THESIS

TOWARDS A MODEL-BASED IMPLEMENTATION IN TECHNOLOGY/PLATFORM LIFE-CYCLE DEVELOPMENT PROCESSES APPLIED TO A THRUST REVERSER ACTUATION SYSTEM (TRAS) CONCEPT

Submitted by
Jayesh B. Narsinghani
Department of Systems Engineering

In partial fulfillment of the requirements
For the Degree of Master of Science
Colorado State University
Fort Collins, Colorado
Fall 2021

Master's Committee:

Advisor: Kamran Eftekhari Shahroudi
Co-Advisor: Daniel R. Herber

Bret Windom

ABSTRACT

TOWARDS A MODEL-BASED IMPLEMENTATION IN TECHNOLOGY/PLATFORM LIFE-CYCLE DEVELOPMENT PROCESSES APPLIED TO A THRUST REVERSER ACTUATION SYSTEM (TRAS) CONCEPT

Modern systems are evolving continuously with rapid technological advancements and refreshments throughout the life cycle. Competitive organizations developing such systems constantly seek to apply holistic techniques for identifying problems that need to be solved and evaluating all possible technological solutions to guide the system development effort. The Model-Based Systems Engineering (MBSE) philosophy and tools are gaining widespread adoption in complex system development and claim several benefits over the traditional document-centric SE approach. This thesis investigates some model-centric implementations in the system development process. An organization's technology/platform development process, referred to as the Product Life Cycle (PLC) process, is analyzed to understand the workflow and expected outcomes necessary to progress in a system development effort. The phase- and gate-based process activities and resulting deliverables in the ideation phase are modeled and analyzed. A data modeling approach is also investigated to integrate the PLC process model with a system model. To develop the system architecture model of a thrust reverser actuation system (TRAS) concept, an MBSE methodology and framework are implemented. The solution boundary for subsystem alternatives is explored, and requirements are analyzed to compare how well the system solution matches the problem. The traditional, text-based requirements are examined to identify deficiencies, and an attribute-based approach is investigated to improve requirements' quality as well as perform requirement development and management activities more interactively and consistently. Overall, the key takeaways of the investigated model- and data-centric approach is summarized, and some insight on the path forward for future implementation is discussed.

ACKNOWLEDGEMENTS

I express my deepest gratitude towards Dr. Kamran Eftekhari Shahroudi and Dr. Daniel R. Herber, as they have always supported me and motivated me throughout my research. I want to thank my committee member Dr. Bret Windom, for providing insight into my research. I would also like to thank the Department of Systems Engineering at Colorado State University, especially Dr. Mike Borky, for mentoring me on my career path and Ingrid Bridge for assisting me throughout my time at CSU. I want to thank the team members at Woodward, Inc., including Jiahua (David) Lu, Leslie Olston, Matt Heath, Frederic Guitteny, Matthieu Moro, Joseph Kopecek, Bill Stone, Matthew Polley, and collaborators from INSA Toulouse, Julien Di Majo, Margot Vacher, and Louis Paulin for their time, expertise and willingness to maintain synergy with our CSU team. I would like to thank my friends Athul Krishna Sundarrajan, Akshay Dalvi, Urusha Silwal, Kalyani Joshi, Harshwardhan Ketkale, James Wheaton, and Sanket Mehrotra for supporting me in my journey. I thank Alexander and Daryani family for making my journey in the US smooth. Finally, I would like to thank my family for all the support and care.

Disclaimer: The opinions expressed herein are those of the author and do not necessarily represent the views of Woodward, Inc. or any other individual.

DEDICATION

Dedicated to my family

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

1.1 Challenges with System Development

Original Equipment Manufacturers (OEMs) constantly strive to keep pace with the continuously evolving technological advancements and disruptive changes to remain competitive in the modern market. OEMs and companies today face several pains when tackling existing and emerging challenges in system development that drive complexity situations when addressing stakeholders' needs, expectations, and concerns [1]. Developments towards new product design practices that meet the changing expectations are trending towards model and data-centric approaches for both standards-compliant and cradle-to-grave system development processes utilizing system architecture models. To support early verification and validation in complex system development, performing Systems Engineering (SE) activities demands a paradigm shift from the document-centric practices (common current state) towards data-centric/model-based practices (desired future state) [2].

An architecture-centric SE approach promises significant benefits to solve problems looming from the siloed/decentralized document-centric SE practices. Model-Based Systems Engineering (MBSE), which utilizes a centralized digital system model, is gaining widespread adoption across government, industry, and academia. This is the right time to explore the benefits of MBSE adoption. A modern intelligent, data-centric approach investigated in this thesis promises to overcome the deficiencies realized in the document-centric approach with a joint philosophy of systems thinking and object-oriented design/analysis. The modeling and research effort focuses on implementing model-centric practices across the early exploration phase in the Product Life Cycle (PLC) through system definition.

An important note is that much of the currently-available literature addresses pain points advocating MBSE adoption more from the "tool" standpoint. On the contrary, the pain points addressed in this modeling and research effort promote MBSE adoption, targeting value creation that directly benefits some specific enterprise and managerial concerns. A team of experienced system engineers and managers in the SE department representing an industry partner were interviewed for their opinions on some of these points. Some specific, deep-rooted pain points/issues and challenges justify the need to move documents to models. The modeling and research effort discussed in this thesis attempt to address the challenges as summarized below:

(*I1*) The *need* for a cultural shift/transformation from documents to models within organizations.

(*I2*) The *need* to break silos of knowledge to coordinate across engineering specialities and fill those gaps with an integrated modeling environment. This challenge of maintaining coordination between technical and management sides is commonly document-centric and, more often verbally communicated, in today's multidisciplinary teams.

(*I3*) The *need* to communicate holistic (big-picture) thinking that enables cross-collaboration among multiple disciplines in a project team.

(*I4*) The *need* to generate data and technology insight from a unified hub of true system information (technical truth) for early decision support capability. Further, exposing the risks and vulnerabilities earlier to realize the impact and consequences of specific design decisions more proactively.

(*I5*) Understanding the *value* of applying MBSE tools and their usage when performing system engineering activities.

1.2 Background

This section presents some fundamental SE concepts to provide a basis for the model-based implementation discussed throughout this thesis. This includes the definition of SE, system architecture, and overviews of MBSE and SysML diagrams. Finally, the section presents an overview of the enabling technology for more electric Thrust Reverser Actuation Systems (TRAS) to which the modeling approach is applied.

1.2.1 System Definition

A *system* is an assemblage or combination of functionally related elements or parts forming a unitary whole. A *system* can also be defined as,“a set of interrelated components functioning together toward some common objective(s) or purpose(s)” [3].

1.2.2 Systems Engineering and System Architecture

Systems Engineering as a discipline [4] can be defined as:

> “an interdisciplinary approach governing the total technical and managerial effort required to transform a set of stakeholder needs, expectations, and constraints into a solution and to support that solution throughout its life”.

SE focuses on ensuring the pieces work together to achieve the objectives of the whole [5]. As a discipline, it is defined under the umbrella of a broader subject of systems science. It originates from systems thinking, where the idea is that the overall capability delivered by the assemblage of the system is more significant than the capability delivered by the individual pieces. For example, consider an automobile/vehicle as a system interacting with the external entities such as environment, occupants/patrons, the source of energy, and supporting entities [6]. The vehicle system is built from subsystems such as a chassis that provides structural support, a drive/powertrain and propulsion subsystem that controls the system’s motion, a power subsystem that energizes the automobile, and infotainment that interfaces with the occupants. System Engineers are the central coordinators of system development practices, accountable for performing requirements analysis and allocation, trade studies, integration and test activities, and verification and validation. One might see a practicing system engineer analogous to the conductor of an orchestra [7], who coordinates the efforts of a team of multidisciplinary experts to ensure that everyone is marching towards the same objective that the system is trying to achieve as a whole. The effort aims to transform the stakeholder requirements into a balanced and optimized solution/system configuration subject to resource constraints [8]. For example, consider the automobile/vehicle system subject to exploration based on propulsion design alternatives. A solving system/implementation of the vehicle concept could be among Internal Combustion Engine (ICE), Hybrid Electric Vehicle (HEV), Plug-in hybrid vehicle (PHEV), or a Battery Electric Vehicle (BEV).

System Architecture—System architecture is an abstract representation of a complex system/entity specified in terms of:

- Structure: what piece parts interact/relate to each other through what interfaces and how? For instance, in the automobile/vehicle system example discussed in Sec. 1.2.2, the subsystems and components that interact with each other as well as the environment to operate the automobile.
- Behavior: what are the functions that each structural entity performs or should perform? For instance, the expected functions that each subsystem does to support the system function, as discussed in Sec. 1.2.2.
- Rules: what has to be satisfied by the system of interest and how it evolves? For instance, a requirement stated as, "the vehicle turning radius shall be less than 4m" must be satisfied by the vehicle and the imposing constraints quantifying the vehicle/system's measure of performance, turning radius determined in meters that improvise the requirement.

System architecture modeling is an approach representing the complexity of modern systems based on the following principles of object orientation:

- Abstraction: a technique for identifying common characteristics shared by a group of real-world entities to represent a broader, more generic entity possessing the shared characteristics. The primary purpose of abstraction is identifying and capturing commonality among entities to create a classifier. For instance, as discussed in Sec. 1.2.2, a braking subsystem is composed of components that are classifiers and group together similar components, like variants of sensors/encoders, valves, calipers, rotors, and pads.
- Generalization: a technique for applying common characteristics from generalized entities to multiple, more specialized entities. For instance, a generalized component caliper can inherit its specifications like outside diameter, bore diameter, spring length to a set of specialized

variants of a caliper such as fixed/floating. The primary purpose of generalization is to define the common characteristics of diverse entities [9].

Architecture modeling helps perform SE activities more efficiently and repeatably with specific work products that are unambiguous and easy to communicate in contrast to document-centric approaches [10]. The *document-centric* practices reflect on the creation of work products using static document formats. Whereas, the *data/model-centric* practices follow creation of artifacts/outcomes that are generated from a formal architecture model [9, 11].

1.2.3 Overview of Model-Based Systems Engineering (MBSE)

The INCOSE Systems Engineering Vision 2020 [12] defines MBSE as:

> “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 enhances the ability to capture, analyze, share, and manage the information associated with a system or product specification. It accomplishes this by formalizing several aspects of SE using models and includes the following:

- Provides an unambiguous definition of the system;
- Ensures rigor, repeatability, and producibility in SE processes;
- Promotes quality, completeness, and correctness in system designs to develop innovative products and systems;
- Reduces risk in SE activities such as requirements analysis, design, integration and test, and V&V;
- Promotes effective stakeholder dialog by enhancing communication and collaboration, synchronizing activities across organizations, and disciplines.

The current emphasis of a state-of-the-art model-based implementation is to build an integrated system model that provides multiple system views. Among other benefits, the modeling effort also serves to bridge the gap between external models that span multiple disciplines. For instance, regarding the automobile example discussed in Sec. 1.2.2, the different views of the underlying model can be used to represent an automobile/vehicle system. A model integrates multidisciplinary views such as architecture, CAD/CAM/CAE modeling and analysis, embedded hardware and software, functional safety, and programmatic planning, unifying the efforts and nurturing cross-collaboration. Moreover, design alternatives can be reused across vehicle/system concepts without duplicated efforts. A model can be queried to assess the errors that might be exposed later in the design process. Thus, saving huge on resource constraints within which projects must operate, such as labor hours, budget, and data. It is further reducing the cost and improving the development timeline. It is faster to convey the SE processes and outcomes using a graphical modeling tool rather than creating new documents that are disconnected from communicating similar information content [13].

Diagrams can be also be analyzed to fill missing gaps and errors. Diagrams are well-suited to communicate simplified ideas about the system, and quality work products can be generated from a centralized source. For instance, performing a quantitative comparison of design alternatives and plotting the results from the model, or generating the interface control document (ICD) more readily and with reduced efforts [14]. A consistent model enables shared understanding among all the team members by representing and maintaining the technical system specification that is reusable [15]. This also enhances change impact assessment. For instance, a change introduced in user specification can be traced to all the interlinked model elements. The propagation is assessed in different views, and changes are identified significantly faster than in traditional documents.

