Traceability

The Threads That Link SE Artifacts Together Across the Lifecycle.

Requirements Working Group

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

Other Sources Concerning Traceability

The INCOSE SE HB v5, Section 3.2.3 greatly expanded the coverage of Traceability.

Members of the RWG coauthored a paper “Traceability – A vision for now and tomorrow” with the Configuration Management (CM) WG that was presented at IS2024 by Adriana D’Souza

34th Annual INCOSE international symposium

hybrid event

Dublin, Ireland
July 2 - 6, 2024

Traceability – A vision for now and tomorrow

Adriana D’Souza (Airbus) ; Louis S. Wheatcraft (Wheatland Consulting, LLC); Tami Katz (BAE Systems, Inc.); A. Larry Gurule (i-Infusion/CMPIC/SAE G33); Michael J. Ryan (Capability Associates Pty Ltd); Aleksander Przybylo (Boeing)

2-6 July 2024 www.incose.org/symp2024 #INCOSEIS 1

Needs and Requirements Manual (NRM)

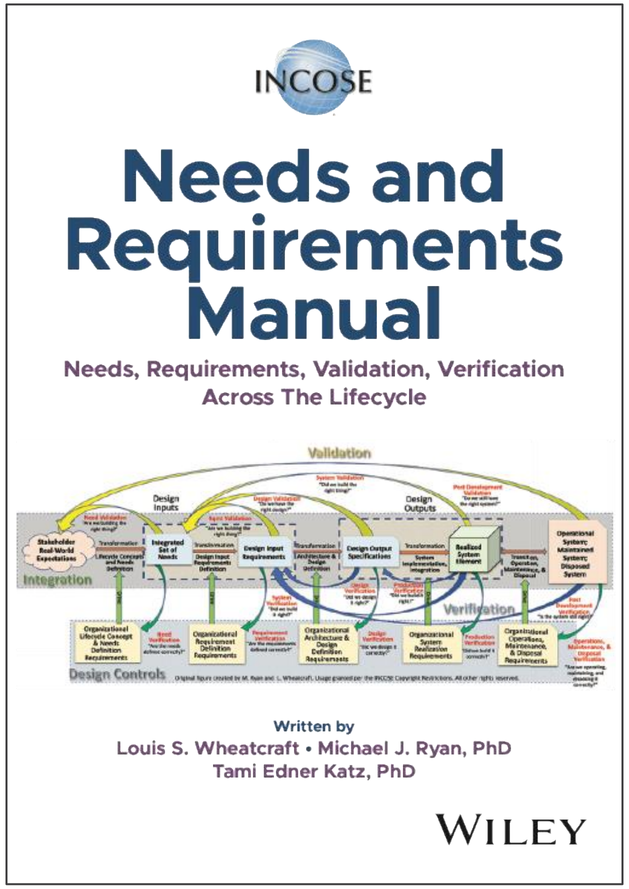
- The NRM is the RWG flagship product, V1 released in January 2022
- V1.1 minor updates in May 2022 to shorten title, add subtitle, and align with other RWG products
- Content aligns with, and expands, the INCOSE SE Handbook version 5 material.
- The NRM has been updated to Version 2, which is has been published by Wiley November 2024.

INCOSE

Needs and Requirements Manual

Needs, Requirements, Verification, Validation Across the Lifecycle

May 2022



INCOSE

Needs and Requirements Manual

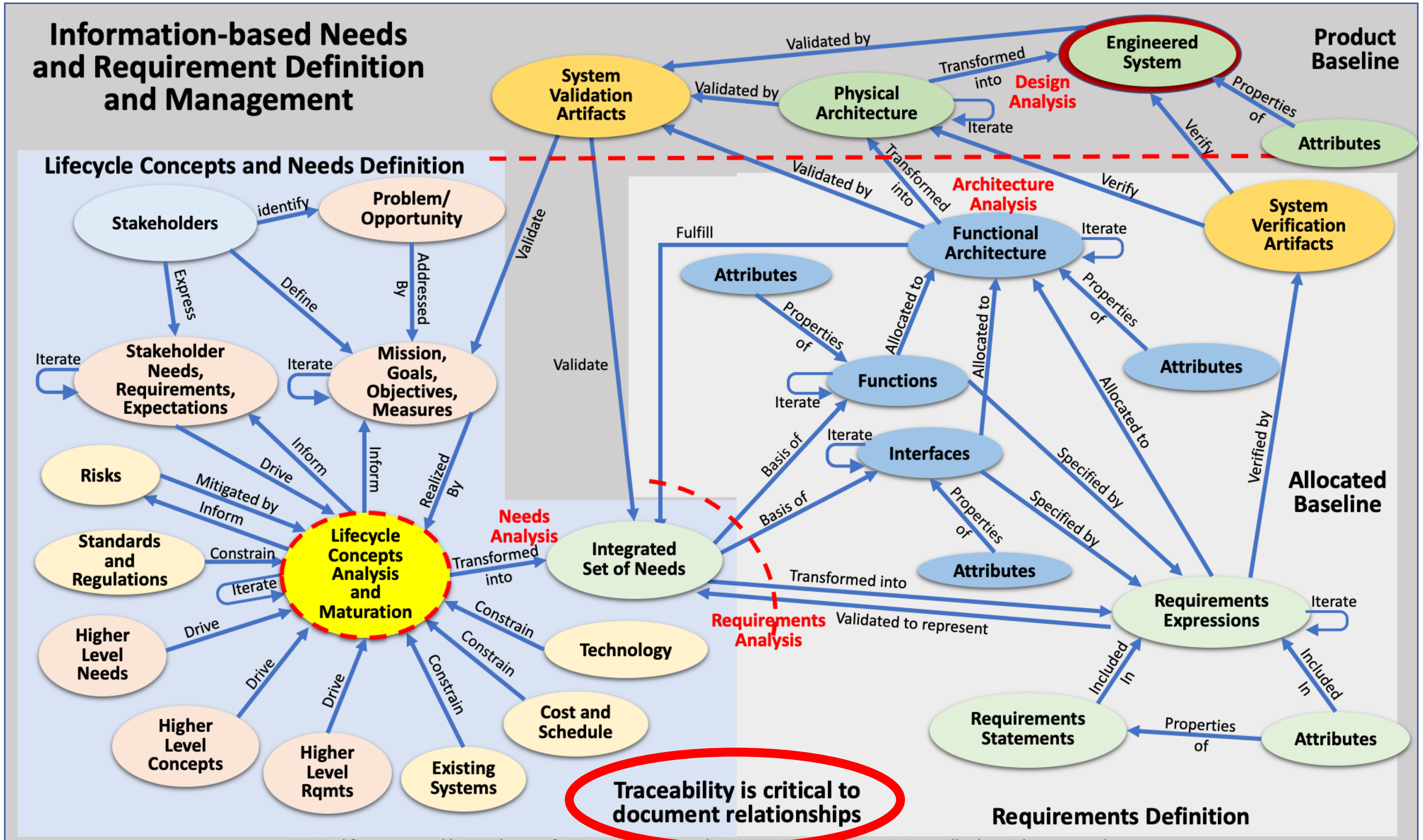
Needs, Requirements, Validation, Verification Across The Lifecycle

Written by
Louis S. Wheatcraft • Michael J. Ryan, PhD
Tami Edner Katz, PhD

WILEY

https://www.wiley.com/en-us/INCOSE+Needs+and+Requirements+Manual%3A+Needs%2C+Requirements%2C+Verification%2C+Validation+Across+the+Lifecycle-p-00386271

MBSE - A Data-Centric Practice of SE



Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

Showing Compliance with Standards & Regulations

- Many standards and regulations require traceability to be established across the lifecycle of the product/system;
  - ARP4754A Section 5.3.1.1 requires requirements dealing with safety to be "uniquely identified and traceable" to "ensure visibility of the safety requirements at the software and electronic hardware design level."
  - ISO26262 Section 6.4.3.2 requires "Safety requirements shall be traceable with a reference being made to:
    - a) each source of a safety requirement at the next upper hierarchical level;
    - b) each derived safety requirement at the next lower hierarchical level, or to its realization in the design; and
    - c) the verification specification in accordance with 9.4.2."
  - USC Title 21 Part 820 requires developing organizations of medical devices to develop and maintain a Device History File (DHF) that "shall contain or reference the records necessary to demonstrate that the design was developed in accordance with the approved design plan and the requirements of this part." **Traceability is a critical part of the DHF**.
  - ISO13485 Section 7.3.2 requires organizations to document "methods to ensure traceability of design and development outputs to design and development inputs."

ARP4754, "Guidelines for Development of Civil Aircraft and Systems"; ISO26262, "Road Vehicles — Functional Safety"; USC Title 21 Part 820, "Quality System Regulation" for Medical Devices ; and ISO13485, "Medical Devices - Quality Management Systems - Requirements for Regulatory Purposes"

Traceability Defined

Traceability

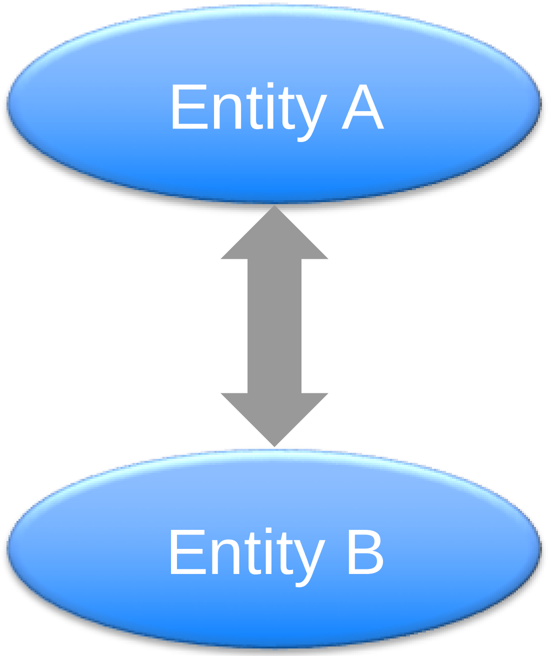
- Individual sets of lifecycle concepts, needs, requirements (CNR), design output specifications, system validation artifacts, and system verification artifacts do not exist in isolation.
  - Rather they represent a multi-level, “spider web” or “tapestry” of relationships that represents a data and information model of the integrated system.
  - These relationships are documented via traceability connections (i.e., links) that allow the relationships to be traced between the entities that are linked both vertically across levels and horizontally across the lifecycle.

This is the fundamental concept of traceability.

- *Traceability* is the ability to establish an association or relationship between two or more entities and to track entities from their origin to the activities and deliverables that satisfy them, as well as assess the effects on artifacts across the lifecycle when change occurs.
- Traceability can be “bidirectional”, “unidirectional”, “horizontal”, or “vertical”.

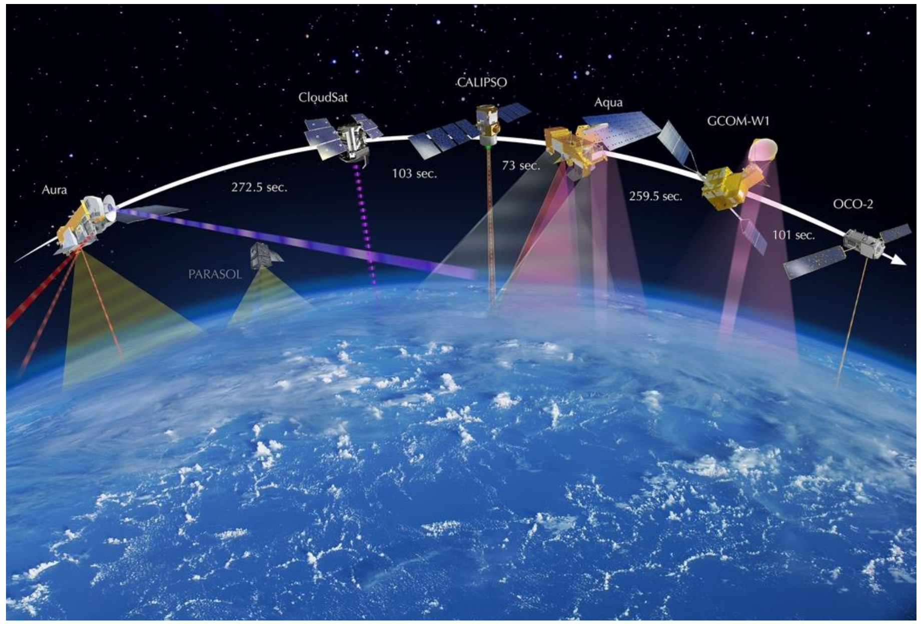
Bidirectional traceability

- **Bidirectional traceability** is the ability to establish a two-way link between entities such that each has knowledge of the other.
  - Enables practitioners to move forward, backward, or up and down digital threads (tapestry) that result from establishing bidirectional traceability.
  - Although the relationship is created in both directions, it is not the same relationship.
    - For example: One direction should be indicated as “traced from” or “traced to”. Similarly, “satisfies” and “satisfied by” are two different directions within a bidirectional “satisfy” relationship.



