Orion Flight Test 1 Architecture – Observed Benefits of a Model Based Engineering Approach

Kimberly A. Simpson, Oleg V. Sindiy, and Thomas I. McVittie
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109

U.S. Government work not protected by U.S. copyright

Abstract—**This paper details how a NASA-led team is using a model-based systems engineering approach to capture, analyze and communicate the end-to-end information system architecture supporting the first unmanned orbital flight of the Orion Multi-Purpose Crew Exploration Vehicle. Along with a brief overview of the approach and its products, the paper focuses on the observed program-level benefits, challenges, and lessons learned; all of which may be applied to improve system engineering tasks for characteristically similarly challenges.**

TABLE OF CONTENTS

1. INTRODUCTION 1
2. BACKGROUND AND SCOPE 1
3. SYSTEM ENGINEERING USING A MODEL FOR ARCHITECTURE DESCRIPTION 2
4. BENEFITS 4
5. CHALLENGES 4
6. LESSONS LEARNED 5
7. SUMMARY 5
ACKNOWLEDGMENT 5
REFERENCES 6
BIOGRAPHIES 6

1. INTRODUCTION

Exploration Flight Test 1 (EFT-1) (formerly Orion Flight Test 1/OFT-1) is an unmanned orbital test of the Orion Multi Purpose Crew Vehicle (MPCV). EFT-1's primary objective is the timely and cost-effective testing of the spacecraft launch, orbital, re-entry, and recovery systems. Supporting this test requires an intricate ground system to configure the vehicle prior to launch, monitor mission progress, manage flight test sensor data, and issue contingency commands to Orion in support of potential anomalous mission events. To minimize non-flight costs, the capabilities of a number of NASA, Lockheed Martin provided systems are being reused and integrated into the core ground system. Understanding how these components are integrated to provide the needed capabilities, and how they will interact with MPCV to achieve the flight test objectives, requires the development of an end-to-end information system (EEIS) architecture.

This paper details how the NASA-led EFT-1 system engineering team transitioned from a largely paper-based

approach to one using formal models to capture and analyze key aspects of the EEIS architecture. Further, this paper articulates how the application of formal model-based engineering techniques has enabled the effective communication of the scope and design of the EFT-1 EEIS's system-of-systems architecture to a variety of the project's stakeholders. The paper concludes with a summary of the observed benefits, challenges, and lessons learned from the common capture and model-based repository of the integrated EFT-1 data architecture and its evolution.

Specifically, this paper highlights observed benefits to project and engineering communities in adopting the model- based system engineering approach in support of program decision making activities. Key observed project-level benefits include: support for requirements gap analysis, enforcement of system engineering rigor in representing the intended design, thorough configuration management of the architectural representation, model reusability, and report generation. Key observed benefits within the systems engineering community, include the ability to propagate systematic changes throughout the modeled architecture, thus transferring time and efforts away from "presentation and document-based engineering" back to integrated design centric techniques. Examples of the required investment into and encountered successes from the EFT-1 architecture model are also discussed, including how the approach may be re-used to support a variety of future program-to-program and project architecture development tasks.

2. BACKGROUND AND SCOPE

Orion MPCV Program and Mission Overview

Orion is designed as an exploration vehicle for carrying crew to space, providing emergency launch abort capabilities, sustaining the crew during space travel, and providing re- entry from deep space return velocities [1]. The Orion MPCV Program is charged with delivering a spacecraft capable of [1]: serving as the primary crew vehicle for mission beyond low Earth orbit (BEO)—such as, crewed missions to asteroids and Mars—and conducting in-space operations—e.g., rendezvous, extra-vehicular activities—in conjunction with payloads delivered by the Space Launch System (SLS) for missions BEO.

