26th annual INCOSE international symposium

Edinburgh, UK
July 18 - 21, 2016

Interface Management for a NASA Flight Project using Model-Based Systems Engineering

Kevin Vipavetz Langley Research Center, NASA Langley

Jim Price Langley Research Center, NASA Langley

Dr. Thomas Shull, Analytical Mechanics Associates

Dr. Samantha Infeld, Analytical Mechanics Associates

What is an Interface?

- An Interface is a common boundary between two systems of interest where they interact
- Each interface can be considered to have a source, destination, and some means to allow interaction
- There are three common types of interfaces:
  - Functional
  - Physical
  - Environmental

Interface Definition

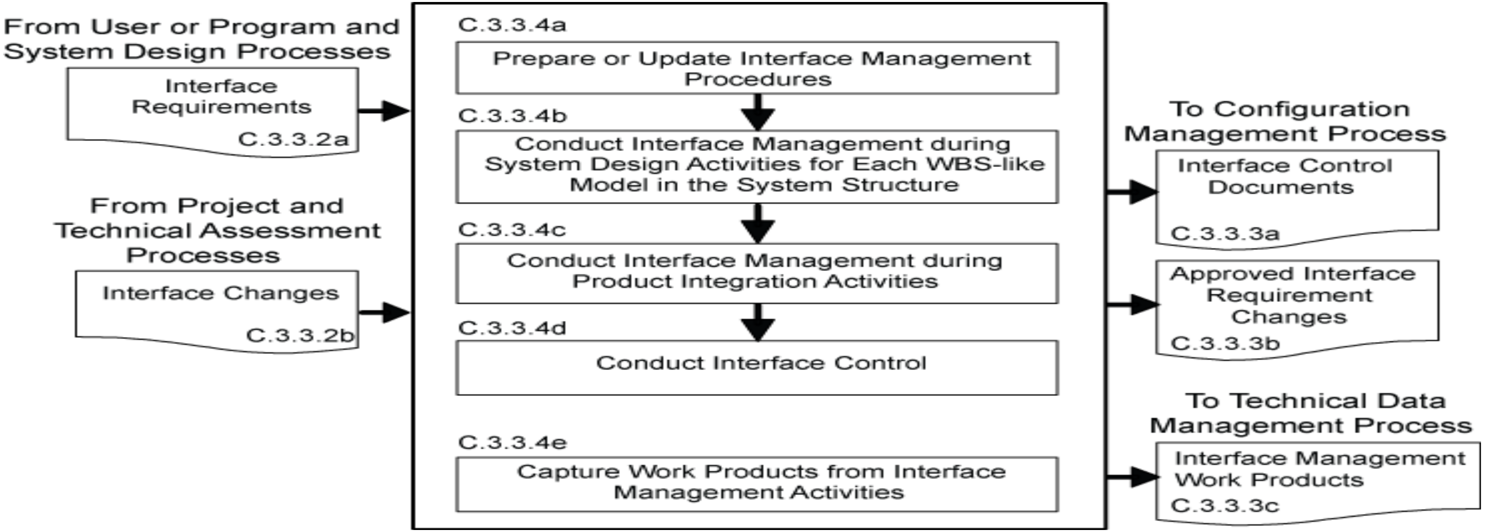
- The interface definition defines the boundary between the two sides
- To develop an interface definition each side must know:
  - The characteristics of each system at the interface
    - Examples: material, structure, mass, loads…
  - The characteristic of what is crossing the interface
    - Examples: current, data, strain, sheer, fluid, heat…
  - What is the media of the interaction
    - Examples: attachment bolts, wires, pipes, environment…
- Interface definitions are captured in a Interface Control Document (ICD)

Systems Engineering Management Plan (SEMP)

- The Lead Systems Engineer should develop the SEMP as soon as possible to manage the technical side of the project
  - The NPR 7123.1B “NASA Systems Engineering Processes and Requirements”, is a good outline for SEMP development
- How interfaces are to be managed by the project are captured in this document
  - The SEMP can point to another document for details or define the details in the SEMP itself
- The SEMP should be a baseline document during project formulation stage

Interface Management Process

- Below is an example from the NPR 7123.1B "NASA Systems Engineering Processes and Requirements"



Interface Accountability

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

- Assign interface custodians for each interface document
  - Accountable for ensuring the interfaces are complete, are following processes, helps track changes, and who herds the cats
  - Maintains documentation
  - Essential in developing interface agreements between external partners
- Assign interface requirement owners for each interface requirement
  - Accountable for lifecycle development, compliance and implementation of the assigned interfaces
- Accountability is different from responsibility
  - The team is responsible for helping develop interfaces, only one is accountable

Configuration Control

- Use your requirements database to configuration manage and control your interface requirements, interface definitions, interface verifications, change requests, deviations and waivers, and for generating baseline requirement documents

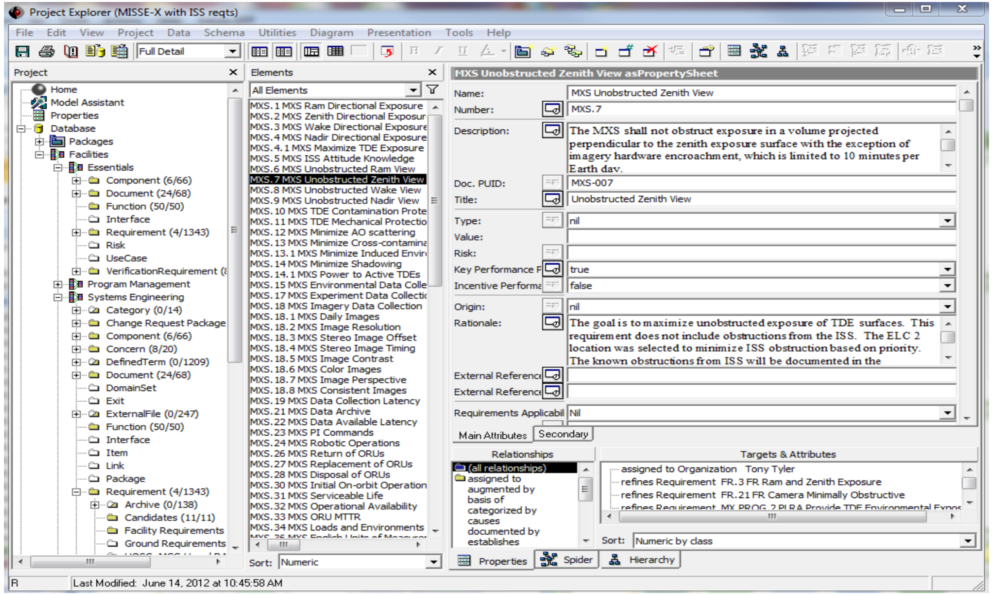
Tip: It is not necessary to wait until the whole interface document is ready for baseline to begin interface development. It can go through partial baseline approvals. This allows for the opportunity to not delay development of already approved critical interfaces.

