A PRIMER FOR DESIGNING A SMALL SPACECRAFT MISSION OPERATIONS ARCHITECTURE

A Thesis

Presented to

the Faculty of the College of Science

Morehead State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

Joseph M. A. Feliciano

April 28, 2022

Accepted by the faculty of the College of Science, Morehead State University, in partial fulfillment of the requirements for the Master of Science degree.

Dr. Benjamin K. Malphrus
Director of Thesis

Master's Committee: ____________________, Chair
Dr. Benjamin K. Malphrus

Nathan D. Fite

Jeffrey Gillette

Date

A PRIMER FOR DESIGNING A SMALL SPACECRAFT MISSION OPERATIONS ARCHITECTURE

Joseph M. A. Feliciano
Morehead State University, 2022

Director of Thesis: ______________________________
Dr. Benjamin K. Malphrus

As the small spacecraft industry matures, so too does the need for ground assets that sustain those missions. While the forecasted cislunar and interplanetary mission profile for small spacecraft remains optimistic, the evolution of small satellite infrastructure towards extra-Earth orbiters is outpacing the ground support elements specific to that infrastructure. Therefore, a forthcoming contrast to that forecast signals attention: the unavailability or absence of exclusive ground assets limiting the advancement of extra-Earth small spacecraft. Ground elements capable of supporting the quantity of small spacecraft in the upcoming generation are present; however, they are integrated in an already stressed architecture and infrastructure oriented more towards larger missions with abundant resource pools. Considering the opportunity, a small spacecraft-based mission would have to adapt towards the established architecture set by large spacecraft-based missions; consequently, requiring a resource pool misaligned with the mission's scope.

The proposed solution is a mission operations architecture, or MOA, for small spacecraft-based missions. Essentially, a ground system design that is inclusive to Earth-based, cislunar, an interplanetary small spacecraft-based missions. The barebone concepts of this design are traced back to the fundamentals of mission operations: the "Who, What, When, Where, How, and Why" of the mission. By correctly identifying the elements that comprise a MOA, the interested team(s) can fabricate a mapping of the underlying data flow; inevitably, establishing a useful guide for the data management of the mission. Since it is crucial for the management of both telemetry and commanding data, this design covers both avenues in two separate configurations and addresses their respective interfaces.

Morehead State University's NASA sponsored Lunar IceCube cislunar CubeSat serves as a case study for the implementation of such a MOA. Additional legacy and present architectures are also included to present the concept of a MOA as a vital piece in the success of any mission and to draw differences between the needs of large and small spacecraft. Overall, the main purposes of this primer are to draw attention to an ongoing paradigm shift: a transition from large, flagship spacecraft to smaller spacecraft; and, concurrently, to stimulate the need for exclusive ground assets counteracting future impediments towards the next generation of extra-Earth small spacecraft.

Accepted by: ______________________________, Chair
Dr. Benjamin K. Malphrus

____
Mr. Nathan D. Fite

____
Mr. Jeffrey Gillette

Table of Contents

Illustrations

Figures

Tables

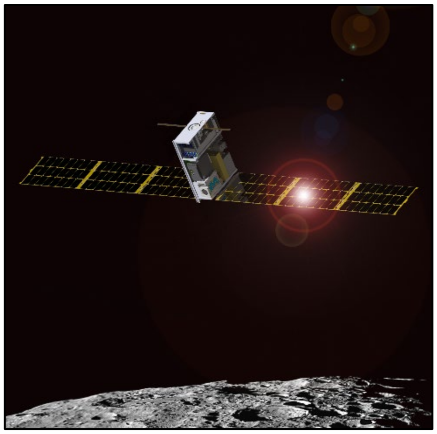


Figure 0. Lunar IceCube CubeSat.

1) The Blueprint of Operations: What is a Mission Operations Architecture?

1.1) Executive Summary

A little over six decades ago humanity initiated its venture for positioning machine, and eventually man, on its next foreground: space. Since its inception, humanity has been utilizing the exploits of space with the prospects of an optimistic future for the next generation on Earth in terms of knowledge, accessibility, and efficiency. However, none of these conceptions and visions for a healthy and sustainable future were possible without the advent of the spacecraft. The spacecraft, as a concept, was the crown jewel of innovation and the brainchild of the early aerospace and aviation industries. Whoever possessed the design, manufacturing, and engineering capacity to produce such a marvel, along with proving its practicality, was launching their name into the history books. From its origin to its current peak, spacecraft have been the fuel of great national collaborations, such as NASA and ROSCOSMOS; and the prowess of corporations seeking prestige, such as SpaceX, Blue Origin, and Virgin Galactic. No matter the fuel or prowess involved, the spacecraft brought with it a disclaimer: it is only as good with respect to whatever is on the ground supporting its operations. Therefore, in parallel with the spacecraft's six-decade history, mission operations came into its own, and brought forth multiple flavors for conducting and managing space missions.

However, “Davids” are rising amongst the Goliaths of the space industry influencing a paradigm shift to smaller more accessible spacecraft and with that their operations. The key elements composing a successful mission are no longer in the hands of these giants but are accessible to those that seek to design, build, launch, and operate a spacecraft without the established imposition of the giants. However, this shift is apparently slow. There is a potential mismatch in the progression of the industry with respect to ground assets when compared to the progression of launch vehicles and the spacecraft. Ground assets have the slowest pace in progression. Therefore, this in turn affects the mission operations on the small spacecraft. While the spacecraft is not the focus here, the operations are. Essentially, what is proposed are the key elements for operating and managing smaller spacecraft in the apparent slow growth of ground assets for said small spacecraft. This being a mission operations architecture for small spacecraft.

There are six Goals for this primer for designing a mission operations architecture for small spacecraft:

I. Explain a mission operations architecture
II. Explain the purpose behind small spacecraft
III. Reflect upon legacy missions and their antiquated architectures to highlight precedents relevant to current architectures
IV. Reflect upon recent missions and their architectures to highlight successes for small spacecraft
V. Define the pertinent elements of a mission operations architecture for small spacecraft
VI. Use a case study (Lunar IceCube) to showcase that the designed architecture can support a complex, deep space small spacecraft-based mission

Goal VI is intended to be the apex of this project, combining the bedrock concepts presented from Goals I-V into a practical example that can be either replicated or customized to the needs of the aerospace industry and community. The case study focuses on a NASA mission named Lunar IceCube (LIC) (Figure 0), a 6U cislunar CubeSat that is managed by Morehead State University in Morehead, KY. LIC is tasked with prospecting for water in solid, liquid, and vapor forms and other lunar volatiles from a low-perigee, highly inclined lunar orbit. Sponsored by NASA's NextSTEP program, LIC is one of ten CubeSats slated to launch as a secondary payload on the Space Launch System (SLS) for the Artemis I mission in the summer of 2022.

1.2) The Significance of Mission Operations

Before the purpose of the mission operations architecture, or MOA, is introduced, some familiarity with mission operations from a systems engineering perspective is required. Systems engineering is defined in a multitude of versions dependent upon the systems engineering standards referenced throughout the mission's lifecycle. Here are three versions under the umbrella that define systems engineering:

> The INCOSE Systems Engineering Handbook: "Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. Successful systems must satisfy the need of its customers, users, and stakeholders."[1]
>
> The NASA Systems Engineering Handbook: "at NASA, 'systems engineering' is defined as a methodical, multidisciplinary approach for the design, realization, technical management, operations, and retirement of a system. A 'system' is the combination of elements that function together to produce the capability required to meet a need. The elements include all hardware, software, equipment, facilities, personnel, processes, and procedures, needed for this purpose."[1]
>
> ESA ECSS E-10 Part 1B: "Systems engineering is the interdisciplinary approach governing the total technical effort to transform a requirement into a system solution."[1]

Briefly, it is important to note that, even though each of these definitions originate from a specific standard, all standards are consistent with the overarching purpose of systems engineering: technical management of product and design development (CH, 1). Therefore, selecting a standard depends on the locality and domain of the product being developed. Moreover, certain standards reflect systems engineering from a generic, base perspective, such as the INCOSE standards, while those from NASA, ESA, CSA, ROSCOSMOS, or JAXA are tailored towards space systems or products specific to that country and pull from various standards not limited to systems engineering.

Returning to the (above) previous formal definitions of systems engineering, each highlights an important concept or two that describes the overarching purpose of systems engineering. First, systems engineering is interdisciplinary or multidisciplinary. While it may not be implied from the systems engineering standards, personnel reflect the cross-disciplinary support for a space mission. Every individual hired to support some action or piece of a space mission is not always going to be cut from the same cloth. For example, a mission for a satellite to survey crop growth and soil composition in the Great Plains of the United States requires subject matter experts in agriculture, horticulture, geology, climatology, remote sensing and instrumentation, material science, and engineering to name a few. Therefore, it is imperative to acknowledge that a mission requires not only those who build, but also those who scientifically support the product after realization. In fact, systems engineering could be considered the balancing of all trades and disciplines applied to a space mission.

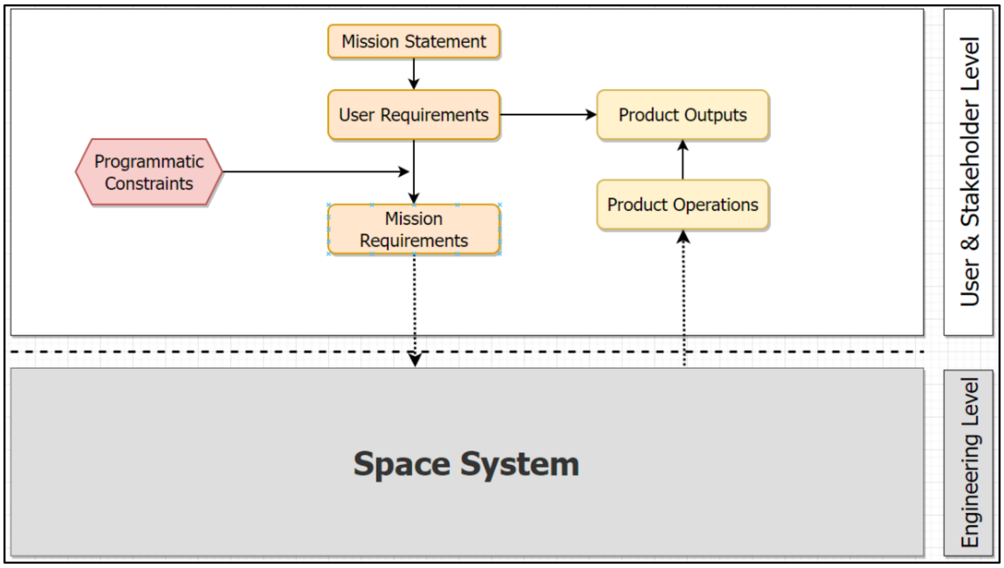


Figure 1-1. The division of a space mission is separated by users and engineers (CH, 12).

Second, systems engineering (or any space mission) involves two levels—user and engineering—with requirements being the contract that binds them. Notice Figure 1-1 that shows the two major divisions in a space mission and their components. On top, the users level includes the stakeholders, the identification of needs and expectations of the mission results, and the outputs of the mission. On the bottom, the engineering level includes where the space system's design, development, and operations originate (CH, 11). Based on the structure of Figure 1-1, the user expects to receive whatever data or products that meet initial expectations. Consequently, the space system is fundamentally a "black box" for the user, where the engineer focuses on its contents and processes (CH, 11). A key process after the user expectations and needs are defined is translating them into mission requirements. A requirement is defined as something that you want or need, or something that you must have to do something else (CH, 11). The requirements at the user and engineer level are seen as an agreement or contract between the users and engineering team. Correct and defect-free definitions on the top-level requirements (user requirements to mission requirements) will allow systems engineers to identify the technical specifications of the space system. Based on these requirements, the design and manufacturing of the system can be carried out, which once integrated, will allow its validation against the requirements, verifying that a correct development has been achieved for operation (CH, 12).

From there, Figure 1-1 can be modified to provide emphasis on the engineering level which leads to the systems engineering V-model as shown in Figure 1-2. This brings up the third concept: systems engineering is a methodical and iterative approach to transforming requirements into a solution. During the design and development of a space system, the process is never perfectly smooth. Most of the time the drivers of the bumpy road to success are space systems not meeting system and/or mission requirements. Major modifications or inclusions of new user or mission requirements, which were initially frozen, might impact the program, and could impose a partial or complete redesign of parts, subsystems, or the entire space system (CH, 12). Additionally, there exist programmatic constraints, such as budgets, schedules, and personnel and staffing limitations that can occur from time to time throughout the development phase. In its entirety, the systems engineering process is generally an iterative one, with reference towards satisfying system and mission requirements.

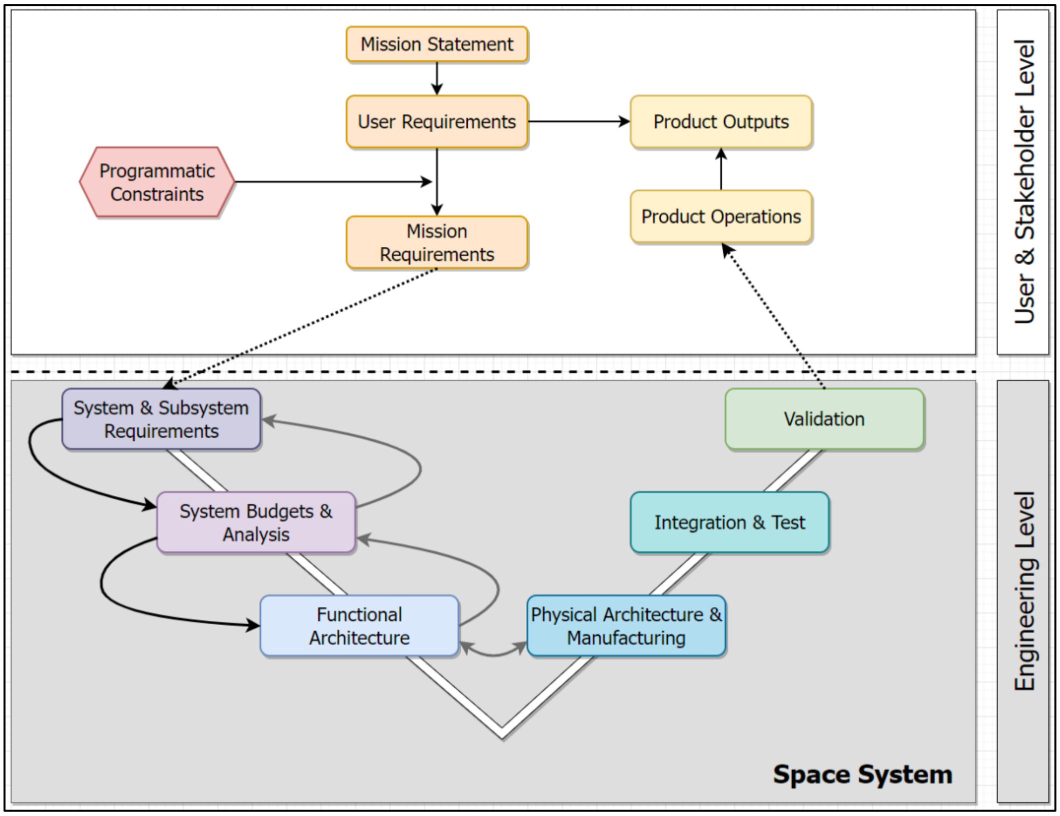
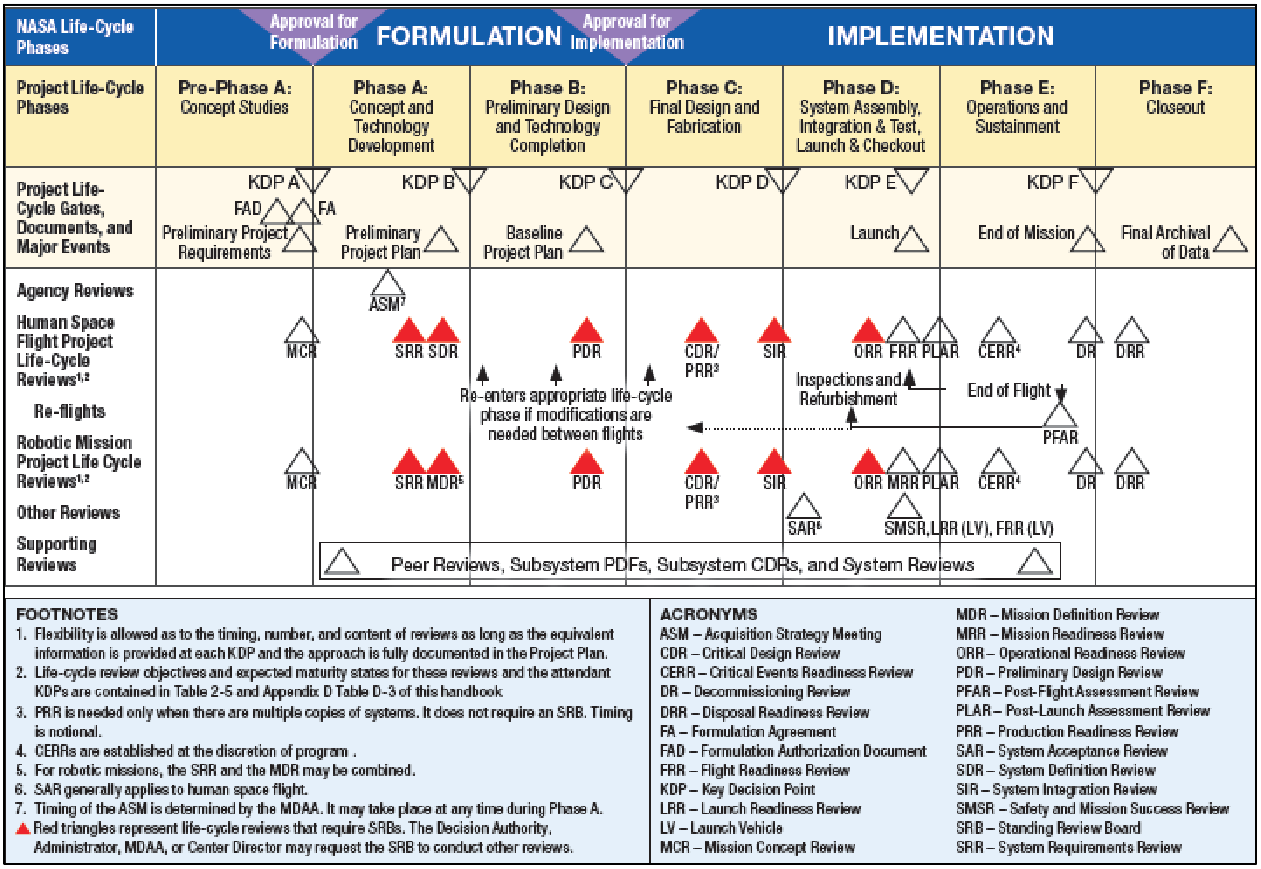


Figure 1-2. The systems engineering V-model (CH, 14).

Since the previous discussion was a brief overview of systems engineering, the assumption here is that the reader(s), team, or organization, has defined their mission and system requirements for their spacecraft. Diving inside systems engineering, there is the program or project lifecycle of the spacecraft, from conception to disposal, which should be a familiar outline for those already involved with space systems. Figure 1-3 gives a snapshot of the NASA program/project lifecycle and Table 1-1 gives the description of each phase throughout the lifecycle. The lifecycle is meant to decompose the entire process into organizable, manageable pieces and provide incremental visibility into the progress being made at points in time.[2] Additionally, from those pieces, or phases as they are mostly known, a programmatic baseline for all deliverables during all phases of the project is defined (CH, 5). Refer to Figure 1-3 for the highlighted deliverables along the lifecycle, and note that depending on the mission and standards, some deliverables may vary. Mission operations is one of those phases (refer to Table 1-1 for all phase descriptions).

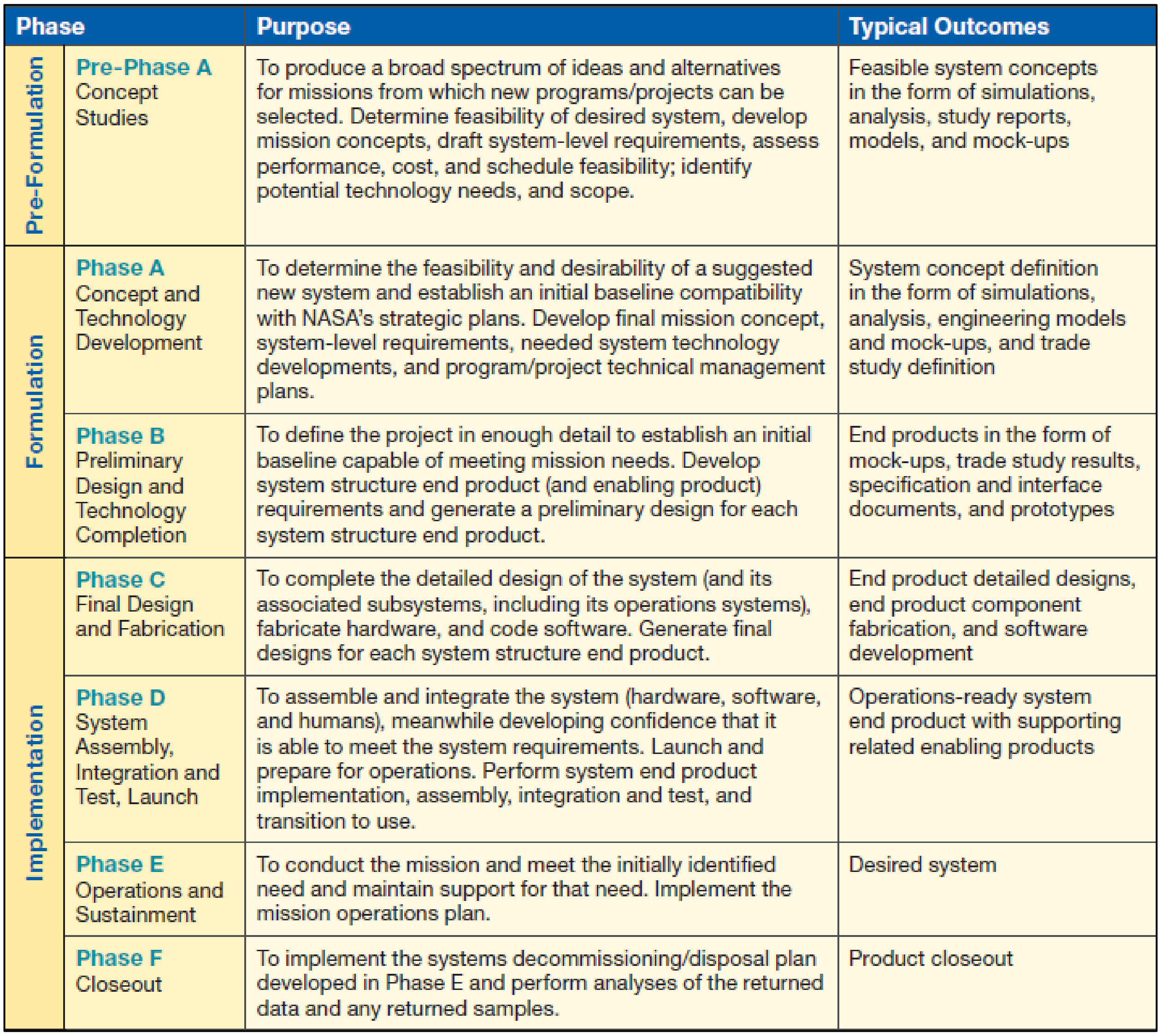


FOOTNOTES
1. Flexibility is allowed as to the timing, number, and content of reviews as long as the equivalent information is provided at each KDP and the approach is fully documented in the Project Plan.
2. Life-cycle review objectives and expected maturity states for these reviews and the attendant KDPs are contained in Table 2-5 and Appendix D Table D-3 of this handbook
3. PRR is needed only when there are multiple copies of systems. It does not require an SRB. Timing is notional.
4. CERRs are established at the discretion of program .
5. For robotic missions, the SRR and the MDR may be combined.
6. SAR generally applies to human space flight.
7. Timing of the ASM is determined by the MDAA. It may take place at any time during Phase A.

▲ Red triangles represent life-cycle reviews that require SRBs. The Decision Authority, Administrator, MDAA, or Center Director may request the SRB to conduct other reviews.

ACRONYMS
ASM – Acquisition Strategy Meeting
CDR – Critical Design Review
CERR – Critical Events Readiness Review
DR – Decommissioning Review
DRR – Disposal Readiness Review
FA – Formulation Agreement
FAD – Formulation Authorization Document
FRR – Flight Readiness Review
KDP – Key Decision Point
LRR – Launch Readiness Review
LV – Launch Vehicle
MCR – Mission Concept Review
MDR – Mission Definition Review
MRR – Mission Readiness Review
ORR – Operational Readiness Review
PDR – Preliminary Design Review
PFAR – Post-Flight Assessment Review
PLAR – Post-Launch Assessment Review
PRR – Production Readiness Review
SAR – System Acceptance Review
SDR – System Definition Review
SIR – System Integration Review
SMSR – Safety and Mission Success Review
SRB – Standing Review Board
SRR – System Requirements Review

Figure 1-3. The NASA Space Flight Project Life Cycle from NPR 7120.5E (NASA SE, 18).



Phase		Purpose	Typical Outcomes
Pre-Formulation	**Pre-Phase A** Concept Studies	To produce a broad spectrum of ideas and alternatives for missions from which new programs/projects can be selected. Determine feasibility of desired system, develop mission concepts, draft system-level requirements, assess performance, cost, and schedule feasibility; identify potential technology needs, and scope.	Feasible system concepts in the form of simulations, analysis, study reports, models, and mock-ups
Formulation	**Phase A** Concept and Technology Development	To determine the feasibility and desirability of a suggested new system and establish an initial baseline compatibility with NASA's strategic plans. Develop final mission concept, system-level requirements, needed system technology developments, and program/project technical management plans.	System concept definition in the form of simulations, analysis, engineering models and mock-ups, and trade study definition
	Phase B Preliminary Design and Technology Completion	To define the project in enough detail to establish an initial baseline capable of meeting mission needs. Develop system structure end product (and enabling product) requirements and generate a preliminary design for each system structure end product.	End products in the form of mock-ups, trade study results, specification and interface documents, and prototypes
Implementation	**Phase C** Final Design and Fabrication	To complete the detailed design of the system (and its associated subsystems, including its operations systems), fabricate hardware, and code software. Generate final designs for each system structure end product.	End product detailed designs, end product component fabrication, and software development
	Phase D System Assembly, Integration and Test, Launch	To assemble and integrate the system (hardware, software, and humans), meanwhile developing confidence that it is able to meet the system requirements. Launch and prepare for operations. Perform system end product implementation, assembly, integration and test, and transition to use.	Operations-ready system end product with supporting related enabling products
	Phase E Operations and Sustainment	To conduct the mission and meet the initially identified need and maintain support for that need. Implement the mission operations plan.	Desired system
	Phase F Closeout	To implement the systems decommissioning/disposal plan developed in Phase E and perform analyses of the returned data and any returned samples.	Product closeout

Table 1-1. NASA program/project life cycle phases (NASA SE, 9).

As defined by the NASA Systems Engineering Handbook, mission operations is the process "to conduct the prime mission to meet the initially identified need and to maintain support for that need" with the products being "the results of the mission and performance of the system" (NASA SE, 31). For a breakdown, the prime mission is defined by the mission statement, which states the subject, users, and the mission operations concept for the spacecraft, while the initially identified need is the requirements set by the users. Recall back to a phrase in the introduction (§1.1): a spacecraft is only as good as the ground systems supporting its operations. This is the sentiment held for the significance of mission operations. The spacecraft, undoubtedly, is the highlight of the mission; however, without support on the ground our use of it equates to nil. Therefore, mission operations serve as the foundation or “ground” for the spacecraft.

Mission operations comprises the entirety of mission planning and management down from the administrative to the technical levels, including all personnel, facilities, equipment, hardware, software, processes, and procedures. More detail on these is provided in the next section (§1.3) and Chapter 3. Table 1-1 defines Phase E as the beginning of mission operations and sustainment of the desired space system, and this is usually consistent with other engineering standards. However, operations do not truly begin at Phase E. It usually begins during Pre-Phase A with the preliminary development of a concept of operations, or ConOps.

A ConOps is produced after the initial user expectations have been established. Usually in the form of a document, this will ensure that the team developing the spacecraft fully understands the expectations and how they may be satisfied by the spacecraft (NASA SE, 50). From there, the team can use the ConOps to define spacecraft requirements and architecture. Since the ConOps is an important driver of requirements, it is considered early in the spacecraft design process. Consequently, it is a stimulant to the development and architecture of other systems and documents supporting the spacecraft, such as the operations plan, the launch and early orbit plan (LEOP), the operations handbook, and provides the foundation for the long-range operational planning activities, such as operational facilities, staffing, and network scheduling (NASA SE, 50). The ConOps also prevents pitfalls in matching systems requirements with mission requirements and cases where requirements and design functions might be overlooked. Most importantly, the ConOps describes scenarios for all significant operational situations, including nominal and off-nominal (critical) operations during integration, test, and launch through disposal (Phases D to F)

(NASA SE, 50). Here are some more aspects considered in the ConOps not already mentioned that are touched upon throughout the lifecycle:

- Description of major phases
- Operational timelines
- Fault management strategies
- Description of human interaction and required training
- End-to-end communications strategy
- Command and data architecture
- Operational facilities
- Integrated logistic support (resupply, maintenance, and assembly)
- Staffing levels and required skill sets

From the introduction of ConOps, there is a discrepancy worth mentioning. The ConOps should not be confused with the operations concept. The NASA Systems Engineering Handbook defines the following for a ConOps[*] and operations concept[†], respectfully:

> "Developed early in Pre-Phase A by the technical team, [a ConOps] describes the overall high-level concept of how the system will be used to meet stakeholder expectations, usually in a time sequenced manner. It describes the system from an operational perspective and helps facilitate an understanding of the system goals. It stimulates the development of the requirements and architecture related to the user elements of the system. It serves as the basis for subsequent definition documents and provides the foundation for the long-range operational planning activities" (NASA SE, 51).[*]
>
> "A description of how the flight system and the ground system are used together to ensure that the concept of operation is reasonable. This might include how mission data of interest, such as engineering or scientific data, are captured, returned to Earth, processed, made available to users, and archived for future reference. It is typically developed by the operational team" (NASA SE, 51).[†]

Both are necessary; however, the main point addressed is that the operations concept serves as a case of verification and validation (V&V) for the ConOps developed and refined from pre-operations in Pre-Phase A to Phase D. Phase E is where the operations concept is initiated and is ultimately a detailed, low-level account of the interactions between the spacecraft and the ground.

In summary, mission operations is a routine of mission and spacecraft sustainment that becomes natural as the team completes operational readiness training, implements the operations concept backed by the ConOps, and the "day in the life" of the spacecraft is established from nominal operations. Formally, the program must pass an operational readiness review, or ORR, before launch to verify and validate the ConOps and operations concept in action. ORR will be revisited in Chapter 4 as a form of V&V for Lunar IceCube.

1.3) The Purpose of an Architecture

Since mission operations has been defined in the proper scope of systems engineering, next is to a MOA. First, reference back to the definition of systems engineering provided by the NASA Systems Engineering Handbook. Here, the definition mentions elements: "hardware, software, facilities, personnel, processes, and procedures" (CH, 1). These are the pillars that define a MOA, or all the infrastructure created to support the spacecraft from launch to decommission (disposal). For instance, consider a blueprint for a building. The blueprint serves as a schematic for the architecture that will serve that building's intent and aims to provide multiple levels of detail, from plumbing to structural integrity. Keeping that mindset, a MOA serves as the blueprint for a purposeful spacecraft with multiple levels and sublevels pertaining to the aspects of spacecraft and mission requirements. If blueprints are intended to provide the means of describing, designing, and eventually defining a building, then different buildings require different blueprints. The equivalent can be stated for a spacecraft. Depending on the mission objectives and spacecraft requirements, one spacecraft will require a different MOA then another spacecraft.

Before a list of the basic elements of a MOA is provided, the reader should understand the underlying principle that will unify these elements to create a cohesive system for an effective, efficient, and substantial mission. Just as mentioned in the previous section (§1.2), mission operations is the center of program or project management. For a MOA, that is comparable to data management. A MOA serves as the medium of data from spacecraft to end-users, and the management of such an asset is crucial for any mission to satisfy the mission objectives. No data mean no results, and no results means unhappy stakeholder and a waste of funding. A team or organizational body wants to avoid such a debacle; therefore, managing data should come down to a science, essentially a formal process of mapping out the entire data flow from spacecraft to end-users. Therefore, a MOA is present for the affirmation of a proper data flow.

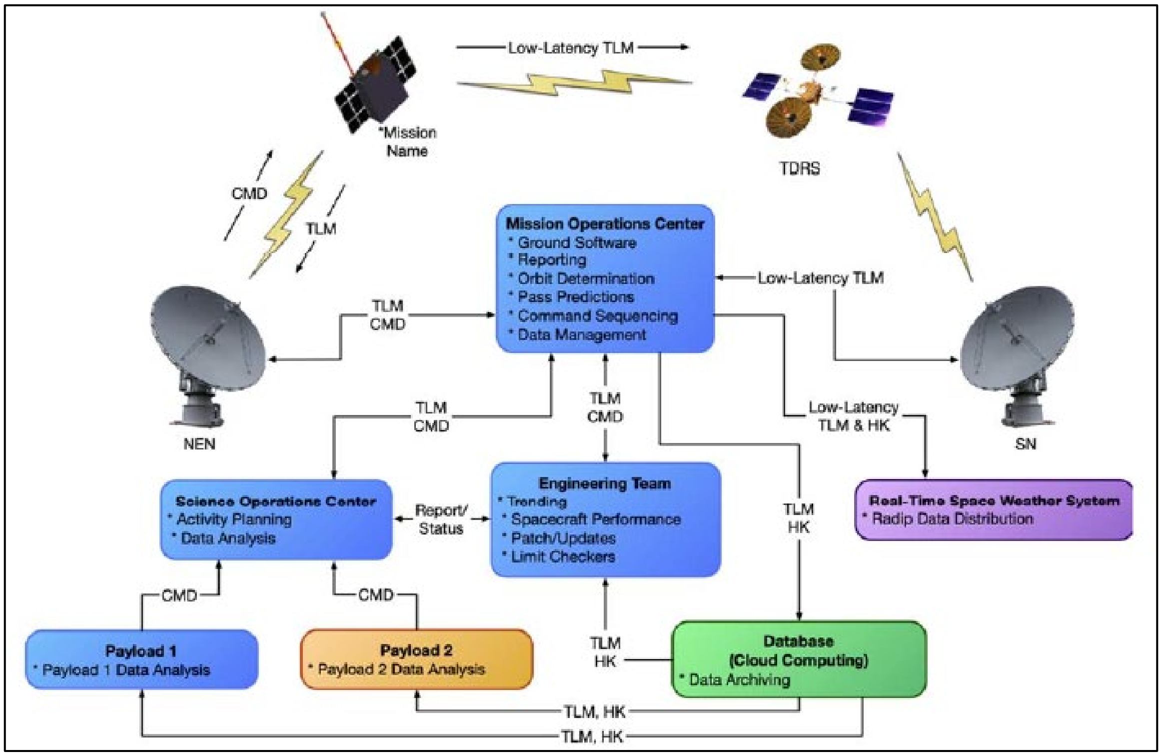


Figure 1-4. Example of a MOA for a small satellite using NASA's Near Earth Network and Space Network (SOA, 219).