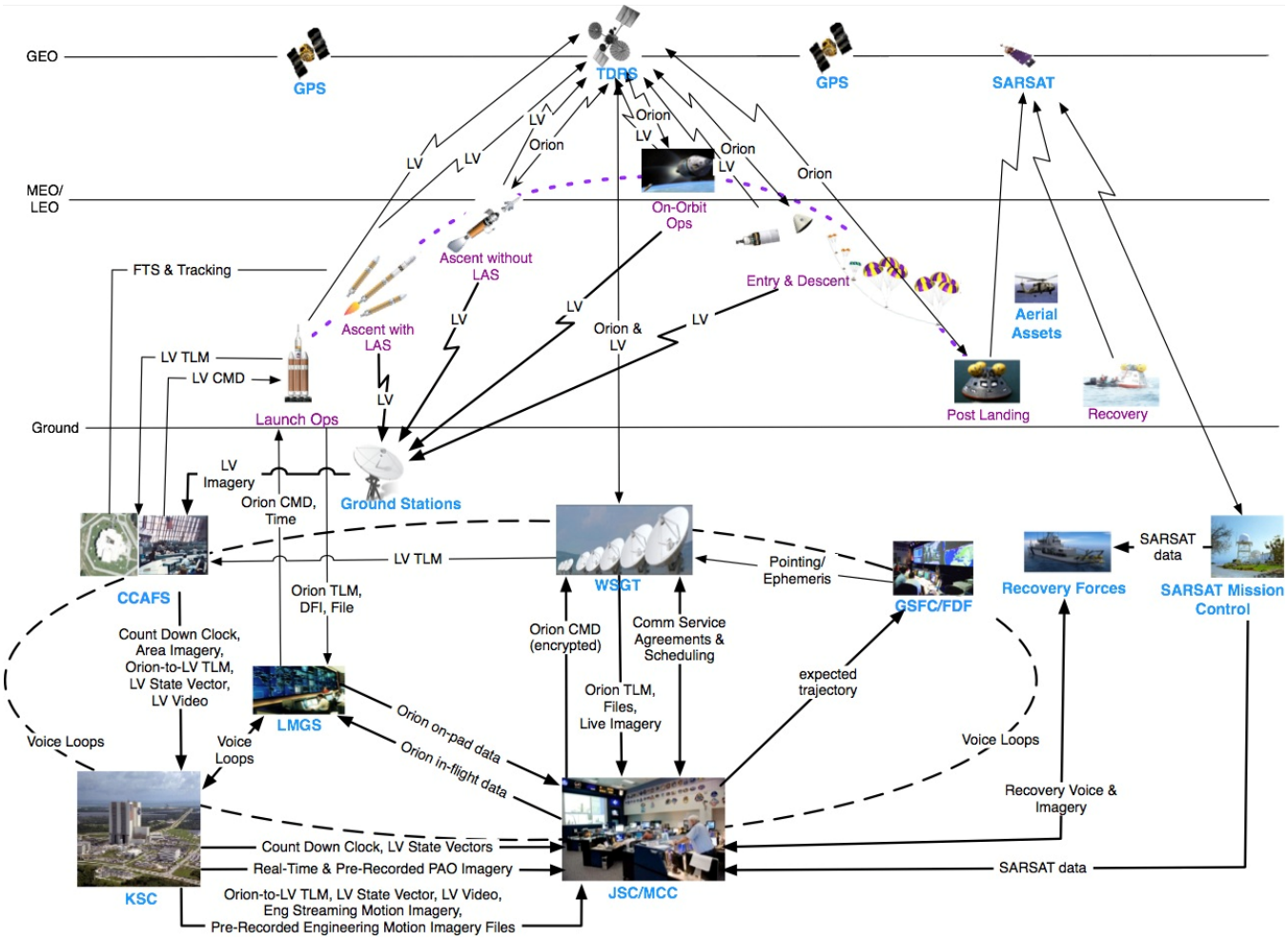

Figure 1 – EFT-1 High-Level Mission Overview of the End-to-End Information System

EFT-1 is a planned debut of the Orion spacecraft aboard an Expendable Launch Vehicle (ELV) on a multi-hour orbital flight test. It is tentatively scheduled to launch aboard a Delta IV-Heavy Launch Vehicle in early 2014. The test will be a jointly operated between Lockheed Martin Space Systems and NASA Mission Operations Directorate (MOD) with support from multiple commercial and government entities. The mission duration is expected to be approximately 5 hours with a trajectory that delivers two low Earth orbits, followed by Hi-Apogee, Hi-Energy re-entry. A splashdown in the ocean, under a parachute system, is expected to be roughly 500 nautical miles off of the Baja Peninsula, with capsule recovery via a Navy well-deck ship.