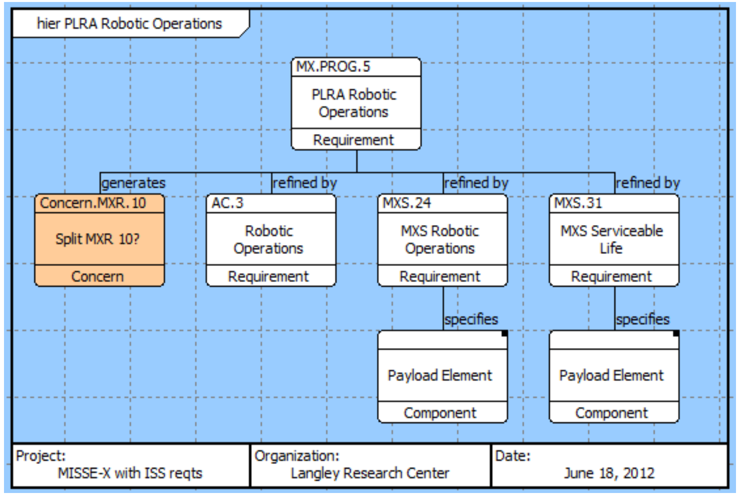
- Example: Draft, Pending, or Baseline status can be assigned to each interface definition and tracked by configuration management.

Requirements Management Tool

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

- You will need a management tool to keep track of your interfaces and requirements





Interface Working Group (IWG)

- Essential to interface development
- Involves the appropriate parties and subject matter experts
- Used to plan development and control of interface processes
- Used to help identify interfaces
- Used to define interface definitions
- Review change requests
- Lays the groundwork for binding agreements between all applicable organizations

Using Model-Based Systems Engineering (MBSE)

- Set up MBSE to do the following:
  - To centralize information
  - To improve consistency
  - To improve communication
- This will promote confidence in your team

Implementing MBSE

- Use an MBSE tool to provide:
  - Use case modeling for Concept of Operations (ConOps) generation
  - Architecture modeling
  - Functional modeling to capture interface functions
  - Interface modeling to capture boundaries and what flows across a boundary
- Link models to a requirements database
- Generate reports based on models
    - Example: Generating a ConOps from Use Cases

Seven Step Interface Approach

1. Identify
   - Perform interface analysis to identify interface boundaries
2. Capture
   - Capture interface requirements in requirement documents (not ICDs)
   - Place under configuration control
3. Define
   - Develop interface definitions or determine applicable interfaces for pre-existing interface documents
   - Place under configuration control
4. Allocate
   - Flow down any interface requirements to next level product

Interface Approach cont…

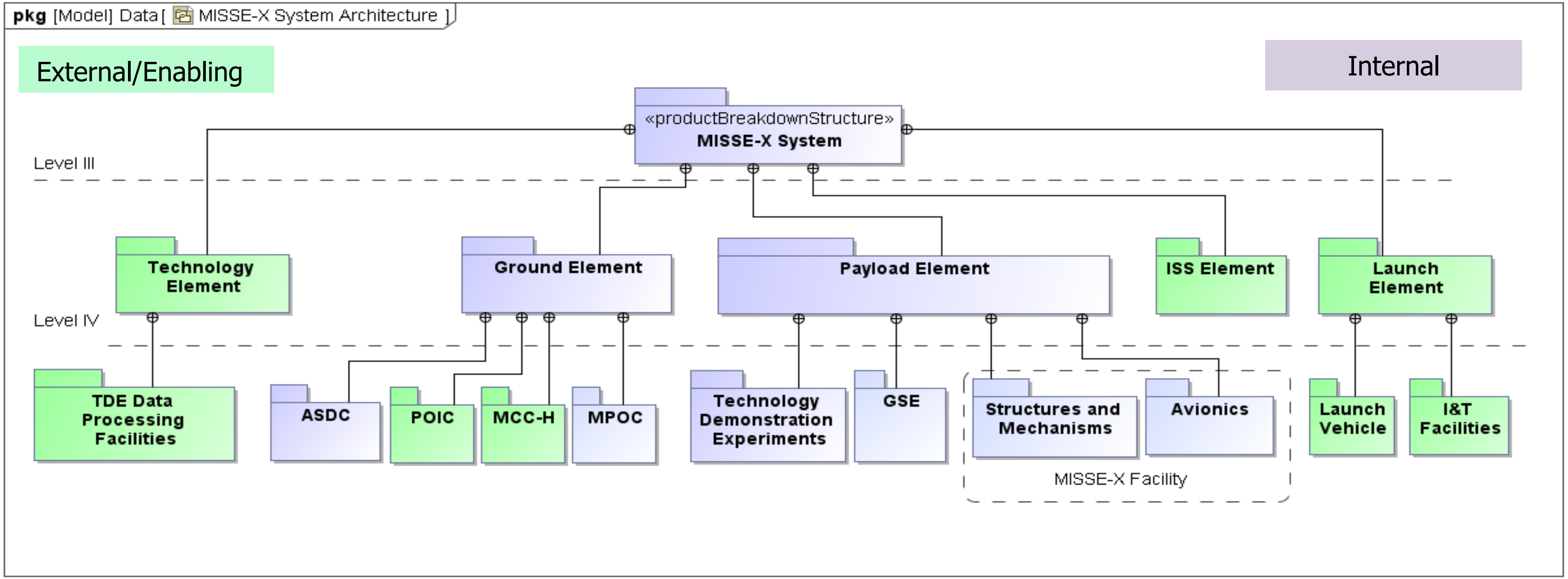
5. Verify
   - Define interface verification activities and success criteria
   - Conduct verification activities
   - Place results under configuration control
6. Comply
   - Write up verification compliance reports
     - Responsibility of the interface requirement owners
     - Interface requirement owners start the configuration control process for closeout and approval
7. Integrate
   - After assembly and testing of an interface side, flow up to next level and checkout integrated interfaces (Note: These are different interface requirement owners at different levels)
   - Repeat steps five through seven until system is complete