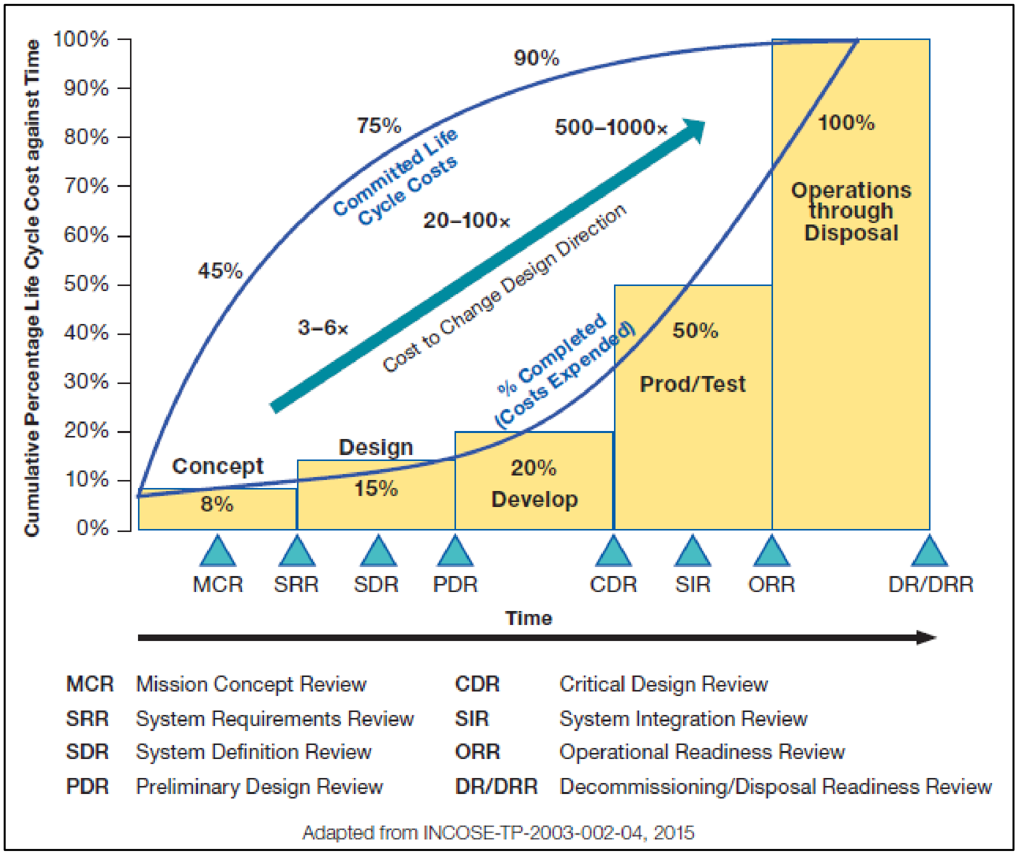
With the idea of data flowing down from spacecraft to end-users intact, the elements of a MOA can be defined. There are two types: "macro" and "micro" elements. Micro-elements are the individual hardware, software, and processes within a system. For example, a transceiver for a ground station; a program that copies incoming telemetry into two repositories—maybe one for testing or one for archiving; or a program that parses telemetry and separates it based on subsystem or housekeeping type. Macro-elements consist of multiple micro-elements that create a standalone, individual system, e.g., a ground station. The following are the macro-elements considered for a MOA: a ground systems architecture (GSA) consisting of either or both a ground station network (GSN) or a single ground station; a ground data system (GDS) for data staging; a mission operations center (MOC); a science operations center (SOC); and external-user facilities (EUFs). An example of a MOA with a spacecraft is considered in Figure 1-4. Note that the term "ground systems architecture" can be used in the same context as a mission operations architecture in other sources.[3] However, while all these macro-elements are considered "on the ground" with respect to the spacecraft, the purpose here is to classify all from a mission operations perspective only on the ground with the spacecraft omitted—the MOA assumes the spacecraft is fully operational. Therefore, the ground systems architecture is specific only to communications with the spacecraft being mentioned when necessary. The dissemination and practice of all MOA elements is presented in Chapter 3 with Lunar IceCube, and their V&V is presented in Chapter 4.

Recall from earlier that different spacecraft, e.g., a LEO-based mission verses a deep-space mission, will require different architectures. Despite the contrasting objectives, the elements of a MOA are streamlined across various mission platforms; hence, they are mostly unchanging up to the integration of subsequent technology. The difference in architectures can be traced to a combination of the differences in mission scope, cost, performance, and risk. While mission scope is determined based on the mission statement and objectives, it is relatively straightforward to determine the differences in assets required for a LEO-based mission verses a deep-space mission. Therefore, cost, performance, and risk are the biggest drivers, and trade studies are performed to localize the designs with the best combinations of each. Nevertheless, there will be tradeoffs. For each cost-effective solution, one of four outcomes arise (NASA SE, 12):

1. To reduce cost at constant risk, performance must be reduced
2. To reduce risk at constant cost, performance must be reduced.

3. To reduce cost at constant performance, higher risks must be accepted
4. To reduce risk at constant performance, higher costs must be accepted

What this describes is that the “best” design for a MOA could contradict the mission objectives and spacecraft requirements—considering programmatic constraints. Therefore, the “best” is implausible to implement.

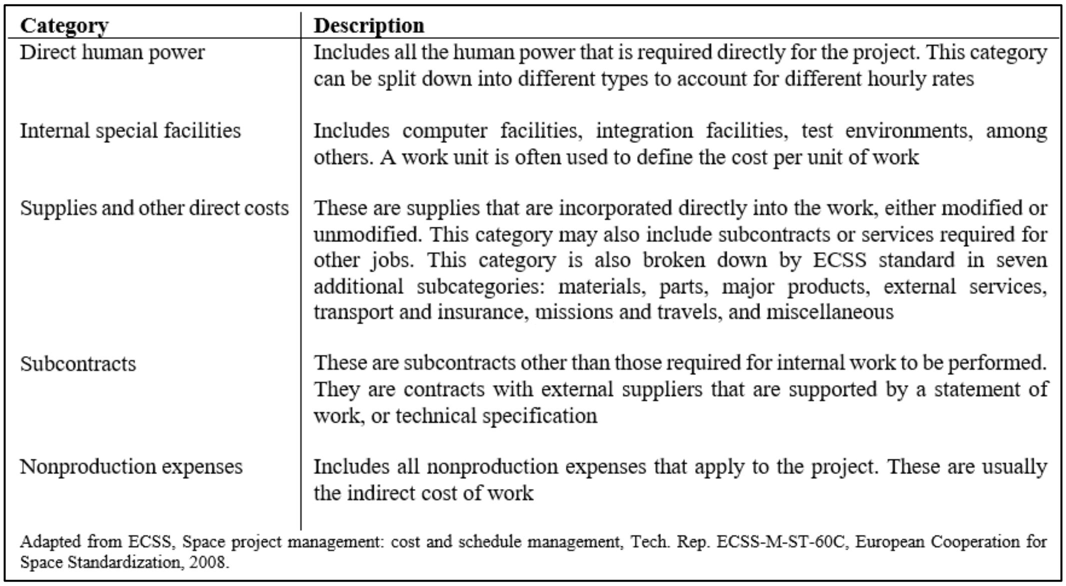


Adapted from INCOSE-TP-2003-002-04, 2015

Figure 1-5. Lifecycle cost impacts from early phase decision-making (NASA SE, 9).

Expanding on the driver of cost, commonly, costs in the lifecycle of a program or project tend to get “locked in” early in design and development—around Phase A to Phase B (NASA SE, 12). Notice Figure 1-5, which shows the lifecycle cost impacts from early phase decision-making for any space system or architecture with the concurrent deliverables. Significance of the iterative systems engineering model comes to light when considering that late identification or fixes to problems of any portion of a system or architecture incur more costs later in the lifecycle (NASA

SE, 12). Conversely, a descope of a system or architecture implemented later verses earlier results in a reduction of cost savings (less costs). Figure E shows four plots: the committed lifecycle costs, the percentage of program/project completion as a representation of costs expended, the cumulative percentage of life cycle costs against time, and the cost to change design direction. As mentioned, changes late in lifecycle incur more costs, and this is represented with an extreme cost multiplier at a 50% completion past the CDR. The cause being the significant increase of committed lifecycle costs occurring at only 15% completion. Therefore, the costs are heavily stacked against program management if changes are necessary later.



Category	Description
Direct human power	Includes all the human power that is required directly for the project. This category can be split down into different types to account for different hourly rates
Internal special facilities	Includes computer facilities, integration facilities, test environments, among others. A work unit is often used to define the cost per unit of work
Supplies and other direct costs	These are supplies that are incorporated directly into the work, either modified or unmodified. This category may also include subcontracts or services required for other jobs. This category is also broken down by ECSS standard in seven additional subcategories: materials, parts, major products, external services, transport and insurance, missions and travels, and miscellaneous
Subcontracts	These are subcontracts other than those required for internal work to be performed. They are contracts with external suppliers that are supported by a statement of work, or technical specification
Nonproduction expenses	Includes all nonproduction expenses that apply to the project. These are usually the indirect cost of work

Adapted from ECSS, Space project management: cost and schedule management, Tech. Rep. ECSS-M-ST-60C, European Cooperation for Space Standardization, 2008.

Table 1-2. CBS top-level categories following the ECSS-M-ST-60C standard used by ESA (CH, 16).

To avoid such derailments in cost, it is important to create a cost breakdown structure of the MOA, or CBS (CH, 16). A MOA is massive in physical and logistical infrastructure, and it could be massive in personnel. Since operations require expertise, training, and planning, personnel are crucial even if they are an indirect element of a MOA. The top-level categories of a CBS often include human resources, facilities, external contracts, indirect costs, material and supplies, transportation, and mission operations, among others (CH, 16). For an example, the ESA CBS top-level categories are presented in Table 1-2. After defining the CBS, a cost estimate for each category of the CBS is required. Since space projects are complex in nature, there exists uncertainties for scope and schedule in the early phases of the lifecycle. Depending on the phase where the cost estimation is made, a margin of uncertainty should be included to account for possible variances. Uncertainty margins used are subject to the area of expertise on which the program/project is developed (CH, 17). The estimated costs rule-of-thumb for space-related projects by phase are expressed as follows (CH, 18):

- Pre-Phase A: -20% (to) +40%
- Phases A and B: -15% (to) +35%
- Phase C: -10% (to) +15%
- Phase D and subsequent: -5% (to) +10%

As maturity of the project progresses, the accuracy of the estimates will progress. Therefore, document and communicate the uncertainties associated with any estimate made on a category. Still, these uncertainties may not be readily available. As stated by the NASA Cost Estimating Handbook, areas of uncertainties can be “pending negotiations; concurrency; schedule risk; performance requirements that are not yet firm; appropriateness of analogies; the level of knowledge about support concepts; critical assumptions; among others” (CH, 18). Note that the cost estimation of each category of the CBS is the first step in conducting a mission cost analysis. For purposes here, this is not within the scope of this writing; therefore, for more detail to a mission cost analysis refer to [1].

Expanding briefly on the driver of risk, acceptable risk is achieved with redundant simple spacecraft that imply a lower cost achieved through focus.[4] Simplicity in the design of a spacecraft can reduce the pressure of risk in the development, launch, and operations of a spacecraft. The concept of redundancy assists simplicity by providing mass manufactured, marketed, capable spacecraft designs that are effective and efficient without the intrusion of complexity. While there may be practical limitations for the simplicity of a spacecraft, the cost and performance can subsequently favor the program. Therefore, simplicity becomes a standard which ultimately provides a limited scope; however, completing a program within a defined scope is attainable when proper considerations are accounted for.

Overall, the reader should have garnered the significance of a MOA by the elements that comprise it: as the hub for data management and the medium for data flow, and the implications and dispositions of tradeoffs among cost, performance, and risk. Up to this point, most of the discussion has revolved around background to setup for a deeper dive into the MOA. However, before this deeper dive, the scope of the MOA needs to be introduced and expanded for more insight as to the purpose for these proceedings. The scope for this MOA is small spacecraft.

1.4) Sizing-up Small Spacecraft

1.4.1) *The Concepts of Small*

Small spacecraft have been situated in the aerospace arena since their conception in the late 1990s, but without significant impact. At the time, this was a result of their limited use, lack of scalable technology and manufacturing (particularly in payload sensors), and disinterest from industry. However, over the past decade, this has overwhelmingly changed. Recently, there has been a boom of small spacecraft missions, with a rare few going interplanetary. This is thanks in part to national and international space organizations and agencies, e.g., NASA's CubeSat Launch Initiative, the National Science Foundation, and ESA's Fly Your Satellite Initiative; more launch opportunities from both public and private corporations, like SpaceX, United Launch Alliance, and Northrop Grumman; also, resources resulting from effective and efficient manufacturing or open software development, leading to reliable, cheaper, and accessible parts, equipment, and software that are integration and flight-ready out-of-the-box. There exist numerous drivers that propelled interest in small spacecraft, but the most popular driver is quite simple in scope and lends to the notion that small spacecraft are just the tipping point of something subsequently greater. Essentially, providing a low-cost mission to make more accessible science to allow questions to be answered, without the cost and effort of standing up to large, flagship spacecraft.[4]

Keeping that driver in mind, small spacecraft have a common set of properties unlike that of large spacecraft. First, the objective of a small spacecraft is narrow and to observe a unique phenomenon.[4] Small spacecraft can be limited in their ability to perform an excessive number of actions and tasks while in operations, and this is usually a result of scale and priority rather than effectiveness and efficiency—this will be expanded upon in §1.4.2. Second, the objective is to acquire multiple vantage points, e.g., in time, space, or frequency.[4] The selection of payload integrated on the spacecraft will set constraints on the number of vantage points allocated throughout operations. Third, objectives are evolving, and the desire is to build upon previous measurements. Moreover, the objectives can be ephemeral, with limited access opportunities.[4] Therefore, the playbook of operations for small spacecraft is usually short when the objectives are met. It is also a plus to the playbook if the spacecraft can confirm hypotheses stated before operations or receive an extension of mission for a new objective.

1.4.2) *The Considerations of Small*

As mentioned in the introduction (§1.1), small spacecraft are considered "Davids" amongst the large, flagship spacecraft. Since the spacecraft's inception in the late 1950s and early 1960s, architectures, infrastructure, and system lifecycles are systemically specific to large spacecraft, with a good portion geared towards manned spacecraft. Therefore, small spacecraft are justifiably underrepresented by the current paradigm of spacecraft design and development. First, large spacecraft have not only size and scale, but also the proper funding, resources, and manpower readily accessible at any given time during a mission. Depending on the funding sponsors and stakeholders, large spacecraft have access to world-class ground station networks, e.g., NASA’s Near Earth (NEN) and Deep Space (DSN) Networks; testing facilities; subject-matter expert inquiries and manpower; and a large overhead in terms of schedule, cost, performance, and risk for requirements.

Since small spacecraft are narrowly scoped from conception, such an architecture is inadequate. Most likely, small spacecraft only have one payload for the prime mission objective unlike that for large spacecraft with multiple payloads. Requirements should be far less in number, and overhead should be lower. However, small spacecraft could be sponsored by national, international, and corporate entities that impose a large spacecraft architecture onto a small spacecraft-based mission. At that point, small spacecraft are in somewhat of a battleground with the large overhead, schedule, priority, cost, performance, and risk imbued into the program by the sponsor. This battleground is usually the result of constrained funding, a lack of resources and equipment, less priority on the part of the sponsor, and less manpower. Therefore, small spacecraft must navigate carefully throughout the process of design and development if the program is to stay afloat and succeed. It is important to understand that these are normal conditions for the design and development for small spacecraft in the aerospace industry. However, after several years of investment and interest, small spacecraft have been launching by the hundreds. The reason: small spacecraft are “custom-flexible.”

First, small spacecraft are “custom” in the sense that they are unique to the mission, bringing quality specifically for that prime mission objective. Large spacecraft are also custom and yield quality but can imply that customization yields complexity and quantity from the multitude of requirements and objectives. Therefore, it can be implied that quality can be hindered when resources are stretched thin. Second, small spacecraft are “flexible” in the sense that programmatic constraints can be selective based on requirements chosen. Meaning, depending on the requirements of the spacecraft and the prime objective of the mission, programs can pick-and-choose the appropriate milestones from their sponsor’s program/project lifecycle. However, some milestones from the sponsor are assumed to be required regardless of the choice of the program, e.g., PDR, CDR, FRR, and ORR.

Touching back on requirements, if small spacecraft are custom-flexible, what requirements are imposed by the sponsor, and how do they effect the prime mission objective and initial spacecraft requirements? Unfortunately, a program will not know all the requirements to adhere to even before the designing of a spacecraft.[5] However, a decent place to begin are the public documents provided by the sponsors and launch providers. Specifically, interface control documents (ICDs). ICDs set the rulebook for small spacecraft and officially state what requirements are to be adhered for a particular launch (CS101, 40). Therefore, the ICDs set the programmatic constraints for a small spacecraft’s requirements. As far as numbers, there might be a handful of ICDs. For example, depending on the type of small spacecraft, e.g., CubeSat, there are specific design specifications; if the spacecraft requires a dispenser, there are dispenser standards and specifications; the launch service provider will provide program level requirements as an ICD for integration with the launch vehicle; and the sponsor will usually provide the mission specific ICD that entails a combination of lower level ICDs, such as the small spacecraft design specifications, the satellite-to-dispenser ICD, the dispenser-to-LV (launch vehicle) ICD, and range safety requirements (CS101, 40). Additionally, if launches are from U.S. soil, small spacecraft must adhere to federal statutes set by the FCC and NOAA, or if internationally, they must adhere to the standards of the host country or conglomerate (CS101, 41).

Launch paradigms are considerations for small spacecraft that usually set the ICDs provided by the sponsor. Since launch vehicle capability usually exceeds primary customer requirements, there is typically mass, volume, and other performance margins to consider for the inclusion of a secondary small spacecraft. Consequently, small spacecraft can exploit this surplus capacity for a relatively inexpensive ride to space. With the increase in small spacecraft demand over the past decade, the launch vehicle market has seen a spike in the creation and advertisement of platforms centered around small spacecraft (SOA, 205). There are three launch paradigms of choice considered for small spacecraft: dedicated, traditional rideshare, and dedicated rideshare. Dedicated launches are where small spacecraft control the mission requirements in whole: what they need, when they want to launch, and where they want to go. Currently, dedicated launchers on the market advertise services to small spacecraft having a mass exceeding 180 kg. Additionally, the most common orbit for dedicated launches is LEO, with very few pushing for HEO, MEO, and GEO. The only downside, when considering the market, is that dedicated launches are expensive (SOA, 205).

Figure 1-6. An Atlas V launching LRO (primary) and LCROSS (secondary). Image by Pat Corkery, United Launch Alliance, "LRO and LCROSS Launch on Lunar Journey," June 18, 2009, https://www.nasa.gov/multimedia/imagegallery/image_feature_1392.html.

Therefore, a majority of small spacecraft are secondary payloads in line with the traditional rideshare paradigm, such as the one presented by the Atlas V with LRO and LCROSS in Figure 1-6. Standard ridesharing consists of a primary mission with surplus mass, volume, and performance margins which are used by the small spacecraft. From the secondary spacecraft designers' perspective, rideshare arrangements provide far more options for immediate launch with demonstrated launch vehicles. Since almost any large launcher can fit a small payload within its mass and volume margins, there is no shortage of options for small spacecraft that envision to fly as secondary spacecraft. The downsides are less control with a majority being lent to the primary spacecraft, such as launch date and trajectory; integration and delivery usually occurs weeks before launch; and secondary spacecraft usually deploy after successful deployment by the primary spacecraft from the launch vehicle (SOA, 205). The third paradigm is dedicated rideshare launches, which use launch vehicles to exclusively launch multiple small spacecraft at once. While quantity is the main advertisement and opportunity here, this paradigm is rarely chosen because of the difficulties in the logistics of managing many small spacecraft with unique visions and needs (SOA, 206).

Another common consideration for small spacecraft is deployment methods that usually depend on the form factor of the spacecraft. There are two types of deployers, CubeSat dispensers and SmallSat dispensers. CubeSat dispensers support CubeSats up to 24 kg in a 12U configuration but can be extended to ~54 kg in a 27U configuration. Considered the most widely manufactured configuration for small spacecraft, it is a container-based integration system that serves as the interface between the CubeSat and the launch vehicle, and it comprises of rectangular box with a hinged door that is opened when commanded, rails for stability and ease of deployment, and a spring deployment mechanism (SOA, 206-207). SmallSat dispensers are manufactured for small spacecraft that do not adhere to the CubeSat standard and require a different separation mechanism. Circular dispensers consist of two rings held together by a clamping mechanism, where one ring is attached to the launch vehicle and the other is attached to the spacecraft. Once the clamping mechanism is released, the two rings separate and are pushed apart by the springs, and each ring then remains with the spacecraft or launch vehicle after deployment (SOA, 208). Another common method of deployment is from ISS, such as the Nanoracks CubeSat Deployer (NRCSD) in Figure 1-7. The NRSCD is a self-contained CubeSat dispenser system that mechanically and electrically isolates CubeSats from the ISS, cargo resupply, and ISS crew. The dispenser consists of a rectangular tube of anodized aluminum plates, base plate assembly, access panels, and deployer doors. Inside, the walls are smooth bore design to minimize and/or preclude hang-up or jamming of CubeSat appendages during deployment, should they become released prematurely (SOA, 211).

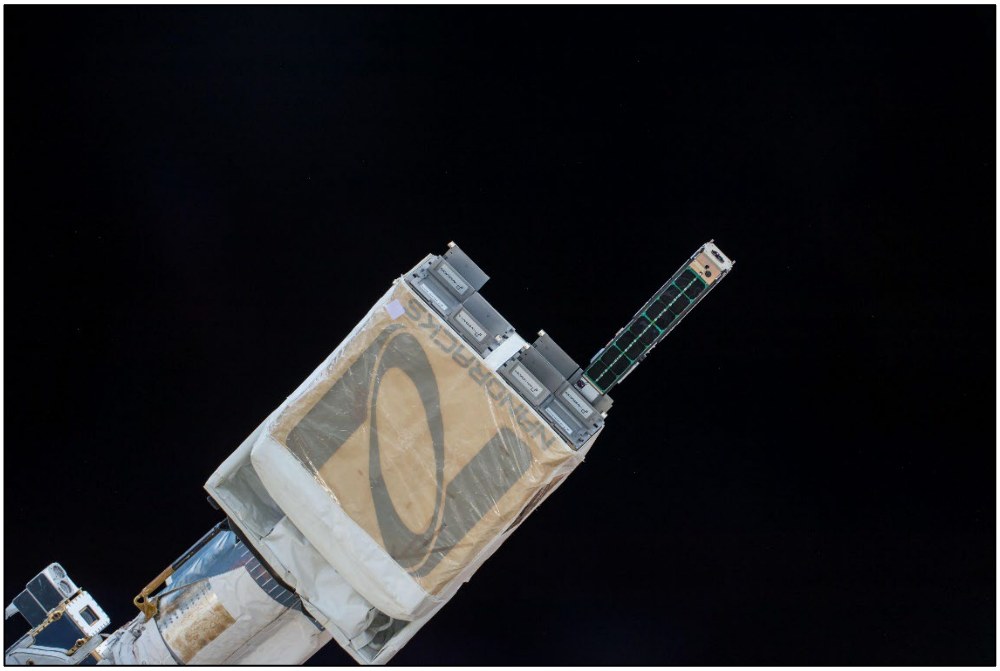


Figure 1-7. The Nanoracks CubeSat Deployer releasing the SHARC (Spacecraft for High Accuracy Radar Calibration) microsatellite into orbit from ISS. Image by NASA. https://www.nasa.gov/mission_pages/station/research/experiments/explorer/Investigation.html?#id=1196.

1.4.3) *The CubeSat*

The CubeSat is the most popular and adopted form factor of any small spacecraft in the aerospace industry worldwide. With a simplistic design boasting its compatibility and expandability to various applications, accessibility to resources and mass-manufactured COTS (commercial-off-the-shelf) parts, and affordability from managerial and programmatic perspectives, it is no longer the toy "Hot Wheel type satellite" or "dumb-idea" as the aerospace industry described in the past (CH, pg. xxv). The foundation of the CubeSat began in 1998 from a project between Stanford University's Aeronautics and Astronautics Department, the Aerospace Corporation, and the Defense Advanced Research Projects Agency (DARPA). Aerospace wanted to launch a picosat the size of a Klondike ice cream bar for the DARPA program. They ultimately requested the help from Stanford. Since 1995, the students at Stanford were already developing picosats. This led to the design of a microsatellite deployer that allowed multiple picosats to launch at one time. Eventually, Orbiting Picosat Automated Launchers, or OPAL was launched under DARPA in 2000 and demonstrated the feasibility for students to work on smaller satellites (CH, xxi). Thereafter, a cancelation of a Russian launch prompted more investigation into a new design, and the concept to design a simplistic picosat never ceased. The CubeSat standard was determined soon after and classified as a 10-cm cube with a total mass of 1 kg (CH, pg. xxii).

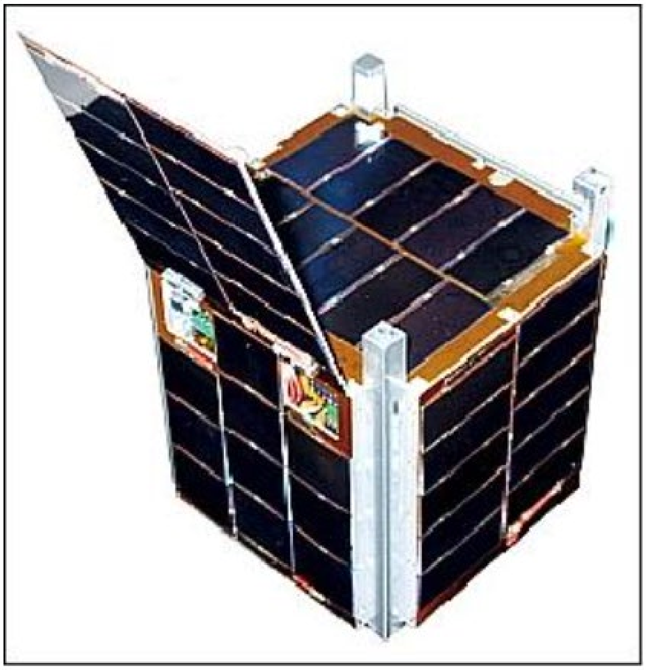


Figure 1-8. CUTE-I in its deployed configuration. Image by Tokyo Institute of Technology. https://directory.eoportal.org/web/eoportal/satellite-missions/c-missions/cute-i.

The first CubeSats launched in June 2003 from Plesetsk, Russia, on a Eurokot, utilizing the Russian Multiple Mission Orbit Service. Cost for the launch of a 1U was set at $30,000 each. Those that launched included the DTUSat built by the Danish Technical University, the Japanese XI-IV and CUTE-1 (Figure 1-8), the Canadian Can X-1, and the Cal Poly led Quakesat in collaboration with Quake Finder from Palo Alto, CA (CH, pg. xxiii). While the Russians were strapped for funds during this time, the launches provided were the only ones of their kind that allowed CubeSats. Unfortunately, the U.S. aerospace industry had no interest in CubeSats at the time. It was not until 2008 when the NSF started noticing the scientific benefit of CubeSats that the aerospace industry started taking notice with NASA following close behind in 2010 to join in the frenzy of this new endeavor (CH, pg. xxv).

There exist several key driving factors behind the worldwide adoption and growth of CubeSats. First, is their accessibility. Their original intention was to be an educational tool for university student teams; with this platform universities were able to afford the financial costs, development time, and expertise to design, launch, and operate a satellite over the course of a student's degree program. These simplicity and cost-effective factors were achieved by creating a simplified design. Ultimately, by using recommended affordable COTS components and accepted specifications and requirements, the design streamlined various stages of the development cycle such as deployment, structural design, and some verification requirements (CH, 53).

Second, is their standardization. The incidental industry came about over time thanks to CubeSats design specifications such as structural dimensions, deployment mechanisms, and the "stackability" of units. Within the boundary conditions of the design specifications, a few engineering solutions became standout options and very quickly became industry standard—such as the PC/104 form factor for electronics and the PPOD common deployment mechanism. These enabled the further streamlining of the development process and mass production of parts. Entire subsystems eventually became COTS products for CubeSats, further reducing costs to developers (CH, 53).

Third, is their entrepreneurship. Over time the result of these net forces led to CubeSats becoming high return-on-investment space platforms, with low start-up or "buy-in" costs. This is what enables CubeSats today to fulfill such a wide variety of roles in such a wide variety of mission scenarios, from a wide variety of developers and customers (CH, 53). Fourth, their technology. Advances in technology have allowed for more miniaturized, distributed autonomous, and higher-performing systems to be flown and more of this flight hardware to become COTS available products (CH, 53). Fifth and final, is the community they influence. This spread of experience, technology, facilities, and industry has helped create a boom in services and customers and one that may not have evolved from traditional aerospace industries or practices. Opportunities for customers from less wealthy of less space-invested countries, industries, or research fields have broadened the scope of experiments and objectives that can be accomplished in space using a CubeSat. A plethora of smaller, more novel, and enterprising companies have formed to fill this niche to support such customers interested in taking their payloads to orbit, regardless of the stage of development (CH, 53).

Because of the CubeSat's engineered design and increasing commonplace use in the space sector, it has become a highly versatile platform with a multitude of applications. First, it is an alternative platform for space-borne instruments that enhance traditional satellite roles in telecommunications, observations of the Earth, and astronomical targets (CH, 54). Second, it is an alternative platform for space-borne experiments and instruments giving life sciences, pharmacology, materials science, radiology, and other fields access to space as an environment (CH, 54). Third, it is an affordable technology-demonstrator and proof-of-concept platform due to the low cost of development and he possibility of using existing components. CubeSats have found a natural application as in-orbit demonstrators for several kinds of new technologies and for demonstrating new mission concepts (CH, 54). Fourth, it is a payload outside of Earth's orbit to provide additional measurements or support for mothership spacecraft or to place instruments in advantageous orbits that would otherwise require dedicated launches (CH, 54). Fifth and final, it is a realization of mega constellations at affordable prices. The use of universal design specifications and evolved industry standards have encouraged CubeSat components and sometimes entire spacecraft to be easier and cheaper to mass produce, significantly reducing costs in producing many for a constellation. The more spacecraft in a constellation, the more coverage is guaranteed, the greater the redundancy is bulti-in, and the shorter revisit time is available (CH, 54).

Aside from the various applications, CubeSats can also be represented as distributed CubeSat systems, or DCS, where the mission functionality is allocated across multiple flight units. DCSs enable new mission capabilities and enhanced mission performance and assurance levels that would be more complex or even not possible to achieve by an individual CubeSat (CH, 123). Constellations were mentioned previously and allow high levels of homogeneity, size of fleet, and spatial separation at the expense of functional interdependence and operational independence breaking from the traditional "monolithic" systems (CH, 125). Clusters, unlike constellations, possess a lower number of identical flight units—typically two to six—that fly at low along-track distance between each other while being mutually dependent from a functional and operational perspective to deliver on their mission objectives (CH, 127). CubeSat series are concepts where identical, or quasi-identical units are flown, usually for technology demonstration purposes or "proof-of-concept" missions such as intersatellite link or formation flight technology (CH, 127). Swarms are closely tied to the concept of clusters with the only difference involving a larger number of flight units that are functionally noninterdependent from each other. Swarms offer the advantage brought in by large numbers of spacecraft, where in mission performance can be designed to degrade more gracefully in case of failure of an individual flight unit. The main different between swarms are constellations is that swarms are interconnected. Large constellations of satellites, like SpaceX's Starlink exist, but swarms have not been implemented. The typical application of swarms is of a scientific nature. As they entail large numbers, swarms bring significant technology challenges in terms of spacecraft in-orbit identification and discrimination after deployment, coordination, and relative navigation (CH, 128). Presently, CubeSats have yet to been implemented in a mission profile consisting of constellations or swarms. Lastly, trains are DCS of different spacecraft, flying in coordinated orbits while performing independent operations. CubeSat trains have not yet appeared in orbit, but they are likely to emerge as a natural evolution of the small satellite industry (CH, 128).

This first chapter was intended to present the overall concept of a MOA and why it is a significant consideration for any space mission. Specifically, a mission for a small spacecraft. Next, small spacecraft were introduced as a concept and important considerations were discussed for overall mission success. Last, the CubeSat was introduced as the design of choice for small spacecraft and how that design upended the entire aerospace industry worldwide for the betterment of access to space. Before diving into deeper waters defining all the elements of a MOA for small spacecraft, it is important to reflect on the MOAs that propelled missions from the origins of manned spaceflight to modern trailblazers.

2) MOA Origins, Precedents, and Trailblazers

To understand the modern concepts and practices for a MOA in the spacecraft and aerospace industry, it is important to reflect upon the origins and precedents set by legacy missions. Project Mercury in the next section will serve as the example of the beginnings of the MOA, and how humanity's first attempt at developing a manned spacecraft was also a first for supporting such a spacecraft from the ground. Since the MOA is the main focal point of this writing, elements of the Project Mercury MOA will be explored. Note that these elements to date represent an antiquated architecture for spacecraft but serves the purpose of how a MOA was first established some six decades ago. After Project Mercury is explored, modern precedents of MOAs will be introduced for deep space small spacecraft, such as the Lunar Reconnaissance Orbiter (LRO), the Lunar Crater Observation and Sensing Satellite (LCROSS), and the Mars Cube One (MarCO) CubeSats. For example, all three cases can serve as the subjects of a trade study that cross-compares MOAs and their specific design and development characteristics.

2.1) The Legacy of the Large: Project Mercury

Project Mercury was the United States' first manned spaceflight project. Planning for the project began shortly before the establishment of NASA on October 1, 1958, by a joint committee of the former National Advisory Committee for Astronautics (NACA) and Advanced Research Project Agency (ARPA). The project was approved shortly after by Dr. T. Keith Glennan, the first administrator of NASA, on October 7, 1958.[6] The Space Task Group (STG)—later the Manned Spacecraft Center (MSC)—was informally organized after this assignment to initiate the action for project accomplishment. The overall management of the program was the responsibility of NASA Headquarters, with project management the responsibility of the STG (Mercury, 16). Since this was an entirely new endeavor for industry and the nation, a joint statement from Mercury Project Manager, Kenneth S. Kleinknecht; Deputy Project Manager, William M. Bland, Jr.; Chief of Engineering Operations, James E. Bost; and Deputy Director for Mission Requirements and Flight Operations, Walter C. Williams described the importance, risk, and complexity at the time:

> "It was recognized from the beginning that this had to be a joint effort of all concerned, and as such, the best knowledge and experience as related to all phases of the program and the cooperation of all personnel was required if success was to be achieved. It was also recognized that it was an extremely complex program that would probably involve more elements of government and industry than any development program undertaken" (Mercury, 16).

