The Evolvable Advanced MultiMission Operations System (AMMOS): Making Systems Interoperable

Adans Y. Ko, Pierre F. Maldague, Tung Bui, Doris Lam, and John McKinney
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

SpaceOps 2010 AIAA Conference, NASA Marshall Center, Huntsville, Alabama, United States of America

This research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with National Aeronautics and Space Administration (NASA)

Introduction

The Advanced MultiMission Operations System (AMMOS) has succeeded in keeping up with the changing functionality required by NASA missions:

- science observatory
- small body impact
- surface explorer
- sample return
- surface rover / lander relay

Here, we focus on a study of AMMOS modernization, and on the benefits of adopting new emergent IT standards, making AMMOS composable, network accessible, and interoperable between NASA centers.

Legacy System Crisis (1 of 3)

- Typical legacy system experience:
  - Evolves under a number of organizational as well as technical constraints
  - Evolution is limited to maintenance, sustaining, and small incremental enhancements
  - Grows in size over time, on the average, by the factor of two to three every decade:
    - becomes brittle and increasingly complex, resulting in its deteriorated structure,
    - becomes less maintainable and sustainable, and
    - Crucially dependent on a small pool of qualified personnel

Legacy System Crisis (2 of 3)

- Lack of up-to-date documentation:
  - Incremental enhancements are seldom documented with the same care as the initial delivery, due to smaller size of maintenance team and piecemeal nature of funding
  - Interaction with domain experts is the best option to perform "as-is" system analysis on legacy tools

- Accumulated design uncertainties:
  - Use of programming scripts as a quick fix method for integrating monolithic applications that were not designed to work together
  - Resolving legacy system's design uncertainties is impossible or difficult

Legacy System Crisis (3 of 3)

- Sporadic use of open source technology:
  - Decisions to use open source are made by individual developers, with little attention paid to software engineering issues across legacy system
  - Open source infusion into legacy system has been accidental and heroic

Legacy System Modernization

- Adopt Risk-Managed Modernization (RMM) methodology for AMMOS legacy tools modernization:
  1. Identify key stakeholders
  2. Understand requirements
  3. Create business cases
  4. Understand the legacy system
  5. Define the target architecture

Identify Key Stakeholders

- AMMOS S/W System Engineers, S/W Architects, and S/W developers:
  - make decisions on design trades, analyze new functional requirements, and promote consistency of AMMOS interfaces within subsystems and with external systems

- Project MOS/GDS system engineers:
  - provide AMMOS with project-based concepts and requirements that reflect the needs of Mission Operations Systems (MOS)

- MGSS Program Managers, and MGSS Program Element Managers:
  - establish funding guidelines, “make buy” decisions, or resourcing decisions

Business Drivers

- The purpose of Multimission Ground System and Services (MGSS) is to develop AMMOS so as to offer NASA missions the following advantages and benefits:
  - Reduce the overall cost to NASA:
    - Project does not have to pay for the development of the AMMOS core capability
  - Reduce the average development time for individual projects:
    - Project adaptation of AMMOS takes less time than development
  - Reduce mission risks with more stable and mature software:
    - Most AMMOS elements have been maintained and improved over a number of years and have been used by a variety of NASA projects in a variety of situations. Many bugs have been discovered and resolved

Requirements

- Examples of MGSS AMMOS modernization requirements:
  - Incorporate technological advances into the AMMOS to maintain reliability and compatibility with future mission flight and ground systems
  - Streamline the Mission Operations System (MOS) Uplink Process from activity planning through execution of commands on board the spacecraft by automation and generalized process improvements
  - Provide end-to-end data accountability
  - Provide tools and services for complex planning missions
  - Provide a fully automated flight system monitoring service

Business Cases

- Examples of mission operations business cases:
  - Develop, optimize and analyze S/C trajectories to achieve mission goals
  - Create conflict-free mission and science activity plans, command sequences to achieve flight project's objectives
  - Process S/C engineering telemetry data and display of actual vs. predicted S/C channel values
  - Capture, distribute, reconcile, catalog, and archive mission files
  - Assess the health and performance of the S/C
  - Deploy and maintain mission Ground Data System during development, Assembly Test Launch Operations (ATLO), and flight operations

Understand Legacy System (1 of 5)

- Adopt the horse-shoe model to reconstruct logical descriptions of the AMMOS Mission Planning and Sequencing (MPS) tools:
  - Interview domain experts and stakeholders
  - Identify existing S/W artifacts
  - Use model-based system engineering to capture logical descriptions of MPS legacy tools in UML/ SysML

Understand Legacy System (2 of 5)

AMMOS Legacy Tools pain point analysis: Interviewed stakeholders, domain experts, and use of mission generated documents of the "AMMOS lessons learned" variety.

Quality Attributes	Pain Points
Affordability†	Tool expenditures for duplicate functionality development. Configuration is labor intensive.
Adaptability†	Much manual configuration. Adaptation requires in depth knowledge.
Modifiability & Extensibility†	Software changes overly impact the system Too many interfaces Hard to add or delete applications
Interoperability†	Different models needed for different subsystems, 50% overlap in information. Too many vocabularies to have to know Information definitions are hard to reuse
Deployability†	Monolithic nature of the applications precludes combining them into integrated GUI clients resulting in awkward workarounds and inefficient Ops procedures. Multiple deployments of the same AMMOS application and automated operational scripts, each accessed via a different endpoint.
Security††	No centralized credential database with consistent interfaces. No clear process for integrating new people. No process for de-provisioning accounts. No formal policy or process for managing transitions between access signatories. Lack of support and control over the procurement and management of browser security certificates (X.509).

