The World's Forum for Aerospace Leadership

A Model-Based Approach to Developing Your Mission Operations System

Robert R. Smith, Kathryn A. Schimmels, Patricia D. Lock, and Charlene P. Valerio

Jet Propulsion Laboratory

AIAA SpaceOps, May 5 – 9, 2014, Pasadena, CA

Introduction

- Traditionally, Mission Operation Systems (MOSs) have been developed through adaptation of previous missions
  - Leads to problems with completeness, traceability, and understanding
- JPL extending a model-based systems engineering (MBSE) approach to MOS design
  - MBSE a paradigm to address needs for complex system design

How Do We Currently Build Our Mission Operations System?

- Document-based approach is a time-proven method
  - Mission Operations System Engineer (MOSE) analyzes a new mission's objectives, high-level requirements, architecture, and constraints
  - Selects a heritage mission's MOS for adaptation
  - Analyzes heritage design for differences, lacks, and unneeded capabilities, and creates a draft design for a new mission.
- Leads to designs that look like this:

How Do We Currently Build Our Mission Operations System?

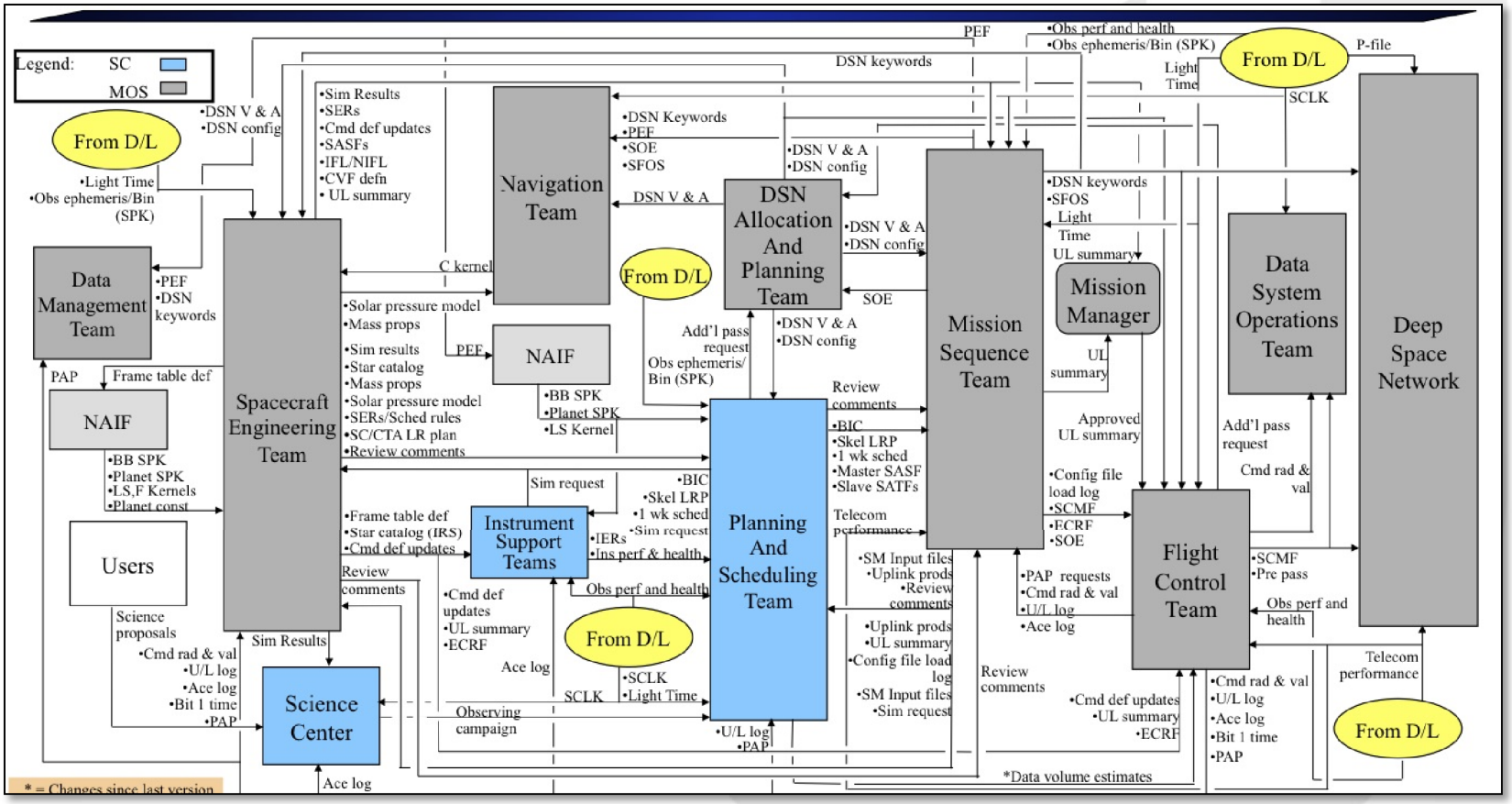
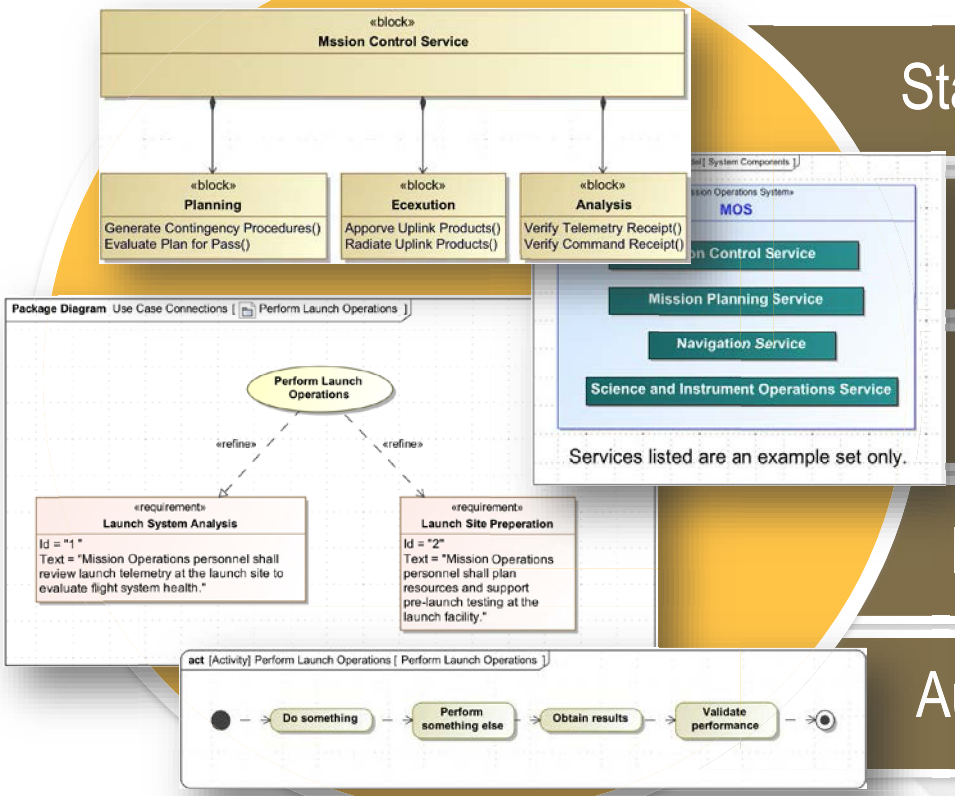


Figure 1. Heritage Mission Operations System Information Exchange Overview from a Document-Based Approach

Why Model a Mission Operations System?



- Standardized language (SysML)
- Single source of truth
- Enhanced traceability
- Knowledge capture between missions
- Automation capability

Document-Based MOS Design vs. Model-Based MOS Design

Top challenges encountered by MOSE's:

- Maintaining a single authoritative source of design information in a dynamic design environment
- Identifying and verifying requirements
- Assessing real impact of flight / ground trades on mission operations
- Capturing detailed interfaces & agreements using a standardized method
- Transitioning key flight system knowledge & information from development to operations

Document-Based MOS Design vs. Model-Based MOS Design

Single Authoritative Source of Design Information

- Generating design documentation/review materials
  - » Generally very time consuming
  - » Suffers from redundancy and overlap
  - » Often out of date; rarely read or used by intended audience
- Need to identify, capture, and visualize the *complete* MOS design
  - » system components, interfaces, internal and external agreements, operations processes, and procedures – <u>and seeing them clearly connected</u>.

