Model-Based System Systems Engineering Pilot Program at NASA Langley

Kevin Vipavetz
Senior Systems Engineer
Langley Research Center
Email: Kevin.G.Vipavetz@nasa.gov
Phone 757-864-3817

Introduction

- Why Model-Based Systems Engineering (MBSE)?
  - The Engineering Directorate and Office of Chief Engineering at Langley Research Center (LaRC) wanted to inject MBSE into a small flight project as a pilot program for evaluation.
  - We were interested in the benefits and challenges of implementation within our Systems Engineering (SE) framework.
- Materials on International Space Station Experiment (MISSE-X) project was used for this pilot due to Interface complexity.
  - MISSE-X is hosted on the International Space Station (ISS).
  - More than 900 ISS interface requirements!
  - Multiple experiments with multiple principle Investigators and other key stakeholders.

Motivation for MBSE

- Complex interfaces
  - Maintaining a consistent understanding of the system concept or design can be laborious and error prone using traditional systems engineering approaches.
  - Communication between discipline experts is complicated by the use of disparate terms, tools and methods.

- End-to-end architectural analysis
  - Being able to integrate perspectives and concerns of various disciplines into the overall system.
  - The need to coherently communicate concepts and constraints between system-level and discipline specific analysis.

MBSE Defined

- INCOSE-TP-2004-004-02, Version 2.03, September 2007
  - Model-based systems engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases.
- MBSE uses models to define, analyze, design, and implement systems.
  - It can be applied to existing systems, systems in development, or conceptual systems.

Goals of the LaRC MBSE Pilot Program

- 1. Centralized Access
- 2. Consistency
- 3. Improved Confidence
- 4. Improved Communication
- 5. The Advancement of MBSE and SysML Skills at LaRC

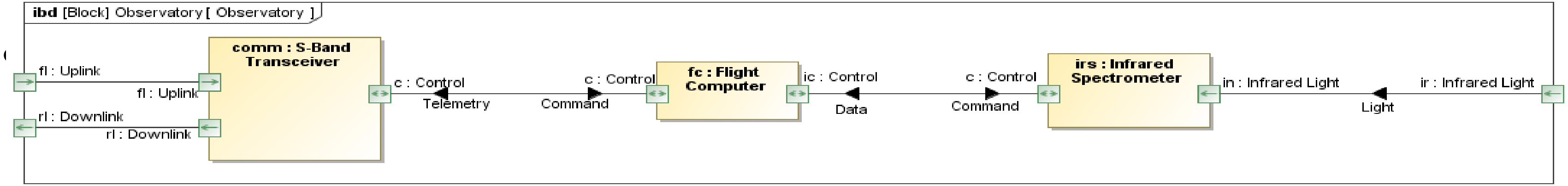
Tools and Methods

- Focus on capturing system architecture and requirements for MISSE-X Systems Requirements Review.
- System architecture in Systems Modeling Language (SysML) – MagicDraw from No Magic Inc.
- Requirements management in relational database – CORE from Vitech Inc.

Note: While MBSE can be used for linking simulations and executable models, this was not the focus of the pilot program.

What is SysML?

- The Object Management Group (OMG) describes SysML as:
  - A general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities. http://www.omgsysml.org/.



MagicDraw and CORE

- A single modeling tool for both requirements and architecture would have been ideal, but not practical.
  - Existing expertise with MagicDraw.
  - Existing use of CORE at LaRC.
  - Lack of features by any one tool to encompass both domains.
- Both tools allow concurrent development on shared databases as opposed to one document centric approach.
- One drawback: requirements had to be exported from CORE to MagicDraw when we needed them (not ideal).