Examples of AMMOS Legacy Tools Pain Point

Understand Legacy System (3 of 5)

MPS S/W tool identifications found in the MGSS CM database

ID	**Mission Planning Tools**	ID	**Sequence Virtual Machine Simulation Tools**
APGEN	Activity Plan Generator	OLVM2	Off-Line VM Engine,Type2
SOA	Science Opportunity Analyzer		**MPS Utilities**
	DSN Scheduling Tools	SEQREVIEW	Sequence Review Tool
CAST	Common Allocation Scheduling Tools	SEQADAPT	Sequence Adapter
	Sequence Generation Tools		VML-to-SATF Converter
SEQGEN	Sequence Generator	RSFOS	Re-Engineered Space Flight Operations Schedules Software
	Command Translation Tools	CTSCOM	CTS Component
SLINC2	Spacecraft Language Interpreter and Collector 2	PAPS	Persistent Apcore Server Software
CTS	Command Translation Subsystem	MPS Editor	MPS Editor
VMLCOMP2	VML Compiler, Type2	ULSGEN	Uplink Summary Generator

Subset of MPS S/W Tools

Understand Legacy System (4 of 5)

MPS tool S/W Interface Specifications (SIS):

- Collected from MPS and mission generated documents
- Interviewed domain experts and project users
- Reconstruct MPS tool component relationships (SIS producer and consumer)

SIS/Product	Producers	Consumers	SIS/Product	Producers	Consumers
APCORE XMLRPC	In-situ Planning	Activity Plan Generator	Sequence file	RSVP-ROSE	Sequence Generator
APGEN Adaptation File	Activity Plan Generator	Activity Plan Generator	Offline Virtual Machine Log file	Off-Line VM Engine, Type2	User
APGEN Plan File	Activity Plan Generator	Activity Plan Generator	PAP API	Activity Plan Generator	Persistent APcore Server Software
Command Packet File	Spacecraft Language Interpreter and Collector 2	DSN acmd_tcwrap Tool	Predicted Events File	Sequence Generator	Re-Engineered Space Flight Operations Schedules Software
Command Translation File (cmdxlt)	Command Translation Subsystem	Spacecraft Language Interpreter and Collector 2			MPS Editor
Conditions File	Sequence Generator	Sequence Generator	Rover Markup Language	In-situ Planning	MPS Editor
Flight Rules Model File	Sequence Adapter	Sequence Generator	SEQGEN User Defined Function	User	Sequence Generator
Multimission Activity Dictionary Language	In-situ Planning	MPS Editor	Master File	Common Allocation Scheduling Tools	Common Allocation Scheduling Tools

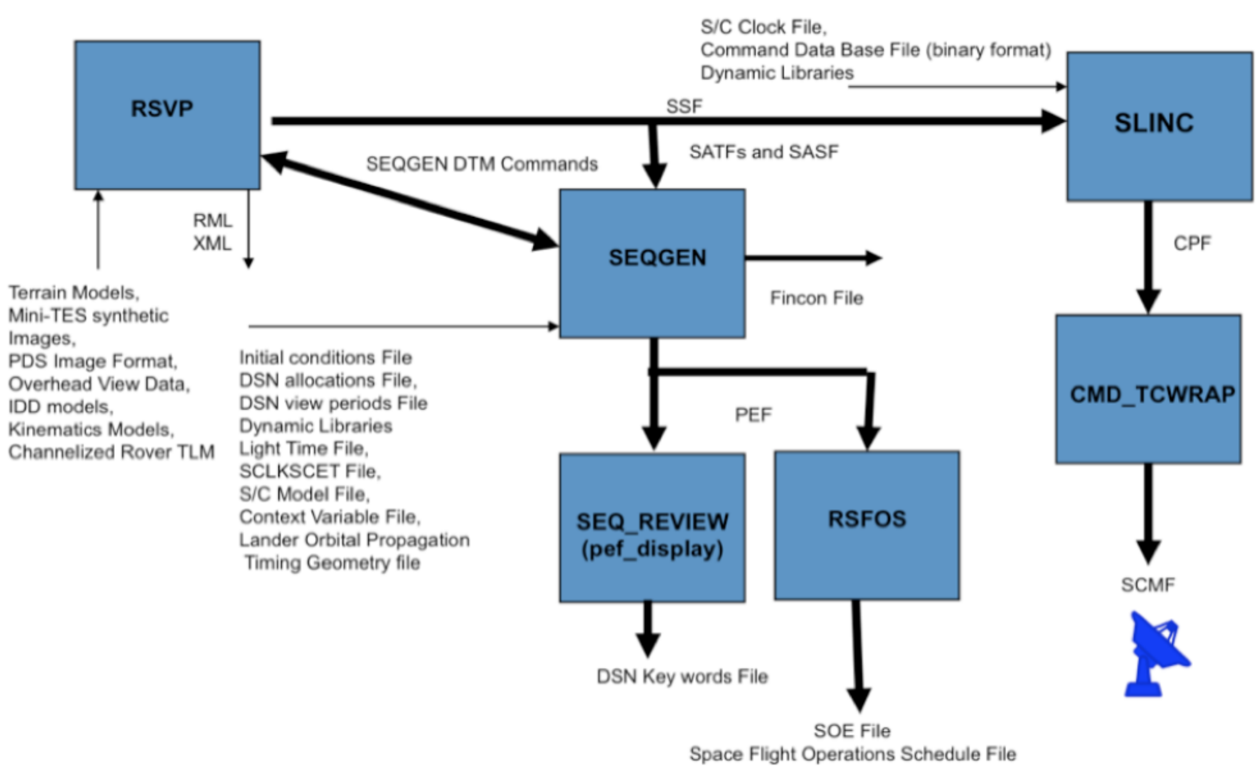
Subset of MPS S/W Interface Specifications

Understand Legacy System (5 of 5)

AMMOS-based Mars Explorer Rover Sequence and Command Generation System is an example of "pipe and filter" architecture.