Document-Based MOS Design vs. Model-Based MOS Design

Desired Features for Scenario Capture and Communication	Wiki	Online Scenario Database	Model-based approach
Collaborative authoring, editing, & reviewing	✓	✓	✓
Single authoritative source	Not always	✓	✓
Document Export & Generation	Copy and paste.	significant post-export editing required to incorporate into documents	• Standard document templates • Post-export editing allowable but not required • links to online model management web-editing capability
Requirements linkage	X	✓ (static)	✓ (dynamic)
MOS products integrated throughout project lifecycle	X	X	Scenarios tied to: • Agreements & Interfaces • Processes • Services, Teams, and Roles
Validation and analysis capability	X	X	• Scenario meets requirements • Interfaces used during scenario • Metrics to check completeness/correctness • Link to analysis tools

Figure 2. ***A Comparison of Methods in Capturing Scenario Information.*** Scenarios are not static – require iteration, collaboration, and are fluid early in the life cycle. Need modeling to capture and analyze scenarios to reduce cost/risk.

Document-Based MOS Design vs. Model-Based MOS Design

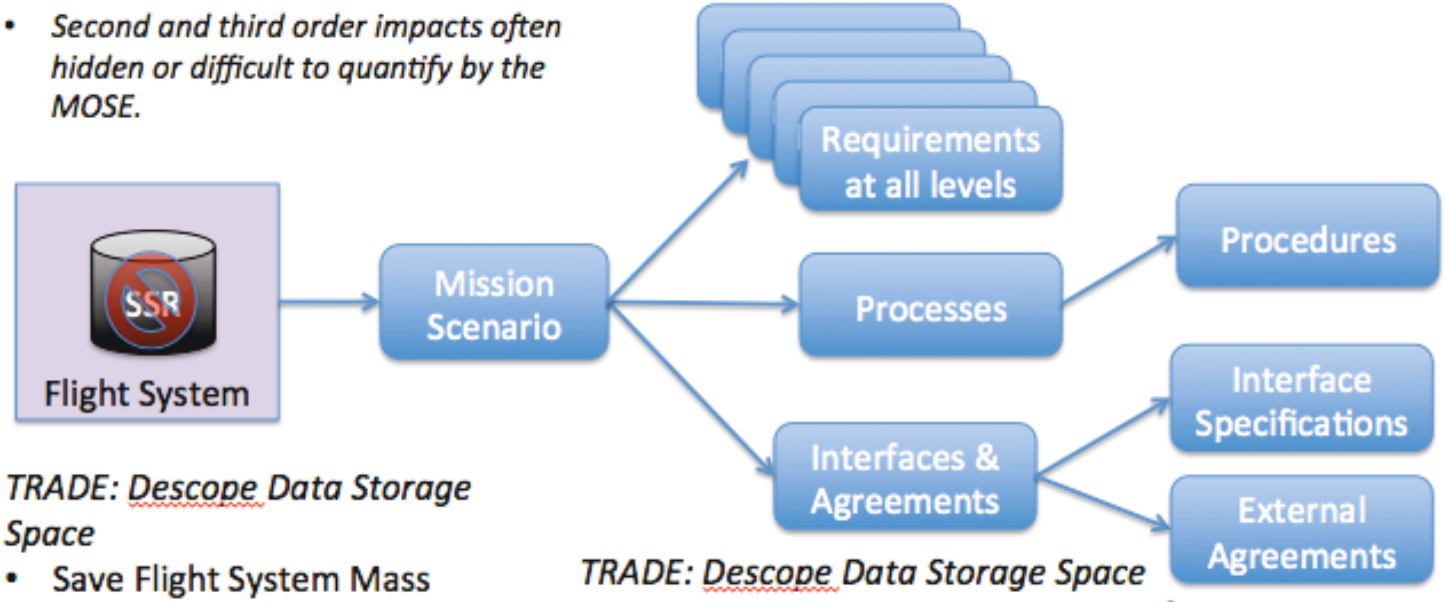
Developing and Verifying Requirements

- Identifying the right set of requirements for your specific MOS (vs. inheritance from previous missions with edits)
  - » Derived from scenario analysis, and mapped clearly to design elements and ops processes for verification
- Requirements change often during the development life cycle, even at the V&V stage
  - » Typical requirements management system provides hierarchical linkage, but lacks ability to interrogate the source and resulting design connections
  - » Respond to changes earlier, verify earlier, mitigate costly fixes later in development

Document-Based MOS Design vs. Model-Based MOS Design

Assessing Flight Ground Trade Impacts

- *Second and third order impacts often hidden or difficult to quantify by the MOSE.*



TRADE: Descope Data Storage Space

- Save Flight System Mass
- Save Flight System Cost
- Resolve Fit issues on S/C Bus

TRADE: Descope Data Storage Space

- Increase in ground station passes → costs
- Decrease in data collection volume/day
- Increase instrument contention for data collection time
- Longer sequences, change to seq dev process/timeline
- Increase in science mission duration
- ***But what else are we missing?***