Identifying Interfaces

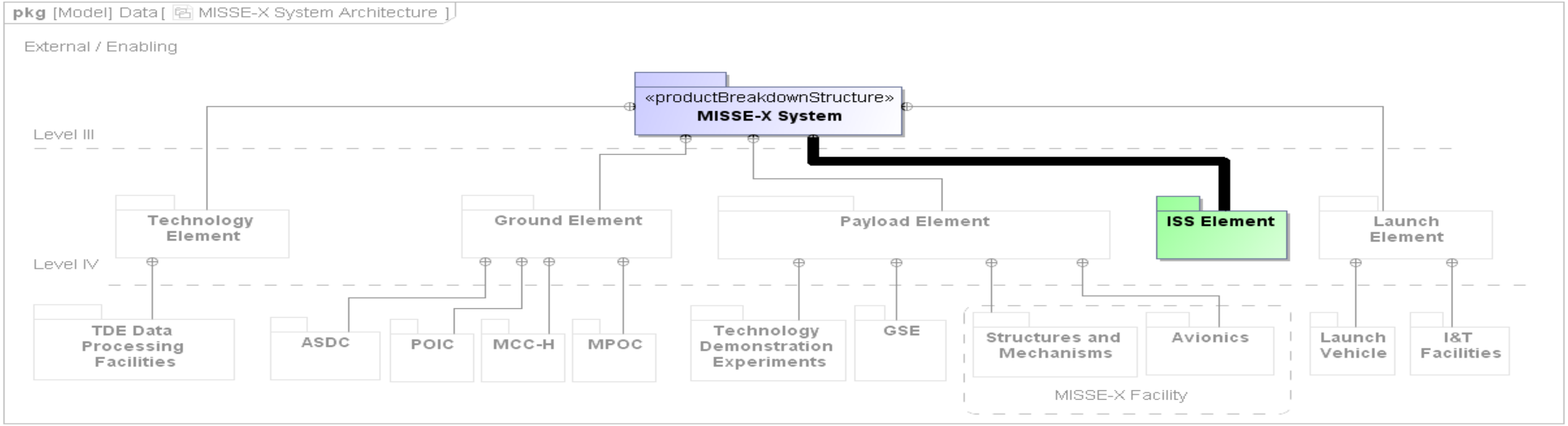
- Develop scenarios from stakeholders for the whole lifecycle of the project
- Integrate into single concept of operations
- Generate a system architecture
- Use system architecture to help identify interfaces

System Architecture Example

(All MISSE-X system modeling performed in MagicDraw®)



Step 1: Identify External Interfaces

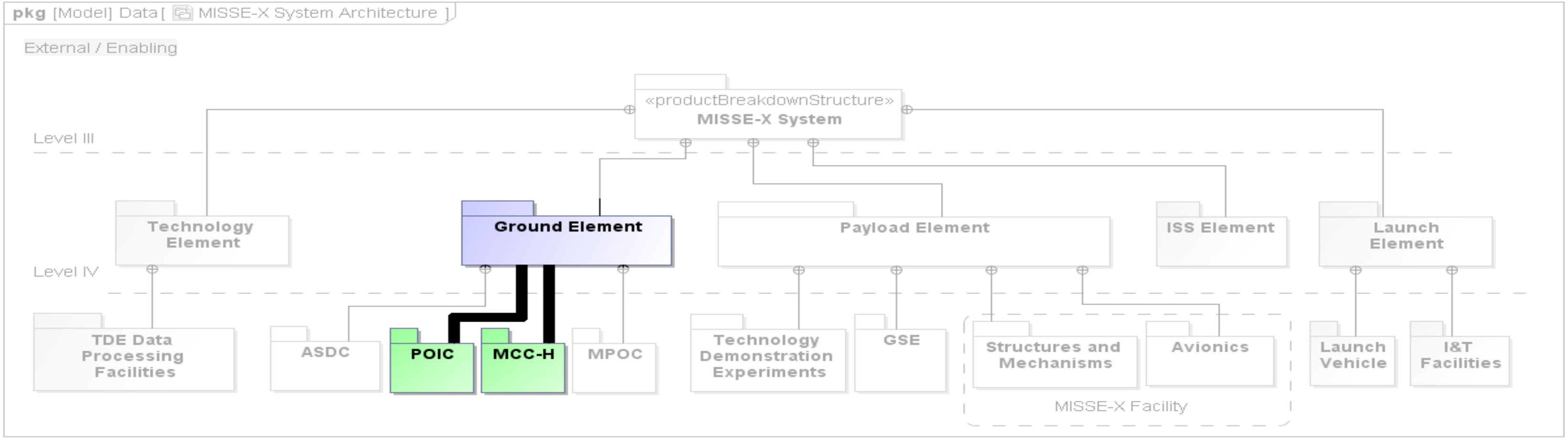


Interfaces:

Mechanical, electrical, thermal, and environment interfaces during:

- **Transfers, installation, de-installation: Extra-vehicular robotics (EVR), including JEM ORU Transfer Interface (JOTI)**
- **Operations: ISS exterior (ELC)**
- **ORU replacement/disposal: ISS interior storage and handling**
- **Contingency: Extra-vehicular Activity (EVA)**