- best for highly automated and predictable system
- inefficient for event driven and exception handling
- majority are file based interfaces

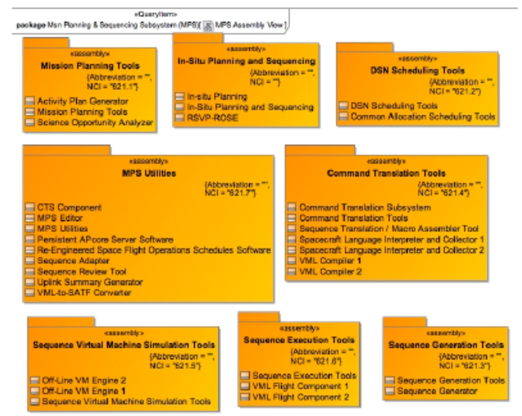
Sequence Generation and Validation Data Flow Format



Model-based Software System Engineering (1 of 4)

ID	Mission Planning Tools	ID	Sequence Virtual Machine Simulation Tools
APGEN	Activity Plan Generator	OLVM2	Off-Line VM Engine,Type2
SOA	Science Opportunity Analyzer		**MPS Utilities**
	DSN Scheduling Tools	SEQREVIEW	Sequence Review Tool
CAST	Common Allocation Scheduling Tools	SEQADAPT	Sequence Adapter
	Sequence Generation Tools		VML-to-SATF Converter
SEQGEN	Sequence Generator	RSFOS	Re-Engineered Space Flight Operations Schedules Software
	Command Translation Tools	CTSCOM	CTS Component
SLINC2	Spacecraft Language Interpreter and Collector 2	PAPS	Persistent Apcore Server Software
CTS	Command Translation Subsystem	MPS Editor	MPS Editor
VMLCOMP2	VML Compiler, Type2	ULSGEN	Uplink Summary Generator

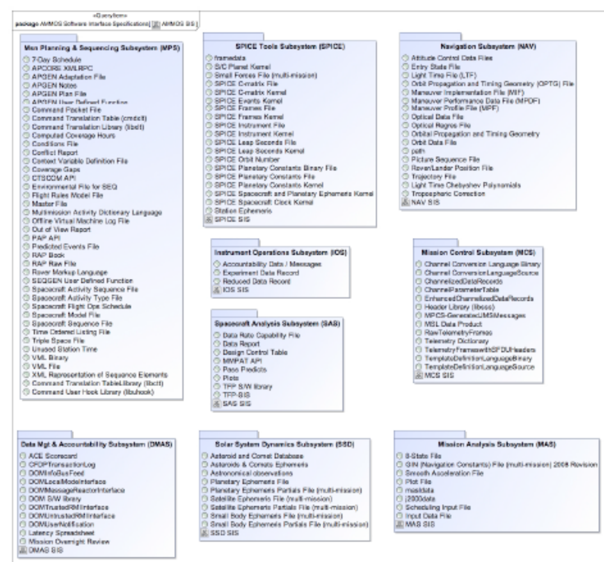
Subset of MPS S/W Tools



MPS S/W Components View

SIS/Product	Producers	Consumers	SIS/Product	Producers	Consumers
APCORE XMLRPC	In-situ Planning	Activity Plan Generator	Sequence file	RSVP-ROSE	Sequence Generator
APGEN Adaptation File	Activity Plan Generator	Activity Plan Generator	Offline Virtual Machine Log file	Off-Line VM Engine, Type2	User
APGEN Plan File	Activity Plan Generator	Activity Plan Generator	PAP API	Activity Plan Generator	Persistent APcore Server Software
Command Packet File	Spacecraft Language Interpreter and Collector 2	DSN acmd_tcwrap Tool	Predicted Events File	Sequence Generator	Re-Engineered Space Flight Operations Schedules Software
Command Translation File (cmdxlt)	Command Translation Subsystem	Spacecraft Language Interpreter and Collector 2			MPS Editor
Conditions File	Sequence Generator	Sequence Generator	Rover Markup Language	In-situ Planning	MPS Editor
Flight Rules Model File	Sequence Adapter	Sequence Generator	SEQGEN User Defined Function	User	Sequence Generator
Multimission Activity Dictionary Language	In-situ Planning	MPS Editor	Master File	Common Allocation Scheduling Tools	Common Allocation Scheduling Tools

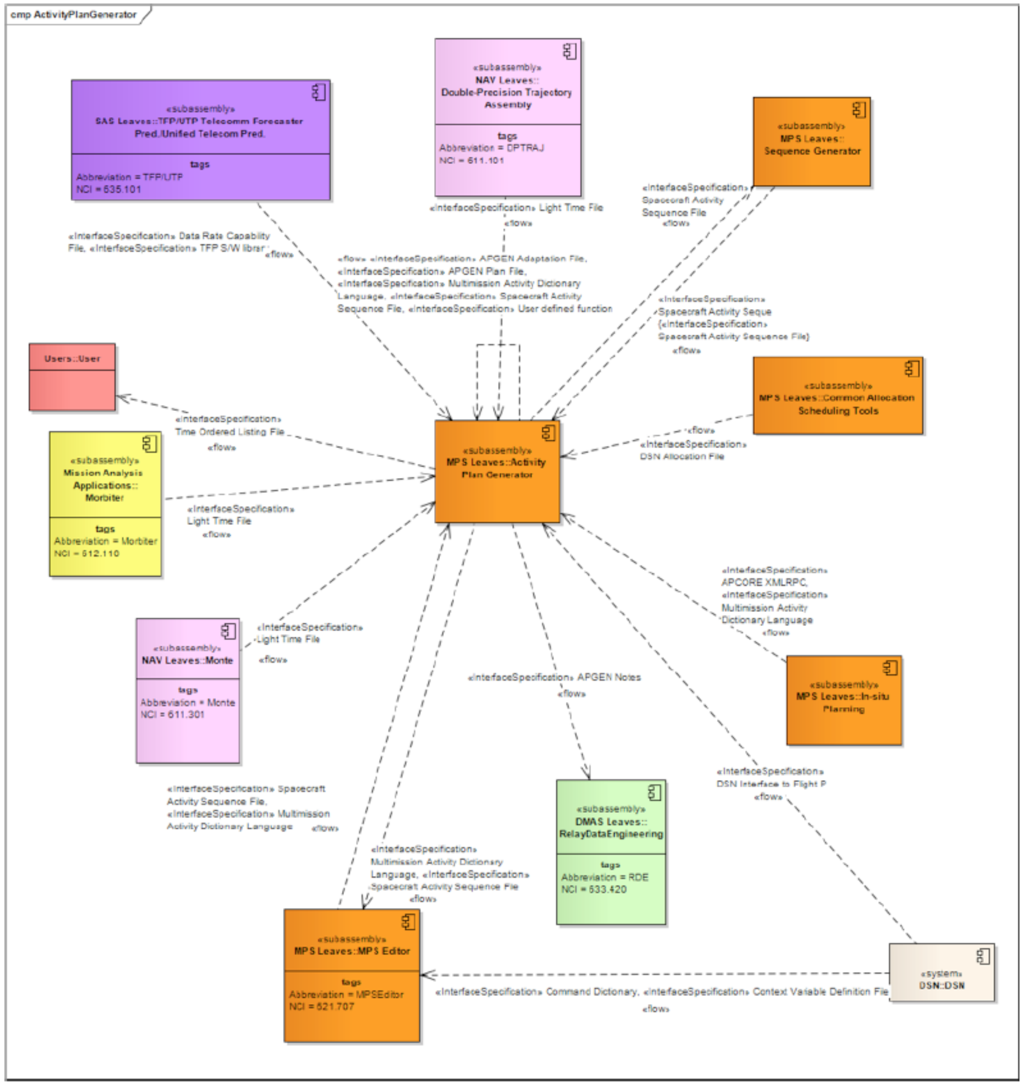
Subset of MPS S/W Interface Specifications



