Everything You Wanted To Know About Interfaces But Were Afraid To Ask!

SYSE 602 Systems Requirements Engineering

18 November 2021

Lou Wheatcraft, Senior Consultant, Wheatland Consulting
Wheatland.consulting@gmail.com

INCOSE

Topics to be Covered

- Managing the interfaces across the system lifecycle
- What an interface is and what it is not
- Three step process
  - Step 1: Identify interface boundaries and interactions across those boundariues
  - Step 2: Define the interactions
  - Step 3: Define interface requirements
- Common interface issues
- Doing an interface audit
- Best practices when managing interfaces

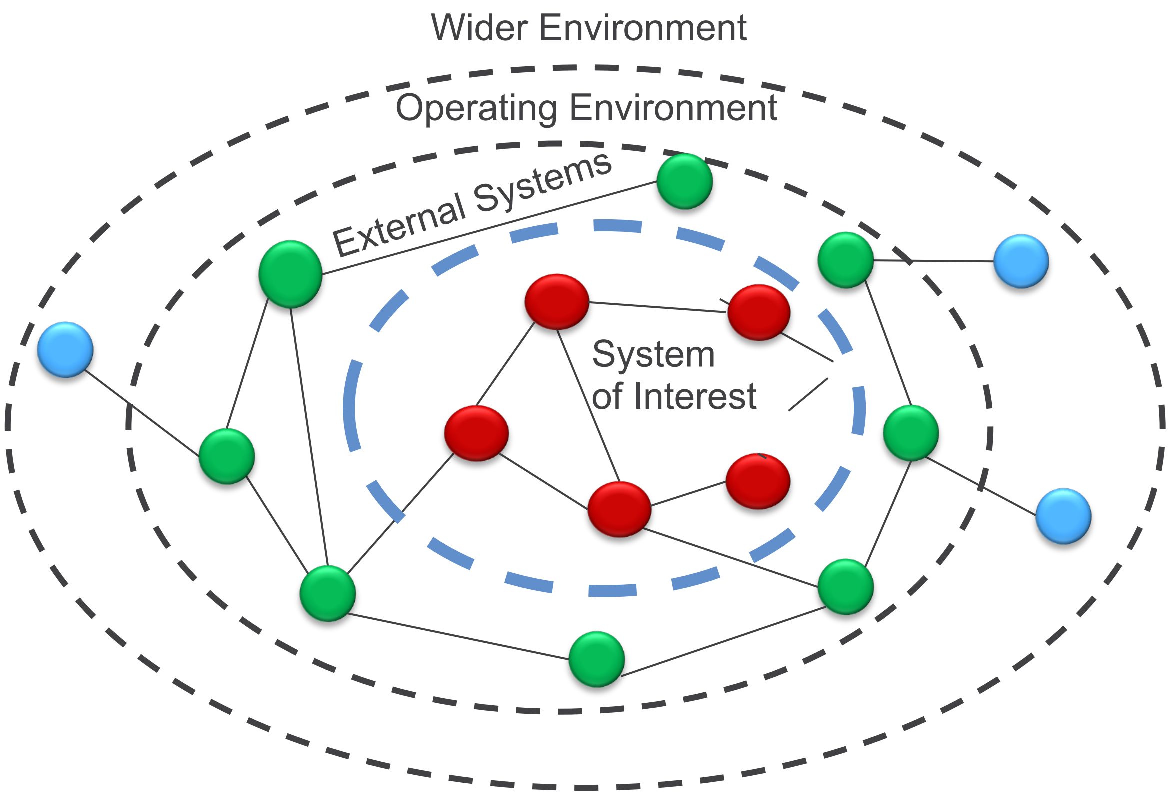
Managing Interfaces Across the System Lifecycle

A System is:

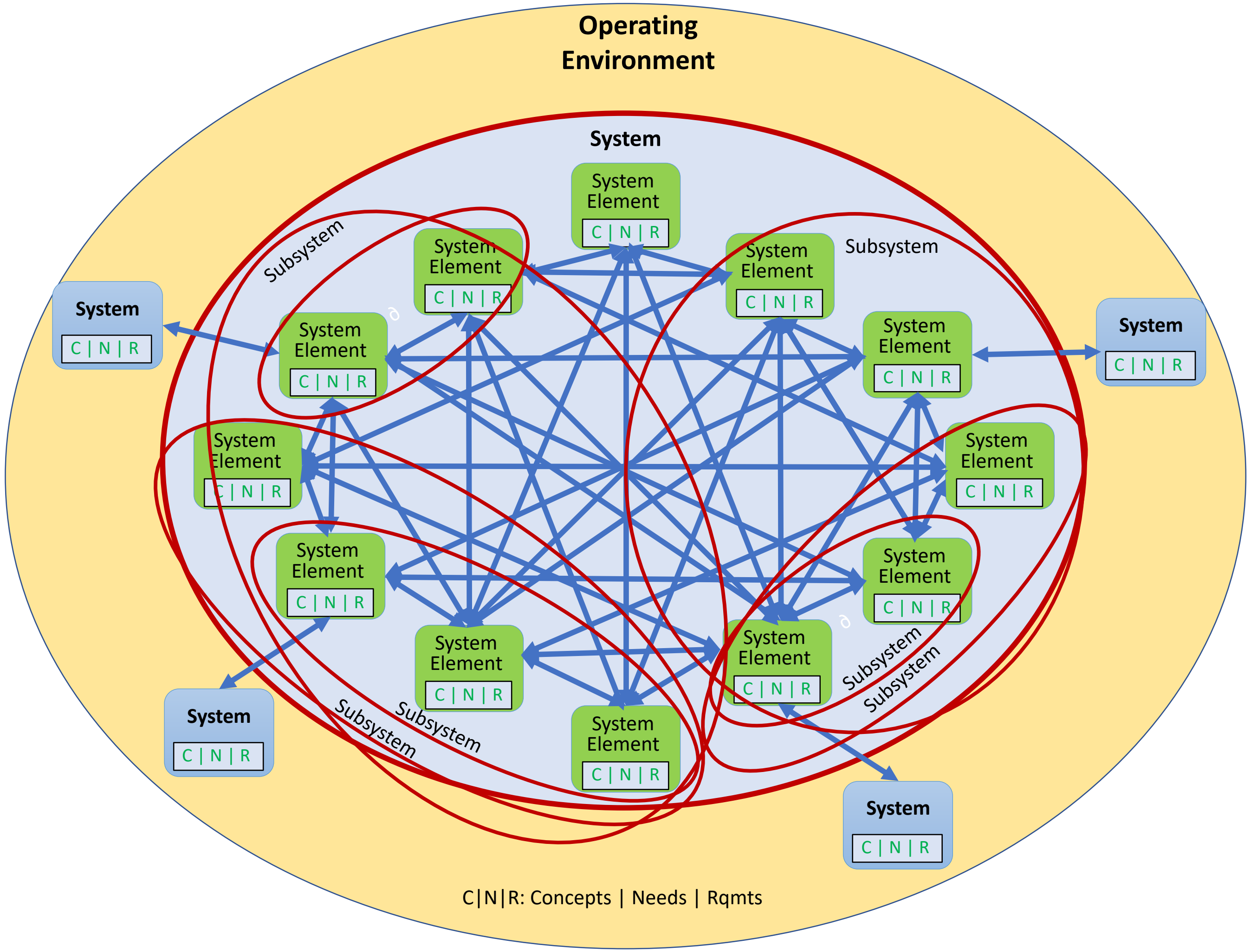
- ✓ is comprised of internal *parts* with <u>*interconnections and interactions*</u> between those parts,