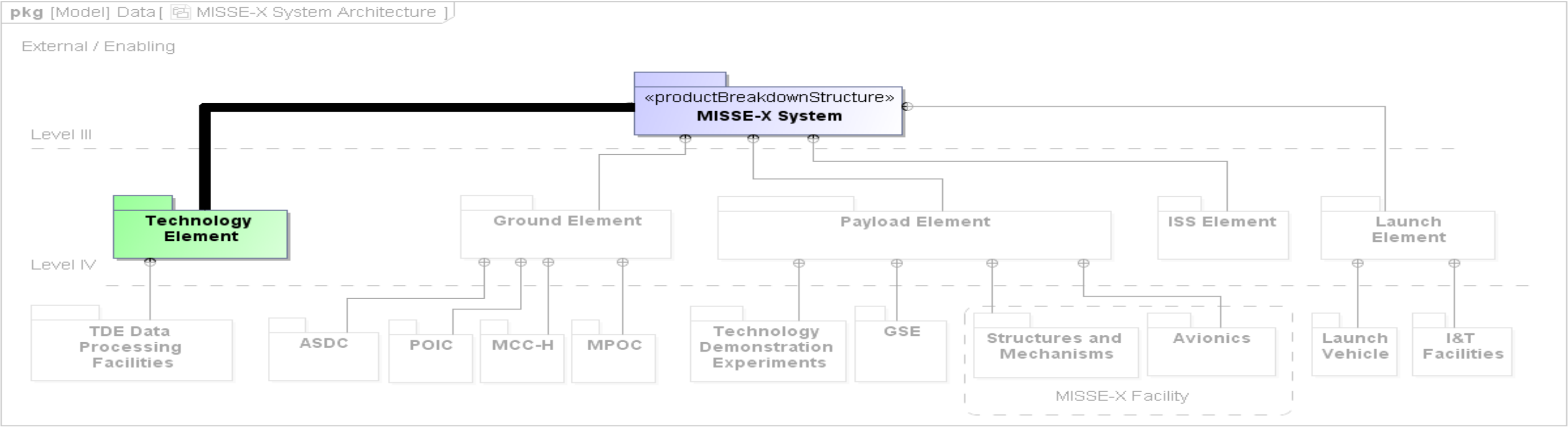
Step 1: Identify External cont…



Interfaces:

- **All data is received by the System from the Payload Operations Integration Center (POIC) in the Ground Element**
- **Communications between MISSE-X and MCC-H through POIC**
  - **Direct contact with MCC-H during robotic operations (installation/retrieval)**

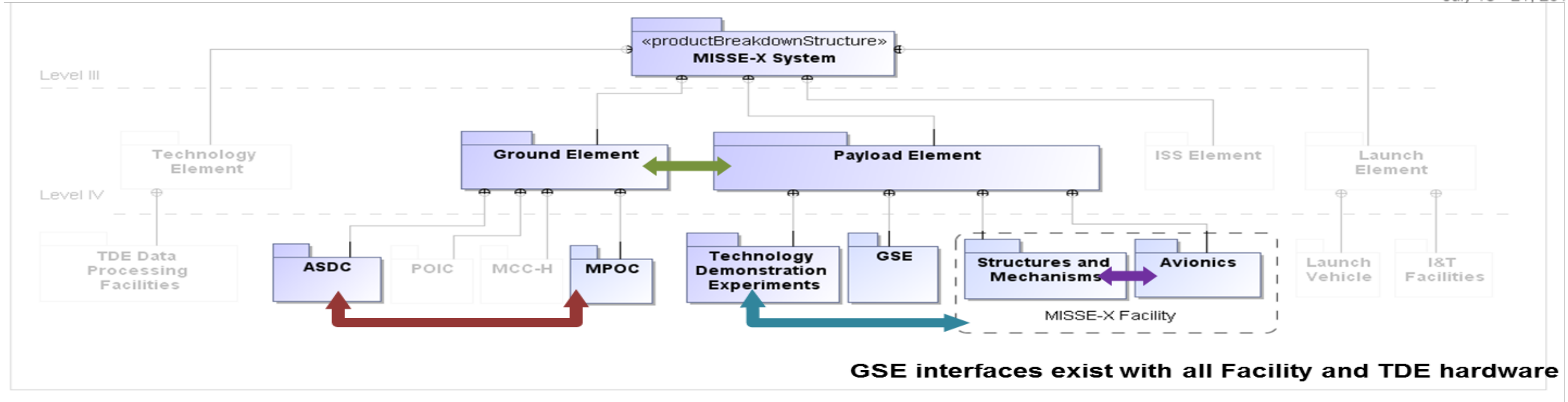
Step 1: Identify External cont…



Interfaces:

- TDE data being collected by the System is delivered to the Technology Element for processing and distribution
- Commands from PI are passed from the Technology Element to the System to control the TDEs

