Principles to Products: Toward Realizing MOS 2.0

Duane L. Bindschadler[*] and Christopher L. Delp[†]
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109

This is a report on the Operations Revitalization Initiative, part of the ongoing NASA-funded Advanced Multi-Mission Operations Systems (AMMOS) program. We are implementing products that significantly improve efficiency and effectiveness of Mission Operations Systems (MOS) for deep-space missions. We take a multi-mission approach, in keeping with our organization's charter to "provide multi-mission tools and services that enable mission customers to operate at a lower total cost to NASA." Focusing first on architectural fundamentals of the MOS, we review the effort's progress. In particular, we note the use of stakeholder interactions and consideration of past lessons learned to motivate a set of Principles that guide the evolution of the AMMOS. Thus guided, we have created essential patterns and connections (detailed in companion papers) that are explicitly modeled and support elaboration at multiple levels of detail (system, sub-system, element…) throughout a MOS. This architecture is realized in design and implementation products that provide lifecycle support to a Mission at the system and subsystem level. The products include adaptable multi-mission engineering documentation that describes essentials such as operational concepts and scenarios, requirements, interfaces and agreements, information models, and mission operations processes. Because we have adopted a model-based system engineering method, these documents and their contents are meaningfully related to one another and to the system model. This means they are both more rigorous and reusable (from mission to mission) than standard system engineering products. The use of models also enables detailed, early (e.g., formulation phase) insight into the impact of changes (e.g., to interfaces or to software) that is rigorous and complete, allowing better decisions on cost or technical trades. Finally, our work provides clear and rigorous specification of operations needs to software developers, further enabling significant gains in productivity.

I. Introduction

This report discusses the methods and results of the in-progress effort to revitalize multi-mission operations within the AMMOS[1,2,3]. After this introduction, we address foundational aspects of our architectural approach (Section II). Examples of our architecture and analysis of the current AMMOS as compared to an eventual future-state MOS 2.0 or "To-Be" system are found in Section III. Section IV describes the products of Ops Revitalization as informed by and stemming from the To-Be architectural vision. We discuss the value of these new products in Section V and offer conclusions in Section VI.

The AMMOS is a system available for use by NASA's deep-space science missions (i.e., those using the Deep Space Network for communications) to execute mission operations. Historically, the AMMOS was focused on mission-configurable or -adaptable software product lines for mission operations. In 2004, the MGSS organization was formed to incorporate this software and a set of multi-mission teams and their associated capabilities into the AMMOS. Before that time, operations teams (and their processes) were the responsibility of one organization, and the AMMOS software another. The incorporation of both parts within a single organization created opportunities for more closely aligning the two; in particular for better ensuring that software capabilities were well aligned with the processes of multi-mission operations.

Today, the purpose of the AMMOS is to provide tools and services that enable mission customers to develop and operate their missions at a lower total cost to NASA than if each mission acquired these capabilities on their own. The AMMOS Operations Revitalization (Ops Revitalization) effort was initiated to better realize this purpose, particularly in providing a wider range of operations services to a more diverse mission set than had been done

---

[*] Asst. Program Manager for Operations, MGSS, M/S 264-235, 4800 Oak Grove Dr., AIAA Senior Member
[†] Systems Architect, Systems Engineering Section, M/S 301-270, 4800 Oak Grove Dr.

before. The Ops Rev effort has focused on the aspects of the AMMOS in particular and MOS in general that are not particular to software, but focus instead on the higher-level functions of the system, the mission processes executed by personnel (using software), and the information taken in, transformed, and sent out to external systems. The scope of the Ops Revitalization effort was first specified in the form of a series of Goals and Objectives (Figure 1). The Goals are each elaborations of the AMMOS purpose, and are discussed in more detail in Bindschadler *et al.*[1]. Objectives are (in general) more specific intent and constraint on the overall Goals.

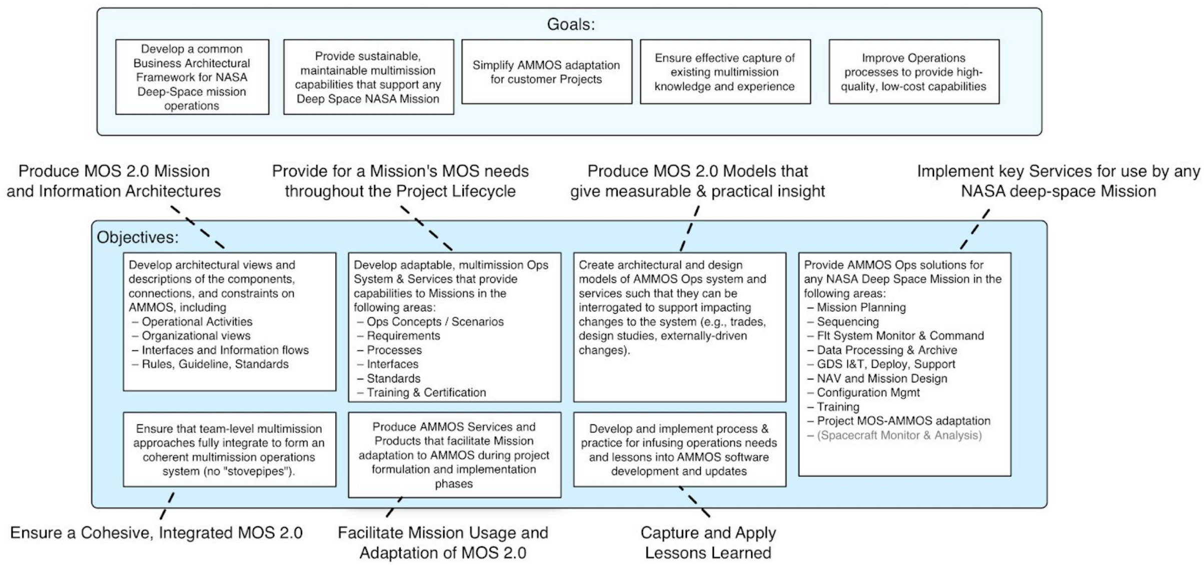

Figure 1. Goals and Objectives. *During the past two years, we have added one Goal ("Improve Operations processes…") after recognizing that it was implicit in the work already accomplished and some of the Objectives have also been modified to reflect that change. Objectives capture an elaboration and more product-centric take on the Goals. It is worth noting that there is no strict a compositional relationship between Goals and Objectives. Instead, Objectives are to be understood as the starting point for how the Goals might be achieved, and to direct and (in some cases) constrain the efforts of the Ops Revitalization Team.*

This gave initial direction to the effort. In the course of Ops Revitalization, our adoption of an architectural approach has lead us to understand underlying Principles and stakeholder concerns that were (in many cases) implicit in the Goals and Objectives. In the following sections we will discuss those Principles and Concerns and how they inform the products of our effort.

II. A Principled Architectural Approach

In this section we briefly review progress to date, and introduce the foundations of Ops Revitalization: our architectural principles, stakeholder interactions, and a model-based systems engineering (MBSE) methodology. Previous papers describe early progress of the Ops Revitalization effort, particularly:

- The motivations behind the effort's Goals and Objectives and the rationale for adopting a Model-Based Systems Engineering (MBSE) approach[1].
- The use of operational concepts and scenarios to capture the foundational aspects of deep-space missions[2].
- The necessity for a principled, architectural approach, and some early results[3].

Over the past year, we have refined our approach to articulating and capturing a To-Be architecture and have begun implementing MOS operations services that are based on that new architecture. These services are discussed in more detail in Section 4 sections.

Here we discuss three foundations of Ops Revitalization:

- The architectural principles that are essential to what a MOS *is* and what it *does.*
- The essential nature of stakeholder interactions, particularly the capture of and response to their concerns.
- The use of a rigorous model-based approach to capture, specify, and utilize MOS design information and to facilitate implementation.

A. Architectural Principles

As both a result of this effort and due to increasing application of enterprise and systems architectural approaches within NASA and JPL, a set of Architectural Principles have been articulated, vetted, and approved for the AMMOS by MGSS Program management. These principles follow industry best practices[4] with each including a definition, rationale, and implications. Here we note a subset (5 out of 9) that is particularly relevant to this work. We discuss those, and we also articulate two additional principles (Authoritative Source of Information and Develop With What You Fly With) that have become apparent as the result of our work in Ops Revitalization.

The principles form a core set of invariants that inform any of the current architectural, design, implementation, and sustaining work that maintains and evolves the AMMOS. Some are invariants that apply to any Mission Operations System. If either ignored or not accorded proper priority and consideration, the result will be an AMMOS that is less effective, less efficient, more risky to develop and operate, and more costly than necessary.

The relevant AMMOS Principles are:

- *Close the Loop*: AMMOS enables closed-loop control of flight assets, including reconciling the reported state of a flight system with science and engineering plans. To enable missions to efficiently and effectively perform planned-to-actual reconciliation, it is necessary to identify, architect, design, build, maintain, and operate elements of the System that respond to this principle.
- *Use of Common Services:* AMMOS provides operations capabilities via adaptable, loosely coupled common services. These expose adaptable portions to customers while maintaining key common aspects as part of the multi-mission system.
- *Learn from Experience:* For the AMMOS to improve and/or maintain capabilities in the face of imperfection and a changing world, it must be maintained and improved over time. This requires intentional effort to acquire and apply the lessons that experience offers.
- *Data/Information Visibility, Accessibility, and Understandability:* Data/Information is defined externally to any given user (including software systems or services) and is readily visible, accessible, and understandable to all authorized AMMOS and external partner users, software systems, and services.
- *Authoritative Source of Information:* Proper execution of flight operations requires having access to and knowing what is the correct, relevant, up-to-date information for the mission. Examples include the state of spacecraft subsystems, the currently running versions of flight and ground software, and the approved version of a command sequence to be transmitted to a spacecraft.
- *Develop with What You Fly With:* The MOS used to operate the mission should be the same as the MOS used during development (formulation, implementation, verification and validation).

As noted in Delp *et al.*[3], these principles provide a foundation for understanding strengths and weaknesses of the current AMMOS, and with understanding how to evolve the AMMOS toward the future state (MOS 2.0). In addition, they are central to the design and implementation choices made along that evolutionary path. Our design and implementation choices are evaluated in comparison to the principles; any that are inconsistent are set aside.