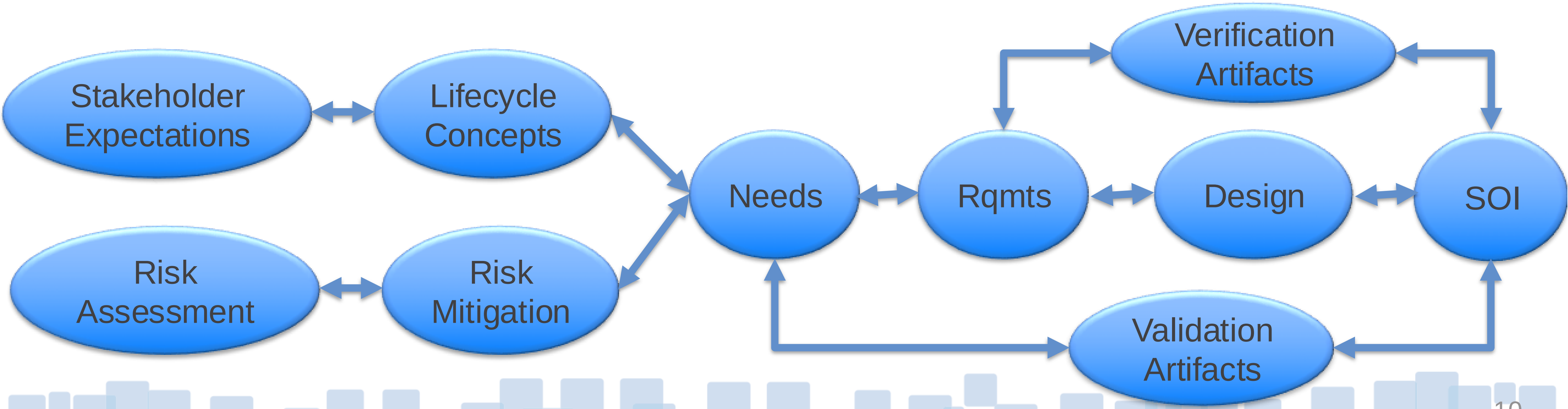
Unidirectional Traceability

- **Unidirectional traceability** is the ability to establish a one-way trace from one entity to another, where the source entity has no knowledge of the receiving entity.
  - Entity B establishes a trace to entity A, but entity A has no knowledge that entity B has a trace to it. Examples include:
    - A GPS satellite does not know the receivers, but the receivers know about the GPS.
    - Broadcasting is also unidirectional as the receivers know the broadcaster, but the broadcaster does not know its receivers.

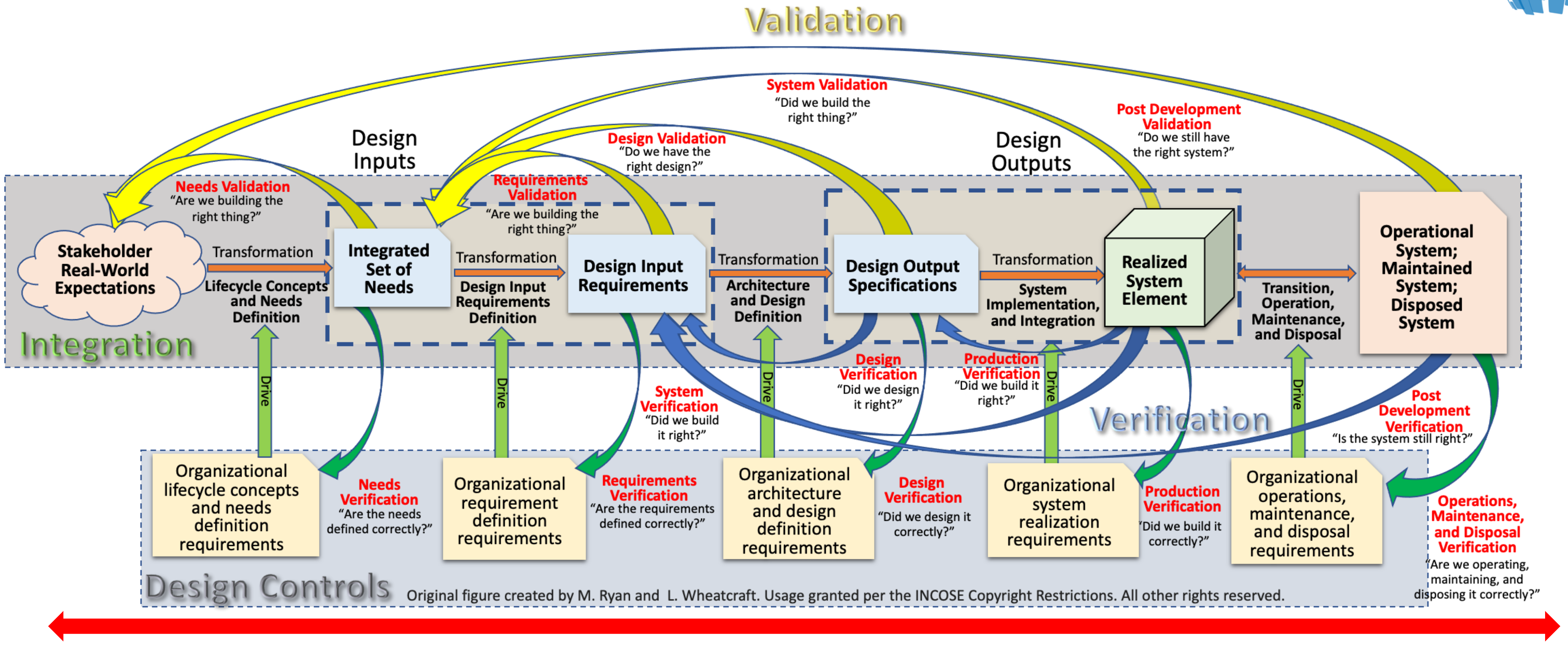


Horizontal Traceability

- **Horizontal traceability** involves the forward and backward traceability between entities across the SOI lifecycle.
- Horizontal traceability links data, information, and artifacts generated in one lifecycle process activity to data, information, and artifacts generated in other lifecycle process activities, resulting in a "digital thread" connecting these data, information, and artifacts across the lifecycle.



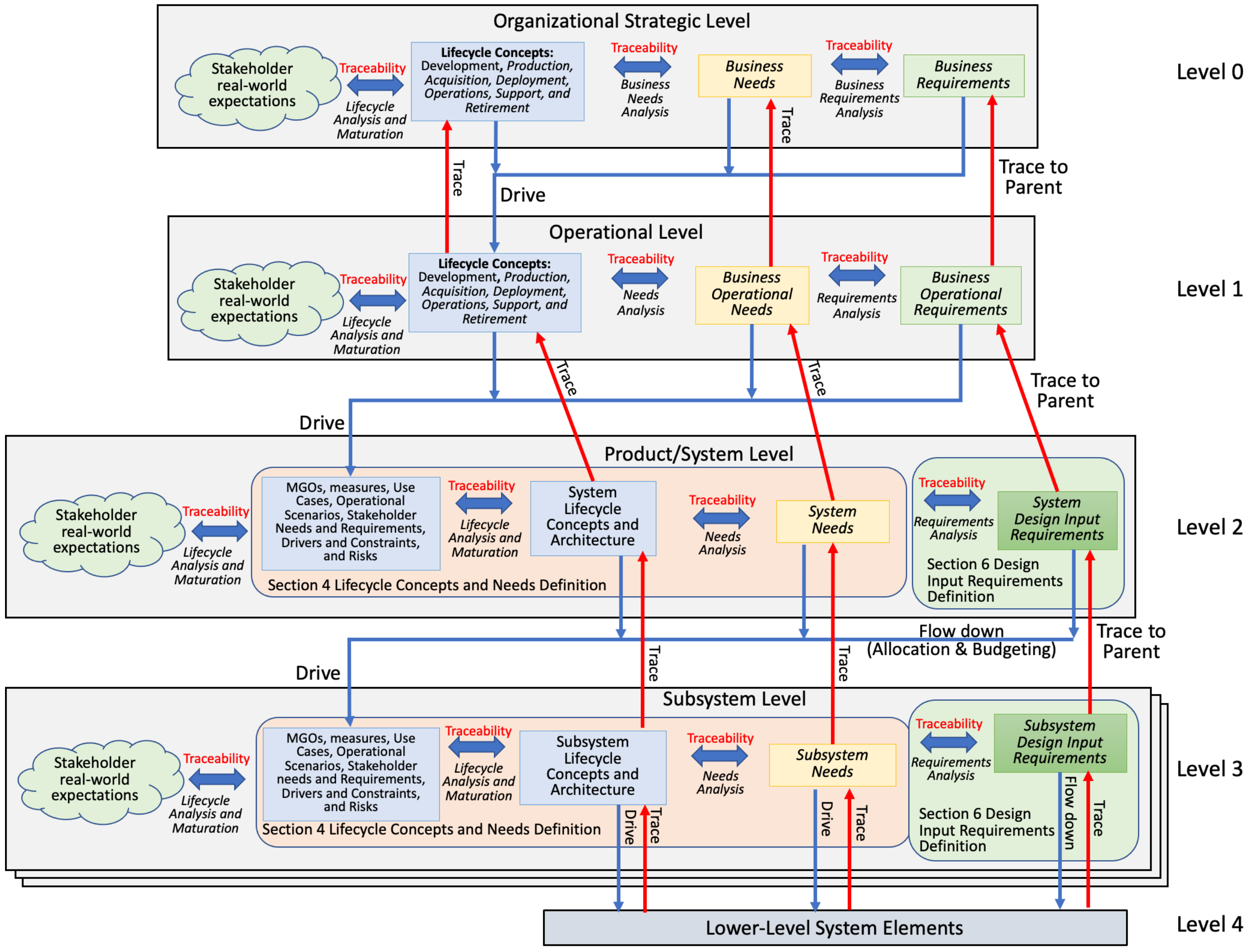
Horizontal Traceability Across the Lifecycle


Validation
System Validation
"Did we build the right thing?"
Post Development Validation
"Do we still have the right system?"
Design Validation
"Do we have the right design?"
Design Inputs
Design Outputs
Needs Validation
"Are we building the right thing?"
Requirements Validation
"Are we building the right thing?"
Stakeholder Real-World Expectations
Transformation
Lifecycle Concepts and Needs Definition
Integrated Set of Needs
Transformation
Design Input Requirements Definition
Design Input Requirements
Transformation
Architecture and Design Definition
Design Output Specifications
Transformation
System Implementation, and Integration
Realized System Element
Transition, Operation, Maintenance, and Disposal
Operational System; Maintained System; Disposed System
Integration
Drive
Drive
Drive
Drive
Drive
System Verification
"Did we build it right?"
Design Verification
"Did we design it right?"
Production Verification
"Did we build it right?"
Verification
Post Development Verification
"Is the system still right?"
Organizational lifecycle concepts and needs definition requirements
Needs Verification
"Are the needs defined correctly?"
Organizational requirement definition requirements
Requirements Verification
"Are the requirements defined correctly?"
Organizational architecture and design definition requirements
Design Verification
"Did we design it correctly?"
Organizational system realization requirements
Production Verification
"Did we build it correctly?"
Organizational operations, maintenance, and disposal requirements
Operations, Maintenance, and Disposal Verification
"Are we operating, maintaining, and disposing it correctly?"
Design Controls
Original figure created by M. Ryan and L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

Needs, Requirements, Verification, and Validation are the common threads that tie all lifecycle activities and artifacts together.