- ✓ is contained within an <u>external *boundary*,</u>

- ✓ <u>with *interfaces*</u> (inputs/outputs) across that boundary to <u>*external systems*</u> that exist in an <u>*operating environment.*</u>



Complexity is a Function of the Number of Interactions Between System Elements



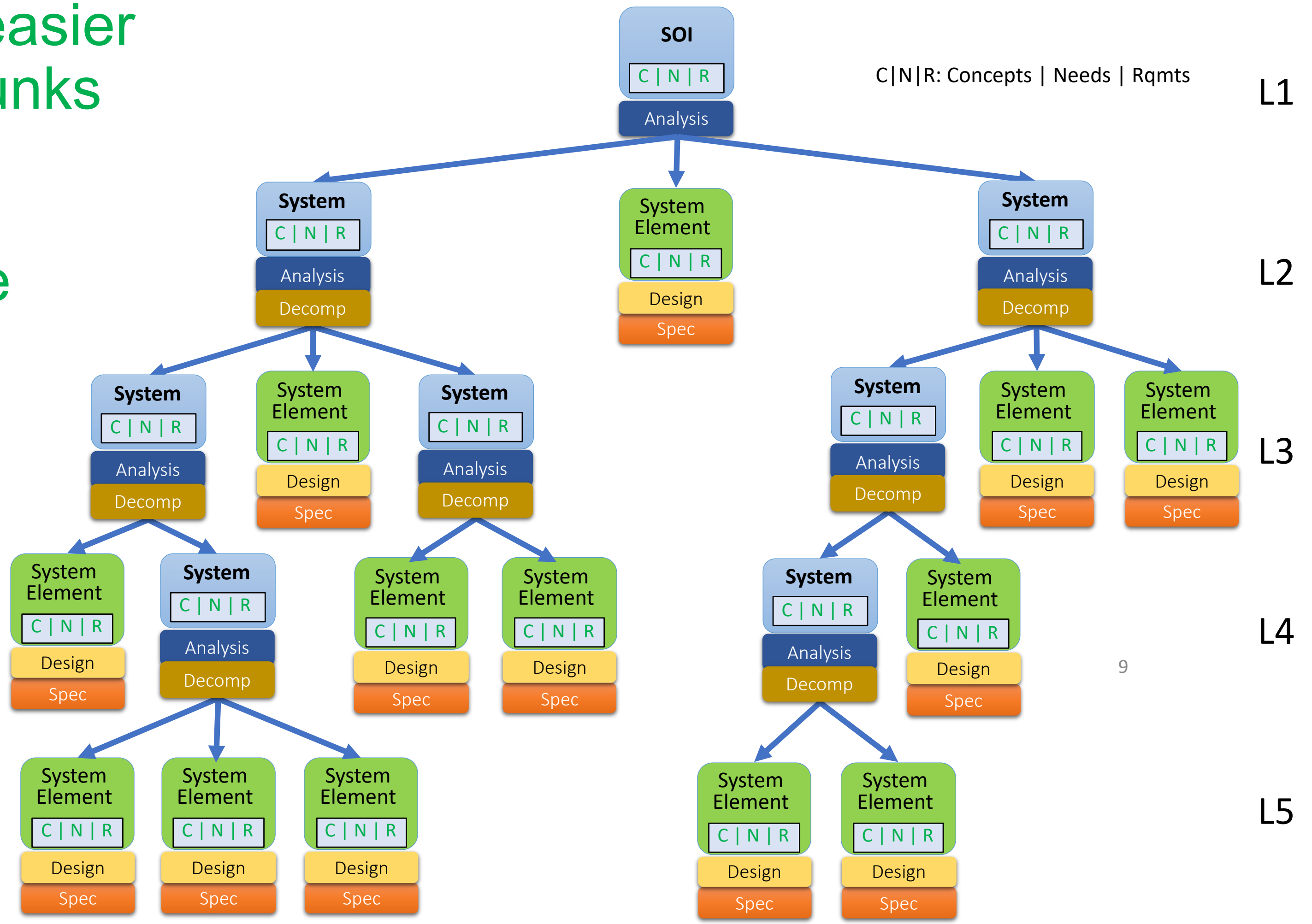
The behavior of a system is a function of **interactions** between the subsystems & system elements that make up the system, as well as **interactions** with external systems & the operational environment.

