Jupiter Europa Orbiter Architecture Definition Process

Robert Rasmussen, Ph.D.
Jet Propulsion Laboratory,
California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109
robert.d.rasmussen@jpl.nasa.gov

Robert Shishko, Ph.D.
Jet Propulsion Laboratory,
California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109
robert.shishko@jpl.nasa.gov

Abstract[1]

The proposed Jupiter Europa Orbiter mission, planned for launch in 2020, is using a new architectural process and framework tool to drive its model-based systems engineering effort. The process focuses on getting the architecture right before writing requirements and developing a point design. A new architecture framework tool provides for the structured entry and retrieval of architecture artifacts based on an emerging architecture meta-model. This paper describes the relationships among these artifacts and how they are used in the systems engineering effort. Some early lessons learned are discussed.

Introduction

The proposed Jupiter Europa Orbiter (JEO) mission, planned for launch in 2020, is using a new architectural process and framework tool to drive its model-based systems engineering effort. The process focuses on getting the architecture right before writing requirements and developing a point design.

Europa is believed to have a saltwater ocean beneath a relatively thin and geodynamically active icy shell. Europa is unique among the large icy satellites because its ocean is in direct contact with its rocky mantle, where the conditions could be similar to those on Earth's biologically rich sea floor, powered by energy and nutrients that result from reactions between the sea water and rock. Consequently, Europa is a prime candidate in the search for habitable environments and life in the solar system. However, the details of the processes that shape Europa's ice shell and that control ocean-ice material exchange, are poorly known.

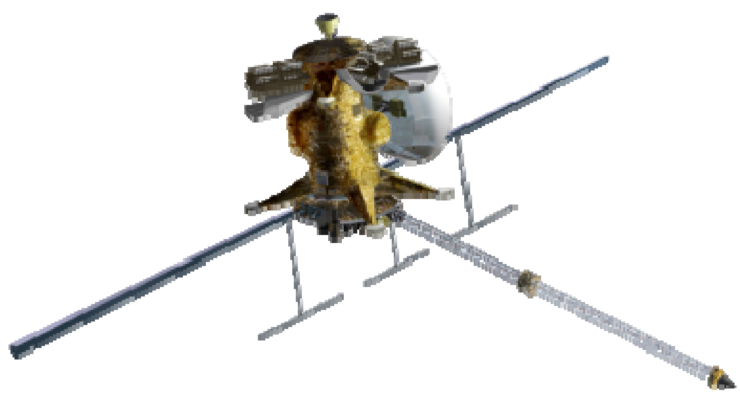

Figure 1. Jupiter Europa Orbiter conceptual spacecraft

In brief then, the JEO mission goal is to investigate Europa's habitability. To do this, the JEO spacecraft (see Figure 1) would conduct an orbital tour of the Jupiter system including close flybys of Io,

[1] This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Europa, Ganymede and Callisto before entering orbit around Europa. It would then carry out an intensive investigation of Europa. JEO would operate in a very low altitude orbit that would enable it to assess the interior of Europa electromagnetically, observe its tidal flexing, and map the surface at high resolution. A sounding radar would probe the ice to characterize its three-dimensional variability and the locations of shallow water. Mass spectroscopy, as well as thermal and hyper-spectral imaging, would be used to investigate the chemistry and search for sites of recent activity. The most promising sites would be potential targets for a future landed astrobiological mission to Europa.

> *"Architecture is the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution."*
>
> *----IEEE Std 1471-2000*

JEO would clearly be a complex major NASA science mission and the systems engineering for JEO is made more challenging by the partnerships that will be in effect. First, JEO is intended to be one of two spacecraft to be launched to the Jovian system at about the same time. The other spacecraft, called JGO (Jupiter Ganymede Orbiter) is a European Space Agency (ESA) mission designed for synergistic science with JEO. Second, JPL is partnering with the Johns Hopkins University Applied Physics Laboratory (APL), who will share systems engineering duties and would ultimately perform instrument integration.