MPS S/W Interface Specifications View

Adans Ko, Pierre Maldague, Tung Bui, Doris Lam, and John McKinney

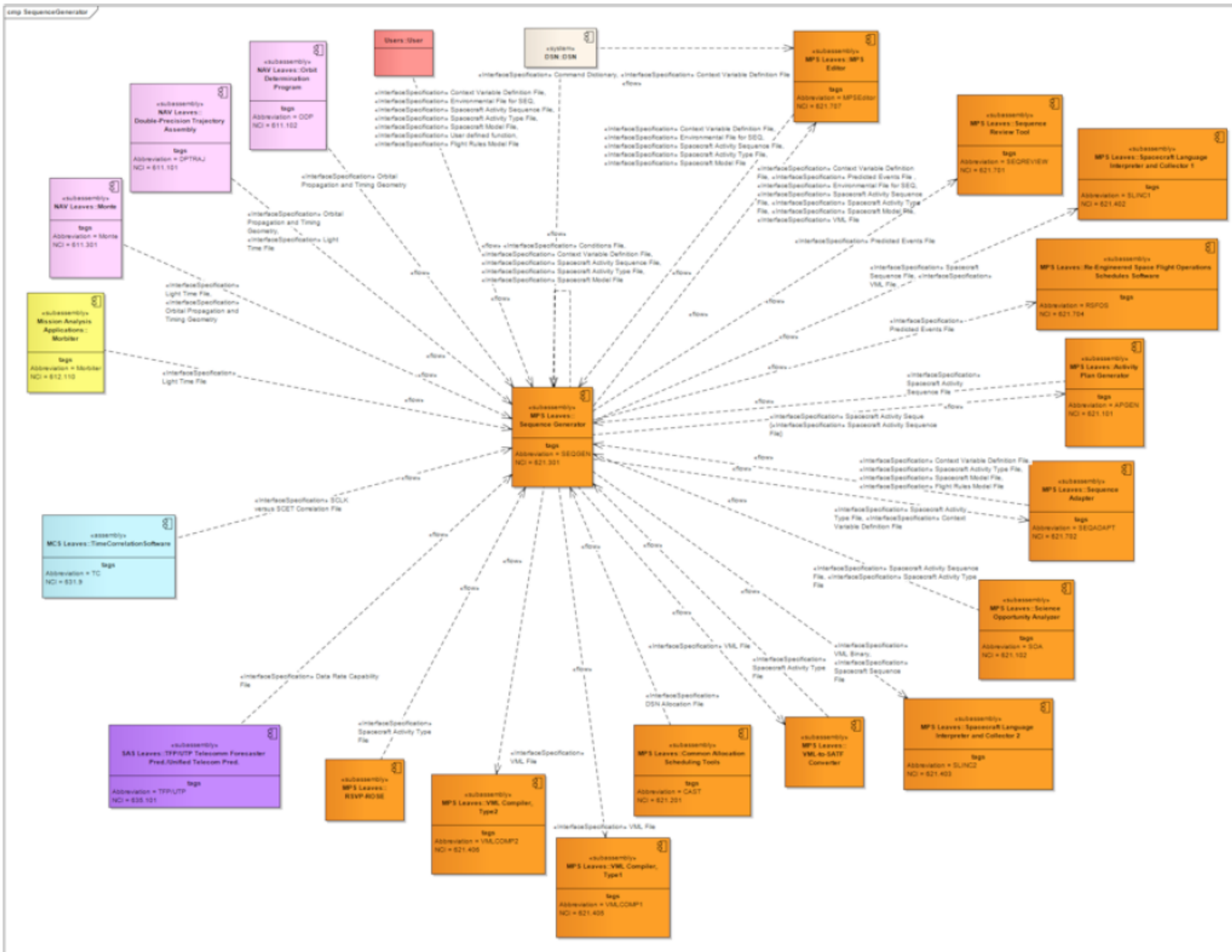
Model-based Software System Engineering (2 of 4)



MPS Planning S/W Components Relationships (internal and external)