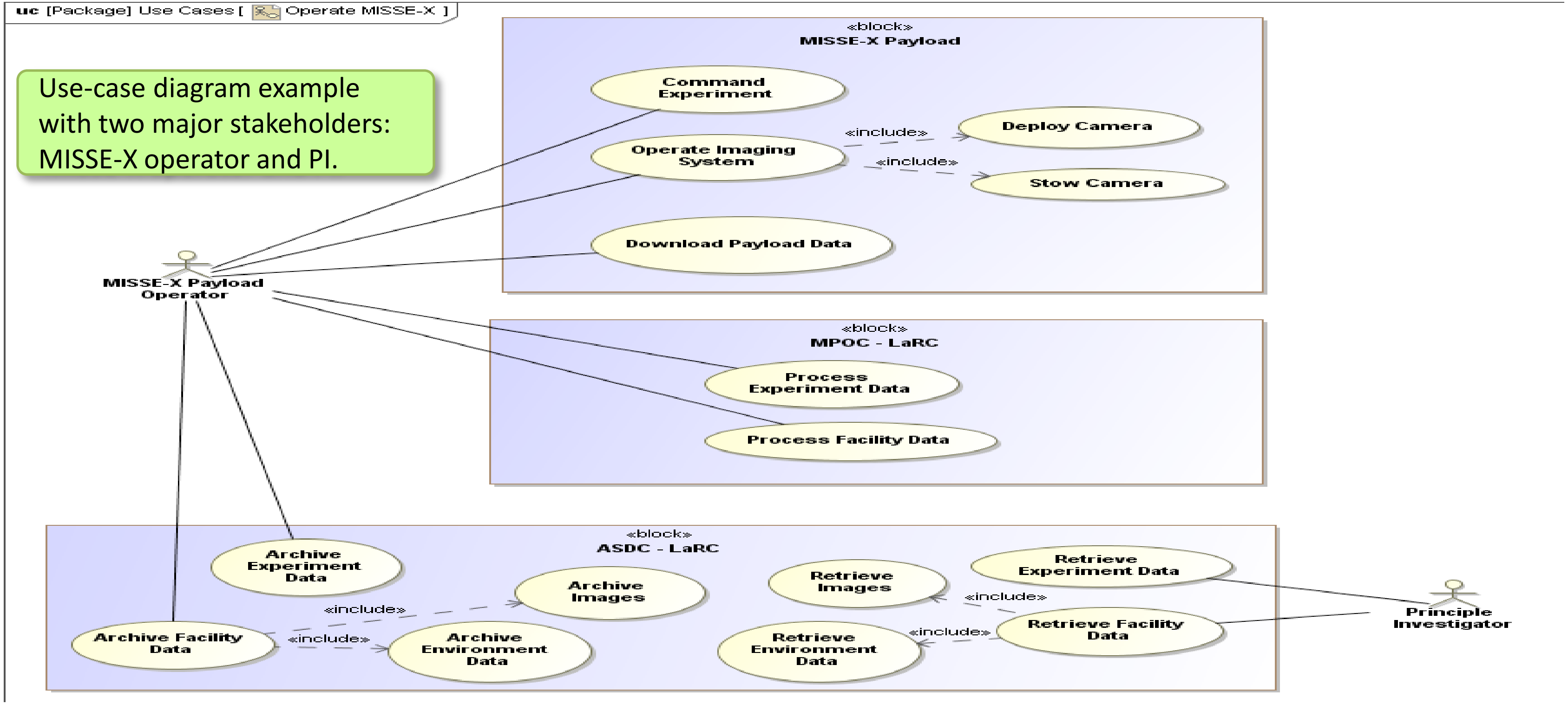
SysML Resources

- The team used the following for guidance during SysML model development:
  - A Practical Guide to SysML: The Systems Modeling Language By Sanford Friedenthal, Alan Moore, Rick Steiner (2009).
  - The results of the INCOSE Telescope Modeling Challenge Team (http://www.omgwiki.org/MBSE/doku.php?id=mbse:telescope) including the SysML cookbook (http://mbse.gfse.de/documents/faq.html).
  - Previous MBSE NASA presentations developed by JPL, JSC, GSFC and others. Particularly:
    - Piloting Model Based Engineering Techniques for Spacecraft Concepts in Early Formulation (Cole, Delp, Donahue, 2010).
    - MBSE Webinars by Dan Dvorak/JPL as part of the NASA Systems Engineering Working Group's (SEWG).