Figure 3. ***A Typical Flight-Ground Trade Example.*** Reducing / removing on-board storage has a number of unanticipated impacts to ops scenarios, processes, interfaces. Modeling analyzes/identifies ripple effects of such a change.

AIAA
The World's Forum for Aerospace Leadership

Document-Based MOS Design vs. Model-Based MOS Design

Capturing Operational Interfaces and Agreements

	Word Doc	Web-based Tool	Model-Based Approach
Collaborative authoring, editing, & reviewing	X	✓	✓
Single authoritative source, robust config mgmt	X	✓	• Approval captured in the model
Consistent/standardized capture of i/f specifications	✓	✓	✓
Identification of complete set of all interfaces	X	X	• Interfaces and Agreements tied to scenarios and processes in the model • Ability to generate reports and metrics
Validation and training capability	X	X	• Identify interfaces early, during scenario dev. • Metrics to check completeness/correctness • Use during table tops and training

Figure 4. ***A Comparison of Methods in Capturing Operational Interface Agreement Information***. Key is to be able to identify not just what the interface is, but why the interface is needed. Tying interfaces to mission operations capabilities, teams, roles, and processes enables this as well as easy early validation/verification

Document-Based MOS Design vs. Model-Based MOS Design

Transition from Development to Flight Operations:

- The MOSE wants to leave a system in place for the Flight Ops Team which will allow the team to easily:
  - Align ground plans with changes in flight system operation
  - Asses change impacts
- The use of a model provides a team with:
  - The big picture view
  - A clearer understanding of system states at all stages
  - Metrics to help identify if a process is incomplete
  - Training capabilities for future operators – nominal and off-nominal situations can be simulated in the model

Document-Based MOS Design vs. Model-Based MOS Design

- Transition of knowledge & information from development
  - Capturing of Flight System information and transitioning it to the Flight Ops Team
  - Model produces useful products for training and Operational Readiness Tests
  - Flight System and MOS models provide an authoritative source for capture of information, and can provide a closed-loop control view of how we operate the flight system.

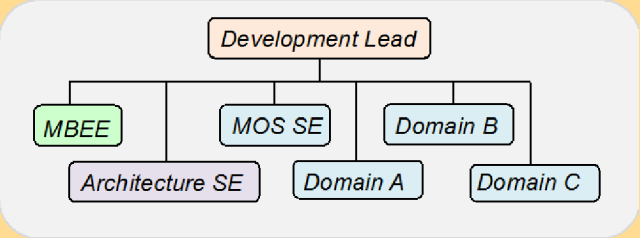
Persistence of a model throughout the entire mission life cycle is key

How Do We Build Models?