In the JEO mission, architecting has been elevated to a more prominent and formal role than has been typical of other JPL projects. This paper describes the new architecting process being used to guide the systems engineering effort. The process requires systems engineers to work differently from the way previous JPL projects (even immensely successful ones) have been formulated and implemented. The obvious question, then, is why introduce a major change in the way JPL does systems engineering on major projects. We believe the answer lies in the need to make systems engineering's basic processes (requirements generation, trade studies, risk management, design and interface control, verification and validation, etc.) more coherent. Specifically, the new architecting process and framework is intended to aid systems engineering in the following ways:

Adding guiding structure — Traditional documentation, teaming, and review processes provide structure to systems engineering development and its documentation, but artifacts tend to be only loosely connected and many issues are conflated within catchall frameworks such as the requirements hierarchy. The goal on JEO is to augment this structure in a manner that prompts more direct consideration of key drivers and relationships.

Providing better integration of the resulting artifacts — Traditional approaches result in broad repetition of information across systems engineering artifacts, creating many opportunities for inconsistency. The goal on JEO is to have a single source of technical "truth," which helps to ensure consistency and accessibility of the information that drives important project decisions.

Ensuring comprehensive attention to important relationships — In traditional approaches, relationships among items are often implicit or underspecified. This hampers any assessment of completeness and correctness. The goal on JEO is to emphasize relationships in the structure of the architectural work in order to ensure that they are given due attention.

Facilitating broad understanding of the architecture — Traditionally, the only well-integrated treatment of a system (relatively speaking) has been in formal requirements, traced through system levels and assigned rationale, which are fragmentary descriptions, relating only with mixed success to architectural justifications. These have been accompanied by review presentations, design documents of several forms, and other artifacts from which a clear, consistent understanding of underlying concepts can be difficult to reconstruct. The goal on JEO is a consolidated, coherent description of the architecture that explains *why* it is the way it is.

Maintaining system integrity over the course of development — Systems that emerge from their constituents in traditional subsystem-focused developments often display characteristics or behaviors that their developers had not anticipated. Such surprises threaten the integrity of the system through delays, increased cost and risk, reduced performance, and an otherwise general lack of elegance that impugns the entire effort. Our goal on JEO is to thoroughly explore, define, and understand the system architecture from all points of view, attentively applying principles of good design, such that throughout subsequent design and implementation efforts the system view is never lost and the resulting system is true to its original conception.

Helping to ensure comprehensive V&V — Verification is focused on requirements, traditionally leaving validation as an ad hoc, catchall pursuit to ensure the integrity and suitability of the system for its mission. The system rediscovered through this process may be less a match to original ambitions than anticipated or desired, but the validation effort will be hampered in revealing such shortcomings if important aspects of the underlying concepts have not been well articulated or communicated. The goal on JEO is to carry architectural concepts throughout the development effort, regularly validating design and implementation against the architecture to ensure that system integrity and suitability are never misplaced.

The JEO Architecture Framework

Numerous architecture frameworks/standards have emerged over the past decade. These include:

- DoDAF
- MoDAF
- IEEE Std 1471-2000
- Kruchten 4+1
- OMG Model Driven Architecture
- RM-ODP
- RASDS
- TOGAF

The artifacts within each of these frameworks are heavily dependent on the nature of the systems and circumstances it was designed to support. For example, the DoDAF (and its predecessor, the C4ISR Architecture Framework) focuses on system-of-systems operations, interoperability, and operational connectivity within a net-centric environment. The JEO architecture framework, however, is perhaps most closely related to the ANSI/IEEE 1471-2000 standard for software-intensive systems. Figure 2 shows the artifacts of that standard expressed as a UML class diagram.

The JEO architecture definition process uses the JEO architecture framework, based on an emerging meta-model shown in Figure 3, as its guiding structure. Some of the architecture artifacts include, but are not limited to, Stakeholders, Concerns, Viewpoints, Views, Analyses, Models, Elements, Scenarios, Properties, and Functions. These artifacts deserve further definition, some description of the relationships among them, and how they are used in the architecting effort:

Stakeholders are influential “outside” people who are affected by or are accountable for the project’s outcome or its actions. Stakeholders have varying degrees of influence over the project architecture, and their sources of influence and authority may be legal, financial, technical, or political. Some stakeholders with specific designated authority or responsibility regarding the project (e.g., the NASA Environmental Management Division) generally represent the interests of a much larger class of individuals (e.g., citizens of Earth). In this example, the stakeholder is a proxy for and acts under a legal authority to represent such constituents. In the JEO architecture definition process, a stakeholder must be someone the architect can actually talk with, not just the title of an official, group, or organization. A part of the architect's job is to help each stakeholder express his/her concerns so that the project can address them in a comprehensive engagement plan.[2]