Scope of the System Engineering Task in Support of the Design of the End-to-End Information System

In late 2010, a response to a high-level program Ground Data System risk identified the need to augment the LM's existing and proposed Ground Data System (GDS) capabilities with NASA provided capabilities needed to support EFT-1. In response, the scope of the MPCV Flight Test Management Office's (FTMO) system engineering activities were expanded to include the NASA portion of the ground data network and NASA-led portions of the information system, and definition of how they would integrate with the LM provided GDS capabilities. The scope of the task was defined to include the capture of EEIS for all of the test and integration configurations, and the operational mission phases (launch through post-mission analysis). For this task, the NASA team chose to employ a model-based system engineering approach with the intention of using the resulting models and products to assemble, represent, and effectively communicate the progression of the evolving EFT-1 EEIS design and development among many of the project's stakeholders. Figure 1 provides a pictorial high-level overview of the notional EFT-1 mission in the context of the EEIS architecture description.

3. System Engineering Using a Model for Architecture Description

System Engineering via a Modeling Approach

The system engineering task was initiated using a small core team with system engineering, GDS design, development, and system-of-systems architecture modeling experience. The selected approach leverages, and improves on, the previous NASA Constellation Program's work in model-based capture of the Computing System Architecture Description.

In implementation, the team's activities focused on: capturing the information provided by the distributed set of multi-organization stakeholders and system designers in a variety of documents, refining their understanding via continuous in person and electronic communications, and then using this information to populate the EFT-1 EEIS architecture model. The resulting model provides an integrated end-to-end architecture description which is then employed to generate multiple representations (i.e., views) that cater to the multitude of evolving concerns of the project's stakeholders. These products have enabled the team to communicate to NASA management and key project stakeholders: system level functionality, as per EFT-1 concept of operations; the end-to-end test, integration, and operational configurations; interface, functional, and design requirements; scope of the system; and assumptions, clarifications, unknowns to be resolved, and alternate implementation options under consideration.

The technical specifics of the applied approach consist of employing System Modeling Language (SysML) [2] plug-in in the MagicDraw modeling tool [3] to produce a conceptual model of the architecture description based on the taxonomy from IEEE 1471 [4] and extraction of the stakeholder driven views for the specific domain of the application problem.

Custom Views of the Modeled Architecture Description

According to the IEEE 1471 standard [4], an *architecture* is the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution. An *architecture description* provides a collection of products employed to document an architecture. Presentation of an architecture description is organized by a set of custom views. Each *view* serves as a representation of a whole system from the perspective of a related set of concerns. Each view conforms to a viewpoint, which is a specification of the conventions for constructing and using a view. A viewpoint provides a pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis. In brief, each view delivers a look through a specialized window into the system architecture which assists in answering some set of questions relevant to the key stakeholders and their concerns.

The set of current viewpoints and views selected to communicate the EEIS design for EFT-1 are illustrated in Figure 2 and are summarized next.

