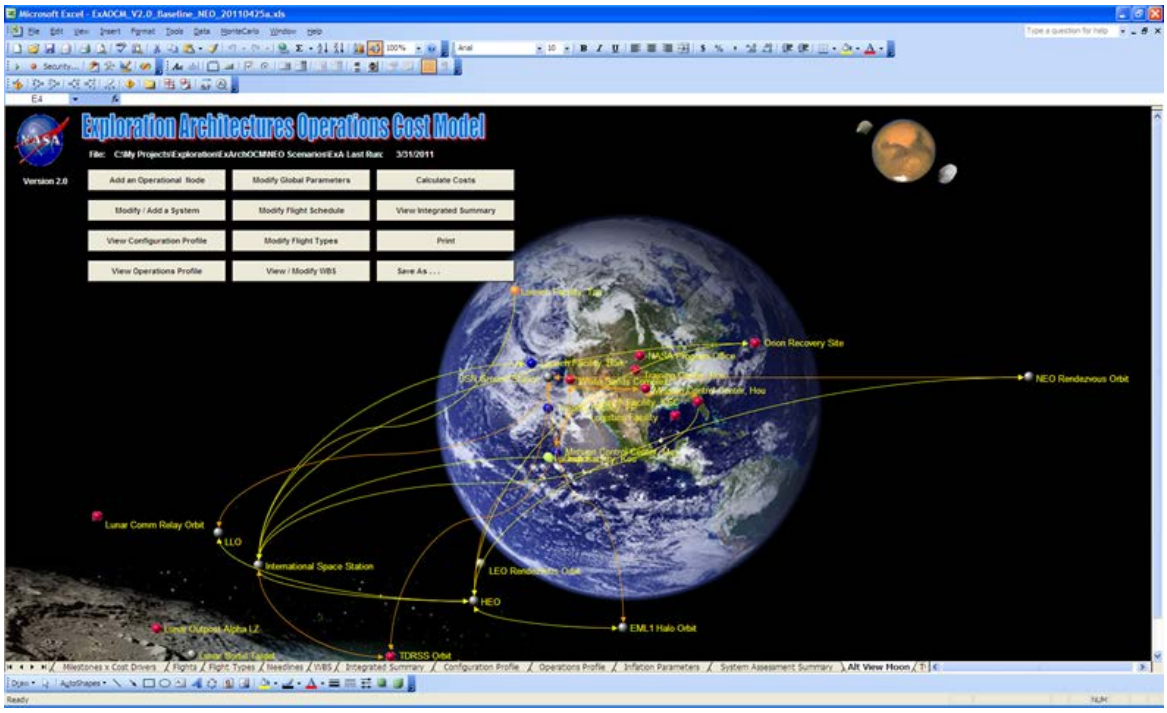

Figure 7. Alternative Display of the Previously Created NEO Exploration Architecture

Once the analyst has identified and entered all the architecture's building blocks (nodes, systems, milestones, etc.), the analyst must begin to tie these together by building relationships through the GUI. This is initiated by clicking on the node at issue, which reveals a dialog box of the possible actions listed below:

1) Change the node name and/or attributes.
2) Add or modify a milestone for the node.
3) Assign an operational function to the node.
4) Assign or delete a system at the node.
5) Assign an operational function to a system at the node.
6) Add or modify a system quantity at the node.
7) Add or modify a cost driver value at the node.
8) Add or modify a system cost driver value at the node.
9) Add or modify a needline at the node.
10) Delete a needline at the node.
11) Delete the node.

C. Building an Operations Work Breakdown Structure (WBS)

Building an operations WBS is one of the key steps in using ExAOCM. The WBS contains all the "leaf" operational functions necessary to generate a full O&S cost, as well as the structure under which these functions and associated costs are rolled up to higher levels. It is generally good practice to develop a product-based WBS even for O&S, though it may be necessary to diverge from this for practical reasons.