Thus, MBSE tools are the key enablers of formal architecture modeling in a tool-supported environment. For instance, a tool like Cameo Systems Modeler was employed in this modeling effort [16]. Such tools also support a lite-version of the Systems Modeling Language (SysML) that focuses on particular aspects tailored to the modeling needs.

Even with the above-mentioned benefits, the existing MBSE tools are bound by certain limitations and deficiencies that are evident and listed as follows:

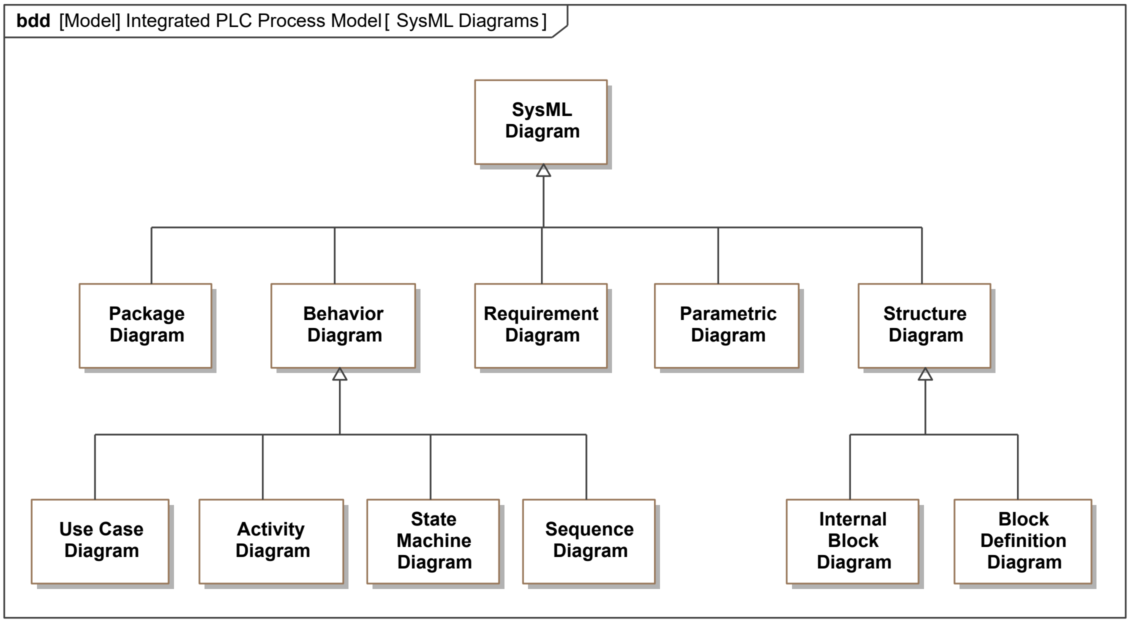


Figure 1.1: Types of SysML diagrams

- Lack of interoperability within different modeling environment, efforts are currently underway to develop mature tools;
- Huge amount of information content needs to be migrated from legacy database from different tool environments. For example, requirements of the legacy system is observed in a requirement management database, such as DOORS;
- The capability and skills required to develop models in the SysML can be involved and demands additional learning across the development teams;
- The modeling environment of the tool is dependent on the user interface developed by a particular vendor.

Systems Modeling Language (SysML)

The SysML is a profile of the Unified Modeling Language (UML) specified by the Object Management Group (OMG) [17]. SysML is a widely-supported graphical modeling language standard and a key enabler MBSE [15]. Figure 1.1 depicts the nine kinds of basic SysML diagrams that are used to present various aspects of a system. These diagrams are summarized below:

- *Package Diagram (PKG)* presents the organization of the model into packages that contain model elements and diagrams. For instance, all structural elements can be created in the *Structure* package, and all requirements go to the *Requirements* package.
- *Requirement Diagram (REQ)* is used to capture the requirements and their relationship with other requirements and model elements. For instance, from Sec. 1.2.2 the vehicle performance requirements can be decomposed to more specific requirement specification such as *vehicle turning radius*, *vehicle mileage*, etc. These can also be represented within a requirement table.
- *Use Case Diagram (UC)* is used to capture the high-level system functions that will be achieved by the User Roles or external systems that interact with it. For instance, *drive the vehicle*, *provide heating*, *maintain the vehicle*, etc.
- *Activity Diagram (AD)* depicts a sequence of interconnected actions. For instance, the Use Case scenario, *drive the vehicle* can be elaborated with an activity diagram to depict a series of actions and events that occur during the vehicle operation.
- *State Machine Diagram (STM)* represents the stateful behavior of an entity using states and transitions. For instance, on ignition, vehicle transitions from the *Off* state to *On* state, during which the vehicle could either be in *Idle* or *Cruise* or *Brake* states.
- *Sequence Diagram (SD)* depicts the interaction as the exchange of messages among parts of a system/entity. For instance, a scenario depicting exchange of signals to drive the vehicle.
- *Block Definition Diagram (BDD)* presents the definition and relationship of structural aspects of a system/entity. For instance, to represent the vehicle's structural decomposition.
- *Internal Block Diagram (IBD)* depicts the internal structure (interconnection and interactions) of the blocks used as parts. For instance, representing the interrelation and interaction of the vehicle with occupants and the external environment using ports and flows.
- *Parametric Diagram (PAR)* is used to capture the mathematical analysis using constraints and expressions. For instance, evaluating the vehicle stopping distance to compare design alternatives as well as to execute external engineering models [18].

1.2.4 Requirements Engineering

> "Requirements Engineering (RE) is concerned with discovering, eliciting, developing, analyzing, determining verification methods, validating, communicating, documenting, and managing requirements [19]."

The goal is to generate high-quality requirements that are complete, correct, consistent, traceable, and verifiable. According to Ref. [20], the subdisciplines of RE are:

Requirements Development—Requirements Development can be defined as the process of defining a project's scope, identifying user representatives for eliciting, analyzing, specifying, and validating requirements [21]. Its product is a set of documented requirements that defines some portion of the product to be built. The activities include eliciting shall statements, analyzing attributes, specifying characteristics, and validating the concerns.

Requirements Management—Requirements Management is concerned with gathering, organizing, expressing analyzing, reviewing, agreeing, tracking, changing, and validating requirement statements. It is a disciplined process that involves working with a defined set of requirements throughout the product's development process and its operational life. The activities include tracking requirements status, managing changes to requirements, controlling versions of requirements specifications, and tracing individual requirements [22].

1.2.5 Overview of Aircraft Thrust Reverser (TR) Technology using Cascade Type Thrust Reverser Actuation System (TRAS)

Thrust Reverser (TR) is a necessary commodity in most commercial aircraft to reduce the ground stopping distance after a touchdown in adverse conditions such as wet/icy, slippery runway by reversing the fan airflow. TRs also reduce brake wear and provide control under emergency scenarios like Refused Takeoff (RTO) and Aborted Landing (AL). Thrust Reverser Actuation Systems (TRAS) enhances operational safety by providing redundancy in levels of safety against accidental deployment during in-flight [23, 24]. TRAS power and control the deployment of aircraft thrust reversers, optimizing aircraft operational safety by minimizing runway stopping distances [25].

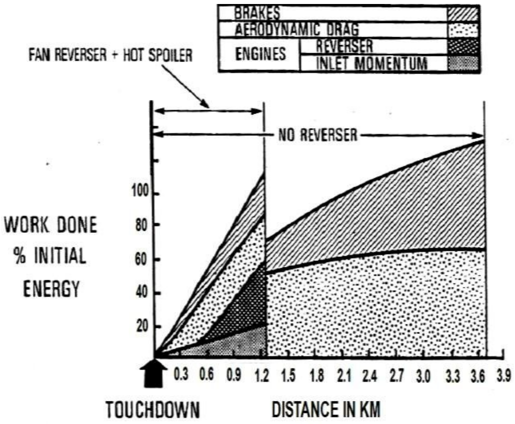


Figure 1.2: Effect of employing a thrust reverser on a wet/icy runway

Some key components integrated to assemble an operational TRAS product can be listed as follows [26, 27]:

- Self-locking and unlocking actuators provide the necessary actuation,
- Primary locks provide the safety during actuation,
- Tertiary cowl locks provide levels of safety and add a degree of redundancy,
- Motion synchronization system that synchronizes actuation,
- Manual drives for manual deployment during maintenance,

Figure 1.2 shows the effect of employing a thrust reverser in an aircraft to significantly reduce the stopping distance on a wet/icy runway. Figure 1.3 illustrates the basic working of a cascade type thrust reverser actuation system under normal operating conditions.

1.3 Thesis Organization

This thesis presents an investigated approach to understanding model-based implementation in an organization's system development practices. The modeling and research effort is focused on architecture model development for a generic aero-actuation system. The following chapters document how the outcomes and findings from the investigation tasks help address some of the inherent challenges in implementing MBSE and are summarized below:

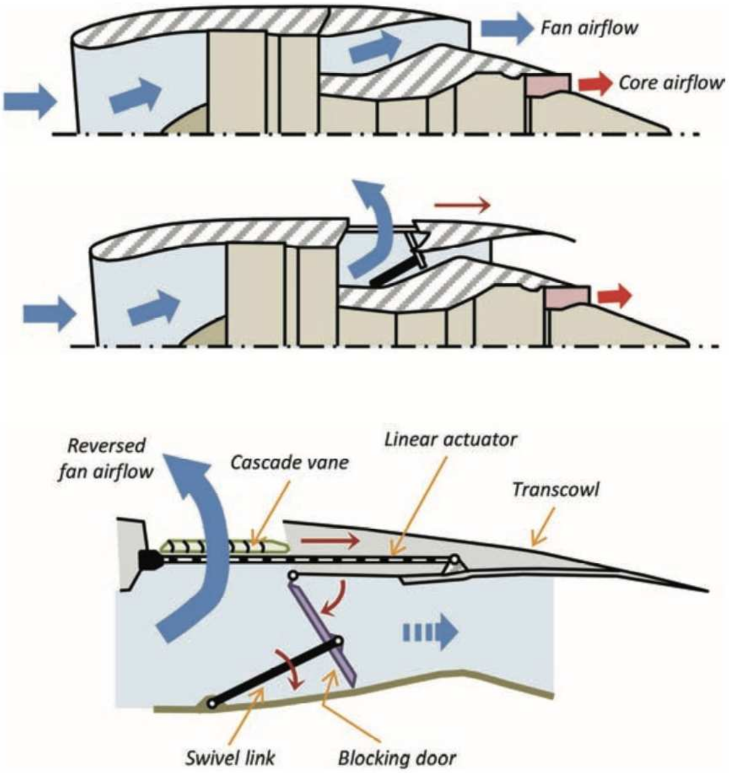


Figure 1.3: Thrust reverser according to the cascade concept(upper: stowed reverser; middle: deployed reverser; lower: main mechanical elements)

- Chapter 2 presents a modern approach to model perspectives of an organization's Product Life Cycle (PLC) development process, introducing the usage of MBSE tools, and proposes a data modeling approach to integrate a process model within the system architecture development framework.

- Chapter 3 focuses heavily on the MBSE tool usage to perform SE activities (implementation and analysis of results) like analysis of alternative architectures to formulate trade study and requirements verification. The outcome/product of this modeling effort is a formal system architecture model in SysML obtained after applying an MBSE methodology to the system of interest. The effort demonstrates an integrated modeling and simulation approach/environment.

- Chapter 4 presents some advanced tool usage techniques to address key pain points related to requirements development and management activities. The outcomes promote using model-based attributes to improve requirements definition and analysis with supporting evidence from elicitation activity, followed by rapid visualization of change propagation within requirements and model elements.

- Chapter 5 concludes the results obtained from the modeling and research effort, documenting the key implementation and the challenges addressed to achieve value and provide insight on the path forward in future work.