B. Stakeholder Interactions, Concerns, and ISO-42010

Interactions with stakeholders are a key feature of this effort[3]. In conformance to the ISO-42010 standard for architecture descriptions of systems, we work with stakeholders to understand their concerns. Those concerns are factored into the architectural description itself in the form of views that respond to concerns. A view may be a diagram, text, table, or some combination of these, and all views are maintained in the model repository along with all other architectural and design work (see section iii below).

This is a straightforward concept but somewhat complicated to put into practice. It commonly requires iteration to "peel the onion" and get to a specific attribute or characteristic of the system that can be addressed. Concerns may conflict with one another. Or, answering them may be outside the scope of the effort or capabilities of the architecture team.

In practice, we began Ops Revitalization with the capture of stakeholder concerns in the form of text and diagram-based scenarios for multi-mission operations[2]. Through analysis, the architecture team was able to articulate the principles of Closed Loop Control and Learn From Experience, as a response to those scenarios and based on their own experiences as MOS personnel. (In essence, this represents the architect or architecture team acting as one of the stakeholders). This was prior to our adoption of an architectural standard (ISO-42010) and an MBSE methodology.

Our practice today is more structured. We first socialize (introduce and explain) a particular aspect of the architecture in a meeting with our key technical stakeholders. This is done to introduce one or more topics and to set context and is followed by individual or small-group discussions between the architecture team and one or two discipline experts, in which details of the discipline's concerns are captured. The architecture team is then responsible for ensuring that the architecture responds to the concerns (if possible). Some iteration may be required to ensure effective communication. These small-group interactions are followed by a technical peer review, in which all the work is made available to the stakeholders for review for a 1-2 week period. A review meeting is then held to allow the group to hear reports from both the architecture team and the individual stakeholders. This approach represents a powerful vetting of the central tenets of the MOS 2.0 architecture, because each interaction represents an opportunity to expose any flaws or incorrect assumptions. It also enables stakeholders to more fully understand the products, easing acceptance of new ways of thinking about the MOS.

C. A Model-Based Methodology

A third foundation of Ops Revitalization is its adoption of MBSE for the implementation of multi-mission MOS products. In the past, systems engineering efforts have relied on document-based methods. We have adopted a model-based method out of necessity -- to address a system as large as complex as AMMOS would have been prohibitive without the efficiencies and insights we have obtained by using MBSE methods. Below are some of the key points behind our rationale for adopting and continuing to use model-based methods:

- They provide rigorous language for architecture and design (SysML and BPMN in Ops Revitalization), which yields better understanding. Implicit concepts are made explicit, and ambiguity of communication is reduced.
- Use of a modeling language and modern software for MBSE gives the system engineer better control over and insight into design information. Storing design information in documents means having to update multiple documents when any changes are made. Documents must be kept in sync. Proper modeling rigor enables updates to occur once - all related artifacts are kept in sync. Connections are frequently implicit or exist only in the reader (or author's) head. Modeling requires explicit connections that are persisted in the model (can't be forgotten). Documents can't be interrogated. Models can report on themselves, including their compliance with validation rules.
- Models are good at patterns. The multi-mission approach is all about reuse. Our Architectural work has identified key, reusable patterns. Models capture those patterns and greatly facilitate reuse.

III. Examples: Application of Architecture to the Multi-Mission MOS

In this section we describe several key examples of the application of architecture to propose solutions to stakeholder concerns. We first describe concerns as captured from stakeholders, then the viewpoint (or perspective) from which the concern is to be analyzed. From that viewpoint we identify opportunities to improve on the current system. According to our principles, we then propose improvements targeted at the concerns. This comparison of the "As-Is" to the "To-Be" system provides the motivation and rationale for the products of Ops Revitalization described in Section 4. Although necessarily less than a complete analysis of all concerns, the discussion below is representative.

A. Stakeholder Concerns

1. Concern #1: State of the Flight System

A key concern of spacecraft operators is having a clear understanding of the most recent state of the spacecraft and its subsystems. This is articulated in a variety of ways, but we see that the ability to quickly and easily assess state on the basis of telemetry, and the ability to make effective predictions of future state are central to operator's needs. This information forms a necessary basis for planning any future activities while maintaining positive knowledge and control over the flight system.

2. Concern #2: Lifecycle Support

A number of different stakeholders presented us with varying aspects of this concern. Flight system personnel emphasized the utility of having MOS support not only for testing of the integrated flight system but earlier in the lifecycle. A number of stakeholders focused on formulation phase and the need for improved ability to quickly and accurately analyze impacts on the MOS in support of mission-level trades or other studies.

3. Concern #3: Authoritative Sources of Information about the MOS

An area of concern for Line management and system engineers responsible across the MOS (for example, Mission Ops Systems Engineers) is the ability to have readily available the most current, complete, and correct information about the developing MOS. How mature or complete is the design? Are all interfaces identified? How complete is their documentation? And how do we ensure that documentation (much of it produced to satisfy internal (JPL) or customer (NASA) needs for Project lifecycle gate reviews) is up to date? In essence there is the strong desire to have the system report on itself and its progress against standards and other criteria.

B. Viewpoints, Views, and Analysis

In order to respond to the concerns, the architect must adopt some particular perspective from which to view the system (analogous to a plan view or an elevation view of a building). This viewpoint allows concerns to be considered in a manner that communicates architectural intent and is used as a basis for agreement or consensus. This is no different than an architect showing sketches or models of a public building to stakeholders (officials, neighbors, those who will work there) to gain consensus (or, in some cases, grudging acceptance).

Based on a particular viewpoint, we construct views that help to compare the current system to a future system concept and illustrate how stated concerns can be addressed. A complete set of views are captured in the architectural models themselves and presented in technical documentation. The diagrams and text shown here excerpted or abstracted from those materials.

For Concern #1 we adopt two viewpoints. From the Closed-loop Principles and Concerns about flight system state, we adopt a Control System point of view. This viewpoint looks at the MOS as a control system, attempting to see how and where the system ensures positive control of a flight system. The second Viewpoint looks at information - in particular the information as needed by operators to assess and predict state, and ultimately to plan and command the next set of flight system activities.

1. Control System Viewpoint, Views, and Analysis

Viewpoint. Figure 2 illustrates the principled, stakeholder-driven view of the system from the Control System viewpoint. It emphasizes the importance of its task to control the state of the flight system, making it a first-class aspect of the MOS and echoing one of our Principles. Considering the concern of flight system state, it is clear that

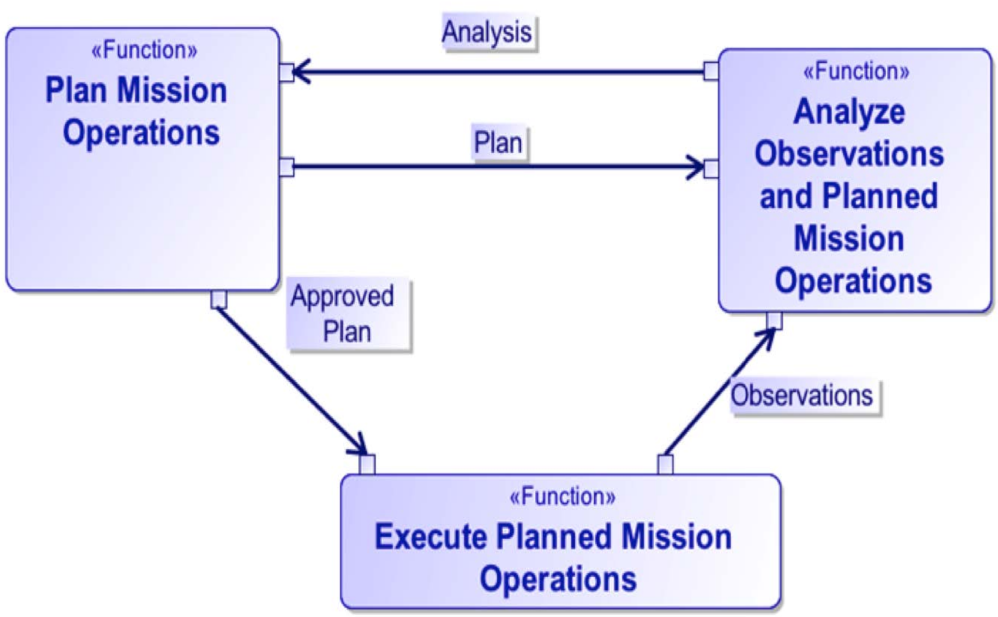

Figure 2. Illustration of MOS from Control System viewpoint.

the job of a MOS is to maintain positive control of the flight system. This is true whether the need is for some engineering activity (trajectory correction, spacecraft turn, or communications pass) or a set of science observations (capture images, radar, EM field, and plasma data during a flyby of Titan). In the To-Be view as illustrated in Figure

2, the MOS operates in a continuous loop. Plans are made, turned into flight system instructions and approved ("Plan Mission Operations"). The MOS must then execute the real-time activities during a communications pass ("Execute"), transmitting approved instructions and monitoring telemetry to spacecraft for it to execute. Observed information (e.g., telemetry, Doppler ranging signals) is then processed and analyzed ("Analyze") in comparison to plan information. The results of that analysis are fed into the next set of plans. As synthesized from stakeholder concerns and architectural principles, this is an essential pattern for the behavior of an MOS.

Views. Figure 3 illustrates the current concept of the MOS as a system. In it, the MOS is treated as a collection of functional entities that provide one another information. One way of describing what the MOS does (how it behaves) is to list the series of functions[5]. This method fails to provide the unifying or shared behavior -- what do all the pieces *do* that makes them a system and not a disparate collection of parts? The current paradigm is to consider the behavior of the system as "Uplink" and "Downlink" where "Uplink" is the set of activities that involves the planning of mission activities, the creation of commands or command sequences, and the transmission and radiation of those commands and sequences to the flight system. "Downlink" is the set of activities that include processing of telemetry and tracking information, and analysis and archiving of that data.