Vertical Traceability

- **Vertical traceability** is most often referred to in the context of levels of organization and architectural levels of the system or product under development.
  - Each level has lifecycle CNR defined for each entity at that level.
  - Organizational-level business requirements drive the development of the operational-level lifecycle CNR.
  - As the operational lifecycle CNR are defined, traceability is established with the higher-level business requirements.
  - This repeats for each level.



Organizational Strategic Level
Stakeholder real-world expectations
Traceability
Lifecycle Analysis and Maturation
Lifecycle Concepts: Development, Production, Acquisition, Deployment, Operations, Support, and Retirement
Traceability
Business Needs Analysis
Business Needs
Traceability
Business Requirements Analysis
Business Requirements
Level 0
Trace
Drive
Trace
Trace to Parent
Operational Level
Stakeholder real-world expectations
Traceability
Lifecycle Analysis and Maturation
Lifecycle Concepts: Development, Production, Acquisition, Deployment, Operations, Support, and Retirement
Traceability
Needs Analysis
Business Operational Needs
Traceability
Requirements Analysis
Business Operational Requirements
Level 1
Trace
Trace
Trace to Parent
Drive
Product/System Level
Stakeholder real-world expectations
Traceability
Lifecycle Analysis and Maturation
MGOs, measures, Use Cases, Operational Scenarios, Stakeholder Needs and Requirements, Drivers and Constraints, and Risks
Traceability
Lifecycle Analysis and Maturation
System Lifecycle Concepts and Architecture
Traceability
Needs Analysis
System Needs
Section 4 Lifecycle Concepts and Needs Definition
Traceability
Requirements Analysis
System Design Input Requirements
Section 6 Design Input Requirements Definition
Level 2
Flow down (Allocation & Budgeting)
Trace to Parent
Trace
Trace
Drive
Subsystem Level
Stakeholder real-world expectations
Traceability
Lifecycle Analysis and Maturation
MGOs, measures, Use Cases, Operational Scenarios, Stakeholder needs and Requirements, Drivers and Constraints, and Risks
Traceability
Lifecycle Analysis and Maturation
Subsystem Lifecycle Concepts and Architecture
Traceability
Needs Analysis
Subsystem Needs
Section 4 Lifecycle Concepts and Needs Definition
Traceability
Requirements Analysis
Subsystem Design Input Requirements
Section 6 Design Input Requirements Definition
Drive
Trace
Drive
Trace
Flow down
Trace
Level 3
Lower-Level System Elements
Level 4

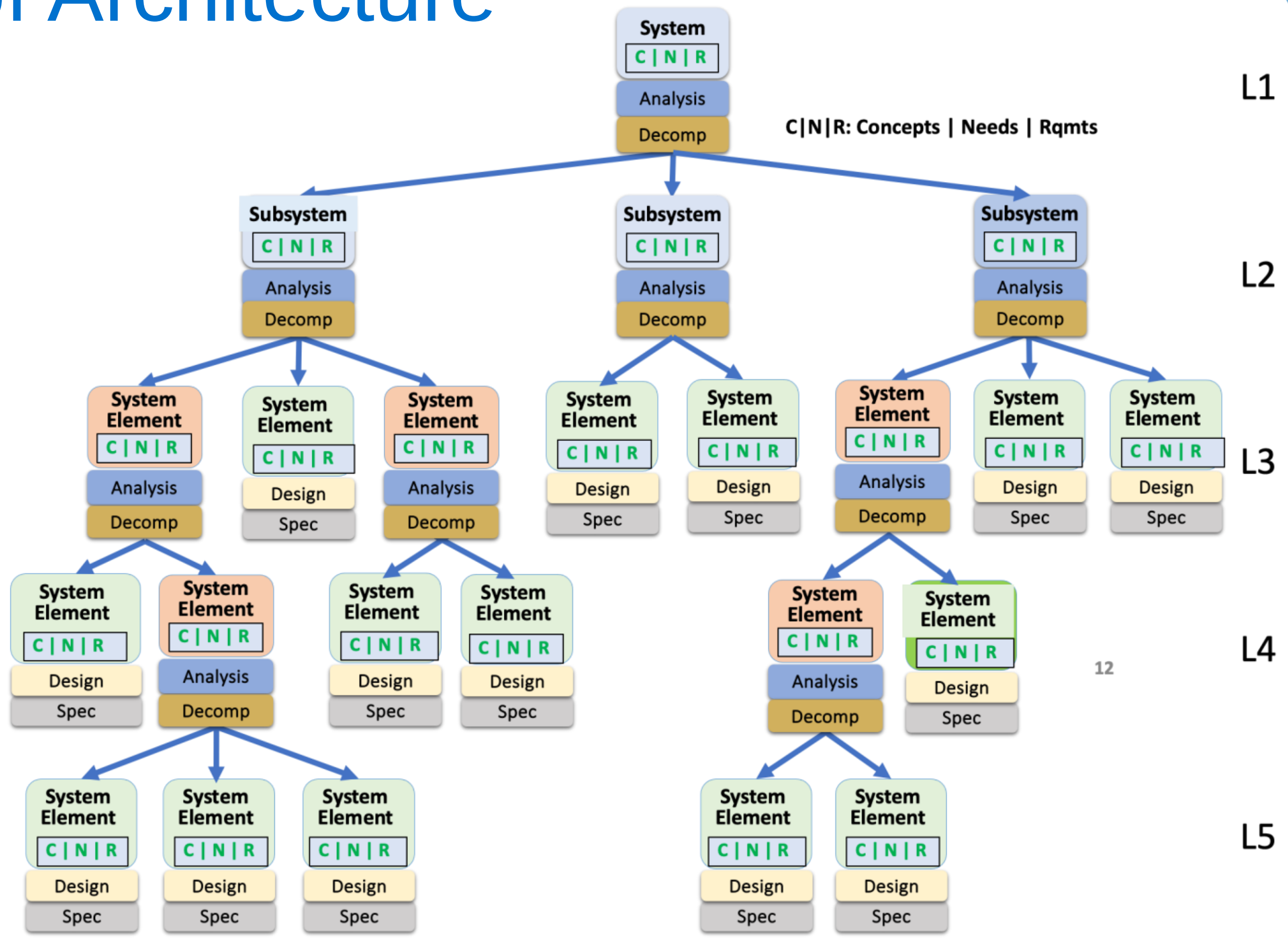
Original figure created by M. Ryan and L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

Vertical Traceability Between Artifacts at Different Levels of Architecture

Each part of the system architecture is represented by a family of systems engineering artifacts.

Artifacts at one level drive the definition of the artifacts at the next lower level.

The artifacts at a lower level, must trace to the artifacts generated at the next higher level.



Original figure created by M. Ryan and L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

A major issue with this view of the architecture is it doesn't show the interactions and relationships between parts of the architecture.

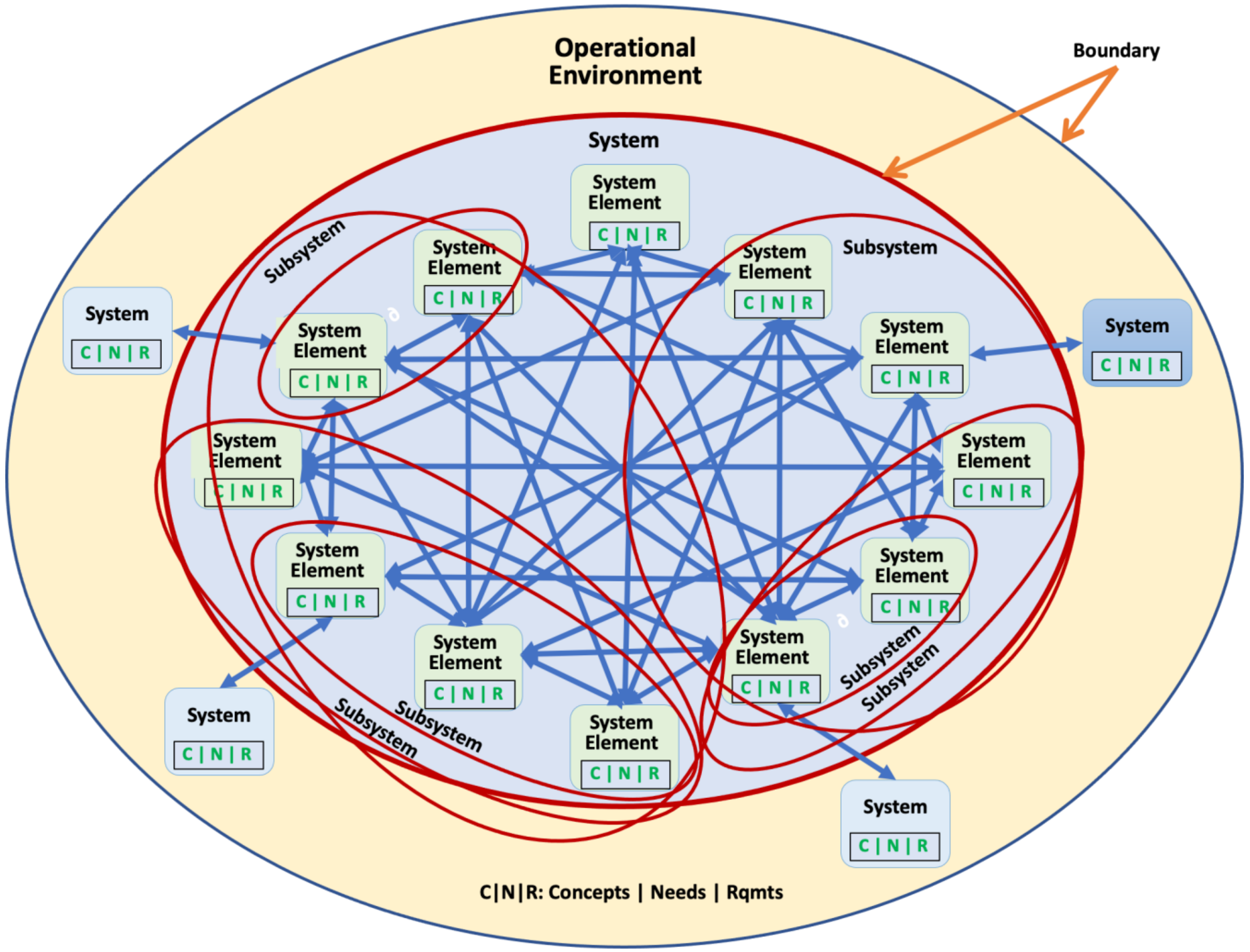
Types of Relationships

Relationships Between Entities

The behavior of a system is a function of the interactions of its parts and interactions of the system with its operational environment.