Chapter 2

Integration of Model-Based System Architecture with Early Phase System Development Processes

The primary outcome of the research and modeling effort discussed in this chapter is a process model for a technology/platform project. It is essential to note from the fundamental concepts described in Sec. 1.2 that MBSE practices are not only applied in the conceptual design phase at the front end but also continues in the development and later life cycle phases as the system evolves. Such practices allow the collaboration of multi-disciplinary teams and support management with better decision-making due to the early detection of faults and errors that can significantly mitigate project risks. The secondary outcome is a reusable/reference architecture framework and methodology for MBSE implementation. The tertiary outcome is a data modeling strategy to integrate phase deliverables from a formal system development process with system architecture artifacts from the system model.

2.1 Model the Formal Process of Idea Generation and Investigation of Objectives

Here, we will demonstrate the creation of a process model representing an organization's standards-compliant product life cycle (PLC) process for technology/platform projects. This modeling effort aims to understand the series of "phase" process activities and deliverables/outcomes as expected tangible work products that enable the system to progress in platform projects by identifying the enabling technologies.

A series of phase process activities that follow an agreed sequence is modeled. Each activity within the workflow is decomposed to elaborate a lower-level workflow of actions. Specific guidance is followed to execute the actions that are necessary to yield the expected deliverable/outcome.

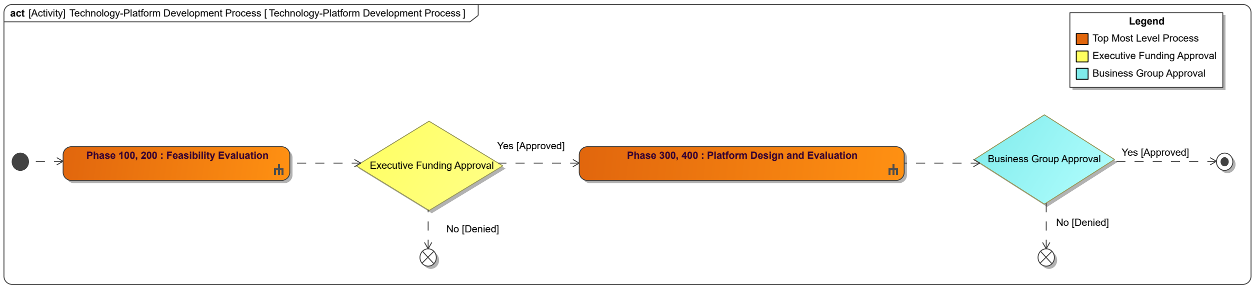

act [Activity] Technology-Platform Development Process [Technology-Platform Development Process]
Legend
Top Most Level Process
Executive Funding Approval
Business Group Approval
Phase 100, 200 : Feasibility Evaluation
Executive Funding Approval
Yes [Approved]
No [Denied]
Phase 300, 400 : Platform Design and Evaluation
Business Group Approval
Yes [Approved]
No [Denied]

Figure 2.1: Top-most level, technology/platform development process

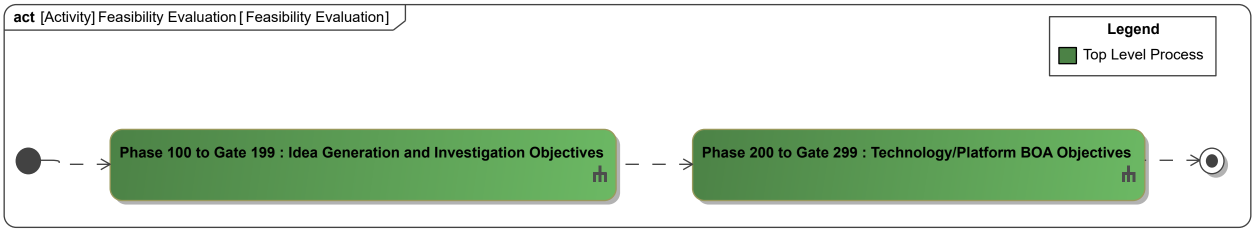

act [Activity] Feasibility Evaluation [Feasibility Evaluation]
Legend
Top Level Process
Phase 100 to Gate 199 : Idea Generation and Investigation Objectives
Phase 200 to Gate 299 : Technology/Platform BOA Objectives

Figure 2.2: Top level PLC process, feasibility evaluation

Each action is allocated to an accountable role that would be responsible for administering them. In this way, the inputs, process actions, and outputs are modeled for each phase activity.

2.1.1 Organize the Process Model

The process model is organized under the main package to capture the overall process and guidance. Nested packages are created to capture a library of process activities representing the phase, gate, and overall PLC processes for easy reusability.

2.1.2 Model the Phase/Gate-Based Activities

First, the topmost level process/workflow in the Technology/Platform Development Process is modeled as a SysML Activity Diagram as shown in Fig. 2.1. The scope of the modeling effort is limited to the early "Phase 100" of the PLC process, and modeling the subsequent phases is planned in future efforts. The feasibility evaluation process denoted Phase 100, 200 is decomposed and elaborated in the top-level process/workflow in an Activity Diagram as depicted in Fig. 2.2. Within the top-level process, the Idea Generation and Investigation Objectives, Phase 100 is further

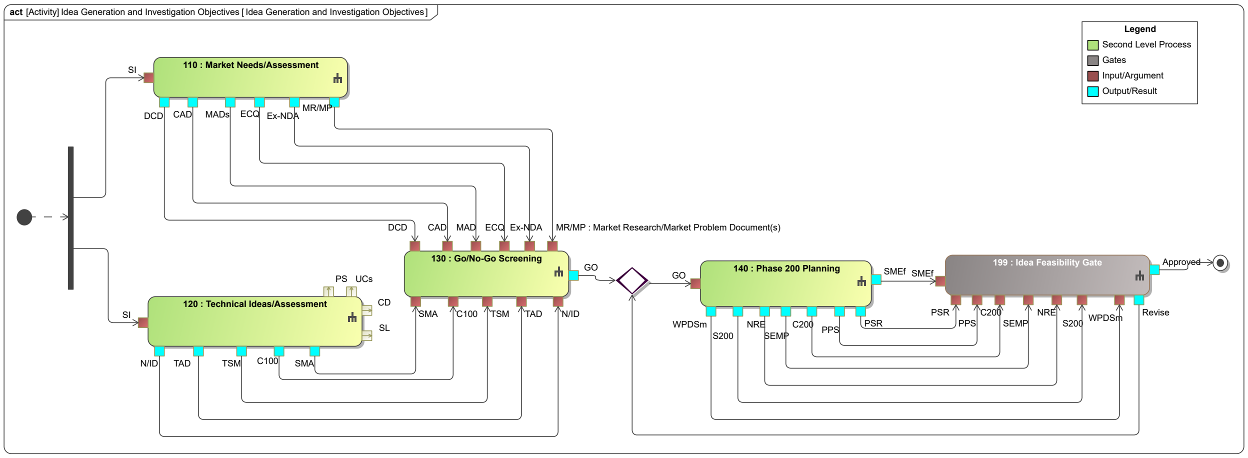


Figure 2.3: Second level process, idea generation and investigation objectives

decomposed down to the second level process/workflow, which remains the main focus in this chapter.

As illustrated in Fig. 2.3, the idea generation and investigation objectives (Phase 100) is elaborated with a series of phase activities. Starting with *Market Needs/Assessment* and *Technical Ideas/Assessment*, the new projects are identified and further reviewed under *Go/No-Go Screening* to select a project that progresses to Phase 200. The decision follows the *Phase 200 Planning* activity to obtain other planning deliverables before entering the *Idea Feasibility Gate* where a final decision is exercised to obtain approval on the commitment of resources from management.

Each phase activity described above is elaborated with a series of specific actions that yield tangible work products, termed as a deliverable/outcome. The deliverable/outcome is modeled as a data object in the activity diagram to instill a placeholder where data/information content will be stored and managed [9]. These outcomes are provided as data input to the following phase activities. Thus, as more information/data becomes available in later stages, it is updated and managed from a single model repository.

Each action is allocated as to an accountable role/stakeholder that is responsible for executing them.

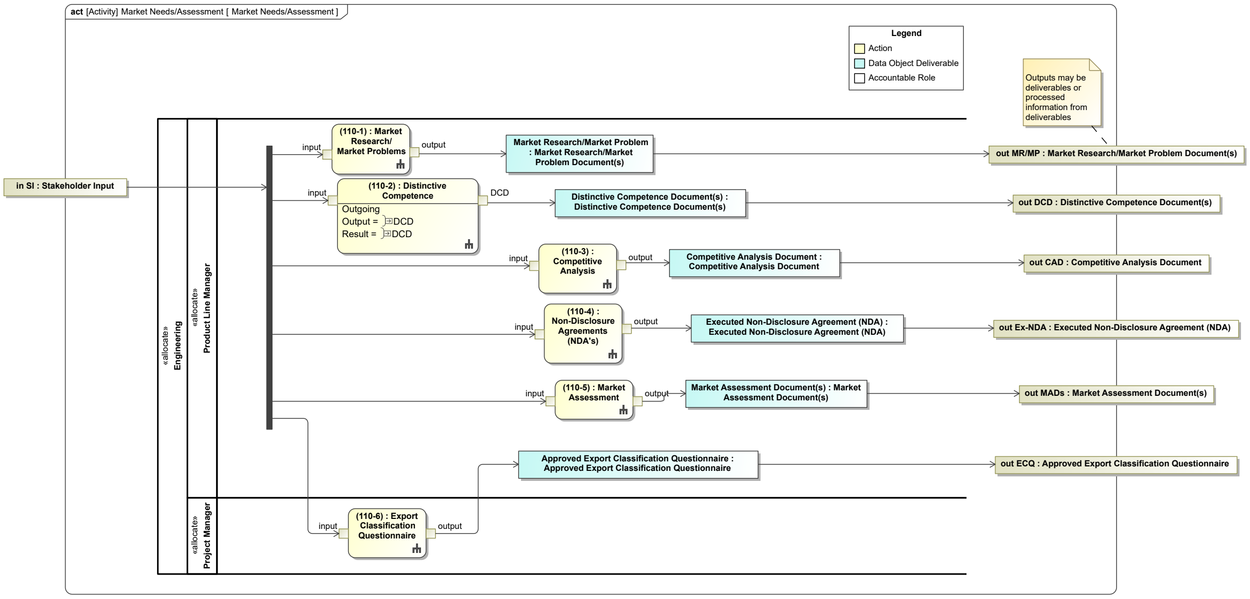

act [Activity] Market Needs/Assessment [Market Needs/Assessment]
Legend
Action
Data Object Deliverable
Accountable Role
Outputs may be deliverables or processed information from deliverables
in SI : Stakeholder Input
«allocate» Engineering
«allocate» Product Line Manager
«allocate» Project Manager
input
(110-1) : Market Research/ Market Problems
output
Market Research/Market Problem : Market Research/Market Problem Document(s)
out MR/MP : Market Research/Market Problem Document(s)
input
(110-2) : Distinctive Competence
Outgoing
Output = DCD
Result = DCD
DCD
Distinctive Competence Document(s) : Distinctive Competence Document(s)
out DCD : Distinctive Competence Document(s)
input
(110-3) : Competitive Analysis
output
Competitive Analysis Document : Competitive Analysis Document
out CAD : Competitive Analysis Document
input
(110-4) : Non-Disclosure Agreements (NDA's)
output
Executed Non-Disclosure Agreement (NDA) : Executed Non-Disclosure Agreement (NDA)
out Ex-NDA : Executed Non-Disclosure Agreement (NDA)
input
(110-5) : Market Assessment
output
Market Assessment Document(s) : Market Assessment Document(s)
out MADs : Market Assessment Document(s)
Approved Export Classification Questionnaire : Approved Export Classification Questionnaire
out ECQ : Approved Export Classification Questionnaire
input
(110-6) : Export Classification Questionnaire
output