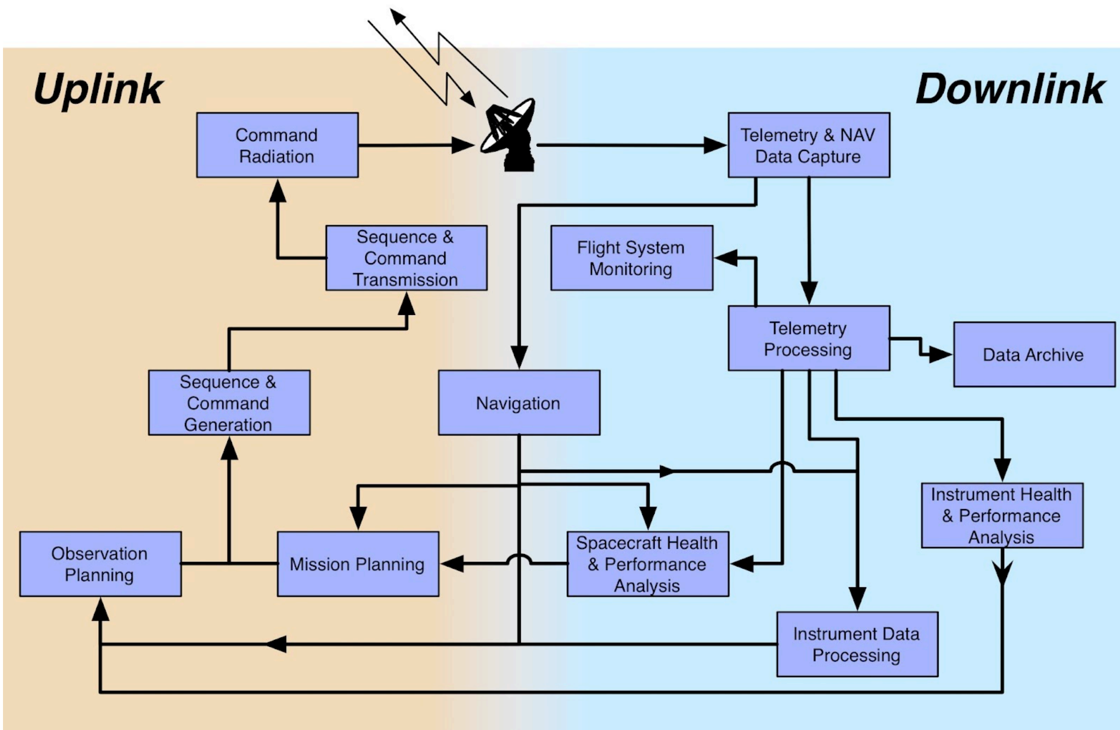

Figure 3. As-Is view of system from Control System viewpoint.

Analysis. In the As-Is view concerns regarding Flight System state are effectively segregated into two functional areas, Spacecraft and Instrument Health & Performance Analysis (Fig. 3), both in the "Downlink" portion of the MOS. This segregation by function has been mirrored in organizational and work breakdown structures. The result is that although analysis of flight system state based on telemetry is well taken care of (locally optimized); the feedback of that state information to "Uplink" (i.e., for the next set of planned activities) is treated as a secondary issue. Moreover, the current view commonly fails to illustrate the necessary feed-forward of predictions (based on plans) from Uplink to Downlink. But the comparison of prediction to measurement is of the utmost importance in spaceflight. In essence, it has not been architected into the system but rather added as just one of many necessary functions. One of the more telling examples of this is that the current system does not have built into it the capability to easily and quickly match planned activities and commands (e.g., for science observations, OTMs, calibrations) to the resulting data and information products. Partial accountability solutions have been built for specific missions (e.g., Mars Reconnaissance Orbiter, Spitzer Space Telescope) but these systems have not proved to be reusable.

Our analysis indicates the opportunity to make the system more efficient by better aligning with the needs of operators (and ultimately, the Mission itself) and focusing on the control pattern and behavior. In particular, this pattern clarifies the primary "job" of an MOS - to properly and safely operate the Flight System for the Mission. We find that the control system pattern recurs in many ways throughout the MOS[3]; we will briefly discuss broader architectural implications in Section 4.

2. Timeline Viewpoint, Views, and Analysis

Viewpoint. We also consider Concern #1 from the point of view of an operator who requires information in order to accomplish their tasks. Based on that Concern and the Principle that information should be visible, accessible, and understandable, we adopt a Timeline viewpoint for information (Figure 4). This viewpoint recognizes that nearly all information processed in an MOS has some basis in time, whether absolute or relative. Examples abound and several are shown in Fig. 4. The presentation of varying types of information in a timeline form (that is, a quantity of interest plotted as a function of time) is common practice in the MOS. It also provides a straightforward way to compare plans and predictions with measurements and the results of analysis It both yield useful visual feedback to operators as well as facilitating analytic comparisons between plans and measurements.

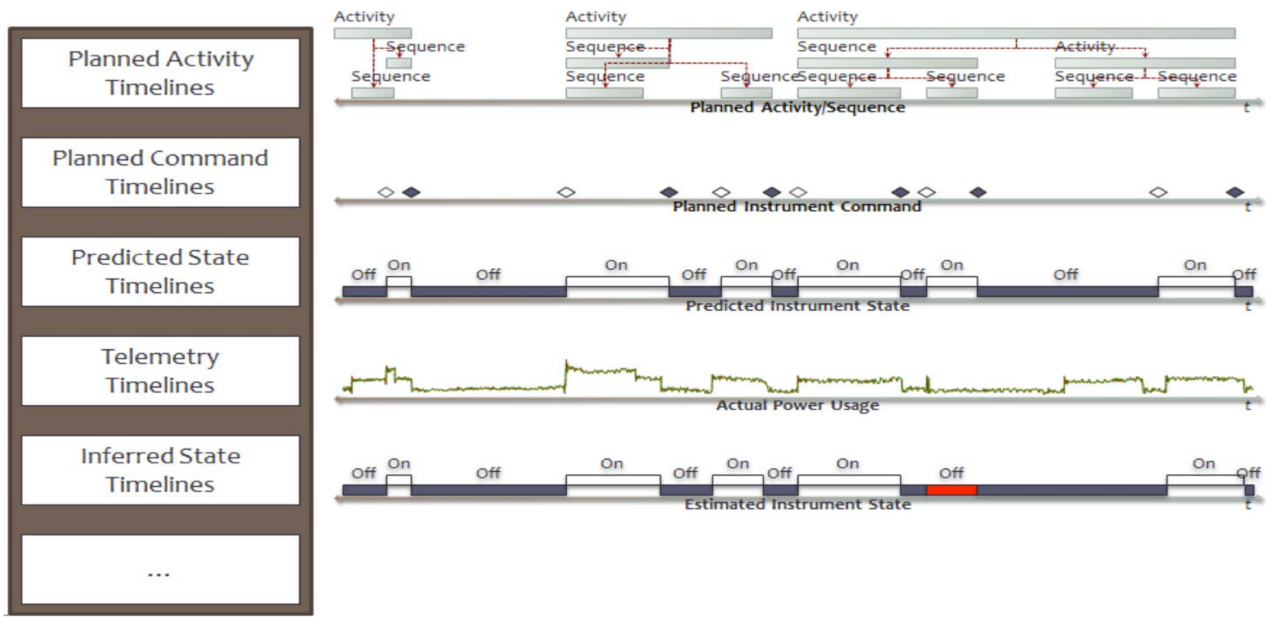

Figure 4. Illustration of MOS Information from the Timeline Viewpoint

When considered in the light of our Principle of Authoritative Sources of Information, we can synthesize the Timeline and Control System concepts in Figures 2 and 4 as shown in Figure 5. Such a synthesis responds to stakeholder concerns about state as well as their need for information, and is consistent with our principles. The diagram envisions that MOS activities are able to share a common, single source of information that facilitates the work of operations. Such a synthesis suggests the use of common repositories of information, supported by common formats and presages the work of Chung et al.[6] and Reinholtz[7], who present the foundation for an AMMOS information architecture that is based on and implemented using Timeline.

View and Analysis (Dependencies). Looking at the current MOS from the Timeline viewpoint, we first note that information is stored and transmitted as files. To obtain the information they need, today's operators must deal with a large number of diverse files. In Figure 6 we illustrate some of the complex dependencies of various files commonly used in the AMMOS. We observe that file names are commonly used as a stand-in for the information content of files (e.g., engineers speak and write about "the SPK" as opposed to "the trajectory" for a given spacecraft). This is a reflection of treating information as secondary to format. As we see below, this results in additional work to extract from files the information necessary for operations work. Further, there are complex and multiple dependencies between files. This reflects the current practice of serial execution of various software applications and transmittal and receipt of files to and from various functional elements of the MOS. The large number of files and their interdependencies requires many interfaces, which grow approximately as the square of the number of different files. This kind of integration scales poorly as the number of files grows – cost, operational efficiency, and risk all increase disproportionately to the number of files added. Moreover, it presents challenges to system integration and test. In the current system, comprehensive records of this web of interconnections are not available and it is difficult, in general, to predict all the effects of making a change to one file or another.

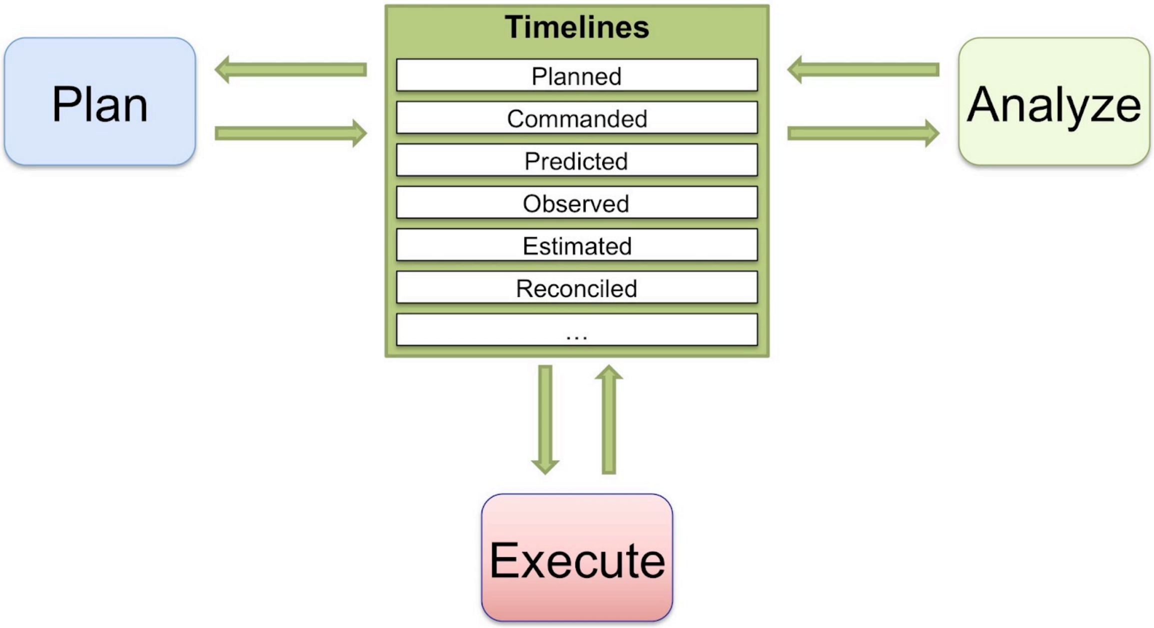

Figure 5. Synthesis of Control System concepts with Timeline according to the Authoritative Source of Information principle.