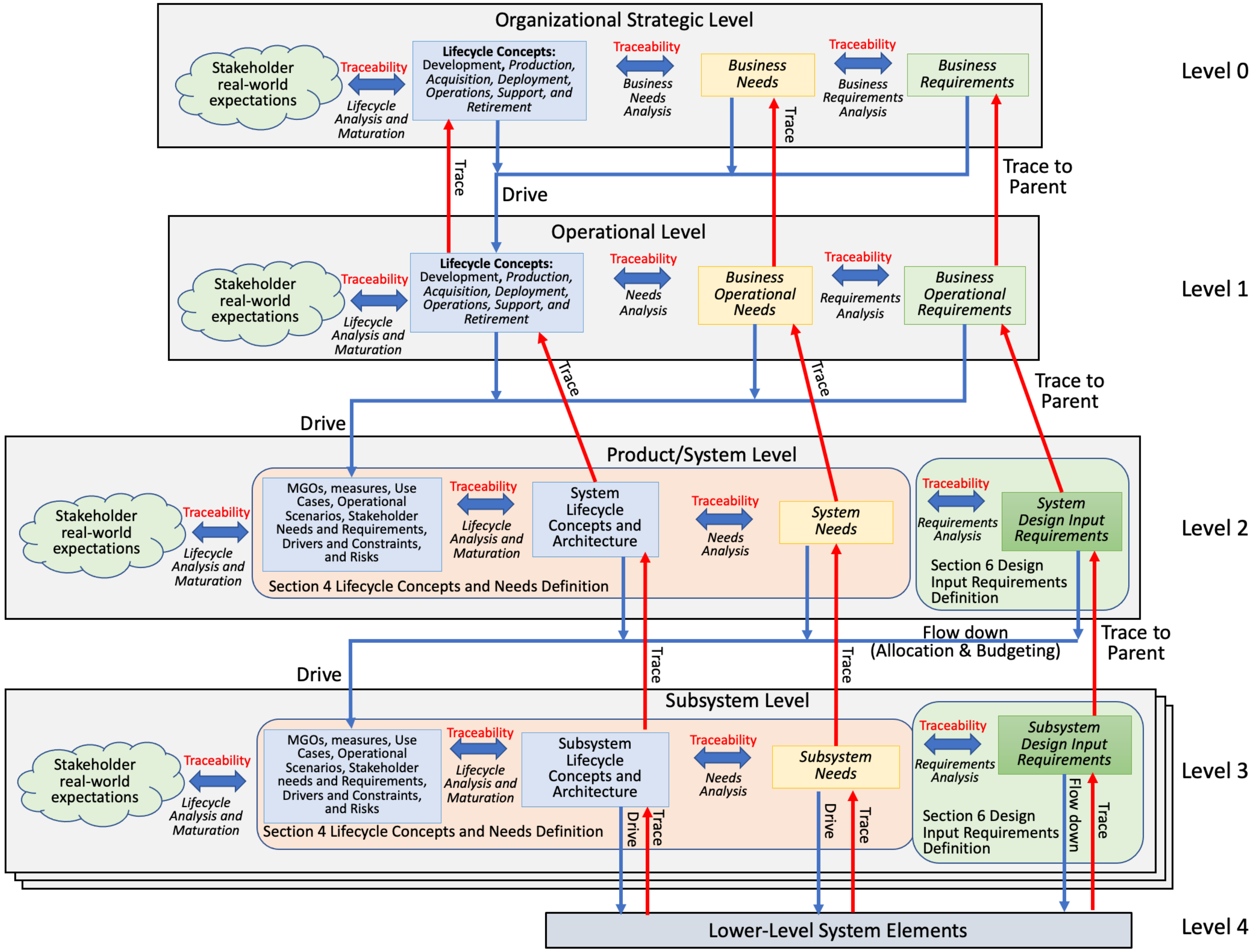
Systems Engineering can be viewed as a system made up of various artifacts and the relationships between artifacts.

Traceability enables the identification and management of the relationships between the various artifacts.



Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

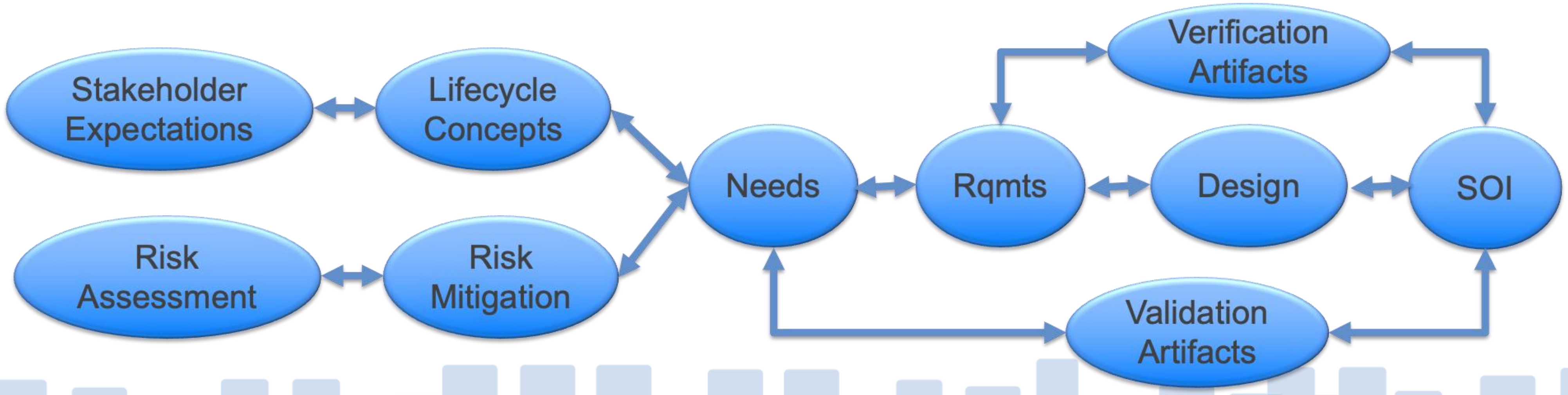
Horizontal and Vertical Relationships


Organizational Strategic Level
Stakeholder real-world expectations
Traceability
Lifecycle Analysis and Maturation
Lifecycle Concepts: Development, Production, Acquisition, Deployment, Operations, Support, and Retirement
Traceability
Business Needs Analysis
Business Needs
Traceability
Business Requirements Analysis
Business Requirements
Level 0
Trace
Drive
Trace
Trace to Parent
Operational Level
Stakeholder real-world expectations
Traceability
Lifecycle Analysis and Maturation
Lifecycle Concepts: Development, Production, Acquisition, Deployment, Operations, Support, and Retirement
Traceability
Needs Analysis
Business Operational Needs
Traceability
Requirements Analysis
Business Operational Requirements
Level 1
Trace
Trace
Trace to Parent
Drive
Product/System Level
Stakeholder real-world expectations
Traceability
Lifecycle Analysis and Maturation
MGOs, measures, Use Cases, Operational Scenarios, Stakeholder Needs and Requirements, Drivers and Constraints, and Risks
Traceability
Lifecycle Analysis and Maturation
System Lifecycle Concepts and Architecture
Traceability
Needs Analysis
System Needs
Section 4 Lifecycle Concepts and Needs Definition
Traceability
Requirements Analysis
System Design Input Requirements
Section 6 Design Input Requirements Definition
Level 2
Flow down (Allocation & Budgeting)
Trace to Parent
Trace
Trace
Drive
Subsystem Level
Stakeholder real-world expectations
Traceability
Lifecycle Analysis and Maturation
MGOs, measures, Use Cases, Operational Scenarios, Stakeholder needs and Requirements, Drivers and Constraints, and Risks
Traceability
Lifecycle Analysis and Maturation
Subsystem Lifecycle Concepts and Architecture
Traceability
Needs Analysis
Subsystem Needs
Section 4 Lifecycle Concepts and Needs Definition
Traceability
Requirements Analysis
Subsystem Design Input Requirements
Section 6 Design Input Requirements Definition
Drive
Trace
Drive
Trace
Flow down
Trace
Level 3
Lower-Level System Elements
Level 4

Original figure created by M. Ryan and L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

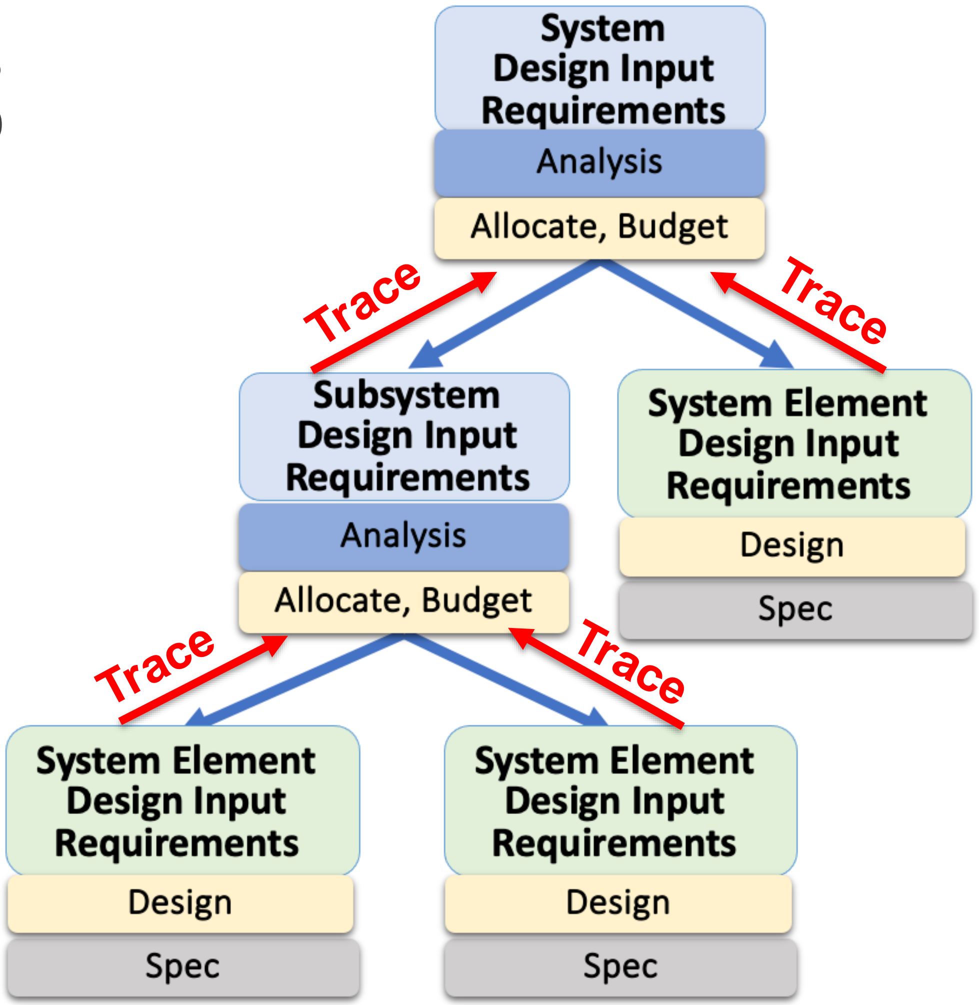
Types of Relationships

- ***Design input requirements to implementing design artifacts*** - shows a relationship between a design input requirement and its design implementation and resulting design output specifications.
- ***Requirements to system verification artifacts*** - displays relationship of requirements to verification artifacts or activities that will provide evidence of requirement satisfaction.
- ***Needs to system validation artifacts*** - displays relationship of needs to validation artifacts or activities that will provide evidence of need satisfaction.