Figure 2.4: Market needs/assessment, activity 110

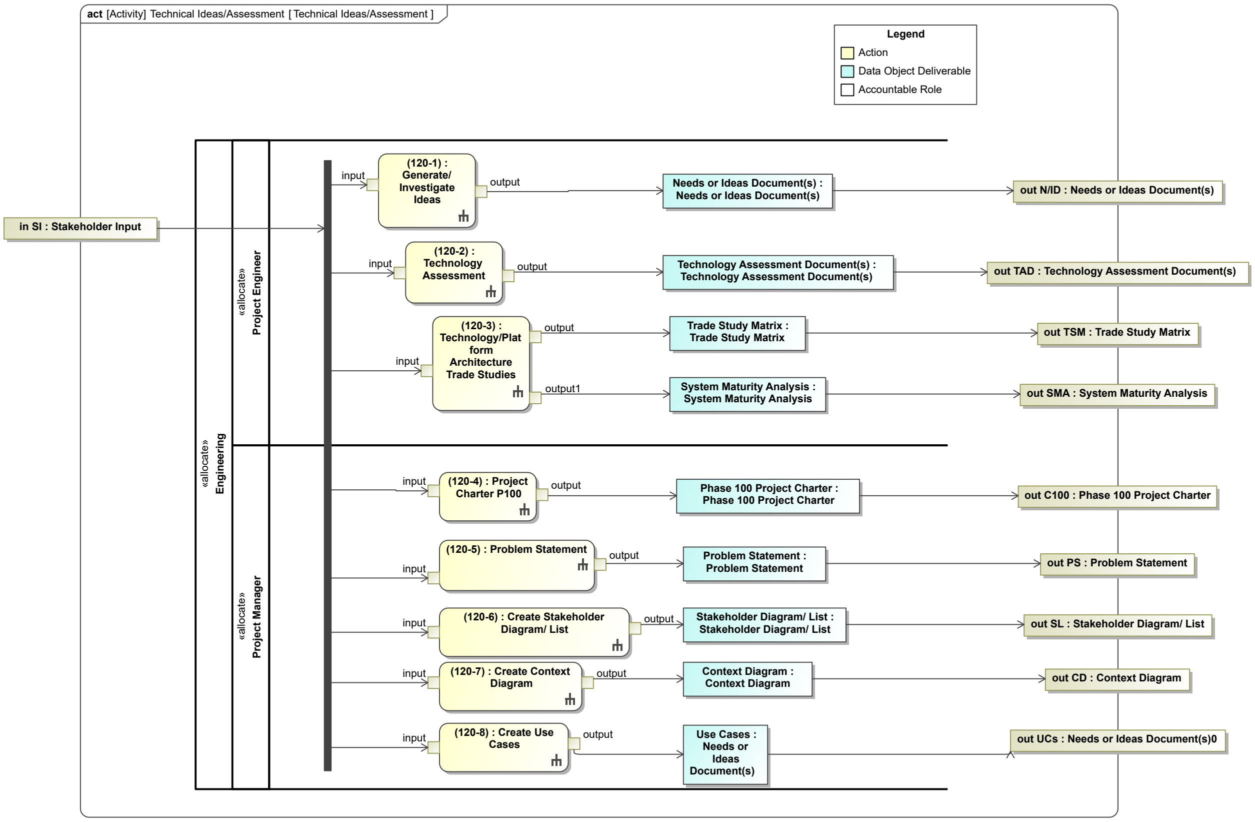

act [Activity] Technical Ideas/Assessment [Technical Ideas/Assessment]
Legend
Action
Data Object Deliverable
Accountable Role
in SI : Stakeholder Input
«allocate» Engineering
«allocate» Project Engineer
«allocate» Project Manager
input
(120-1) : Generate/ Investigate Ideas
output
Needs or Ideas Document(s) : Needs or Ideas Document(s)
out N/ID : Needs or Ideas Document(s)
input
(120-2) : Technology Assessment
output
Technology Assessment Document(s) : Technology Assessment Document(s)
out TAD : Technology Assessment Document(s)
input
(120-3) : Technology/Plat form Architecture Trade Studies
output
Trade Study Matrix : Trade Study Matrix
out TSM : Trade Study Matrix
output1
System Maturity Analysis : System Maturity Analysis
out SMA : System Maturity Analysis
input
(120-4) : Project Charter P100
output
Phase 100 Project Charter : Phase 100 Project Charter
out C100 : Phase 100 Project Charter
input
(120-5) : Problem Statement
output
Problem Statement : Problem Statement
out PS : Problem Statement
input
(120-6) : Create Stakeholder Diagram/ List
output
Stakeholder Diagram/ List : Stakeholder Diagram/ List
out SL : Stakeholder Diagram/ List
input
(120-7) : Create Context Diagram
output
Context Diagram : Context Diagram
out CD : Context Diagram
input
(120-8) : Create Use Cases
output
Use Cases : Needs or Ideas Document(s)
out UCs : Needs or Ideas Document(s)0

Figure 2.5: Technical ideas/assessment, activity 120

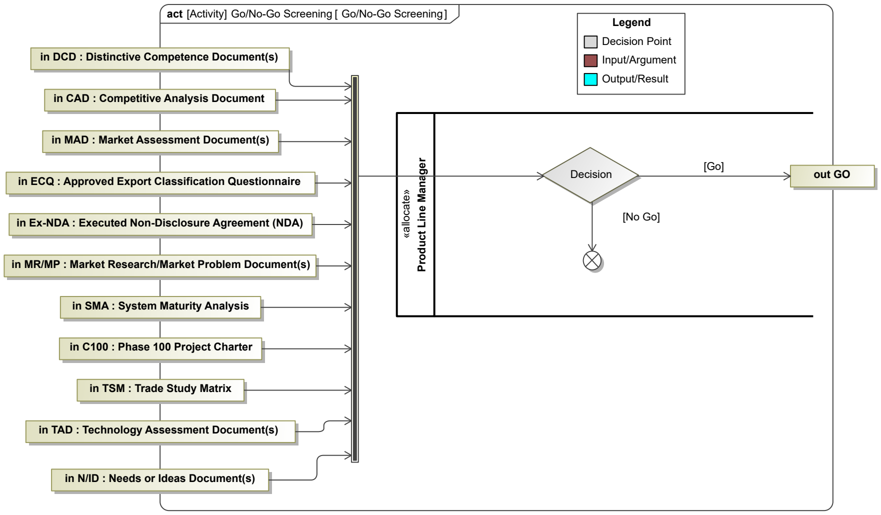

act [Activity] Go/No-Go Screening [Go/No-Go Screening]
Legend
Decision Point
Input/Argument
Output/Result
in DCD : Distinctive Competence Document(s)
in CAD : Competitive Analysis Document
in MAD : Market Assessment Document(s)
in ECQ : Approved Export Classification Questionnaire
in Ex-NDA : Executed Non-Disclosure Agreement (NDA)
in MR/MP : Market Research/Market Problem Document(s)
in SMA : System Maturity Analysis
in C100 : Phase 100 Project Charter
in TSM : Trade Study Matrix
in TAD : Technology Assessment Document(s)
in N/ID : Needs or Ideas Document(s)
«allocate»
Product Line Manager
Decision
[Go]
[No Go]
out GO

Figure 2.6: Go/No-Go screening, activity 130

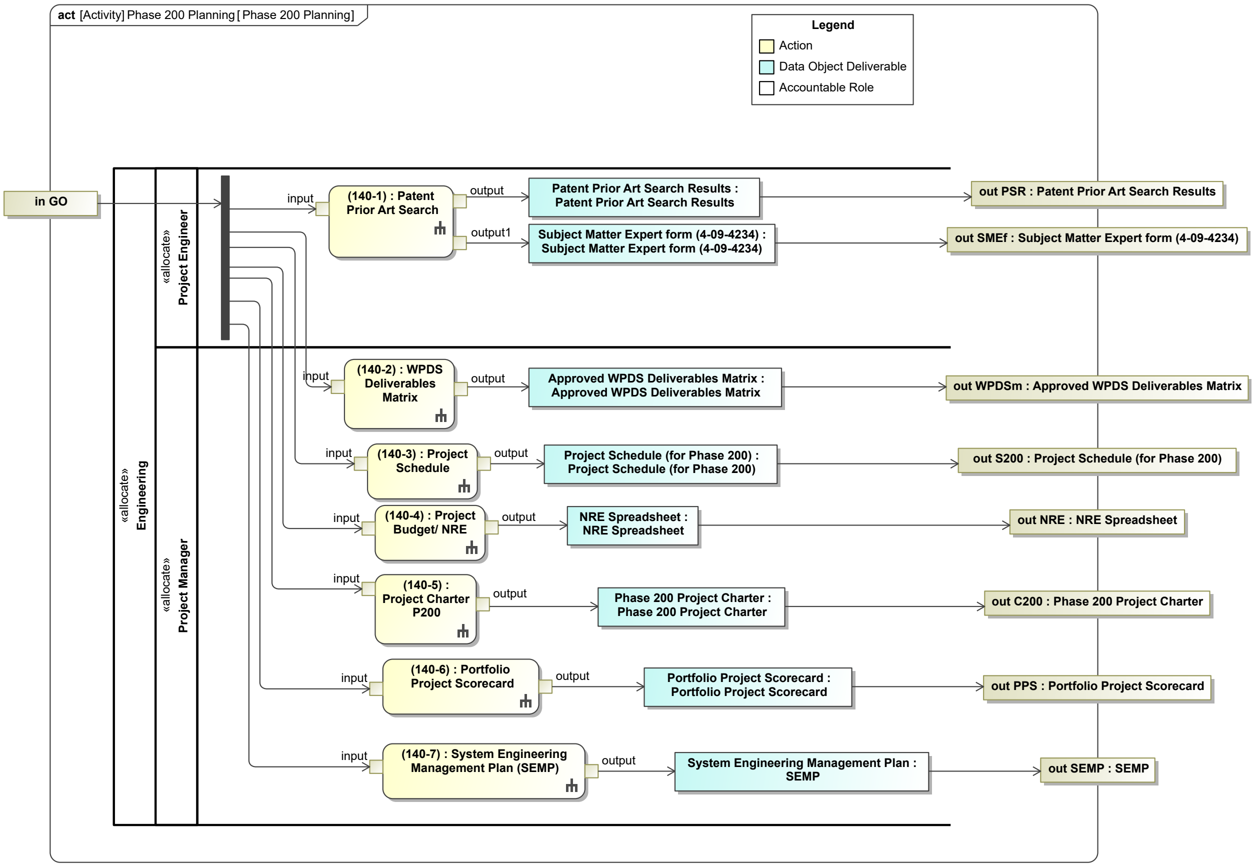

act [Activity] Phase 200 Planning [Phase 200 Planning]
Legend
Action
Data Object Deliverable
Accountable Role
in GO
«allocate»
Engineering
«allocate»
Project Engineer
«allocate»
Project Manager
input
(140-1) : Patent Prior Art Search
output
output1
Patent Prior Art Search Results : Patent Prior Art Search Results
Subject Matter Expert form (4-09-4234) : Subject Matter Expert form (4-09-4234)
out PSR : Patent Prior Art Search Results
out SMEf : Subject Matter Expert form (4-09-4234)
input
(140-2) : WPDS Deliverables Matrix
output
Approved WPDS Deliverables Matrix : Approved WPDS Deliverables Matrix
out WPDSm : Approved WPDS Deliverables Matrix
input
(140-3) : Project Schedule
output
Project Schedule (for Phase 200) : Project Schedule (for Phase 200)
out S200 : Project Schedule (for Phase 200)
input
(140-4) : Project Budget/ NRE
output
NRE Spreadsheet : NRE Spreadsheet
out NRE : NRE Spreadsheet
input
(140-5) : Project Charter P200
output
Phase 200 Project Charter : Phase 200 Project Charter
out C200 : Phase 200 Project Charter
input
(140-6) : Portfolio Project Scorecard
output
Portfolio Project Scorecard : Portfolio Project Scorecard
out PPS : Portfolio Project Scorecard
input
(140-7) : System Engineering Management Plan (SEMP)
output
System Engineering Management Plan : SEMP
out SEMP : SEMP