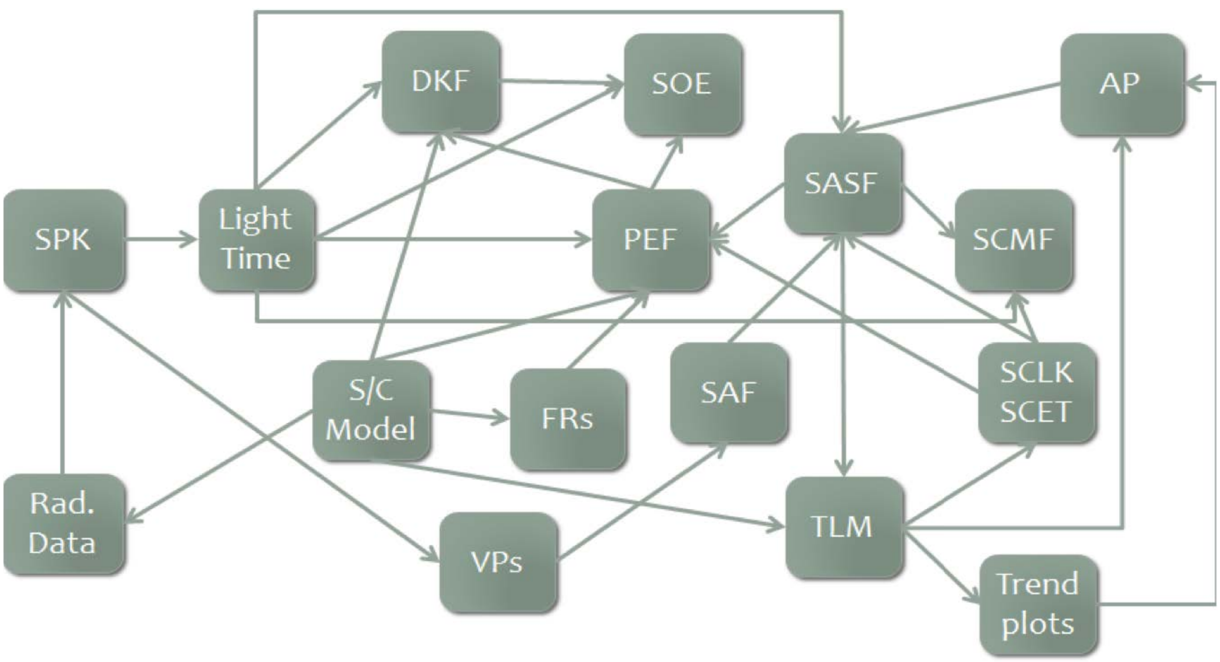

Figure 6. As-Is dependencies between various MOS files.

View and Analysis (Redundancy) From a Timeline Viewpoint (Fig. 4) we also look at the information content of the various files in the current system. The result is shown in Figure 7, which is a representation of the redundancy and information overlap in the current set of files. For example, information in the Predicted Events File will also be present in the DKF and Light Time files. The question then arises as to which is the most correct or most current version of that information -- which is the authoritative source? In the current system, we have strict procedures, file management, and configuration management and control systems to ensure that only the "right" version of a particular set of information is utilized. Similar to the information dependencies (Figure 6), overlap and redundancy impose cost (e.g., for configuration management systems) and risk (e.g., of confusion as to which version of a file to use) on the system.

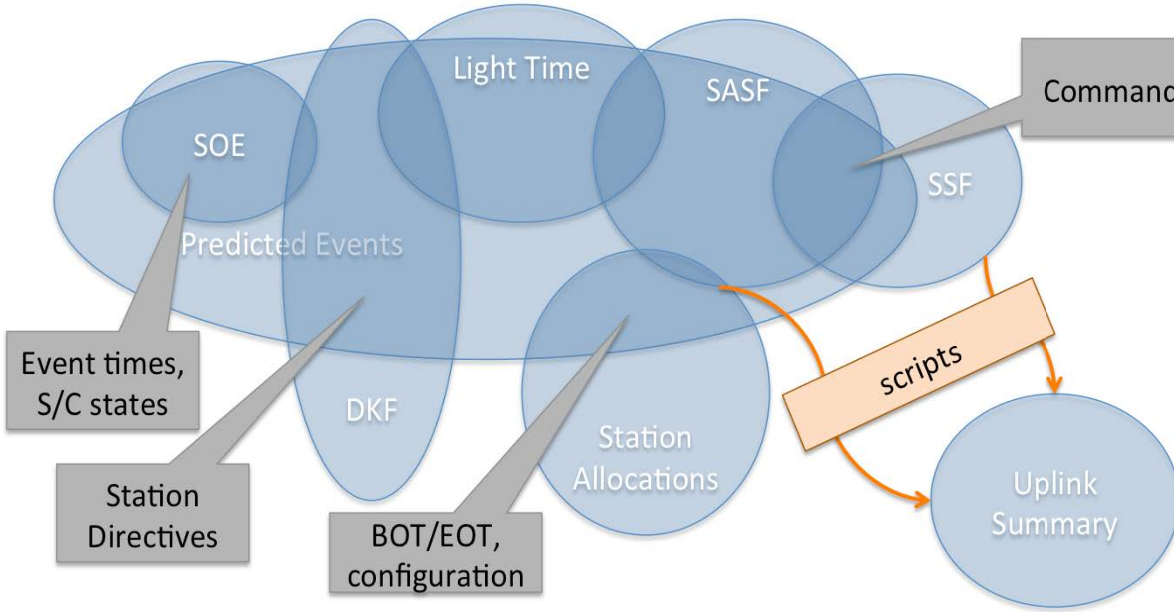

Figure 7. Illustration of information redundancy and overlap in the current system.

Figure 7 also illustrates a tendency to continue to add software and files to the system. Because the current files and their presentation of information do not satisfy operator's concerns, operators request new representations of data. The requests are accommodated by software (commonly scripts) that produces additional files. These become ingrained in operations and are added to the files and software already requiring configuration management, maintenance, and updating. Or, worse, they are added in late and are only managed informally and represent hidden risk in the MOS.

Our analysis indicates that by properly architecting our MOS information based on timelines, we can take advantage of opportunities to

- Better align representations of data with operators needs for information.
- Decrease the number of unique information interfaces and decrease cost and risk of integration of those interfaces.
- Decrease the effort (and cost) required to ensure proper configuration management and simplify the identification of authoritative sources of information.

3. Adaptation Viewpoint, Views, and Analysis

Viewpoint. Considerations of Concerns #2 and #3 require that we view the AMMOS from a perspective that includes MOS development in a Project context (formulation and implementation, aka Phases A, B, C and D), including support of a multi-mission product line across multiple project lifecycles, as well as on a continuous, ongoing basis. We also focus on the role of AMMOS during Project development, as it undergoes mission adaptation. This viewpoint responds to all the Principles listed in Section 2 with the exception of Data/Information Visibility.

Views. This part of the adaptation viewpoint is illustrated in Figure 8. Here, AMMOS is deployed such that it conforms to the "Develop With What You Fly With" Principle. Products chosen by a Project are deployed in an operational state to a

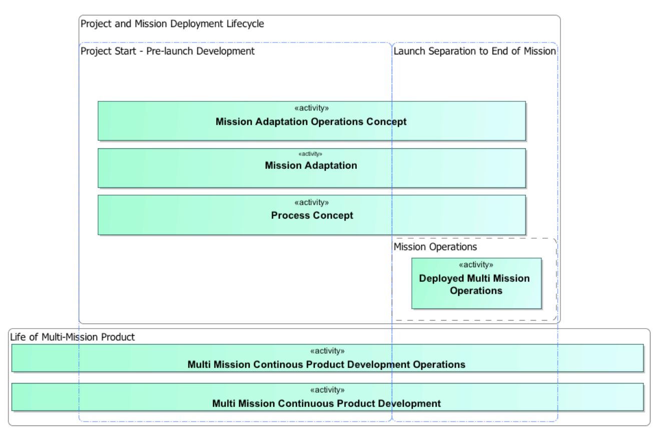

Figure 8. AMMOS Products Lifecycle

project (Mission Adaptation Operations) enabling tabletop (or more sophisticated) simulations even in formulation phase. The initial deployment to the project undergoes adaptation to better support mission needs (Mission Adaptation). Adaptation continues, as necessary, until the mission completes its objectives and is retired. The AMMOS has internal processes for responding to and tracking requests for work (Process Concept). Independent of Project lifecycles, MGSS continuously develops the multi-mission product to keep pace with evolving mission needs, changing technology, and innovation (Multi-Mission Continuous Product Development). It conforms to the principle of Learn from Experience and leverages its Operations capability and experience gathered from previous missions to vet these updates (Multi-Mission Continuous Product Development Operations).

We add to this point of view a consideration for how AMMOS products will undergo adaptation for the purposes of providing a working MOS that meets the needs of the Mission. This addresses concerns about the state of readiness of AMMOS products and their verification and validation. In keeping with the principle of Develop With What You Fly With, we consider the usage of the multi-mission system throughout development. And consistent with a principle of Common Services, we adopt a model in which the AMMOS is a service and consists of services, each service having its own internal workflow for accomplishing its tasks (Figure 9).