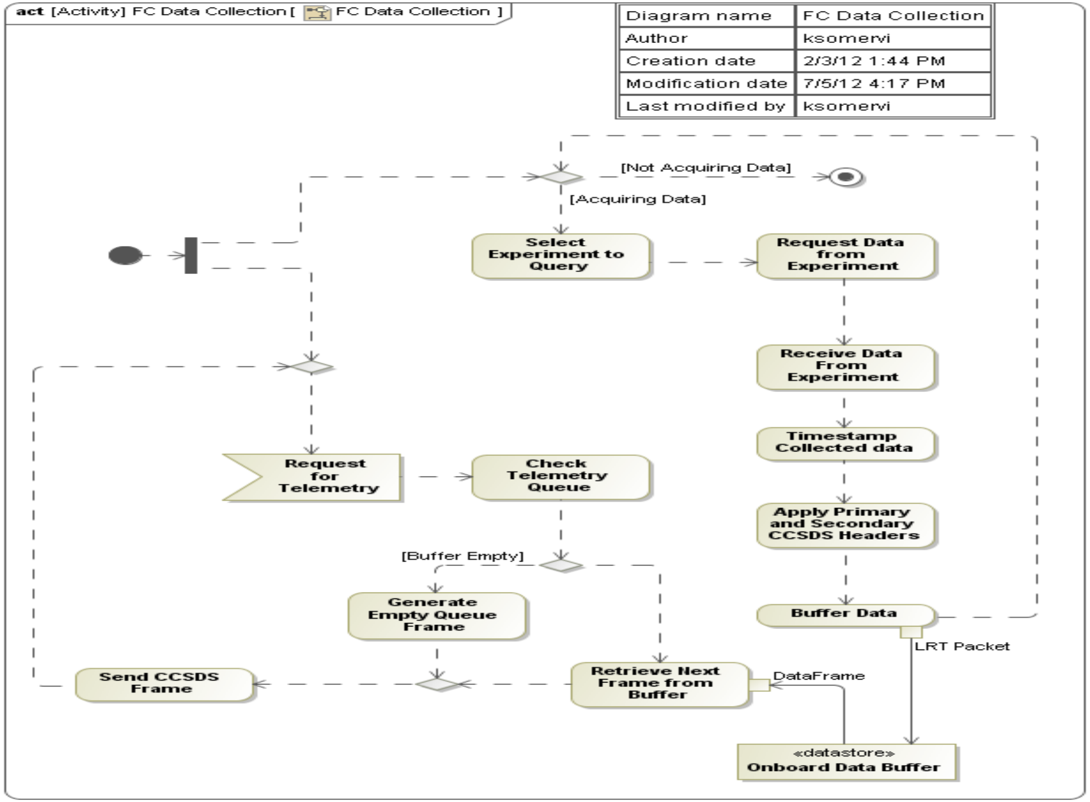
Our Process

- Generally aligned with Object Oriented Systems Engineering.
  - Use-case diagrams: stakeholder interactions captured.
  - Activity diagrams: system functions documented.
  - Package diagrams: system architecture and external boundaries developed.
  - Internal block definition diagram: flows and interfaces between systems identified.
- A well defined systems architecture benefitted the development of the concept of operations (ConOps).

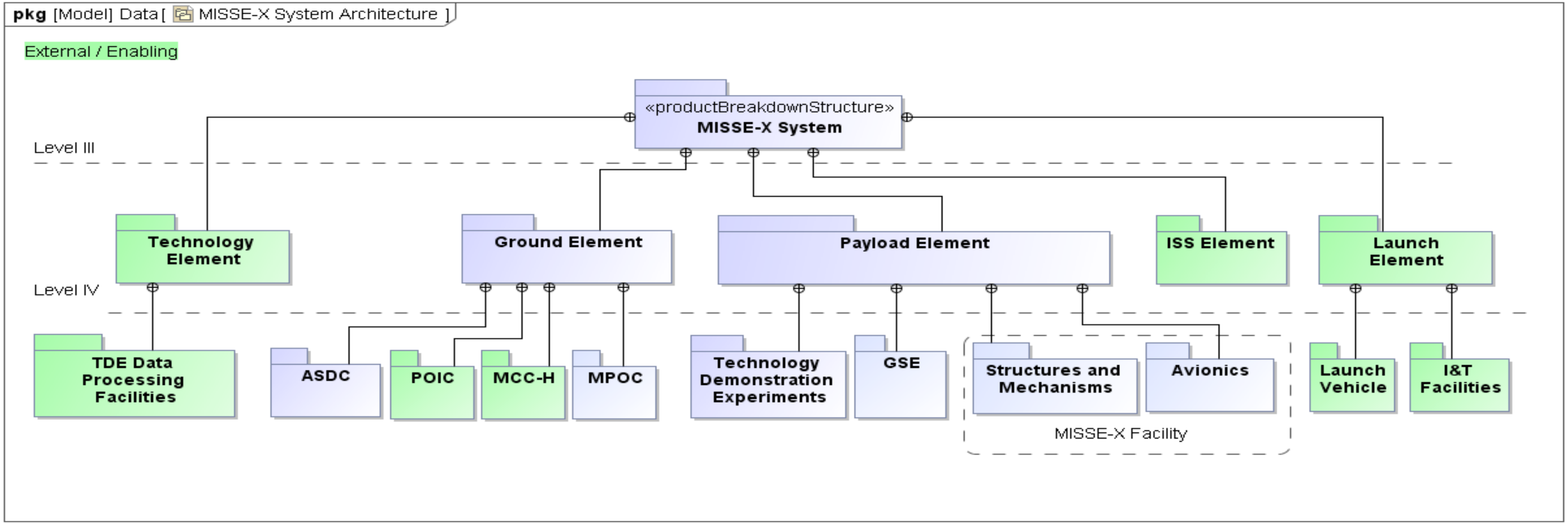
Use Case Example



Activity diagram example showing the major functions of the MISSE-X Facility Controller, a part of the facility avionics system.