The WBS sheet is shown in Figure 8. The WBS sheet contains columns B through F for entering WBS element numbers in indentured form (to five levels). Columns G through K provide five levels of indenture for entering the corresponding WBS element names. It is important that the indenturing be done properly, since ExAOCM uses it to perform the cost roll-up and grouping of subordinate elements. The analyst may insert or delete rows in the WBS in the usual Excel manner and enter new WBS element numbers and names as needed; the roll-up of costs and the grouping of subordinate elements are automatically re-established at run-time (i.e., when costs are calculated).[‡]

The choice of WBS element names in the WBS sheet is entirely at the discretion of the analyst. Of course, names meaningful to the human exploration program being estimated are helpful, if a WBS has not already been provided. Even if a WBS has been provided, it may contain only high level elements, say, to the third level of indenture. The analyst can provide additional detail in order to match the level of a particular cost algorithm.

Once the WBS is developed, the analyst creates the WBS dictionary by filling out the Functions datatable, modifying or adding the leaf functions. This is needed because the WBS element names and element numbers in the WBS sheet are just text; they need to be captured in the ExAOCM database. The analyst clicks the appropriate button to invoke the corresponding dialog box. This dialog box asks the analyst to provide basic information about each leaf function such as an operational function name and description. The function's unique ID is used later in association with a particular node or system at a node to target a specific WBS element number.