Managing interfaces within the system architecture

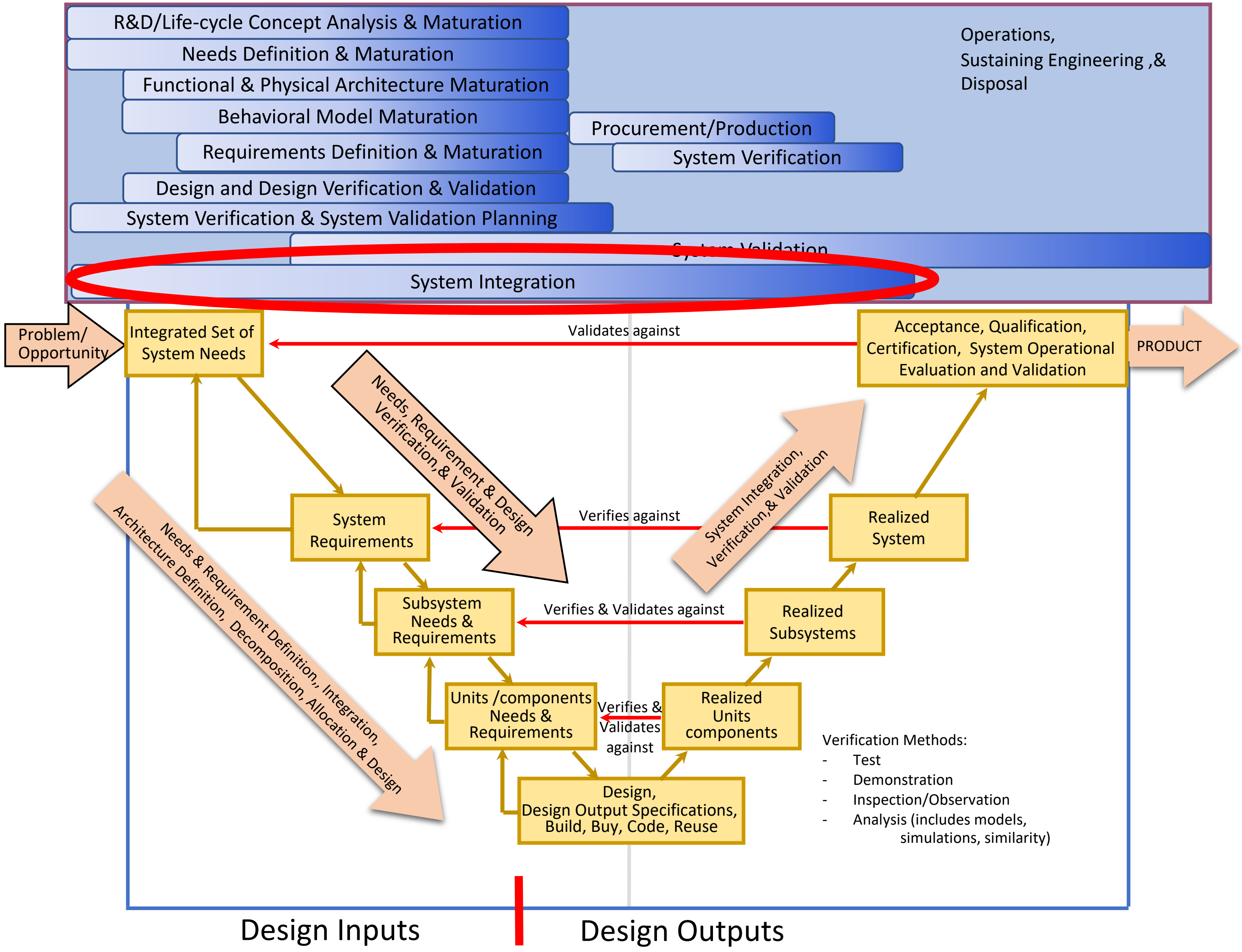
Focus on decomposition makes it easier to develop the SOI in bite sized chunks across multiple organizational units (internal and external) based on specialize knowledge and expertise

Interactions (Interfaces) not shown in this view

A major challenge is managing both internal and external interfaces when parts of the architecture are developed by different organizations both internally and externally



System Integration Across the Lifecycle



The integrated SOI must be managed from the beginning of the project across all lifecycle stages

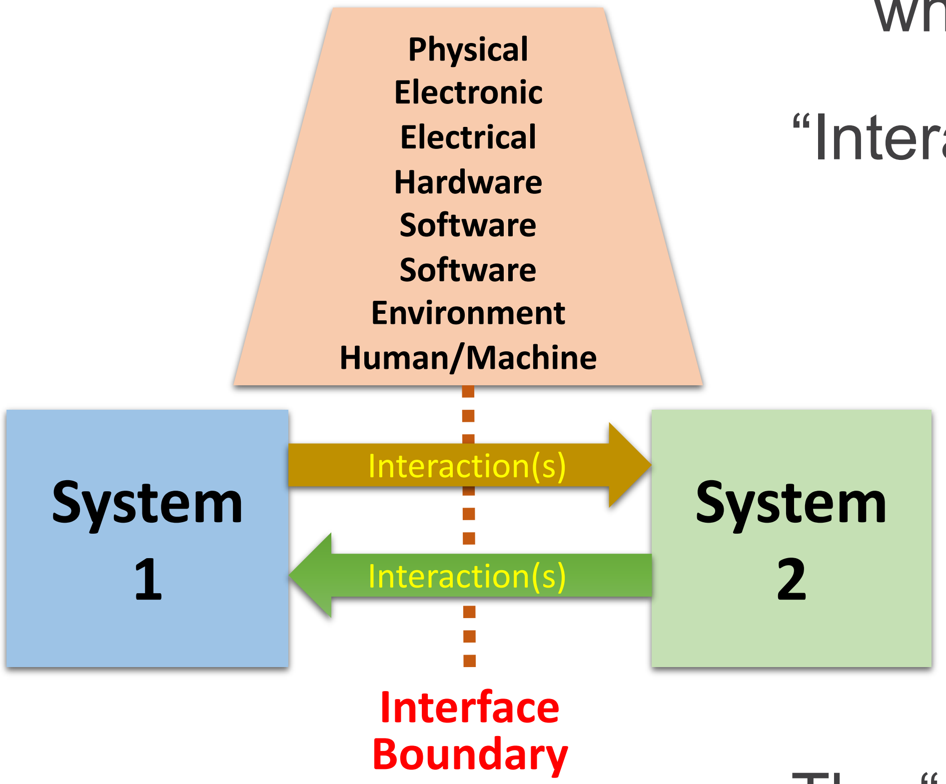
The major focus must be the interactions and interfaces throughout the lifecycle

The customer/supplier relationship must allow management of the interfaces

Adapted from Ryan, M. J.; Wheatcraft, L.S., “On the Use of the Terms Verification and Validation”, February 2017 and INCOSE SE HB, Version 4, Figures 4.15 & 4.19

What is an interface?

“An interface is a **boundary** where, or across which, two or more systems **interact.**”



“Interact” involves a function verb/object pair.

The “function verb” indicates some action concerning receiving an input or supplying an output

The “object” is that actual “thing” that is crossing the interface boundary.

The “performance” aspect is included in a definition of the characteristics of the interaction, often contained in an interface definition or control type document

What an interface is not.

- A general rule is that the word “interface” should not be used in a requirement statement either as a noun or a verb.
  - As a **noun**, it implies the interface is a thing, which it is not – it is a boundary across which, or at, two systems interact.
  - As a **verb**, it is ambiguous, in that often there are multiple interactions between systems across a single interface boundary.
- It is a best practice to focus on individual interactions when writing interface requirements.
  - This is important from both a system verification perspective and an allocation perspective.