With these sentiments in mind, the following are the objectives, guidelines, and spacecraft requirements for the Mercury Project (Mercury, 2):

Objectives:

1. Place a manned spacecraft in orbital flight around the earth
2. Investigate man's performance capabilities and his ability to function in the environment of space
3. Recover the man and the spacecraft safely

Guidelines:

1. Existing technology and off-the-shelf equipment should be used wherever practical
2. The simplest and most reliable approach to system design would be followed
3. An existing launch vehicle would be employed to place the spacecraft into orbit
4. A progressive and logical test program would be conducted

Spacecraft Requirements:

1. The spacecraft must be fitted with a reliable launch-escape system to separate the spacecraft and its crew from the launch vehicle in case of impending failure
2. The pilot must be given the capability of manually controlling the spacecraft attitude
3. The spacecraft must carry a retrorocket system capable of reliably providing the necessary impulse to bring the spacecraft out of orbit
4. A zero-lift body utilizing drag breaking would be used for reentry
5. The spacecraft design must satisfy the requirements for a water landing

A schematic of the Mercury capsule is given in Figure 2-1.

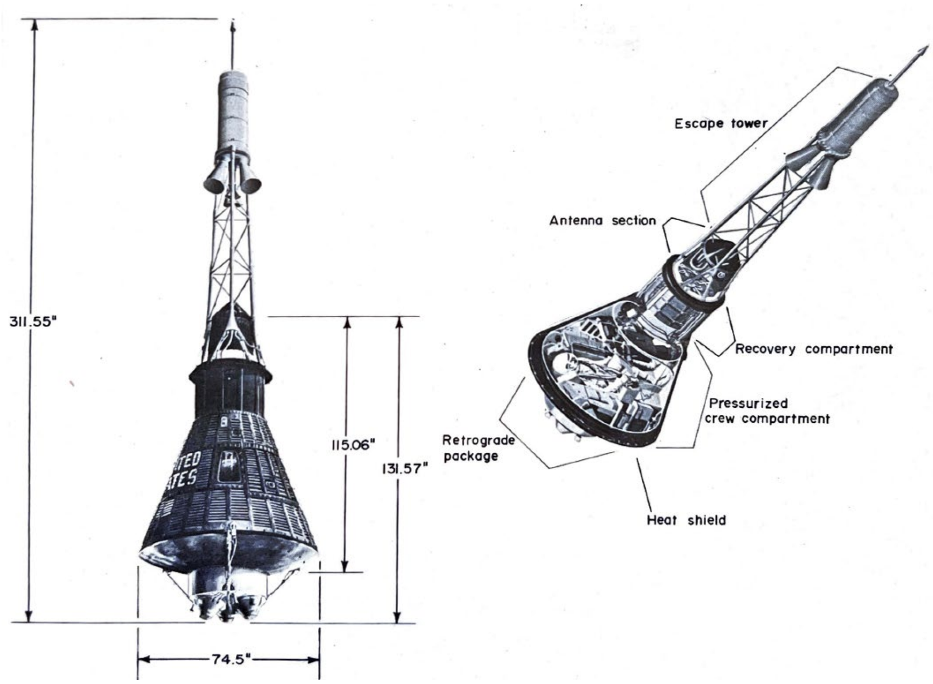
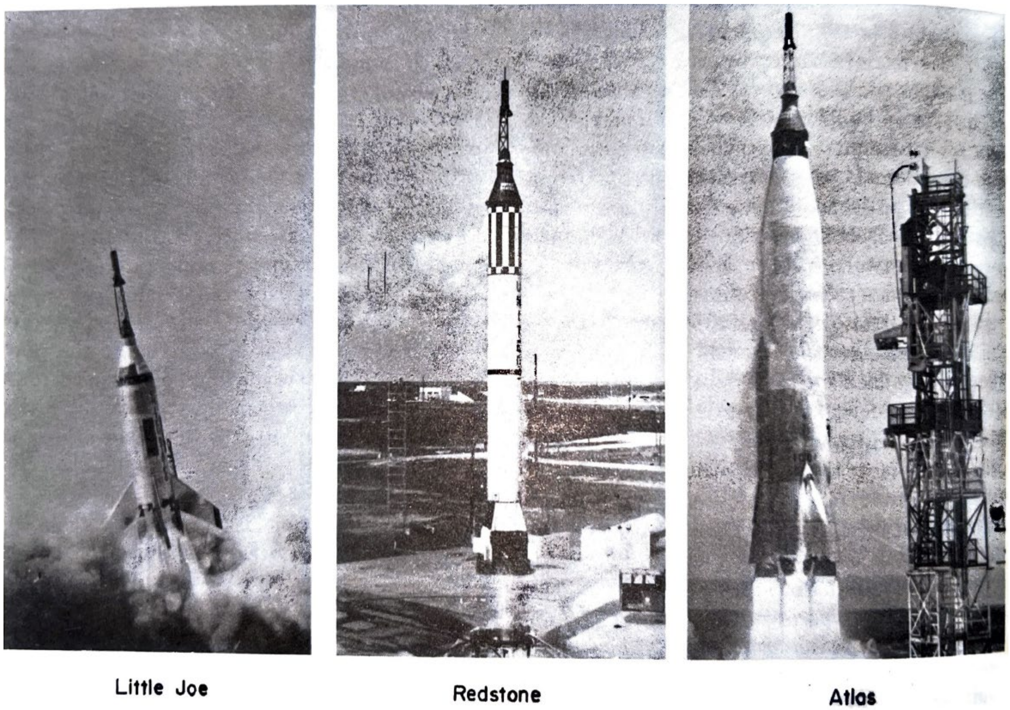


Figure 2-1. Schematic of the Mercury capsule (Mercury, 3).



Little Joe Redstone Atlas

Figure 2-2. Project Mercury space launch vehicles from left to right: Little Joe, Redstone, Atlas (Mercury, 14).

Project Mercury, from conception to retirement, encompassed 57 months or four and two-thirds years of work. Twenty-five major flight tests were conducted within a 45-month period; six of which were manned (Mercury, 24). The entirety of Project Mercury ‘s objectives were achieved 3 ½ years after official project approval, with the completion of astronaut John Glenn’s successful orbital flight on February 20, 1962. Subsequently, astronaut M. Scott Carpenter completed a similar mission. Then, astronauts Wally M. Schirra, Jr. and L. Gordon Cooper completed orbital missions of increased duration to provide additional information about man’s performance capabilities and functional characteristics in the space environment (Mercury, 14). Cape Canaveral, FL and Wallops Station, Wallops Island, VA had a combined twenty-three launches—seven Little Joe, six Mercury-Redstone, and ten Mercury-Atlas as shown in Figure 2-2. Over two million personnel directly or indirectly supported Project Mercury, with industry accounting for the bulk of the numbers (Mercury, 24). Finally, based upon 1963 dollars, the total accounting for the project showed a total cost of $384,131,000; however, final auditing had not been completed at the time of the reference's published date. Refer to [6] for more details. These cost figures include the Mercury tracking network which was used after the end of the program, and the cost of the operational and recovery support supplied for the mission (Mercury, 25).

With the introduction of Project Mercury completed, each of the elements of its MOA will be explored. Three attributes of the MOA are identified as three modern equivalents to present MOAs: the GSN/ground station, the GDS, and the MOC.

2.1.1) *The Mercury Worldwide Network*

The Mercury Worldwide Network (MWN) was an early MOA required for effective ground control during the unmanned and manned phases of Project Mercury. In its basic form, the MWN was required to provide all functions necessary for ground control and monitoring for a Mercury mission from launch to landing. This network provided everything from continuous tracking, telemetry, voice communications, flight control, computation, and data handling within defined parameters and margins. The function of the network was to end when the spacecraft had landed and the best possible information on the location of landing point had been supplied to the recovery team (Mercury, 127-128). The following were the functional requirements established for the MWN:

1. Provision of adequate tracking and computing to determine launch and orbital parameters and spacecraft location for both normal and aborted missions
2. Voice and telemetry communications with the spacecraft with periods of interruption not to exceed 10 minutes during the early orbits, contact at least once per hour thereafter, and communications to be available for at least 4 minutes over each station
3. Command capability to allow ground-initiated reentry for landing in preferred recovery areas and to initiate abort during critical phases of launch and insertion
4. Ground communications between the ground stations and the control center

The MWN utilized a total of sixteen stations comparable to a modern GSN. These stations were selected on the considerations of the flight plan and on the character of the spacecraft electronic systems consistent with the basic requirements. Additionally, compromises due to the constraints of Earth's rotation and the lack of geographic locations for a three-orbit mission resulted in gaps, primarily on the third pass leading to greater than the desired ten minutes (Mercury, 128). Figure 2-5 (on page 46) shows the locations of the Mercury stations and Table 2-1 (on page 46) defines the location's acronym.

Two Centers were also required. First, the Mercury Control Center (MCC), located at Cape Canaveral, was to provide a means of centralizing control and coordination of all the activities associated with a Mercury mission; the equivalent to a modern MOC (Mercury, 140). Second, the Computing and Communications Center located at Goddard Space Flight Center (GSFC) in Greenbelt, Maryland, was to provide for communications control, switching, and distribution; also, it was to provide all computations necessary to monitor and control the mission from launch to landing; the equivalent to a modern GDS (Mercury, 128).

Figure 2-3 (left). Acquisition-aid antenna installation at Guaymas, Mexico (Mercury, 132).
Figure 2-4 (right). Mercury computing center at GSFC (Mercury, 133).

Regarding the equipment systems utilized for the MWN, the three major considerations were economics, time, and safety. Although several stations had to be constructed and retrofitted to meet design requirements, the MWN's attempts to make use of existing facilities and proven equipment held true for most of the program's lifetime. The design requirements included, but were not limited to, reliability, redundancy, diversity, and the verifiability of the systems. To provide mission support, the systems of the MWN had to provide the following major functions (Mercury, 129):

1. Ground radar tracking of the spacecraft and transmission of the radar data to the Goddard Space Flight Center computers
2. Launch, orbital, and reentry computations during the flight with real-time display data being transferred to MCC
3. Real-time telemetry display data at the sites
4. Command capability at various stations for controlling specific spacecraft functions from the ground
5. Voice communications between the spacecraft and the ground, and maintenance of a network for voice, teletype, and radar data communications

The interested reader can dive into the specifics for each individual system provided by [6] at the end of the paper. These include the continuous radar tracking and active acquisition aid systems (Figure 2-3), the computing system (Figure 2-4), the telemetry system, the air-ground communication system, the command system, ground (network) communications system, and the timing system. Figure 2-5 shows the tracks for the Mercury capsule with respect to each ground station, while Table 2-1 lists the equipment present at each MWN station.

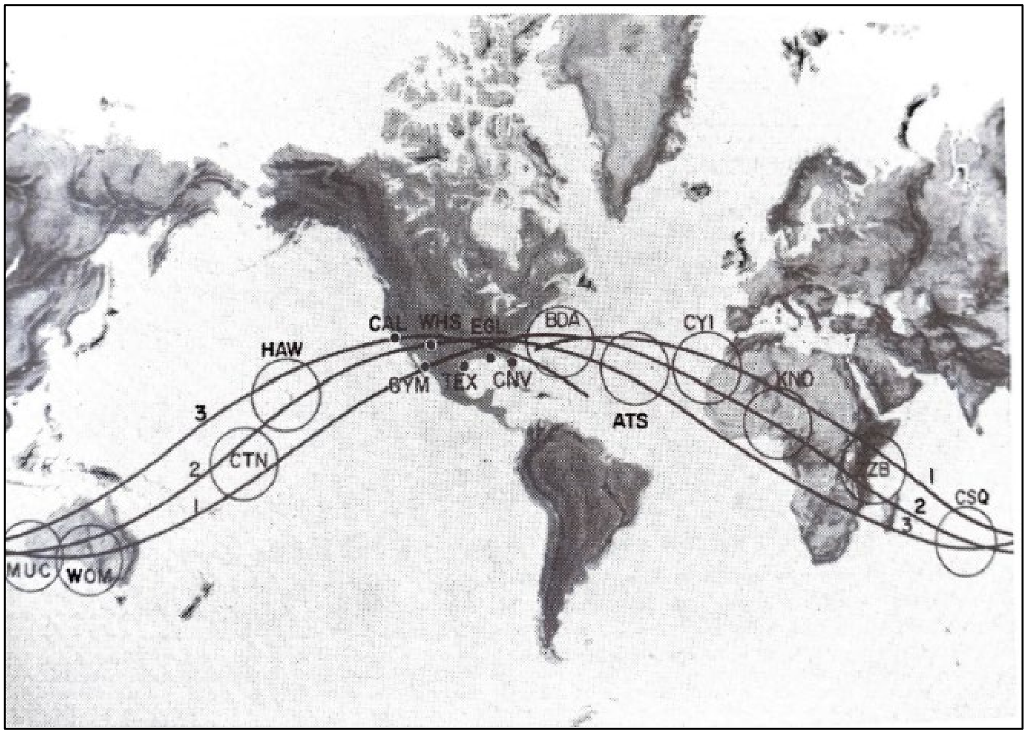
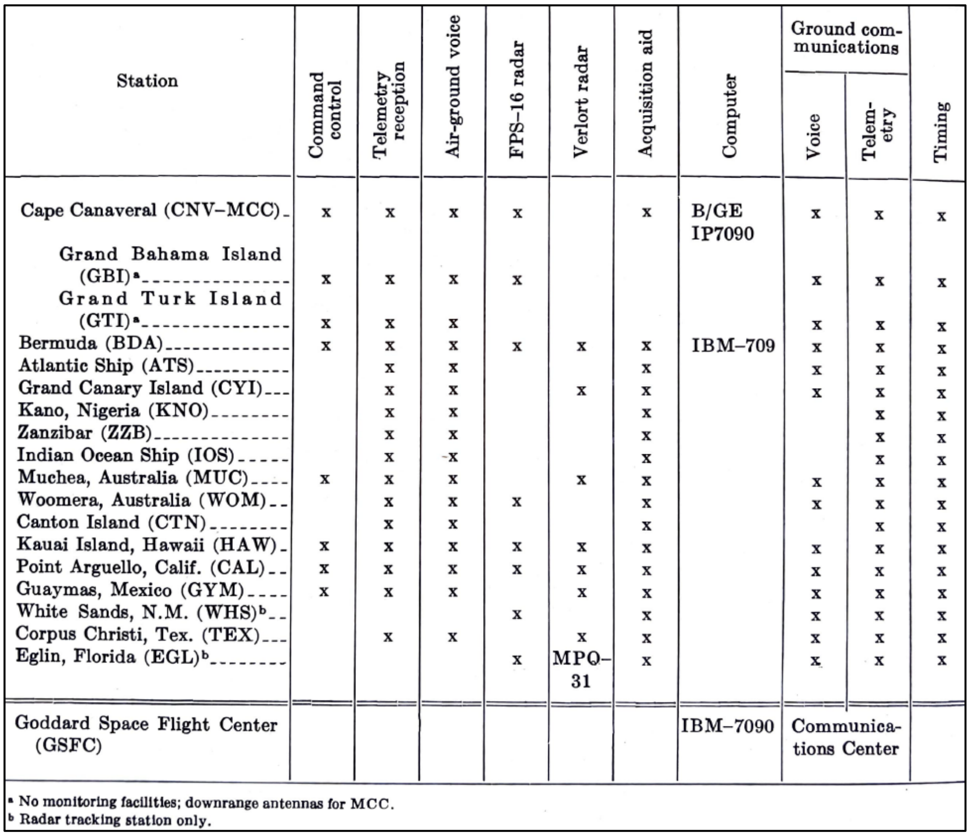


Figure 2-5 (above). Locations of the selected Mercury stations with ground tracks (Mercury, 128).

Table 2-1 (below). Mercury station list of abbreviations and equipment (Mercury, 146).



Station	Command control	Telemetry reception	Air-ground voice	FPS-16 radar	Verlort radar	Acquisition aid	Computer	Ground communications		Timing
								Voice	Telemetry	
Cape Canaveral (CNV–MCC)	x	x	x	x		x	B/GE IP7090	x	x	x
Grand Bahama Island (GBI)[a]	x	x	x	x				x	x	x
Grand Turk Island (GTI)[a]	x	x	x					x	x	x
Bermuda (BDA)	x	x	x	x	x	x	IBM–709	x	x	x
Atlantic Ship (ATS)		x	x			x		x	x	x
Grand Canary Island (CYI)		x	x		x	x		x	x	x
Kano, Nigeria (KNO)		x	x			x			x	x
Zanzibar (ZZB)		x	x			x			x	x
Indian Ocean Ship (IOS)		x	x			x			x	x
Muchea, Australia (MUC)	x	x	x		x	x		x	x	x
Woomera, Australia (WOM)		x	x	x		x		x	x	x
Canton Island (CTN)		x	x			x			x	x
Kauai Island, Hawaii (HAW)	x	x	x	x	x	x		x	x	x
Point Arguello, Calif. (CAL)	x	x	x	x	x	x		x	x	x
Guaymas, Mexico (GYM)	x	x	x		x	x		x	x	x
White Sands, N.M. (WHS)[b]				x		x		x	x	x
Corpus Christi, Tex. (TEX)		x	x		x	x		x	x	x
Eglin, Florida (EGL)[b]				x	MPQ–31	x		x	x	x
Goddard Space Flight Center (GSFC)							IBM–7090	Communications Center		

[a] No monitoring facilities; downrange antennas for MCC.
[b] Radar tracking station only.

2.2) Big and Tiny Trailblazers

2.2.1) *Lunar Reconnaissance Orbiter*

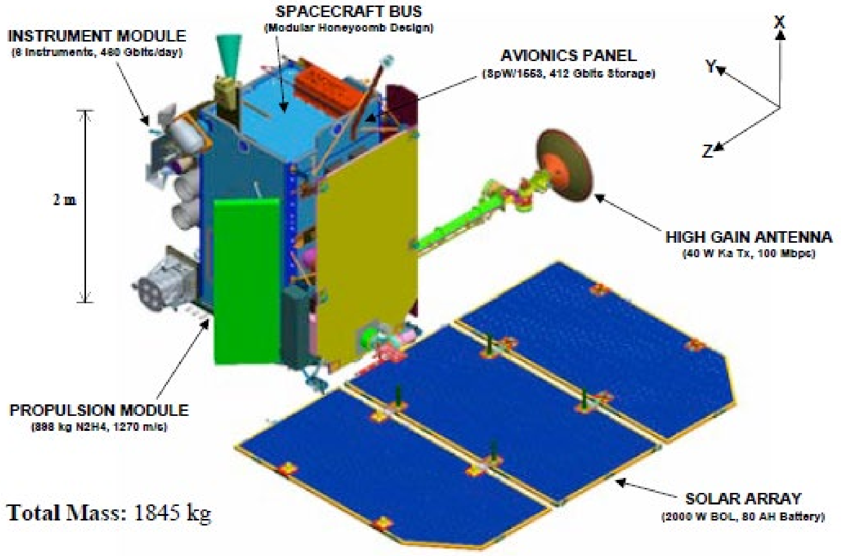


Figure 2-6. NASA's Lunar Reconnaissance Orbiter.[8]

Currently operating as a lunar mission managed by NASA's Science Mission Directorate, the Lunar Reconnaissance Orbiter (LRO) (Figure 2-6) has been the cream-of-the-crop in expanding our understanding and knowledge of the Moon. Originally under the authority of NASA's Exploration Systems Directorate's Lunar Precursor Robotic Program (LPRP) during its first year of operation, LRO launched June 18, 2009, where it set out to create a comprehensive atlas of the Moon's features and identifying available resources, including water. After that first year, LRO transitioned to a science phase under NASA's Science Mission Directorate with its payload carrying instruments with considerable heritage from previous planetary science missions. The objectives of the LRO mission are to aid NASA in identifying landing sites, locate potential resources, describe the current radiation environment, and demonstrate new technology.[7] Here was LRO's mission profile before launch:

> "LRO will launch on an Atlas V 401 rocket and the trip to the moon will take approximately four days. LRO will then enter an elliptical orbit, also called the commissioning orbit.

> From there, it will be moved into its final orbit—a circular polar orbit approximately 50 km (31 miles) above the moon's surface. LRO will spend at least one year in low polar orbit collecting detailed information about the moon and its environment" (PK, 14).

To expand on LRO's MOA, LRO's systems (flight and ground) are specially designed to manage the substantial amounts of data that are generated over the course of the mission. The key drivers to the LRO mission design and operations concept are its orbit and data requirements, which are expected for a cislunar mission.[8] When LRO began its lunar orbital insertion maneuvers, or LOIs, LRO's "universe" was beginning its transition to become moon centered. Once lunar capture was successful at the mission's required orbit, LRO's perspective switched to the Earth circumnavigating the Moon once a month, and the Sun circumnavigating the Moon once a year. Overall, LRO's orbit has a mean period of 113 minutes and a maximum eclipse time of 48 minutes (Houghton, 4). Therefore, twice a month LRO's orbit is in full view of the Earth for about two days, and twice a year LRO's orbit will be in full view of the Sun for about one month.

With those orbital characteristics set for operations, LRO's data handling can be mentioned. With the use of its high-gain antenna, LRO makes use of NASA's Deep Space Network (DSN) through S-band up/down-link and Ka-band downlink. The S-band is used for nominal spacecraft tracking (30 minutes per orbit) and Ka-band for downlink of all stored instrument and spacecraft data. The latter generates 460 Gbits on-board the spacecraft and downlinks at 100 Mbps. With S-band passes occurring twelve times per day and Ka-band passes occurring (on average) four times per day, nominally, LRO is never out of contact with the ground for more than 1 hour at a time. Station-keeping maneuvers and instrument calibrations occur once a month. Momentum management maneuvers occur every two weeks. Most of the instruments operate autonomously over the course of a single orbit, while one requires a tailored command timeline. Nominally, LRO receives a new command timeline from the ground once per day

(Houghton, 5). LRO's MOC resides at NASA GSFC in Greenbelt, Maryland and makes use of NASA's Planetary Data System for the storage of its data and data products. The project's life cost is approximately $500 million.

2.2.2) *Lunar Crater Observation and Sensing Satellite*

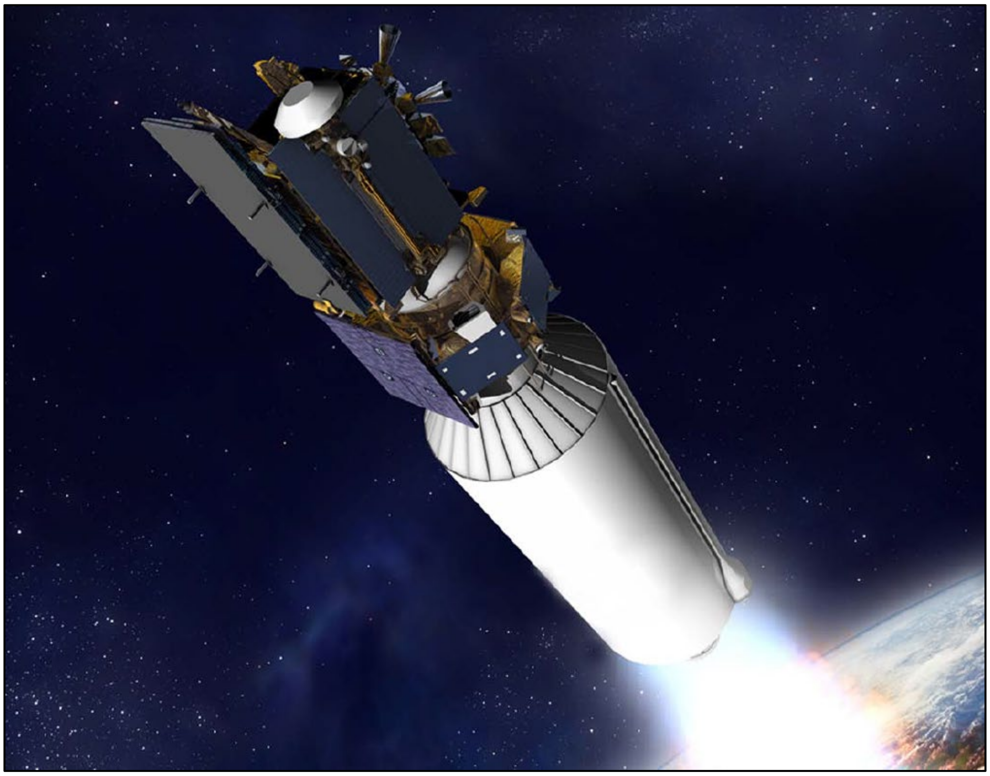


Figure 2-7. LRO (top), LCROSS S-S/C (center), ULA Centaur upper stage (bottom) (PK, 27).

Co-manifested for launch with LRO, LCROSS (Lunar Crater Observation and Sensing Satellite) (Figure 2-7) was a fast-paced, low-cost, mission selected under NASA's Exploration Systems Directorate's Lunar Precursor Robotic Program (LPRP). Representative of a new generation of fast-development, cost-capped missions that use off-the-shelf hardware and flight-proven software to achieve mission goals, LCROSS set the standard for a secondary mission with LRO set as the primary. While LRO remotely senses evidence of resources such as water ice in cold regions of the Moon, LCROSS directly determined whether water ice formed in the permanently shadowed regions near the lunar poles (PK, 5). Scientifically, the detection of the nature of hydrogen in those regions.[9]

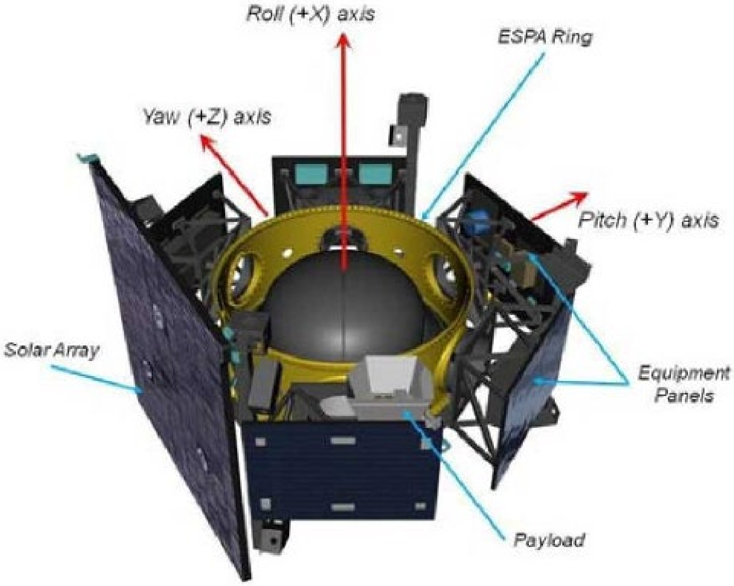


Figure 2-8. LCROSS Shepherding Spacecraft (PK, 27).

As described from its mission profile, rather than incur the high cost and mission complexity associated with landing at a lunar pole, LCROSS precisely guided a ULA Centaur (rocket) upper stage to a lunar target as a kinetic impactor to raise a plume of lunar regolith from the surface. The location of impact was chosen to be a lunar polar crater named Cabeus. After releasing the Centaur, the guiding Shepherding Spacecraft (S-S/C) (Figure 2-8) employed a suite of nine science instruments to analyze the impact flash and plume materials at close range, with it itself impacting the surface around four to five minutes later. From launch with LRO on June 18, 2009, to impact on October 9, 2009, LCROSS remained in operation for 112 days in flight. After one-month of post-impact analysis, the LCROSS science team announced the positive identification of water on the floor of Cabeus (Tompkins, 2).

To expand on LCROSS's MOA, the spacecraft was designated as a Class D mission, indicating NASA's willingness to accept greater levels of programmatic and operational risk. The mission presented several inherent challenges—planning for and achieving a precise impact at a lunar pole, guiding a Centaur upper stage far beyond its operational lifetime, co-launching as a secondary payload with LRO, and preparing for and operating such a mission under a highly constrained budget and short development cycle. LCROSS had four operational phases: Launch (liftoff to S-S/C activation), Transfer (activation to lunar gravity assist), Cruise (gravity assist to TCM 8) (Trajectory Correction Maneuver), and Final Targeting and Impact (TCM 8 through S-S/C impact). Regarding impact, to achieve the desired impact geometry, LCROSS performed a lunar gravity assist to transfer from its trans-lunar orbit to a lunar gravity assist lunar return orbit (LGALRO), an elliptical, Earth-centered orbit of roughly lunar semi-major axis but inclined steeply with respect to the lunar orbit plane (Tompkins, 3).

LCROSS mission operations were conducted primarily at NASA Ames Research Center (ARC) from a seven-seat MOC room, a mission support room for planning and off-line analysis, and a SOC. Additionally, several remote operations facilities and personnel supported the mission, including the Jet Propulsion Laboratory (JPL) (navigation and link scheduling), NASA GSFC (maneuver design), and two Northrop Grumman sectors (real-time and off-line engineering support). Support was also provided by LRO, and the Hubble Space Telescope (HST) for impact observations. The flight team was divided into two shifts of real-time spacecraft support, and another team devoted to maneuver design, communications scheduling, and engineering analysis. Specifically for communications, LCROSS utilized the DSN. Both uplink and downlink were conducted with 34-meter antennas; however, downlink also was conducted through 70-meter antennas (for high-rate science downlink applications) and the 26-meter antenna at Canberra (for initial acquisition) (Tompkins, 3).

Ridesharing with LRO on the Atlas V 401 left an enormous impact on LCROSS. Though

LRO imposed strict limitations on the LCROSS design and partially restricted its use of the DSN in the first week of operations, the LRO team provided invaluable support to the LCROSS for impact site selection and conducted pre- and post-impact observations in support of the LCROSS science mission. LCROSS was required to follow a policy of non-interference with the LRO mission that governed many aspects of the program and spacecraft design. Therefore, compromises ensued, such as power-off launch and automatic activation after LRO separation and DSN resource contention and resolution during critical events. Tompkins, et al. describes both as notable examples of constraints a secondary spacecraft must endure for mission success (Tompkins, 8). LCROSS was designed and built by Northrop Grumman Aerospace Systems in Redondo Beach, California with a project cost of $79 million plus additional funding from LPRP to cover delays in launch (PK, 7). The Atlas V 401 configuration for LRO and LCROSS is given in Figure 2-9.

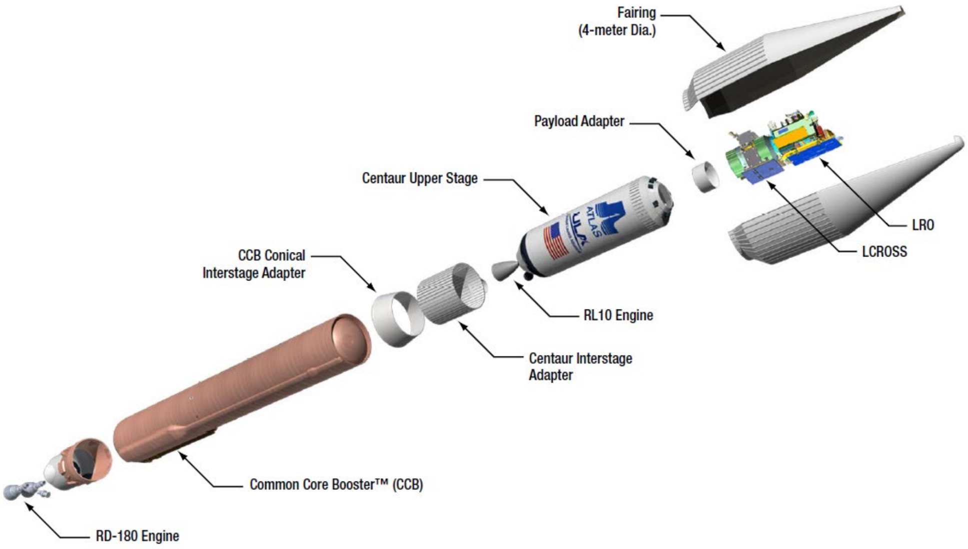


Figure 2-9. LRO/LCROSS launch configuration ULA Atlas V 401(PK, 27).

2.2.3) *Mars Cube One*

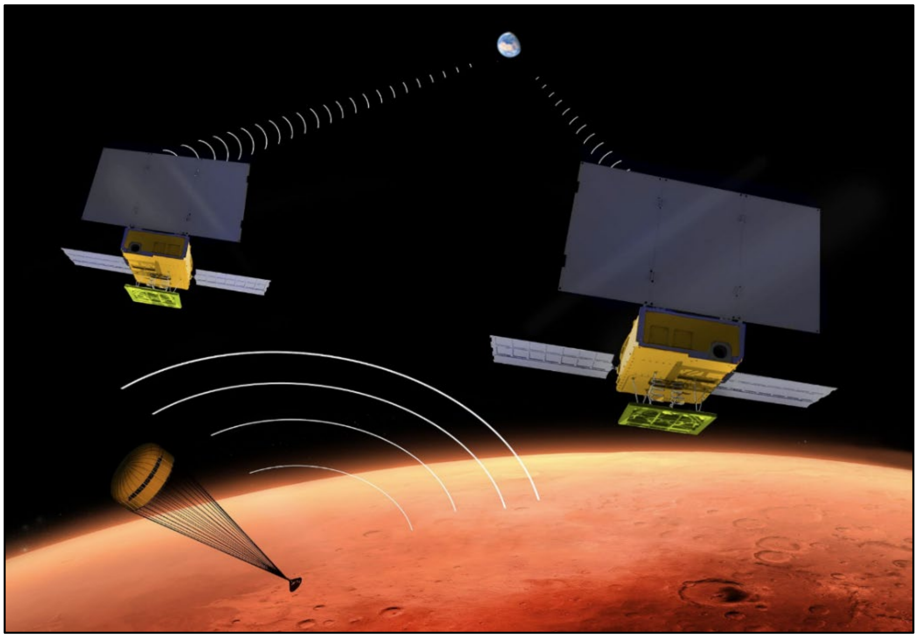


Figure 2-10. Artist conception of MarCO-A and B monitoring InSight during EDL, June 12, 2015. Image by NASA/JPL-Caltech. http://photojournal.jpl.nasa.gov/jpeg/PIA19388.jpg.