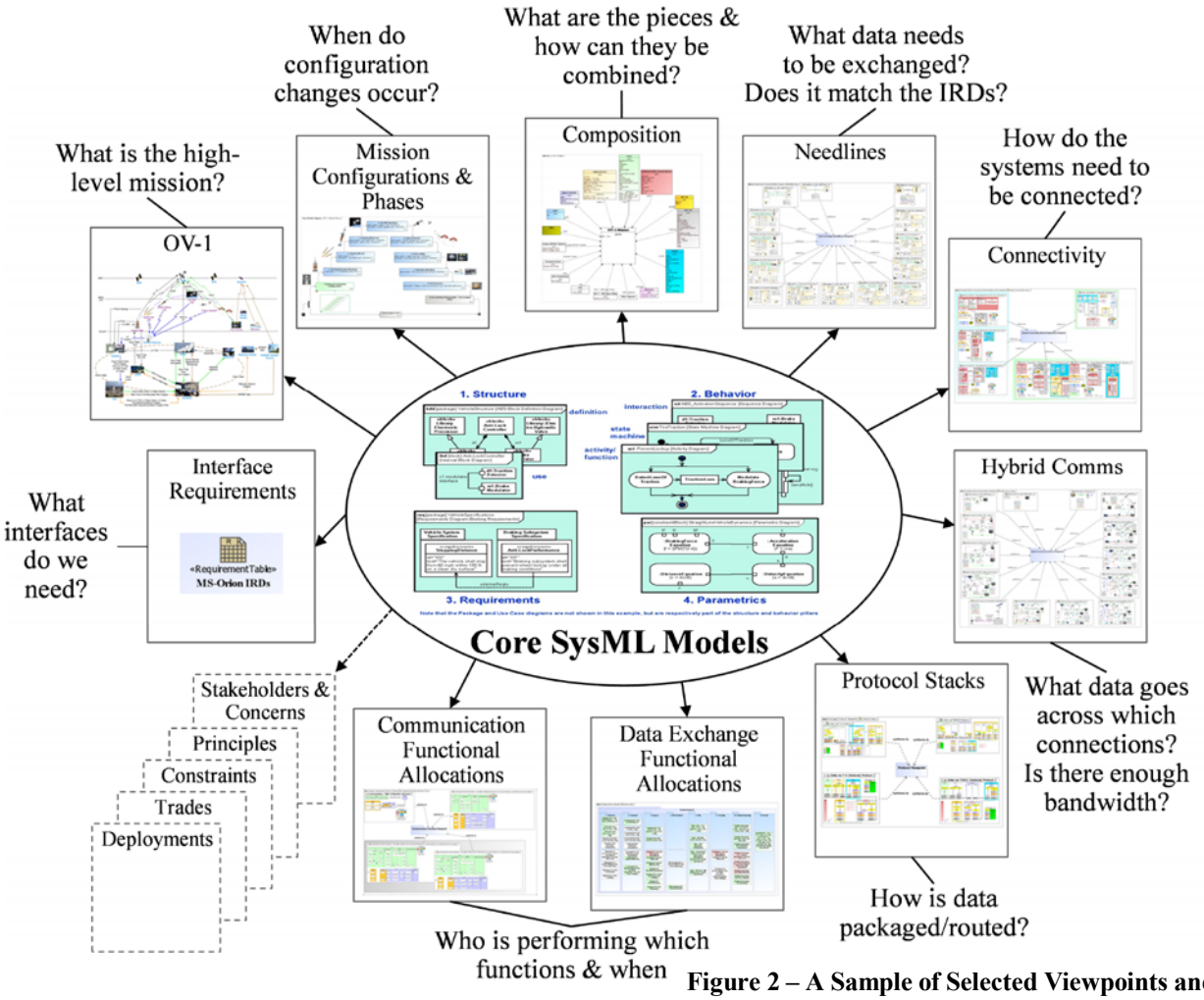

Figure 2 – A Sample of Selected Viewpoints and Views from the Model of the EFT-1 EEIS Architecture

High-level Mission Operations View: provides a basic, high-level overview diagram that defines the key nodes/systems of the mission and how they interact over time. This view is based on Operation View 1 (OV-1) defined in the DoDAF [5], which recommends the use of a high level operation concept graphic to offer a graphical and textual description of the operational concept for the architecture under consideration. Figure 1 is an example of this view.

Mission Configurations and Phases View: defines the major mission testing and integration configurations, operational mission phases, and the transitions between them. The configurations and phases represent some fundamental change in the architecture which alters one or more of the aspects of the architecture description required to describe system.

Composition Views: setups the decomposition of the system-of-systems by location, owners, facilities, systems, hardware, including how this system-of-systems evolves over time. The ability to associate various hardware and software builds with each of these configurations and phases provided an invaluable tool for reasoning about the System's capabilities during any given phase or even when specific integration tests could be performed.

Needlines Views: defines the major exchanges of data, as driven by mission Concept of Operations, Interface Requirement Documents (IRDs), and Mission System Requirements Document (MSRD) between the various organizational and physical nodes during each configuration or phase of the mission. This viewpoint is based on the DODAF OV-2 [5], where a needline view documents the required or actual exchanges of resources.

Requirements: documentation of the relevant requirements (e.g., IRDs, MSRD) that drive the design of the architecture and which may be used to validate the design captured in the architecture description.

Hybrid Communications Views: details the data flows from the needlines views between the origin and destination nodes, with the addition of the relay nodes and major communication network employed to get the data from the sources and the sinks. This viewpoint was created for this project to better communicate data presented in several related views; in particular, to better match the traditional diagrams the project stakeholders were accustomed to seeing.

Data Exchange Functional Allocation Views: identifies key operational functions associated with the processing, transmission, and distribution of data—e.g., create, combine, send, receive, process, store, etc.—involved in realizing each needline during every mission phase or configuration.

Connectivity Views: defines how major facilities and nodes are connected across the end-to-end network. The view shows layer 2-physical and layer 3-logical connections.

Protocol Stacks Views: details the major communications protocol stacks for each system involved in enabling the flow of needlines through the network. The viewpoint is based on the protocol stacks presented in RASDAS [6].