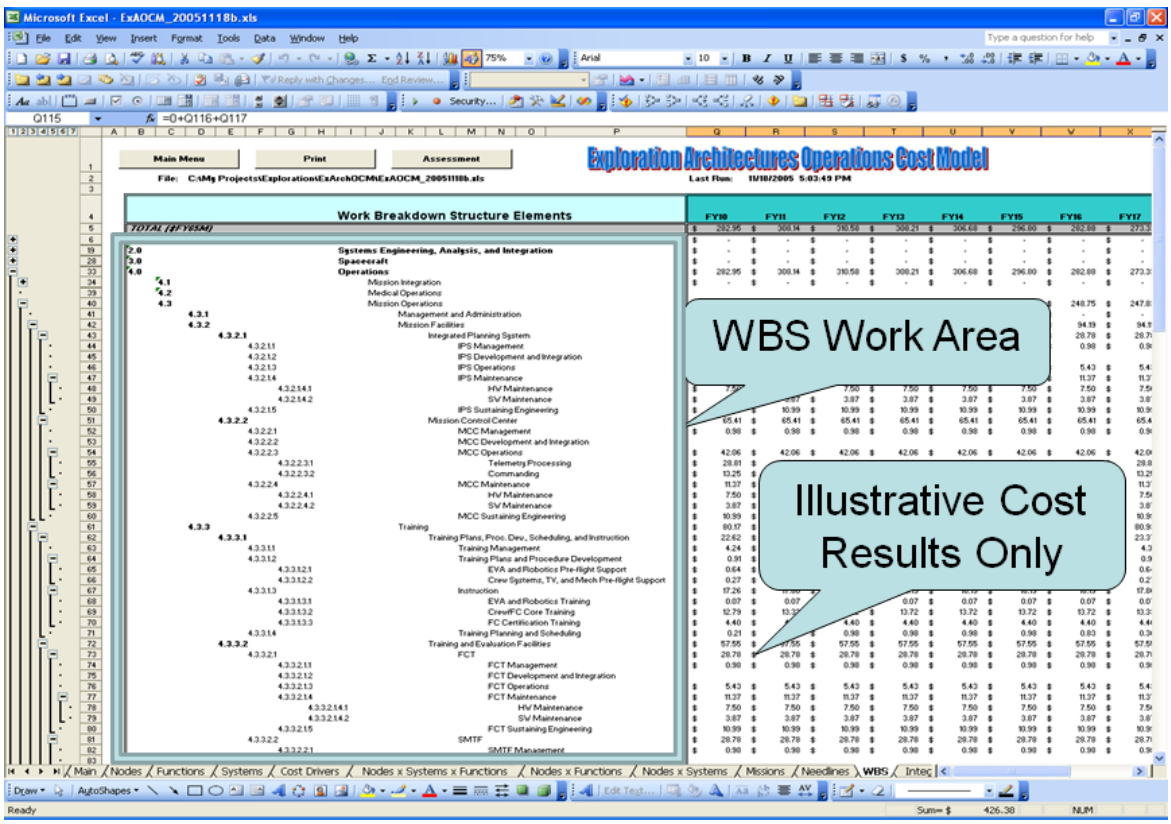

Figure 8. Display of a WBS in ExAOCM

D. Creating Flight Types

Flight rate is perhaps the most significant scenario-determined cost driver. In ExAOCM, the analyst must first create flight types, which are typically defined by where (i.e., from which node) they are launched, where (i.e., to which node) they are going, what launch vehicle is used, and what space vehicle/carrier is being transported. (Both the launch and space vehicle/carrier are considered systems in ExAOCM, and the analyst should ensure that they are

[‡] The cost information contained in this document is of a budgetary and planning nature and is intended for informational purposes only. It does not constitute a commitment on the part of JPL and/or Caltech.

already entered in the Systems datatable.) Other characteristics may be used to distinguish flight types, such as number of crewpersons and total mission duration, but the nodes and systems used are paramount. The analyst may add as many flight types as needed to accurately capture the architecture's concept of operations. To add a new flight type or modify the attributes of an existing flight type, the analyst clicks on the button with that caption. This invokes the dialog box 'Select a Destination Node'. This dialog box asks the analyst to provide a destination node from the list of in-space nodes previously created.

Selecting a destination node and clicking 'OK' invokes the second dialog box shown in Figure 9. The analyst adds or modifies flight types to the selected destination node in the Flight Types datatable by providing additional basic data such as the departure node and the launch vehicle and space vehicle/carrier associated with the flight type. This is accomplished using the dialog box's pull-down menus of previously identified systems. Additional data may be entered to help establish distinguishing characteristics of a particular flight type.