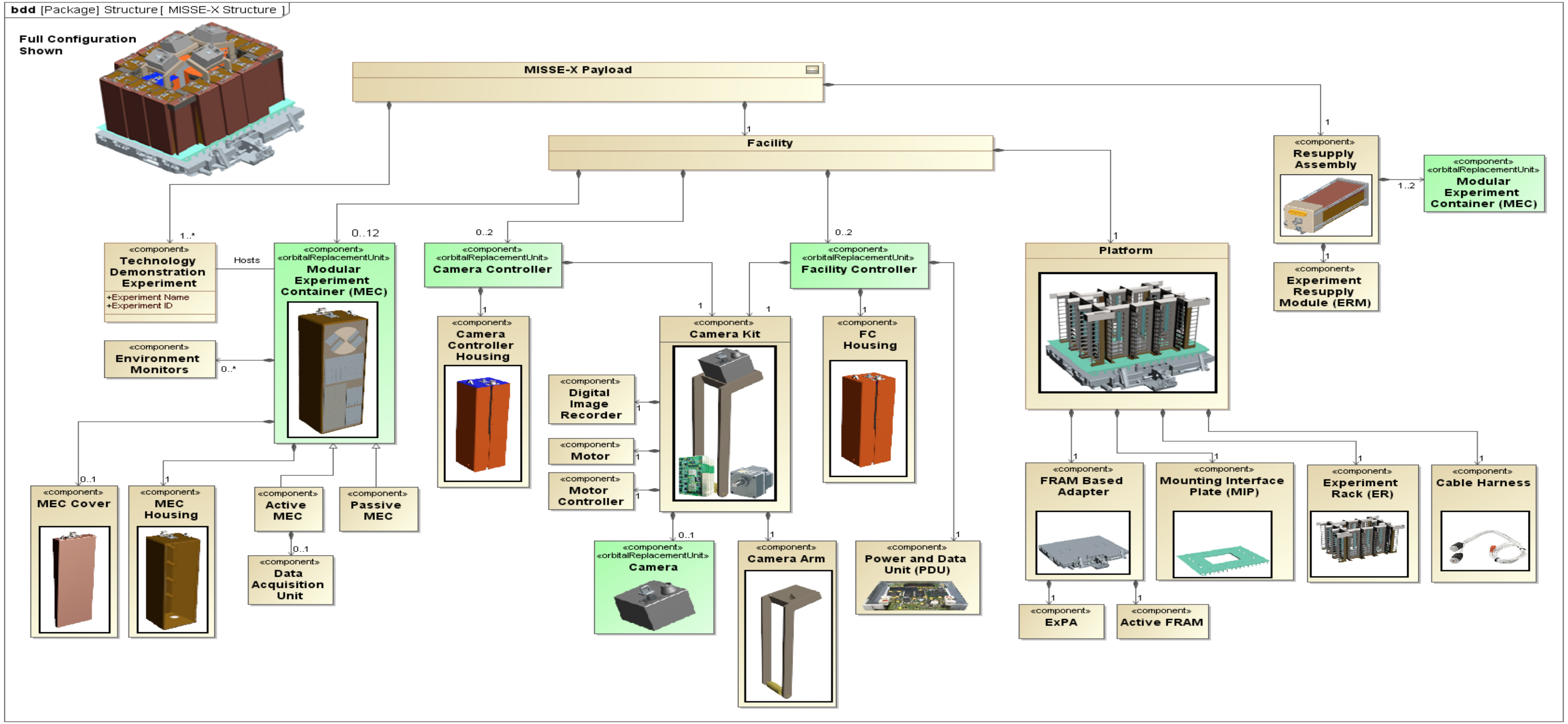
A SysML Package Diagram example depicting the MISSE-X Systems Architecture. The connections between the system-of-interest (blue boxes) and external/ enabling systems (green boxes) defined key external interfaces.