Communication Functional Allocation Views: details the functional decomposition of the communication services (functions) by core mission systems. The view shows the flow of data through the various communication functions of the nodes involved in selected mission configurations and phases.

Beyond the ones discussed, numerous other viewpoints and ensuing views are possible. These can be implemented as evolving concerns of the project stakeholders drive a need for them. For example, the EFT-1 system engineering team can integrate additional details into the architecture to produce views that capture detailed specifics designs in technical domains—e.g., system-of-system level trades, parametrics (e.g., data volume, latency), explicit mapping of stakeholders to their concerns and which views address their needs, verification and validation, and so forth.

4. BENEFITS

The key benefits observed in implementing the described approach for the EFT-1 end-to-end information system include:

- support to various levels and layers of design definition
- definition of cross-system interface development
- enabling system-level trade analysis responsive to program constraints facilitation of architectural alignment with flight test objectives and operations concepts enforcement of system engineering rigor
- implementation of thorough configuration management of the architectural design and its description among a distributed and multi-organizational set of stakeholders
- enabling of reusability for various future system configurations; both the approach and the architecture captured in the model for EFT-1 can be adopted for other proposed Orion MPCV program missions: Ascent Abort-2 (AA-2) and so on.
- providing an integrated design centric approach, that is aligned with vision outlined in NASA's Human Exploration and Operations Mission Directorate's (HEOMD) capability-driven approach framework

5. CHALLENGES

Some of the challenges observed are as follows:

Changing the culture: one of the fundamental challenges when working with a distributed set of stakeholders from multiple organizations is to create a vehicle and mechanism where instead of updating their own private representations

of the architecture (usually in Visio or PowerPoint), that they instead are able to provide periodic up-dates to the integrated architecture model, and then have the model generate the products/representations they need in a timely manner. The maximum benefit of employing a single integrated end-to-end description may only be realized if it serves as a single source of truth for capturing, representing, and communicating the description of the latest architecture design by the majority of the stakeholders involved.

Representation styles: it has been critical to our communication efforts that the representative views look like the products that the stakeholders are used to seeing, but with the added benefits of a model-based approach and use of standards. The modeling team's use of a SysML specification sets a project standard that other SysML practitioners are able to understand. Although, until it is embraced by the majority of the stakeholders involved, it will continue to be somewhat of a challenge to clearly communicate all aspects of the captured system description without iterating what the standard specific representation denotes.

Management of alternative options: especially for problems as complex as those characterized as system-of-systems, it is programmatically and technically challenging to capture and manage all of the alternatives under consideration by the various design teams; including how the various options propagate/influence the different architectural variants in the architecture description. However, investment into the model-based capture of a carefully pruned set of options can enable designers to understand the architecture-wide implications of various decisions, which informs them of emergent insights that could not otherwise be observed.

Tool limitations and stability: modeling tools to perform system-of-system architecture capture and representation have yet to achieve the kind of stability and breadth as other model-based engineering tools—e.g., ones in the computer-aided design and drafting, structural analysis, electronics, dynamics and controls domains. We found it to be imperative to establish a direct relationship with tool vendor(s) to fix bugs and add additional tool functionality, as well as allow for the internal project capability to "bend" the capabilities of the tool, to meet the immediate needs of the project.

6. LESSONS LEARNED

In implementing this approach, we also gathered a list of lessons learned to help us apply this type of an approach for comparable design challenges in the future. To-date, some of these lessons are:

Based on many historical usages, using terminology "Model-Based System Engineering (MBSE)" can confuse and introduce unwarranted bias toward the approach by some of the stakeholders. For EFT-1, we chose not to openly use this terminology, and instead emphasize our task as a "system engineering using a modeling approach." We found this to be more effective in gaining acceptance across the majority of the organizations and stakeholders involved.

Take the time to understand what you need to communicate with the allotted workforce and within the available task duration. This includes strong upfront investment in planning, so as to: (1) clearly understand stakeholder needs, and (2) determine how best to utilize the architecture description model in support of the specific system engineering needs and objectives.

The approach requires discipline experts using the modeling tool to perform their system engineering functions. It is not about just using a tool to draw elaborate diagrams, but rather producing elaborate diagrams from a common database for the entire architecture description.

It was found to be more effective to keep the core system engineering team, which employs the modeling approach, small (less than 5 persons) and tightly coupled. This allowed the team to control the capture of architectural/technical baseline, to provide a single source of information. In this process, all of the stakeholders can use the single source of information as a reference by having the modeling team periodically export updates views to facilitate communications amongst the larger program team.