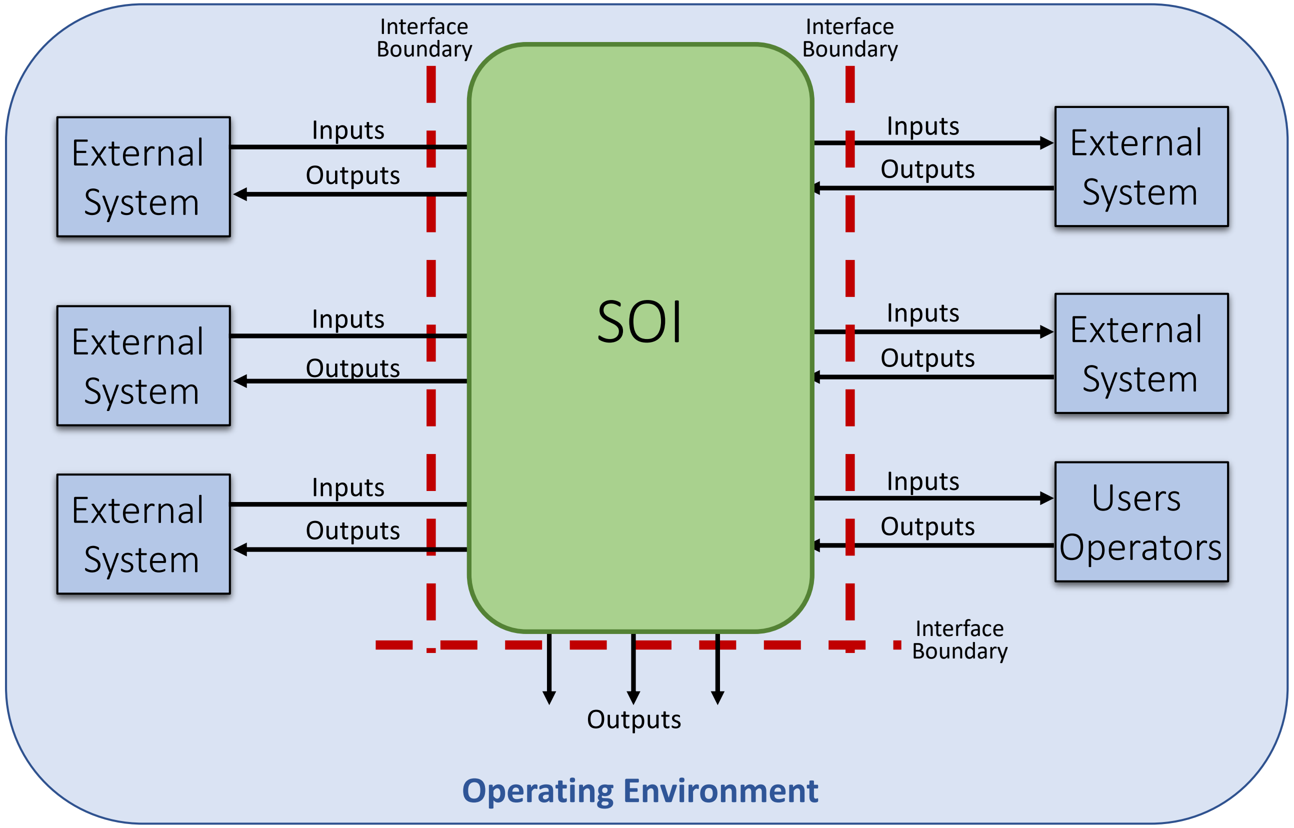
Examples on how **not** to write an interface requirement:

- *The digital **interface** shall maintain full operational capability after two failures*.
  - This requirement assumes the interface is a system and has functionality – this is not true.
- *The **interfaces** between the spacecraft and payload shall be designed to …*
  - This is a requirement on the designers and also assumes the interfaces are things. The requirement should be on accessibility of connectors, bolts, etc.
- The **interfaces** between the spacecraft and payload shall have standard labels, controls, and displays.
  - This requirement again assumes the interface is a thing. Also where is "standard" defined?
- *The electrical **interface** between the spacecraft and payload shall have a reliability of 0.99999.*
  - This requirement again assumes the interface is a thing. The requirement is on each of the systems and apply to any hardware or software of the system involved in interfacing with another system.
- *"The SOI shall interface with ……………"*
  - This requirement is ambiguous because it does not focus on a specific interaction.

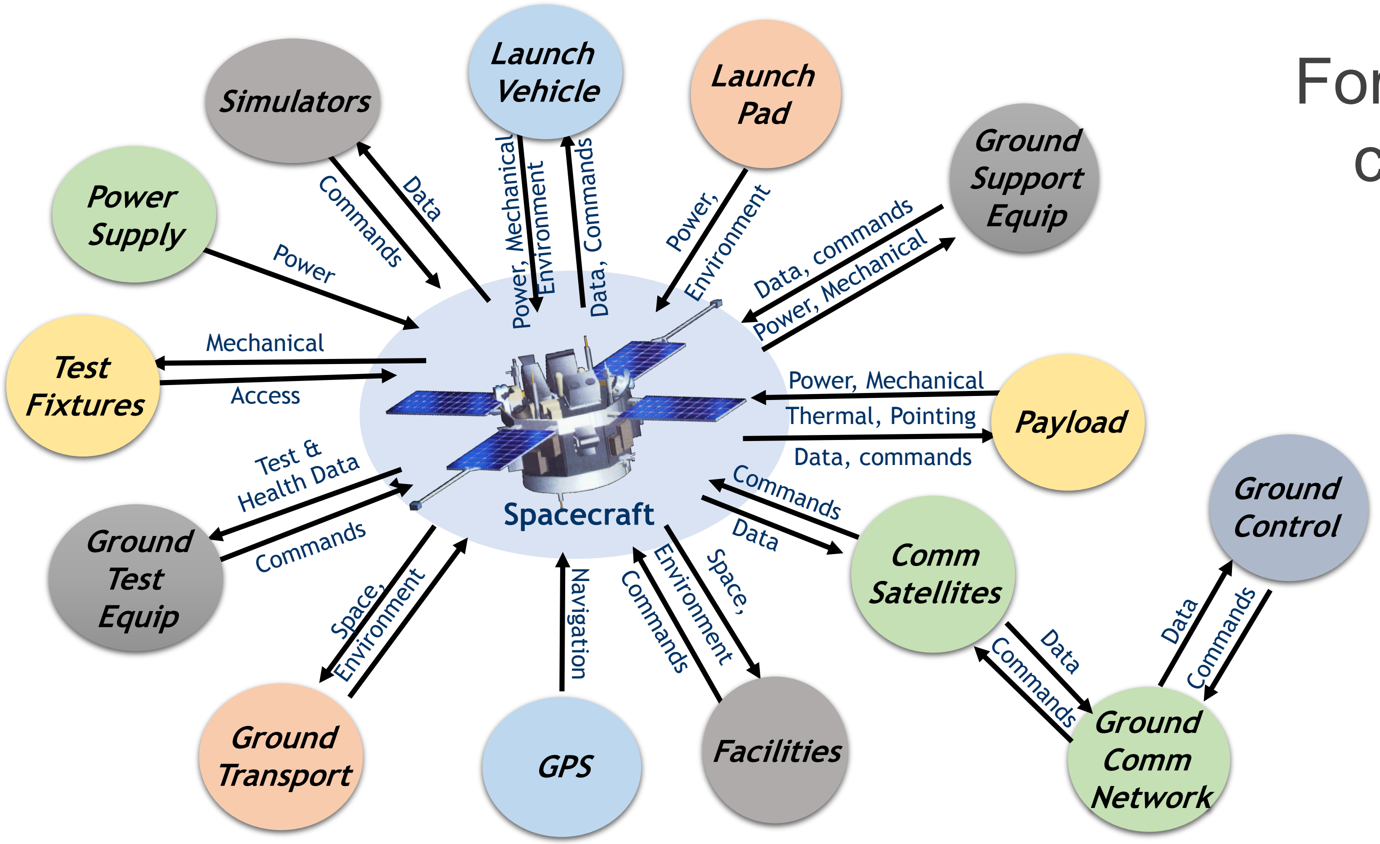
Three step process:

- Step 1: Identify the interface boundaries and interactions across those boundaries
- Step 2: Define the interactions across the interface boundaries
- Step 3: Write the interface requirements.

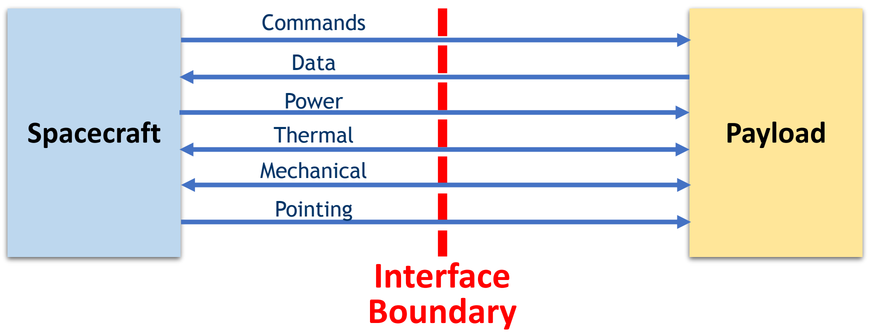
Step 1: Identify the Interface Boundaries and Interactions Across the Boundaries



Step 1: Identify the Interface Boundaries and Interactions Across the Boundaries



For each of the 14 interface boundaries create and interface block diagram



Identify each individual interaction across each boundary

Enabling Systems

- Enabling systems are systems external to the SOI needed to “enable” certain lifecycle activities or enable the SOI to operate.
- Enabling systems include lifecycle support systems and services across the SOI lifecycle, e.g., development, production, integration, system verification, system validation, deployment, training, operations, maintenance, and disposal.
  - This could consist of laboratories, test equipment, test fixtures, power supplies, clean rooms, transportation, storage, integration facilities, to name a few.
- Enabling systems are often “owned” by external organizations
  - Must budget for their use
  - Must address their use in the project master schedule
  - Must address whether they can be used as is or need to be modified to meet the project needs.
- Enabling systems are a major source of interfaces
- Enabling systems are a major source of constraints

External Interface Analysis

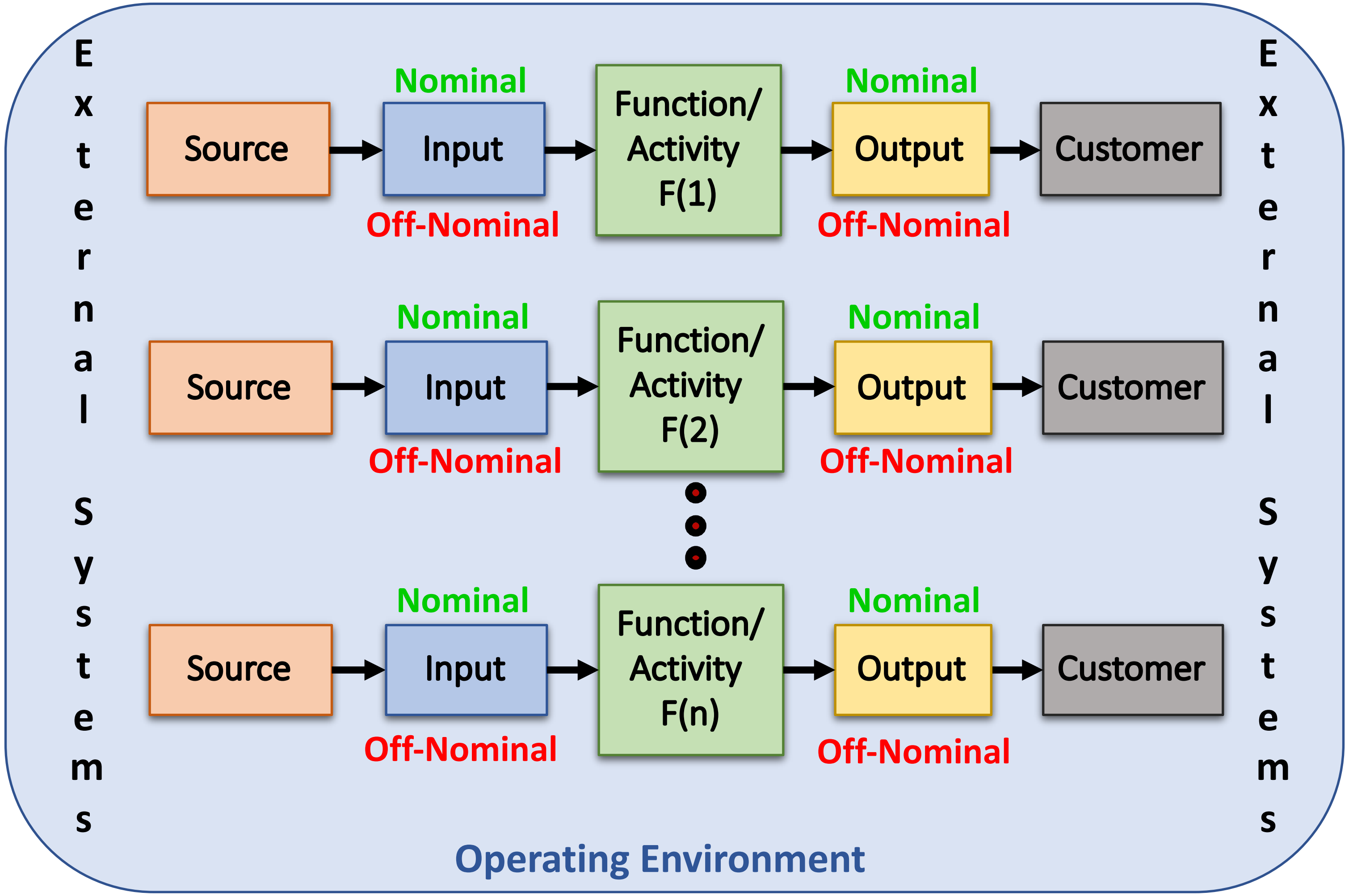
- Which external systems are existing vs. which are being developed concurrently with the SOI?
- Who are the stakeholders for the external systems, and have they been involved in the elicitation activities?
- Have all the specific interactions (inputs and outputs) between the SOI and each external system been defined?
- From a threat perspective, have the risks been assessed concerning bad things that could happen across this interface in either direction?
- Is an existing external system likely to change how it interacts with the SOI across the interface boundary during the development or after the SOI is in use? How will you know if it does change?
- For existing systems, is the documentation (e.g., Interface Control Documents (ICDs)) that defines the interface boundary and interactions across that boundary available and current? If not, how will this information be obtained?
- For new systems being developed concurrently, what is the process to be followed to document and agree on the specific interactions. Who is responsible for recording the interactions and getting approval? Who will have configuration control of those definitions? What is the schedule for doing so?
- For software systems, what standard, application programmer's interface (API), etc. apply to the interactions?