Mars Cube One (MarCO) (Figure 2-10) was a set of two 6U interplanetary CubeSats (MarCO-A and B) (34 x 22 in or ~87 x 56 cm at 30 lbs. or ~13.6 kg) that launched with the InSight Mars lander on May 5, 2018. Shortly after on November 26, 2018, MarCO set the standard for what an interplanetary CubeSat is capable of. Before diving into its mission profile and MOA, a quick brief on interplanetary CubeSats is required. Interplanetary CubeSats take advantage of the CubeSat paradigm and the availability of commercial components developed for LEO missions, but they are specifically designed to explore deep space. As a result, interplanetary CubeSats are essentially different from LEO CubeSats in the three primary technological areas: propulsion, radiation tolerance, and telecommunications. While fundamental differences exist in these areas, interplanetary CubeSats require changes to almost every subsystem (CH, 85). Succinctly put, interplanetary CubeSats are different because their orbits take them far from home and its familiar environment. To be successful in LEO, a CubeSat must be relatively complex in a manner that affords powering and survival of subsystems, maintaining specified orientations, and responding to commands, all while acquiring and transmitting health status and data to meet mission objectives. Once a CubeSat leaves Earth's vicinity, all these functions must be provided, but for some these functions the "how" requires new techniques and complexity. The interested reader can reference authors Malphrus, Freeman, Staehle, Klesh, and Walker diving deeper into the specifics of the differences, particularly with propulsion, telecommunications, and navigation (CH, 89).

From its mission profile, the objective of MarCO was to provide a critical real-time communication link (relay) to Earth for InSight's entry, decent, and landing (EDL) at Mars. The spacecraft were developed from commercially available components obtained from CubeSat system suppliers (CH, 99):

- CDH/electronics: AstroDev of Ann Arbor, MI
- ADCS: the XACT by Blue Canyon Technologies of Boulder, CO
- Solar arrays: MMA Design LLC also of Boulder, CO
- Dispenser: Tyvak NanoSatellite Systems of San Luis Obispo, CA
- Cold gas micropropulsion system (R-236FA propellant): VACCO Industries of South El Monte, CA
- Flat Panel Reflector High Gain Antenna: JPL of Pasadena, CA

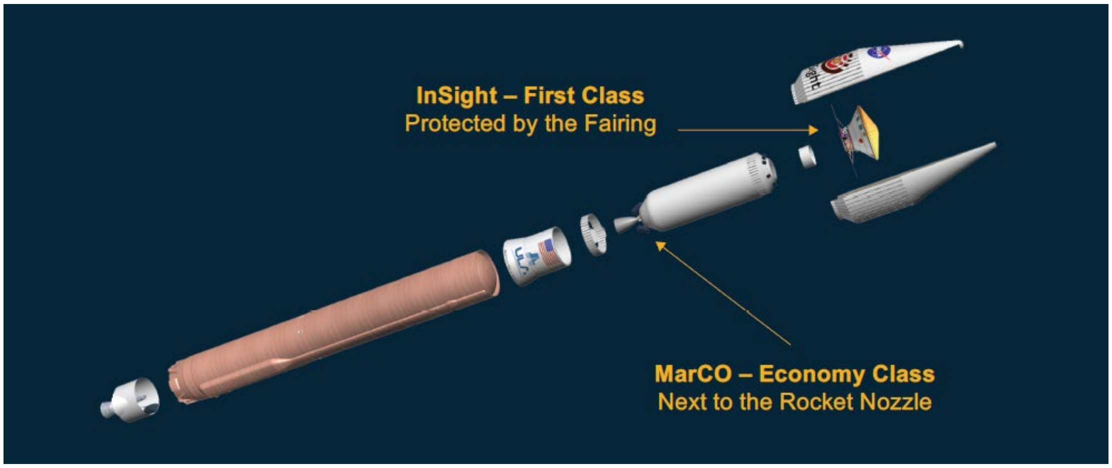


Figure 2-11. InSight/MarCO launch configuration.[4]

All systems were proven to be effective during deep space operations. During the mission, MarCO performed as intended, receiving the UHF telemetry from InSight during its EDL and relaying this telemetry at X-band frequencies to DSN as well as images of Mars and the Earth-Moon system (CH, 100). Now, while MarCO was the supporting role for InSight, it also holds recognition as a successful technology demonstration. It utilized a miniaturized deep space radio (IRIS) that performed successful uplink at 62.5 bps – 1 kbps, downlink at 62.5 bps – 16 kbps, ranging, and Delta-DOR (delta-differential one-way ranging); successful use of the Flat Panel Antenna; and TCMs performed on a CubeSat. Launch was conducted on the ULA Atlas V 401 configuration as shown in Figure 2-11. The MOC for MarCO was located at JPL in Pasadena, California. The project cost was $18.5 million with a mission duration of 1.25 years.[4]

3) The Breakdown of a MOA

3.1) Work Breakdown Structure: The M2m Approach

With any project it is vital to have a methodology towards the implementation of said project. Therefore, a work breakdown structure, or WBS, is required to ensure those assigned to the project are aware of the methodology towards successfully accomplishing project goals. For designing a MOA for small spacecraft, the WBS will utilize the concept of a MOA being comprised of macroscopic and microscopic elements; the M2m approach. Figure 3-1 presents the WBS for the project. The purpose of this WBS is to examine every macro and micro-element and exploit its purpose in a MOA.

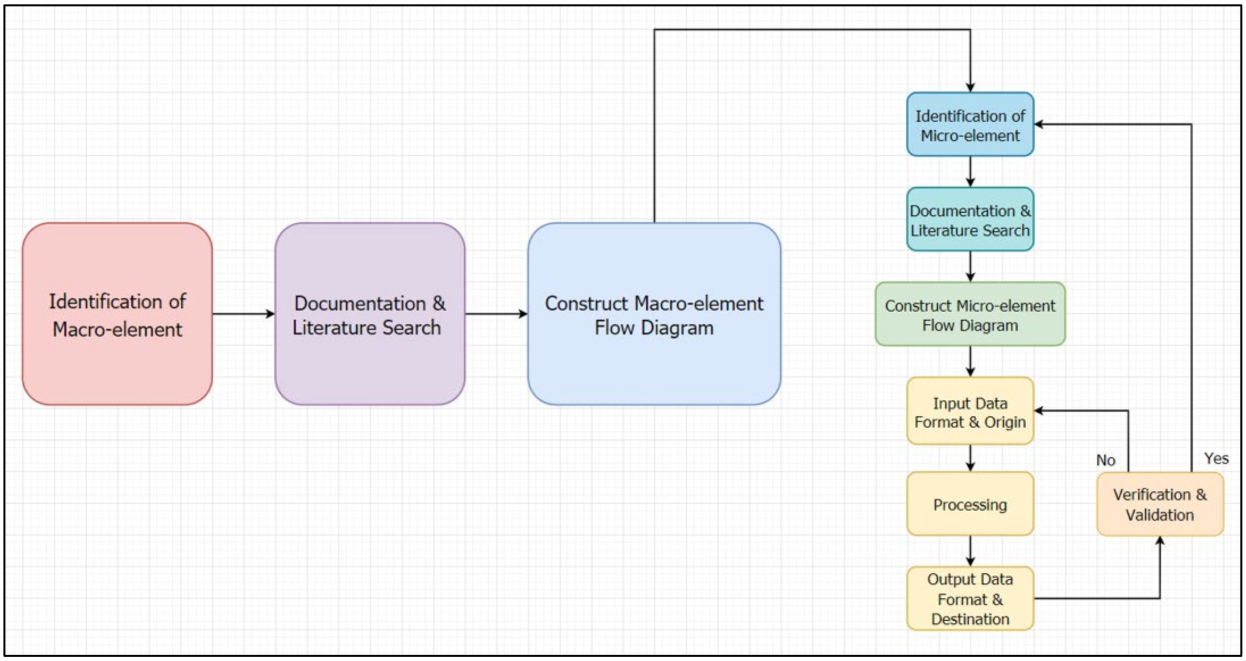


Figure 3-1. Project work breakdown structure.

There are four items of priority to address when examining each element: (1) the input from the previous element; (2) The processes of the current element; (3) The output to the next element; and (4) verification and validation for the next element. Item (1) addresses the beginning of the element: Where did the input data come from? What does it contain and what is its configuration? Item (2) addresses the processes of the element: What does the element do with the data? In other words, what is its priority? Item (3) addresses the end-result of the element, or the output: What is the new configuration of the data and where does it go? Item (4) addresses the verification and validation of the element: Does the element operate/process as intended?

Taking a deeper dive into the project WBS from Figure A, the first step is identifying a macro-element of the MOA, a system incorporating multiple handlers and processes at once, such as a ground data system. Finding information on the processes and skeleton of the macro-element will require a documentation or literature search thereafter, and if necessary, a documentation creation period should be introduced if material is unique or unavailable. After a thorough information search, the team should map the skeleton of the macro-element through a flow diagram. The connections or arrows between each element will signify the “data handshake” between the elements, or the transfer of data from one element to the next. Once the flow diagram of the macro-element is completed, the team already has the basis for identifying the micro-elements. When a micro-element is identified, a documentation and literature review should be conducted to garner the necessary info based on the four principles previously stated. A separate flow-diagram may need to be created for extra emphasis if the micro-element involves pertinent sub-elements to the MOA that should not be overlooked. Next, once the micro-element has been fully assimilated by the team, a live, iterative data flow through the MOC should ensure that the four principles are accurate based on documentation gathered and organized. Last, if all micro-elements have been accounted for, verified, and validated, the next macro-element can be visited, and this iterative process continues until all macro-elements are accounted for in the MOA. At that instance, the MOA can be considered completely mapped. This WBS is applied to the case of LIC’s MOA for this chapter and the V&V of LIC’s MOA in Chapter 4. A macroscopic view of LIC’s MOA is presented as Figure 3-2. Throughout this chapter, a section of Figure 3-2 is highlighted to indicate the current macro-element being examined.

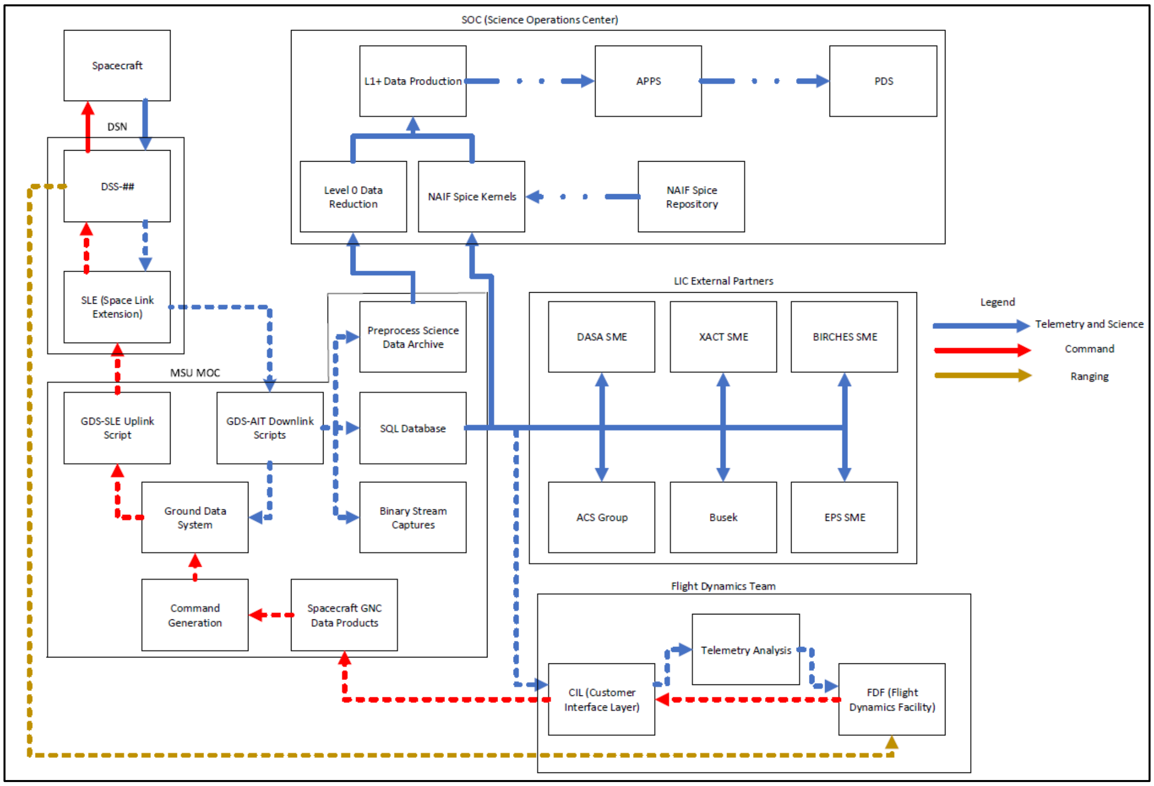


Figure 3-2. Macroscopic view of LIC's MOA.

3.2) The Ground Station Architecture

This section considers the macro-elements of a ground station architecture (GSA). The GSA will be exclusive to Direct-to-Earth, or DTE, ground services for a deep space small spacecraft-based mission. DTE services provide direct point-to-point access with antennas at ground stations which are strategically located and equipped with telemetry, command, and tracking services. Additionally, DTE is especially effective for missions needing frequent, short-duration contacts with high data throughput, and are also capable of handling longer latency durations due to orbital dynamics and stations visibility. This is such a case for deep space small spacecraft (SOA, 221). A deep space small spacecraft can have one of two options: contract with a ground station network (GSN) or utilize an independent ground station. Because every mission does not have access to an exclusive ground station and dish antenna, both cases are considered separately; however, there are missions that do obtain such assets. LIC is one of those missions, and a highlight is its use of a GSN (refer to §3.2.2) and an exclusive ground station consisting of a 21-meter dish antenna (refer to §3.2.4). An important note: since there is a mutual relationship between LIC's GSN and its ground station, the case study separates each macro-element for lack of confusion and improved clarity; therefore, there should be no loss of generality.

3.2.1) *Ground Station Networks*

A GSN is a conglomerate of tethered ground stations that serve as the central communications conduit for all space-based communications with a multitude of space assets and their respective GDS, MOC, EUFs, and various supporting infrastructure. For deep space small spacecraft-based missions, understanding the number of ground stations required to support the mission is the first step in designing a MOA and determining if a GSN is necessary (SOA, 225). The number of ground stations required for a mission depends upon multiple factors, including the number of satellites, the orbit regimes and inclinations, and the data latency or data volume requirements. For example, if a satellite has an orbit that regularly crosses over the same position on Earth, that mission could be supported by a single ground station at that frequently revisited position. However, if a satellite's orbit does not frequently revisit the same position on Earth, then multiple ground stations will be required to support that mission (SOA, 225). Similarly, if a mission requires the satellite to downlink collected data as soon as possible, i.e., low data latency requirements, or if the mission will generate a large volume of data during each orbit, e.g., many remote sensing missions, then more ground stations will be required to support the mission (SOA, 225). Another consequence of determining the number of ground stations required is the size of the GSN. Missions that require a single ground station with a co-located MOC have a very simple network (refer to §3.2.3 and §3.2.4). Missions that require multiple, geographically dispersed ground stations will have a larger GSN with provisions to ensure the MOC can communicate effectively with each of those ground stations (SOA, 225).

3.2.2) *Lunar IceCube and the Deep Space Network*

For a typical deep space small spacecraft-based mission, a GSN would be a realistic requirement when considering the depth and breadth of a mission beyond Earth orbit. Such is the case for LIC, which utilizes NASA's Deep Space Network, or DSN. As stated by the DSN's MSIPP, the DSN is responsible for providing command, telemetry, and tracking services using the DTE methodology of services mentioned previously. These capabilities can be conducted either at DSN facilities or by arranging support from other foreign or commercial space assets.[10]

Unique is just one word to describe the capabilities of the DSN. For any deep space mission, there are three constraints for communications from Earth to spacecraft: weak downlink signals, distant uplink, and precision. To detect weak downlink signals, cryogenically cooled (6 to 12 K) very low noise amplifiers are required at each node, not to mention precise frequency and timing standards (utilizing a hydrogen maser), ultra-stable receivers, and data rates from 10 bps to 6.6 Mbps. Next, to combat distant uplink, high power (20 kW) transmitters with large apertures yields high EIRP (high effective isotropic radiated power), phase continuous ramped uplinks, and high stability. Lastly, timing precision affects radiometric data that provides ranging distance and dynamics (Doppler measurements); therefore, ranging is conducted down the nanosecond level.[11]

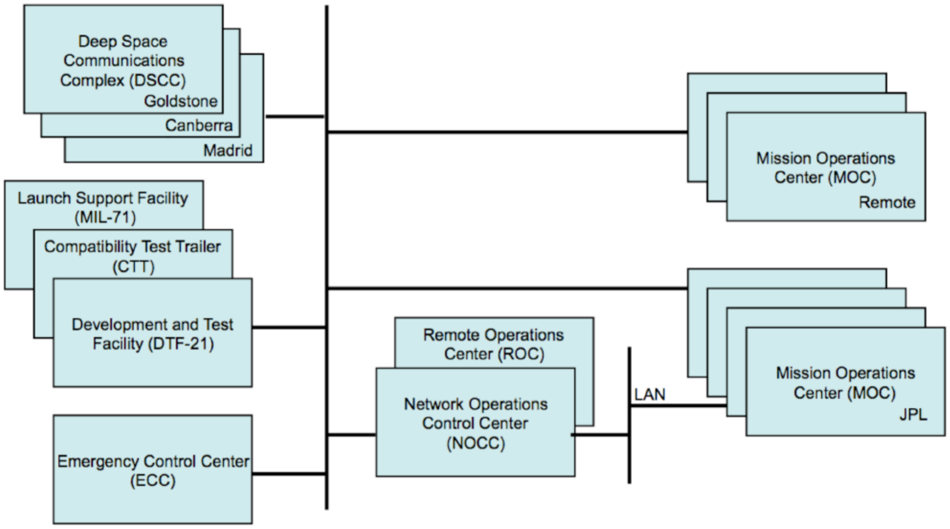


Figure 3-3. Physical view of the Deep Space Network (DSNSC, 2-2).

Figure 3-4 (left). CDSCC in Canberra, Australia, April 1, 2010. Image by Robert Kerton, CSIRO. http://www.scienceimage.csiro.au/pages/about/.
Figure 3-5 (right). DSS-17, 21-meter antenna at night, August 18, 2021. Image by Morehead State University. https://www.moreheadstate.edu/news/2021/august/morehead-state-s-21m-antenna-to-support-nasa-deep?feed=News.

The DSN consists of several key facilities in the context of LIC's MOA depicted in Figure 3-3.[12] First, there are three Deep Space Communications Complexes (DSCC): Barstow near Goldstone, CA (GDSCC); Madrid, Spain (MDSCC); Canberra, Australia (CDSCC). Currently, each complex consists of a signal processing center (SPC) and several beam wave guide dish antennas called deep space stations (DSS) (the exception is DSS-65 as a high efficiency antenna):

- GDSCC: one 70-meter (DSS-14), three 34-meters (DSS-24, DSS-25, DSS-26)
- MDSCC: one 70-meter (DSS-63), five 34-meters (DSS-53, DSS-54, DSS-55, DSS-56, DSS-65)
- CDSCC: one 70-meter (DSS-43), three 34-meters (DSS-34, DSS-35, DSS-36) (Figure 3-4)
- MSU (Morehead, KY): one 21-meter (DSS-17) (Figure 3-5)

All nodes on the DSN downlink (telemetry) at the S, X, K, and Ka bands, while uplink for commands is conducted on the S and X bands (DSNSC, 2-3). While LIC is commissioned to use any node for X-band uplink and downlink, LIC's particular interest lies with DSS-17 located on the campus of Morehead State University in Morehead, KY. As previously mentioned, LIC's MOA is unique because of its exclusivity of using a ground station. DSS-17 is a 21-meter beam waveguide dish antenna that has recently become the first non-NASA affiliated node on the DSN. DSS-17 utilizes both S and X-bands for uplink and downlink. More on the physicality of DSS-17 as a macro-element is presented in §3.2.4.

Second, there are three test facilities identified by the DSN Services Catalogue. The first is the Development and Test Facility (DTF-21), located at Monrovia, CA near JPL, is used to conduct tests of RF compatibility between the DSN and LIC's flight system, and as developmental testing ground for modifications to be implemented in the DSN. Next, the Compatibility Test Trailer (CTT-22) is a transportable facility for conducting tests of RF compatibility between the DSN and systems at the LIC MOC. Last, the Launch Support Facility (MIL-71) provides re-verification prior to launch operations at NASA's Kennedy Space Center (DSNSC, 2-4).

Third, there are several operations centers aside from the MOC for LIC. The first is the Deep Space Operations Center, or DSOC, located at the JPL Oak Grove site in Pasadena, CA. At DSOC the computing and communications equipment and personnel provide central monitor and control of the DSN, coordination between the DSCCs, as well as a data processing and data delivery interface to the MOC (DSNSC, 2-4). DSOC is the highlight for LIC's GDS regarding data processing, delivery, and staging; therefore, refer to §3.3. Next, the Emergency Control Center (ECC) is a DSN facility located at GSDCC that is a scaled down operations center intended to enable limited DSN continuity of operations and flight operations if a natural disaster or other catastrophe disables operations at JPL (DSNSC, 2-4). Last, the Remote Operations Center (ROC), also in Monrovia, CA, comprises of the facilities, equipment, and personnel that provide everyday operations, engineering, and support functions for the DSN (DSNSC, 2-4). It is important to note that while LIC contracts with the DSN, these facilities are directly and indirectly associated with LIC's MOA. Therefore, they can be considered as a portion of the expanded representation of the DSN. Figure 3-2 shows the first macro-element of the LIC MOA, DSN/DSS-17, highlighted below.

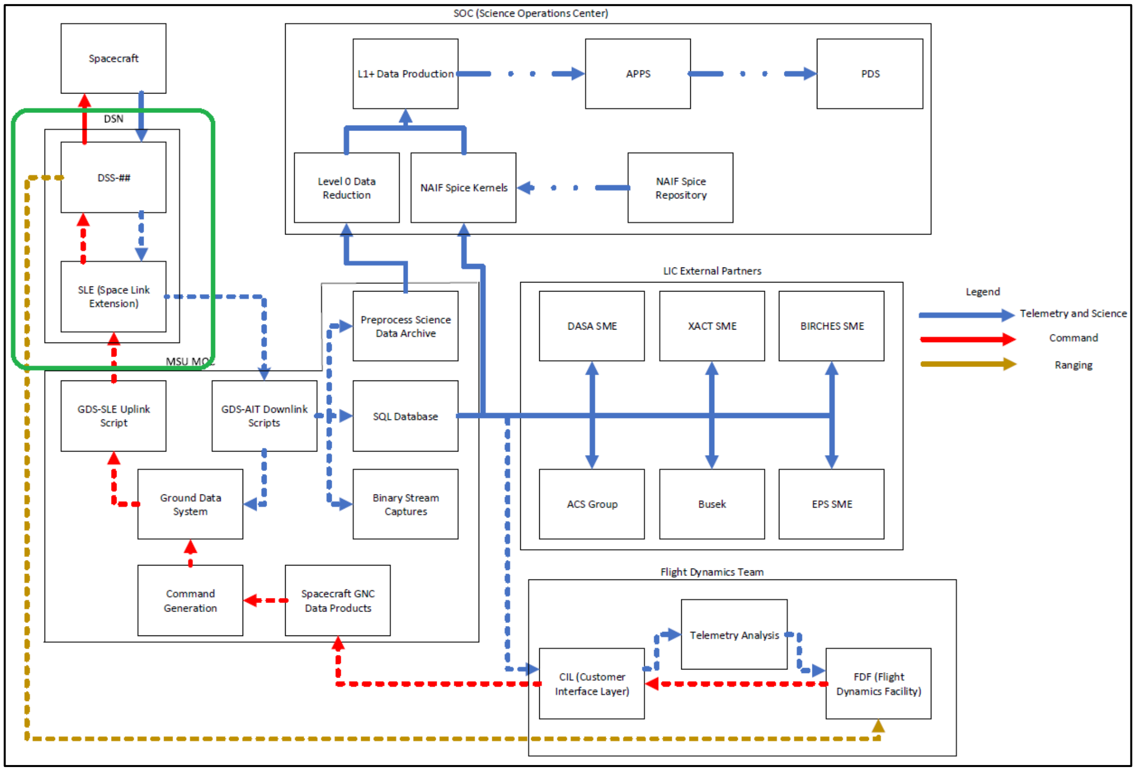


Figure 3-2. Macroscopic view of LIC's MOA. DSN/DSS-17 macro-element highlighted in green.

Diving into the micro-elements of the DSN, each DSCC, including DSS-17, possesses three micro-elements: an uplink subsystem (UPL), a downlink tracking and telemetry subsystem (DTT), and a data capture and delivery subsystem (DCD). While the antennas at each DSCC can be considered a fourth micro-element, a singular ground station/antenna is treated as a macro-element with DSS-17 as the representative (refer to §3.2.4). Recall that every spacecraft either has a ground station/antenna to utilize or contracts with a GSN. The UPL is a collection of hardware and software that handles data processing for uplink communications with LIC, and consolidates the command, exciter, transmitter, and uplink ranging systems; all command-related actions for LIC. Like the UPL, the DTT is also a collection of hardware and software that processes the radio frequency input (signals from LIC) from the antenna to the output of telemetry frames and, additionally, processes input for tracking, ranging, and Doppler estimates; all telemetry-related actions for LIC. Lastly, the DCD, captures and gathers all data generated from the DTT and manages the delivery of said data.[13] Since LIC utilizes both DSN and DSS-17, the UPL, DTT, and DCD are mutual for both macro-elements; therefore, more specifics are given in §3.2.4.

The DSN-provided services for LIC are classified as two actions: the exchange of telemetry data and the exchange of command data; respectively, downlink and uplink. This is the macroscopic beginning for LIC's data management. Therefore, these two actions set the interplay between the elements of the DSN, GDS, and the LIC MOC. From DSN and LIC's OICD, a high-level brief is given for the telemetry and command exchanges, respectively:

> "The exchange of telemetry data between the DSN and the Mission Operations [Center] is accomplished via the DSN Space Link Extension (SLE) Gateway at the JPL [DSOC]. The tracking data [is] delivered to the MOC at MSU and GSFC FDF."[14]

> "The exchange of command data between the DSN and the Mission Operations [Center] is accomplished via the DSN Space Link Extension (SLE) Command Link Transmission Unit (CLTU) service in the DSN Uplink Subsystem (UPL) at the [DSCC]."[14]

It is understood that the telemetry exchange involves the DTT and DCD, and the command exchange involves the UPL. Now, the Space Link Extension (SLE) is a framework of interfaces between the DSN and LIC MOC provided by the GDS though DSOC at JPL (SLE Gateway) and standardized by the Consultative Committee for Space Data Systems (CCSDS). Discussion regarding the CCSDS SLE interfaces for telemetry and command is held off until §3.3.1. An analysis of the command and telemetry data based on those interfaces is given in §3.4.1 and §3.4.3, respectively.

3.2.3) *Ground Stations*

Whether it be called “the ground segment” or “the ground station,” the overall purpose is clear: deep space communications for tracking, telemetry, and command of spacecraft. Because the previous section involved ground stations tied to the DSN, the focus remains on ground stations supporting deep space small spacecraft. According to José M. L. Agra and Alberto G. Muíño of Alén Space in Nigrán, Spain, “The ground segment is a key component to the success of any [small spacecraft] mission…[it] gives support to the space segment, relaying payload data to the user segment and managing Tracking, Telemetry, and Control (TT&C) for both the platform and the payload” (CH, 341). While there are many ground stations with various capabilities, such capabilities must be investigated to attain reliable performance during operations. Higher frequencies, such as S, X, K, and Ka, are gaining momentum is the small spacecraft market; especially for complex missions that require high data rates. What is commonly seen for these situations is the use of the parabolic reflector or parabolic dish antenna, such as the nodes provided by the DSN. A dish antenna requires a feed system to capture the radio frequency signal and guide it to the transceiver through a coaxial cable or waveguide. Vice versa for the transmission. It is also recommended that the dish feed supports configurable polarizations without the need for a setup modification (CH, 344). The next section involves a brief on LIC's main ground station, DSS-17. Note that while the suite of hardware and software that encompass DSS-17 are not covered in their entirety in this case study, please refer to [13] and online sources provided by the Morehead State University Ronald G. Eaglin Space Science Center.

3.2.4) ***Lunar IceCube and DSS-17***

Refer to the accompanying flow-diagrams in §A.2 (Figures A-7a and A-7b) for this section. As mentioned in §3.2.2, LIC utilizes the X-band range of frequencies for its telemetry and command. It's main ground station, DSS-17, is a 21-meter dish antenna system that provides telemetry, tracking, ranging, and communication services for LEO, MEO, and "near Earth" deep space missions independently and as an affiliated node on the DSN. LIC is classified as a cislunar mission (within the confines of the Earth and the Moon); therefore, it meets the criteria. DSS-17 has the capabilities of deep space S-band for uplink and downlink at 2.020 – 2.120, and 2.200 – 2.700 GHz, respectively; then, deep space X-band for uplink and downlink at 7.145 – 7.235, and 8.400 – 8.500 GHz, respectively.[15] Additionally, the station has the capabilities of multiple receive polarization: RHCP, LHCP, VERT, and HORZ. The breakdown of DSS-17 is divided into uplink, downlink, and tracking. Mostly, all activity occurs at MSU, but there are DSN micro-elements that require attention such as those in the previous section. Note that the current DSS-17 architecture flow presented here is for X-band communications.

First, uplink. Uplink is associated with the commanding of LIC through command data (CMD). All commands are generated from the LIC MOC that jointly serves as the control room for DSS-17. Those commands are then processed by the UPL provided by the DSN. As mentioned in §3.2.2, the UPL handles data processing for uplink communications with LIC. The UPL receives directives and command data from LIC MOC, modulates the uplink carrier based on that data, amplifies the modulated carrier for transmission, and provides uplink carrier phase data for Doppler measurements as well as ranging phase data for ranging measurements. The UPL is compatible with the PSK and BPSK modulations schemes as well as acceptable CCSDS Command Link Transmission Units, or CLTUs (refer to §3.3.1 and §3.4.1) (Wilczewski, 25). After the command data is processed by the UPL and outputs an IF signal of ~300 MHz, it is then upconverted by an IF/RF upconverter monitored and controlled (M&C) by a Raspberry Pi single-board computer. An upconverter utilizes mixers and local oscillators that takes as input a low frequency signal and converts it to a high frequency signal as an output; the output from the IF/RF upconverter is ~7.1 GHz. The upconverted signal is then sent to the 21-meter antenna which experiences a gain of 62.7 dBi at 8.4 GHz. The result are commands sent as RF signals to LIC.

Next, downlink. Downlink is associated with the telemetry of LIC through telemetry data (TLM). The requests for telemetry data, like that for commands, are produced from the LIC MOC. An RF signal is received from LIC at the antenna and is then passed to a low noise amplifier, or LNA. In general, power constraints set the pace of small spacecraft missions, such as LIC. Very low power is available for transmissions; therefore, a weak signal is captured by the antenna and it must overcome the feed line losses experienced in the system before the receiver. An LNA is used to combat this common scenario; a group of electronic amplifiers with low impact in the SNR that are placed close (immediately after) to the antenna to keep the overall noise figure as low as possible (CH, 345). The LNA of DSS-17 is a compact and cryogenically cooled system that is designed to be very low maintenance and low power (Wilczewski, 16). The nominal operational temperature of the LNA using the X-band feed is < 20 K with the system at < 100 K (MSU, 1). Using the M&C, after the RF signal passes through the LNA, it is downconverted to IF by the RF/IF downconverter. For the case of S-band and higher frequencies, the use of a low-noise downconverter is convenient, since it will not only amplify the signal but also convert it to a lower frequency—the difference between the input RF frequency and the local oscillator frequency of ~8.1 GHz (Wilczewski, 18)—without a severe impact on the SNR. Moreover, this allows avoiding high losses during the signal's travel through the transmission line to the receiver, which is usually a coaxial cable (CH, 345). An important consideration for the downconverter is adjusting the gain of the LNA or downconverter itself to avoid saturation on the reception, particularly with short distances between the antenna and the transceiver (CH, 345). Moreover, a noise diode array (NDA) is an added analysis micro-element that is also under the influence of the M&C to measure and quantify system noise temperature at the LNA-downconverter chain of the downlink (Wilczewski, 17). The overall signal after down-conversion represents an IF of ~300 MHz before reaching the DTT.

As mentioned in §3.2.2, the DTT handles data processing for all downlink communications with LIC. The DTT takes the incoming telemetry data IF signal as an input and demodulates the carrier, subcarrier, and the symbol stream where it then converts the symbols to bits and performs frame synchronization and decoding (digitization). Additionally, the DTT also performs ranging correlation and provides carrier phase data for Doppler measurements. Considering modulation, the DTT is compatible with the PSK, BPSK, QPSK, and OQPSK schemes (Wilczewski, 24). Next, the DCD captures telemetry and tracking data from the DTT, manages the delivery of said data based on the grade of service and priority, and then catalogues the data for future reference (Wilczewski, 26). Now, there are a few more intermediate micro-elements before telemetry is received at the LIC MOC; however, since most of those micro-elements are offsite through JPL, those are reviewed in the GDS discussion (refer to §3.3.2). This concludes the discussion and breakdown of the GSA—DSN and DSS-17—for LIC. For a detailed mapping of the DSN and DSS-17 in LIC's MOA, refer to the accompanying flow-diagrams in §A.2 of the Appendix: Figures A-6a, A-6b, A-7a, and A-7b.

3.3) The Ground Data System

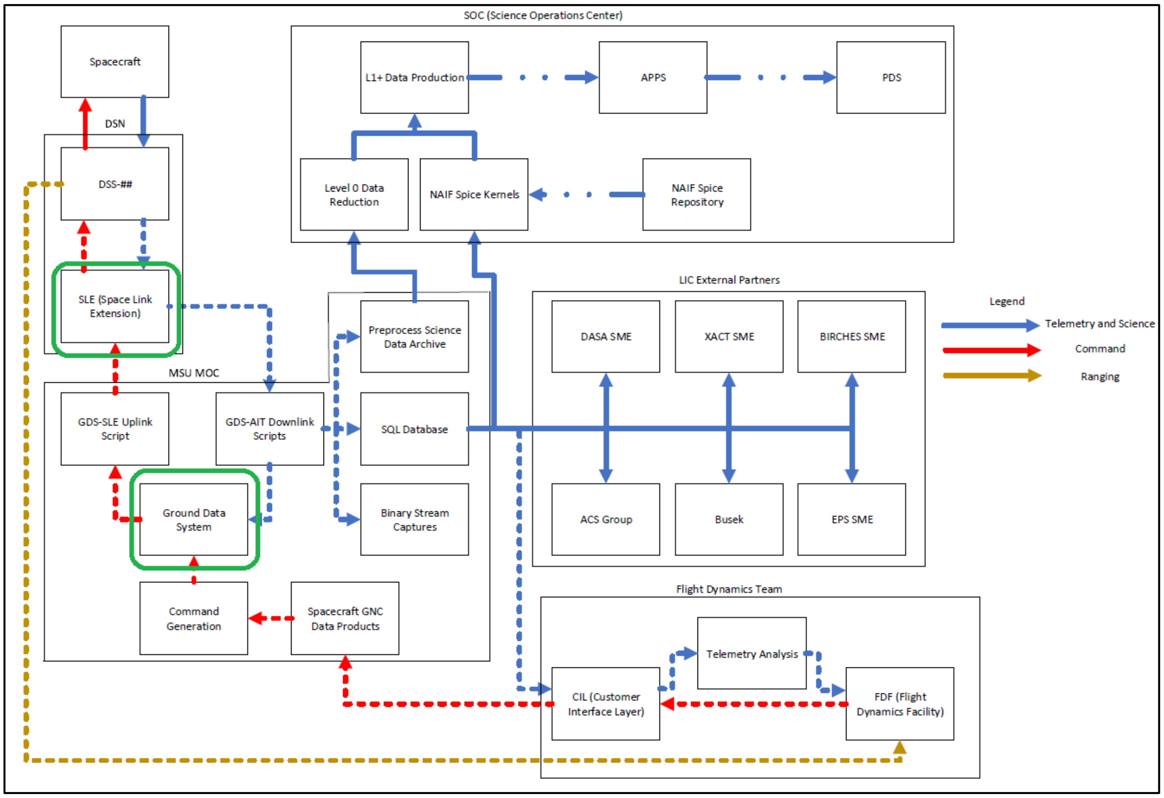


Figure 3-2. Macroscopic view of LIC's MOA. GDS macro-element highlighted in green.