Communicate…communicate…communicate. Use of the model as the vehicle for communicating system design to management, stakeholders, and the technical leads is essential for success in large scale, multi-stakeholder, and widely distributed programs.

Review…review…review. Reviews are critical to confirm that all stakeholders' needs are addressed, the end-to-end system, as designed, is sufficiently specified, and the design is correctly captured and represented.

7. SUMMARY

This paper presented a summary of the overarching benefits, challenges, and lessons learned in the process of applying a modeling approach for the system engineering of the end-to-end information system for Orion MPCV Program's Exploration Flight Test 1. Overall, the paper demonstrated how the application of formal model-based engineering techniques has been invaluable to effectively communicate the scope of, and perfect effective system engineering for, a system-of-systems architecture design among many of the project's stakeholders.

ACKNOWLEDGMENT

This task was managed out of the Jet Propulsion Laboratory, a division of the California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The work was funded and performed under the leadership, guidance, and support of the Orion MPCV Flight Test Management Office (FTMO) manager, Donald Reed of NASA's Johnson Space Center.

REFERENCES

[1] NASA, "About the Crew Vehicle and Launch System," 14 Septeber 2011, http://www.nasa.gov/exploration/systems/index.html [Accessed 11 October 2011]

[2] OMG Systems Modeling Language (SysML) Specification, Object Managemetn Group (OMG), Version 1.2, 1 June 2010.

[3] MagicDraw, Software Package, Version 17, No Magic Inc., Allen, TX, 2011.

[4] ISO/IEC Standard for Systems and Software Engineering - Recommended Practice for Architectural Description of Software-Intensive Systems, ISO/IEC 42010 IEEE Standard 1471-2000, 1st Ed., 15 July 2007.

[5] U.S. Department of Defense, "Department of Defense Architecture Framework (DoDAF)," Version 2.02, August 2010.

[6] Reference Architecture for Space Data Systems (RASDS) Recommended Practice, The Consultive Committee for Space Data Systems (CCSDS), Washington DC, USA, Issue 1, CCSDS 311.0-M-1, September 2008.

BIOGRAPHIES

Kimberly Simpson is an assistant program manager in the Mission Systems Concepts section at NASA's Jet Propulsion Laboratory and the Orion Multi-Purpose Crew Vehicle Program Network Integration lead. Over the past six years, Ms. Simpson has held a variety of leadership positions within Constellation Program and the current human spaceflight program responsible for program integration of end-to-end hardware and software data systems. She has also held many diverse roles within various technical sections and divisions performing system engineering, ground software development, and spacecraft integration and testing. She is the recipient of the Technical Award for Excellence, Explorer and Ranger Awards, and has received multiple NASA Certificates of Recognition for her support to JPL flight projects. Ms. Simpson earned a Bachelor of Science degree in Computer Science from California State Polytechnic University-Pomona.

Dr. Oleg Sindiy is a software systems architect in the Ground Systems Engineering section at NASA's Jet Propulsion Laboratory and the co-architect and modeler of the end-to-end information architecture description for the Orion Multi-Purpose Crew Vehicle Program. He is currently also supporting JPL's Ground System process modeling work and the trade space analyses for NASA's modernization of the Space Communications and Navigation Network (SCaN) assets. Oleg Sindiy is a recent PhD graduate from the School of Aeronautical and Astronautical Engineering at Purdue University, where his dissertation research focused on the "Model-Based System-of-Systems Engineering for Space-Based Command, Control, Communication, and Information Architecture Design." He earned a Master of Science degree in Aeronautical and Astronautical Engineering from Purdue University and a Bachelors of Science degree in Aerospace Engineering from Embry-Riddle Aeronautical University- Prescott.

Dr. Thomas McVittie is a principal software systems engineer at NASA's Jet Propulsion Laboratory where he specializes in the engineering, design and implementation of secure and reliable command and control systems. He currently supports the end-to-end information systems design for the upcoming Orion test flights and the design of resilient and secure power grids for the Los Angeles Department of Water and Power. Previously, he was the Chief Architect for the Constellation Program's Software and Avionics Integration Office and numerous Department of Defense projects. Dr. McVittie earned a PhD in Electrical and Computer Engineering, and a Master of Science in Computer Architecture from University of California-Santa Barbara.