Adans Ko, Pierre Maldague, Tung Bui, Doris Lam, and John McKinney

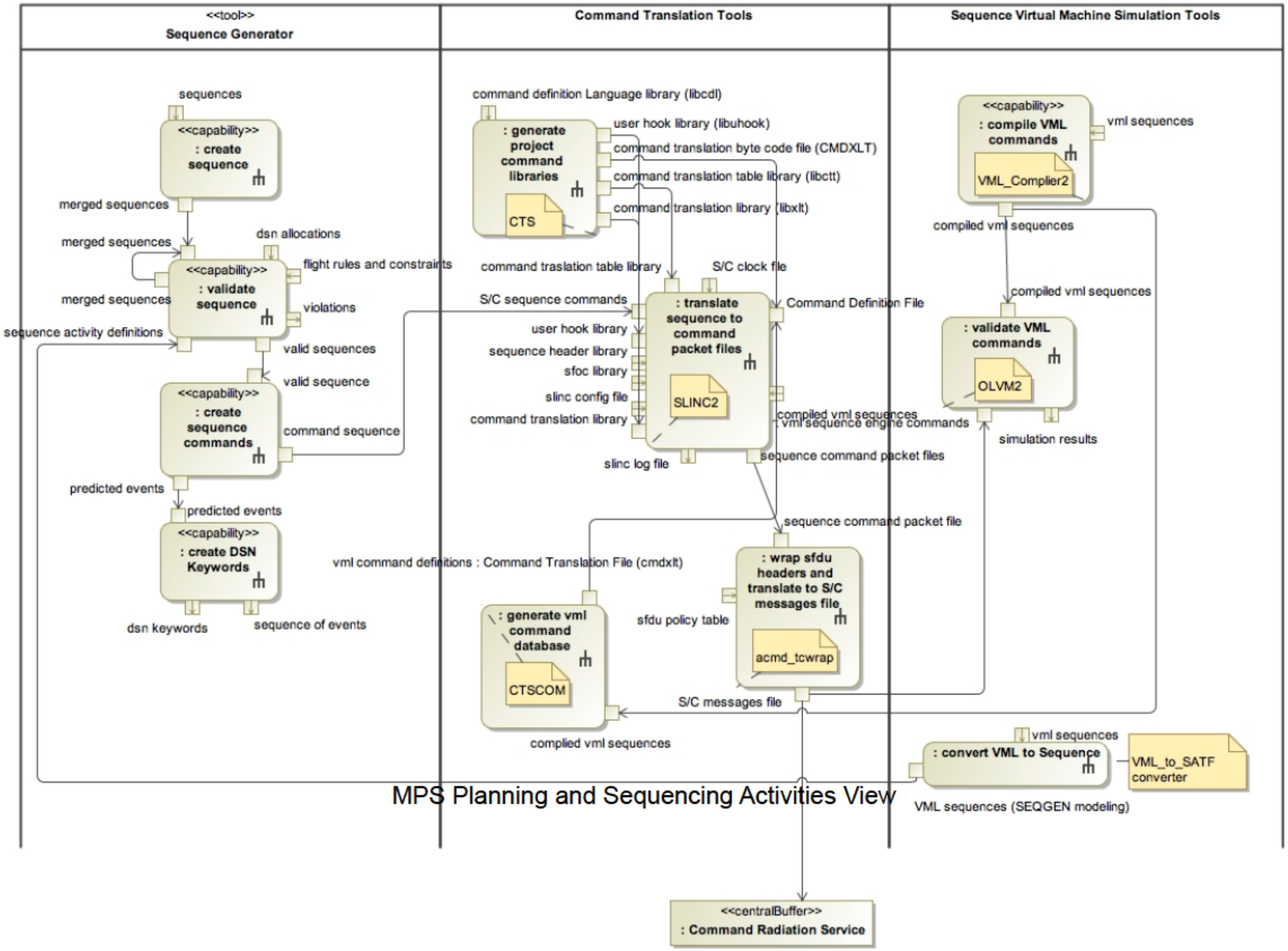
Model-based Software System Engineering (3 of 4)



MPS Sequencing S/W Components Relationships (internal and external)

Model-based Software System Engineering (4 of 4)

Create conflict-free mission and science activity plans, command sequences to achieve flight project's objectives



MPS Planning and Sequencing Activities View

Define Target Architecture (1 of 3)

AMMOS modernization is adopting a Service Oriented Architecture (SOA) based Deep Space Information Services Architecture (DISA). For examples, according to SOA core principles, components of a system should be:

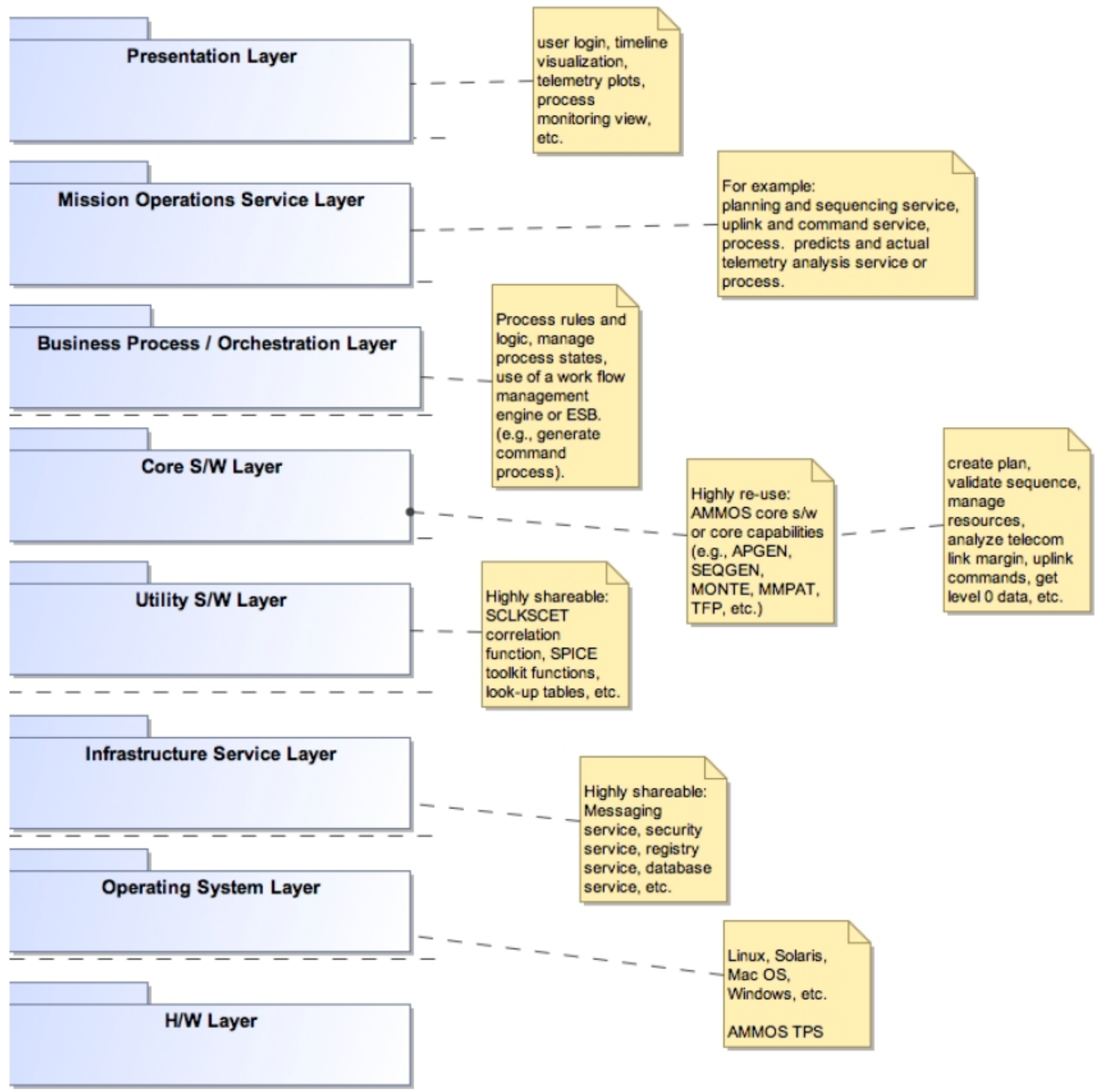
- Loosely coupled – message based and network accessible interfaces
- Composable – allow flexibility using alternative software components or services when necessary
- Standardized - use IT standards to deliver consistent services and enable reuse behavior
- Modular – reduction of dependency and component based software.

Define Target Architecture (2 of 3)

In this paper, an illustration of AMMOS MPS modernization is done using a SOA layered architecture concept:

- Presentation layer: MOS content specific presentation, user interface, navigation
- Service layer: services that consolidate major AMMOS capabilities as common services to flight project's MOS in a consistent manner. Strongly MOS business process and mission data oriented
- Business process layer: MOS business process orchestration. It manages automated workflow and process state for units of AMMOS services, that are required to be common across two or more MOS operation processes.
- Core software layer: software components that are domain specific to support AMMOS services.
- Utility software layer: software components or mission data that are highly shared by core software components.
- Shared infrastructure service layer: COTS / GOTS based infrastructure services such as messaging, discovery , storage, and security.

Define Target Architecture (3 of 3)



Layered Architecture Representation

System to System Interoperability (1 of 4)

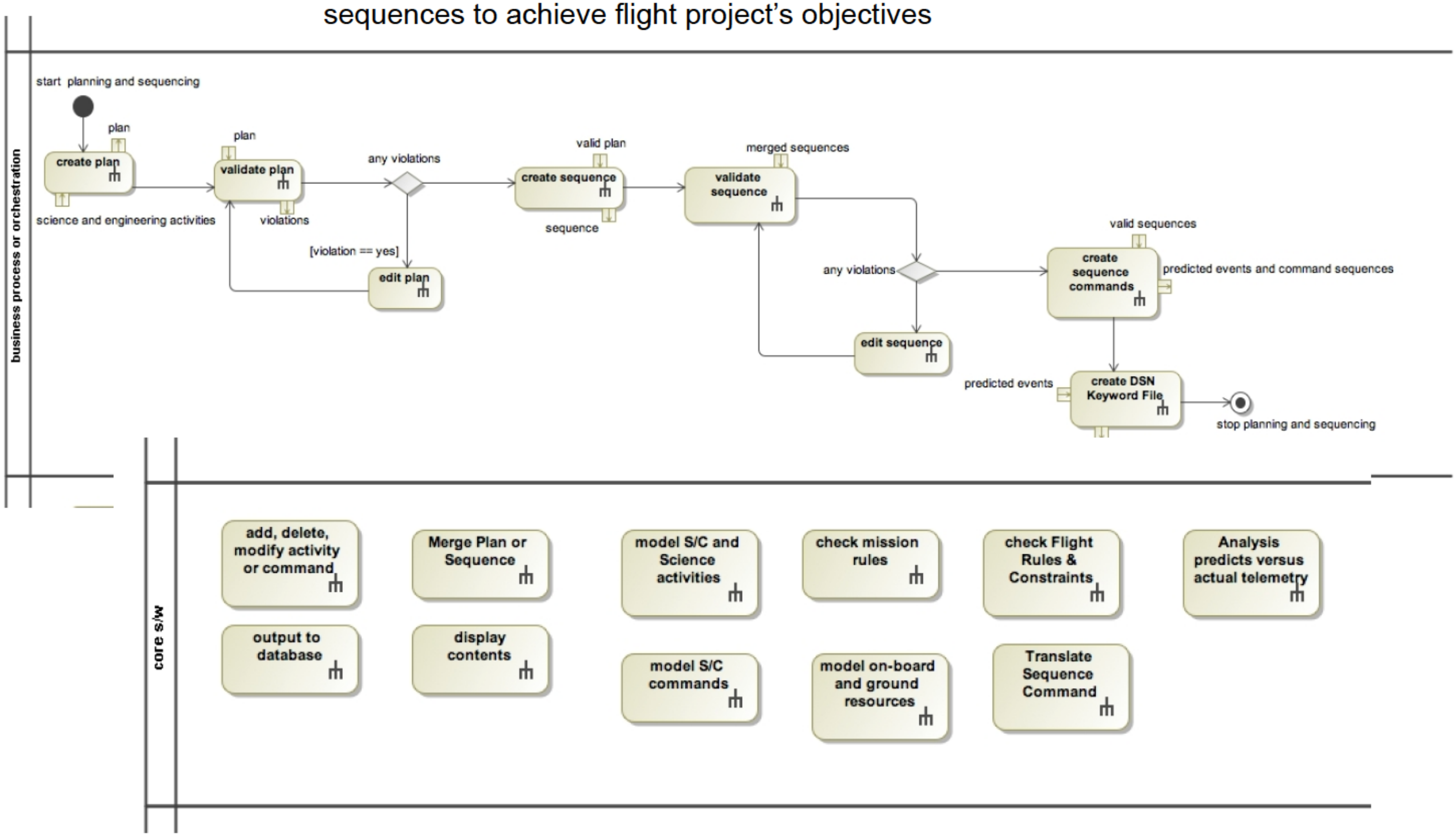
- Demo feasibility of system to system interoperability through legacy tool modernization:
  - refactoring monolithic applications into interoperable components
  - Business process oriented design
  - loose-coupled interfaces via shared messaging services