Now that the GSA has been introduced as the macro-element that serves as the hub for all space-borne communications, emphasis shifts towards the macro-element that establishes the parameters for data management and provides the interfaces between the GSA and the MOC: the ground data system, or GDS. The GDS is analogous to highways that connect two major cities, city A to city B, with multiple routes, but each unique depending on their usage. Here, the linkage is between the communication and the control of the spacecraft, with the GDS serving as those highways. The GDS can be severed into two distinct avenues; therefore, continuing with the case of LIC, the same scheme is applied as from the GSA where two exchanges occur: command and telemetry. The GDS for LIC is ultimately managed and organized by the assets of JPL, including the data services and interfaces that link the DSN with the LIC MOC. Before diving into the layouts of the command and telemetry avenues of the LIC GDS, a brief on the mission service interface that links the DSN with LIC MOC is required.

3.3.1) ***The Space Link Extension***

Like systems engineering standards discussed in §1.2, data exchange or transfer related to spacecraft are standardized. For all intents and purposes, the main ideas surrounding this standardization are most likely for security protocols. With LIC being a spacecraft supported by the DSN, the GDS utilizes CCSDS Space Link Extension (SLE) interfaces and services (or service instances) between the DSN and LIC MOC. SLE is a standardized public medium of transferring command and telemetry data to and from the spacecraft. Since this is public domain, SLE takes many forms depending on its usage and management but must comply with the protocols set by the CCSDS. In general, SLE services must be configured for each application by means of service management (OICD, 9-1). For LIC, management of these service and interface protocols are left to the DSN SLE Gateway located at JPL DSOC.

As previewed in §3.2.2, there are two SLE interfaces that serve the exchange or transfer of command and telemetry data between the DSN and LIC MOC: the SLE forward command link transmission unit interface, or CLTU (FCLTU is also common), and the SLE return all frames interface, or RAF protocol, respectively (OICD, 3-1). It is important to note that LIC also makes use of DSN tracking services, but for now that is held off until the next section (refer to §3.3.2). First, macroscopically, the CLTU and RAF interfaces are considered services; however, secondly, there can be multiple "instances" for each service. More specifically, although the GDS involves at most three distinct service types (RAF, CLTU, and tracking), the SLE framework embeds the concept of a service instance to distinguish data transfers that require special or specific handling inherent to the managerial or technical facets of the GDS. For example, the spacecraft as the destination for commands and the source of telemetry; the DSS being used to radiate commands and acquire telemetry; the space link physical channel (regulated frequency); the SLE service type; the SLE delivery mode; and the SLE instance identifier, are used to differentiate, and distinguish data transfers (OICD, 3-2).

Consider a nominal operational scenario. An LIC MOC user, such as a spacecraft operator, initiates the provision of service for a session by "handshaking" with the DSN SLE service provider—DSOC in this case—for each service instance applicable to the session. This handshake is like a bind that connects the LIC MOC (initiator) and the DSN (responder), and this must be performed for each applicable service instance. One of the parameters required in this handshake is the service instance identifier, which is a unique name that enables one service instance to be distinguished from another (OICD, 3-3). With every service instance, there are several attributes that affect the provision of service for that service instance. For example, RAF service instances can have different delivery modes. While there are several modes defined by the RAF specification, those most important for DSN users like LIC are the online complete (data delivery in order of arrival) and offline (post-pass data delivery with specified start/end time) (OICD, 3-3). For a continuation of both CLTU and RAF, more specifically their impact on the format and contents of the command and telemetry data, refer to the discussion on the MOC macro-element (§3.4.1 and §3.4.3, respectively).

3.3.2) *Lunar IceCube and the Ground Data System*

Keeping in mind the pretext from §3.3.1 that the GDS supports two avenues with two respective interfaces: command (CLTU) and telemetry (RAF). Most of the activity for the GDS in centered at JPL DSOC; therefore, the GDS is centered between the flow diagrams of the DSN/DSS-17 and LIC MOC. First, consider command. Command is treated first due to its simplicity. While there are no distinct micro-elements between the MOC and the UPL for DSS-17 or any DSCC, the main highlight is the flow between them (refer to the GDS flow diagram in §A.2.3). Notice that commands are sent from LIC MOC straight through DSOC to the UPL of DSS-17/DSCC. DSOC serves as mediator of the handshake between LIC MOC and the UPL; therefore, commands from LIC MOC are affiliated with the CLTU interface and are radiated by DSS-17/DSSC to LIC. Commonly, personnel such as ground controllers associated with GDS tests can witness this handshake rather than spacecraft operators or flight controllers (OICD, 3-2).

Second, consider telemetry. The telemetry avenue of the LIC GDS does have distinct micro-elements. When LIC MOC requests telemetry, the following explains the flow related to telemetry. First, telemetry/tracking data is downlinked from LIC via a DSS station either at a DSCC or DSS-17 and imported to the DSCC/DSS-17 DCD for export to the JPL DCD at DSOC. Next at DSOC, the data imported from the DSN is split into two paths: one for telemetry and one for tracking. Focusing on the telemetry, data is sent from the JPL DCD to a database labeled the DSN SLE Gateway. Recall that this where the RAF interface is utilized to commence a handshake between DSN and LIC MOC for telemetry. The requested telemetry data is then forwarded from the SLE Gateway database to the LIC MOC where it is decomposed and translated into human readable data (OICD, 3-2).

Third, consider tracking. While the focus remains on command and telemetry, tracking data is important for the LIC SOC and EUF flows. Therefore, there are numerous micro-elements within the LIC GDS supporting tracking data. The following explains the flow related to LIC tracking data. First, as mentioned from the downlinking of telemetry data, tracking data joins telemetry. That combination of data is received at the JPL DCD at DSOC. Next, tracking data is routed to the tracking data delivery subsystem (TDDS). Before continuing with the tracking data, notice a second flow from the DSN to the GDS. This begins with the collection of antenna weather data (conditions upon tracking) from a DSS that is syphoned to the antenna controller assembly (ASA) for media calibration. Next, this data is exported to a media modeling micro-element at DSOC where the data is processed before being utilized to support concurrent tracking data from the TDDS. Returning to tracking data, the tracking data imported from the TDDS meets the media calibration data within the radiometric data conditioning subsystem, or RMDC. It is at the RMDC where both data are processed into tracking files. Lastly, these files are stored on a server/database called OSCARX which are then accessible by the Flight Dynamics Facility (FDF) at GSFC for non-live data. Live, raw tracking data is additionally transferred to FDF from TDDS via the Ground Interface Facility and its Special Function Gateway (GIF/SFG) (OICD, 3-2). This concludes the discussion and breakdown of the LIC GDS. For a detailed mapping of the GDS in LIC's MOA refer to the accompanying flow-diagrams in §A.2: Figures A-8a and A-8b. A more expansive breakdown of the LIC SOC and EUFs and its reliance on tracking data is presented in §3.5; however, the scheme of both command and telemetry continues into the next section focusing on the MOC macro-element where data management reaches its peak.

3.4) The Mission Operations Center

Considered the hub for mission planning, spacecraft/satellite commanding, and telemetry analysis and archiving, the mission operations center, or MOC, is the backbone of a MOA. While the GSA and GDS serve as the multi-structured catalyst between the spacecraft and mission personnel, the MOC serves as the venue for mission personnel to perform tasks, actions, and procedures in a complex and repetitious environment. The environment itself is typically a secured room enclosing a multitude of spacecraft operating equipment with several consoles staffed by subsystem experts; however, this is the MOC only at face value (SOA, 228). Underneath this visage is a multi-layered data highway representative of a "data backbone" distinguishing the MOC as the center of data management. A MOC is configured physically and technologically to meet mission objectives and spacecraft requirements; therefore, making it inherently complex from a managerial and technological perspective. It is the joint responsibility of the mission operations team and subsystem experts to minimize this complexity with the use of written procedures and operational training to achieve adequate simplicity. Since the consideration here is for a small spacecraft, size, scale, and scope are limited; therefore, complexity is also limited. However, this is a space mission, and complexity does hold in all circumstances regardless of the mission. The LIC MOC is shown as Figure 3-6, and a future configuration is given as Figure 3-7.

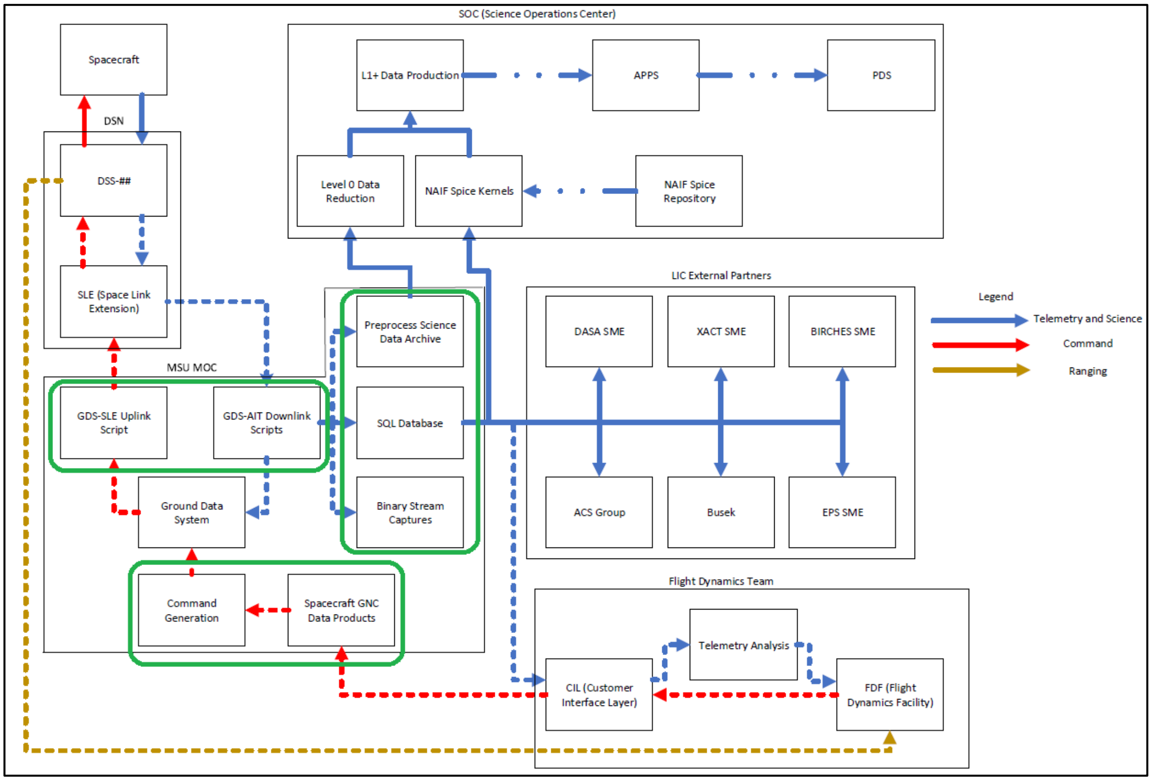


Figure 3-2. Macroscopic view of LIC's MOA. MOC macro-element highlighted in green.

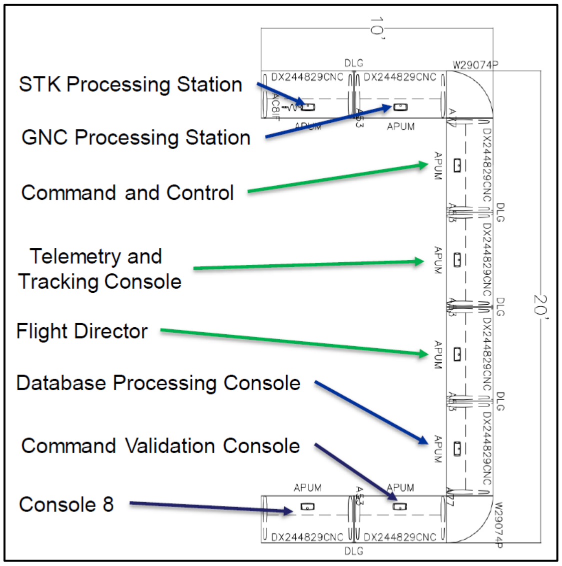


Figure 3-6 (left). LIC MOC and DSS-17 control center.
Figure 3-7 (right). Future LIC MOC configuration and console/station positions.

The MOC also serves as the venue for mission planning and testing. The mission operations team submits tasking requests for future spacecraft operations. With these tasking requests, command plans are generated and tested against a simulation to verify if those plans are sufficient with confidence. If not sufficient with simulations, those plans are tested with active engineering model hardware prior to approval for uplink. The same team is also responsible for downlink. Plans are generated to determine what telemetry and science data is necessary and what ground resources are available (SOA, 229). As mentioned, the MOC either has access to a GSN or its own ground station(s), so implementation of uplink and downlink can vary. However, the concept remains intact for each case. For example, the MOC submits the data necessary for commanding the spacecraft for uplink which includes commands and parameters settings for the payloads, a schedule of events for the flight computer (CDH), and ephemeris and pointing tables for the ACS along with its own timeline of events. For that same contact, the MOC also submits commands to downlink specific telemetry and science data. When the contact is complete, the data is sent back to the MOC by either a node on the GSN or its own ground station(s) (SOA, 229).

Before diving in further, a note of the proceedings is required. The following refers to LIC MOC and the configurations set by the mission objectives and spacecraft requirements. Note that LIC is a small spacecraft mission; therefore, not all the responsibilities are held within its MOC. Some responsibilities are held by outside entities, but that is held for a later discussion regarding the SOC and EUFs (refer to §3.5). LIC MOC involves two configurations: uplink (command) and downlink (telemetry). For each, there will be a general overview of its data interface followed by the breakdown of its physical data flow.

3.4.1) ***Uplink Configuration Part 1: The CLTU Interface***

As mentioned in §3.3.1, the LIC MOC utilizes the CCSDS SLE Forward CLTU command service interface for uplink. To begin, consider the macroscopic depiction for the uplink of commands: LIC MOC generates commands in the form of CLTUs on the ground which are then radiated to LIC for reception and processing. The program makes use of DFE, or the Direct-from-Earth, protocols that are an adaptation of the CCSDS telecommand standard for the generation of commands.[16] The macroscopic command protocol structure known as the command load is given by Figure 3-8. A command load consists of one or more uplink sessions, each of which conveys one or more command packages. LIC utilizes three different types of commands: critical, immediate, and file loads. Before diving into a general breakdown of each, discussion must build to how the CLTU becomes involved as an interface for commands and its relationship with the command load.

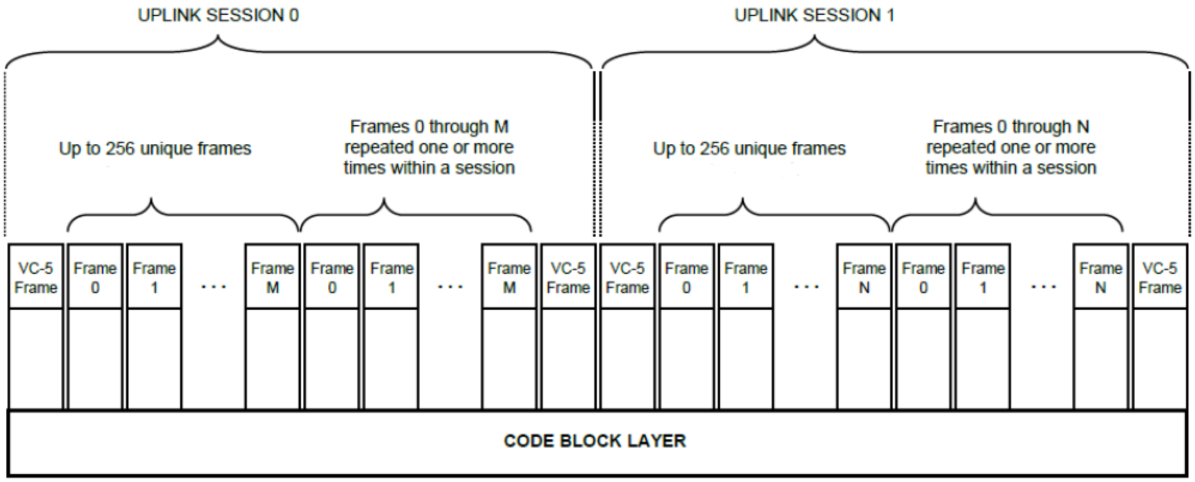


Figure 3-8. The command load for uplink sessions.^{16}

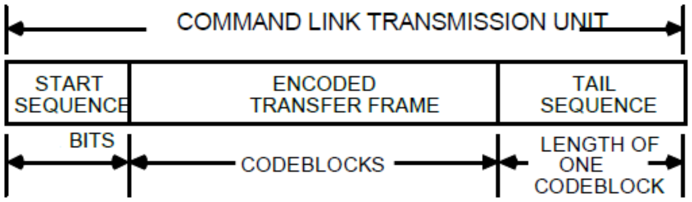


Figure 3-9. The structure of a CLTU (FGICD, 19).

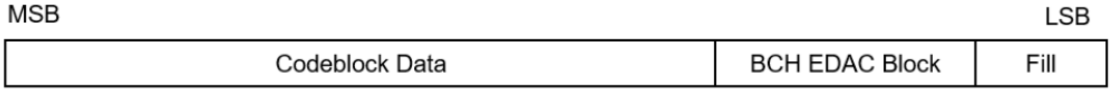


Figure 3-10. The structure of a code-block. MSB is the most-significant bit, while LSB is the least-significant bit (FGICD, 19).

A CLTU is partitioned into three parts: a start sequence, an encoded transfer frame, and a tail sequence (refer to Figure 3-9). The start sequence is essentially a group of bits that delineate the beginning of the uplinked transfer frame, while the tail sequence is another group of bits that mark the end of the transfer frame and the search for the next available transfer frame (FGICD, 44). The transfer frame is the standardized high-level command initiated by the MOC. Now, while the CLTU is defined by its division of parts represented as bits, CLTUs are tacked onto the command load's coding layer, where the CLTU is represented by "code-blocks" (refer to Figure 3-10). A CLTU transfer frame has no direct limit on the number of code-blocks it can use for the command package, except for how the code-blocks are utilized to convey the three different commands (FGICD, 19). However, code-blocks do have a limited bit-size. For example, consider a command package that contains enough bytes (8 bits = 1 byte) to utilize eleven code-blocks. The first ten are filled; therefore, the rest is deposited into an eleventh, but it does not fill up that entire code-block. For this case, the eleventh code-block is then "padded-out" with alternating 0's and 1's to indicate that the CLTU transfer frame content does not complete the last code-block. This is represented by the Fill block in Figure 3-10. Finally, the BCH EDAC Block denotes the error detection and correction (EDAC) utilized on the code-block. BCH represents the Bose-Chaudhuri-Hocquenghem method of error-correcting cyclic codes (FGICD, 19).

Now that the CLTU has been properly defined in context of the command load's coding layer, focus can now be directed towards the three command types. Uplink sessions for CLTUs encompass no more than 256 unique transfer frames. The beginning of each session is denoted by an acquisition sequence used to acquire bit synchronization on the uplink bit-stream. Additionally, a session is segmented into four discrete layers (from high level to low level). First, the coding layer as mentioned previously. Second, the application layer including the CLTU transfer frames assembled from the data portions of the code-blocks. Third, the command message layer including the transfer frame data portion of individual transfer frames. Last, the command package layer containing the information that is being transferred from the MOC to the spacecraft with no overhead (FGICD, 44).

With the use of the CLTU's transfer frame, the MOC can designate the type of commands as either critical, immediate, or file load using a virtual channel. A virtual channel is only a field of data that classifies the types of commands, and the spacecraft controls the processing and routing of those commands based on what fills the field. Critical commands execute immediately upon receipt and verification without requiring the use of flight software. A critical command is contained in one code-block with no frame error correction word (FECW) within the transfer frame located on the application layer (FGICD, 21). Refer to Figure A-1 for the complete graphical description of critical commands. Immediate commands follow the CCSDS space packet protocol (SPP) format conventions like that for non-critical commands (refer to §3.4.3 for more on SPP). Unlike the special case with critical commands where only one code-block can be utilized, an immediate command is not limited by the number of code-blocks it utilizes to convey the command package; however, there is a specified maximum length present for each immediate command. A FECW is also present. Additionally, like non-critical commands, if a command parameter field is not an even number of bytes, the field is padded with 0's. However, if the fields are file paths, the fields are padded with "nulls" to reach the specified maximum length. Note also that immediate commands are not executed on the spacecraft "immediately" in all circumstances. "Immediate" here means on reception from the subsystem as there is no strict real time to interrupt an executing task (FGICD, 16). Refer to Figure A-2 for the complete graphical depiction of immediate commands.

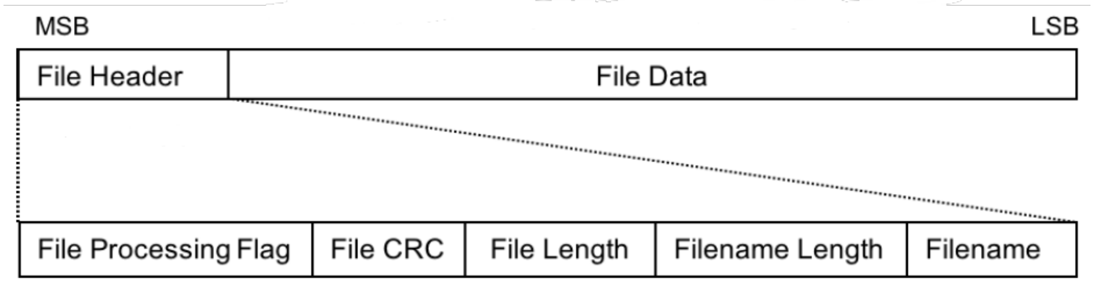


Figure 3-11. Format of a file load command package (FGICD, 24).

Finally, file loads are configured towards the CCSDS File Delivery Protocol, or CFDP. These file loads are specially structured SPP messages generated by a ground station CFDP engine. Once they are passed to the spacecraft, the on-board CFDP engine processes them before execution (FGICD, 16). An important note: a single uplink session from the MOC can contain either one or more SPP messages, or a single CFDP session. Both are limited to the maximum of 256 unique transfer frames. Therefore, for file loads, a single file load cannot exceed 256 frames. Consider a detailed command package layer in Figure 3-11. Figure 3-11 shows the format for a file load command package that includes a file header and file data. Additionally, the file header includes the following groups of bits: a file processing flag that indicates the overwriting status of the file and the file type; a file CRC, or cyclic redundancy check, to verify the integrity of the file contents; the file length in bytes; the filename length; and the filename itself (FGICD, 23). In conformance with CCSDS standards, if the MOC requires to upload a file longer than the allocated 256 transfer frames, an option granted to the MOC is to split the original file into multiple files of lesser length, upload the files to the desired directory, then concatenate the files to recreate the original file. The action of the splitting of the file load is depicted by Figure A-3, where the command package layer is contained in Frame 0 of the first CLTU and Frame 1 of the second CLTU (FGICD, 24). Refer to Figure A-3 for a complete graphical depiction of CFDP file loads.

3.4.2) *Uplink Configuration Part 2: The Uplink Flow*

As mentioned in §3.4.1, a single uplink session can contain either SPP commands or a single CFDP file load. First, consider the easiest command flow concerning SPP commands. These are basic commands that are generated separately from a graphical user interface, or GUI, accessed from LIC MOC using NASA's AMMOS Instrument Toolkit (AIT). AIT is a collection of ground software for command and telemetry data, and was used throughout LIC flight software development, integration, and testing. Essentially, the SPP commands are routed as "click-button" commands that simplify the execution process through the AIT GUI. The AIT software exports a spacecraft command to the Encode block, where a script will then encode the commands and format them to meet the CLTU interface standard before being exported as forward CLTUs to the

UPL of a DSS.

Next, consider CFDP files loads. Files generated by the LIC MOC are meant to take the place of routine commands that would normally be executed as SPP commands. Essentially, a collection of commands that upon upload would allow LIC to execute tasks autonomously. Therefore, consider some files. These files are exported to the Sender block, which is a script that encompasses the LIC MOC's CFDP engine capabilities to format files being uplinked to LIC. The CFDP engine Sender then exports the formatted files to a virtual machine for external purposes (refer to §3.4.4 for more detail on the virtual machine). The virtual machine then exports the formatted CFDP files to Encode and formats them to the CLTU interface before being exported as forward CLTUs to the UPL of a DSS. This concludes the consideration of the uplink configuration for the LIC MOC. For a detailed mapping of the uplink configuration for the MOC in LIC's MOA, refer to the accompanying flow-diagram in §A.2: Figure A-9a. Next, consideration turns towards the downlink configuration.

3.4.3) ***Downlink Configuration Part 1: The RAF Interface***

As mentioned in §3.3.1, the LIC MOC utilizes the CCSDS SLE RAF telemetry service interface. To begin, consider the macroscopic depiction for the downlink of telemetry: LIC MOC requests telemetry using the RAF interface and classifies telemetry data based on channelization. There are three types of telemetry data: pre-channelized, ground-channelized, and non-channelized. Pre-channelized data is data channelized onboard LIC that provide spacecraft and instrument engineering health and status information. Ground-channelized data is primarily used for engineering information generated by instruments and is channelized on the ground via a ground map. Non-channelized data is used for event reporting; conveyance of memory readouts, logs, and data dumps; and for transportation of science data (FGICD, 38). The program makes use of the DTE telemetry protocols that are an adaptation of the relevant CCSDS standards, such as SPP and the Advanced Orbiting System (AOS) Space Data Link Protocol.

Like the command structure in §3.4.1, the telemetry strcuture has four distinct layers. The use of code-blocks is reintroduced on the first high-level coding layer, along with transfer frames; however, what is new is the usage of the term "turbo." Recall that transfer frames are a fixed-length data structure that allows reliable transmission of telemetry, even in noisy environments. Turbo encoding is a class of encoding schemes that aid in the mitigation and reduction of noise to produce a robust data structure for the downlink of telemetry (FGICD, 28). The specifics of turbo encoding are beyond the scope of this writing but is essentially implemented as a recursive convolution encoder. The turbo encoded symbol stream is parallel in nature to the code-block layer in the command structure from §3.4.1. Here, the stream is segmented into turbo code-blocks, also known as CVCDUs. The turbo code-blocks contain an encoded transfer frame (VCDU) that contains the first highlights of the telemetry data and a Trellis termination—a coded reset switch for the encoder. "Turbo transfer frame" is one name to give this encoded transfer frame. Another name is the AOS transfer frame, and LIC, along with LIC MOC, utilize two formats resulting from the turbo encoding scheme. The main difference in their format is the sizing of the frame data field: a small format and a large format. With that said, stepping down into the application layer, the format of the AOS transfer frame is visited next.

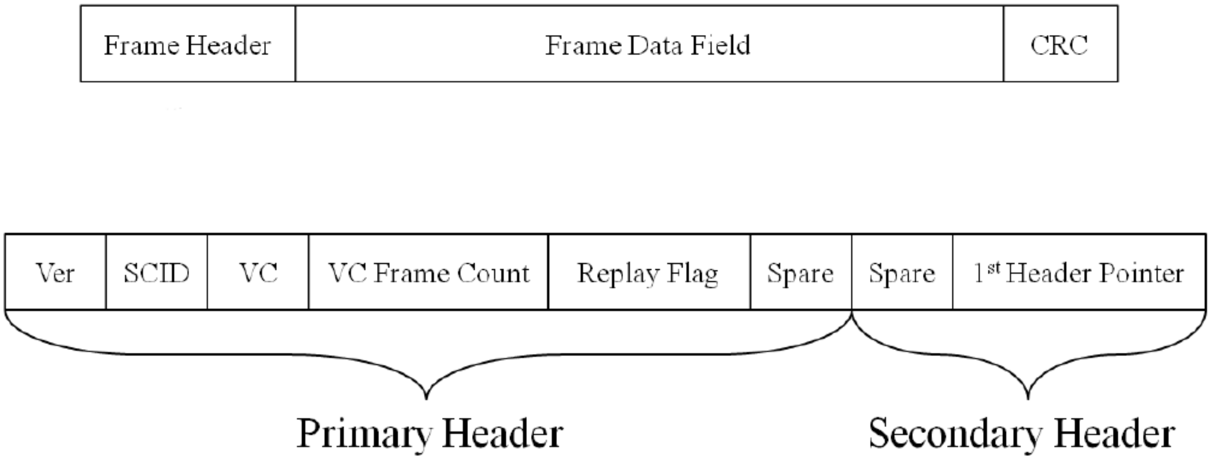


Figure 3-12. Format of Turbo/AOS Frame (FGICD, 31).

It is important to note that the only difference between the two AOS transfer frame formats is in the number of bits allowed in the frame data field as depicted in Figure 3-12. Now, the option to switch between formats depends on the state of the MOC and the spacecraft with respect to available ground stations. Diving into the AOS frame making up the application layer, there are three segments: the frame data field as mentioned, the frame header, and a CRC. The frame header denotes information about the contents of the frame data field: the version number for the AOS structure; the spacecraft identifier (SCID) that tags the source of the telemetry data; the virtual channel (VC) of the telemetry data; the virtual channel frame count that sequentially counts the number of transfer frames within a virtual channel; a replay flag to determine if the data has or has not been retransmitted; a couple of spare fields; and the first header pointer, which contains an incrementing binary counter pointing to the number octet containing the first octet of the first packet header (FGICD, 31). Throughout downlink, the spacecraft ensures that the frame counts within the virtual channel are monotonic. Additionally, the MOC rejects any downlink transfer frame that mismatches the standardized transfer frame fields. In the case when no telemetry data is available for transmission, the spacecraft generates idle frames to maintain a continuous downlink data stream (FGICD, 32).

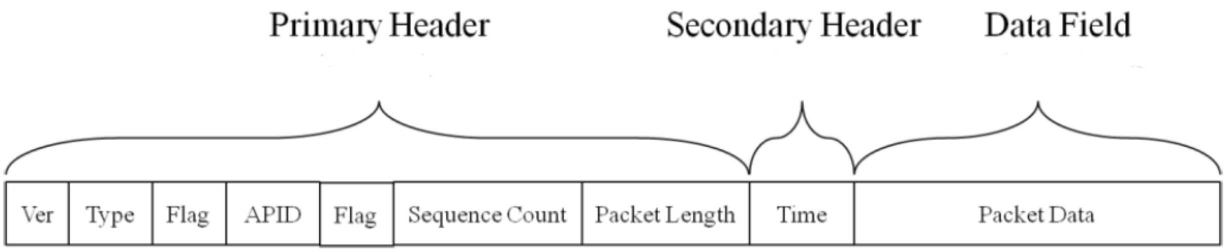
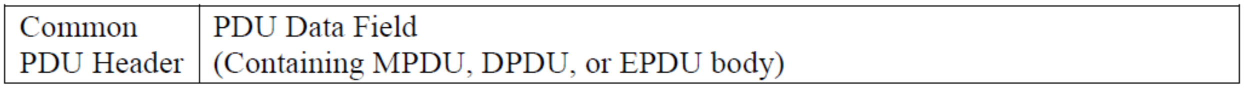


Figure 3-13. Format of SPP packet (FGICD, 33).

Moving lower into the message and package layers, the telemetry data begins to show itself. Skipping the message layer for a moment, telemetry can take one of two formats in the package layer: SPP packets or CFDP Protocol Data Units (PDUs). First, consider SPP packets. The purpose behind packets is to transport logically related datasets to and from the spacecraft. The SPP packet format is given in Figure 3-13. SPP packets have a fixed length of bytes, and contents differ on a case-by-case basis. SPP packets contains two headers: primary and secondary. These headers, like that of the AOS header for the AOS frame, are there to convey information about the contents of the packet data field. The contents of the primary header are as follows: the version number of the of the SPP; the packet type (always telemetry); a secondary header flag to show if a secondary header is present; the application process identifier, or APID, that indicates the kind of packet (subsystem specific designations); grouping flags to identify if a packet is part of a segmented series or just a non-segmented (single) packet; the source sequence counter that counts the number of packets generated; and the length of the packet. The secondary header only includes the time of packet generation in seconds and sub-seconds (FGICD, 33).

As mentioned previously contents can differ on a case-by-case basis. The MOC verifies that all SPP packets match the formatting set by the CCSDS standards, and that the proper designations have been given within the headers to validate that these are genuine SPP packets requested from the spacecraft. For the case of packet size, if telemetry for a certain APID is too large to contain in one packet, the spacecraft then segments the data across multiple packets. A series of packets representing one dataset is indicated by each individual packet's header; therefore, all should have matching APIDs and time stamps, continuous sequential source packet sequence counts (SPSCs), and packet segmentation flags set as either “first”, “continuous”, or “final” (FGICD, 34). Packet data from the message layer is then transposed and segmented into multiple AOS transfer frames when stepping up to the application layer. Finally, the AOS transfer frames are integrated into turbo code-blocks encompassing the coding layer. For the case when no data is available for transmission, the spacecraft generates idle packets to pad-out the transfer frames. Finally, the MOC rejects and reports unidentifiable packets with unknown APIDs and reports missing packets detected by gaps in the source packet sequence count for each APID (FGICD, 34).



Common PDU Header	PDU Data Field (Containing MPDU, DPDU, or EPDU body)

Figure 3-14. Format of CFDP PDU (FGICD, 34).

Next, consider CFDP PDUs. CFDP is used to transport engineering data as a data product by utilizing the same packet data fields as the SPP packets in the package layer. However, this field contains one or more CFDP PDUs. The PDUs are formatted per the CFDP specification and consist of a common PDU header followed by a specific PDU data field (refer to Figure 3-14). LIC utilizes three types of PDUs: metadata PDUs (MPDUs), file data PDUs (DPDUs), and end-of-file PDUs (EPDUs). A telemetry data product becomes a group PDUs by creating one MPDU, one or more DPDUs, and one EPDU. By convention, the MPDU is the first PDU produced for the telemetry that contains general and specific information about the contents of the data product. By requirement, the MPDU is contained in one packet. Next, the telemetry is then encased in one or more DPDUs, where the PDU boundaries are subject to the maximum PDU size and/or data product record boundaries (FGICD, 34). By requirement, each DPDU is contained in one packet. Finally, the EPDU is the last PDU produced for the telemetry and contains additional specific information about its contents. Contents of the PDU header serve a similar purpose as those of the AOS and SPP headers but holds many more parameters conveying information about the PDU data field (FGICD, 35). For example, the source entity ID and the destination entity ID, identifying LIC and the LIC MOC, respectively.