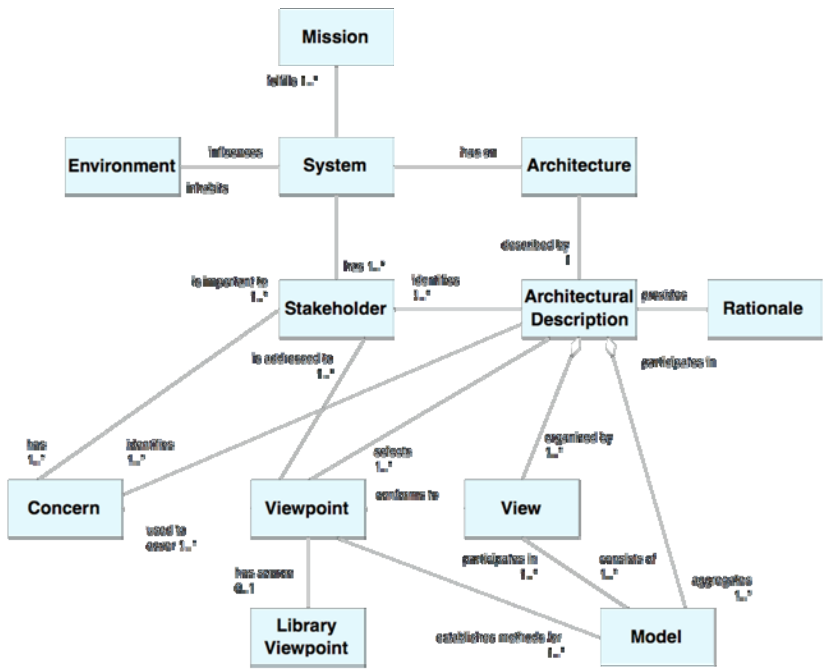

Figure 2. The ANSI/IEEE 1471-2000 conceptual framework expressed as a UML class diagram

Concerns are matters that demand attention in the architecture. Concerns generally arise from stakeholders, but some may be self-generated by the architect. In the JEO architecture process, each concern is accompanied by a stated success criterion (or criteria), which objectively measures the relative success in addressing the concern. The success criterion is intended to be expressed quantitatively, where feasible, as the value of one or more properties of the architecture (or its elements) when evaluated under pertinent scenarios. In those cases, one may think of the success criterion as identifying a measure of effectiveness and the quantitative value it must achieve, as required by the "Define Stakeholder Expectations" activity in NPR 7123.1. The success criterion is usually negotiated with the stakeholders early in the formulation process. To demonstrate achievement in meeting the success criterion during later formulation

[2] Project engagement of stakeholders is the first required activity identified in NPR 7123.1, where it is called "Define Stakeholder Expectations."

and implementation, the JEO architecture definition process develops particular viewpoints and views to show to stakeholders as evidence that a particular concern has been addressed.

Viewpoints are a set of rules, conventions, methods, and possibly templates for constructing views. They basically spell out what data (properties) will be presented, how the data will be calculated (in analyses using various models, if computation is needed), and what presentation format(s) will be used to best foster stakeholder understanding. In the JEO architecture definition process, viewpoints may also be negotiated with stakeholders so they know the form of the views they will be offered later.