Figure 2.7: Phase 200 planning, activity 140

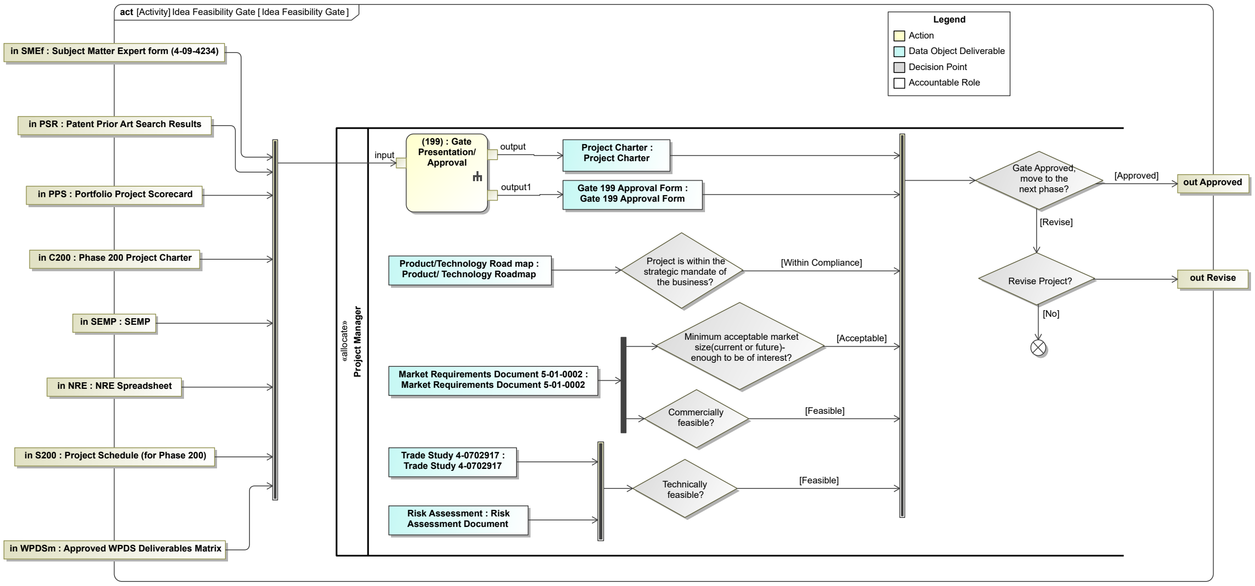

act [Activity] Idea Feasibility Gate [Idea Feasibility Gate]
Legend
Action
Data Object Deliverable
Decision Point
Accountable Role
in SMEf : Subject Matter Expert form (4-09-4234)
in PSR : Patent Prior Art Search Results
in PPS : Portfolio Project Scorecard
in C200 : Phase 200 Project Charter
in SEMP : SEMP
in NRE : NRE Spreadsheet
in S200 : Project Schedule (for Phase 200)
in WPDSm : Approved WPDS Deliverables Matrix
«allocate»
Project Manager
input
(199) : Gate Presentation/ Approval
output
output1
Project Charter : Project Charter
Gate 199 Approval Form : Gate 199 Approval Form
Product/Technology Road map : Product/ Technology Roadmap
Project is within the strategic mandate of the business?
[Within Compliance]
Market Requirements Document 5-01-0002 : Market Requirements Document 5-01-0002
Minimum acceptable market size(current or future)- enough to be of interest?
[Acceptable]
Commercially feasible?
[Feasible]
Trade Study 4-0702917 : Trade Study 4-0702917
Risk Assessment : Risk Assessment Document
Technically feasible?
[Feasible]
Gate Approved, move to the next phase?
[Approved]
out Approved
[Revise]
Revise Project?
out Revise
[No]

Figure 2.8: Idea Feasibility Gate, activity 199

The diagrams shown in Figs. 2.4, 2.5, 2.6, 2.7, and 2.8 illustrate the details of the phase activities and outcomes. The overall process model decomposition is summarised in Fig. 2.9 by creating a SysML Activity Decomposition Map (a derived view of the model).

2.1.3 N-Squared Analysis

After modeling the early phase processes of the PLC, the input-process-output of the workflow is analyzed by creating an N-squared diagram as depicted in Fig. 2.10. N-squared analysis enables understanding necessary functional interfaces and input/output relationships by identifying multi-level indirect relations. A customized relationship criterion is established to query the model and represent the relationships in a matrix structure. The objective is to clearly understand and quickly read the inputs and outputs for every phase/gate (as currently modeled) within the PLC process model. The criteria of relationships are modified to simplify the original diagram that reflects only the second level process.

Here, a series of related properties are tailored with meta-chain navigation to represent indirectly related functions.

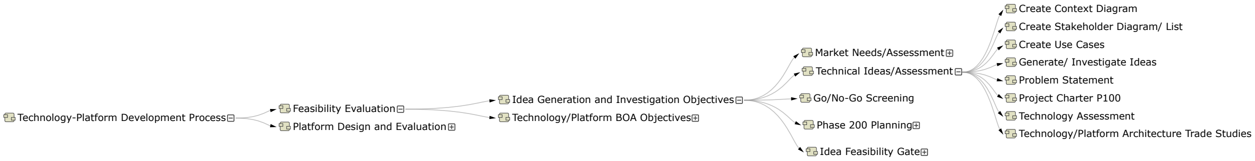
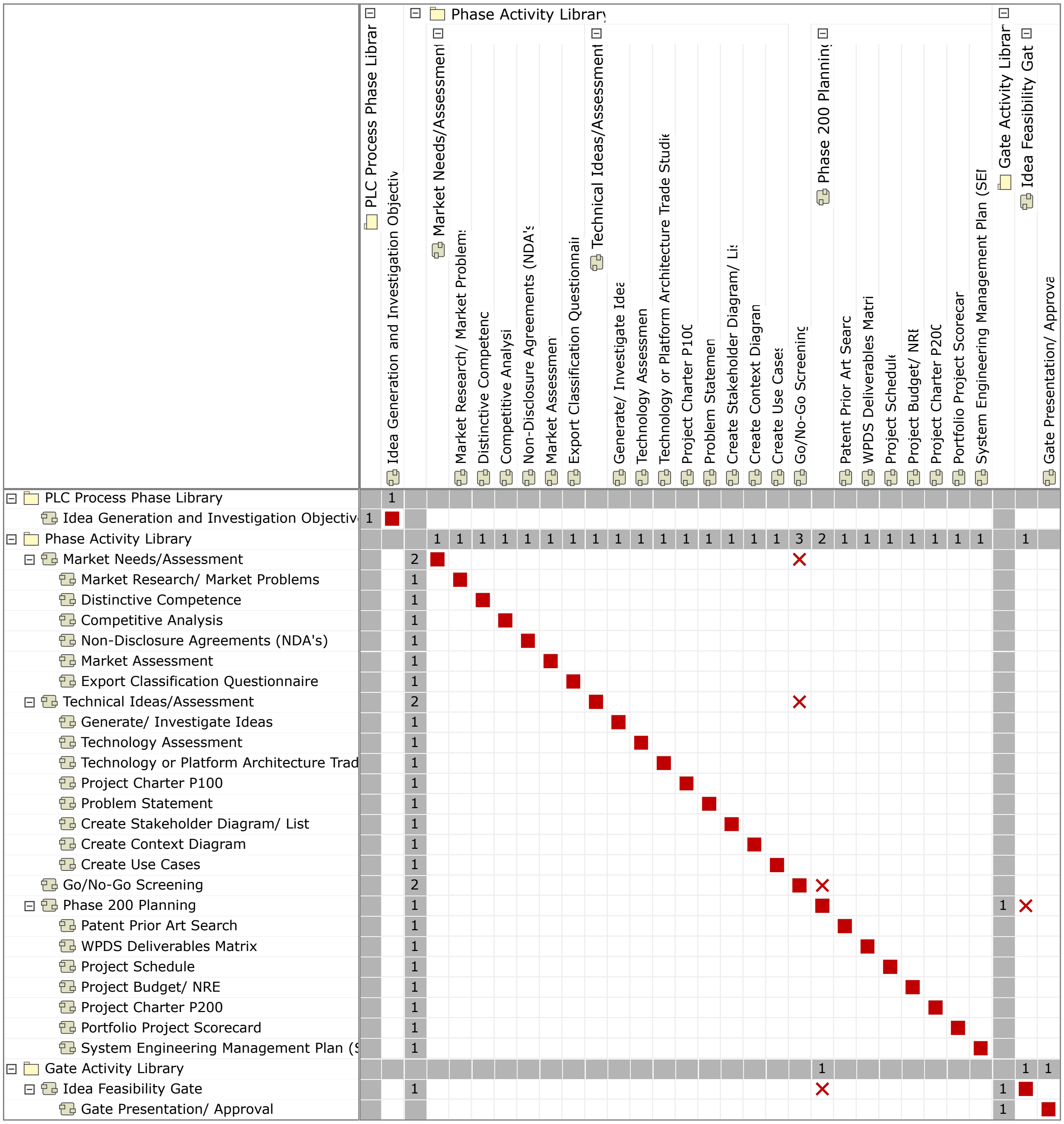
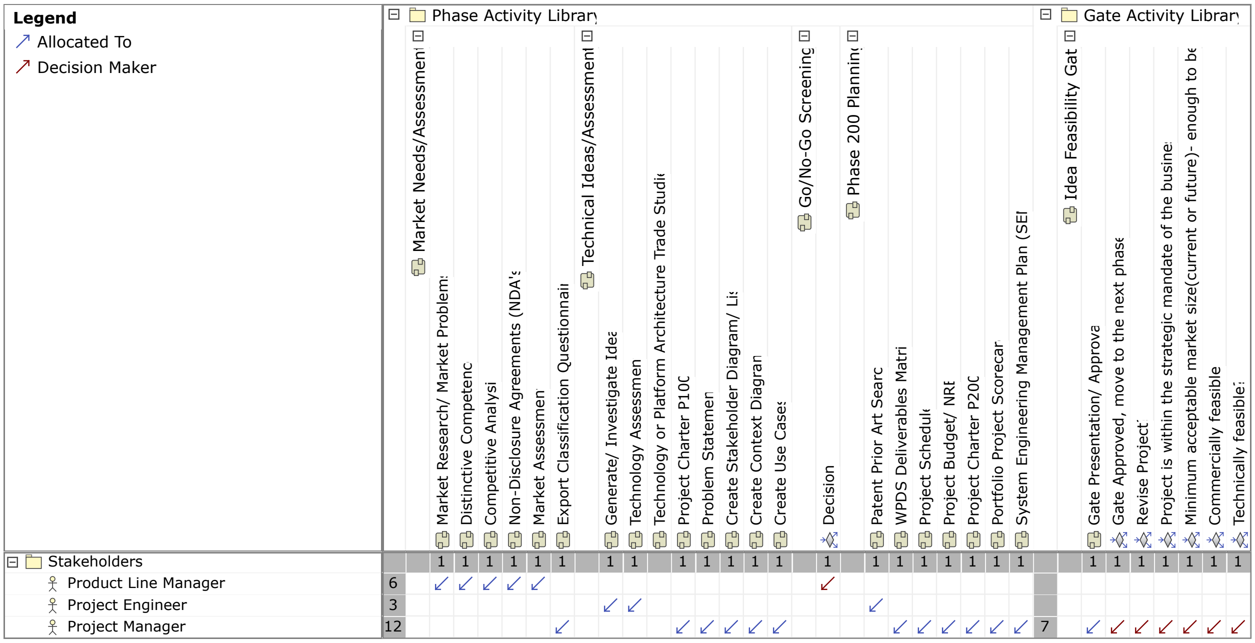

Technology-Platform Development Process
Feasibility Evaluation
Platform Design and Evaluation
Idea Generation and Investigation Objectives
Technology/Platform BOA Objectives
Market Needs/Assessment
Technical Ideas/Assessment
Go/No-Go Screening
Phase 200 Planning
Idea Feasibility Gate
Create Context Diagram
Create Stakeholder Diagram/ List
Create Use Cases
Generate/ Investigate Ideas
Problem Statement
Project Charter P100
Technology Assessment
Technology/Platform Architecture Trade Studies

Figure 2.9: Overview of the technology-platform development process model activities