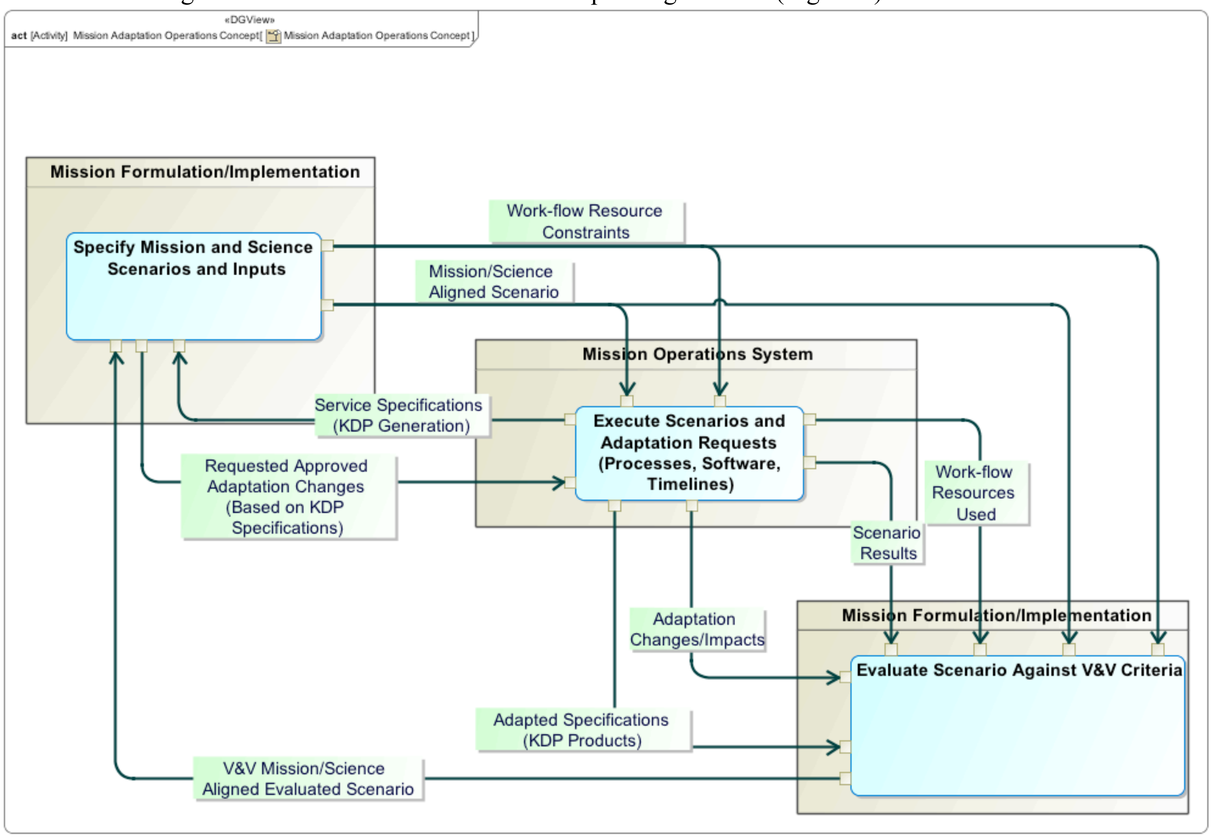

Figure 9. Adaptation of AMMOS in Mission Context, using scenarios to drive adaptation throughout the lifecycle (formulation, implementation, and V&V).

Use of multi-mission scenarios and service-internal workflow allows incremental evaluation and adaptation for the purposes of the Mission. Scenarios are used to describe how the AMMOS products will perform work. Workflow constraints are used to assess whether performance meets the Mission's needs. V&V is performed at each step throughout the entire adaptation effort - from formulation through implementation and deployment to operations. Each increment of adaptation thus proceeds:

1. Utilize Mission and Science Scenarios to determine validity of AMMOS products for Mission.
2. Elaborate Mission Scenarios and adapt (if needed).
3. Incrementally verify and Validate throughout development of the mission.
4. Upon deployment - continue to update (if needed) though adaptation.

In response to concerns about authoritative information about the MOS itself (#3), AMMOS uses introspection to provide the detailed specifications of its implementation (e.g., design documentation, required at Key Decision

Points (KDP) such as PDR and CDR). This reporting capability also provides information the Project can use to perform trades and/or evaluations necessary to determine feasibility of adaptation. We note that this reporting also responds to the Close the Loop Principle. Adaptation Impact reports illuminate the deltas between the multi-mission specification and the mission specific adaptation as implemented and "close the loop" on Project expectations. This information allows the project to maintain maximum multi-mission utilization.

Analysis. Comparison of the current and To-Be systems shows points of similarity, but these are commonly either implicit or incomplete. For example:

- AMMOS software products do consider both Project lifecycles and their own continuous updates. But AMMOS Teams (the current analog to Services) are almost exclusively focused on Project lifecycles and have no strong concept or funding for ongoing maintenance or updating of their capabilities.
- Although V&V is considered necessary, it is commonly provided in the form of a V&V systems engineer who is brought on as late as possible (for cost reasons) and not as an integral part of the formulation and implementation of the MOS.
- Reports from the current system come in the form of documents that are treated as distinct deliverable items, and in some cases are considered a formality rather than being integral to the development of the MOS.

Finally, there is a strong overall tendency for documentation, processes and procedures, and even some software to be informally handed down from one mission to another. This kind of inheritance, referred to as "clone and own," occurs because of the lack of a clearly defined mechanism for feeding mission specific lessons and applicable enhancements back into the AMMOS. For software, MGSS is currently developing an open-source repository to facilitate incorporation of mission-developed tools into the AMMOS. But no systematic mechanism exists at this time for ensuring capture of other key products such as MOS documentation of designs, process, procedures, etc.

From the Project Lifecycle point of view, the following opportunities exist:

- Explicit feedback of mission development experience back into the AMMOS to ensure that productivity enhancements are not lost (or re-invented) from mission to mission.
- V&V can become a continuous activity rather than a late-phase challenge. Errors are discovered and fixed early and cheaply as opposed to late in implementation at much expense (or are accepted as operational risk).
- Reports (including KDP gate transition documents) are produced as a routine and integral part of the engineering work. The future system uses models to capture design information, so documents are simply reports from the authoritative models, rather than standalone artifacts.
- The use of introspection to allow the system to better report and assess its own progress. Such knowledge helps missions and the AMMOS to know what adaptation is truly required and to maintain a clear record of adaptation work that has occurred.

C. Summary

This section illustrates our usage of stakeholder-focused, principled architectural approach to identify opportunities to improve upon the current AMMOS. The Control System, Timeline, and Lifecycle and Adaptation concepts each offer value propositions that translate to reduced costs, lower risk, and/or increased efficiencies. They represent fundamental and pervasive MOS architectural patterns. Such patterns are particularly useful when leveraged using model-based methods (which enable efficient application of patterns to various aspects of the system) and will be the subject of a future paper [Delp and Bindschadler, manuscript in prep.]. In the next section we describe the results of leveraging those patterns using MBSE techniques to produce a robust set of Mission Services that provide Missions with the opportunity to develop and operate a more efficient and higher-quality MOS at lower cost and risk.

IV. Realizing the Architecture as MOS Products

In creating a set of AMMOS Mission Services we use the architecture to inform them, and MBSE methods to execute them. Additional analysis (not described in this paper) has also lead us to apply a Services model to our implementation. This follows from our principle of Use of Common Services as well as stakeholder input, and provides a method for disambiguating functional concerns of the system from organizational concerns.

The result of our work is an integrated system that includes both Services and the "connective tissue" (the System) that enables them to function smoothly together. The products of Ops Revitalization completed or in progress to date include:

- Information on the System and the Services - the architectural/design models as captured in a modeling tool and stored in a model repository (database). The latter is the authoritative source for information about the To-Be System and its Services.
- Reporting capabilities - model constructs and software enable and facilitate production of reports and documentation from the models. Standard documentation sets include KDP gate transition documents (for NASA Projects), adaptation reports, and training materials.
- An adaptation methodology and guide giving information on how to adapt Services to Mission needs.
- Catalog information products for (1) timelines and (2) processes. These contain products that are either invariant from mission to mission or that can be customized in the course of mission adaptation. (Catalogs for items like telemetry and command dictionaries would have similar value but are outside the scope of this effort).

A. Mission Services

The Mission Services defined thus far represent a core set of MOS functionality and are directly engaged in the business of acting as a control system for a deep-space science mission. Other supporting services (e.g., Configuration Management) could be defined but are not currently planned. A short definition of each Mission Service is shown in Table 1. Note that Service names are provisional at this time and thus subject to change.

Table 1. Mission Services

Mission Engineering	Mission Engineering controls the integrated ground activities for the Mission during Operations and Development. Among other tasks, it ensures coordination of planning, sequencing, and commanding during operations, maintains external interfaces and agreements, and monitors MOS performance throughout the lifecycle
Flight Systems Engineering	Flight Systems Engineering is responsible for flying the spacecraft under control, including its subsystems, during Operations, and for the development of such capabilities during earlier Phases. Among its major tasks is the planning of spacecraft activities and monitoring of performance vs. predicts so as to maintain spacecraft health.
Science & Instruments	Science and Instruments Service is responsible for control of the Science payload during Operations, and for the development of such capabilities during earlier Phases. Further, it is responsible for planning and integration of science activities and commands, and "closing the loop" between planned science activities and the return of science data.
Flight-Ground Communications	The Flight-Ground Communications Service provides Project-side control of the DSN ground stations. It communicates Project intent for ground station configuration and evaluates ground station telemetry against that intent.
Navigation	Navigation Engineering is responsible for control (planning, predicting, estimating, and analyzing) spacecraft trajectory.

Each Service includes the capability to generate its own engineering specifications, which satisfy needs for self-reporting and generation of technical documentation at KDP reviews (management documents such as implementation plans or policy documents are not in scope). The standard set of documents and reports are illustrated in Figure 10. Definitions of each are provided in the Glossary in Appendix B. Below we give a brief summary of the documents.