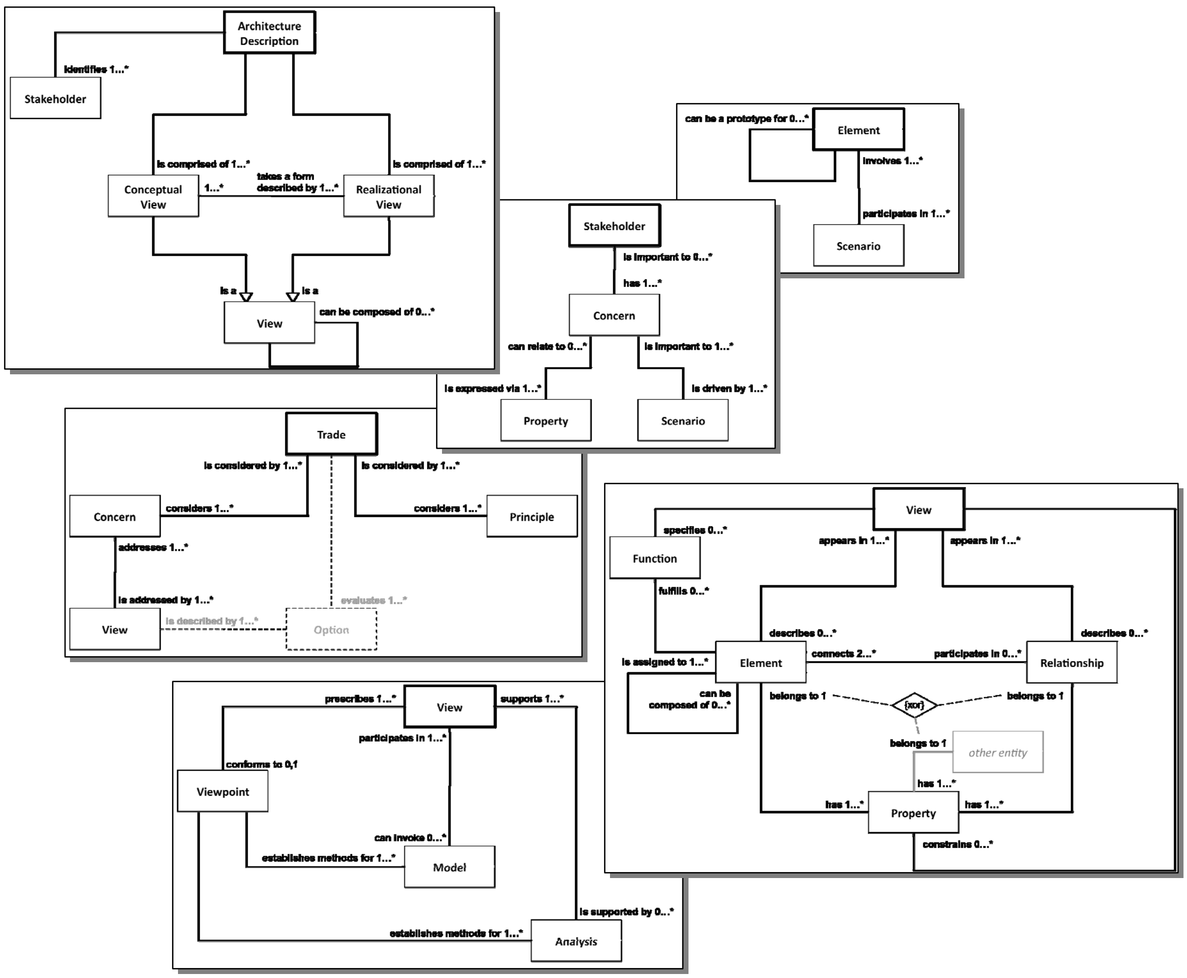

Figure 3. JEO Architecture Framework Evolving Metamodel

Views contain the actual data to be offered as evidence that a particular concern has been addressed. Each view conforms to one exactly one viewpoint. A hierarchy of sub-views is also supported in the framework. Views (and viewpoints) will naturally have diverse content. Typical views will cover decompositions of the architecture into elements along with properties of those elements (e.g., Mass Equipment Lists and functional allocations); models of behavior (e.g., Fault Modes and Effects Analyses); configurations (e.g., block diagrams); and programmatics (e.g., schedules). Many views of architectural significance will involve the results of analyses.

Analyses are calculations, simulations, or other examinations that are performed under the conditions of driving scenarios using validated models. Typically, an analysis is performed to assess the value of a property. Each evaluated property belongs to some architectural element described in a scenario and/or model.