System to System Interoperability (2 of 4)

- Refactoring:
  - Remove a tightly coupled GUI and associated communication functions from a monolithic planning / sequencing application
  - Replace deleted elements with an industrial standardized communication design
  - Preserve well tested and tuned and encapsulate P&S expertise legacy modules

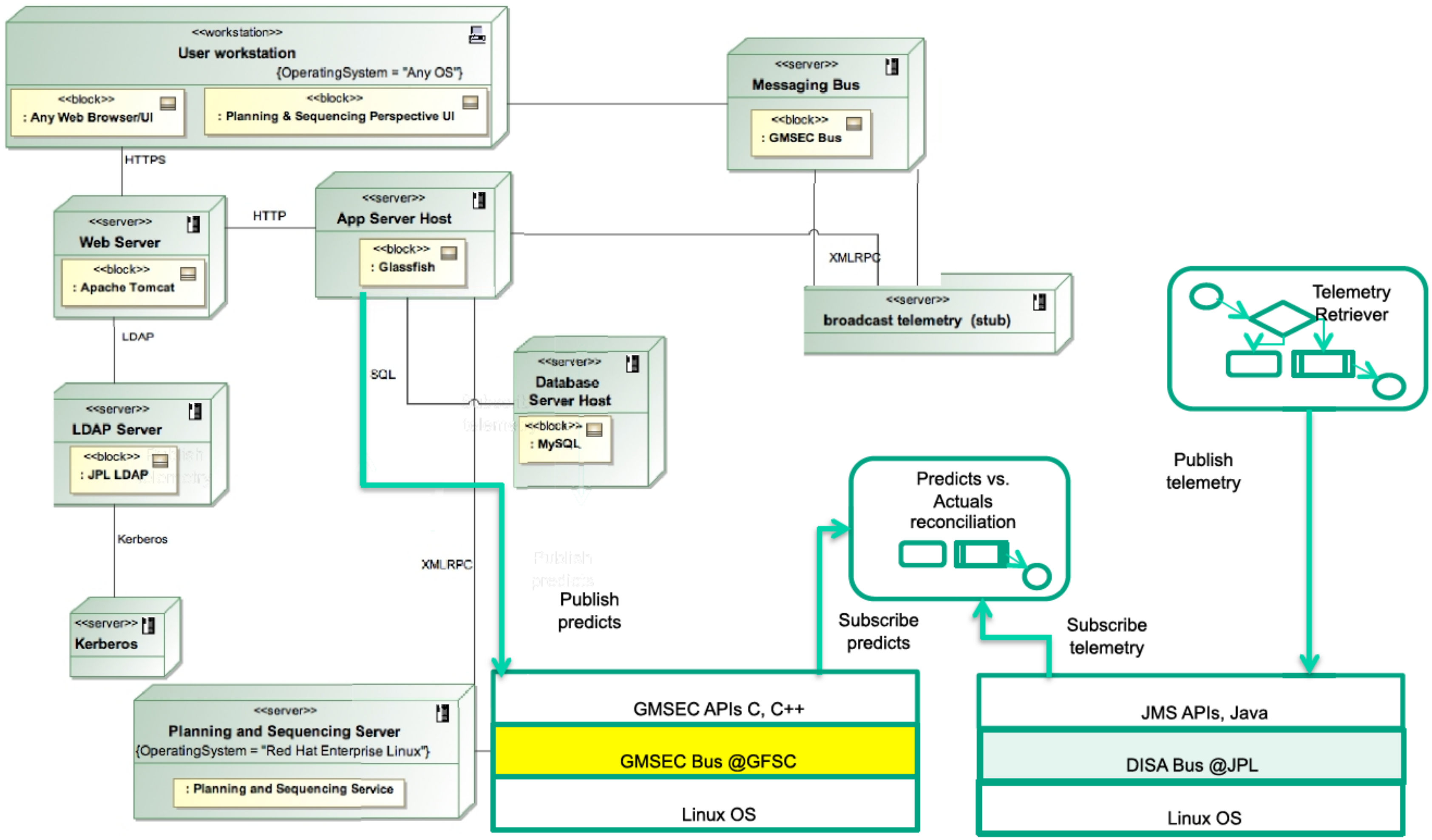
System to System Interoperability (3 of 4)

Business Process Oriented Create conflict-free activity plans and command sequences to achieve flight project's objectives



Adans Ko, Pierre Maldague, Tung Bui, Doris Lam, and John McKinney

System to System Interoperability (4 of 4)



Acknowledgements

We would like to acknowledge the many individuals who participated in the development of the AMMOS "as-is" architecture UML and SysML models and EPOXI mission operation scenario, their valuable insights and thoughts in the AMMOS system, Mission Operations System (MOS), architectural views standards, UML and SysML modeling strategy to support AMMOS model based system engineering practices. The individuals are too numerous to mention here for their contributions to this effort. Nonetheless, I wish to acknowledge the excellent work done on the AMMOS UML/SysML models and on the AMMOS “as-is” architecture concept prototype work by the following individuals, to whom I extend my special thanks: Steve Wissler (EPOXI S/C Team Chief and Flight Director), Mark Dutra, Kathy Rockwell, and Will Sun. Last but not least, I am grateful for the support of MGSS management, which special thanks to Dave Linick, Andy Dowen, and Ben Smith.