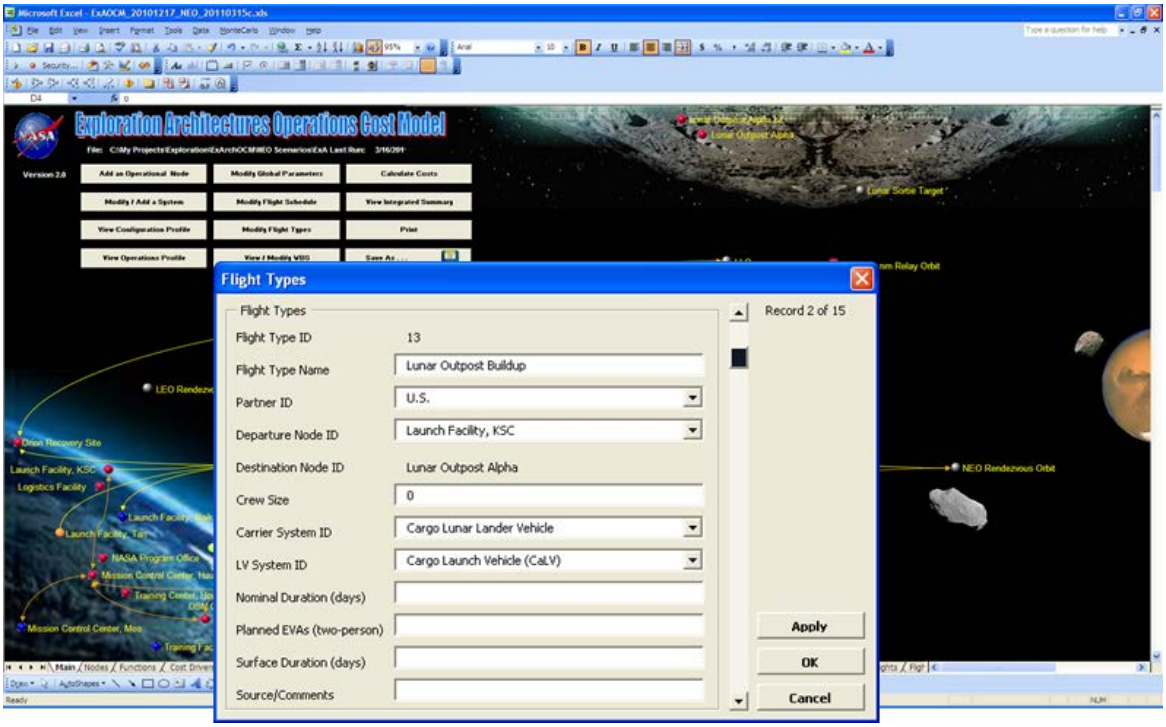

Figure 9. Dialog Box to Define Flight Types

E. Building or Modifying a Scenario Flight Schedule

Once the needed flight types have been defined, the analyst can create (or modify) a flight schedule that defines a scenario or campaign. The flight schedule is a list of all flights by type along with the anticipated launch date for each. The analyst may add as many flights to the flight schedule as needed to accurately capture a scenario or campaign. To add a new flight or modify the attributes of an existing flight, the analyst clicks on the button with that caption. This invokes the dialog box 'Which Flight Type.' This dialog box asks the analyst to provide a flight type from the list of those already entered in ExAOCM.

Selecting a flight type and clicking 'OK' invokes the second dialog box shown in Figure 10. The analyst adds or modifies flights of the selected flight type in the Flights datatable by providing the nominal flight start (i.e., launch) and arrival dates. An arrival date is optional, but it is sometimes helpful in establishing architecture milestones. The nominal flight arrival date, if provided, should be consistent with trajectory and astrodynamics constraints. The analyst can provide additional information such as the payload manifested on that specific flight. This is helpful in documenting the scenario or campaign, since it provides an important distinguishing attribute when many flights of the same type make up the scenario or campaign.

F. Using ExAOCM

Developing a full scenario for a complex exploration architecture such as an outpost-first HLR requires the work of an engineering team with significant skills. The O&S cost analyst needs to be fully familiar with their

products, and with the resulting architecture's artifacts (nodes, systems, etc.) and concept of operations. For just one such HLR scenario that the CAT-Lunar team investigated, there were 20 operational nodes, 125 individual (ground, transportation, and lunar surface) systems, and approximately 100 milestones that had to be represented in ExAOCM. ExAOCM also contained 55 separate O&S functions and 235 unique cost drivers, and datatables containing cross-products were hundreds of rows long. Because of this data volume, a logical ontology and data model are absolutely essential, and the DoDAF-based ExAOCM fulfills that requirement. Nevertheless, the cost analyst, being human, may enter data incorrectly or may incorrectly associate one data element with another. Fortunately, the analyst can exploit a number of ExAOCM-performed checks on the data. In fact, it is highly recommended.