Models are abstract representations of selected architectural elements (or collections of related elements). A valid model describes the essential characteristics of a chosen architectural element sufficiently well enough to be used for analysis or simulation of the element, or where appropriate, as an exemplar for its construction and integration. Models can take a wide variety of forms ranging from hard technical details in equations to general information captured in diagrams and prose. Models are the intellectual connective tissue of architecting that provides the link between the concerns stakeholders have on the one hand, and what developers will implement and operators will use on the other. In the JEO architecture definition process, requirements will spring from these models and the properties of the concerns whose assessment they enable. Rationale for these requirements will spring from the trades enabled by these models and their ties backs to concerns. A substantial part of the architecting effort is in the modeling that supports it.[3]

Elements are the entities that arise in the decomposition of the architecture and that realize their function (or purpose) through their relationships with other elements. Elements are diverse and include such entities as spacecraft subsystems and components, planets, trajectories, operational teams, flight rules, etc. In the JEO architecture definition process, they may be either abstract entities or concrete entities (designated as 'conceptual' or 'realizational,' respectively), depending on the nature of the model or view in which they appear, the level of refinement available, and the specificity required to characterize the model in which the element resides.

As with views, elements can be defined hierarchically. No dominant hierarchy is imposed on either element or view hierarchies in order to maintain a good separation of concerns. It is the convergence of conceptual entities in their assigned realizational instantiations that accomplishes the required mapping to (i.e., requirements on) concrete product and work breakdowns.

Scenarios are an organized set of activities and events, and the associated conditions that govern their progress. Scenarios can be pertinent to certain mission phases or activities, or can be triggered by events or the presence of certain conditions, or can be subject to negotiated intervals of time, resource limits, or other constraints.

Properties are attributes, qualities, or characteristics of an architectural entity (including quality attributes ('ilities'), measures of effectiveness, measures of performance, etc.). A property may belong to an element, a relationship, a scenario, and other architectural entities (with some restrictions). Properties include at least those defined through inheritance from one or more class affiliations. That is, all members of a given type will be describable through a common set of properties, in addition to those that are unique to themselves. These classes may be hierarchically defined. For example, all spacecraft hardware elements possess the property 'mass', and all powered hardware elements also possess the property 'peak power', but some on-board instruments may uniquely possess the property 'vertical depth resolution.' In the architecture definition process, emphasis is on properties that are of interest, either directly to a

[3] The approach to modeling in the JEO project is described in the JEO Model Management Plan.

stakeholder, or indirectly, as a necessary part of the description of a concept or realization that supports an analysis.

Functions are the purposes for which an element is intended. This can be an activity performed by the element or the usage of this element by other parts of the system.

Making It Work

The role of architecting within the larger systems engineering effort is described in the JEO Systems Engineering Management Plan (SEMP). The system architecting effort is an overarching aspect of the systems engineering effort on JEO. Consequently, it should not be viewed as a separate systems engineering task, but rather as a method and timeline for accomplishing systems engineering tasks. Further, architecting in JEO is neither just a top-level effort, nor is it exclusively a systems engineering effort. Rather it will be threaded through all engineering levels, as necessary to capture the concerns of JEO stakeholders, and discipline experts will be engaged to ensure sensible treatment of all issues.

The role of architecting also changes as a project progresses. Its early role is to establish the perimeter of a design space that addresses stakeholder concerns in a reasonably balanced manner and within which subsequent design development can take place. Establishing a conceptual architecture (grounded in realizations, as necessary) for the mission and system, and establishing a set of formal requirements for each of the major elements defined in the architecture are the primary architecting tasks through the end of Phase A.[4] As we have stated, rationales for these requirements will be drawn from the architecture definition.

To make all this happen, the JEO architecture framework must contain the copious amounts of information associated with the critical architecture artifacts described above, plus additional artifacts not mentioned such as the results of trade studies. All of this information must be reliable and made available whenever needed. The JEO Architecture Framework Tool (AFT), developed in 2010 as an independent effort within JPL's Integrated Model-Centric Engineering (IMCE) initiative, supports the structured entry, storage, and retrieval of this information. *The AFT provides the means by which JEO will move away from document-based systems engineering to a more modern approach based on information technology.*

As the guiding structure, the architecture framework specifies how artifacts are to be categorized and related, but not which artifacts are primary and should be produced first. It can certainly be said that working on one artifact prompts consideration of another. Major artifacts, however, need a more thoughtful top-down approach. Moreover, it is important to understand the broad outline and timing needs of architecting products before devoting resources to their detailed composition.