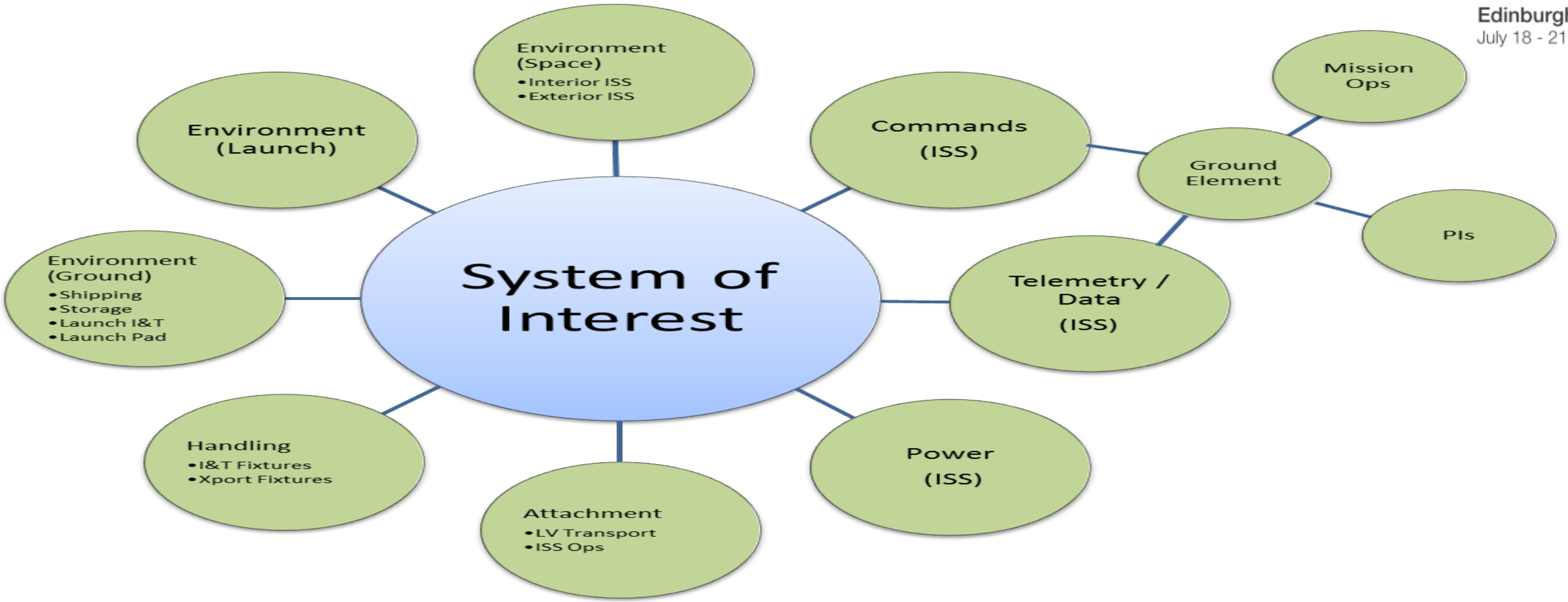
Step 1: Identify Internal Interfaces



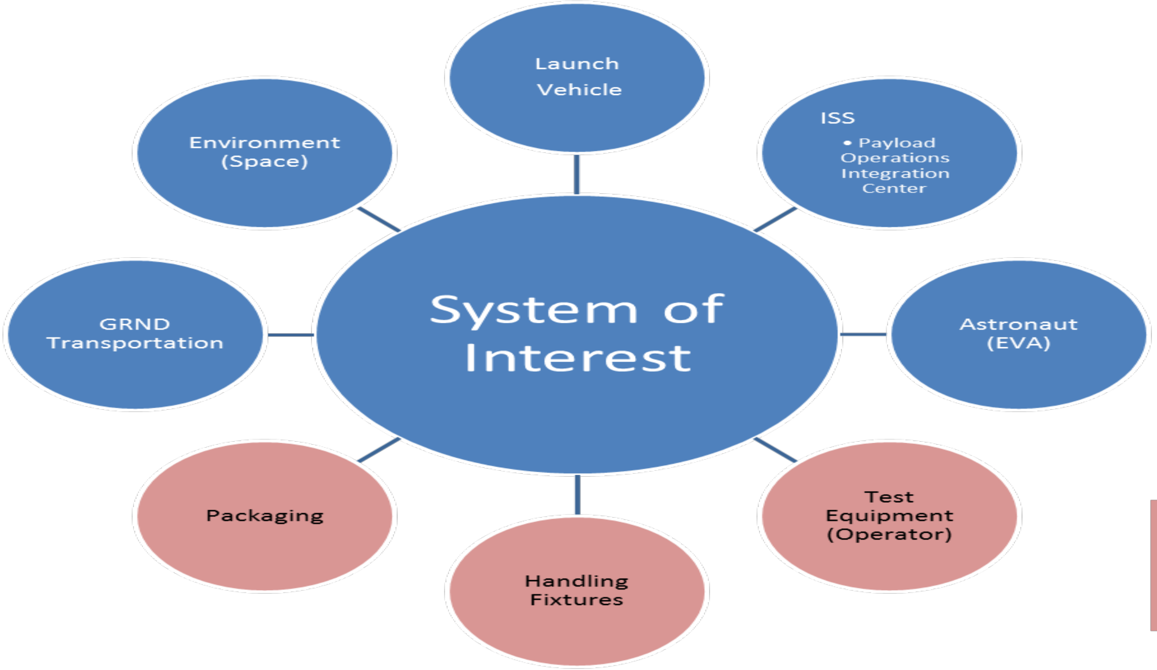
- Ground / Payload: Primarily through external interfaces
- ASDC / MPOC: Interface documented in an agreement
- TDE / Facility: Interface control & product requirements in Experiment Integration Agreements
- S&M / Avionics: Interface documented in configuration managed drawings