Types of Relationships

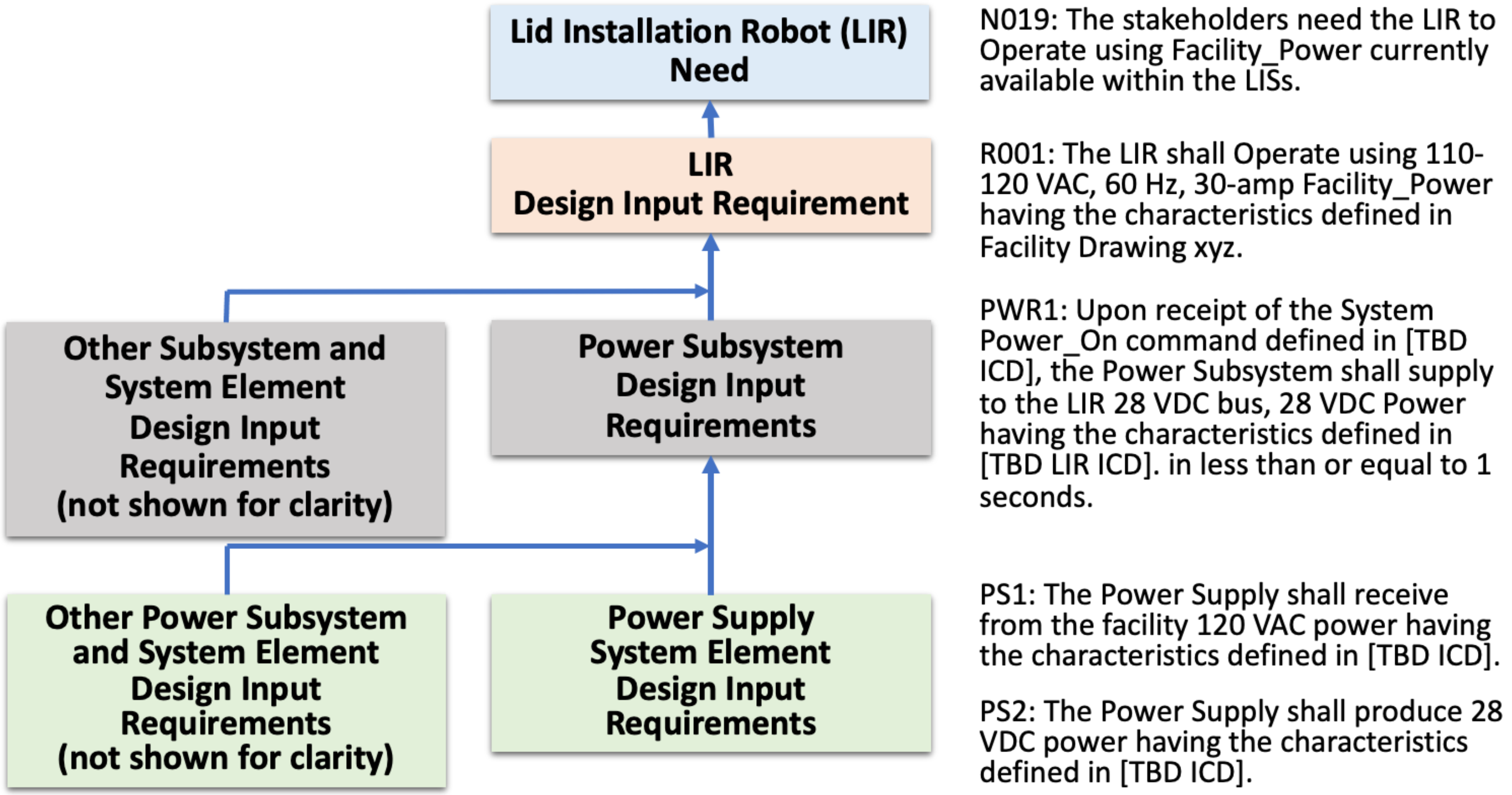
- ***Flow down of a requirement to a lower-level entity (allocation)*** – allows a link from a higher-level requirement to a lower-level subsystem or system element the parent requirement is allocated or budgeted to.
  - *Note: this is not a requirement-to-requirement link, but a requirement to a lower-level subsystem or system element within the SOI architecture, for which child requirements will be defined and then traced back to the allocated parent requirement.*
  - *The trace is child requirement to parent requirement*


System Design Input Requirements
Analysis
Allocate, Budget
Trace
Trace
Subsystem Design Input Requirements
Analysis
Allocate, Budget
System Element Design Input Requirements
Design
Spec
Trace
Trace
System Element Design Input Requirements
Design
Spec
System Element Design Input Requirements
Design
Spec

Original figure created by L. Wheatcraft.
Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

Types of Relationships

Parent/child – shows a connection of child requirement(s) to a higher-level allocated parent requirement that, when realized, will result in the intent of the parent to be met.


Lid Installation Robot (LIR) Need
LIR Design Input Requirement
Other Subsystem and System Element Design Input Requirements (not shown for clarity)
Power Subsystem Design Input Requirements
Other Power Subsystem and System Element Design Input Requirements (not shown for clarity)
Power Supply System Element Design Input Requirements

N019: The stakeholders need the LIR to Operate using Facility_Power currently available within the LISs.

R001: The LIR shall Operate using 110-120 VAC, 60 Hz, 30-amp Facility_Power having the characteristics defined in Facility Drawing xyz.

PWR1: Upon receipt of the System Power_On command defined in [TBD ICD], the Power Subsystem shall supply to the LIR 28 VDC bus, 28 VDC Power having the characteristics defined in [TBD LIR ICD]. in less than or equal to 1 seconds.

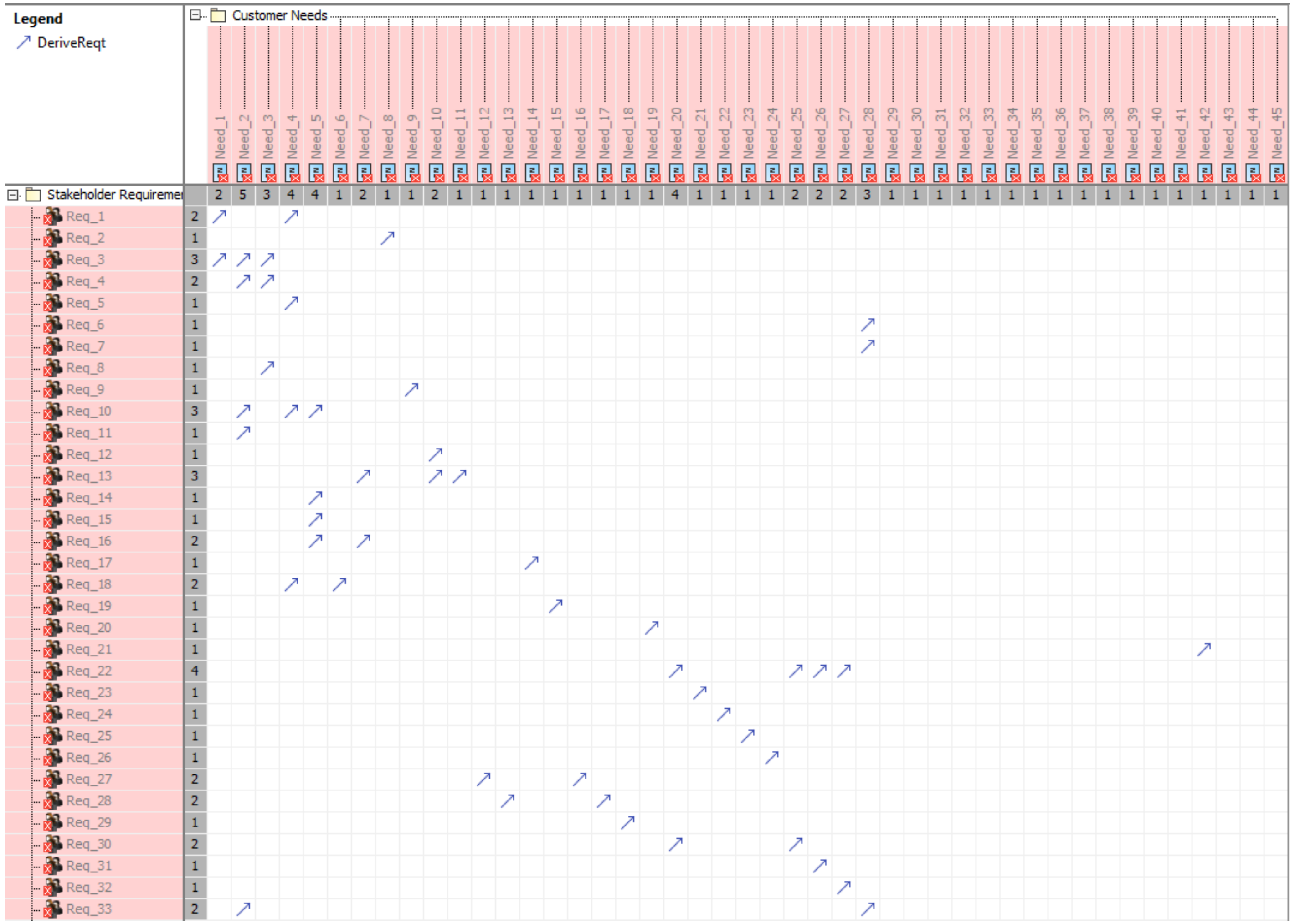
PS1: The Power Supply shall receive from the facility 120 VAC power having the characteristics defined in [TBD ICD].

PS2: The Power Supply shall produce 28 VDC power having the characteristics defined in [TBD ICD].

Use of Trace Matrices to View Relationships

System Need	System Requirement	Subsystem Requirement	System Element Requirement
N017: The stakeholders need the LIR to Operate using Facility-Power currently available within the LISs.	R026: The LIR shall Operate using 110-120 VAC, 60 Hz, 30-amp Facility_Power having the characteristics defined in Facility Drawing xyz.	PWR1: Upon receipt of the System Power_On command defined in [TBD ICD], the Power Subsystem shall supply to the LIR 28 VDC bus, 28 VDC power having the characteristics defined in [TBD LIR ICD] in less than or equal to 1 seconds.	PS1: The Power Supply shall receive facility120 VAC power having the characteristics defined in [TBD ICD]. PS2: The Power Supply shall produce 28 VDC power having the characteristics defined in [TBD] ICD.
		[Additional subsystem requirements would be shown below]	

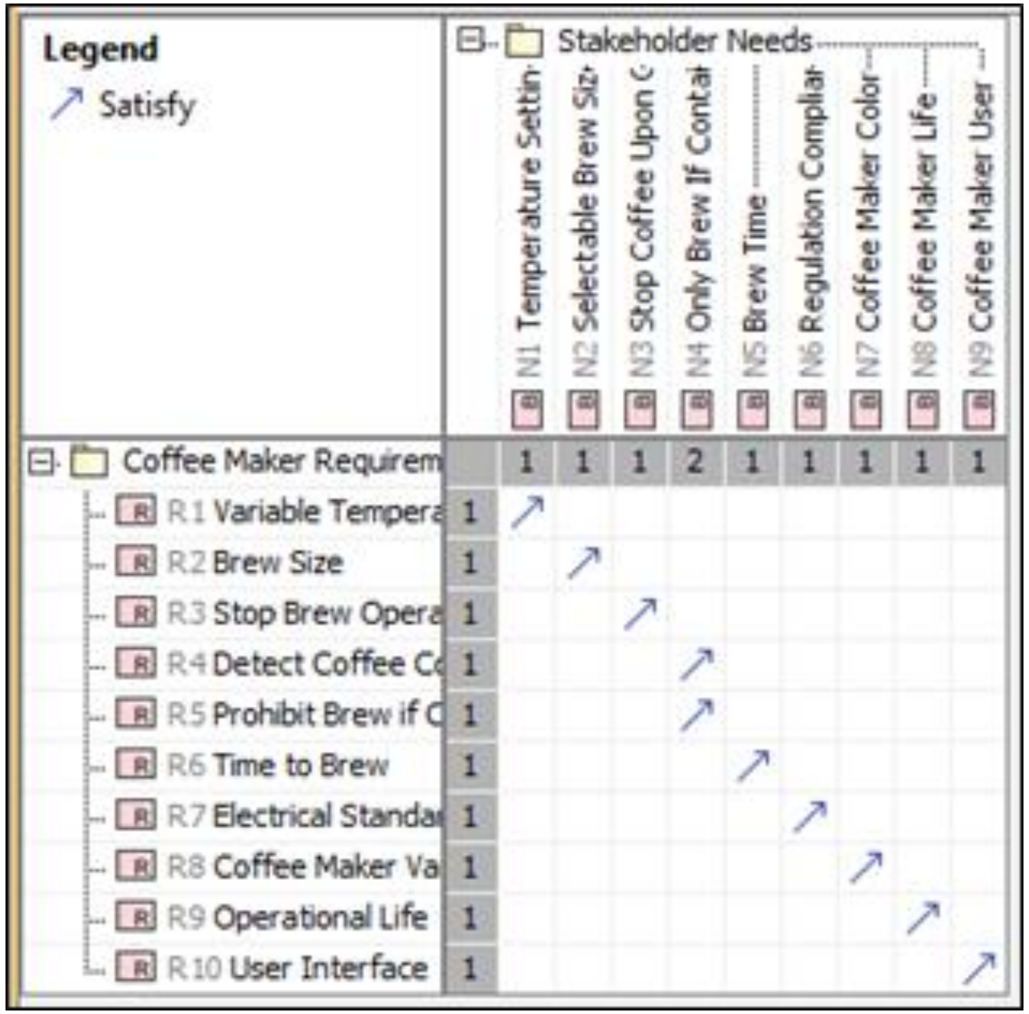
Use of Trace Matrices to View Relationships Within a SySML model