Consequently, development of the JEO architecture artifacts will be staged in regular intervals, called iterations. These define the major cadence of development, with time reserved in each iteration for reassessment of priorities, scope, and assignments. Iterations will typically range from three to four months between starts, with adjustments made to align with major project milestones. Some overlap between iterations is expected, as planning and preparation for the next begins while another is wrapping up. Within each iteration, the architecting effort will be divided into modestly sized tasks, called increments, each producing an extension or refinement of the architecture that

---

[4] In the NASA project cycle, Phase A culminates in a System Requirements Review (SRR) and Mission Definition Review (MDR).

can be effectively assigned and tracked through the systems engineering workflow management process. Increments will likely vary in size and scope, but should average a few workweeks.

Architecting products are subject to informal peer review in each increment as part of the closeout of each task. Reviewers will be drawn from the project's system engineering team, as appropriate, and from subsystem/instrument and discipline engineers working on the project. In addition, outside peer reviewers will be drawn in at the discretion of line and project managers. Steadfastly, the architecting team will not consider an architecting task complete in the increment workflow until the peer-reviewed results have been captured in the AFT.

As we have already mentioned, the development and use of models will be another critical enabler of the architecture definition process. Because of the intense radiation flux around Jupiter, the JEO spacecraft would encounter within a few months following Europa orbit insertion a life-limiting radiation dose comparable to the total dose received by the *Galileo* spacecraft over the course of its extended mission. Consequently, one of the early efforts in JEO is the development of a system radiation model (an architecture artifact in itself) to produce a stochastic estimate of the spacecraft lifetime. This model will be a significant improvement over previous radiation models and will be ultimately updated with data from the *Juno* mission to be launched in August of this year. Another early model in development is the science traceability model, which will be the basis for future trades involving the instrument suite and its relationship to the spacecraft system and mission.

Early Lessons Learned

During the first several iterations beginning in 2010 up to the present, the JEO architecting team focused on five major objectives: (1) identifying and capturing stakeholders and their concerns; (2) developing the content for and capturing viewpoints and views related to the concerns; (3) identifying and initiating trades that are needed in the near-term; (4) maturing the models that are needed to support those trades; and (5) AFT training for the growing architecting team. What was learned from these iterations is discussed next.

Initially, all architecting team members were asked to generate potential stakeholders. This resulted in a great deal of duplication, unevenness, and mis-specification of the "true" stakeholder. The Chief Architect stepped in to scrub the stakeholder entries down to a complete, but manageable, set of about 20. From that pared-down list, individual architecting team members were assigned, based on their expertise, to build stakeholder descriptions, concerns, and potential viewpoints (and views, where possible) for a small number of stakeholders. This was done to deepen the documentation of stakeholders and their concerns. Some stakeholders' concerns could be determined from statutory requirements, but for others, an interview was conducted with an informed proxy. This enabled the architecting team members to cask a somewhat wider net in gathering concerns. The Chief Architect again performed a consistency scrub of the stakeholder descriptions, concerns, and engagement plans, so that by the end of the second iteration, a fairly stable data set was ready for peer-review. The effort also resulted in some 'gold standard' examples that could be used in training newer members of the architecting team. The lessons learned here were that it is neither obvious who the real stakeholders are, nor is it easy to describe their concerns accurately and completely. Both iteration and discipline are needed.

In order to capture an initial set of viewpoints and views (including supporting scenarios, analyses, elements, relationships, models, properties, and so on) from pre-cursor Europa mission studies, the architecting team had to deconstruct a large volume of work and understand the underlying logic and rationales for those studies' conclusions. This effort proved to be resource- and time-intensive due to some incompletely documented assumptions, analyses, models, and so on. The lesson learned here is that capturing data from previous studies would have been significantly easier if they had been more architecturally minded and formally organized. Had that been so, assumptions, analyses, models, and so on would already have been captured and their linkage to stakeholders and concerns would have been clear.

The early identification of the project's core models, their maturation schedule, and those responsible for delivering them is also needed. The architecture for the integration of selected models needs to be pulled together as well. The lesson learned is that a Model Management Plan as a subordinate document to the SEMP should be created.