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

1. Seacord R.C., Plakosh, D., and Lewis, G. A., "*Modernizing Legacy Systems*", Addison Wesley, Boston, 2003, pp. 6-7.
2. Muller, H. A., Wong, K., and Tilley, S.R., "Understanding Software Systems Using Reverse Engineering Technology", *The 62nd Congress of L'Association Canadienne Francaise pour l'Avancement des Sciences Proceedings (ACFAS)*, 1994, pp. 41-48. Vol. 26. No. 4.
3. Lehman, M. M. and Belady, L., "Program Evolution: Processes of Software Change" *London: Academic Press*, 1985.
4. Seacord R.C., Plakosh, D., and Lewis, G. A., "*Modernizing Legacy Systems*", Addison Wesley, Boston, 2003, p. 28.
5. Maldague, P., Ko, A., Page, D., and Starbird, T., "APGEN: A multi-mission semi-automated planning tool", *First International NASA Workshop on Planning and Scheduling*, Oxnard, CA, 1998.
6. Seacord R.C., Plakosh, D., and Lewis, G. A., "*Modernizing Legacy Systems*", Addison Wesley, Boston, 2003, pp. 58-59.
7. Ko, A., Maldague, P., Page, D., Bixler, J, Lever, S, and Cheung, K.M., "Design and Architecture of Planning and Sequence System for Mars Exploration Rover (MER) Operations, AIAA SpaceOps Conference, Montreal, Canada, 2004.
8. McVittie, Thom, "DSMS Software Architecture Overview: Web-based GDS," Working Draft (Internal Document), Jet Propulsion Laboratory, California Institute of Technology, September 14, 2004.
9. Estefan, J. A., "The Importance of Architecture Governance for Achieving Operationally Responsive Ground Systems", *GSAW08 Architecture-Centric Evolution (ACE) Working Group Session*, April 2, 2008.
10. Sprott, D., Dodd, J., and Gilbert, M., "*Introducing Service Architecture and Engineering*", CDBI Journal, July/August 2006, p. 6.
11. Butler, J. and Ellis, D., "*Enhancing the Enterprise Architecture with Service Orientation*", CDBI Journal, October 2007.

Author Biographies

1. ***Mr. Adans Y. Ko*** *is a Software Engineer in Ground Software Architecture and System Engineering Group at JPL. Currently, he is the Multi-Mission Ground System and Services (MGSS) Software Architect for Advanced Multi-Mission Operation System (AMMOS), and he is responsible for Ground System software architecture for the current and future AMMOS. He was a system engineer and development manager for the AMMOS Mission Planning and Sequencing Subsystem (MPS) over 10 years. He has in-depth knowledge of the Multi-mission Ground Data System uplink tools, which include mission planning, sequence generation, and sequence flight software. In the private sector, he was a project manager for credit card systems for the Navy at CitiBank Development Center, Los Angeles, CA and a Principle Engineer for e-Commerce Consulting at MarchFirst Consulting Firm, in Los Angeles, CA. He received NASA "Turn a goal to reality" award for his work on technology infusion of AMES planning and scheduling technology to AMMOS planning and sequencing legacy system. He has also received NASA's Exceptional Service Medal for his work on Voyager's Onboard Computer Command Subsystem for missions to Uranus and Neptune. He got his B.S.C.S. degree from Utah State University, Logan, Utah in 1982 and his M.B.A. degree from University of California, Los Angeles in 1993.*

2. ***Dr. Pierre F. Maldague*** *is a senior member of the Mission Systems Engineering Section at JPL. Currently, he is the Cognizant Programmer for the planning tool APGEN; he has also made major contributions to the design and implementation of sequence tools such as SEQ_REVIEW (a smart editor/translator for sequencing information) and SOA (Science Opportunity Analyzer). Dr. Maldague holds a Ph. D. in Theoretical Physics from M.I.T. After exploring his research interests in Quantum Mechanics at Purdue University and I.B.M., he joined Ford Motor Company in 1979 to work on computer-aided design, flame theory and engine manufacturing. He left Ford in 1984 to work on a variety of projects that included machine vision, robotics, image processing, and three-dimensional data visualization before joining Section 314 in 1993.*

4. ***Mr. Tung Bui*** *received the BS from California University of Irvine in 1994 and the MS from California State University of Long Beach in 1999, both in Computer Science. In 2001 he joined the Jet Propulsion Laboratory to take part in designing and implementing the Service Preparation Subsystem (SPS), a replacement of the Network Support Subsystem (NSS), on supporting missions using end-to-end service-based architectures in a way to enable DSMS to deliver more efficient and cost-effective support services to missions, eliminating unnecessary data interfaces, building direct user-DSN communication paths and pave a way for future Service Management era.*

5. ***Ms. Doris Lam*** *joined JPL as a software engineer in 2008 after graduating magna cum laude from UCLA with a bachelor's degree in Computer Science. She had worked as an undergrad at UCLA's Center for Embedded Networked Sensing lab, and likes to work on and experiment with new technologies and frameworks.*

6. ***Mr. John C. McKinney*** *has over 40 years experience in Space System Ground Data and Mission Operations Systems Management, operations and Systems Engineering in particular for NASA Deep Space Missions and multi-mission ground support systems and services. Currently the Chief System Engineer for the Advanced Multi-mission Operations System (AMMOS) at the Jet Propulsion Laboratory.*