Legend
↗ DeriveReqt

Customer Needs

		Need_1	Need_2	Need_3	Need_4	Need_5	Need_6	Need_7	Need_8	Need_9	Need_10	Need_11	Need_12	Need_13	Need_14	Need_15	Need_16	Need_17	Need_18	Need_19	Need_20	Need_21	Need_22	Need_23	Need_24	Need_25	Need_26	Need_27	Need_28	Need_29	Need_30	Need_31	Need_32	Need_33	Need_34	Need_35	Need_36	Need_37	Need_38	Need_39	Need_40	Need_41	Need_42	Need_43	Need_44	Need_45
Stakeholder Requiremer		2	5	3	4	4	1	2	1	1	2	1	1	1	1	1	1	1	1	1	4	1	1	1	1	2	2	2	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Req_1	2	↗			↗																																									
Req_2	1								↗																																					
Req_3	3	↗	↗	↗																																										
Req_4	2		↗	↗																																										
Req_5	1				↗																																									
Req_6	1																												↗																	
Req_7	1																												↗																	
Req_8	1			↗																																										
Req_9	1									↗																																				
Req_10	3		↗		↗	↗																																								
Req_11	1		↗																																											
Req_12	1										↗																																			
Req_13	3							↗			↗	↗																																		
Req_14	1					↗																																								
Req_15	1					↗																																								
Req_16	2					↗		↗																																						
Req_17	1														↗																															
Req_18	2				↗		↗																																							
Req_19	1															↗																														
Req_20	1																			↗																										
Req_21	1																																										↗			
Req_22	4																				↗					↗	↗	↗																		
Req_23	1																					↗																								
Req_24	1																						↗																							
Req_25	1																							↗																						
Req_26	1																								↗																					
Req_27	2												↗				↗																													
Req_28	2													↗				↗																												
Req_29	1																		↗																											
Req_30	2																				↗					↗																				
Req_31	1																										↗																			
Req_32	1																											↗																		
Req_33	2		↗																										↗																	



Legend
↗ Satisfy

Stakeholder Needs

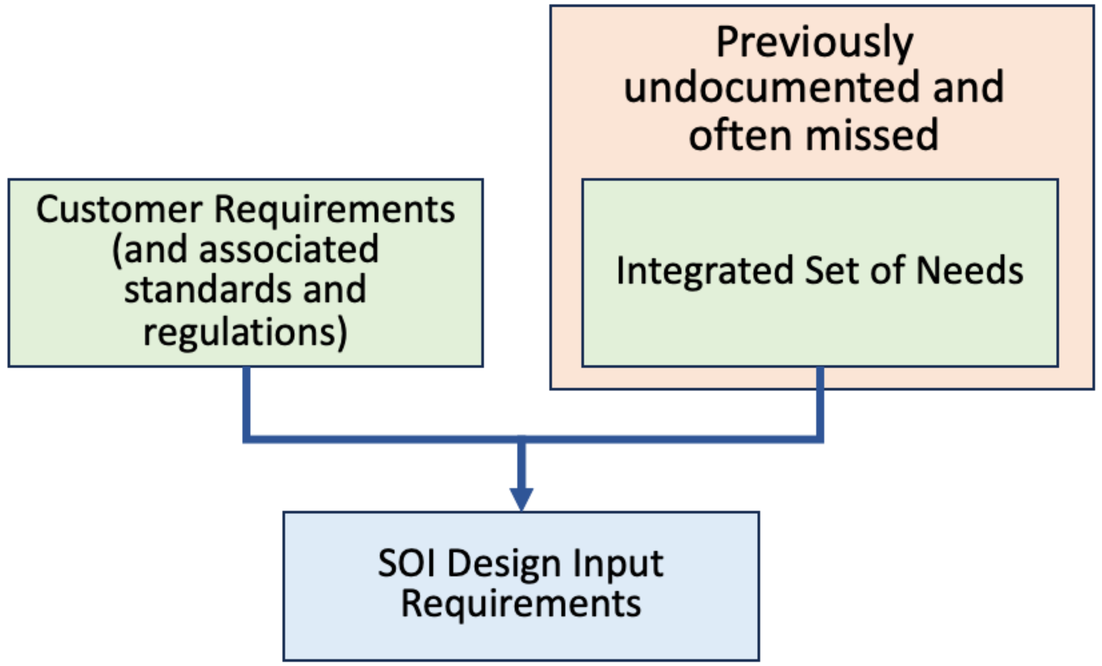
		N1 Temperature Settin	N2 Selectable Brew Siz	N3 Stop Coffee Upon C	N4 Only Brew If Contai	N5 Brew Time	N6 Regulation Compliar	N7 Coffee Maker Color	N8 Coffee Maker Life	N9 Coffee Maker User
Coffee Maker Requirem		1	1	1	2	1	1	1	1	1
R1 Variable Tempera	1	↗								
R2 Brew Size	1		↗							
R3 Stop Brew Opera	1			↗						
R4 Detect Coffee Cc	1				↗					
R5 Prohibit Brew if C	1				↗					
R6 Time to Brew	1					↗				
R7 Electrical Standar	1						↗			
R8 Coffee Maker Va	1							↗		
R9 Operational Life	1								↗	
R10 User Interface	1									↗

Types of Relationships

Requirement to a source - shows connection to where requirement content was derived (such as a constraint, standard, regulation, MGOs, measures, concept, risk, a model, analysis, or need)

In the past, the focus was on traceability of a child requirement to a parent allocated requirement as discussed in the previous slide.

In reality, the parent is just one source of a requirement.

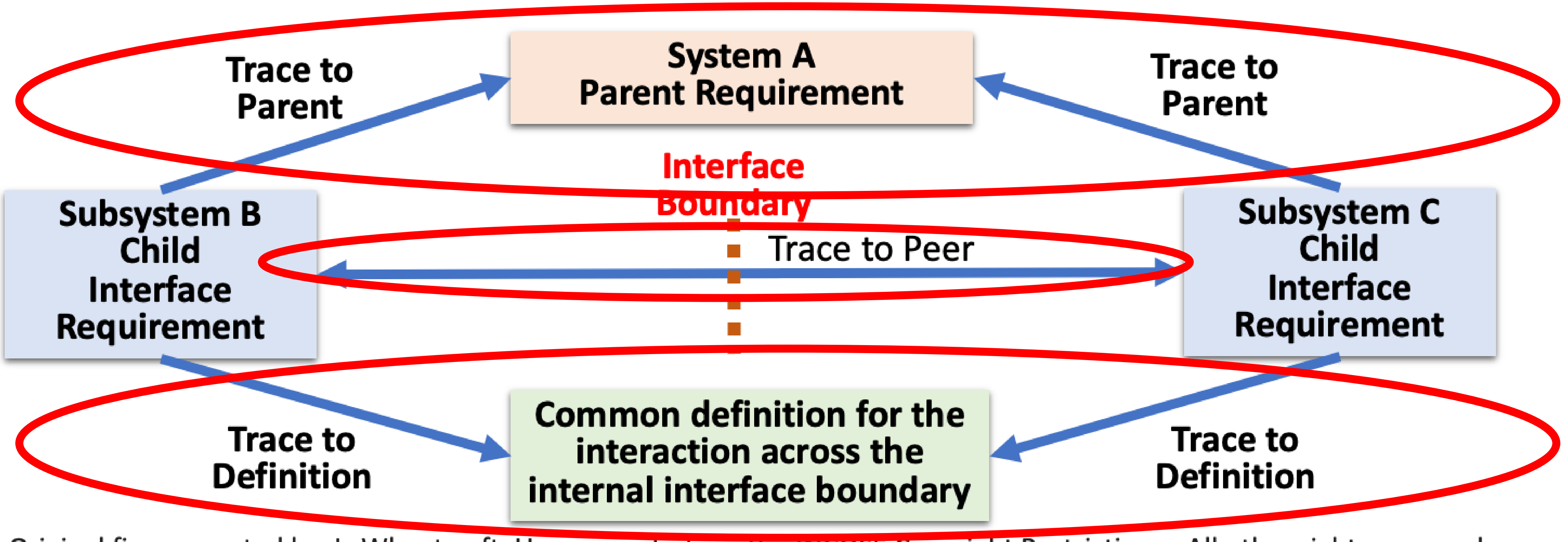

Previously undocumented and often missed
Customer Requirements (and associated standards and regulations)
Integrated Set of Needs
SOI Design Input Requirements

Original figure created by T. Katz and L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

In addition, we must also think of the source of the needs contained within the integrated set of needs.

Types of Relationships

Interface requirement to an interface definition - provides traceability between an interface requirement and the agreed to definition concerning the interaction across an interface boundary between the SOI and another entity.


System A Parent Requirement
Trace to Parent
Trace to Parent
Interface Boundary
Subsystem B Child Interface Requirement
Trace to Peer
Subsystem C Child Interface Requirement
Trace to Definition
Common definition for the interaction across the internal interface boundary
Trace to Definition

Types of Relationships

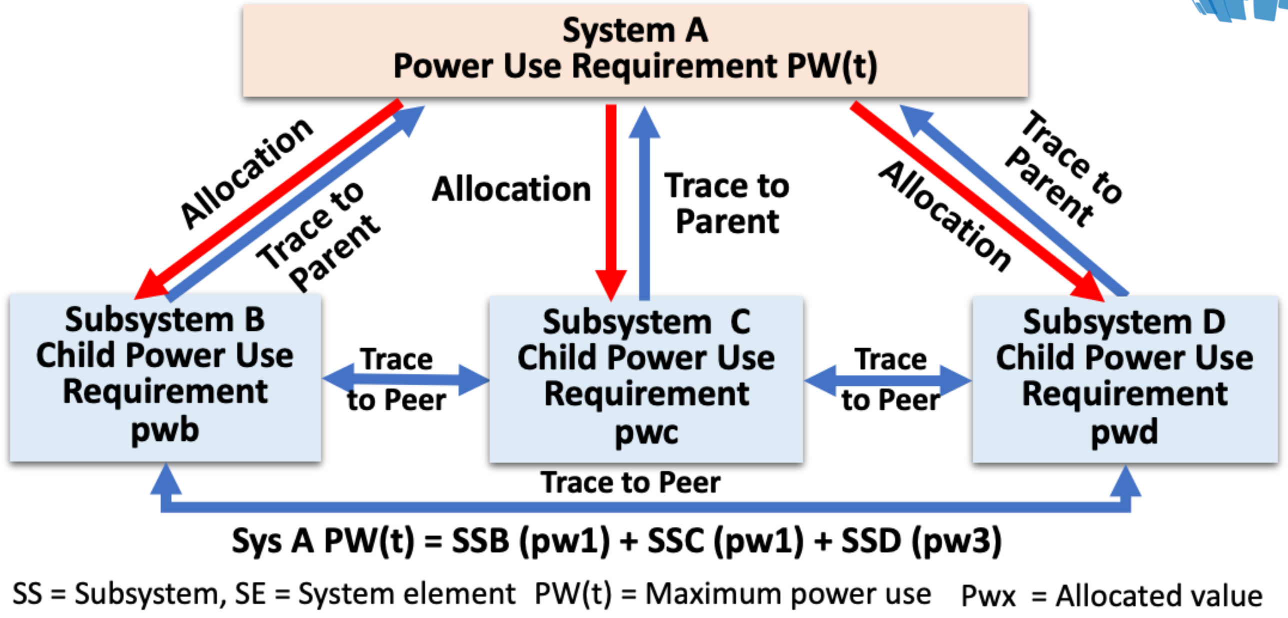
- ***Peer to peer-*** establishes relationships among requirements at the same level, either within the same set or requirements contained in different subsystem or system element sets of requirements.
  - The relationship could be general or grouped by topic (such as connecting requirements of a particular theme, connecting interface requirements, requirements sharing a common budgeted quantity, and connecting performance requirements that relate to a common function).
- ***Dependency of a requirement on another*** - shows a relationship of one requirement to one or more requirements in which there is a dependency
  - For example items that must be completed together to be satisfied (can be other requirements, or representation of specific conditions) or a relationship to another requirement such that a change in one will result in the need to change the other.
  - The other requirement could be in a separate set of requirements for a different system, subsystem, or system element.

Allocation and Budgeting

As part of allocation, performance, form, and quality requirements are apportioned (budgeted) to each system, subsystem, and system element that has a role in the realization of the budgeted value.