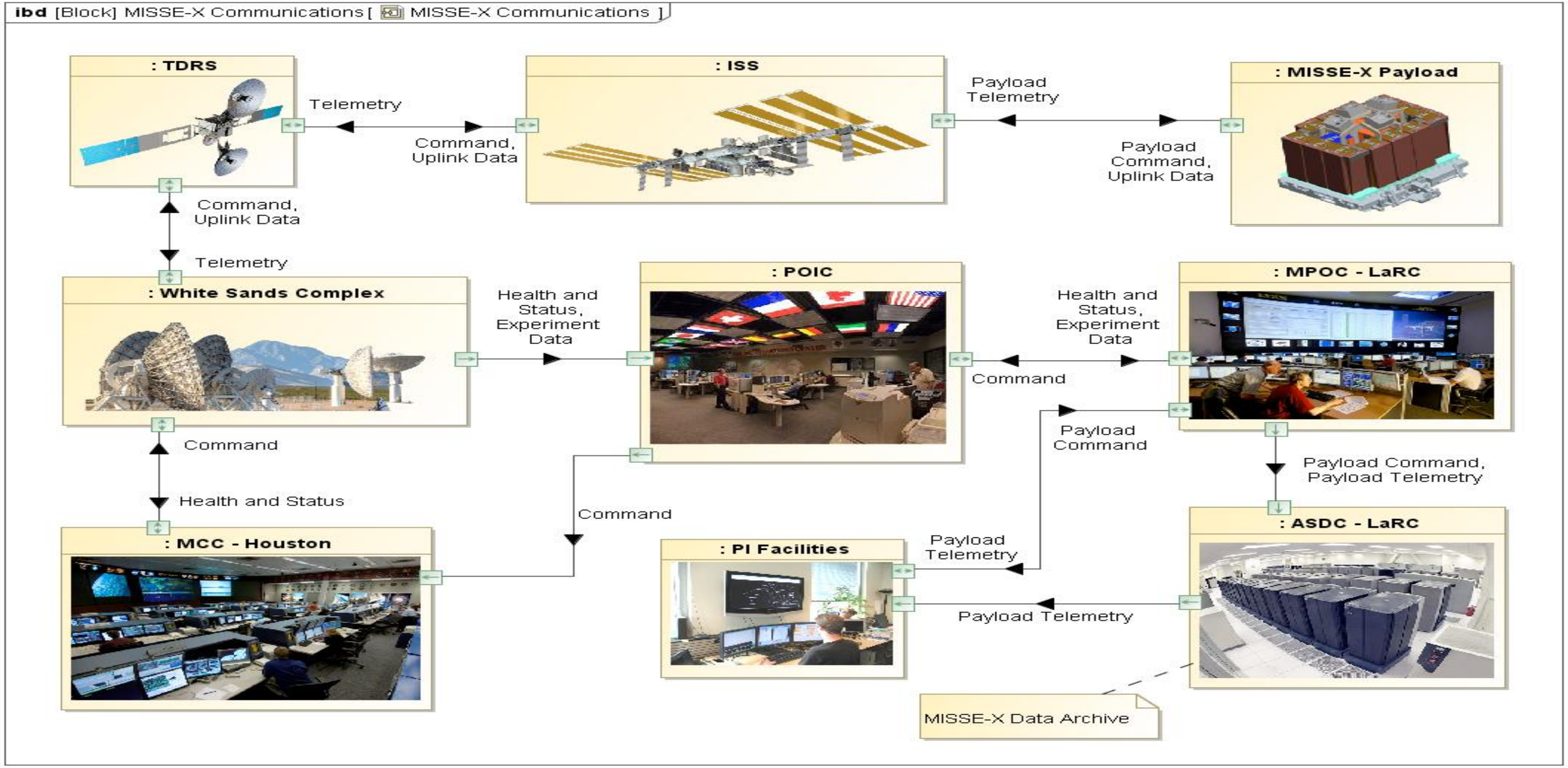
Modeling Product Details

- Block definition diagrams
  - Organizing hierarchical assemblies and physical structures.
- Activity diagrams
  - Creating operational flows.
  - Map out functions and create requirements.
- Internal block diagrams
  - Identify internal interfaces.
- Modeling ensured a shared understanding of the system from top down and bottoms up perspective as it was being developed.

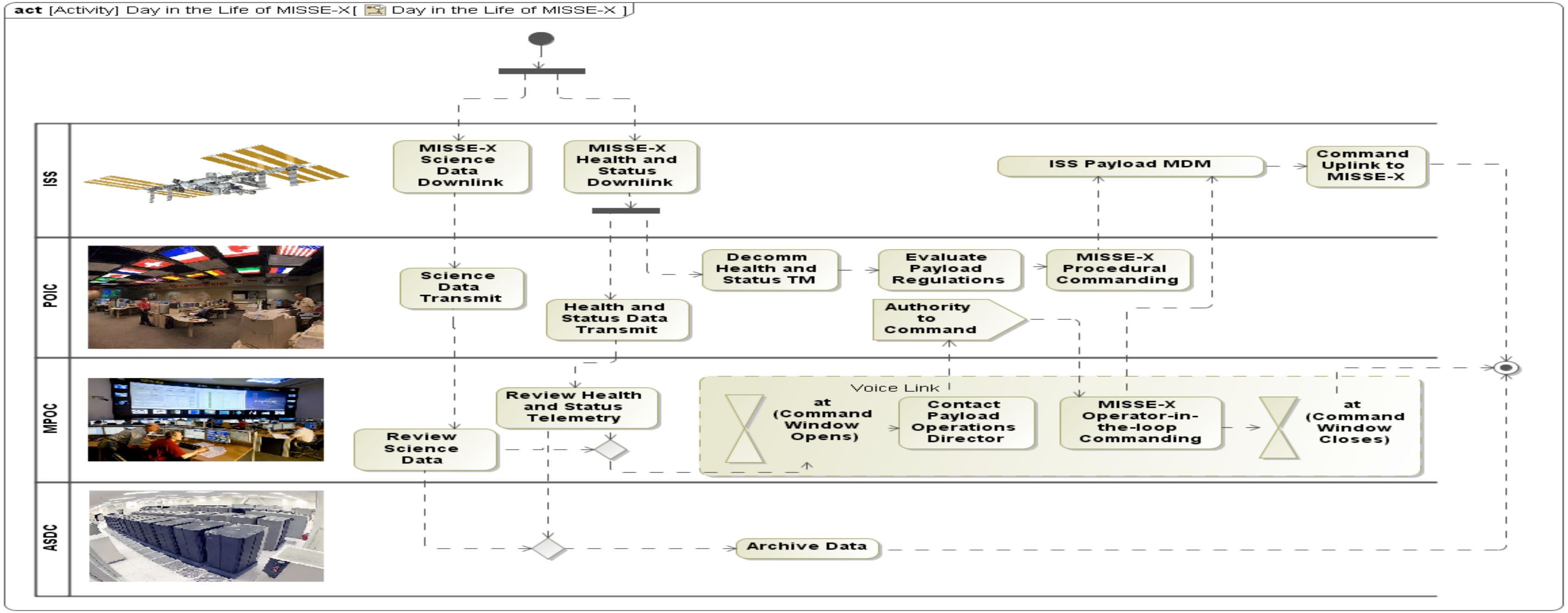
Block definition diagram example showing the composition of the payload's physical structure.