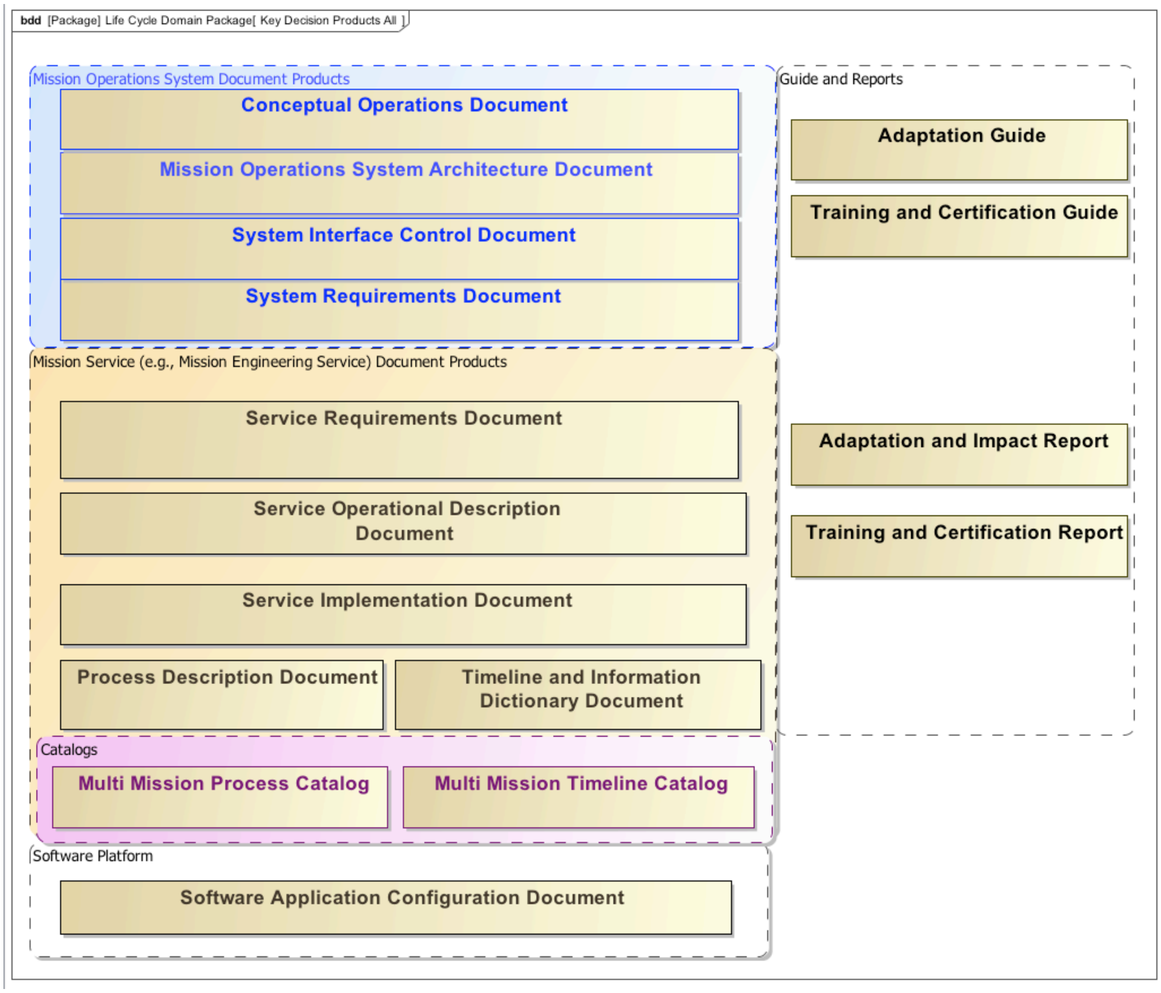

Figure 10. Standard set of System- and Service-level documents. *See Appendix B (Glossary) for brief definitions of each document. Note that each Service has its own complete set of documents (Requirements, Operational Description, etc.).*

B. System-Level Document Products

System-level documents include an Conceptual Operations Document that contains scenarios that give context to (and may drive) requirements, a requirements document, a design document, and an ICD capturing the interfaces between the MOS and other systems (e.g., DSN, Flight System) along with the functional dependencies in the MOS. Guides for adaptation and for training and certification, and corresponding reports round out the set. With the exception of the adaptation guide & reports, this is a conventional set of MOS documentation. The difference is that these documents fully linked and consistent because of the underlying model. Changes to a portion of the model are reflected as soon as documents are regenerated. And because document generation is built into the models and facilitated by DocGen[8], changes can be made available on the Web within minutes.

C. Service-Level Document Products

The Service-level documents provide a reviewable specification of each Service and contain operations concepts and scenarios, requirements, interfaces and agreements, processes and procedures mapped to the roles that perform them, and a specification of the information (timeline document) for which the Service is responsible. See Appendix B for more details.

Informed by our architectural work, documentation responds to our core Principles and Stakeholder Concerns. The ability to quickly and comprehensively update documents responds to Concerns #2 and #3 (Lifecycle and Authoritative Information about the MOS). This is made possible by System and Service models that are founded on the Authoritative Source principle. The creation and application of multi-mission catalogs for Timelines and for Processes is in response to the Learn from Experience Principle.

D. Examples and trace back to Principles/Concerns

In the following we examine two examples from the System and Services models as illustrations of how they realize a principled architecture.

1. Control Interactions and Interfaces Example

Realizing the MOS as a control system raises the issue of how closed-loop control information is passed back and forth and expressed between the MOS and other Systems, and between Services. Figure 11 provides a notional view of control relationships from the MOS perspective.

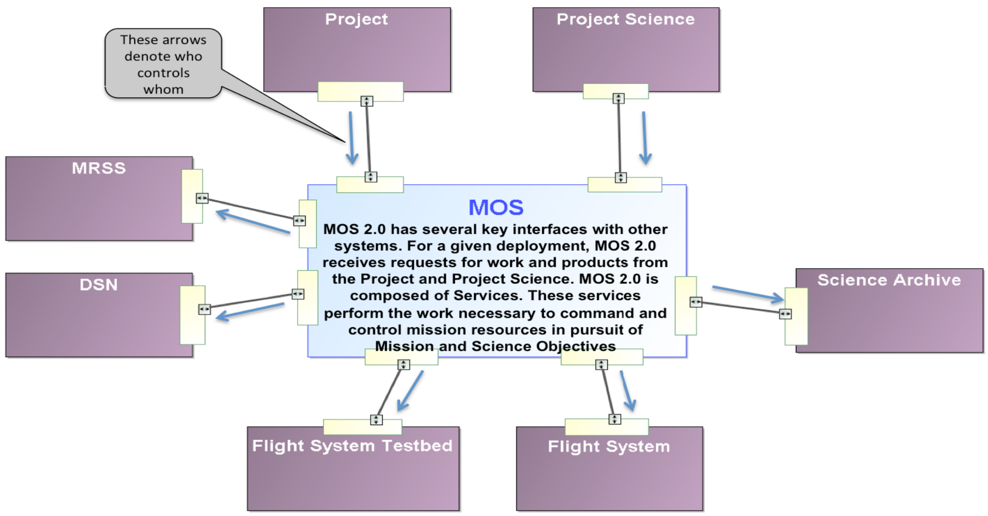

Figure 11. Control relationships and interfaces between MOS and other Systems. Arrows denote whether a system issues commands or directives to the MOS or accepts them.

In detail, System interfaces require bidirectional flow - commands in one direction and responses the other. An example is shown in Figure 12. Here the MOS provides commands for the configuration of a DSN antenna station in the form of a DSN Keyword file (DKF). The station responds with DSN Monitor Data. The MOS can then analyze the provided data and compare with predictions to assess antenna performance and ensure that the configuration is the proper one for a particular spacecraft contact. The same pattern applies to the MOS to Flight System interface - any commands or directives need to be matched with a response that can be used to maintain positive control, as captured in the Principles.

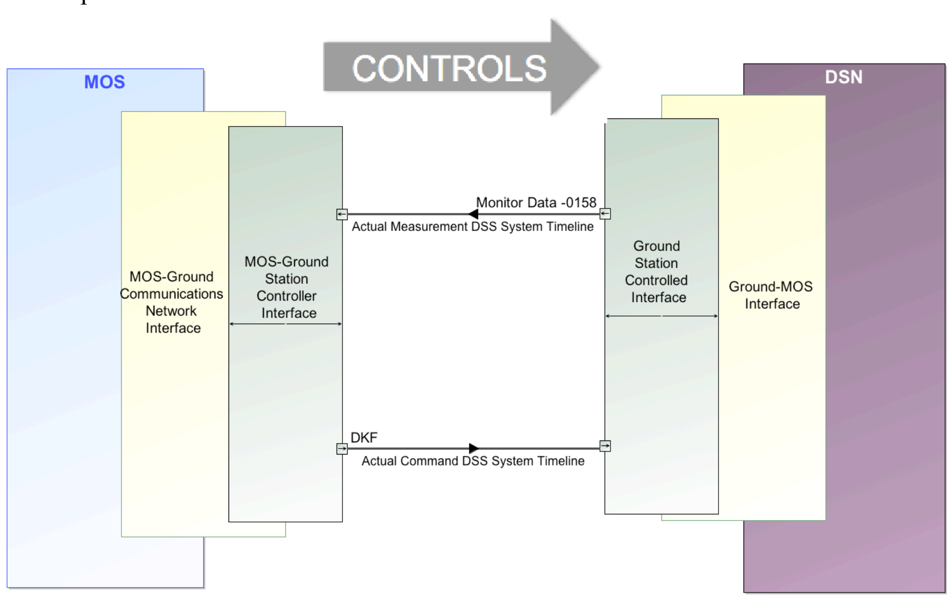

Figure 12. Example System-System interfaces. *The DKF provides the configuration commands for the DSN antenna operators. DSN Monitor data is the telemetry from the antenna used to verify proper configuration.*

The DSN-MOS interface is straightforward in that commands and carried out with minimal latitude (e.g., a transmitter is either on or off). Other interfaces are less deterministic and can be characterized as one entity *directing* another. Figure 13 shows an example of a "directing" interface between two Services. In this case, Mission Engineering (MES) establishes the general course of mission activities and sets forth Mission-level priorities and constraints (based on Project and Project Science directives - see Fig. 11). The directing entity (MES) cannot rigorously predict the response to its directives, but rather constrains them to acceptable solutions. For example, a Mission Timeline may contain timing and duration for an orbital insertion maneuver. Some or all science instruments must be off or cannot collect useful data during the maneuver. The directed entity (Science & Instruments) must respond within that constraint but has latitude to choose which set of payload activities is the best, proper, or most effective one.