	PLC Process Phase Librar	Idea Generation and Investigation Objectiv	Phase Activity Library	Market Needs/Assessment	Market Research/ Market Problem:	Distinctive Competenc	Competitive Analysi	Non-Disclosure Agreements (NDA's	Market Assessmen	Export Classification Questionnai	Technical Ideas/Assessment	Generate/ Investigate Idea	Technology Assessmen	Technology or Platform Architecture Trade Studie	Project Charter P10C	Problem Statemen	Create Stakeholder Diagram/ Lis	Create Context Diagran	Create Use Cases	Go/No-Go Screening	Phase 200 Plannin	Patent Prior Art Searc	WPDS Deliverables Matri	Project Schedule	Project Budget/ NRE	Project Charter P20C	Portfolio Project Scorecar	System Engineering Management Plan (SEI	Gate Activity Librar	Idea Feasibility Gat	Gate Presentation/ Approva
PLC Process Phase Library		1																													
Idea Generation and Investigation Objectiv	1	🟥																													
Phase Activity Library				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3	2	1	1	1	1	1	1	1		1	
Market Needs/Assessment			2	🟥																✕											
Market Research/ Market Problems			1		🟥																										
Distinctive Competence			1			🟥																									
Competitive Analysis			1				🟥																								
Non-Disclosure Agreements (NDA's)			1					🟥																							
Market Assessment			1						🟥																						
Export Classification Questionnaire			1							🟥																					
Technical Ideas/Assessment			2								🟥									✕											
Generate/ Investigate Ideas			1									🟥																			
Technology Assessment			1										🟥																		
Technology or Platform Architecture Trad			1											🟥																	
Project Charter P100			1												🟥																
Problem Statement			1													🟥															
Create Stakeholder Diagram/ List			1														🟥														
Create Context Diagram			1															🟥													
Create Use Cases			1																🟥												
Go/No-Go Screening			2																	🟥	✕										
Phase 200 Planning			1																		🟥								1	✕	
Patent Prior Art Search			1																			🟥									
WPDS Deliverables Matrix			1																				🟥								
Project Schedule			1																					🟥							
Project Budget/ NRE			1																						🟥						
Project Charter P200			1																							🟥					
Portfolio Project Scorecard			1																								🟥				
System Engineering Management Plan (S			1																									🟥			
Gate Activity Library																					1									1	1
Idea Feasibility Gate			1																		✕								1	🟥	
Gate Presentation/ Approval																													1		🟥

Figure 2.10: N-squared diagram for the idea generation and investigation objectives process



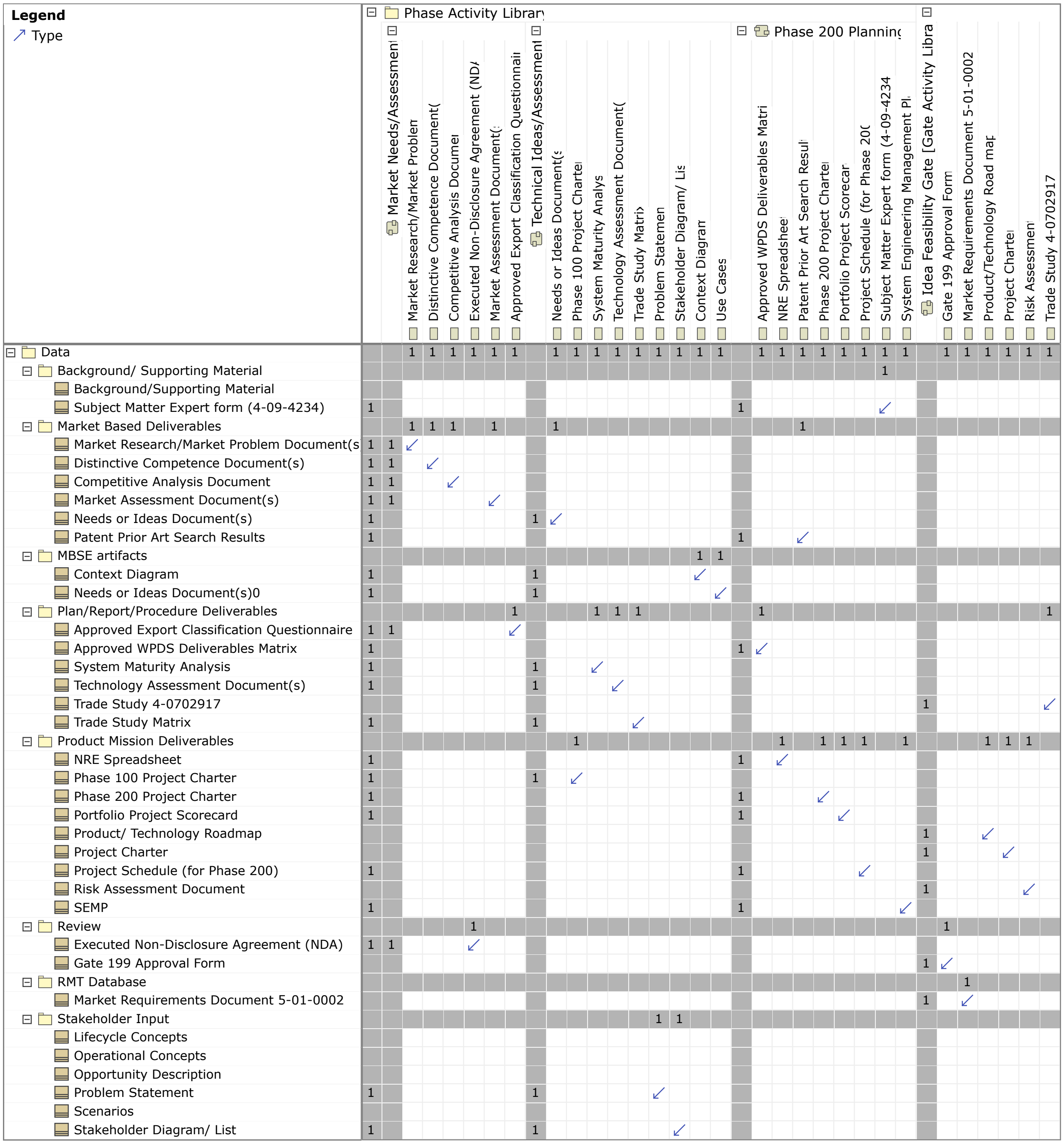
Legend
↗ Allocated To
↗ Decision Maker

	Phase Activity Library	Market Needs/Assessment	Market Research/ Market Problems	Distinctive Competenc	Competitive Analysi	Non-Disclosure Agreements (NDA's	Market Assessmen	Export Classification Questionnai	Technical Ideas/Assessment	Generate/ Investigate Idea	Technology Assessmen	Technology or Platform Architecture Trade Studi	Project Charter P10C	Problem Statemen	Create Stakeholder Diagram/ Lis	Create Context Diagran	Create Use Cases	Go/No-Go Screening	Decision	Phase 200 Plannin	Patent Prior Art Searc	WPDS Deliverables Matri	Project Schedul	Project Budget/ NRE	Project Charter P20C	Portfolio Project Scorecar	System Engineering Management Plan (SEI	Gate Activity Library	Idea Feasibility Gat	Gate Presentation/ Approva	Gate Approved, move to the next phase	Revise Project	Project is within the strategic mandate of the busine	Minimum acceptable market size(current or future)- enough to be	Commercially feasible	Technically feasible
Stakeholders			1	1	1	1	1	1		1	1		1	1	1	1	1		1		1	1	1	1	1	1	1			1	1	1	1	1	1	1
Product Line Manager	6		↙	↙	↙	↙	↙												↙																	
Project Engineer	3									↙	↙										↙															
Project Manager	12							↙					↙	↙	↙	↙	↙					↙	↙	↙	↙	↙	↙	7		↙	↙	↙	↙	↙	↙	↙

Figure 2.11: List of stakeholders involved in decision making at the phase process activities

A data modeling strategy is established to create/retrieve/update/delete data in an expected deliverable by channelling the information content from a unified system data source [28]. This is achieved by creating Data Objects as model elements to include in activities where it stores or conveys data items during process execution. For instance, within the *Technical Ideas/Assessment* phase activity, the *Project Manager* is accountable for administering the action of creating a stakeholder list to obtain the deliverable–*Stakeholder List*. Figure 2.13 shows the stakeholder list as a table generated within the system model that is channeled to the phase activity through the data object deliverable [11].

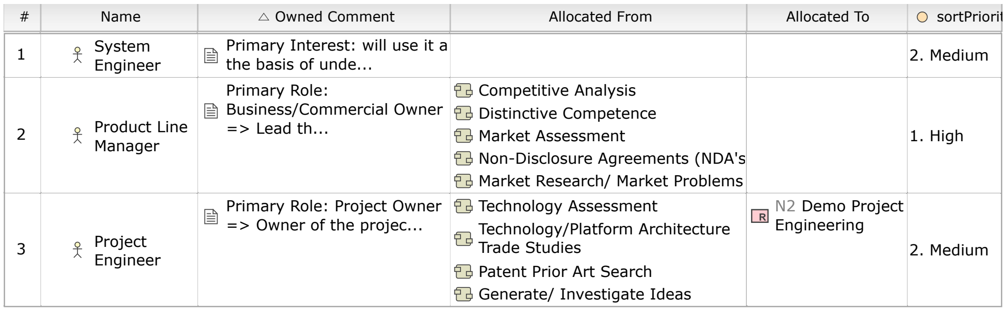
Consequently, this modeling effort addresses *(I1)* by exposing views of the process model and demonstrates capabilities of a model-centric approach that can transform the culture of the document-centric practices.



Legend
↗ Type

Data	Phase Activity Librar	Market Needs/Assessmen	Market Research/Market Problen	Distinctive Competence Document(	Competitive Analysis Documer	Executed Non-Disclosure Agreement (ND/	Market Assessment Document(	Approved Export Classification Questionnai	Technical Ideas/Assessment	Needs or Ideas Document(	Phase 100 Project Charte	System Maturity Analys	Technology Assessment Document(	Trade Study Matrix	Problem Statemen	Stakeholder Diagram/ Lis	Context Diagram	Use Cases	Phase 200 Planning	Approved WPDS Deliverables Matri	NRE Spreadshee	Patent Prior Art Search Resul	Phase 200 Project Charte	Portfolio Project Scorecar	Project Schedule (for Phase 20(	Subject Matter Expert form (4-09-4234	System Engineering Management Pl	Idea Feasibility Gate [Gate Activity Libra	Gate 199 Approval Form	Market Requirements Document 5-01-0002	Product/Technology Road map	Project Charte	Risk Assessmen	Trade Study 4-0702917
Data			1	1	1	1	1	1		1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1		1	1	1	1	1	1
Background/ Supporting Material																										1								
Background/Supporting Material																																		
Subject Matter Expert form (4-09-4234)	1																		1							↙								
Market Based Deliverables			1	1	1		1			1												1												
Market Research/Market Problem Document(s	1	1	↙																															
Distinctive Competence Document(s)	1	1		↙																														
Competitive Analysis Document	1	1			↙																													
Market Assessment Document(s)	1	1					↙																											
Needs or Ideas Document(s)	1								1	↙																								
Patent Prior Art Search Results	1																		1			↙												
MBSE artifacts																	1	1																
Context Diagram	1								1								↙																	
Needs or Ideas Document(s)0	1								1									↙																
Plan/Report/Procedure Deliverables								1				1	1	1						1														1
Approved Export Classification Questionnaire	1	1						↙																										
Approved WPDS Deliverables Matrix	1																		1	↙														
System Maturity Analysis	1								1			↙																						
Technology Assessment Document(s)	1								1				↙																					
Trade Study 4-0702917																												1						↙
Trade Study Matrix	1								1					↙																				
Product Mission Deliverables											1										1		1	1	1		1				1	1	1	
NRE Spreadsheet	1																		1		↙													
Phase 100 Project Charter	1								1		↙																							
Phase 200 Project Charter	1																		1				↙											
Portfolio Project Scorecard	1																		1					↙										
Product/ Technology Roadmap																												1			↙			
Project Charter																												1				↙		
Project Schedule (for Phase 200)	1																		1						↙									
Risk Assessment Document																												1					↙	
SEMP	1																		1								↙							
Review						1																							1					
Executed Non-Disclosure Agreement (NDA)	1	1				↙																												
Gate 199 Approval Form																												1	↙					
RMT Database																														1				
Market Requirements Document 5-01-0002																												1		↙				
Stakeholder Input															1	1																		
Lifecycle Concepts																																		
Operational Concepts																																		
Opportunity Description																																		
Problem Statement	1								1						↙																			
Scenarios																																		
Stakeholder Diagram/ List	1								1							↙																		