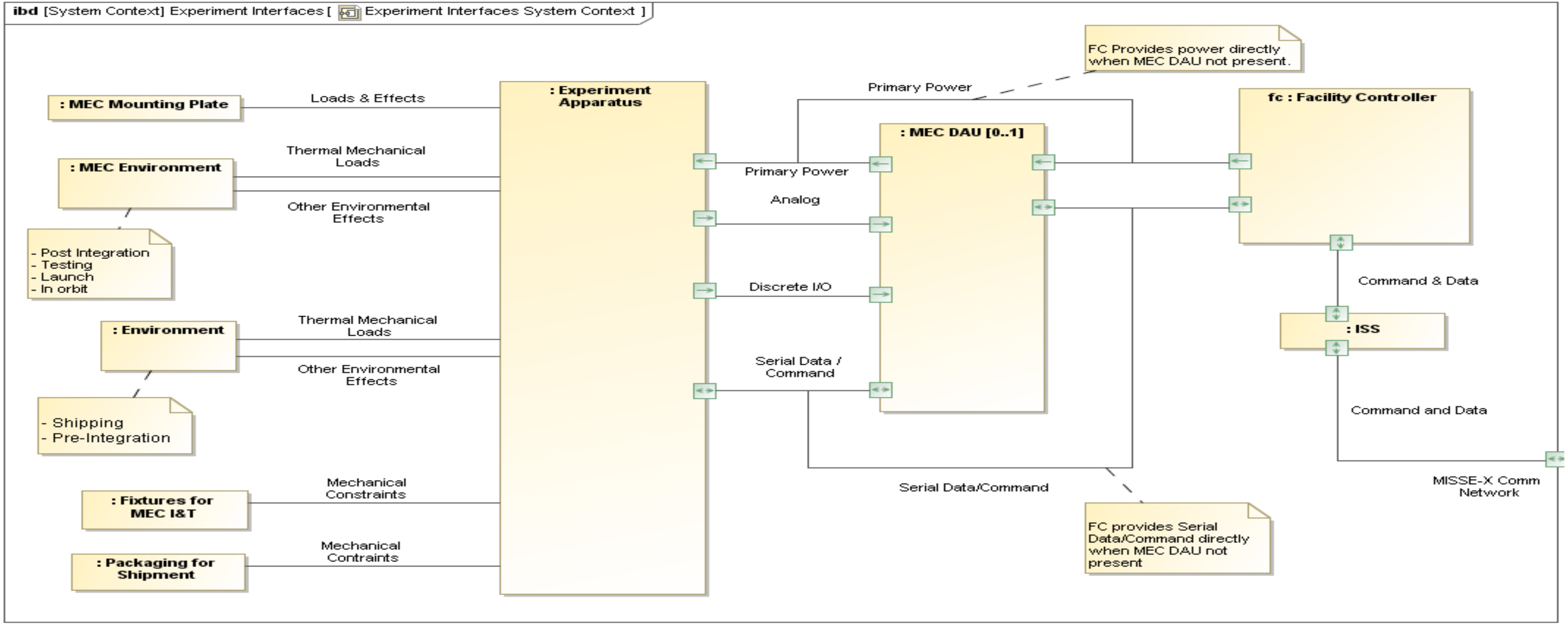
Internal block diagram example showing the communications connections and flow through the system and external supporting elements.



Activity diagram example of a day in the life of MISSE-X operations.



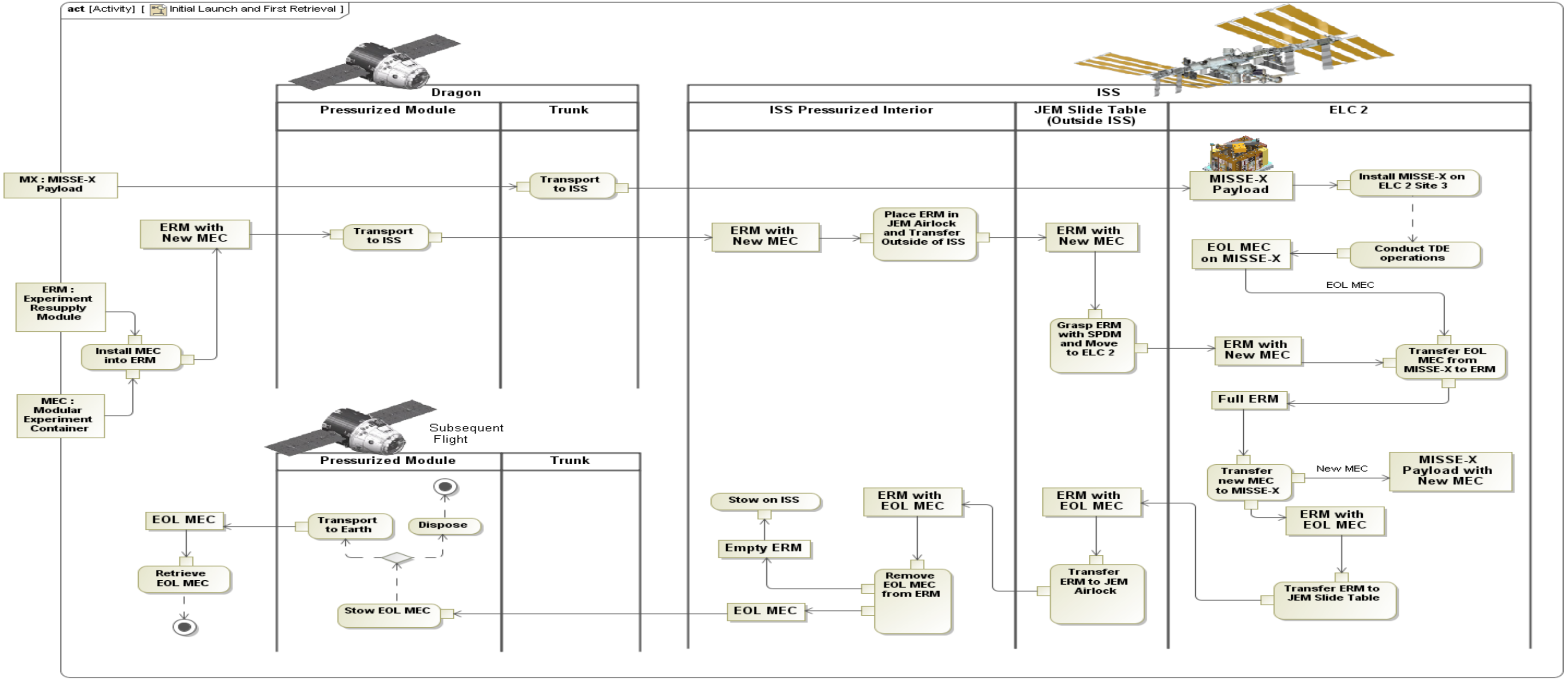
Internal block diagram example showing the interfaces with experiment apparatus. This diagram helped define and communicate the resources, constraints, and environment that will affect experiments on MISSE-X.