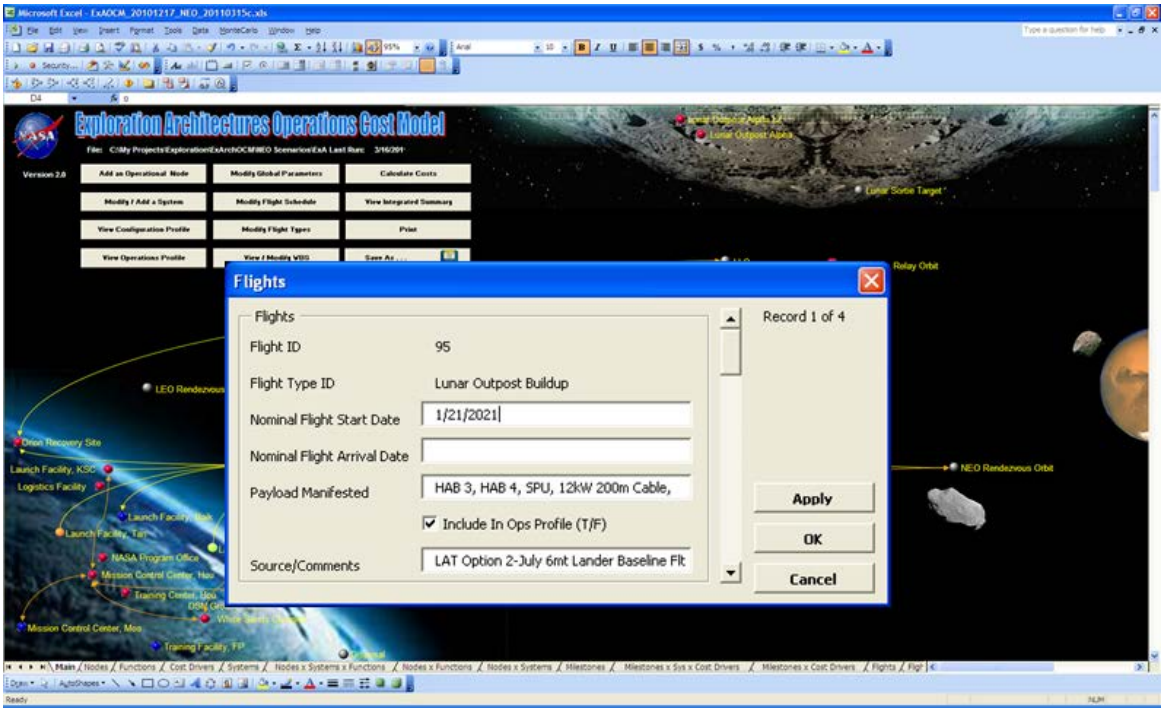

Figure 10. Dialog Box to Create a Flight Schedule Within a Scenario

There are four audits the analyst can request by clicking on the appropriate buttons. Each audit queries the ExAOCM database and produces a human-readable report (in Excel). First, the analyst can see the 'Configuration Profile', which shows for each fiscal year the quantity of each system that is associated with each node. This is extremely useful for verifying the correctness of an assembly sequence. Second, the analyst can view the 'Operations Profile', which depicts the number of flights of each type in each fiscal year. This display uses the analyst-supplied flight schedule and flight type data, and groups flights according to their respective in-space destination node. Third, the analyst can order a WBS-System audit, which for a selected system tracks all the WBS elements in which that system participates. These data are presented for each node with which the system is associated, and is useful is determining completeness of the data. Lastly, the analyst can order a full System audit, which goes beyond the WBS-System audit. The full System audit contains the same data as the simpler WBS-System audit, but also reports for each milestone, the value of every cost driver used to estimate the WBS element cost. This is useful in tracking how cost driver values may change throughout a scenario. Once the analyst is satisfied that all the data are correctly entered, clicking the upper right button on the main screen executes the model.

G. Algorithm Development for ExAOCM

ExAOCM algorithms were developed or derived from a variety of sources. For Constellation's Mission Operations Project at JSC, workforce and cost algorithms were co-developed as a joint activity of Mission Operations Directorate (MOD) personnel and the author. For Constellation's Ground Operations Project at the Kennedy Space Center (KSC), the workforce and cost algorithms drew from earlier work by the Launch and

Landing Effects Ground Operations (LLEGO) cost model team.[16] Figure 4 shows the relationship between these models and ExAOCM.

MESSOC algorithms were also coded, but could only be applied when the available data matched the algorithm's cost drivers. For example, the detailed MESSOC algorithm for spares and repair costs was coded in ExAOCM, but not used for the CAT-Lunar work, because failure data on lunar surface systems was not available at the time of the estimate. Instead, a simpler algorithm was used that relied only on the lunar surface system landed mass and known spares and repair cost factors from the ISS Program.

H. ExAOCM Verification and Validation Activities

The Verification, Validation, and Accreditation (VV&A) process for ExAOCM used NASA-STD-7009.[17] The Exploration Architectures Operations Cost Model (ExAOCM) Capability Report v1.1 was submitted to the Constellation Program Accreditation Authority in August 2008. This report documents the software development and configuration management approaches used, and the extensive software verification and validation procedures performed.