Before moving on to the LIC MOC downlink flow, one final application is required. Since LIC is a unique case of having a subscription with the DSN, including with its own ground station, DSS-17, there is one more data-layer to consider for downlinking telemetry. Note that this is exclusive to spacecraft subscribed with the DSN. For missions that require telemetry, such as LIC, the standard formatted data unit (SFDU) is utilized. A telemetry SFDU is a self-identifying, self-delimiting data structure that is used to encapsulate a portion of the telemetry data acquired from a spacecraft by the DSN for delivery to a mission. Typically, each SFDU encapsulates one transfer frame. A given sequence of telemetry SFDUs may encapsulate one physical channel from a spacecraft, or it may encapsulate only a part of a physical channel (e.g., a virtual channel for LIC). Each SFDU also contains additional information related to the telemetry data encapsulated in the SFDU.[17]

The SFDU as a concept that was originally established by the CCSDS to provide a standardizes and internationally recognized methodology for information exchange. Now, the SFDU specification required by the DSN is slightly different than the international specification set by the CCSDS (Telecomm, 1-2). Here is how the SFDU is incorporated into the downlink process. First, once the RF downlink from the spacecraft is downconverted, amplified, demodulated, and bit synchronized to produce a stream of digital symbols, depending on the requirements of the mission, further processing may be performed by the DSN (Telecomm, 2-1). The minimum required processing that is needed to acquire the physical channel, or the stream of digital symbols, include, if applicable, demodulation of the RF carrier and subcarrier, symbol synchronization, differential decoding, and convolutional decoding (Telecomm, 2-2). For the case of LIC, turbo coding is utilized and requires that the frame synchronization be performed before the decoding from symbols to bits is to be conducted. It is also strongly recommended that—whenever applicable and under ordinary circumstances—the mission use a CRC in the transfer frame, which LIC does incorporate. Additionally, only frames that are successfully synchronized and decoded are delivered to the mission (Telecomm, 2-3).

The resultant telemetry stream is annotated with DSN status, configuration, and performance information, which is then formatted into telemetry SFDUs. The turbo telemetry SFDUs are transported from the DSS to the MOC or other specified sites where they are made available to mission controllers, engineers, and scientists. To protect against potential data losses that may be caused by certain classes of GDS failures, all telemetry SFDUs are recorded by the DSN and are recoverable from that recording for some time thereafter (Telecomm, 2-1). Since the SFDU interface is exclusive to spacecraft on the DSN, specifics are left to the reader. While details could be provided, a significant majority of the LIC MOC content also applies to a significant majority of other MOCs for other missions (not just deep space missions). A very small fraction are realistically subscribed with the DSN.

3.4.4) *Downlink Configuration Part 2: The Downlink Flow*

As mentioned previously, downlinking of telemetry involves two formats: SPP packets and CFDP PDUs. The LIC MOC downlink flow is configured to support both formats. Once the handshake (bind) between the SLE Gateway and LIC MOC is established via the RAF interface, telemetry data is accessible to the operators. All spacecraft telemetry is routed to a main downlink script that acknowledges the SFDU, AOS, SPP, and CFDP standards. The script ultimately sorts and filters all telemetry based on two parameters: classification (packets or PDUs) and bundle protocol (bundled or non-bundled telemetry). The bundle protocol is an experimental protocol included with LIC's mission objectives and is essentially a technology demonstration for deep space spacecraft under NASA's discretion. Details of this protocol are beyond the scope of this writing, but the protocol is included to prevent the negligence of a data flow branch that is well established inside the LIC MOC.

First, consider non-bundled telemetry. Non-bundled SPP packets are exported by the downlink script to a "Split" block. The Split is a script that carbon-copies the imported data and splits it into separate streams. For now, consider these: one to the graphical user interface (GUI), one to the binary string capture (BSC), and one to the Archive. The GUI is an interface that allows LIC operators to monitor telemetry and send commands via the AIT server. The BSC is a database consisting of copies of raw (binary or machine language) data streams. Essentially, the same human-readable data, but at the lowest level. This data can be revisited and re-streamed for testing and analysis purposes. The Archive is a database storing all human readable telemetry data for the lifetime of LIC's mission. Non-bundled CFDP PDUs are imported into the same CFDP engine used to uplink command files, but now a Receiver script is used to acquire incoming PDUs for de-formatting. The telemetry is then exported from the Receiver and imported into an Ingest script that preps the data for deposit into the Archive.

Next, consider the stream for bundled telemetry. Bundled telemetry is imported into a virtual machine aligned to the technology demonstration of the bundle protocol. A Query is utilized to keep track of the number of “bundles” received from LIC throughout the stream, and LIC MOC generates acknowledgements of reception. LIC MOC then uplinks these acknowledgements to LIC using the same process through the Encode script to the UPL. Bundled SPP packets are imported to the same Split script and carbon-copied to the GUI, BSC, and Archive. Meanwhile, bundled CFDP PDUs are collected by the Receiver script. Now, the only difference for bundled PDUs is that they are imported to AIT rather than straight to the Ingest script before deposit into the Archive. This concludes the discussion and breakdown of the LIC MOC for both the uplink of commands and the downlink of telemetry. For detailed mapping of the downlink configuration for the MOC in LIC’s MOA, refer to the accompanying flow-diagram in §A.2: Figure A-9b. Next, LIC’s external user facilities and science operations center are introduced as the final macro-elements of the LIC MOA.

3.5) External User Facilities and the Science Operations Center

The final macro-element under consideration for a deep space small spacecraft-based mission are the inclusion of external user facilities, or EUFs, and the science operations center, or SOC. While most of the data management that occurs behind the scenes is prevalent from the LIC MOC as told by §3.4, the EUFs and SOC are vital to the generation and management of high-level data products, such as navigation and science. Usually, EUFs take the form of companies employing their own SMEs, or subject matter experts, as ambassadors for subsystems developed for the mission by the company. These are either off-the-shelf or custom made, but either way provide a lifeline from program management to the company itself. However, an EUF doesn’t have to be a sponsored company, it could also be a national space agency. Space agencies have their own pool of SMEs that provide technical assistance with subsystems and contribute a multitude of assets within the allocated constraints of the sponsored program. On the other hand, a SOC is the focal point for all mission science data and data resources (SOA, 229). The program science team stores and analyzes the data to confirm if findings are parallel with that of hypotheses and objectives formed during mission conception.

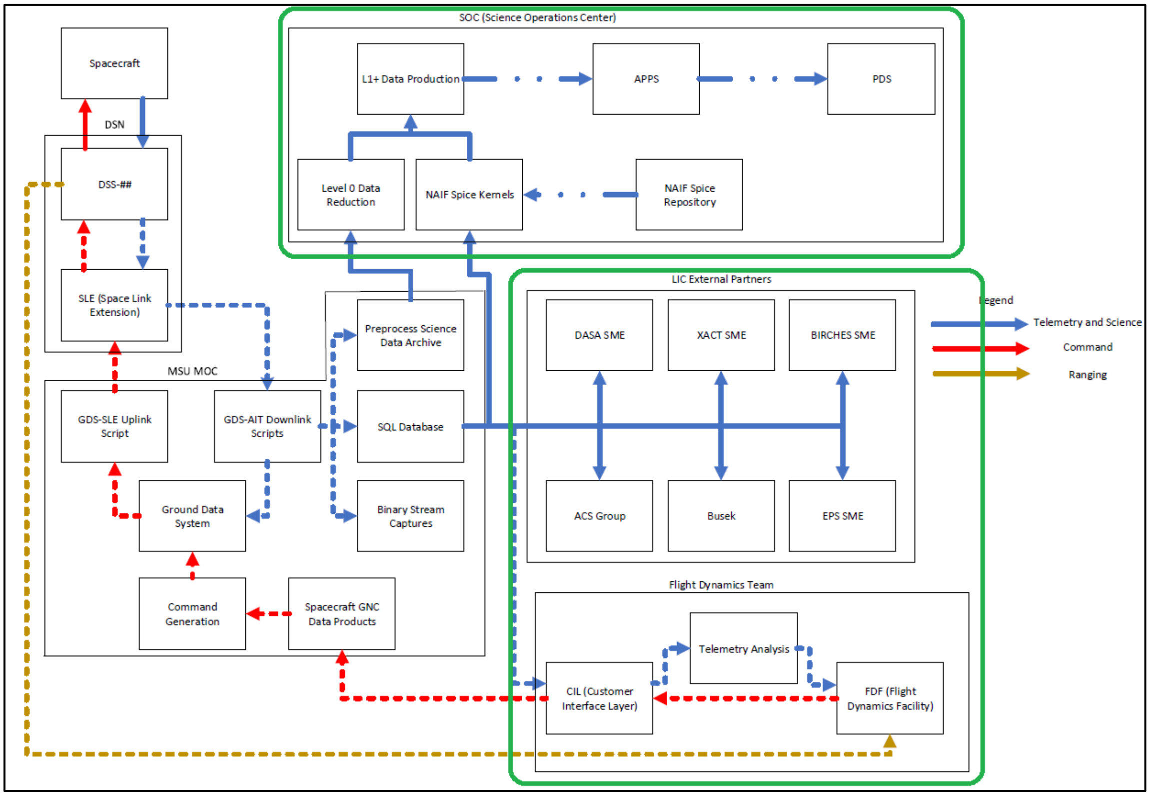


Figure 3-2. Macroscopic view of LIC's MOA. EUFs and SOC macro-element highlighted in green.

3.5.1) *Lunar IceCube and EUFs*

There are multitude of EUFs that are involved with a MOA, and the usual candidates to consider are the producers of subsystems. Usually when a mission program has been granted an opportunity, the program starts scouting for subsystems on the market that meet spacecraft requirements. In the process, the company provides a sort of special customer support to the program. By providing SMEs from the company itself who know about the specialized subsystem in question, the SMEs act as liaisons between the company and the program itself. Simply put, the company becomes a sponsor for the spacecraft just by means of a simple transaction. While this sponsorship provides short-term satisfaction for the program, it provides long-term benefits to the company producing the subsystems as it garners more credibility in the aerospace industry for providing a unique and specialized subsystem. The same can also be said for space agencies, such as NASA and ESA. The only difference here is that space agencies are usually the reason smaller spacecraft missions can make it past conception and design and into the actual building, integrating, launching, and operating of the spacecraft. Agencies ultimately help a program evolve into something greater than before. SMEs and resources are provided to the program in the same way a sponsored company would provide its resources.

For the case of LIC, it is both company and agency sponsored. LIC at its highest level is a NASA sponsored spacecraft utilizing the SMEs and resources of the DSN and FDF at GSFC. While the DSN has been extensively covered in §3.2.2, the FDF is a NASA sponsored EUF. FDF is involved with trajectory analyses, orbit determination and maneuver planning, determination of view periods and station locations, tracking data from DSN, clock/frequency offsets, media calibration data during scheduled tracks, and real time monitoring at GSFC. In return, FDF provides ephemeris prediction and reconstruction, navigation tracking requests, and maneuver interface and light time files to name a few. Starting with uplink, FDF provides the planning, data, and resources necessary for command generation by LIC MOC through a customer interface, or CI. LIC MOC pulls the necessary tools from the CI to generate guidance, navigation, and control (GNC) data products that aid in the generation of spacecraft commands. Next, consider downlink. The Database micro-element was not introduced in the discussion for LIC MOC due to its specific nature with FDF; however, it is integral in the exportation of telemetry data to FDF. The Database micro-element is a fourth branch off the Split that feeds telemetry through the CI to FDF, and by doing so FDF can conduct its telemetry analyses by pulling from the CI the necessary telemetry data. Refer to Figures A-9a and A-10 for a detailed mapping of the uplink and downlink for EUFs in LIC's MOA, respectively.

3.5.2) *Lunar IceCube and the SOC*

While SOCs can be considered off-site entities for small spacecraft missions, modernity has made them virtual or cloud-based for ease of access by the public. Spacecraft science teams usually incorporate a multitude of personnel, including those from companies, space agencies, and the program itself. The main goal of the science team is to provide various levels of science data through the means of collection by the spacecraft payload and distribution by the SOC. The term "levels of science data" are synonymous with the CLTU and RAF structures, except in reverse. The high-level data products are the most refined data sets that provide correlations and lead SMEs towards profound conclusions, such as images, graphs, and charts. The lowest-level data products (Level 0 data, or L0) are the original "raw" science data from the payload that requires iterative processing to achieve a distinguishable result before being upgraded to the next level. Moreover, ancillary engineering data accompanies the science data to signify the status of the spacecraft in terms of state, location, and orientation. This ancillary data is basically telemetry and are formatted into NAIF SPICE kernels before uniting with L0 data to form Level 1, or L1, data. These "kernels" or "kernel files" are composed of the navigation and ancillary data for precision observation geometry.

LIC will utilize NASA's Planetary Data System (PDS) by means of the APPS, or AMMOS-PDS Pipleine Service, for the transfer storage of all science and ancillary data that is obtained throughout the lifetime of the mission. A preprocessed science data archive (refer to Figure A-9b, fifth branch added to Split) is included in the LIC MOC to house the entire library of Level 0 science data before it is processed by the SOC with other data analysis resources. The

SOC also provides input for command sequence generation and planning to acquire new targets of opportunity for observation. Refer to Figures A-9a and A-11 for SOC involvement in uplink and downlink, respectively. This concludes the discussion and breakdown of a MOA in its entirety and showcasing the LIC MOA as an example of a possible configuration that can be suitable for Earth and deep space missions for small spacecraft. The next chapter pushes the practicality of the MOA using V&V methodology with the LIC MOA subjected to this methodology.

4) Lunar IceCube and V&V

Now that the entire construction and mapping of a MOA has been communicated, it must be tested to prove it is a competent model for the practicality of daily operations. Specifically, the verification and validation, or V&V, of the MOA. Recall that the systems engineering process is full of iteration, and that iteration is purposeful to exploit any product's deviations from the concentrated requirements and mission objectives. V&V is the tool that systems engineers utilize to confirm that the conceptualized, designed, built, integrated, launched, and/or operated product does what its intended. The words "verify" and "validate" are often confused, eliciting inaccurate, synonymous definitions. To bring clarity, here are the definitions as recorded from Dictionary.com (the word "valid" is used in place of "validate" to remove the case of parallel definitions):

> Verify: 1. to prove the truth of, as by evidence or testimony; confirm; substantiate
> 2. to ascertain the truth or correctness of, as by examination, research, or comparison
> Validate: 1. to make sound, just, or well-founded
> 2. to produce the desired result; to make effective

As is worded, "verifying" has a closer tie with "truth" than "validating". "Validating" involves judgment based on effectiveness and solid arguments. Here are two excerpts from NASA's systems engineering competency model describing "product verification" and "product validation," respectively (NASA SE, 14-15):

> "Proving the end product conforms to its requirements. This includes preparing for the verification efforts, analyzing the outcomes of verification (including identifying anomalies and establishing corrective actions), and preparing a product verification report providing the evidence of product conformance with the applicable requirements."
>
> "Confirming that a verified end product satisfies stakeholders expectations for its intended use when placed in its intended environment and ensuring that any anomalies discovered during validation are appropriately resolved prior to product transition. This includes preparing to conduct product validation, performing the product validation, analyzing the results of validation (including identifying anomalies and establishing recommended corrective actions), and preparing a product validation report providing the evidence of product conformance with the stakeholder expectations baseline."

From the competency model, the verification of a product is tuned towards the product conforming to the requirements, while the validation of a product is tuned towards the product satisfying stakeholder (customer) expectations. What these definitions describe is that a product presents truth in the form of confirmation and substantiation that it itself is "the product" as defined by the requirements. Then, "sound, just, [and/or] well-founded" arguments or demonstrations present "the product" as the "the product for this mission" as defined by objectives set forth by customer expectations.

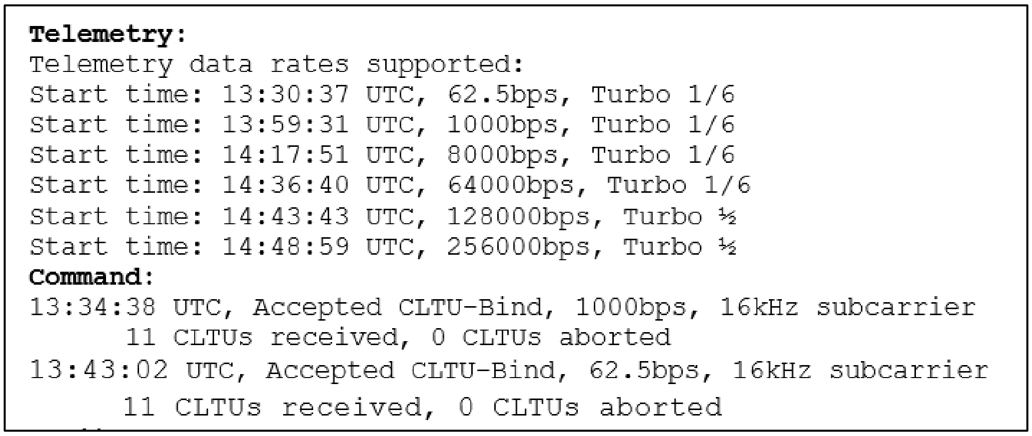
Anything can take the place of "product" in the previous explanation, so a spacecraft would be the default consideration; especially considering a space mission. A MOA is an indirect product—a system in reality—of the spacecraft. It too must be subjected to the processes of V&V to function in conjunction with its spacecraft. Most of the V&V that occurs with a MOA relates to the testing of the various macro and micro-elements and the interfaces between them. Before the launch of LIC, the LIC MOA was subjected to various tests to show the accuracy and practicality of the system during operations. These tests were a sort of "rite of passage" to prove that the system is ready for live operations just after LIC is released from SLS. In addition, ORR is the validation of the culmination of the LIC and the LIC MOA systems, and it too is presented in this chapter as a practical “certification” on the part of the sponsor that its expectations were satisfied.

4.1) Lunar IceCube and Ground Data Systems Testing

Consider the LIC GDS as examined in §3.3 using Figures A-8a and A-8b. The purpose of a GDS test was to verify the end-to-end command and telemetry connectivity between the DSN and the LIC MOC. While the GDS tests possessed the same basic procedures, each varied their use of a DSS. In fact, since LIC utilizes DSS-17 as its ground station, GDS tests were divided into two classifications: DSS-17 specific and outside DSS specific, such as DSS-36 at CDSCC in Canberra, Australia. GDS tests can also be classified as their integration into a MSPA, or multiple spacecraft per aperture test. This allows DSN nodes to support multiple spacecraft within the same beam that schedule to share the same node with each spacecraft downlinking to a separate receiver. Now, since these GDS tests are a form of V&V, their duration varies from swift to substantial. As a system, the LIC MOA is expansive; therefore, anomalies and faults can occur unexpectedly with the smallest element creating a cascading domino-effect in the process. A GDS test can involve multiple iterations of the same procedures to achieve the goals of the test set forth by JPL DSOC. With that said, the following lists the goals for a common GDS test with a DSS:

1. Establish a command bind from the LIC MOC to the DSN
2. Send a command from the LIC MOC to the DSN
3. Establish a telemetry bind from the LIC MOC through the SLE Gateway to DSOC at JPL
4. Confirm receipt of telemetry at LIC MOC
5. Confirm full telemetry processing capability into GDS at LIC MOC
6. Execute clean telemetry and command unbind

As explained in §3.3.1, the handshake or bind between the LIC MOC and the DSN is to verify the interface, whether it be for telemetry or command. The LIC MOC was expected to bind to the DSN SLE Gateway portal for RAF services and to the DSS UPL for forward CLTU services. Therefore, two binds are occurring simultaneously in a GDS test. Additionally, there are specific data rates configured for each interface set by the program. These data rates are measured in bps (bits per second) and sym/s (symbols per second). LIC commands from the DSN occur at two data rates: 62.5 bps and 1 kbps. LIC telemetry from the DSN can occur at five data rates utilizing two different turbo encoding schemes: 62.5 bps (374.99 sym/s) at Turbo 1/6; 1 kbps (6000.89 sym/s) at Turbo 1/6; 8 kbps (48076.9 sym/s) at Turbo 1/6; 64 kbps (378787.3 sym/s) at Turbo 1/6; 128 kbps (255102 sym/s) at Turbo ½; and 256 kbps (520833.1 sym/s) at Turbo ½. Figure 4-1 is a segment of a mission services activity report with DSS-36 that reveals the implied successful bind on telemetry at all data rates at specific times and the CLTU-bind status with the number of CLTUs received at DSS-36.



```
Telemetry:
Telemetry data rates supported:
Start time: 13:30:37 UTC, 62.5bps, Turbo 1/6
Start time: 13:59:31 UTC, 1000bps, Turbo 1/6
Start time: 14:17:51 UTC, 8000bps, Turbo 1/6
Start time: 14:36:40 UTC, 64000bps, Turbo 1/6
Start time: 14:43:43 UTC, 128000bps, Turbo ½
Start time: 14:48:59 UTC, 256000bps, Turbo ½
Command:
13:34:38 UTC, Accepted CLTU-Bind, 1000bps, 16kHz subcarrier
      11 CLTUs received, 0 CLTUs aborted
13:43:02 UTC, Accepted CLTU-Bind, 62.5bps, 16kHz subcarrier
      11 CLTUs received, 0 CLTUs aborted
```

Figure 4-1. Segment of a DSS-36 mission services activity report showing LIC MOC's live telemetry times, rates, and turbo encoding scheme. Additionally, the times and number of CLTUs received by DSS-36.

For the MOC in reference to §3.4 and Figures A-9a and A-9b, the active elements under testing for GDS tests are AIT, Encode, the Downlink Script, and the GUI. For the uplink portion of the GDS test, the Encode script initiates the command, or CLTU, bind with the DSS UPL. Next, all commands are initiated by operators using the GUI and are verified side-by-side using the Encode script. Last, the DSS reports the reception of the CLTUs. For the downlink portion of the GDS test, the execution of the Downlink Script allows operators to view the live stream of telemetry/symbols provided by the DSS and gives a visual interpretation of the data rates utilized.

4.2) Lunar IceCube and FDF-to-MOC Testing

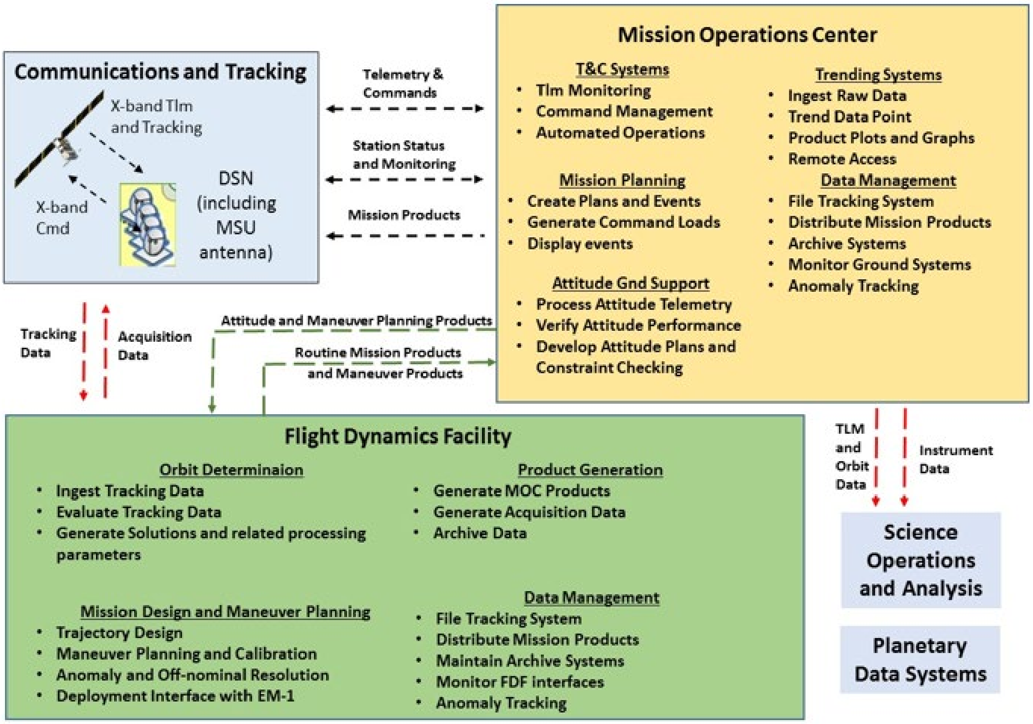


Figure 4-2. More specific functions and traces for LIC MOC and FDF at GSFC.[18]

As mentioned in §3.4 and §3.5, respectively, the LIC MOC and FDF provide important roles in the sustainment of LIC. However, jointly, they form a strong multiplier. Figure 4-2 gives a perspective on how much weight each brings to the table and how each provides for one another. To prove and produce this effective relationship, an FDF-to-MOC test serves as the V&V tool.

This test provides a concrete basis for the efficient and effective generation of data products between the LIC MOC and FDF. First, a review of the responsibilities of the LIC MOC is required. Recall that the LIC MOC provides real-time monitoring; commanding; data and telemetry reception; DSN scheduling; overall mission planning; data management and processing of science data; onboard system updates; attitude support functions; and an interface with the PDS.[19] Next, recall that the LIC MOC as a facility serves as the main telemetry and command interface to LIC, where it processes housekeeping data to monitor LIC health and safety. The LIC MOC also distributes measurement data to the LIC science team along with other required mission products; provides data storage for all raw measurement data for the life of the mission; receives any required instrument command sequences from the science team and processes them before uplink; and distributes real-time telemetry (EBCS, 7). Finally, recall that the facility houses LIC mission planning that leads to command load generation, trend analysis, and attitude determination functions.

Second, consider the responsibilities of FDF. Recall that FDF supports maneuver planning; orbit determination; orbit products generation and distribution; and DSN data and interfaces (EBCS, 6). The facility itself is a Navigation and Mission Design Branch (NMDB) facility that provides launch support; orbit determination and tracking data evaluation; ground and spacecraft data product generation; mission trajectory design; and maneuver planning and reconstruction. The NMBD provides flight dynamic services, supplying a navigating team and a mission design team composed of contractors and civil servants to support LIC. The mission design team has access to the orbit determination solutions and the required subsets of mission telemetry data (e.g., propulsion data, power, and attitude data) sent to the FDF via the FDF's CI. The CI is also employed for delivery to the MOC of maneuver plans, trajectory data, and other planning products. The CI consists of servers that provide incoming and outgoing data access for FDF mission support. For orbit determination, the FDF receives the tracking data from the DSN, processes that data to generate a reconstructed and predicted orbit and generate mission products. In addition to pre-mission trajectory and orbit planning, civil servants on the GSFC flight dynamics team monitor and plan from trajectory maneuvers during the cruise, capture maneuvers, and station-keeping maneuvers; the FDF supplies these maneuver plans to the MOC for command uplink and LIC execution (EBCS, 7).

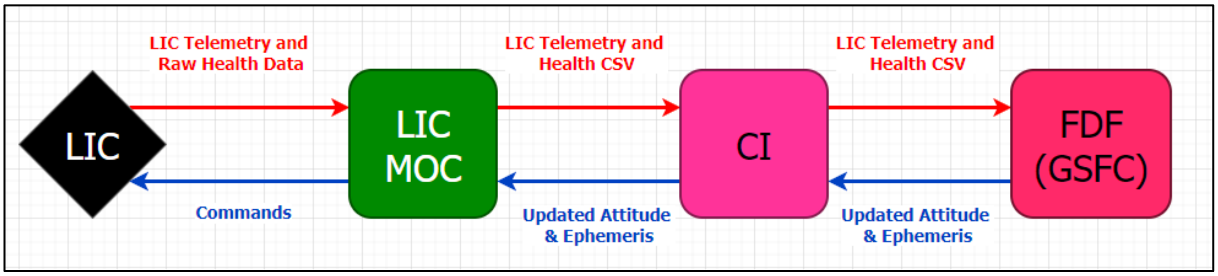


Figure 4-3. FDF-to-MOC data flows for testing and operations. Reproduced (EBCS, 9).

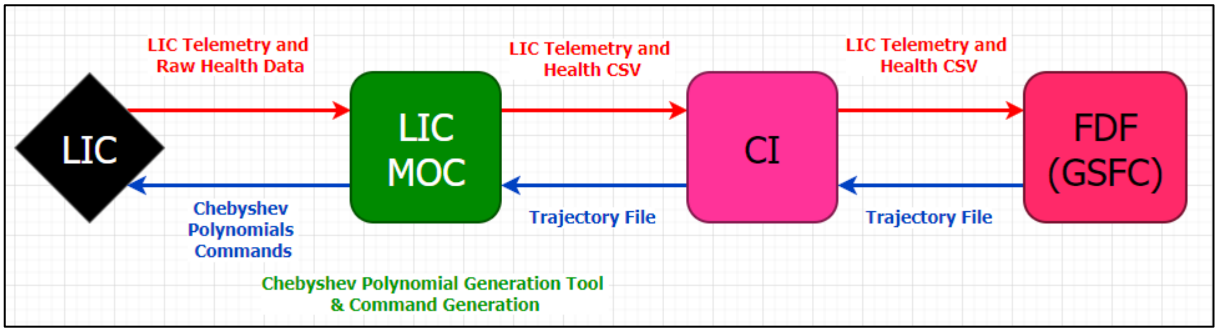


Figure 4-4. FDF-to-MOC data flows for Chebyshev polynomials. Reproduced (EBCS, 10).

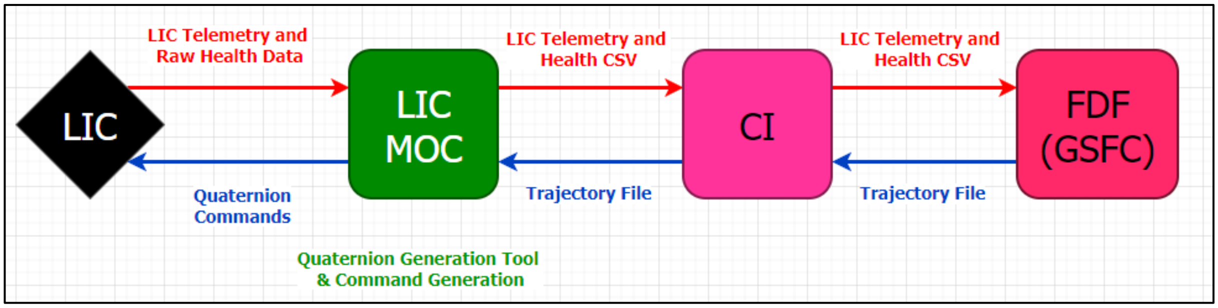


Figure 4-5. FDF-to-MOC data flows for quaternions. Reproduced (EBCS, 13).

Now that the responsibilities and the functions of both the LIC MOC and FDF have been reiterated in detail, refer to Figure 4-3. Figure 4-3 shows a simpler depiction of the telemetry and command flow between LIC MOC and FDF under this test. The LIC MOC utilizes a database utility tool that reads the raw data from the Database to generate a human readable (CSV) file with time-stamped data. This file contains datapoints from the ACS (XACT) and the propulsion subsystem (BIT-3), including a suite of spacecraft health. This information is uploaded to the CI to be received by FDF, where ephemeris files, attitude files, and trajectory changes are generated. FDF uploads these files onto the CI for the LIC MOC to receive and convert into CLTUs using Encode. The CLTUs are then sent to a DSS UPL for transmission to LIC (EBCS, 8).

This test is not exclusive to confirming the effectiveness and efficiency of the telemetry and command flows. It also confirms the impact of Chebyshev polynomial and quaternion generation on the flows. Chebyshev polynomials are used as a mathematical means of determining trajectories, with each consisting of a set of Chebyshev coefficients. As the mission continues, the polynomials and their coefficients are updated for each scheduled trajectory based on the tracking data received at FDF from the DSN. Figure 4-4 presents the same flow as Figure 4-3, except FDF provides trajectory files. These trajectory files are uploaded to the CI for use by the LIC MOC. Once the LIC MOC pulls the files from the CI, they are imported to a Chebyshev generation tool to construct the Chebyshev polynomial for the next scheduled trajectory. The team will define a resolution and time of the Chebyshev polynomials based upon the shape of the planned trajectory as well as how much resolution is needed. Resolution is quantified based upon curvature and out-of-plane motion and is limited by the byte space. The resolution ultimately determines how many Chebyshev polynomials can be uploaded to the ACS at a given time. LIC MOC then sends the polynomials as CLTUs to the ACS onboard LIC to facilitate the requested trajectory (EBCS, 10).

Quaternion generation works in the same manner as shown by Figure 4-5. LIC MOC inputs the trajectory file into an LIC simulation to account for pointing constraints. This then allows for the generation of a report containing information regarding how the spacecraft needs to be oriented. The report is converted into quaternion commands, which are then sent to LIC as CLTUs (EBCS, 13).

4.3) Lunar IceCube and Operational Readiness

Operational readiness, for any spacecraft-ground system, is the V&V process whereby a series of tests and trainings occur either pre- or post-delivery of the spacecraft, but before launch to certify the robustness of the system. This robustness is streamlined by the awareness of known impediments towards a healthy system and the application of concepts and results from previous testing conducted after integration. Essentially, operational readiness is sort of like the "seal-of-approval" for a system undergoing daily usage for a particular mission. Two important events are conducted for a spacecraft-ground system to be considered "operationally ready:" operational readiness tests (ORTs) and an operational readiness review (ORR). The LIC MOA is used once again as an example to provide small-spacecraft teams insight into the importance, process, and criteria of both events. Before diving in, an important note: as of the time of this writing the LIC program has not conducted ORTs. However, LIC's ORR has been conducted, and content related to this milestone is provided. ORTs are still introduced for conceptual and preparational purposes so teams can investigate and understand what to expect beforehand.

4.3.1) *Operational Readiness Testing*

ORTs are training scenarios designed to mimic real-space operations and an opportunity to practice nominal commanding and anomaly response in a lower-stakes environment than real flight operations (CH, 367). ORTs range from a few days to a week, but the duration and timeline vary on a case-by-case basis—more on that in a moment. Fidelity is the main design objective for ORTs, or to achieve the most realistic form of on-orbit conditions as possible subject to the availability of flight or flight-like hardware and software. Along with the MOA, it is strongly preferred to having the real flight-unit (if pre-delivery) or a high-fidelity engineering unit (if post-delivery) in the loop. Telemetry from preflight testing is an option if actual flight or flight-like hardware is unavailable. Procedures and tools should be used in the same manner as if used during actual in-orbit operations (CH, 368).