Centralization

- Benefits:
  - Connected model elements facilitate the comprehension of the context and dependencies of the system, especially as the design details are developed.
  - A single model provides a consistent point of departure.
- Challenges:
  - More than one database – had two.
  - Required special clients to access the database (as opposed to familiar MS Office applications).

Activity diagram example of the transportation concept for the initial launch of MISSE-X and a subsequent resupply mission. “Swim lanes” identify locations of the activities and SysML central buffer nodes are used to represent MISSE-X hardware.



Consistency

- Benefits:
  - Benefit is analogous to using Unified Model Language (UML) for software – re-use of elements guarantees that they are exactly the same object.
  - For SysML, this was useful in tracking connections between elements and element names.
  - For CORE, this was useful in making sure requirements wording was identical in every place that it appeared.
- Challenges:
  - Had to manually port the requirements from CORE to MagicDraw.

Confidence

- Benefits:
  - Helped provide design feasibility that meets the project need and risk posture.
  - Provided traceability to requirements (using CORE).
  - Complex and numerous interfaces were addressed through CORE and internal block diagrams.
  - Complex avionics were addressed through block definition diagrams.
  - The modeling approach enhances thoroughness that increases confidence in the validation of the design.
- Challenges:
  - How do you know if the model is right? Important to have sufficient MagicDraw expertise and subject matter experts familiar with SysML diagrams to review models.
  - Lack of models for our domain. Would be nice to draw from a library of SysML models for ISS hosted payloads.

Communications

- Benefits:
  - SysML is less ambiguous than the English Language.
  - Standard specifications enable depiction of a consistent set functions and features.
  - SysML provided a good shorthand for conveying a lot of information.
  - Motivated model developers to think about the system in an object-oriented way.
  - In the case of MISSE-X, this allowed stakeholders versed in SysML to converse more efficiently.
- Challenges:
  - SysML learning curve for authors.
  - SysML learning curve for reviewers and other stakeholders.