VI. Conclusion

This paper has presented some of the history and capabilities of the O&S cost models developed and used in three human exploration programs (SSF, ISS, and Constellation). Over the past 25 years, it is clear that the capabilities and software usability of human exploration O&S cost models have dramatically increased. It is also clear that increasingly defensible estimates for full life-cycle costs for future human exploration programs will be demanded by those funding such programs. With sufficient program definition, ExAOCM is ready to provide the O&S cost portion for any post-Constellation NASA human exploration program.

Figure 11 is a recapitulation of the conceptual flow from an exploration architecture to an O&S cost estimate that also shows the implementation in ExAOCM software. Just as DoDAF evolution has focused on its data model, ExAOCM, as an evolution of previous models, is the result of focusing on the organization of the data needed to build an estimate and on establishing the proper data relationships.

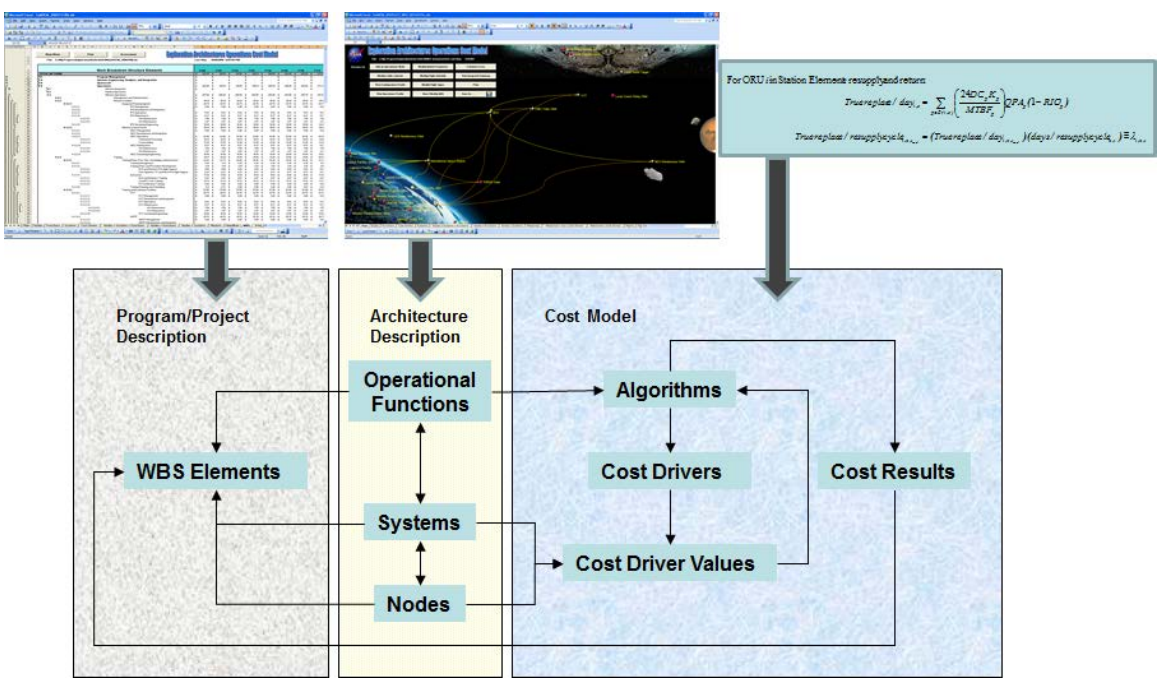

Figure 11. Conceptual Flow Diagram with ExAOCM Implementation

ExAOCM software was developed for NASA human exploration programs, but has already demonstrated some extensibility. A robotic mission version has been developed with algorithms that support the JPL standard operations

WBS. Other programs could benefit without too much difficulty. Because it is DoDAF-based, ExAOCM could be adapted to estimate the O&S costs of DoD space programs, especially those with complex ground infrastructures. Systems engineering in future human exploration programs will likely be performed using model-enhanced methods and formal tools like SysML. It is not hard to imagine that much of the data ExAOCM needs could be obtained directly from such systems engineering models, thus directly linking O&S cost estimates to architecture, concept of operations, and detailed design choices.