Functional Interfaces Identified

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

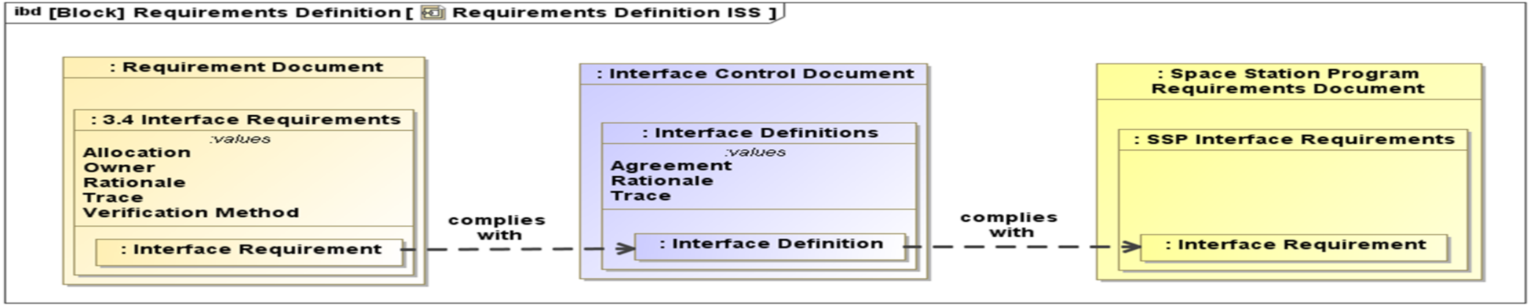


Physical Interfaces Identified

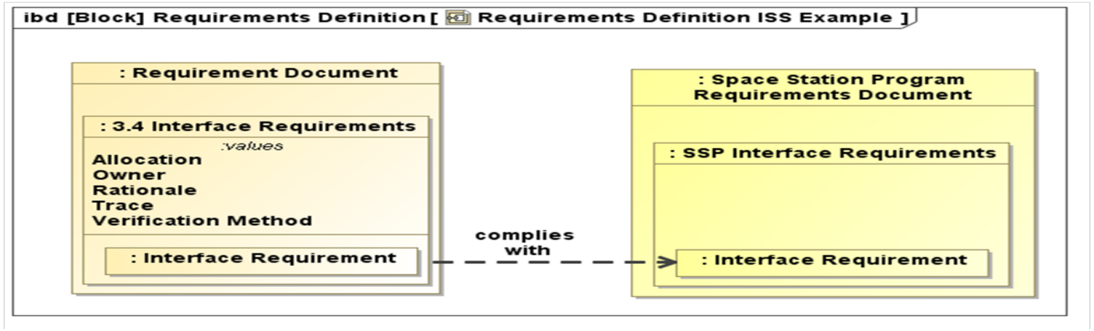


Could be part of System

Step 2: Capture Interfaces

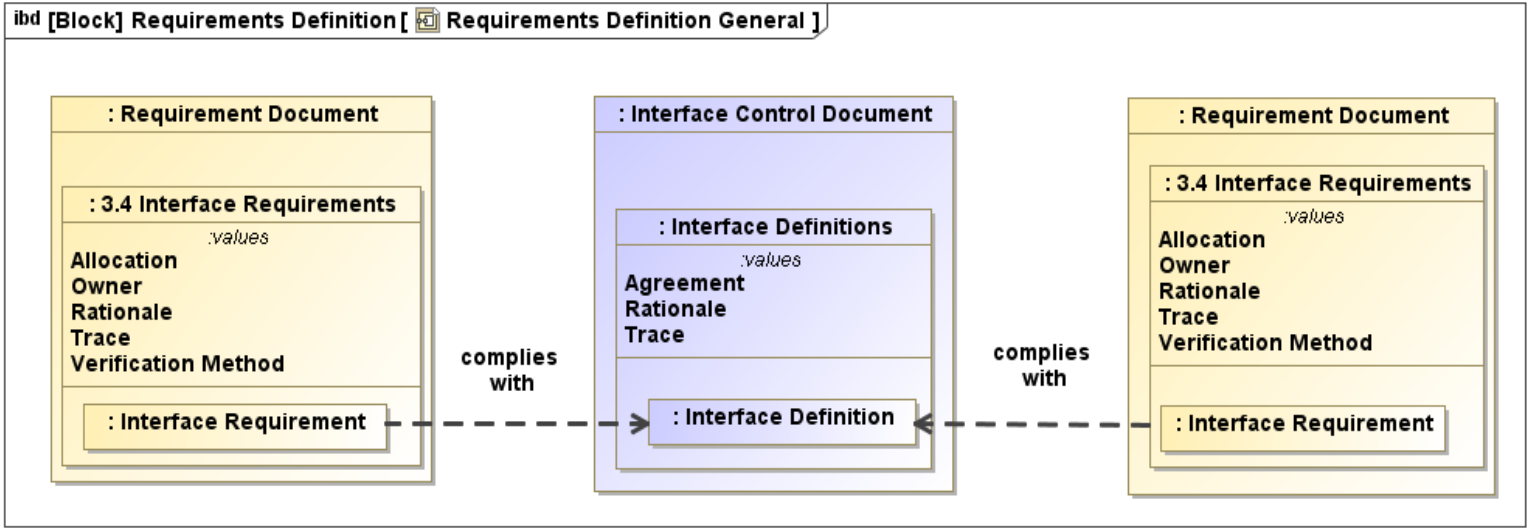


OR



Non-Existing ICD Development

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016



Tip: Add any interface agreements between external parties into Interface Control Documents. These become binding when they become a baseline which are rarely covered in contracts.

Capture into a Requirements Database

- [MXS-ISS-001] MXS ISS SSP57012 Structural/Mechanical Interfaces

The MXS shall meet requirements of SSP-57012, Section 3.1, Structural/Mechanical Requirements as deemed applicable by Table 4.2-X(MXS-SRD-TBR-008) of SSP-53XXX(MXS-SRD-TBR-007).

Rationale: *The ISS Program levies the launch requirements found in 57012 on all unpressurized, FRAM based payloads.*

Traceability: MXS.C.1

Verification Method: Inspection

- [MXS-ISS-002] MXS ISS SSP57012 Electrical Interfaces

The MXS shall meet requirements of SSP-57012, Section 3.2, Electrical Requirements as deemed applicable by Table 4.2-X (MXS-SRD-TBR-008) of SSP-53XXX(MXS-SRD-TBR-007).

Rationale: *The ISS Program levies the launch requirements found in 57012 on all unpressurized, FRAM based payloads.*

Traceability: MXS.C.1

Verification Method: Inspection

Tip: Group by how you want to verify your interfaces, usually by similar categories and by requirement owners

Step 3: Define

- Once the interface requirements are captured and under configuration control, you can begin defining the detailed interface definitions
- There are no shall statements in interface definitions
  - Reason: It is an end item spec and becomes a statement of fact
- Interface definitions become the success criteria for interface verification activities
- Can be more than one interface definition per interface requirement
- Can be shared at multiple levels

Tip: Add rationales to justify the interface definitions. This improves record management, provides how the definition was reached, and leaves a good audit trail.

Interface Requirement vs. Interface Definition

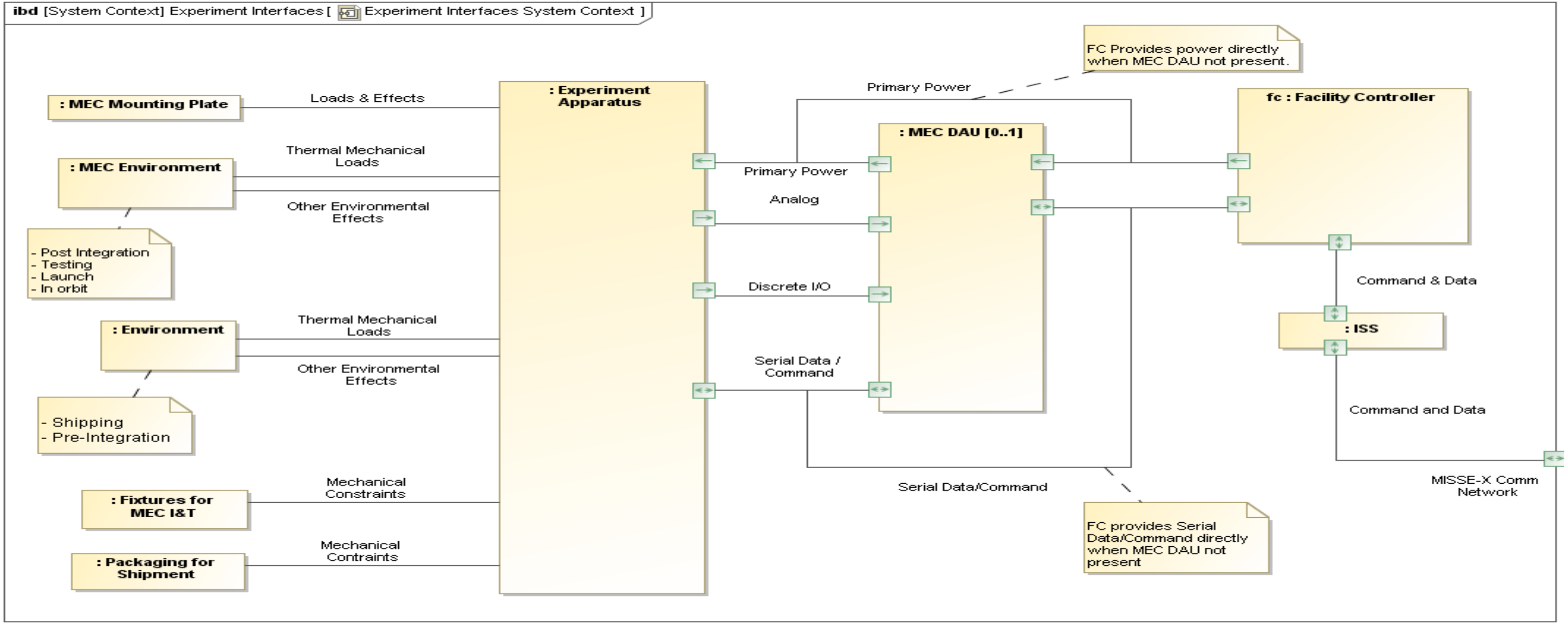
System Requirements Document contains interface requirements