Risk Analysis

Always ask: “What is the worse thing that can happen across the interface boundary?

It is a best practice to do an interface FMEA (IFMEA) for each interaction.

Define and manage each identified risk



Step 2: Define the Interactions Across each Interface Boundary

- To completely define an interaction at or across an interface boundary, three things must be addressed:
  - *The characteristics of the thing crossing the interface boundary.*
  - *The characteristics of each system at the interface boundary.*
  - *The media involved in the interaction.*
- The characteristics of the thing crossing the interface boundary are important design inputs as they communicate the information the designers need to know to realize the interactions.
- The characteristics of each system at the interface boundary and media are defined as design outputs.

Step 2: Define Interactions Across each Interface Boundary

- In some cases, there are standards defined for both the media and the characteristics of the systems at the interface boundary for example, Ethernet, USB, Networks (IEEE 802.xx), wiring, pressure lines, etc.
  - These standards are important from safety, security, and interoperability perspectives.
  - Standards associated with interfaces need to be invoked within the set of design input requirements.
  - These standards both constrain and drive the design.
- For developing systems, not all this information will be known until the design is complete.
  - In these cases, the definitions evolve with the design
  - This is a <u>major challenge</u> for parts of the architecture that are contracted out to a supplier
  - The SOW or SA must clearly state how the supplier will address the evolving nature of the interface definitions.

The general format of an interface definition statements

- [Thing being defined] is[are] [whatever the definition is]. or
- [Thing being defined] is as shown in [Drawing xxxxxx] or [Figure yyyyyy].
- [Thing being defined] has the characteristics shown/defined in [Table or Graph zzzzzzzz.]

- Examples:
  - The DC voltage [*supplied by System 1*] has the characteristics shown in Table xyz or Figure 123.
  - The mechanical attach points [*between System 1 and System 2*] are as shown in Drawing xyz.
  - The fluid [*supplied by System 1*] has the characteristics defined in Table xyz (pressure, quality, flow rate, temperature, etc.)
  - The leak rate at the connection [*between System 1 and System 2*] is less than xxx units per time period.
  - The commands [*sent by System 1*] are defined in table xyz.
  - The data stream [*accepted from System 2*] has the characteristics defined in ……….
  - The data parameters used within the SOI are defined in Data Dictionary xxxxx
  - System 1 printer port complies with USB 3.0 standard.

No Shall Statements – just statements of agreed to facts!!!

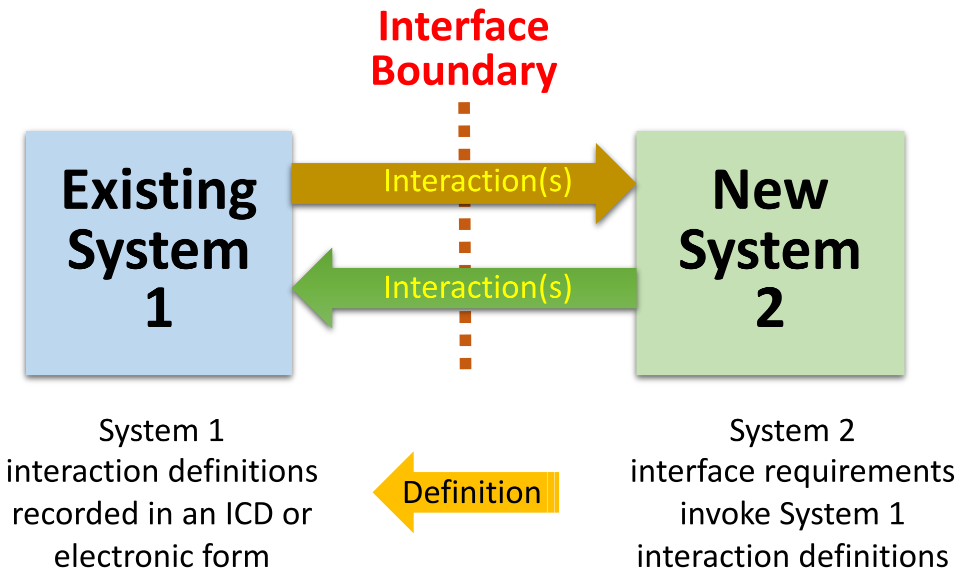
Step 3: Write the Interface Requirements

- An interface requirement is a design input requirement that involves a defined interaction across an interface boundary with another system.
- The format includes
  - a function/verb indicating directionality (input/output; supplier/receiver),
  - the name of the object involved in the interaction,
  - AND a reference (pointer) to the specific location where the definition of the specific interaction across the interface boundary is located.
- All interface requirements have the same general form: (if the interaction is in response to some trigger event, then that trigger event would be included in the requirement text.)
  - *[My System] shall [interact (function verb/object) with] [Another System] as defined in [location where the interaction is defined].”*
  - *[My System] shall [use/provide from/to] [Another System] [something] having the characteristics defined in [location where the something is defined].”*
- As discussed earlier, the word “interface” is not included in an interface requirement as a noun or a verb.