Budgets need to be managed and controlled at the integrated system level.

A critical concept associated with requirements and budgeting is that the budgeted quantities result in requirements that have a dependency - a change in one will result in the need to change another.

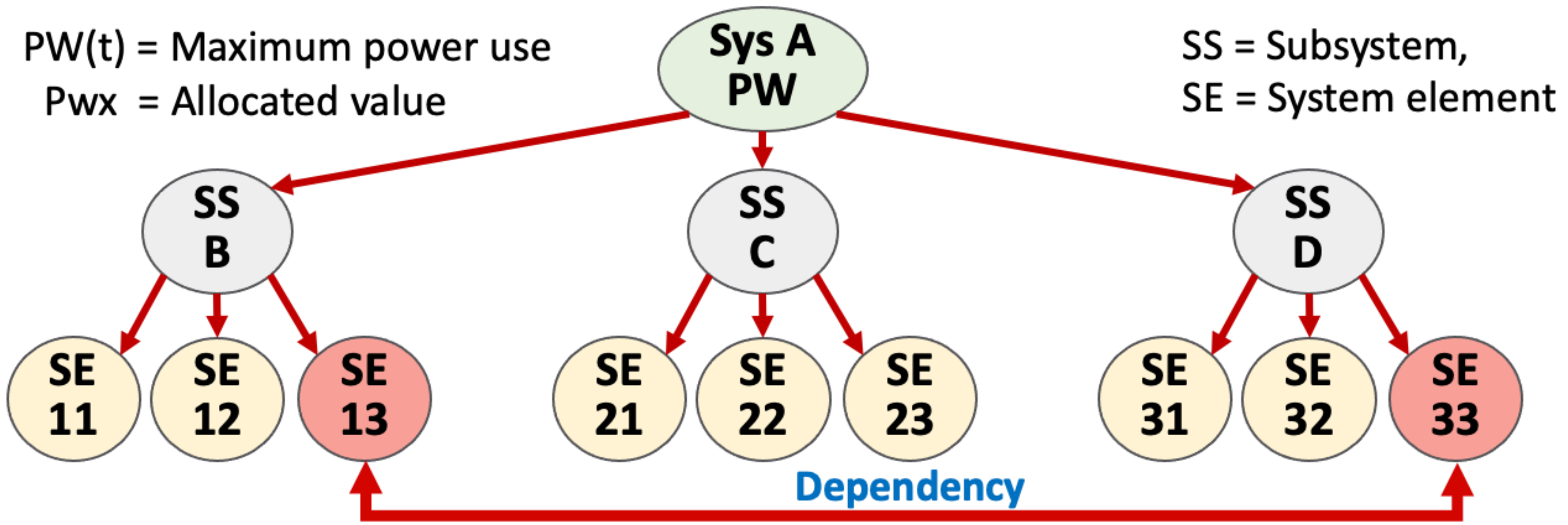

System A
Power Use Requirement PW(t)
Allocation
Trace to Parent
Allocation
Trace to Parent
Allocation
Trace to Parent
Subsystem B
Child Power Use Requirement
pwb
Trace to Peer
Subsystem C
Child Power Use Requirement
pwc
Trace to Peer
Subsystem D
Child Power Use Requirement
pwd
Trace to Peer
Sys A PW(t) = SSB (pw1) + SSC (pw1) + SSD (pw3)
SS = Subsystem, SE = System element PW(t) = Maximum power use Pwx = Allocated value

Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

Budgets must be carefully controlled as they tend to change as the design matures.

Both trace to parent and trace to peer are critical to manage and control budgeted values.

Allocation and Budgeting



Sys PW(t) = SSB(pwb) + SSC (pwc) + SSD (pwd)

= SE31(pw31) + SE32 (pw32) + SE33(pw33)

= SE21(pw21) + SE22 (pw22) + SE23(pw23)

= SE11(pw11) + SE12 (pw12) + SE13(pw13)

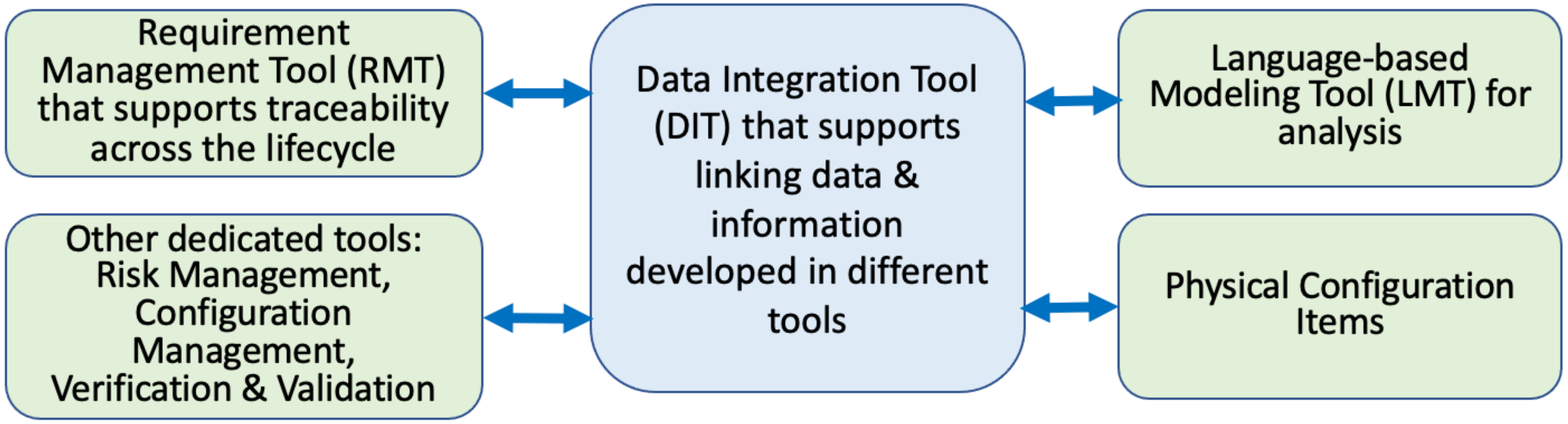
Dependency

Dependencies at lower levels may not be obvious, but it is critical to identify and manage them.

The use of models should help identify dependencies across multiple levels of an integrated system model.

Types of Relationships

- ***Textual requirement in an RMT to the equivalent requirement within a model*** - a trace between a requirement and an entity within a model. For example, a functional requirement would be linked to a function within a functional model; a performance characteristic of a function would be linked to that function within the model.



Requirement Management Tool (RMT) that supports traceability across the lifecycle
Other dedicated tools: Risk Management, Configuration Management, Verification & Validation
Data Integration Tool (DIT) that supports linking data & information developed in different tools
Language-based Modeling Tool (LMT) for analysis
Physical Configuration Items

Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

Must establish the ability to share and link data between tools to ensure consistency and establish an ASoT.

Relationships within Models

Relationships within Models

- Many of the benefits of using traceability relationships are additionally enhanced when requirements are developed using modeling techniques, such as with SysML or other language-based modeling tools.
- When using requirement content in SysML tools, there are additional ways requirements can show relationships to other requirements or other elements within the model using requirements diagrams.
  - For example, in SysML applications, requirement diagrams are used to display textual requirements, the relationships between requirements, and the relationships between requirements and other model elements.
- Most requirement relationships in SysML are based on the UML dependency.
  - The arrow points from the dependent model element (client) to the independent model element (supplier).
  - Hence in SysML, the arrow's direction is opposite to that typically used for requirements flows, where the higher-level requirement points to the lower-level requirement.

Types of relationships and dependencies within SysML

- **Containment** - showing how a model element is contained within a larger package, such as showing how requirements are contained within a specific theme package. For requirements, this could be used to indicate a requirement is contained within a given set for an entity within the system architecture.
- **Trace** - showing that requirements have a dependency, changes to a connected requirement could result in the need to change a dependent requirement.
- **Derive requirement** - type of dependency showing a requirement was derived from the connected requirement. Derived requirements can correspond to child requirements at the next level of the architecture indicating a parent/child relationship.
- **Refine** - type of dependency, connecting a requirement to a model element that provides more details, such as a use case, misuse case, loss scenario, user story, or operational concept.
- **Satisfy** - type of dependency, connecting a requirement to a model element that provides fulfillment of the requirement within the design, such as a design output specification.
- **Verify** - type of dependency, connecting a requirement to a model element that provides objective evidence of requirement satisfaction, such as a test case, verification *Activity,* verification *Procedure*, or other verification artifacts such as verification *Method*, *Success Criteria*, or *Strategy*.

Combine Allocation & Traceability to Manage Requirements

Combine Allocation & Traceability to Manage Requirements

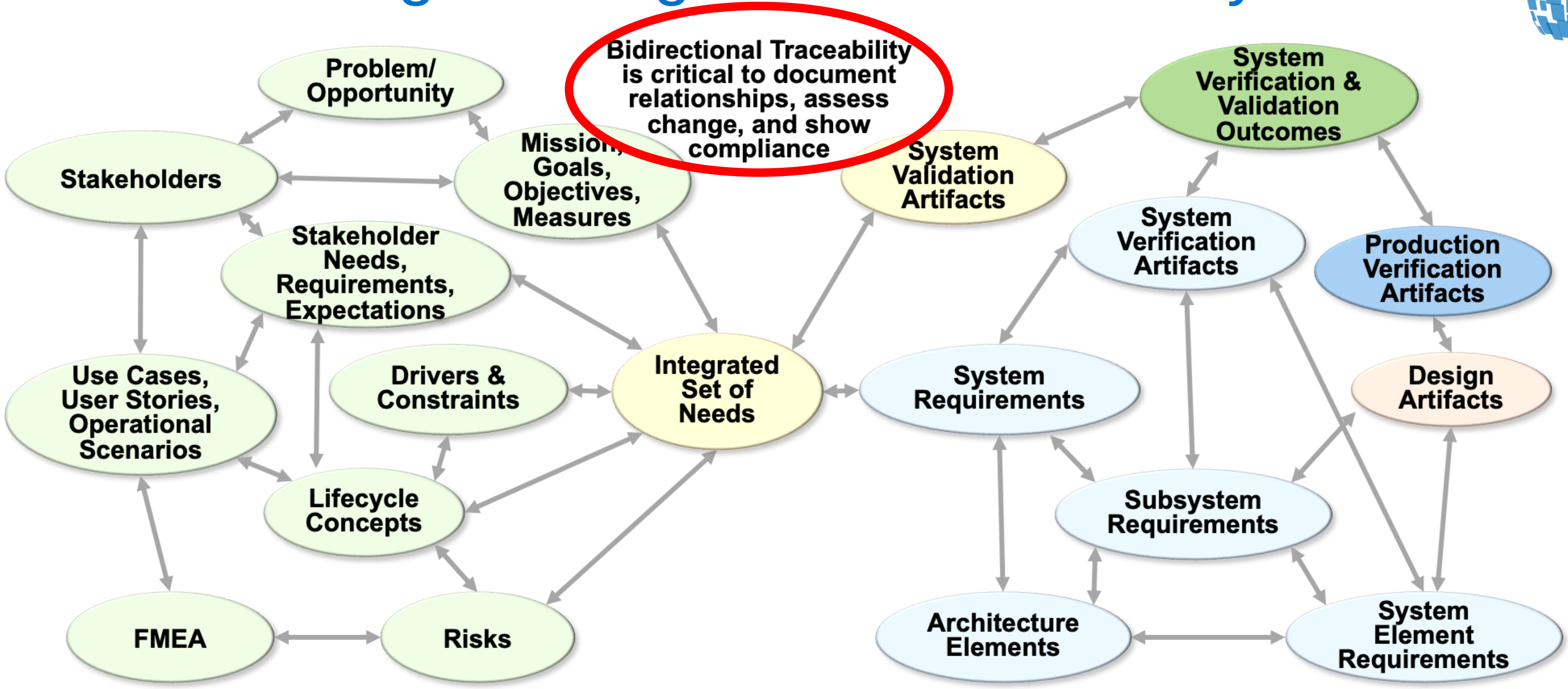
- Combining the concepts of allocation and traceability provides a powerful method to manage the design input requirements, especially across levels and across subsystems and system elements within a specific level.
- Unfortunately, the concept of allocation is not as well understood as traceability. There are several common misconceptions and bad practices concerning allocation and traceability:
  - *Some feel they can use the traceability matrices to assess allocation.*
    - The thought process is that if a parent has child requirements defined for a subsystem or system element, it is a safe assumption that the parent must have been allocated to that subsystem or system element – *this is often a bad assumption*!
  - *Missing information in the traceability matrix.*
    - Another issue is that when the traceability matrices are developed, rather than including the requirement text (or at least a summary or title of the requirement) in the matrix, only requirement numbers are used – *thus the validity and correctness of the allocation and traces is hard to assess.*
  - *Not all tool vendors understand the real meaning of allocation and thus do not always implement the concept correctly within their tools.*
    - One way to test this is to ask the tool vendors if they can generate a report that lists all child requirements that trace to a parent requirement that was NOT allocated to the subsystem or system element in which the child requirements exist.