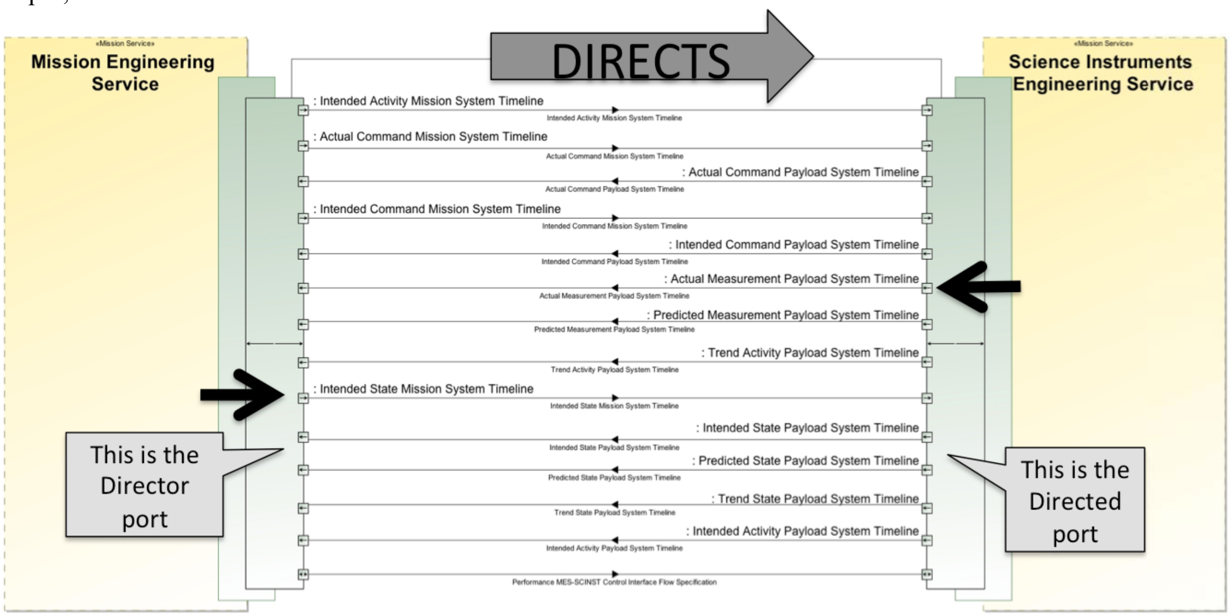

Figure 13. Example of interfaces between two Services in a Director-Directed relationship.

The application of the Close The Loop principle has significant utility in that it insists on accountability - for any command or directive, there should be some response from the commanded/directed entity. This gives the engineer a powerful heuristic for ensuring completeness of interfaces. It responds to opportunities to simplify and clarify the job of the various parts of the MOS - by focusing on closing the loop of commands or directives, actions taken in response, and reporting of results, we capture the drivers on work to be done by the System or its Services.

2. Timeline Expressing State Over Time Example

A considerable portion of the System- and Service-level models are concerned with the application of Timelines to the MOS. Here we show an example of how the state of a specific spacecraft component (in this case a Li-Ion battery) is represented as a set of Timelines and how those Timelines are applied.

In this example, we show how the State of Charge (SOC) of the battery is represented as timelines and how those timelines are used to provide the operator with necessary information to ensure positive control. The battery and SOC are the responsibility of the Flight System Service. From the generic SOC timeline, the set of specific timelines are specified to provide the operator with control information (Figure 14). Specifically, the operator needs to (1) communicate intention, (2) make predictions as to what measurements will result, (3) have a record of actual commands and measurements and (4) perform trending to forecast SOC into the future and provide inputs to the next planning cycle.

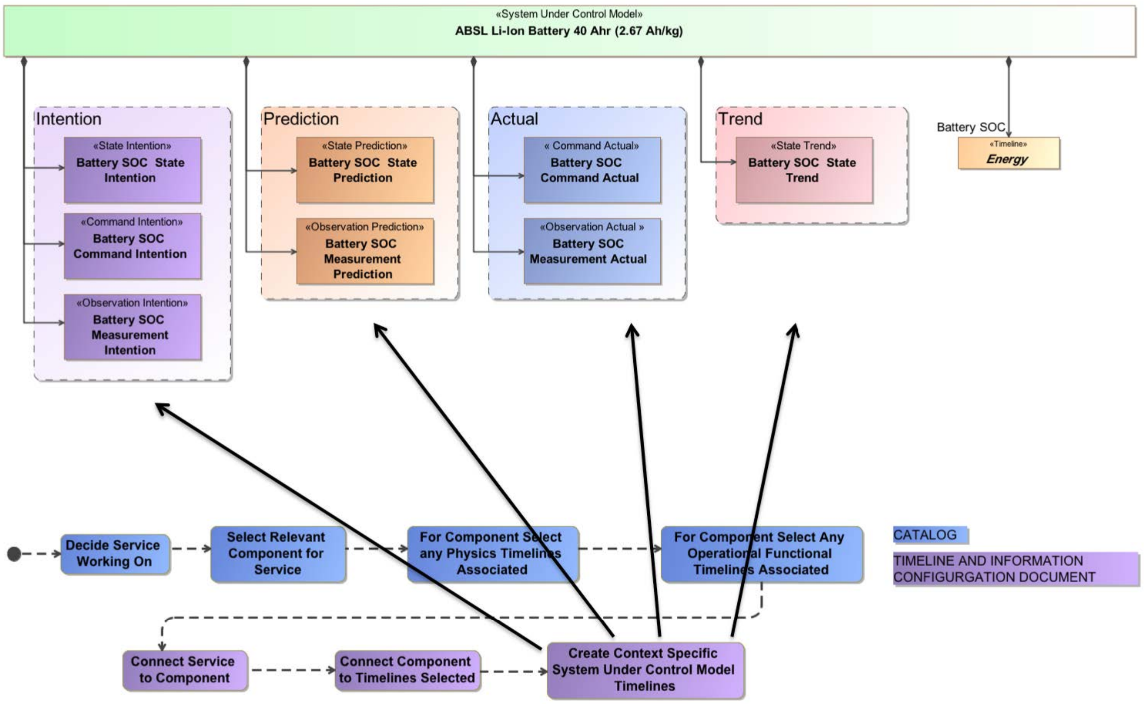

Figure 14. Definition of various timeline types needed for control of Battery SOC (upper part of diagram) along with steps taken (lower portion) to select a generic timeline and specialize it to meet mission needs. Colors in the lower part indicate where timelines used or created are captured in documentation.

In general, all these types of information (timelines) are necessary if the operator is to maintain proper knowledge and control. This can be illustrated by placing the various timelines in the control-loop context (Figure 15). The intention timelines (State, Observation, Command) are each different options for operators to represent an intention for the SOC (during adaptation, a Mission may select one, two, or all three as needed). From Intention timelines, Predictions (of either State or Observation) and Command Actual timelines are produced. The latter are sent to the flight system (not shown) and result in an Observation Actual timeline consisting of telemetry readings. Analysis is done based on Actual and Prediction information, and a Trend timeline is produced that shows SOC state as estimated from observed information and projected forward in time. The Trend information at a particular time is fed to the Intention timelines, setting the initial condition for SOC at the start of the next planning interval.

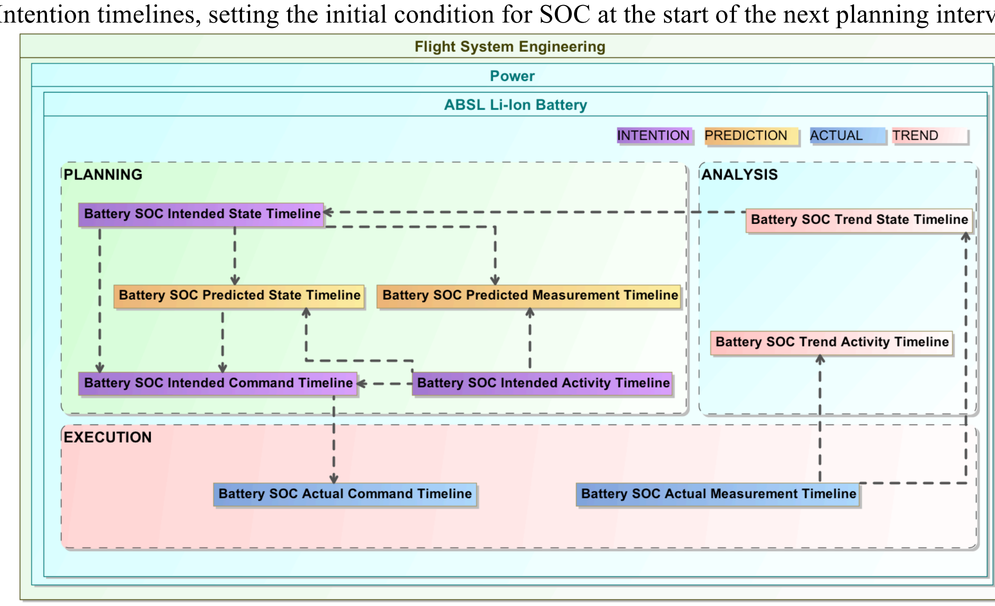

Figure 15. Battery State of Charge timelines shown in an operational context.

This application of timelines addresses opportunities to make data align more transparently with operation's need for information. In contrast to a file-based system, where each of the eight timelines would need its own file and the associated effort to manage and control versions of those files, the Control/Timeline synthesis (Figure 5) illustrates use of a central repository to minimize the cost and effort of that control. Moreover, in the current system, Intention is in one file format, Prediction in another, Observation in a third, and Trend in a fourth.

E. Summary

In this section we have illustrated examples of how a principled, stakeholder-driven architecture is translated into system and service designs. Supporting this translation are architectural patterns for Control System, Timeline, Services and other key components of the architecture, synthesized as our Mission Services Architecture Framework. How those patterns are created and synthesized and the details of their application are the subject of a future paper [Delp and Bindschadler, manuscript in prep.]. For now, we simply note that the Framework is the fundamental structure that synthesizes Principles, Concerns, and architectural patterns. This structure enables us to produce services that respond to these various drivers and still maintain the overall coherence required to reduce cost and risk and increase quality and efficiency.