Process of Building Models:

1. **Define the scope of the system to be modeled**
   - System boundaries
   - Level of detail for the model
   - Design documents to be produced

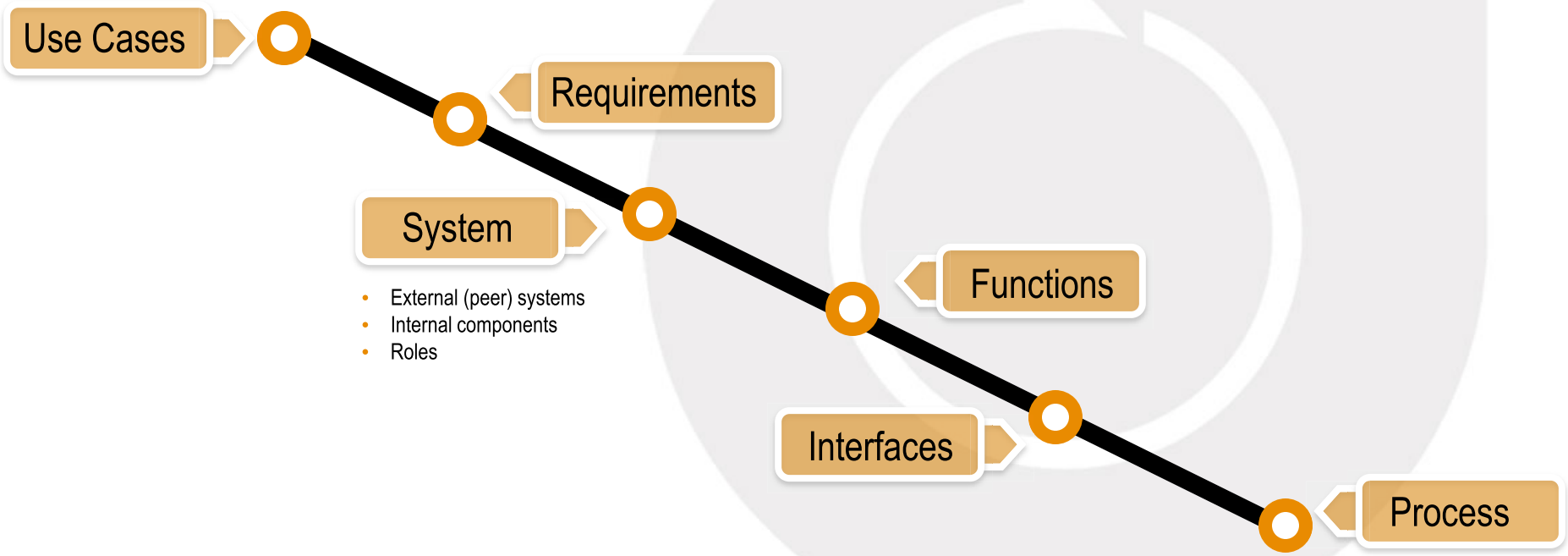
2. **Build the Team**
   - Model-based system enabling environment
   - Model architect
   - Domain experts



3. **Establish Reviews & Criteria**
   - Management review – *"Is it within budget, schedule, scope, and risk?"*
   - System being engineered – *"Are we building the right system"*
   - Model architecture – *"Are we building the system right?"*

How Do We Build Models?

- Utilize a top-down approach in designing the system model



What Have We Built?

- OpsRev
  - A subset of the MOS 2.0 initiative developing a next-generation operations system
  - MOS 2.0 is planned to incorporate a number of models in its tool suite, while also encapsulating operations practices
- MOSE and GDSE Development Procedures Model
  - Text-to-model conversion
  - Ambiguity and conflicts aligned

Future Work

- GDS Model to Match OpsRev model
- Connection of the MOSE and GDSE system engineering procedures to the OpsRev model, streamlining system engineering workflow.
  - Also more detail to be added, capture of research resources, “tips and tricks”, lessons learned, and historical contacts
- Mars 2020 and Europa Clipper missions intend to use architecture-driven model-based systems engineering for both flight system and ground system development.

Acknowledgments

- The work described in this presentation was performed at the Jet Propulsion Laboratory (JPL), managed by The California Institute of Technology (CalTech), under contract to the National Aeronautics and Space Administration (NASA).

AIAA

The World's Forum for Aerospace Leadership