A key goal of ORTs is to “stress test” the operations team and train them to make careful but timely decisions about how to respond to anomalous system behavior. Every system, or for this matter every MOA, has quirky behavior in one or more elements, and ORT scenarios should account for those behaviors. Scenarios provide an opportunity to respond to previously observed elemental and sub-elemental misbehavior during previous MOA testing. If a specific element is known to exhibit anomalistic behavior, procedures should be tuned and exercised to address that fault so operators can recognize and mitigate any further complications when conducting operations with the spacecraft. The cycle should contain several rounds of operations planning, simulated ground contacts with spacecraft telemetry and commanding, and after action debriefs and evaluations. After ORTs are conducted, the operations team should assess any gaps in preparation or tools and work to address them before in-orbit operations begin (CH, 368). Since the timeline of each mission is unique, the delivery, launch, and deployment schedules set the timeline of ORTs. Missions that expect a long delay between delivery and launch could conduct ORTs during the schedule gap. If small satellites are deployed from the ISS, there may be a delay of several months between launch and deployment, providing an excellent opportunity for training and preparation. Launch and deployment dates are often uncertain, so it is highly advisable to perform at least a minimal set of ORTs before the spacecraft is delivered to the launch provider (CH, 369).

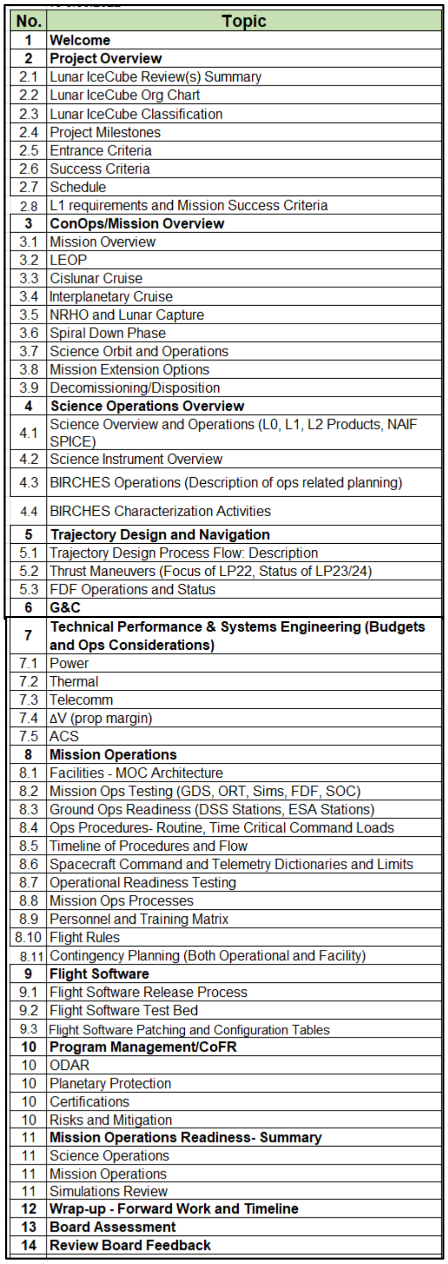
4.3.2) *Operational Readiness Review*

From JPL's project support office, an operational readiness review, or ORR, is defined as the event that "evaluates the readiness of the MOS [Mission Operations System] (or Instrument Operations System (IOS)) and all of its component elements, e.g., GDS (or Science Data System (SDS)), to support launch and flight operations."[20] Using this definition for this writing, an ORR can be modified as this: an evaluation of the readiness of the MOA and all of its component elements—macro and micro—to support launch and flight operations. The review's main purpose is to ensure that the operations team possesses the resources, processes, experience, personnel, and budget to operate successfully. An ORR is held because the rigor of operations is important. Consider the position of both project management and the operations team for a moment. ORR benefits both project management and the project operators. Project management is assured that the operations team possesses the resources and the knowledge to operate safely. The operations team is held accountable for updating processes from relevant ORTs, and ORR is the final chance for the team to receive constructive criticism and experienced feedback prior to in-orbit operations (Pasquale, 8). Mission operations by itself is a rigorous and pedantic process, but without those two distinctions missions themselves will flop. In the worst-case scenario, those stakeholders involved—represented by those on the ORR review panel—will not see any benefit in commissioning a system, meaning a waste of time and funding on the part of those stakeholders.

The goal is not to present a worst-case scenario or a system that "just works." To truly present a system that operates as expected based on the mission objectives and requirements, rigor is assumed and established within the process. Therefore, act accordingly. Here is the scope for an ORR used by JPL (Pasquale, 10):

- The operations plans, workforce plans, and schedules
- The readiness of the MOS (IOS)—including the GDS (SDS), personnel, processes and procedures, operations software, contingency plans for anomaly response, interfaces, and facilities—to support operations
- The verification and validation status, including tests and training exercises, and experience gained from used of operations personnel, systems, and procedures in flight system (instrument) I&T
- Exceptions to "Test-As-You-Fly/Fly-As-You-Test" principle
- The state of the documentation
- The status of the transition from development to operations
- Developments, if any, deferred to after launch, and associated risk
- Significant mission risks, including aggregate risk, and mitigations
- Project budget status and budget reserve, schedule status, and schedule margin(s), including the operations phase
- Work-to-go

As is noticeable from the list above, there is much detail to be addressed in an ORR. In fact, more detail is present when considering the specific agenda for such a review. LIC's agenda is provided as Table 4-1. Additionally, note that overall, the idea is to cover most of the bases regarding mission operations. Notice from Table 4-1 that content generated for the MOA can take only a fraction of the total content generated for ORR. While that fraction is the highlight of this writing, it is only just a piece of the total under consideration. However, the ORR is just a picture of mission operations; realistically, it might just be a possibility considering how the MOA and spacecraft are integrated as a system. In other words, it is never static, but dynamic. ORRs are meant to show an adequate/required capability, not an ideal capability. There can and will always be ways to improve or streamline the spacecraft-MOA system as the mission continues, and probably ones that were not considered before ORR and commissioning. Therefore, the program should have a process for updating the mission system to align with the accuracy, efficiency, and effectiveness required considering cost, performance, and the risk involved (Pasquale, 15).



No.	Topic
1	**Welcome**
2	**Project Overview**
2.1	Lunar IceCube Review(s) Summary
2.2	Lunar IceCube Org Chart
2.3	Lunar IceCube Classification
2.4	Project Milestones
2.5	Entrance Criteria
2.6	Success Criteria
2.7	Schedule
2.8	L1 requirements and Mission Success Criteria
3	**ConOps/Mission Overview**
3.1	Mission Overview
3.2	LEOP
3.3	Cislunar Cruise
3.4	Interplanetary Cruise
3.5	NRHO and Lunar Capture
3.6	Spiral Down Phase
3.7	Science Orbit and Operations
3.8	Mission Extension Options
3.9	Decomissioning/Disposition
4	**Science Operations Overview**
4.1	Science Overview and Operations (L0, L1, L2 Products, NAIF SPICE)
4.2	Science Instrument Overview
4.3	BIRCHES Operations (Description of ops related planning)
4.4	BIRCHES Characterization Activities
5	**Trajectory Design and Navigation**
5.1	Trajectory Design Process Flow: Description
5.2	Thrust Maneuvers (Focus of LP22, Status of LP23/24)
5.3	FDF Operations and Status
6	**G&C**
7	**Technical Performance & Systems Engineering (Budgets and Ops Considerations)**
7.1	Power
7.2	Thermal
7.3	Telecomm
7.4	ΔV (prop margin)
7.5	ACS
8	**Mission Operations**
8.1	Facilities - MOC Architecture
8.2	Mission Ops Testing (GDS, ORT, Sims, FDF, SOC)
8.3	Ground Ops Readiness (DSS Stations, ESA Stations)
8.4	Ops Procedures- Routine, Time Critical Command Loads
8.5	Timeline of Procedures and Flow
8.6	Spacecraft Command and Telemetry Dictionaries and Limits
8.7	Operational Readiness Testing
8.8	Mission Ops Processes
8.9	Personnel and Training Matrix
8.10	Flight Rules
8.11	Contingency Planning (Both Operational and Facility)
9	**Flight Software**
9.1	Flight Software Release Process
9.2	Flight Software Test Bed
9.3	Flight Software Patching and Configuration Tables
10	**Program Management/CoFR**
10	ODAR
10	Planetary Protection
10	Certifications
10	Risks and Mitigation
11	**Mission Operations Readiness- Summary**
11	Science Operations
11	Mission Operations
11	Simulations Review
12	**Wrap-up - Forward Work and Timeline**
13	**Board Assessment**
14	**Review Board Feedback**

Table 4-1. Example of LIC ORR agenda.

Typically, the timeline places ORR after the planned and scheduled ORTs. Now, for small spacecraft-based missions this may not always be true. Most of time secondary payloads are at the mercy of the primary payload, which then sets the mind for the launch provider. Therefore, dates are fluid. For example, LIC was delivered to Kennedy Space Center on July 12, 2021, ORR was conducted on April 14, 2022, and ORTs will soon follow before a summer 2022 launch date. In general, running ORTs and ORR near operations is advantageous since processes and procedures will be fresh on the mind for operators, but successful ORTs paired with a successful ORR should ensure the program has the products and resources necessary to operate at any point in time (Pasquale, 6).

5) Executive Epilogue: The Paradigm Shift

The reader(s) should have a general to in-depth understanding of the importance of a mission operations architecture for small spacecraft-based missions and the relevant elements and V&V provided as examples from the LIC mission. Since a small spacecraft-based mission would already have sufficient development on the side of ground support, the mission would also have sufficient documentation on the processes, procedures, and provisions allocated to that ground support. At this point it is more than likely that all the documentation has not been consolidated to define the MOA for the mission, but is documented separately to define individual, discrete elements. The effort should switch towards consolidating the information at hand, whether it be from procedures, manuals, or testing evaluations. Once that consolidation takes place, the next step would be to identify the relationship between elements and their discrete interfaces. The last step would be to create a physical mapping of the MOA through a flow-diagram. A project of this nature, such as the one under consideration for this writing, is a rigorous process that requires patience and diligence; it might even require more testing to confirm the findings. Overall, the process is based around the logical flow of the data from the spacecraft to the ground and vice-versa.

On a final note, it is important to wrap-up by discussing the higher-level meanings behind this project. As mentioned from the executive summary (§1.1), the advent of the spacecraft was an entirely new concept. With that concept, everything that came with it was “new.” Society was pushing towards that goal of space, and at the same time engineering and technology were pushing for something new to accommodate the “new” that was the spacecraft. Much “new” was achieved during the time of the 1960s: Mercury, Gemini, Apollo, Soyuz. Once society proved that it could achieve great deeds in space, it decided to push a little further in the 1970s and 1980s: Skylab, the Hubble Space Telescope, MIR, and STS, or the Space Shuttle. From the 1990s into the 2000s, society pushed harder: the Mars Pathfinder, Sojourner, Spirit, Opportunity, and the ISS. Even currently, from the 2010s onward, that persistence is striking: Curiosity; InSight; Perseverance; the James Webb Space Telescope; the Artemis program with the SLS and the Orion capsule; the Commercial Crew program with NASA, SpaceX, ULA, and Boeing. There are many missions omitted, but the picture should be clear. While “new” is the common theme here for missions, much of this “new” is what is seen on the launchpad or “seen” in space, but not so much behind the scenes.

Expanding on this, recall that small spacecraft-based missions are the focus group for this project. Additionally, recall that current architectures and infrastructure—ground support—for spacecraft are attuned towards large, flagship missions rather than small spacecraft-based missions. Here is the reason why. All the previous missions carried out by major space agencies, such as NASA, for the most part, were a success or resulted in a favorable outcome. Now, if something is successful, it sets precedents for future spacecraft missions, so why de-integrate everything established from the old spacecraft mission if it can be the foundation for a new spacecraft mission. Essentially, an update or upgrade of the old. Usually this means scrapping the older launch vehicle and spacecraft, inheriting then transitioning all the assets of the old spacecraft mission towards the creation of a new launch vehicle and spacecraft, and simultaneously create new assets to support the new spacecraft mission. However, since the new mission is inheriting the old mission's assets, this should impose the risks of the old mission onto the new mission. Next, compound that over several decades of new missions. There should exist an apparent large amount of risk with every new mission created that needs to be minimized. Therefore, a risk-averse trade-space is created: longer durations of development and testing for achieving minimal risk. From a third-person perspective, it seems space agencies are unable to scale back or come down to the level of small because of their imposition of being suited to large systems that adopt much risk. They are sort of stuck between a rock and a hard place. While it can achieve results and bring about innovation, it seems less accommodating, and the time between mission conception and launch appears longer.

Currently, there is some relief from the risk-averse trade-space. Commercial entities, such as SpaceX, Blue Origin, and Virgin Galactic, are approaching a trade-space where risk is manageable rather than minimized. In other words, a risk-tolerant trade-space. SpaceX is a clear example. The Falcon 9 is a partially reusable two-stage-to-orbit medium-lift launch vehicle supporting their partially reusable Dragon capsule. Currently, as from the Axiom-1 mission that launched the first private astronauts to ISS on April 8, 2022, a Falcon 9 first stage was used for a fifth time, while the Dragon capsule, Endeavour, was used for a third time. SpaceX has pioneered the risk-tolerant trade-space of reusing systems. It is accepting inherent risk by utilizing the same systems and processes; concurrently, it is updating and upgrading its design, manufacturing, and operational capabilities along the way. As the program develops, so too does the trade-space. This does not appear as an update or upgrade of an old system by disposing of an old system and acquiring the old assets towards a new system; therefore, durations of development and testing are shorter and are seen as accommodating with time between mission conception and launch.

However, both trade-spaces—risk-averse and risk-tolerant—are apparently still not accommodating enough for smaller spacecraft-based missions. While the aerospace industry has fully adopted the paradigm shift to transition from government-backed space agencies to commercial providers, the paradigm shift for support from large, flagship missions to small spacecraft-based missions is still ongoing. Since the trade-spaces encapsulate large, flagship missions, small spacecraft-based missions are somewhat unable to fit entirely into the mold; they are sort of "on-the-bubble".

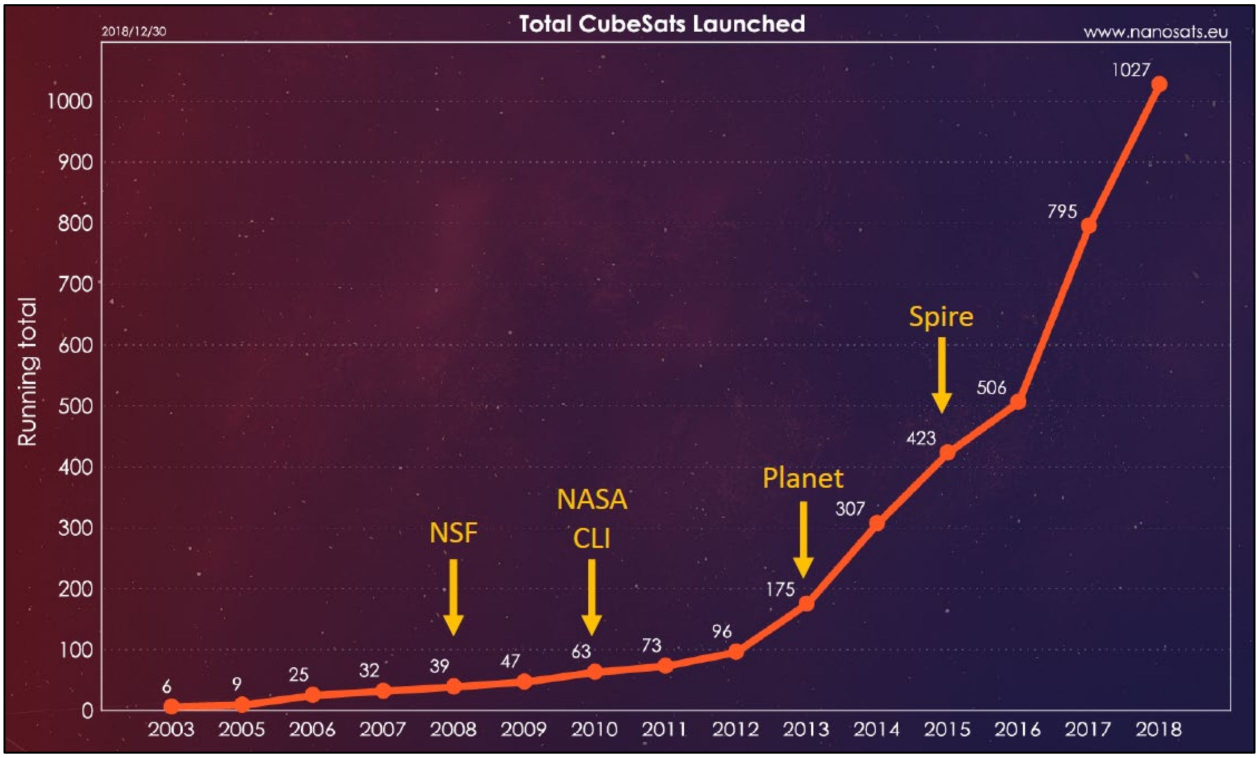


Figure 5-1. Running total of CubeSat launches in the years 2003-2018 (Klesh, 10).

Now, refer to Figure 5-1. This plot tracks the total number of CubeSats launched by year, and after initiatives are injected into the industry, such as through the NSF, NASA, and third-party companies, launches per year increased. Next, consider a pie chart that represents a mission divided into three slices: spacecraft, launch vehicle, and ground support. Small spacecraft have been accepted into the aerospace industry, and launch services are expanding to the point where providers are opting into only servicing small spacecraft-based missions. That is two-thirds of the pie accounted for small spacecraft in industry. What about the ground support, the ground assets, that final piece? Considering the scale, it is nonexistent to some extent. If a small spacecraft-based mission were to be selected as a prospective mission, all support would be provided through the same avenues as larger, flagship missions. Therefore, there is no distinction between missions other than the priority they are given. Consider LIC; it and nine other 6U CubeSats will be launched using NASA's SLS. Since these CubeSats are extra-Earth missions, they do not have comparable resource pools to maintain such a mission like those for larger spacecraft. For example, every CubeSat is contracted with NASA's DSN, which is the only telecommunications asset of its caliber on Earth to support deep space missions. However, it's important to note that the DSN already services tens of Earth-based and deep space missions within a tight subscription schedule. Therefore, with the addition of ten more spacecraft, the DSN becomes tighter on their service schedule and their ability to provide an ample amount of "viewing" time for that mission. Overall, if smaller spacecraft and even larger spacecraft are added to the DSN in the future, the service itself can become oversubscribed and missions might be left holding the bag on trying to find alternatives.

The DSN example represents a commonality for all small spacecraft-based missions: there is no architecture, infrastructure, and/or ground assets exclusive to small spacecraft. What exists is only an integrated, temporary patch rather than a defining solution. It is not that the larger trade-spaces do not give small spacecraft-based missions an opportunity or allow innovation in space to halt. It is mostly a matter of time and money. This should inevitably assume adaptation towards a current system, rather than adoption of new system. However, there needs to be an effort within the aerospace industry by leaders, such as NASA and SpaceX, to address that the paradigm shift, while present, is incomplete on the side of ground support for small spacecraft. The pace of the progress between all three assets is inconsistent, and there should be accountability in signaling that discrepancy before that discrepancy hinders the advancement of extra-Earth small spacecraft into the next generation. Therefore, this primer's intentions were to show that if a program is defined by the contents of this writing and is conducting a small spacecraft-based mission selected by a national space agency or commercial provider, this is a possible solution. Essentially, a MOA that can be that missing third piece of the pie representing the ground support necessary to make a small spacecraft-based mission a success.

Acknowledgements

First and foremost, I would like to thank my advisor, Nathan D. Fite, lead spacecraft engineer for Lunar IceCube, for his mentorship throughout this arduous process. His experience was instrumental in shaping the ideas for this project and applying them to paper. Second, I would like to thank my committee member, Jeffrey Gillette of Relative Dynamics, Inc., for providing industry perspective and feedback on my content. While lengthy and time consuming for him, his words held breadth and depth. Third, I would like to thank Dr. Benjamin K. Malphrus, executive director for the MSU Space Science Center, for his outstanding encouragement and support throughout my two years here as a master's student in space systems engineering. He brought me into this program with just a simple email, but what a turning point it was. Fourth, to my graduate student colleagues, thank you for your affirmations and conversations. You guys will reach higher than stars one day—hopefully I do as well. Finally, I want to thank the faculty and staff of Morehead State University's Space Science Center. You are all a gem in the mountains. I hope more will pass through your doors in the future to expand their horizons on this endeavor.

Ad astra.

Appendix

A.1) *Data Structure Diagrams*

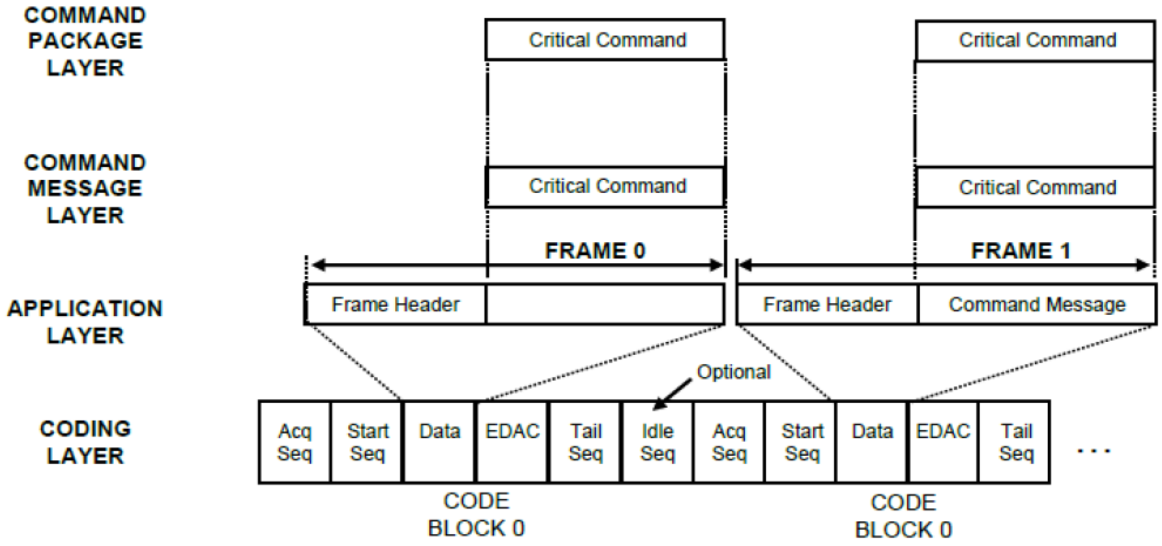


Figure A-1. CLTU uplink session layers for critical commands (FGICD, 17).

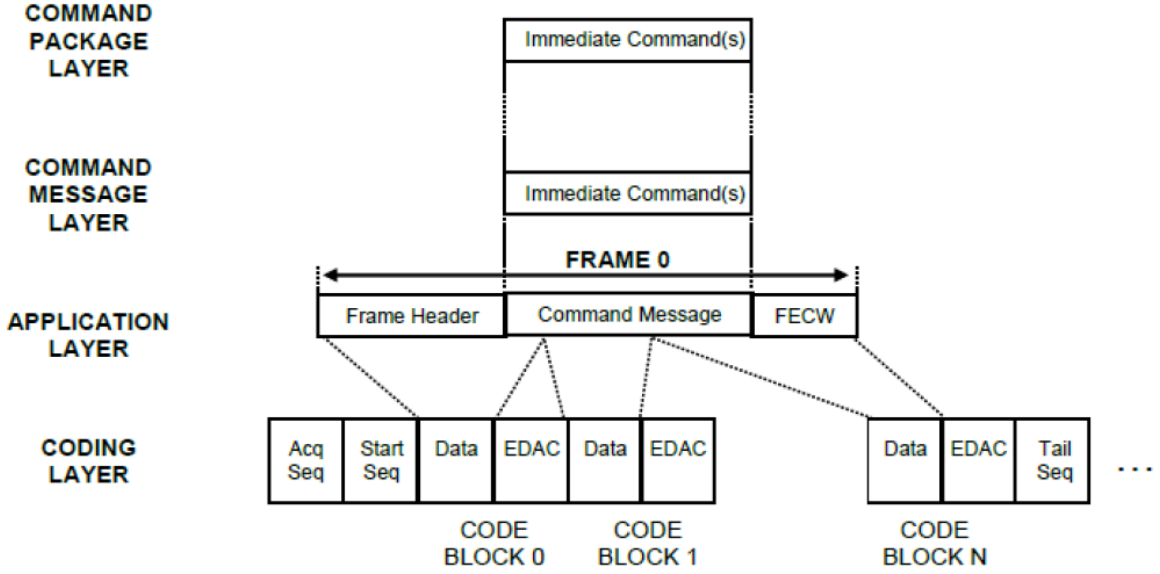


Figure A-2. CLTU uplink session layers for immediate commands (FGICD, 18).

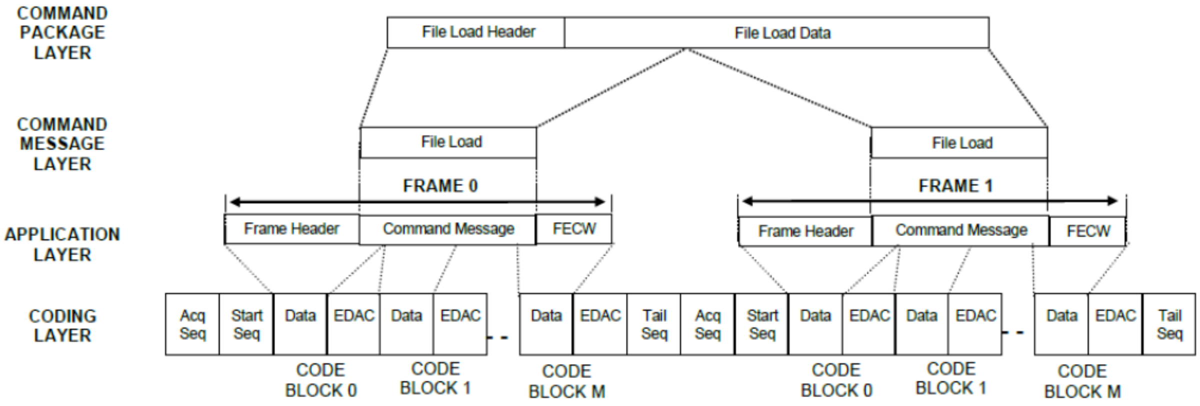


Figure A-3. CLTU uplink session layers for file loads (FGICD, 18).

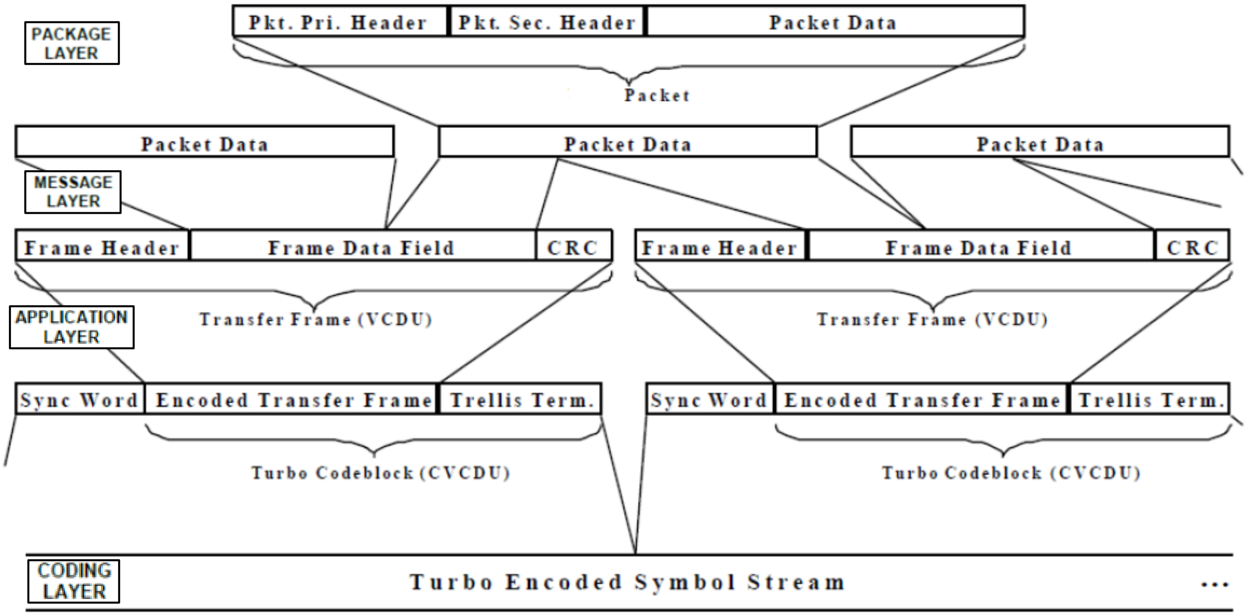


Figure A-4. RAF downlink protocol structure (FGICD, 29).

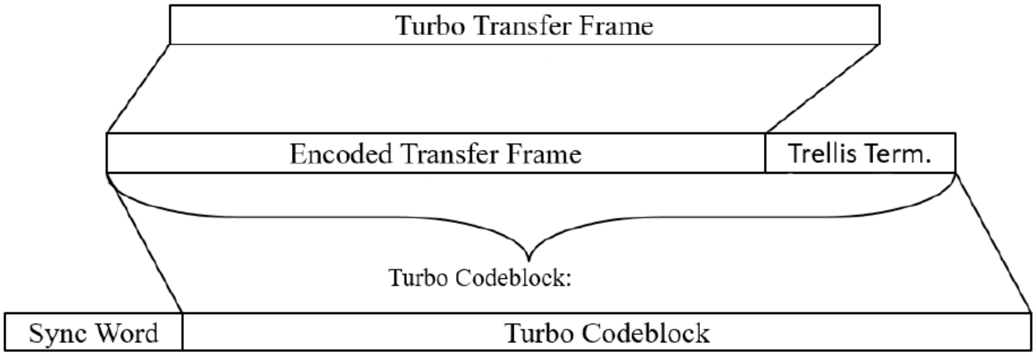


Figure A-5. Turbo code-block structure, including Turbo/AOS transfer frame (FGICD, 30).

A.2) *LIC MOA Flow-Diagrams*

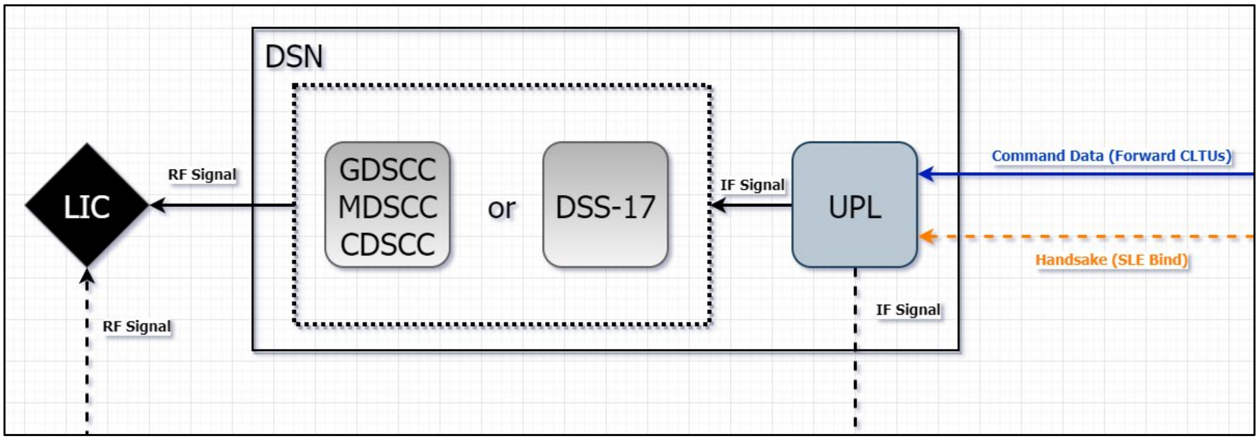


Figure A-6a. DSN uplink flow.

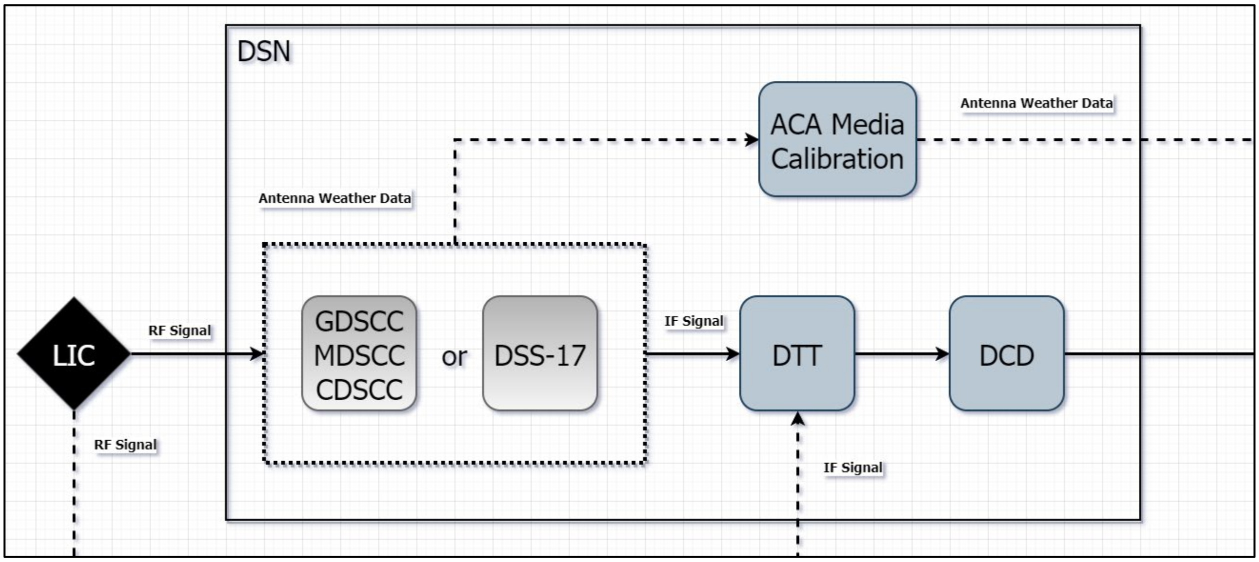


Figure A-6b. DSN downlink flow.