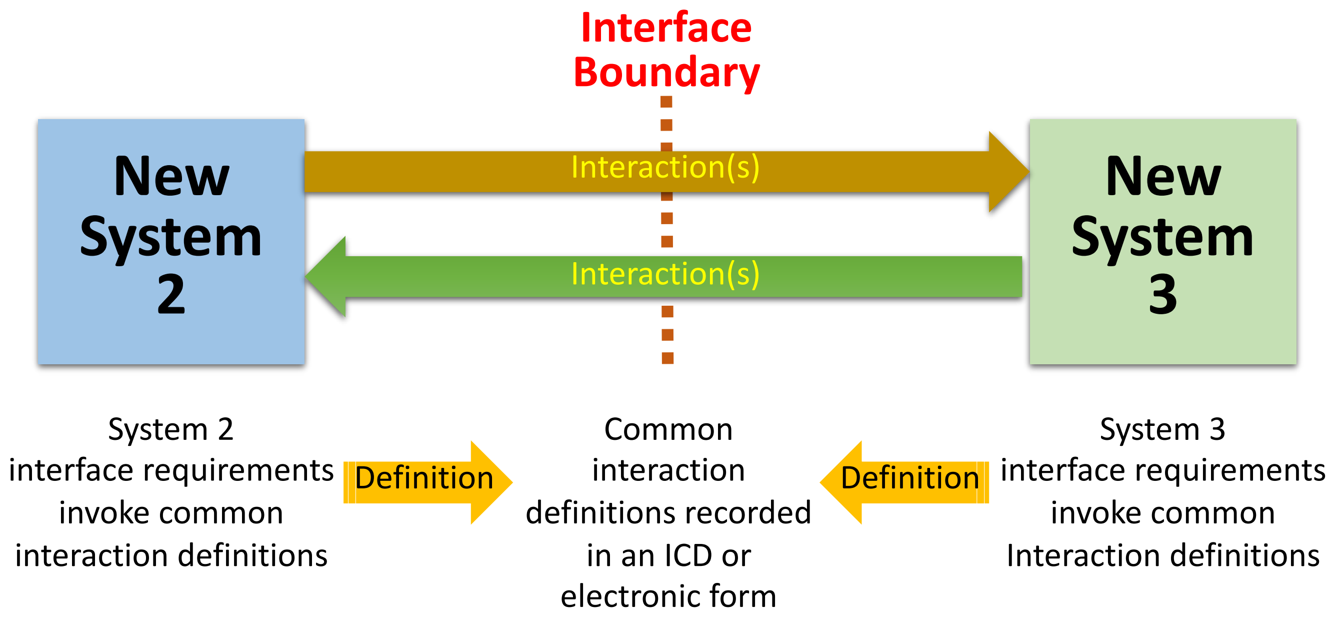
Examples for a payload/spacecraft:

- The Payload shall communicate with the Spacecraft processor via a 1553 bus as shown in Spacecraft ICD 1234, Figure 6. *[Specifies the media used for communicating payload sensor data to the Spacecraft.]*
- The Payload shall send Sensor A data to the Spacecraft having the characteristics defined in Spacecraft ICD 1234, Table 5.4. [*Specifies the format the payload needs to send the Sensor A data to the spacecraft over the 1553 bus.]*
- The Payload shall receive from a [ground power supply] ground power having the characteristics described in Spacecraft ICD 1234, Table 4.2. [Specifies the characteristics of power used during development and testing. The characteristics are the same as that provided by the Spacecraft during operations.]
- The Payload shall receive from the Spacecraft, 28-volt power having the characteristics described in Spacecraft ICD 1234, Table 4.2. [*Specifies the characteristics of the power the spacecraft is providing to the payload.*]
- The Payload shall receive 28-volt power from the Spacecraft per the connections defined in Spacecraft ICD 1234, Drawing 2-5. [Specifies the characteristics of the physical connections the Payload needs in order to connect to the Spacecraft 28-volt payload power bus.]
- The Payload shall receive from the Spacecraft, pointing data having characteristics defined in Spacecraft ICD 1234, Table 7.2. [*Specifies the characteristics and format of the pointing data supplied by the Spacecraft to the Payload.]*
- The Payload shall mechanically attach to the Spacecraft per the mechanical connections shown in Spacecraft ICD 1234, Drawing 5-5. [*Specifies the mechanical connections, bolt holes, bolt sizes, torque values, etc. to be used to mechanical attach the Payload to the Spacecraft.]*

Existing vs New System Development



New System 2 interfacing With an Existing System 1



New System 2 and System 3 interfacing With Each Other

Interface requirements when both systems are being developed concurrently

- If two systems are being developed concurrently, each system will need to include in their set of design input requirements an interface requirement for each of its interactions with the other system.
- Because of this, interface requirements for these systems are developed in pairs.
  - *"[System 2 shall [interact (function verb/object)] with [System 3] [as defined in or having the characteristics shown in] [common location where the interaction is defined]."*
  - *"[System 3] shall [interact (function verb/object)] with [System 2] [as defined in or having the characteristics shown in] [common location where the interaction is defined]."*
- For the Spacecraft/Payload example, for each of the "The Payload shall……" interface requirements there would be a corresponding "The Spacecraft shall……" interface requirement.

Common defects and issues concerning interface requirements

- Writing interface requirement statements that include the word “interface” either as a noun or verb.
- Interface requirements not written in the form of an interface requirement.
  - The most common defect is a failure to include a pointer within the requirement statement to where the interaction is defined and agreed to.
- Failing to identify all external systems in which the SOI interacts with across an interface boundary.
- Identifying an interface boundary with another system, but not addressing all interactions across that boundary within the set of design input requirements.
- Failing to define, agree to, and configuration manage all interactions across an interface boundary.
- Failure to include all interactions (functional and physical) with external systems in the system model and assess overall system behavior as a function of the interactions with external systems.
- Failure to include all interactions (functional and physical) within the system model and assess overall system behavior as a function of the interactions of the subsystems and system elements that make up the SOI.