Figure 2.12: Defining data for data objects tied within the phase activities



#	Name	△ Owned Comment	Allocated From	Allocated To	○ sortPriorit
1	System Engineer	Primary Interest: will use it a the basis of unde...			2. Medium
2	Product Line Manager	Primary Role: Business/Commercial Owner => Lead th...	Competitive Analysis Distinctive Competence Market Assessment Non-Disclosure Agreements (NDA's Market Research/ Market Problems		1. High
3	Project Engineer	Primary Role: Project Owner => Owner of the projec...	Technology Assessment Technology/Platform Architecture Trade Studies Patent Prior Art Search Generate/ Investigate Ideas	R N2 Demo Project Engineering	2. Medium

Figure 2.13: Create stakeholder diagram/list deliverable

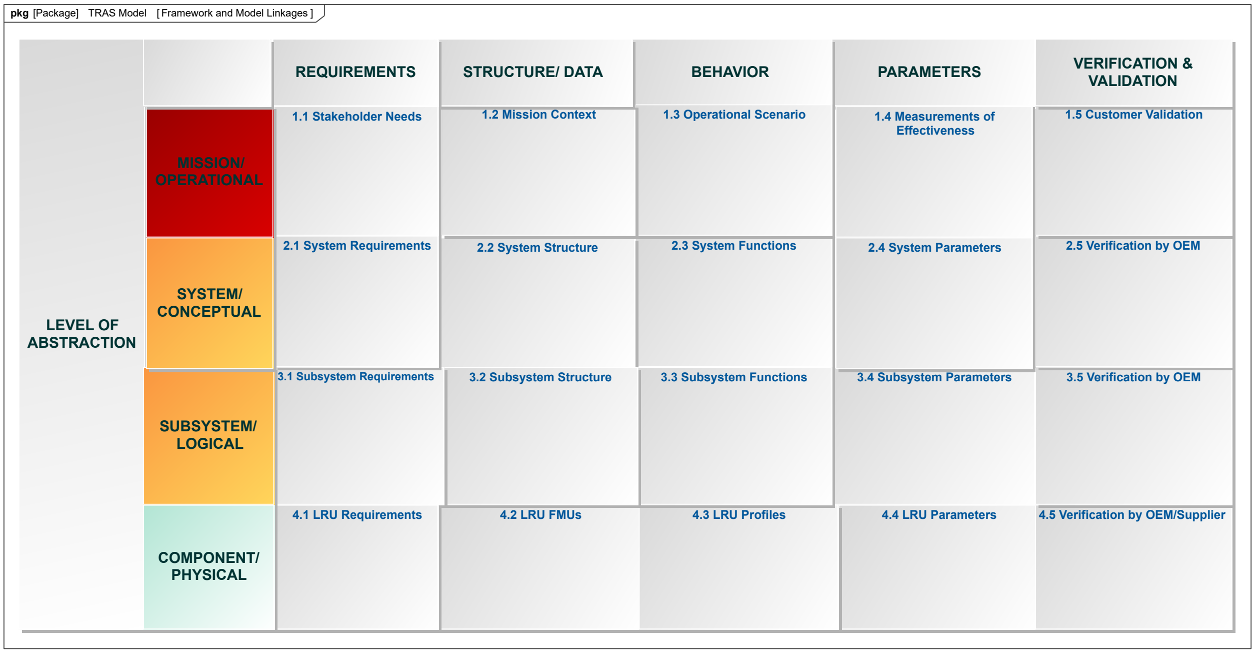
2.2 Augment an MBSE Framework and Methodology for System Architecture Development

This section presents a generic architecture framework and architecting methodology for applying MBSE principles to a complex system. The primary objective of augmenting such an approach is to understand what architecture content should be defined and how it would be created through a well-defined methodology. The proposed approach is repeatable and reusable for the development of a system architecture model.

From a holistic standpoint, architecture is concerned with maintaining a broad-base understanding of the whole system-of-interest without constraining the implementation to a specific solution boundary. An architecture model can be viewed as the foundation for a consistent, controlled, high-quality data-centric SE process for replacing document-centric processes. The model also provides support at implementing upgrades, technology refreshment, and sustainment throughout the life cycle. In order to perform SE activities on the foundation of such an architecture enabled/architecture-centric model, we need both:

1. An architecture framework defined using a structured approach to visualize and capture a complex system in terms of the key system aspects (as per the language standard, SysML) while considering appropriate abstraction as discussed in Sec. 1.2.

2. An architecting methodology presented with specific architecture content follows a structured approach (What to create? When to create and in what sequence?).



pkg [Package] TRAS Model [Framework and Model Linkages]

LEVEL OF ABSTRACTION		REQUIREMENTS	STRUCTURE/ DATA	BEHAVIOR	PARAMETERS	VERIFICATION & VALIDATION
	MISSION/ OPERATIONAL	1.1 Stakeholder Needs	1.2 Mission Context	1.3 Operational Scenario	1.4 Measurements of Effectiveness	1.5 Customer Validation
	SYSTEM/ CONCEPTUAL	2.1 System Requirements	2.2 System Structure	2.3 System Functions	2.4 System Parameters	2.5 Verification by OEM
	SUBSYSTEM/ LOGICAL	3.1 Subsystem Requirements	3.2 Subsystem Structure	3.3 Subsystem Functions	3.4 Subsystem Parameters	3.5 Verification by OEM
	COMPONENT/ PHYSICAL	4.1 LRU Requirements	4.2 LRU FMUs	4.3 LRU Profiles	4.4 LRU Parameters	4.5 Verification by OEM/Supplier

Figure 2.14: Augmenting a generic approach for MBSE adoption/implementation

The architecture model can be developed by augmenting a comprehensive system modeling strategy that captures the operational environment, magnifies the system-of-interest to focus on functional capabilities, decomposes them for specifying the logical design, and finally synthesizes to in-depth physical/point design. Such an approach shall contribute towards constructing a complete, clear and unambiguous, thorough abstract representation of a complex system that will manifest in a unified model repository or more often referred to as a "digital thread".

2.2.1 System Architecture Modeling Framework

As illustrated in Fig. 2.14, a generic framework is formulated for applying the MBSE philosophy and tools in the pursuit of manifesting a formal system architecture model. The proposed framework is organized in terms of structured *rows* and *columns* intersecting at *cells* to specify the rules and guidance for a consistent modeling workflow. The inspiration for this matrix style framework is adopted from the Zachman Framework for Enterprise Architecture$^{\text{TM}}$ [29, 30].

The first four *columns*, namely *Requirements*, *Structure/Data*, *Behavior*, and *Constraints/Parameters* depict the key aspects for specifying a system in compliance with the language standard, SysML. The individual *rows* within this framework represents a classifier in terms of the level/layer of abstraction. Each level provides a basis for an abstract representation of a set of system entities in terms of shared characteristics. The proposed philosophy of modeling the aspects of system hierarchy/environment within the levels of abstraction helps realize the progression of a system architecture from abstract to detailed as more information becomes available. The resulting views are captured within the *cells* of the matrix. A *cell* represents a view that captures instance(s) of system technical specification in terms of SysML diagram(s) referred to as *artifact(s)*. This approach enables visualizing the views of a complex system and capturing it as architecture content that follows a well-defined modeling workflow [7].

It is important to note that a fifth column, *Verification and Validation* is introduced as an independent aspect for evaluating and verifying compliance of architecture content with requirements and validating against stakeholder concerns. The verification activities within this column also support attaining agreement with the stakeholders on the verification methods.

As discussed earlier, to develop a complete digital thread of a system, the modeling effort must follow a well-defined architecting methodology. The methodology helps understand what model elements need to be created in each activity of the modeling workflow and provides detailed guidance on how they should be interlinked, thus creating dependencies across model elements and artifacts. The framework is structured to capture the key SE activities/outcomes by following a specified sequence that constitutes the modeling workflow to develop a system architecture model. The following section elaborates these activities and describes the kind of artifacts that should be created as model content.

2.2.2 System Architecting Methodology

An architecting methodology for applying MBSE is described in terms of a comprehensive workflow that follows a sequence of the modeling activities as depicted in Fig. 2.14. Each modeling activity is directed with a purpose that contributes value to the SE effort. Architecture content for each *cell* of the *matrix* is identified by investigating artifacts needed to accomplish the specific

SE tasks. Suitable relationships/dependencies are established across model elements. A detailed description for each modeling activity (What goes in each cell? What is the definition and guidance to create them?) is discussed in the following paragraphs.

1. **Mission/Operational Level**

   The key purpose of modeling effort at this level is to transform the needs and requirements into the architecture context by partitioning them into structure, behavior, and constraint aspects.

1.1 Stakeholder Needs:

The first activity, stakeholder needs analysis, begins with capturing the stakeholder's information, including their names, level of interest, and influence. A clear definition of the system and context boundaries will supports the correct definition of requirements.

1.2 Mission Context:

The operational structure is defined with the top-level partitioning of the system, its users, and external systems that participate within the system context. This activity involves intensive communication with stakeholders where multi-disciplinary teams apply holistic systems thinking approaches to visualize the different aspects and facets/dimensions of the system-of-interest near its boundary. Thinking about the system from different mindsets helps establish a problem statement that stimulates the desire for a solution or commonly referred to as a solving system shall help identify the aspects of the system context. The system context facilitates a quick representation of the vision to all the cross-disciplines for answering the fundamental question, "what are we looking at?". The system context is perceived from different aspects such as the subject/participating objects/events, its usage/functionality with external participants, the operational and technical environment, and the development process/methods/tools. The context aspects are further classified based on factors that affect the system, such as the source of requirement, stakeholders and legacy systems, context objects such as users, and material/immaterial characteristics and relationships of context objects. Hence, based on this structuring criteria, the real-world entities outside the system boundary and inside the context boundary that affect the development of the system-of-interest can be defined as the system context. Any unknown/ambiguous entities within the facets and context aspects can be placed under the 'grey area' for later consideration. The main challenge is identifying the concerned influencers of the system design and bringing them together in the proper context to discover risks and errors early in the life cycle and mitigate any ramifications to schedule, budget, quality, and collaboration.

1.3 Operational Scenarios:

Operational Scenarios establish the operational environment of the system to capture the users and participating systems that might influence successful operation of the system. An operational scenario narrates a day-in-the-life of the system, subject to all possible operating conditions. A scenario is modeled as a Use Case and further elaborated by a flow/sequence of actions or interactions to present the functions performed by the system while interacting with users/external systems. The operational scenarios provide relevant information to refine mission requirements.

1.4 Measures of Effectiveness:

Measures of Effectiveness (MOEs) describe how well the system performs relative to mission-oriented characteristics in the operational environment. Effectiveness is specified by comparative measures based on quantitative values.

1.5 Stakeholder Validation:

This activity involves the creation of plans required for validation of needs and requirements using methods like analysis, inspection, test, and demonstration. This activity helps identify the planning efforts to validate stakeholder needs and provide a basis for agreement with stakeholders on the planning activities.

2. **System/Conceptual Level**

The modeling effort in this level establishes the functional architecture where the different aspects of system architecture are identified, designed, and developed.

2.1 System Requirements

System Requirements capture the functional/non-functional capabilities of a system (what it does and what it has) elicited as shall statements. System requirements are derived from the mission requirements and grouped among categories of requirement types. The requirement specifications are elicited to specify the required system characteristics. Finally, the requirements are allocated to system structure and behavior/Use Cases with appropriate relationships.