Acknowledgments

This research was carried out at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under a contract with the National Aeronautics and Space Administration (NASA). © 2013 California Institute of Technology. Government sponsorship is acknowledged.

References

[1]SDTM, Station Design Tradeoff Model, Software Package, Ver. 1.3, JPL Systems Office/NASA Space Station Freedom, Reston, VA, October 1990.

[2]MESSOC, Model for Estimating Space Station Operations Costs, Software Package, Ver. 2.2, JPL Systems Office/NASA Space Station Freedom, Reston, VA, December 1990.

[3]Shishko, R., "Application of MESSOC to the Repair-in-Orbit (RIO) Issue," *Space Station Operations Task Force Panel 4 Report: Management Integration, Appendix B*, NASA-TM-101819, NASA, December 1987.

[4]Smith, C.A., "Probabilistic Risk Assessment for the International Space Station," *Joint ESA-NASA Space-Flight Safety Conference*, edited by B. Battrick and C. Preyssi, European Space Agency, ESA SP-486, Noordwijk, NE, 2002, pp. 319-324.

[5]NASA, Memorandum from Philip E. Culbertson, Associate Administrator, Office of Space Station to John Hodge, Manager, Space Station Program, Johnson Space Center, dated December 28, 1984, archived at JPL, Pasadena, CA.

[6]O'Malley, T.J., Kline, R.C., and Sherbrooke, C.C., "Model for Estimating Space Station Operations Costs (MESSOC) Assessment," Logistics Management Institute (LMI) Briefing Book NS901TB1, October 1989.

[7]NASA, Memorandum from Gary L. Oleson, Member of the Technical Staff, Program Engineering and Assessments to B. Moxon, G. Williams, Space Station Utilization, et al., dated June 28, 1989, archived at JPL, Pasadena, CA.

[8]Young, A. Thomas (Chairman), "Report by the International Space Station (ISS) Management and Cost Evaluation (IMCE) Task Force to the NASA Advisory Council," NASA, Washington, DC, November 1, 2001.

[9]Shishko, R., "Model for Estimating Space Station Operations Costs," Briefing to J. Hamaker, NASA Cost Analysis Division, dated October 15, 2002, archived at JPL, Pasadena, CA.

[10]DoD Architecture Framework Working Group, "DoD Architecture Framework Version 1.0, Volume I: Definitions and Guidelines," August 2003.

[11]Shishko, R., "The Application of Architecture Frameworks to Modeling Exploration Operations Costs," *Proceedings of the 16th Annual INCOSE International Symposium*, International Council on Systems Engineering, July 2006.

[12]NASA, "NASA's Exploration Systems Architecture Study Final Report," NASA-TM-2005-214062, November 2005, URL: www.nasa.gov/exploration/news/ESAS_report.html [cited 27 July 2013].

[13]Culbert, C., "Lunar Surface Systems Project Overview," Briefing to the U.S. Chamber of Commerce Programmatic Workshop on NASA Lunar Surface Systems Concepts, February 2009, URL: www.nasa.gov/pdf/315828main_LSS_Overview_for_Industry_Culbert.pdf [cited 27 July 2013].

[14]NASA, "Human Exploration of Mars Design Reference Architecture 5.0," edited by Bret Drake, NASA-SP-2009-566, July 2009.

[15]Shishko, R. "Exploration Architectures Operations Cost Model User's Guide Version 2.0," Jet Propulsion Laboratory, April 2011.

[16]LLEGO Public Website, Zapata, E. (contact), URL: http://science.ksc.nasa.gov/shuttle/nextgen/Models/LLEGO.htm [cited 27 July 2013].

[17]NASA, *Standard for Models and Simulations*, NASA-STD-7009, Approved July 11, 2008.