Combine Allocation & Traceability to Manage Requirements

- Common defects that can be identified by combining the concepts of allocation and traceability include:
  - Parent requirements with no child requirements.
  - Needs with no implementing design input requirements.
  - Orphan needs that do not trace to a source.
  - Orphan requirements that do not trace to a need, parent, or a source.
  - Needs, sources, or requirements with incorrect or missing implementing children.
  - Child requirements with an incorrect parent or source**.**
  - Sets of child requirements are not necessary and sufficient to implement the parent requirement, need, or source from which it was transformed/derived.
- Addressing the issues discussed above is hard and can take a considerable amount of time and resources.
- Using a data-centric approach as advocated in the NRM can reduce the occurrence of these issues as well as make it much easier to identify and correct these issues.

Managing Change

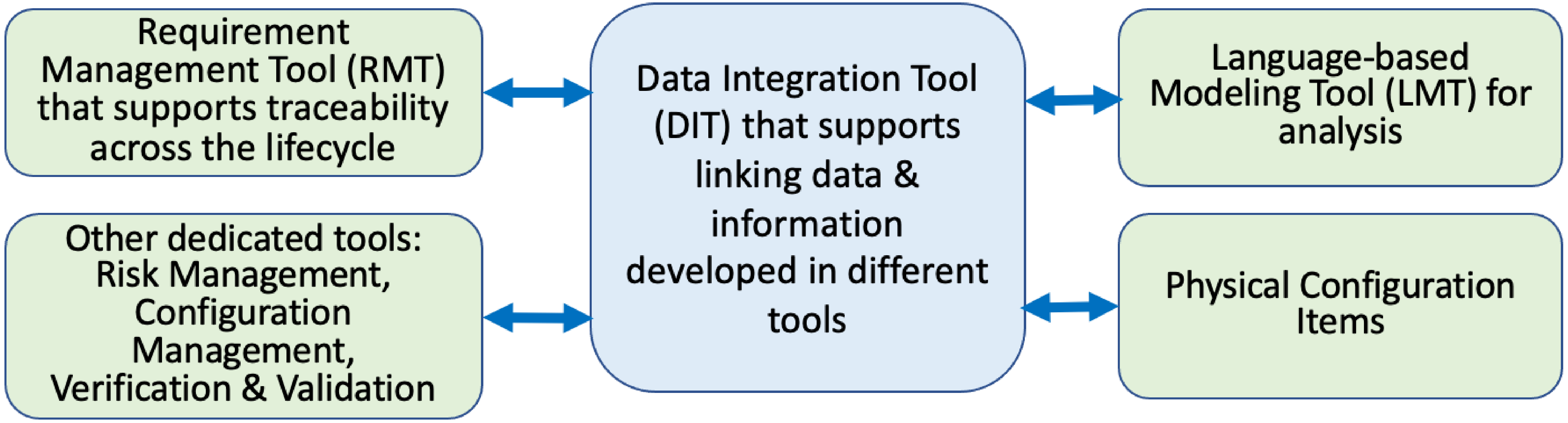
Manage Change Across the Lifecycle


Bidirectional Traceability is critical to document relationships, assess change, and show compliance
Problem/ Opportunity
Stakeholders
Mission, Goals, Objectives, Measures
Stakeholder Needs, Requirements, Expectations
Use Cases, User Stories, Operational Scenarios
Drivers & Constraints
Integrated Set of Needs
Lifecycle Concepts
FMEA
Risks
System Validation Artifacts
System Verification & Validation Outcomes
System Verification Artifacts
Production Verification Artifacts
System Requirements
Design Artifacts
Subsystem Requirements
Architecture Elements
System Element Requirements

Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

Define a Traceability Relationship Meta-model at the beginning of the project and implement within the project toolset.

Managing Change to Ensure Consistency Across the Project Toolset


Requirement Management Tool (RMT) that supports traceability across the lifecycle
Other dedicated tools: Risk Management, Configuration Management, Verification & Validation
Data Integration Tool (DIT) that supports linking data & information developed in different tools
Language-based Modeling Tool (LMT) for analysis
Physical Configuration Items

Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

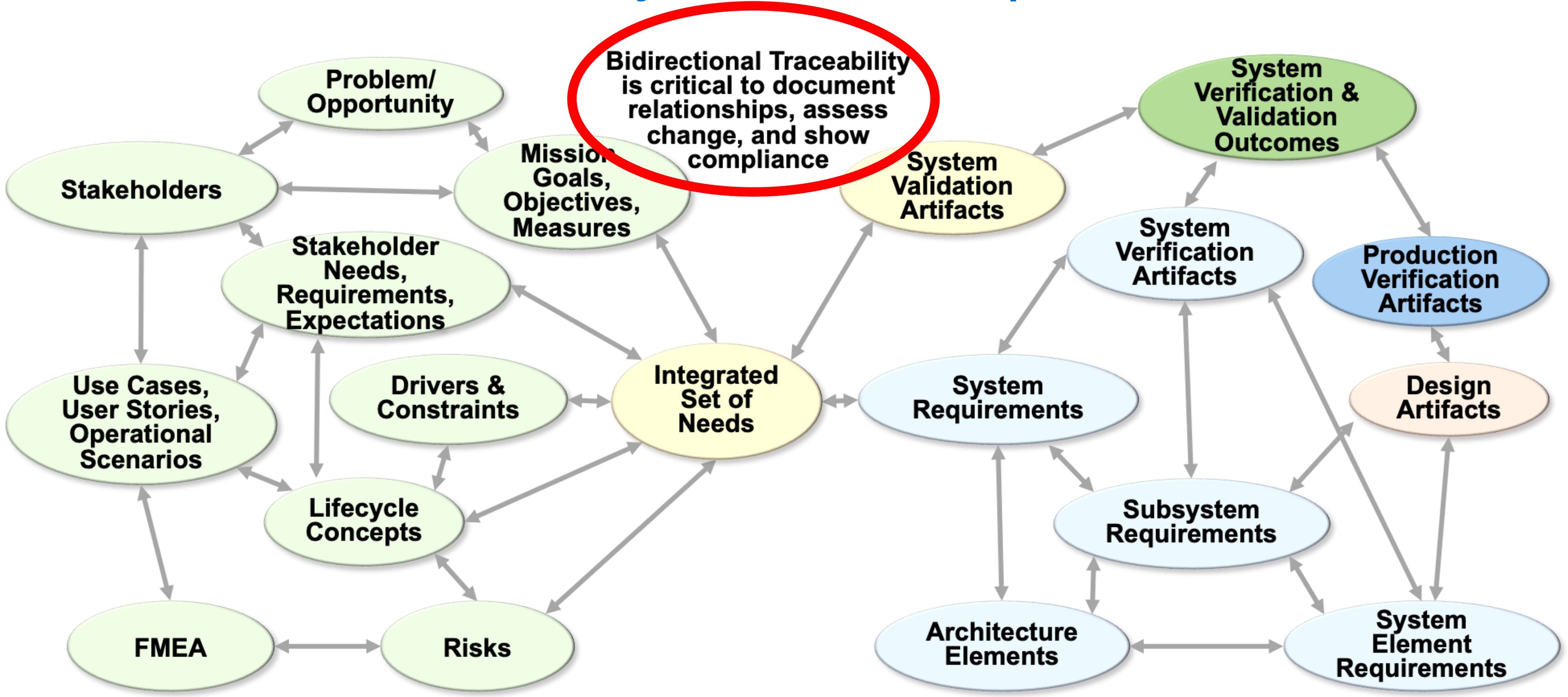
Must establish the ability to share and link data between tools to ensure consistency and establish an ASoT.

Guidelines for Recording & Managing Traceability

Guidelines for Recording & Managing Traceability

- There is a range of tools available for establishing and managing traceability
  - Unless there is a small set of requirements, it is highly recommended that the project team uses an RMT or other application.
  - The tools used to establish traceability and create and manage traceability records should be identified at the beginning of the project, as traceability can quickly become complex; switching tools mid-project could present major challenges.
- When models are used as an analysis tool to identify needs and design input requirements, the tools in the project toolset need to enable traceability between requirements and the models from which they were derived, even if the data are defined and maintained within different software applications.
  - This is key to being able to maintain consistency and correctness of the needs and requirements no matter which tool is used to view them. This enables the ability to change a need or requirement in one tool and have that change reflected in the other tool, so there is an ASoT. Refer to NRM Chapter 16 concerning features a project toolset should have.
- Design input requirements are transformed from the integrated set of needs, the trace between the requirement and its related need should be established when the requirements are initially defined and recorded within the RMT.
- Child requirements created in response to an allocated parent requirement should establish a trace to its parent as it is initially defined and recorded within the RMT.

Traceability Relationship Meta-model


Bidirectional Traceability is critical to document relationships, assess change, and show compliance
Problem/ Opportunity
Stakeholders
Mission Goals, Objectives, Measures
System Validation Artifacts
System Verification & Validation Outcomes
Stakeholder Needs, Requirements, Expectations
System Verification Artifacts
Production Verification Artifacts
Use Cases, User Stories, Operational Scenarios
Drivers & Constraints
Integrated Set of Needs
System Requirements
Design Artifacts
Lifecycle Concepts
Subsystem Requirements
FMEA
Risks
Architecture Elements
System Element Requirements

Original figure created by L. Wheatcraft. Usage granted per the INCOSE Copyright Restrictions. All other rights reserved.

Modern RMTs allow the project to define a traceability and dependency model within the tool and define rules concerning required traceability.
The tool will then enforce these rules, helping to minimize traceability issues.

Guidelines for Recording & Managing Traceability

- All requirements must trace to a need, a source, or a parent. Requirements that do not have this trace, are referred to as “orphan requirements”.
  - Even if the requirement does not trace to an allocated parent, there should be a source or need from which it was transformed. There should be no orphans.
- It is a best practice to define the system verification attributes when a requirement is defined.
  - Traceability to the system verification attributes should be established when the requirement is initially defined and recorded within the RMT (note that the verification data may exist in other applications in the digital ecosystem).
- It is also a best practice to trace requirements to the test cases within a verification plan or procedure that executes the system verification activity for that requirement.
- When a child requirement traces to multiple parents, needs, or sources, each trace must be assessed for its validity.
- The trace of all requirements to a parent, source, or need should be verified independently, if possible, to ensure that the requirements trace is correct during requirement verification.

Questions and Discussion

Lou Wheatcraft

- **Lou Wheatcraft** is a senior consultant and managing member of Wheatland Consulting, LLC. Lou is an international expert in systems engineering with a focus on needs and requirements development, management, verification, and validation across the system lifecycle.
- Lou's goal is to help systems engineers practice better systems engineering from a needs and requirements perspective across all life cycle stages of system/product development. Getting the needs and requirements right upfront is key to a successful project.
- 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 23 years. Lou supports clients from government and industries involved in developing and managing systems and products including aerospace, defense, medical devices, consumer goods, transportation, and energy.
- Lou is very active in the INCOSE and is the current chair of the RWG. Lou is a principal author of several RWG manuals and guides including the Needs and Requirements Manual (NRM), Guide to Needs and Requirements GtNR), Guide to Verification and Validation (GtVV), and newly released version 4 of the Guide to Writing Requirements (GtWR).
- 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.

INCOSE®

Requirements Working Group