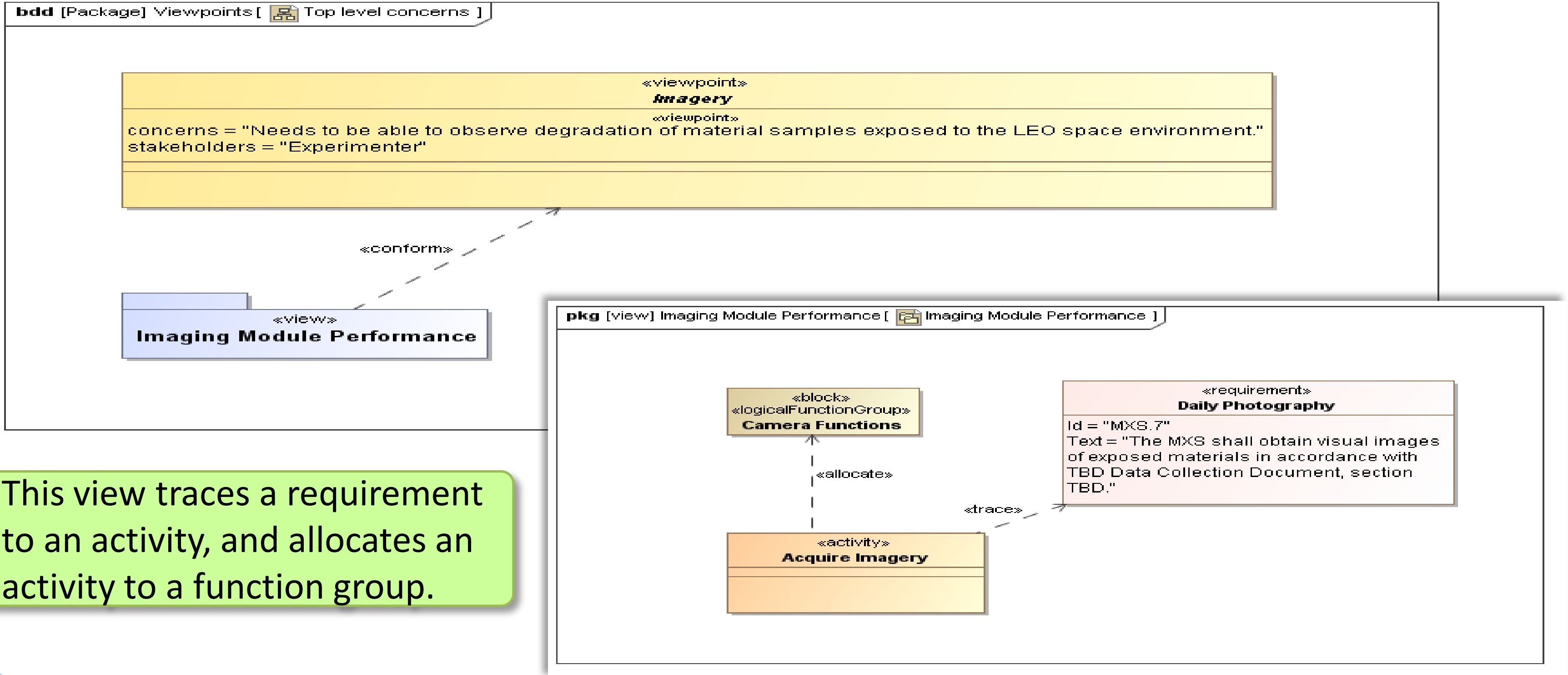
Advancement of MBSE Skills

- Working reviews of SysML diagrams advanced skills in learning the language.
  - Activity diagrams and internal block definition diagrams were the easiest to digest for SysML beginners.
  - Block definition diagrams were recognizable to software developers.
- SysML diagrams were used in the baseline project documentation, which helped advance SysML familiarity.
- Advancing to a fully realized MBSE implementation will need a model-centric configuration control approach.

Next Steps

- Design Reference Missions
  - Creating specific MISSE-X configurations to support a series of design analysis cycles. All nine SysML diagram types will likely be used.
- Test development & documentation
  - Utilizing activity diagrams and ibds to development test plans and document test setup (including cabling diagrams).
- Views & Viewpoints
  - A technique identified by IEEE Standard 1471-2000 (and more recently ISO/IEC/IEEE 42010) to capture the concerns of particular stakeholders and express the model with respect to these concerns.

Examples of viewpoint: Below identifies a stakeholder concern and points to a relevant view of the model.



This view traces a requirement to an activity, and allocates an activity to a function group.

Conclusion

- The Pilot Program showed:
  - The investment of effort in the model-based approach is significant, but can yield significant returns.
  - Some of the benefits that UML provides the software community were also provided by SysML for the SE activity on MISSE-X.
- Some final recommendations:
  - *Start simple.*
    - Use the basic elements of SysML to document high level topics such as the product breakdown structure.
  - *Support the team infrastructure.*
    - The MBSE tools will only realize their potential when the models become the authoritative technical description of the product.
  - *Go for breadth*.
    - Involving more of the subsystems leads at a beginner level could be more beneficial than having one or two SysML experts.

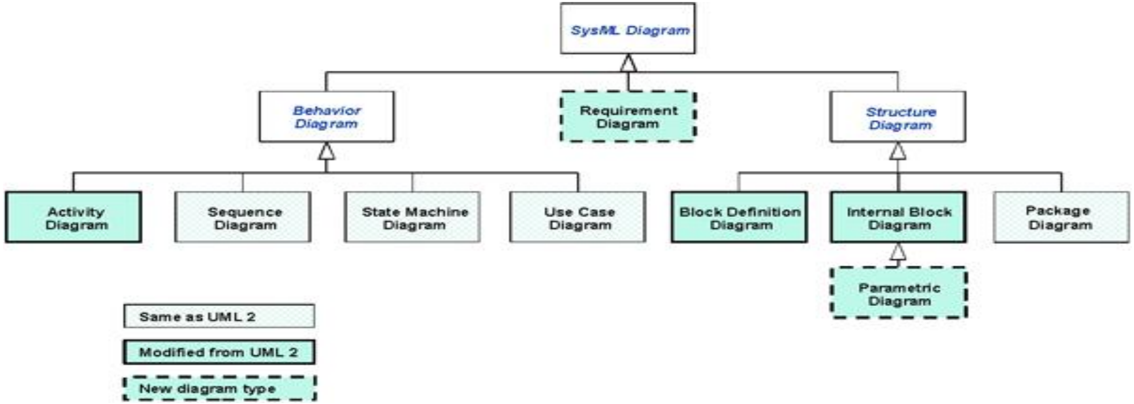
Acknowledgements

- Special thanks to former MISSE-X teammates Doug Murphy and Dr. Samantha Infeld who helped produce the MBSE LaRC Pilot AIAA paper and with the development of an earlier AIAA version of this presentation.

Backup

SysML Diagram Types

- **Structure**
  - Package diagram (pkg)
  - Block definition diagram (bdd)
  - Internal block diagram (ibd)
  - Parametric diagram (par)
- **Behavior**
  - Activity diagram (act)
  - Sequence diagram (sd)
  - State machine diagram (sm)
  - Use case diagram (uc)
- **Requirements diagram (req)**



Source: http://www.omgsysml.org/