- Sys 2 shall obtain power from Sys 1 per the connections defined in ICD 2345 Drawing 3-4
- Sys 2 shall operate on power obtained from Sys 1 having the characteristics defined in ICD 2345 Table 3.6

Interface Control Document defines the interface (interface definition)

- Drawing 3-4 contains the connector, pin assignments, and grounding information in order to obtain power
- Table 3-6 contains power characteristics such as voltage, current, noise, filtering

This Internal block diagram example showing some of the interface flow with experiment apparatus. This diagram helps define and communicate the resources, constraints, and environment that will affect experiments on MISSE-X.



This activity diagram example shows the major functions of the MISSE-X Facility Controller, a part of the facility avionics system.

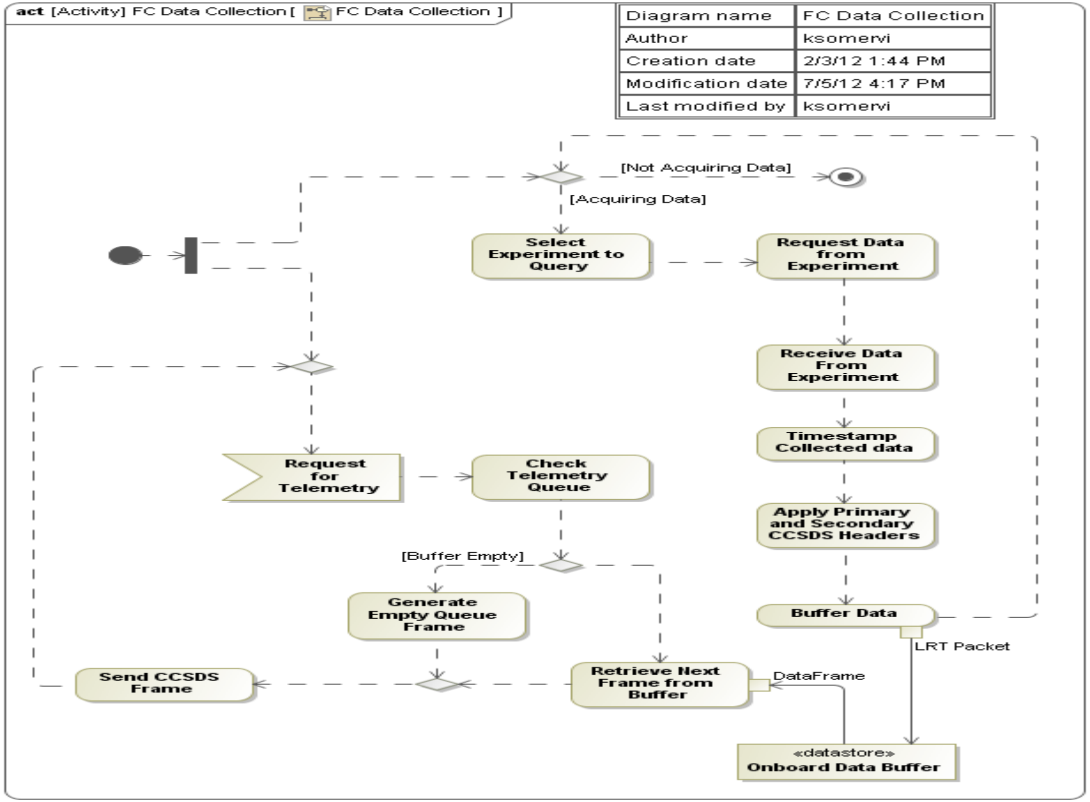


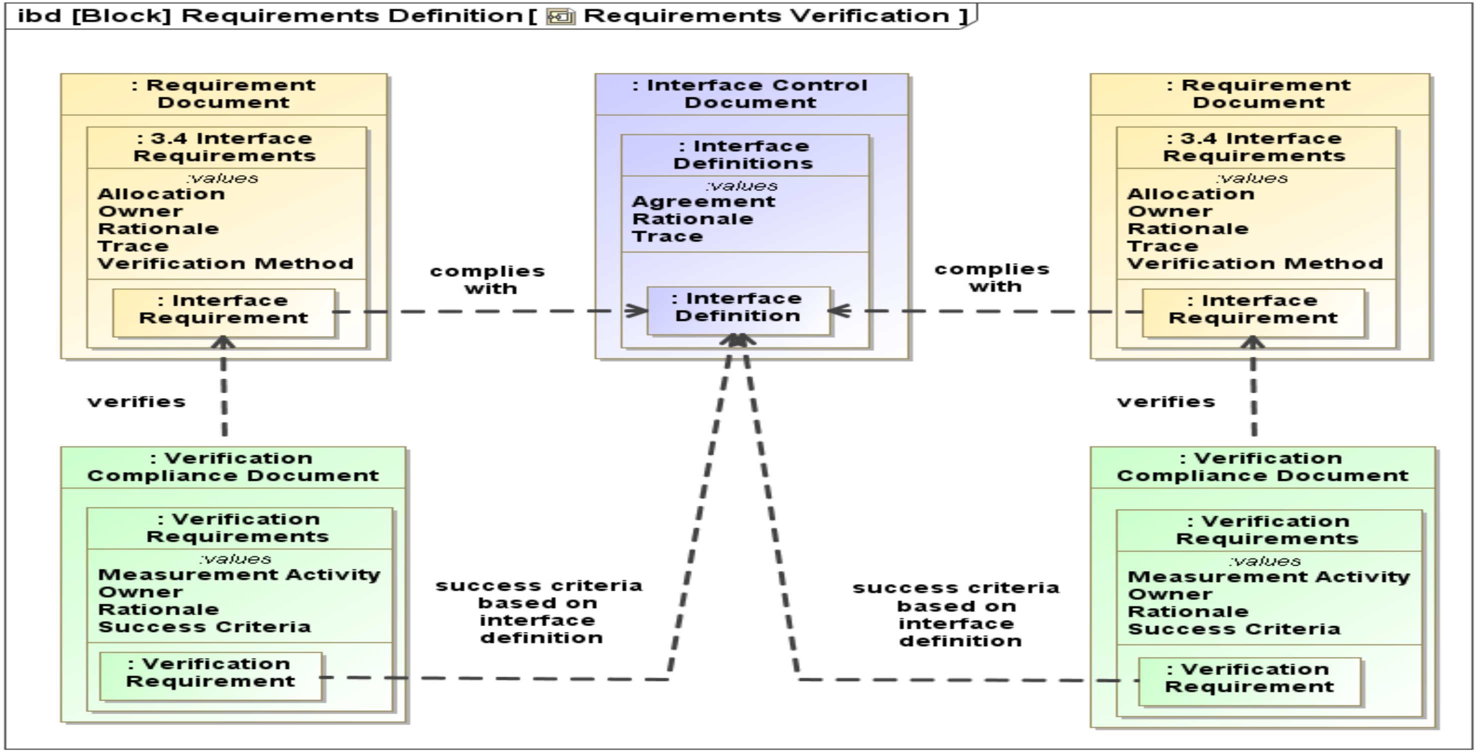
Diagram name	FC Data Collection
Author	ksomervi
Creation date	2/3/12 1:44 PM
Modification date	7/5/12 4:17 PM
Last modified by	ksomervi

Step 4: Allocate

- If there is further decomposition needed for interfaces, then allocate to the next lower level and capture the interface requirements in the requirements document at that level
- Continue until you are at a level that is not a composite of other component interfaces