Common defects and issues concerning interface requirements

- Failure to verify a system or system element meets all its interface requirements prior to integration into the macro system it is a part.
- Assuming that design verification of all functional interactions across system boundaries using a model of the system is an adequate substitute for verification of the actual physical system against its interface requirements.
- Failure to do a threat and failure analysis for each interaction.
  - What could go wrong?
  - What happens if an input or output are not as defined and agreed to?
  - For security critical interactions, how could an unintended user interact across the system boundary such that the intended use of the system is compromised?
  - How does the SOI address these issues?
- From an integrated system behavior, another issue is cascading failures.
  - A non-nominal input results in a function to produce an off-nominal output which is then an off-nominal input to another function, and so on. In today's software intensive systems, cascading failures across interface boundaries must be addressed.

Interface Requirement Audit

SOI Interface Requirements	SOI Interaction Definitions	External System Interaction Definition	External System	External Systems Interface Requirements
SOI IR 1	Def 111	Def 111	ES1	ES1 IR 1
SOI IR 2	**Def 112**	**Def 123**	ES1	ES1 IR 2
SOI IR 3	Def 113		ES1	**Missing**
SOI IR 4	**TBD**	**TBR**	ES1	ES1 IR 4
SOI IR 5	**Missing**	Def 134	ES2	ES2 IR 1
Missing		Def 135	ES2	ES2 IR 2
SOI IR 6	Def 114	Def 114	ES2	ES2 IR 3
SOI IR 7	Def 115		ES2	**Missing**
SOI IR 8	**Def 116**	**Def 128**	ES3	ES3 IR 1
SOI IR 9	Def 117		ES3	**Missing**
SOI IR 10	**Missing**	Def 131	ES3	ES3 IR 1
SOI IR 11	**TBD**	**TBR**	ES3	ES3 IR 3
Missing		Def 131	ES4	ES4 IR 1
SOI IR 12	Def 138	Def 138	ES4	ES4 IR 2
SOI IR 13	Def 119		ES4	**Missing**
Missing		Def 135	ES4	ES4 IR 3

Best Practices Associated with Managing Interfaces

- Define how the project team will manage interfaces in their SEMP and NRDM Plan.
- Assign responsibility for the identification of all interface boundaries, definition of the interactions across those boundaries, and management of artifacts associated with interface management
- Involve all interface stakeholders
- Include all interactions within the SOI's set of system requirements.
- Ensure all interactions have been defined appropriate for the lifecycle stage, recorded, baselined, and all interface artifacts are put under configuration control.
- For developing systems, ensure a process is in place to define, agree, and configuration manage how the developing systems will interact with external systems across the interface boundary.
- For systems being developed by suppliers, insure there is a plan in place concerning how interfaces will be managed within the contract, especially for cases where the interface definitions are evolving with the design.
- Make all interface documentation available to developers as well as to those responsible for the external systems the SOI needs to interact.
- Track/monitor changes to all interface artifacts.
  - A change to an interface definition can impact both sides of the interface as well as the parent requirement.
  - Failing to manage interfaces, is a major reason for issues during system integration and for systems failing system verification and system validation.

Best Practices Associated with Managing Interfaces

- Incorporate the use of an Interface Control Working Group (ICWG), which include members responsible for each of the interfacing systems and system elements.
- Identify interfaces early as part of the definition of drivers and constraints and lifecycle concept analysis and maturation.
  - Because of their importance, all interfaces must be identified to ensure compatibility with other systems, define the system's boundaries, and manage risks associated with interfaces.
- Obtain copies of all interface documentation for existing systems.
  - If the interface definitions are not documented, work with the owners of the existing systems to document them.
  - Ensure their current configuration is defined in sufficient detail the developing SOI can be designed to successfully interact with the existing system across each interface boundary.
- Plan for system verification of all interface requirements, system integration, and assessing and validating the behavior of the integrated system as a function of the interactions across the interface boundaries.
  - Verification that the interactions are as required may be a prerequisite to integrate the SOI with the other system
  - The system on the other side of the interface may not be available
  - Often there will be a need for special equipment to assess the interactions across the interface boundary - simulators or emulators (software and/or hardware)
    - These must be budgeted for, developed, and go through their own system verification and system validation per a schedule consistent with the SOI's system integration and system verification, and system validation schedule.

Questions and Discussion

Lou Wheatcraft

- **Lou Wheatcraft** is a senior consultant and managing member of Wheatland Consulting, LLC. Lou is an expert in systems engineering with a focus on needs and requirements definition, management, verification, & validation. Lou provides consulting and mentoring services to clients on the importance of well-formed needs & requirements helping them implement needs & requirement definition and management processes, reviewing and providing comments on their needs and requirements, and helping clients write well-formed needs & requirements.
- Specialties include: Understanding & documenting the problem; defining project & product scope; defining and maturing system concepts; assessing, mitigating, & managing risk; documenting stakeholder needs; transforming needs into well formed design input requirements; allocation, budgeting, and traceability; interface management, requirement management; & verification and validation.
- Lou's goal is to help clients practice better systems engineering from a needs & requirements perspective across all life cycle stages of system/product development. Getting the needs & requirements right upfront is key to a successful project. Poor needs & requirements can triple the chances of project failure.
- Lou has over 50 years' experience in systems engineering, including 22 years in the United States Air Force. Lou has taught over 200 requirement seminars over the last 21 years. Lou supports clients from all industries involved in developing and managing systems and products including aerospace, defense, medical devices, consumer goods, transportation, and energy.
- Lou has spoken at Project Management Institute (PMI) chapter meetings and INCOSE conferences and chapter meetings. Lou has published and presented many papers concerning needs and requirement for NASA's *PM Challenge*, INCOSE, INCOSE *INSIGHT Magazine*, and *Crosstalk Magazine*. Lou is a member of INCOSE, past Chair and current Co-Chair of the INCOSE Requirements Working Group (RWG), the Software Engineering Institute (SEI), and the National Honor Society of Pi Alpha Alpha.
- Lou has a BS degree in Electrical Engineering from Oklahoma State University; an MA degree in Computer Information Systems; an MS degree in Environmental Management; and has completed the course work for an MS degree in Studies of the Future from the University of Houston – Clear Lake.