V. Product Usage and Value Proposition

The results of the Ops Revitalization effort will be a set of multi-mission products that are ready for Project adaptation, along with Adaptation Guides for how to do so. The Guides are still in development; we briefly outline here what we know at this time. The models of the System and Services, including multi-mission catalogs (timeline and process), and the modeling infrastructure used to create them represent significant leverage for a Project MOS development team. Adaptation of the System and Mission Services will follow an adaptation process that parallels the Ops Revitalization development, greatly simplified and facilitated by virtue of requirements, timelines, interfaces, agreements, and processes that only require minor changes as compared to invention (or re-invention). Along with a delivered repository of models, model profiles, productivity scripts, report and document generation (DocGen) capability, and document management (DocWeb), we plan to have web applications that facilitate adaptation and usage of model-based processes.

The result is a repository of MOS knowledge and structure along with tools that enable interrogation, update, and customization (adaptation) of that knowledge to serve a Mission's MOS -- and provides significantly more value than an electronic repository or bookshelf full of electronic or paper documents. There are manifold arguments to be made for the value of the products outlined here; we focus on three:

- *Productivity / Affordability / Quality:* We have demonstrated the ability to produce significant, high-quality engineering products for much less effort than the conventional MOS systems engineering practice. In our experience the work to produce documents (and the engineering behind them) can be done for less than one-half the current effort required. Use of modeling techniques enables automated validation and error checking of interfaces, processes, or other products so as to increase quality without increased cost. And because analyses are facilitated, larger design or trade spaces can be investigated for a given cost.
- *Management of complexity:* The evolution of technology and increasing sophistication of science investigations drive missions to be more capable and more complex. This is reflected in the tasks that an MOS must perform -- and commonly at a lower budget than the previous mission. The use of an authoritative set of models supported by software acts as a prosthetic memory, particularly for the routine, the obscure, or the extremely detailed particulars that define our software-intensive MOS today.
- *Pathways to Automation:* The ability to rigorously model MOS processes, and to capture interfaces and detailed information models is enabling for automating, particularly routine tasks. Today's MOS commonly captures such opportunities as informally developed software scripts. These are mostly informal low-cost, one-off solutions that are rarely reused and introduce risk. A strong reference architecture and a deliberate process of continuous improvement better enable single-point, grass-roots solutions to become incorporated as part of a reusable set of MOS capabilities that are fully integrated with the rest of the multi-mission system. Better choices can be made about which functions to automate. And looking farther forward, application of these methods to a unified Mission Architecture (flight and ground) would enable migration of functionality from the ground system to the flight system in cases where risk and cost allow. Such capability has been in demonstration for some time on EO-1[9,10] and has enabled significant

savings while retaining high science return and observational efficiency, but has not gained wide acceptance. We suggest that a necessary part of gaining such acceptance is similar architectural work applied to the overall Mission and to the Flight System.

The value proposition for MOS 2.0 as the future state of the AMMOS is simple. Adaptation is a proven road to affordability and risk reduction where AMMOS software has always excelled. Stakeholders across mission operations have not only advocated for such a solution for operations, they have participated in our effort and helped drive the design through the architecting process. They have contributed in identifying the key issues with the existing system and have told us that the work to date meets the goals we've set out in the architecture.

VI. Conclusions

By taking a principled and stakeholder-focused approach to architecture, we have identified multiple opportunities to improve on the current system in terms of cost, risk, efficiency, quality, and responsiveness to operators. Identification of powerful architectural concepts and patterns enable translation of those opportunities into engineering specifications for a business architecture, information architecture, and Mission Services that can be phased into the AMMOS over time. These will also drive a new generation of software capabilities that better respond to the new architecture -- and have already begun to do so. For example, an updated set of planning and sequencing software is currently in work as an update to the venerable SEQGEN software applications. This new software includes core concepts like timeline and an authoritative source of information as part of its design[7]. In the next two years we anticipate completion of the Mission Services and other capabilities discussed in Section 4 and look forward to application to missions either in flight or in development.

Acronym List

MOS	Mission Operations System
AMMOS	Advanced Multi-mission Operations System
MGSS	Multi-mission Ground System and Services Program Office
MBSE	Model-Based Systems Engineering
SysML	Systems Modeling Language
BPMN	Business Process Modeling Notation
ICD	Interface Control Document
DSN	Deep Space Network

Glossary

Architecture:	Architecture is the structure – in terms of components, connections, and constraints – of a product, process, or element [Rechtin and Maier, The Art of Systems Architecting]
BPMN:	The Business Process Modeling Notation (BPMN) is a graphical notation that depicts the steps in a business process.
Concern	A statement by a stakeholder expressing his interest in a quality or characteristic of the System
Principle:	Principles are general rules and guidelines, intended to be enduring and seldom amended, that inform and support the way in which an organization sets about fulfilling its mission
Stakeholder:	A stakeholder is a person or group of people, who have something to gain or lose by the actions of the project because of the selected architecture.
SysML	The Systems Modeling Language (SysML) is general-purpose visual modeling language for systems engineering applications.

Glossary of Standard Documents

System-Level Documents

Conceptual Operations Description	Captures key MOS concerns and responds with scenarios that cover those concerns across the lifecycle of the Mission. Also provides mapping of operational scenarios (i.e., from the perspective of ground operations) to mission scenarios (i.e., from the perspective of execution of mission activities by the flight system).
Mission Operations System Architecture Document	Documents how the Conceptual Operations is implemented in the System. Shows how closed-loop control is achieved and describes the information needs of the MOS. Elaborates Services and their capabilities as needed to realize the control strategy. Establishes system-level interfaces and agreements and their delegation to individual Services. Reports on the complete set of processes executed across the System and the Roles (expertise) required to fulfill them.
System ICD	Detailed specification of Interfaces and Agreements between the MOS and external Systems (e.g. DSN, Flight-Ground), lists dependencies to MOS functions.
System Requirements	Level 3 functional and performance requirements
Training and Certification Guide (System/Service)	Contains process for System/Service operations training and certification criteria for operations personnel.
Adaptation Guide (System/Service)	Describes the process for performing mission-specific adaptation to the System/Service models, along with verification criteria for adaptation
Adaptation and Impact Report (System/Service)	Standard report that can be used to analyze and impact proposed changes to the system (whether due to adaptation or system trades). Also can be used to provide record of all changes made in the course of mission adaptation.
Training and Certification Report (System/Service)	Standard report to indicate status of training of personnel; record of certification for operations
Multi-mission Timeline Catalog	Catalog of all reusable or adaptable timelines.
Multi-mission Process Catalog	Catalog of all reusable or adaptable processes.

Mission Service-Level Documents

Service Requirements	Level 4 Functional and Performance Requirements
Service Operational Description	Describes how the service performs in operations. Includes scenarios for information flows, provides context for Interface Specifications, and Agreements with other Systems/Services.
Service Design Description	Describes the processes and roles needed to realize the Service's capabilities, defines the Interface Specifications, and Agreements with other Systems or Services, lists the s/w applications used to support process execution and the information (timelines) utilized and output by the Service.
Process and Procedure Description	Complete specification of detailed processes and procedures for a Service.

Service Timeline and Information Specification	Specifies the detailed Information Model for a Service. Specifies the Timelines the Service is responsible for, their behavioral relationships and their mappings, to Command and Telemetry Dictionaries
Software Application Configuration	Specifies the Information and Software Application dependencies of process/procedures. Provides a detailed mapping of software and information required for each process and procedure.

Acknowledgements

We acknowledge the work of the Ops Revitalization team making this paper possible. Special thanks for support with diagrams from Carlos Carrion, Elyse Fosse and Louise Anderson, and Seung Chung, and editing and commentary from Michelle McCullar. We also thank our MOS Stakeholders for their generosity in sharing their insights. This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1]Bindschadler, D.L., Boyles, C.A., Carrion, C., and Delp, C.L., "MOS 2.0: The Next Generation in Mission Operations Systems," SpaceOps 2010 Conference Proceedings, Huntsville, AL, Apr 25-30, 2010.

[2]Carrion, C., Delp, C.L., Illsley, J., and Liepack, O., "Use of Operational Scenarios in Architecting MOS 2.0", SpaceOps 2010 Conference Proceedings, Huntsville, AL, Apr 25-30, 2010.

[3]Delp, C. L., Bindschadler, D., Wollaeger, R., Carrion, C., McCullar, M., Jackson, M., Sarrel, M., Anderson, L., Lam, D., "MOS 2.0 - Modeling the next revolutionary mission operations system," 2011 IEEE Aerospace Conference Proceedings, AERO 2011, May 13, 2011.

[4]TOGAF™ Version 9, Enterprise Edition, The Open Group Architecture Framework, (TOGAF), Van Haren Publishing, November 2008, pp. 265-280.

[5]Boden, D. G., & Larson, W. J., "Cost-effective Space Mission Operations". New York: McGraw-Hill, 1996, Ch. 5.

[6]Chung, S.E., "Timeline-based Mission Operations Architecture," SpaceOps 2012 Conference Proceedings, Stockholm, Sweden, June 11-15, 2012.

[7]Reinholtz, K., "Timeline as Unifying Concept for Spacecraft Operations," SpaceOps 2012 Conference Proceedings, Stockholm, Sweden, June 11-15, 2012.

[8]Jackson, M., Delp, C., Bindschadler, D., Sarrel, M., Wollaeger, R., Lam, D., "Dynamic gate product and artifact generation from system models," 2011 IEEE Aerospace Conference Proceedings, AERO 2011, May 13, 2011.

[9]Chien, S., Sherwood, R., Tran, D., Cichy, B., Rabideau, G., Castano, R., Davies, A., Mandl, D., Frye, S., Trout, B., Shulman, S., Boyer, D., "Using Autonomy Flight Software to Improve Science Return on Earth Observing One," Journal of Aerospace Computing, Information, & Communication, April 2005, AIAA.

[10]Chien, S., Tran, D., Rabideau, G., Schaffer, S., Mandl, D., Frye, S., "Timeline-Based Space Operations Scheduling with External Constraints," Proceedings International Conference on Automated Planning and Scheduling, Toronto, Canada, May 2010.