Note: Interface requirements can share or point to same interface definitions even though they are at different levels.

Step 5: Verification



Step 6: Compliance

- Once the verifications are complete, the process for compliance closeout can begin
- Within the Req Tool, a compliance report is generated by the requirement owner
- The requirement owner starts the electronic signature approval process
- On completion of compliance, it is recorded and placed under configuration control

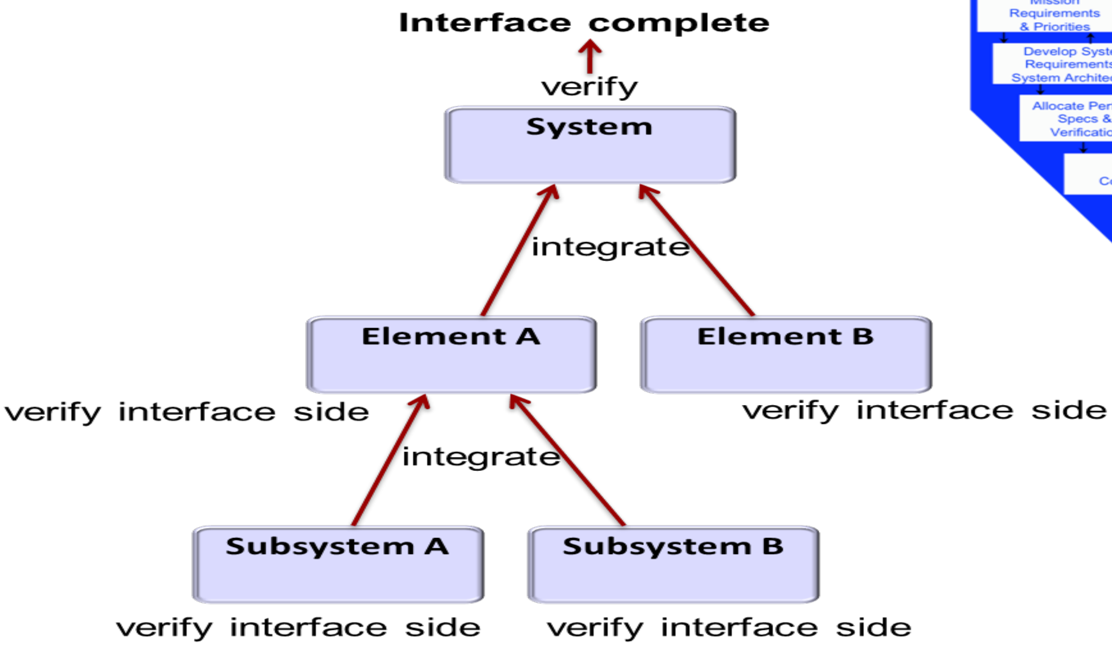
Tip: It is not necessary for each side to complete their interface verification to begin compliance.

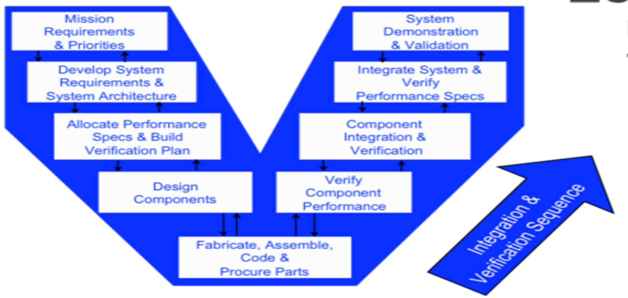
Step 7: Integrate

- Once each side of the interfaces are closed out at a lower level, it can move up for integration and *checkout* at the next level up
- The Requirement Owner at the next higher level can now begin verification of integrated interfaces to meet compliance at the that level
- Process repeats steps five through seven until the system is complete

Tip: Each side can verify without the other side, integrated sides are checked out at the next level up

Step 7: Integrate





Further SE support

- Systems Engineering Technical Excellence Community
  https://sites.larc.nasa.gov/setec/
- For a copy of a Requirements Guide LMS-CP-5526 send a request to kevin.g.vipavetz@nasa.gov or james.e.price@nasa.gov

Questions?