Another set of lessons learned in early iterations concerned training. While some formal training in the use of the AFT was conducted, it became clear that the architecting team also needed an informal venue for the practical issues that unfolded in using it. Consequently, a regularly scheduled meeting was established during which the architecture team members could air these issues and get immediate feedback from other members and the Chief Architect.

Summary

JPL is pioneering an information- and model-based approach to system architecting as a way of advancing the practice of system engineering. The currently perceived benefits from this approach relate to the deeper understanding of architectural choices that it will provide, i.e., why the system is the way it is. Specifically, it is hoped that the new approach will allow the project to consistently address both technical and programmatic (e.g., cost) issues, will allow a smoother transition from formulation activities to implementation, and will strengthen the explicit consideration of downstream concerns (i.e., operability).

While some aspects of defining the approach were supported by previous internal research efforts, the architecting process outlined in this paper is no longer a research project; it's actually being used by the JEO project with strong management "buy-in." There will be no shadow process within which the "real" work gets done. However, this new architecting process is presumed neither to be flawless nor immutable, so the intent is to correct the approach as shortcomings are discovered. In other words, its application on the JEO project will be a "learn-by-doing" experience.

The evolving JEO Architecture Framework Tool (AFT) is being initially populated by the architecting team. Successive versions will be augmented and refined with the aim of convergence to a complete and internally consistent formulation of the JEO architecture. Throughout, the AFT is intended to be the "single source of truth." As appropriate and feasible, documentation products required of the systems engineering effort will be generated from the contents of this data repository, so for example, major review material will be assembled from the text, diagrams, and model results already in the AFT.

Early iterations of the architecting effort have produced a considerable amount of data, but several more will be needed to achieve the critical mass of information needed to support a successful Mission Concept Review (MCR), the key gate for entering Phase A. Beyond that, the architecture definition itself must begin to stabilize in preparation for the impending design work of Phase B. Key to realizing this stabilization will be a collective confidence that nothing important has been overlooked among all the various concerns, viewpoints, analyses, models, and the many relationships among them.

References

ANSI/IEEE Standard 1471-2000, *Recommended Practice for Architectural Description of Software-Intensive Systems*

Department of Defense, *DoDAF Architecture Framework Version 1.5, Volume I: Definitions and Guidelines*, Washington, DC, 23 April 2007

Maier, M.W., and E. Rechtin, *The Art of System Architecting*, 2nd Edition, CRC Press, Boca Raton, FL, 2000

National Aeronautics and Space Administration, NPR 7123.1, *Systems Engineering Processes and Requirements*, Washington, DC, 2007

Biographies

Dr. Robert Rasmussen is a Fellow of the Jet Propulsion Laboratory. He joined JPL in 1975 after receiving his Ph.D. in Electrical Engineering from Iowa State University. Since then, he has contributed to several JPL missions, including Voyager, Galileo, and Cassini, with significant roles in spacecraft system design, guidance and control, avionics and computer systems, test and flight operations, and automation and autonomy. In addition to his flight project experience, he has conducted research in fault-tolerant multicomputers; he helped initiate and advised the Remote Agent autonomy experiment on Deep Space 1; he has been Chief Architect of the Mission Data System project, which developed a methodology and associated architecture for unified, model-based systems and software engineering; he has been Division Technologist for JPL's Information Technologies and Software Systems Division and Chief Engineer of JPL's Systems and Software Division; and he is presently Chief Architect of the JEO pre-project.

Dr. Robert Shishko is a Principal Systems Engineer and Economist at the Jet Propulsion Laboratory (JPL), California Institute of Technology. Dr. Shishko received two S.B. degrees from M.I.T., and his M.Phil. and Ph.D. in economics from Yale University. He has been at JPL since 1983, and has worked on the ISS and Constellation programs and Mars Pathfinder project. He is presently part of the JEO architecting team. Dr. Shishko was the principal contributor to the *NASA Systems Engineering Handbook*, SP-6105. He has also contributed to several books in the Space Technology Series, including the most recent *Applied Space Systems Engineering*. He is currently co-editing an AIAA book, entitled *Space Logistics, Supply Chain Management, and Operations: Opening New Frontiers*.