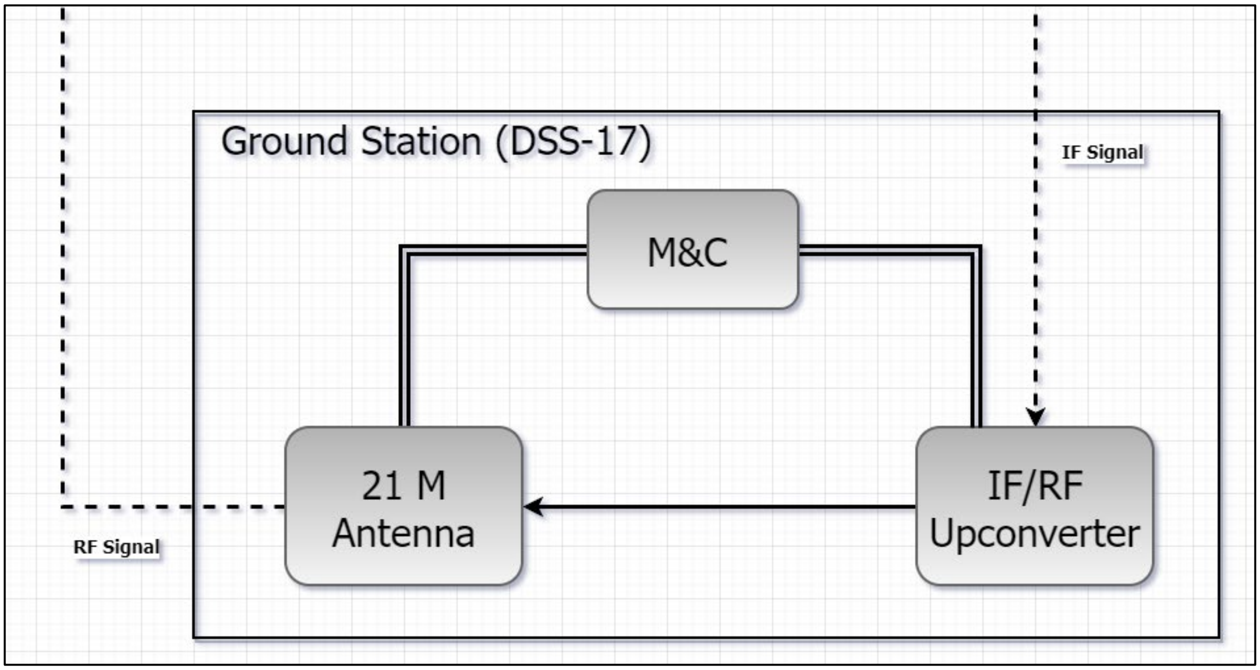


Figure A-7a. DSS-17 uplink flow.

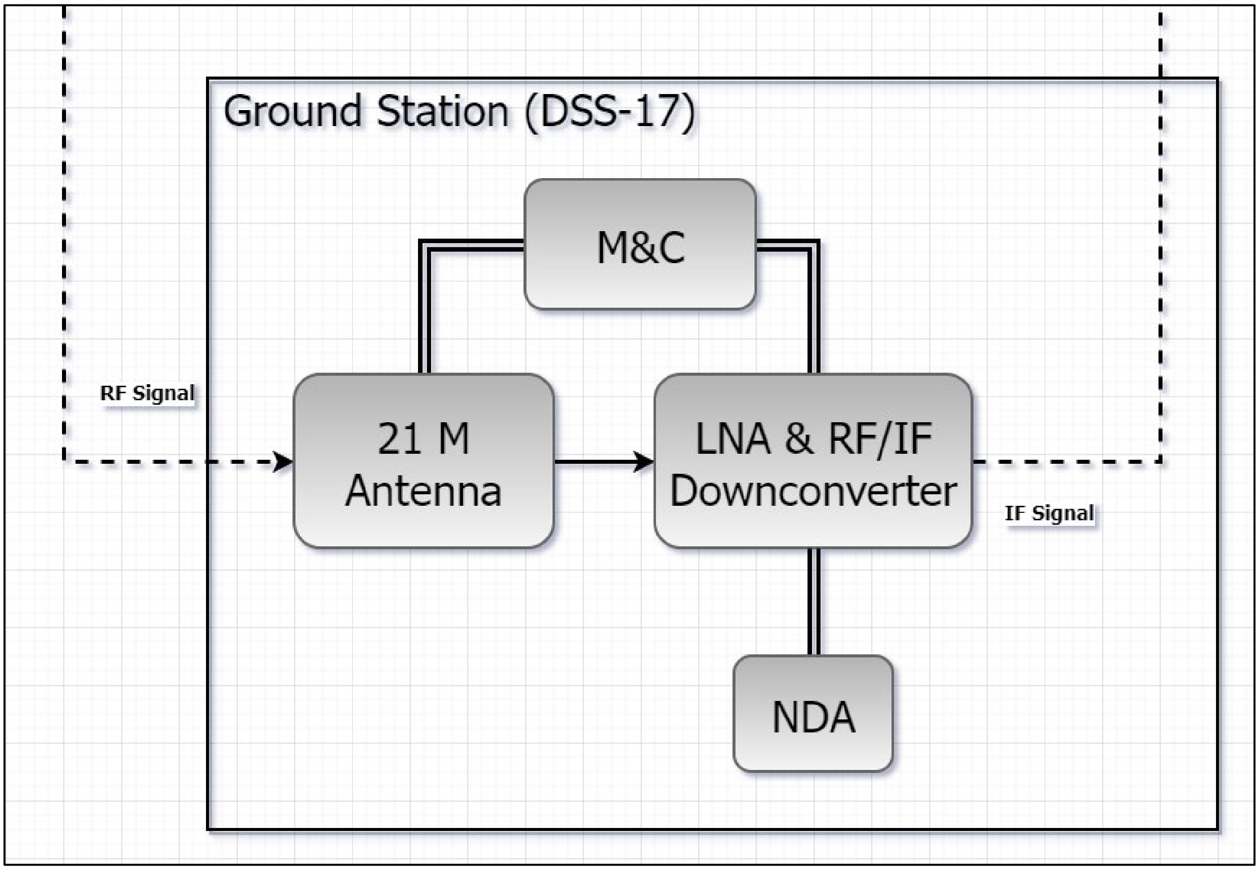


Figure A-7b. DSS-17 downlink flow.

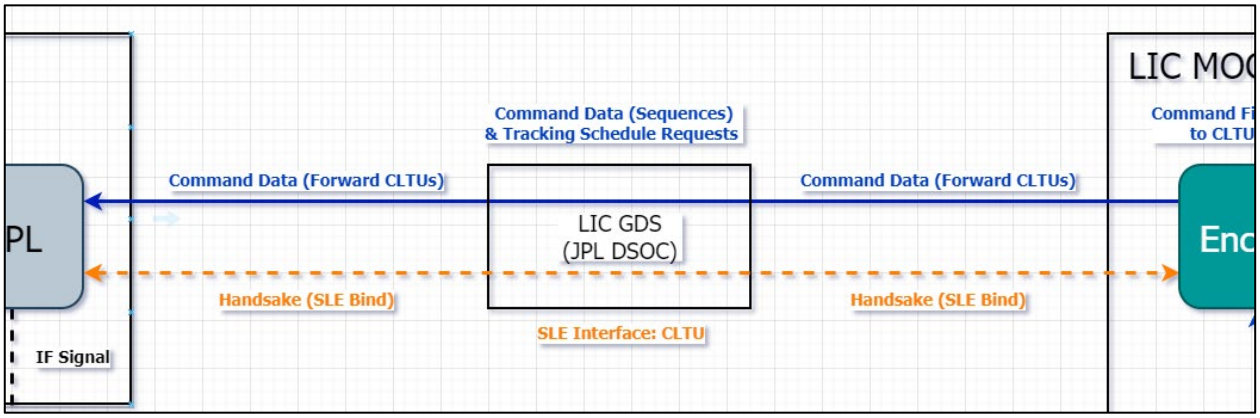


Figure A-8a. LIC GDS uplink flow.

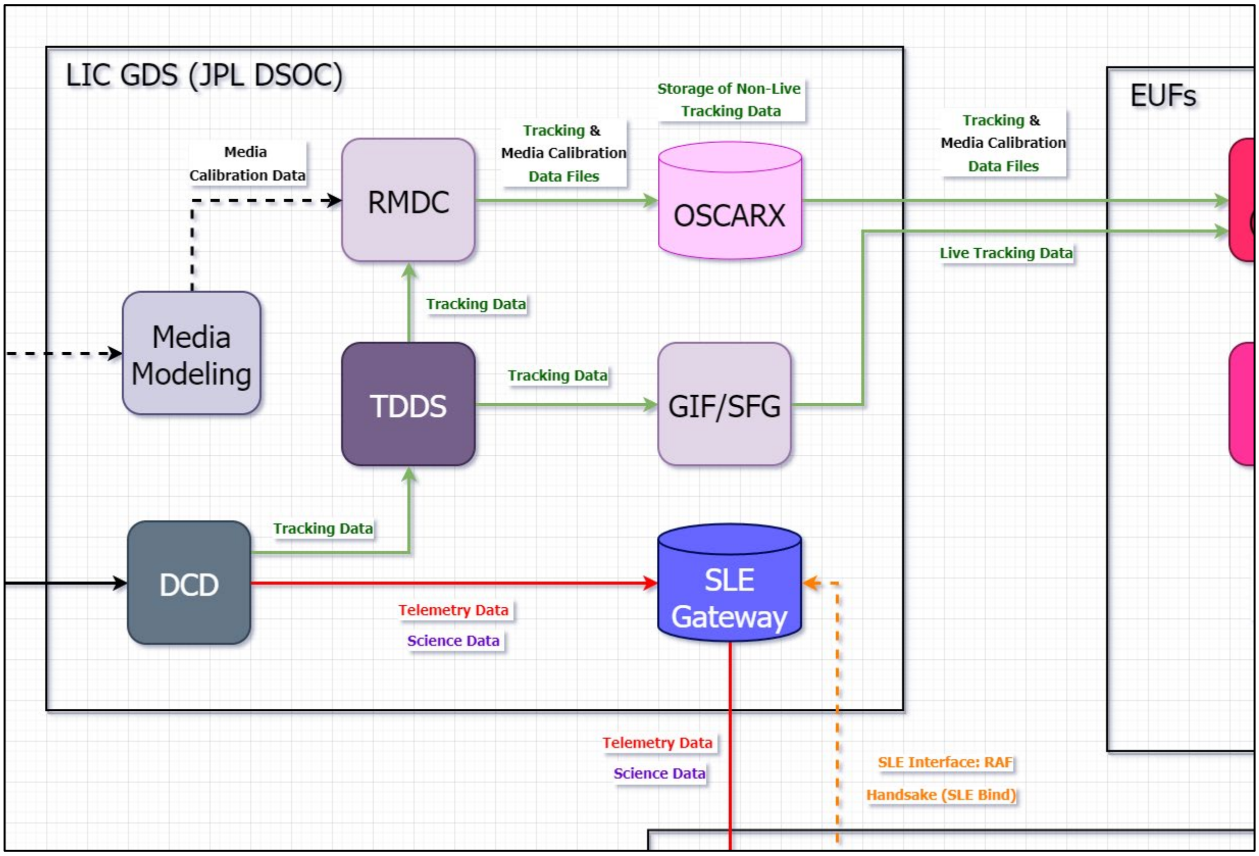


Figure A-8b. LIC GDS downlink flow.

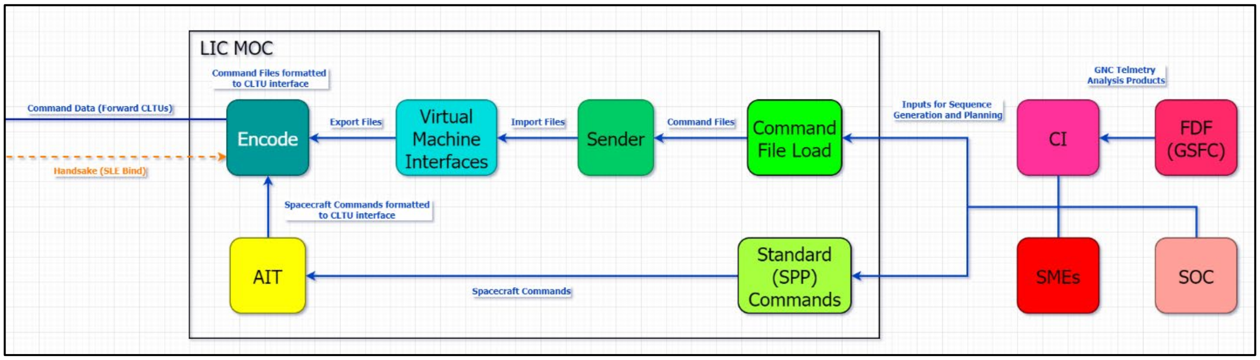


Figure A-9a. LIC MOC, EUFs, and SOC uplink flow.

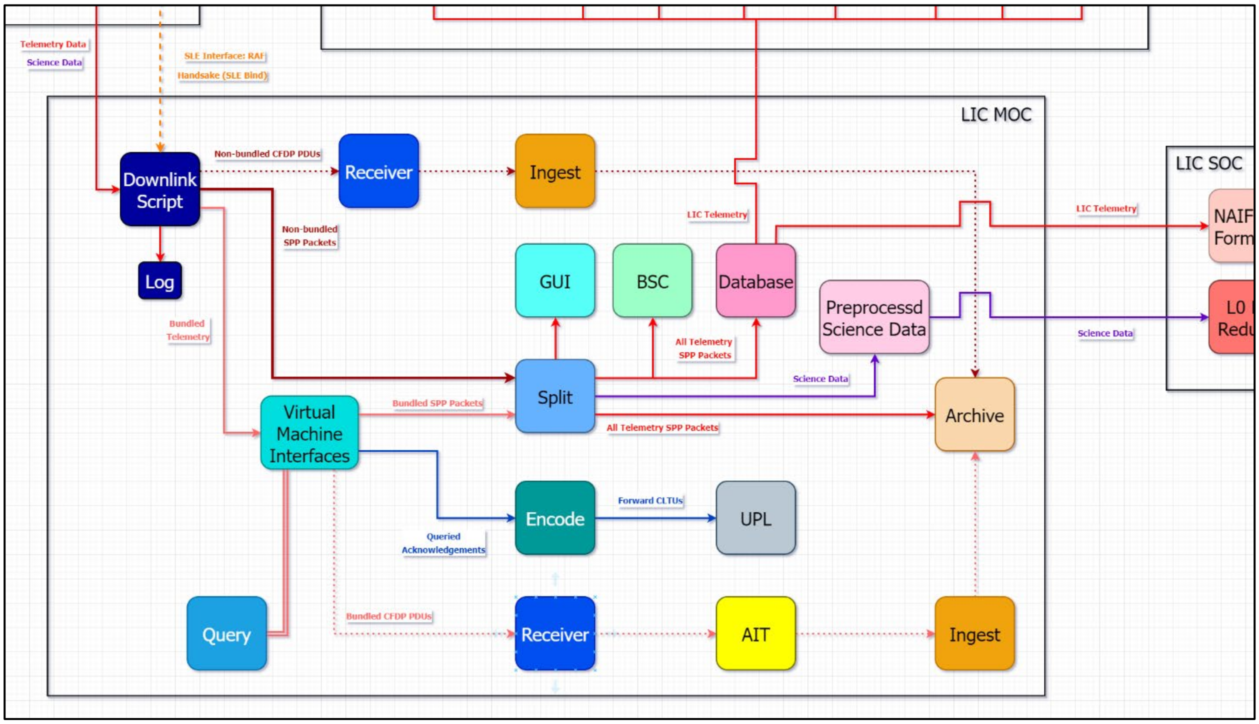


Figure A-9b. LIC MOC downlink flow.

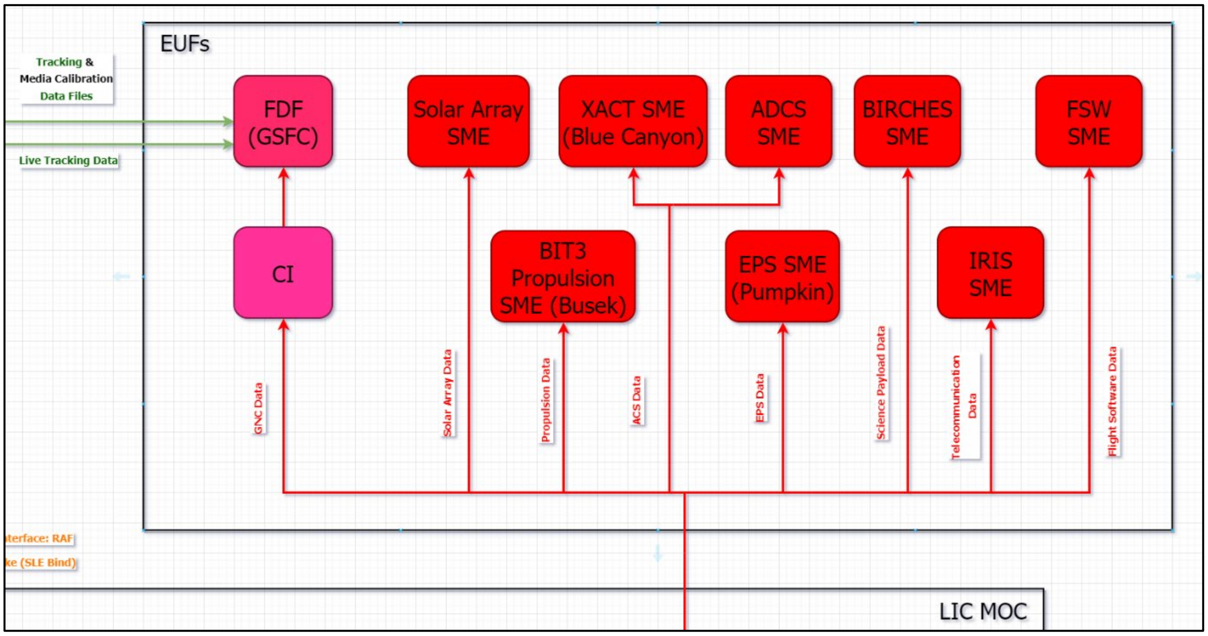


Figure A-10. LIC EUFs downlink flow.

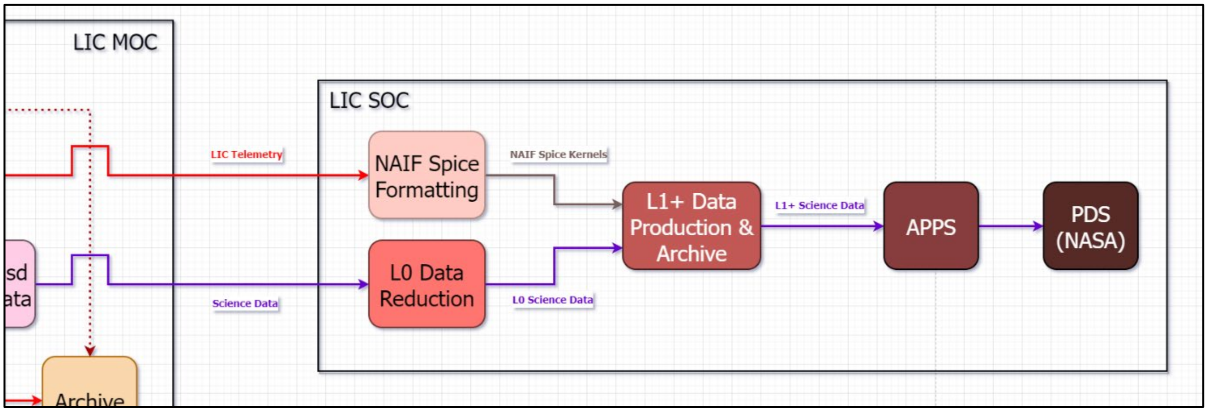


Figure A-11. LIC SOC downlink flow.

List of Abbreviations & Acronyms

ACS or ADCS – Attitude Control Subsystem or Attitude Determination Control Subsystem
AIT – AMMOS Instrument Toolkit
AMMOS – Advanced Multimission Operations System
AOS – Advanced Orbiting System
APID – Application Process Identifier
ARC – Ames Research Center (NASA)
ARPA – Advanced Research Project Agency
ASA – Antenna Controller Assembly
BCH – Bose-Chaudhuri-Hocquenghem
BPSK – Binary Phase-Shift Keying
CBS – Cost Breakdown Structure
CCSDS – Consultive Committee for Space Data Systems
CDH – Command & Data Handling Subsystem
CDR – Critical Design Review
CDSCC – Canberra Deep Space Communications Complex (NASA)
CFDP – CCSDS File Delivery Protocol
CI – Customer Interface
CLTU – Command Link Transmission Unit
CMD – Command Data
ConOps – Concept of Operations
COTS – Commercial-Off-the-Shelf
CRC – Cyclic Redundancy Check
CSA – Canadian Space Agency
CVCDU – Coded Virtual Channel Data Unit
DARPA – Defense Advanced Research Projects Agency
DCD – Data Capture and Delivery Subsystem
DCS – Distributed CubeSat Systems
Delta-DOR – Delta-Differential One-way Ranging
DFE – Direct-from-Earth
DSCC – Deep Space Communications Complex
DSN – Deep Space Network (NASA)
DSOC – Deep Space Operations Center (JPL)
DSS – Deep Space Station
DTE – Direct-to-Earth
DTT – Downlink Tracking and Telemetry Subsystem
ECC – Emergency Control Center (JPL)
ECSS – European Cooperation for Space Standardization
EDAC – Error Detection and Correction
EDL – Entry, Decent, and Landing
ESA – European Space Agency
EUF – External User Facility
FCC – Federal Communications Commission
FDF – Flight Dynamics Facility (GSFC)
FGICD – Flight-to-Ground ICD

FRR – Flight Readiness Review
FSW – Flight Software
GDS – Ground Data System
GDSCC – Goldstone Deep Space Communications Complex (NASA)
GEO – Geostationary Orbit
GIF – Ground Interface Facility
GSA – Ground Station Architecture
GSFC – Goddard Space Flight Center (NASA)
GSN – Ground Station Network
GNC – Guidance, Navigation, & Control
GUI – Graphical User Interface
HEO – Highly Elliptical Orbit
HORZ – Horizontal Polarization
HST – Hubble Space Telescope (NASA/ESA)
ICD – Interface Control Document
INCOSE – International Council on Systems Engineering
IF – Intermediate Frequency
ISS – International Space Station
JAXA – Japanese Aerospace Exploration Agency
JPL – Jet Propulsion Laboratory (Caltech/NASA)
LCROSS – Lunar Crater Observation and Sensing Satellite (NASA)
LEO – Low Earth Orbit
LEOP – Launch and Early Orbit Plan
LGALRO – Lunar Gravity Assist Lunar Return Orbit
LHCP – Left-Hand Circular Polarization
LIC – Lunar IceCube (Morehead State University)
LNA – Low Noise Amplifier
LOI – Lunar Orbital Insertion
LPRP – Lunar Precursor Robotic Program (NASA)
LRO – Lunar Reconnaissance Orbiter (NASA)
M2m – Macroscopic to Microscopic
M&C – Monitor and Control
MarCO – Mars Cube One
MDSCC – Madrid Deep Space Communications Complex (NASA)
MEO – Medium Earth Orbit
MOA – Mission Operations Architecture
MOC – Mission Operations Center
MSC – Manned Spacecraft Center (NASA)
MSU – Morehead State University
MWN – Mercury Worldwide Network (NASA)
NACA – National Advisory Council for Astronautics
NAIF – Navigation and Ancillary Information Facility (NASA)
NASA – National Aeronautics and Space Administration
NEN – Near Earth Network (NASA)
NextSTEP – Next Space Technologies for Exploration Partnerships (NASA)
NMDB – Navigation and Mission Design Branch (GSFC)

NOAA – National Oceanic and Atmospheric Administration
NRCSD – Nanoracks CubeSat Deployer
OICD – Operations Interface Control Document
OPAL – Orbiting Picosat Automated Launchers (Stanford)
OQPSK – Offset Quadrature Phase-Shift Keying
ORR – Operational Readiness Review
PDR – Preliminary Design Review
PDU – Protocol Data Unit
PDS – Planetary Data System (NASA)
PPOD – Poly Picosat Orbiting Deployer (Cal Poly)
PSK – Phase Shift Keying
QPSK – Quadrature Phase-Shift Keying
RAF – Return All Frames
RHCP – Right-Hand Circular Polarization
RF – Radio Frequency
RMDC – Radiometric Data Conditioning
ROC – Remote Operations Center (JPL)
ROSCOSMOS – Russian State Space Agency
SCID – Spacecraft Identifier
SFDU – Standard Formatted Data Unit
SFG – Special Function Gateway
SLE – Space Link Extension
SLS – Space Launch System
SME – Subject Matter Expert
SNR – Signal-to-Noise Ratio
SOC – Science Operations Center
SPICE – Spacecraft ephemeris, Planet (or target body), Instrument, C-matrix (orientation), Event
SPP – Space Packet Protocol
SPSC – Source Packet Sequence Count
STG – Space Task Group (NASA)
TCM – Trajectory Correction Maneuver
TDDS – Tracking Data Delivery Subsystem
TLM – Telemetry Data
UHF – Ultra High Frequency
ULA – United Launch Alliance (U.S.A.)
UPL – Uplink Subsystem
V&V – Verification and Validation
VC – Virtual Channel
VCDU – Virtual Channel Data Unit
VERT – Vertical Polarization
VHF – Very High Frequency
WBS – Work Breakdown Structure

Endnotes

1. Chantal Cappelletti, Simone Battistini, and Benjamin K. Malphrus, eds., *CubeSat Handbook: From Mission Design to Operations* (Cambridge, MA: Academic Press, 2021), 1 (cited in text as CH).

2. National Aeronautics and Space Administration, *NASA Systems Engineering Handbook*, 2nd rev. ed., NASA SP-2016-6105 Rev2 (Washington, DC, 2016), 17 (cited in text as NASA SE).

3. National Aeronautics and Space Administration, *State-of-the-Art Small Spacecraft Technology*, NASA TP-2020-5008734 (Moffett Field, CA, 2020), 218 (cited in text as SOA).

4. Andrew Klesh, "Lecture 1: Introduction," (virtual lecture for Deep Space Mission Ops for Small Spacecraft by Naval Postgraduate School, Jet Propulsion Laboratory/California Institute of Technology, Pasadena, CA, March 4, 2021).

5. National Aeronautics and Space Administration, *CubeSat 101: Basic Concepts and Processes for First-Time CubeSat Developers*, rev. ed, NASA NP-2017-10-2470-HQ (Washington, DC, 2017), 39 (cited in text as CS101).

6. National Aeronautics and Space Administration, *Mercury Project Summary including results from the Fourth Manned Orbital Flight, May 15 and 16, 1963*, NASA SP-45 (Washington, DC, 1963), 16 (cited in text as Mercury).

7. National Aeronautics and Space Administration, *Lunar Reconnaissance Orbiter (LRO): Leading NASA's Way Back to the Moon & Lunar Crater Observation and Sensing Satellite (LCROSS): NASA's Mission to Search for Water on the Moon Press Kit/June 2009*, Press Kit Ver2 (Washington, DC, 2009), 14 (cited in text as PK).

8. Martin B. Houghton, Craig R. Tooley, and Richard S. Saylor, Jr., "Mission Design and Operations Considerations for NASA's Lunar Reconnaissance Orbiter," NASA Goddard Space Flight Center and Honeywell Technology Solutions Inc. (published paper, Greenbelt, MD, October 30, 2007), 1.

9. Paul D. Tompkins, et al., "Flight Operations for the LCROSS Impactor Mission," Stinger Ghaffarian Technologies, NASA Ames Research Center, and Northrop Grumman Aerospace Systems (paper presented at SpaceOps 2010 Conference of the American Institute of Aeronautics and Astronautics, Huntsville, AL, April 2010), 1.

10. Jet Propulsion Laboratory/California Institute of Technology, *DSN Mission Service Interfaces, Policies, and Practices (MSIPP)*, owned by Glen Elliot, DSN No. 875-0001, Rev. H, JPL D-26688 (Pasadena, CA, July 31, 2020), 2-1.

11. Benjamin K. Malphrus, "Deep Space Communications and the NASA Deep Space Network" (virtual lecture for Deep Space Mission Ops for Small Spacecraft by Naval Postgraduate School, Morehead State University, Morehead, KY, March 16, 2021).

12. Jet Propulsion Laboratory/California Institute of Technology, *Deep Space Network Services Catalogue*, owned by Jeff Berner, prepared by Timothy Pham, DSN No. 820-100, Rev. G (Pasadena, CA, September 8, 2020), 2-2 (cited in text as DSNSC).

13. Sarah E. Wilczewski, “Design of the 21-Meter Network Monitor and Control System for Deep Space Network Cross Support” (master’s thesis, Morehead State University, 2019), 23 – 26, ProQuest Dissertations & Theses Global, https://scholarworks.moreheadstate.edu/msu_theses_dissertations/359 (cited in text as Wilczewski).

14. Jet Propulsion Laboratory/California Institute of Technology, *DSN-Morehead State University Lunar IceCube (MLIC) Mission Operations Control Document (OICD)*, owned by Karen Yetter, DSN No. 875-0041, Rev. A (Pasadena, CA, January 12, 2022), 3-1 (cited in text as OICD).

15. Morehead State University Ronald G. Eaglin Space Science Center, *21 M Space Tracking Antenna-Deep Space Station-17 Quicksheet*, owned by Benjamin K. Malphrus and Amanda Holbrook (Morehead, KY, 2020), 1 (cited as MSU).

16. Morehead State University, *Lunar IceCube Flight-Ground Interface Control Document (FG-ICD) Pre-Release*, prepared by Sarah Wilczewski, approved by Nathan D. Fite, concurred by Benjamin K. Malphrus (Morehead, KY, November 12, 2018), 16 (cited in text as FGICD).

17. Jet Propulsion Laboratory/California Institute of Technology, *Telemetry Standard Formatted Data Unit (SFDU) Interface*, 0161-Telecomm, DSN No. 820-013, Rev. A, prepared by Tracy Clark, owned by Harvey Soldan, concurred by Jeff Berner, approved by T. Pham and J. Stipanuk (Pasadena, CA, August 29, 2008), 1-1 (cited in text as Telecomm).

18. NASA, Goddard Space Flight Center, *Lunar IceCube Project Flight Dynamics Facility (FDF) Mission Operations Center (MOC) Interface Control Document*, prepared by David Folta, reviewed by Steve Slojkowski and John Lynch (Greenbelt, MD, March 1, 2019), 2-1.

19. Morehead State University, *MOC to FDF ICD*, prepared by Emily Ballantyne and Christo Smith, reviewed by Nathan D. Fite (Morehead, KY, March 8, 2022), 6 (cited in text as EBCS).

20. Peter Di Pasquale, “Operational Readiness Review Overview,” (virtual lecture for Deep Space Mission Ops for Small Spacecraft by Naval Postgraduate School, Jet Propulsion Laboratory/California Institute of Technology, Pasadena, CA, April 22, 2021), 4 (cited as Pasquale).

Bibliography

Cappelletti, Chantal, Simone Battistini, and Benjamin K. Malphrus, eds. *CubeSat Handbook: From Mission Design to Operations*. Cambridge, MA: Academic Press, 2021.

Houghton, Martin B., Craig R. Tooley, and Richard S. Saylor. “Mission Design and Operations Considerations for NASA’s Lunar Reconnaissance Orbiter.” Paper published by NASA Goddard Space Flight Center and Honeywell Technology Solutions Inc., Greenbelt, MD, October 30, 2007.

Jet Propulsion Laboratory/California Institute of Technology, *Deep Space Network Services Catalogue*. Owned by Jeff Berner. Prepared by Timothy Pham. DSN No. 820-100, Rev. G, Pasadena, CA, September 8, 2020.

Jet Propulsion Laboratory/California Institute of Technology, *DSN Mission Service Interfaces, Polices, and Practices (MSIPP)*. Owned by Glen Elliot. DSN No. 875-0001, Rev. H, JPL D-26688, Pasadena, CA, July 31, 2020.

Jet Propulsion Laboratory/California Institute of Technology, *DSN-Morehead State University Lunar IceCube (MLIC) Mission Operations Control Document (OICD)*. Owned by Karen Yetter, DSN No. 875-0001, Rev. A, Pasadena, CA, October 6, 2021.

Jet Propulsion Laboratory/California Institute of Technology, *Telemetry Standard Formatted Data Unit (SFDU) Interface*. Prepared by Tracy Clark. Concurred by Jeff Berner. Approved by T. Pham and J. Stipanuk. 0161-Telecomm, DSN No. 820-013, Rev. A, Pasadena, CA, August 29, 2008.

Klesh, Andrew. “Lecture 1: Introduction.” Virtual lecture for Deep Space Ops for Small Spacecraft by Naval Postgraduate School, Jet Propulsion Laboratory/California Institute of Technology. Pasadena, CA, March 4, 2021.

Malphrus, Benjamin K. “Deep Space Communications and the NASA Deep Space Network.” Virtual lecture for Deep Space Mission Ops for Small Spacecraft by Naval Postgraduate School, Morehead State University, Morehead, KY, March 16, 2021.

Morehead State University, *21 M Space Tracking Antenna-Deep Space Station-17 Quicksheet*. Owned by Benjamin K. Malphrus and Amanda Holbrook, Morehead, KY, 2020.

Morehead State University, *Lunar IceCube Flight-Ground Interface Control Document (FG-ICD) Pre-Release*. Prepared by Sarah Wilczewski, approved and concurred by Nathan D. Fite and Benjamin K. Malphrus, Morehead, KY, November 12, 2018.

Morehead State University, *MOC to FDF ICD*. Prepared by Emily Ballantyne and Christo Smith. Reviewed by Nathan D. Fite. Morehead, KY, March 8, 2022.

National Aeronautics and Space Administration. *CubeSat 101: Basic Concepts and Processes for First-Time CubeSat Developers*. Rev. ed. NASA NP-2017-10-2470-HQ. Washington, DC, 2017.

National Aeronautics and Space Administration. *Lunar Reconnaissance Orbiter (LRO): Leading NASA's Way Back to the Moon & Lunar Crater Observation and Sensing Satellite (LCROSS): NASA's Mission to Search for Water on the Moon Press Kit/June 2009*. Press Kit Ver2, Washington, DC, June 2009.

National Aeronautics and Space Administration, Manned Spacecraft Center, Project Mercury. *Mercury Project Summary including results for the Fourth Manned Orbital Flight, May 15 and 16, 1963,* NASA SP-45, by NASA Office of Scientific and Technical Information. Washington, DC, October 1963.

National Aeronautics and Space Administration. *NASA Systems Engineering Handbook*. 2nd rev. ed. NASA SP-2016-6105. Washington, DC, 2016.

National Aeronautics and Space Administration. *State-of-the-Art Small Spacecraft Technology*. NASA-TP-2020-5008734. Moffett Field, CA, 2020.

NASA, Goddard Space Flight Center, *Lunar IceCube Project Flight Dynamics Facility (FDF) Mission Operations Center (MOC) Interface Control Document*. Prepared by David Folta. Reviewed by Steve Slojkowski and John Lynch. Greenbelt, MD, March 1, 2019.

Pasquale, Peter Di. "Operational Readiness Review Overview." Virtual lecture for Deep Space Mission Ops for Small Spacecraft by Naval Postgraduate School, Jet Propulsion Laboratory/California Institute of Technology. Pasadena, CA, April 22, 2021.

Tompkins, Paul D., et al. "Flight Operations for the LCROSS Lunar Impactor Mission." Paper presented at the SpaceOps 2010 Conference of the American Institute of Aeronautics and Astronautics, Huntsville, AL, April 25-30, 2010. https://arc.aiaa.org/doi/10.2514/6.2010-1986.

Wilczewski, Sarah E., "Design of the 21-Meter Network Monitor and Control System for Deep Space Network Cross Support." Master's thesis, Morehead State University, 2019, ProQuest Dissertations & Theses Global, https://scholarworks.moreheadstate.edu/msu_theses_dissertations/359.