2.2 System Structure

The system-of-interest is identified as an assemblage of functionally-related elements forming a unitary whole. The system structure is viewed externally as a black box to understand what is known about the system and what is unknown. Further, separate views are created to present the internal structure (white-box) to understand how the system is interconnected and what interactions occur through what interfaces. The system is decomposed into logical subsystems without constricting to a specific physical implementations.

2.3 System Functions

System Functions presents the behavioral aspects of the system in terms of required functionality (what does the system do? how does it operate?) under certain operating conditions. These can be used to create executable behaviors of the system.

2.4 System Parameters

The System Parameters represent the technical performance measures of how well the system performs subject to limiting constraints.

2.5 Verification by OEM

This activity describes the procedures for technical evaluation of the system by an organization's internal team. This activity also involves the creation of verification plans required to specify the verification methods and obtain agreement with internal stakeholders.

3. **Subsystem/Logical Level**

The modeling effort at this level establishes the logical design/functionality and identifies the subsystems that make up the system.

3.1 Subsystem Requirements

The subsystem requirements capture the logical capability/functionality of the subsystems. Subsystem requirements are derived from corresponding system requirements. Each subsystem requirement specification is elicited as a shall statement that addresses the expected subsystem characteristics.

3.2 Subsystem Structure

This describes what each subsystem is and its characteristics. Here, the broad subsystem configuration alternatives are specified, and each subsystem is decomposed into components/Line Replaceable Units (LRUs). The subsystem structure represents the assembly of various logical mechanisms that helps understand what components exist within a said configuration of the subsystem.

3.3 Subsystem Functions

This activity describes the behavioral aspects of the subsystems (what it does?). Primarily, stateful behavior is specified to capture the states and transitions that represent the functionality of each subsystem.

3.4 Subsystem Parameters

These represent the constraints to express the performance characteristics of each subsystem. It can be specified using mathematical relations or external engineering models to obtain engineering analysis and simulations results. Parameters are captured using appropriate quality kinds and units.

3.5 Verification by OEM

This activity describes the procedures for technical evaluation of the subsystems to verify the internal team's requirements and concerns. This activity also includes creating verification plans required to specify the verification methods like 1-D analysis, unit tests, and inspection for obtaining agreement with internal stakeholders.

4. **Component/Physical Level**

The modeling effort at this level establishes the physical/point design by synthesizing the physical architecture by creating a component catalog.

4.1 Component Requirement

Component requirement specifications capture the point design/detailed characteristics of components/LRUs. The component requirements are derived from the subsystem requirements or specified from the component catalog as Commercial off-the-Shelf (COTS) items.

4.2 Component Structure

These represent the series of components/LRUs that are composed within the subsystem configurations. Each LRU is described with the piece-part specification obtained from COTS specifications and made available as LRU variants. This activity establishes a component catalog of all types of LRUs and their variants.

4.3 Component Functions

These represent the standard LRU functions available as COTS parts. For instance, the standard function of a switch is to turn on/off.

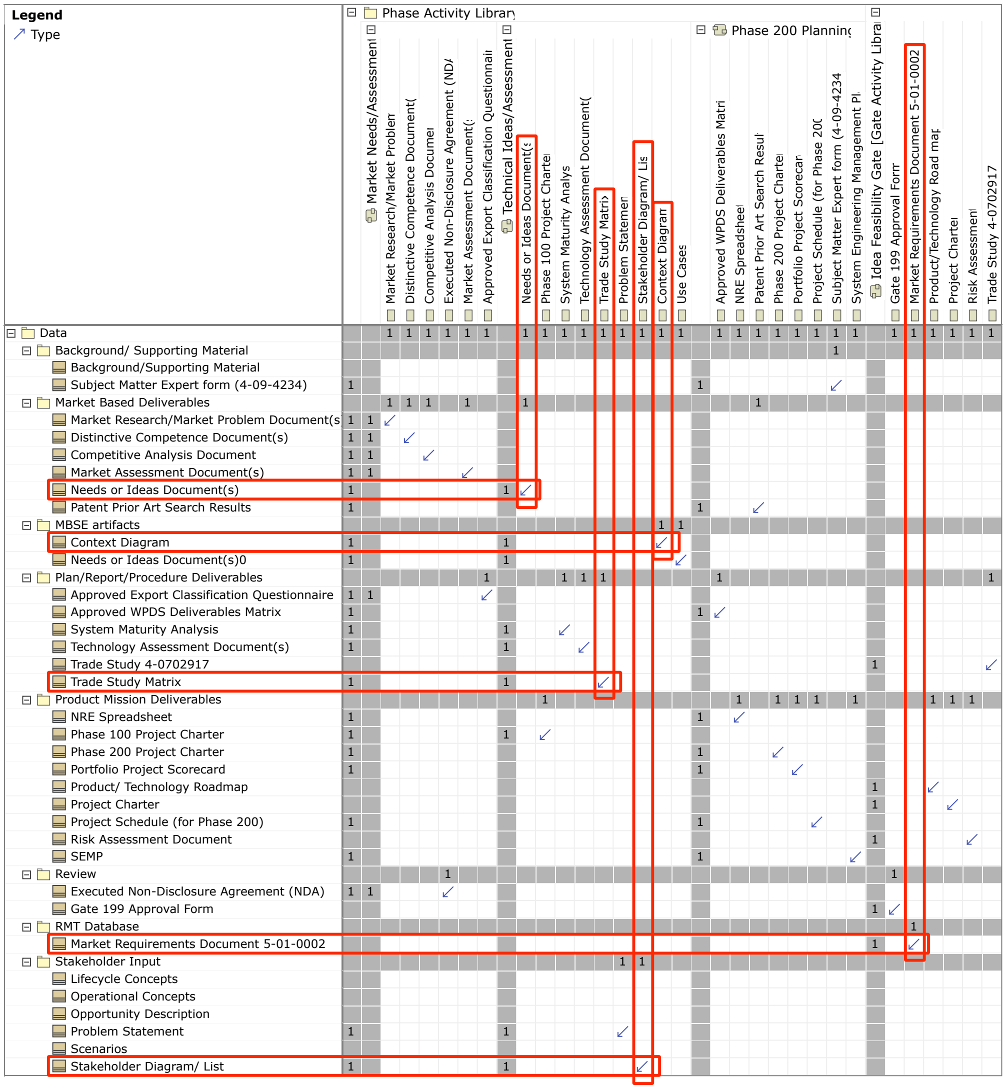
4.4 Component Parameters

These represent the physical characteristics in terms of quantities and units for each LRU, available as COTS.

4.5 Verification by OEM/Supplier

This activity describes the procedures for technical evaluation of the trade study results and point design decisions for verification purposes by pedigree suppliers/vendors. This activity might also include creating verification plans to specify the verification methods like 1-D analysis, unit tests, and inspection for obtaining agreement with suppliers.

In conclusion, augmenting this approach can fulfill the need to obtain a structured modeling framework that follows a comprehensive and well-defined methodology for effectively applying the MBSE philosophy and tools. Consequently, the modeling and research effort addresses (*I3*).



Legend

↗ Type

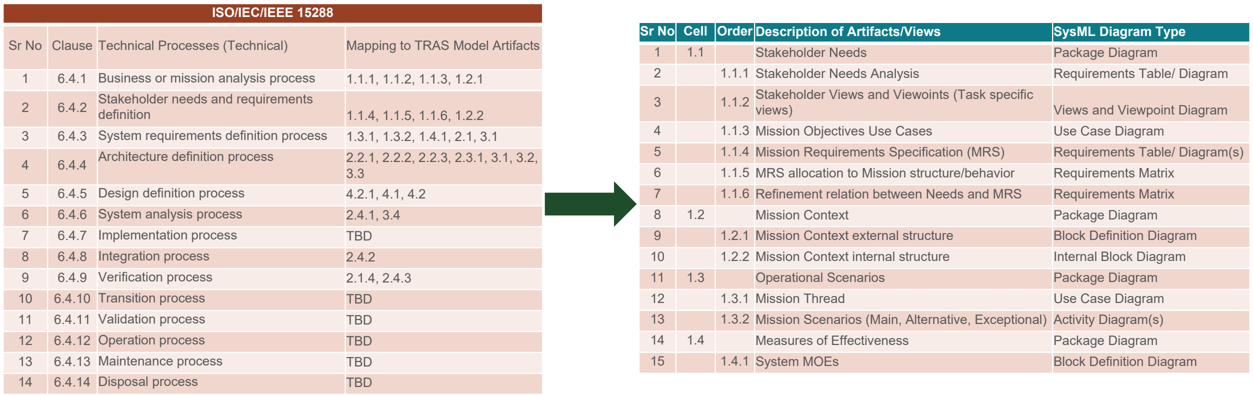
	Phase Activity Librar	Market Needs/Assessmen	Market Research/Market Problen	Distinctive Competence Document(	Competitive Analysis Documen	Executed Non-Disclosure Agreement (ND/	Market Assessment Document(:	Approved Export Classification Questionnai	Technical Ideas/Assessmen	Needs or Ideas Document(:	Phase 100 Project Charte	System Maturity Analys	Technology Assessment Document(	Trade Study Matrix	Problem Statemen	Stakeholder Diagram/ Lis	Context Diagran	Use Cases	Phase 200 Planning	Approved WPDS Deliverables Matri	NRE Spreadshee	Patent Prior Art Search Resul	Phase 200 Project Charte	Portfolio Project Scorecar	Project Schedule (for Phase 20C	Subject Matter Expert form (4-09-4234	System Engineering Management Pl	Idea Feasibility Gate [Gate Activity Librai	Gate 199 Approval Forr	Market Requirements Document 5-01-0002	Product/Technology Road maµ	Project Charte	Risk Assessmen	Trade Study 4-0702917
⊟ Data			1	1	1	1	1	1		1	1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1		1	1	1	1	1	1
⊟ Background/ Supporting Material																										1								
Background/Supporting Material																																		
Subject Matter Expert form (4-09-4234)	1																		1							↙								
⊟ Market Based Deliverables			1	1	1		1			1												1												
Market Research/Market Problem Document(s	1	1	↙																															
Distinctive Competence Document(s)	1	1		↙																														
Competitive Analysis Document	1	1			↙																													
Market Assessment Document(s)	1	1					↙																											
Needs or Ideas Document(s)	1								1	↙																								
Patent Prior Art Search Results	1																		1			↙												
⊟ MBSE artifacts																	1	1																
Context Diagram	1								1								↙																	
Needs or Ideas Document(s)0	1								1									↙																
⊟ Plan/Report/Procedure Deliverables								1				1	1	1						1														1
Approved Export Classification Questionnaire	1	1						↙																										
Approved WPDS Deliverables Matrix	1																		1	↙														
System Maturity Analysis	1								1			↙																						
Technology Assessment Document(s)	1								1				↙																					
Trade Study 4-0702917																												1						↙
Trade Study Matrix	1								1					↙																				
⊟ Product Mission Deliverables											1										1		1	1	1		1				1	1	1	
NRE Spreadsheet	1																		1		↙													
Phase 100 Project Charter	1								1		↙																							
Phase 200 Project Charter	1																		1				↙											
Portfolio Project Scorecard	1																		1					↙										
Product/ Technology Roadmap																												1			↙			
Project Charter																												1				↙		
Project Schedule (for Phase 200)	1																		1						↙									
Risk Assessment Document																												1					↙	
SEMP	1																		1								↙							
⊟ Review						1																							1					
Executed Non-Disclosure Agreement (NDA)	1	1				↙																												
Gate 199 Approval Form																												1	↙					
⊟ RMT Database																														1				
Market Requirements Document 5-01-0002																												1		↙				
⊟ Stakeholder Input															1	1																		
Lifecycle Concepts																																		
Operational Concepts																																		
Opportunity Description																																		
Problem Statement	1								1						↙																			
Scenarios																																		
Stakeholder Diagram/ List	1								1							↙																		

Figure 2.15: Coordinate the PLC deliverables to TRAS model artifacts

2.3 Coordinate the PLC Process with the System Architecture

Figure 2.15 highlights some of the deliverables required within the *Technical Ideas/Assessment* phase process that can be coordinated with the TRAS model artifacts. For instance:

- The *Stakeholder Diagram/List* and *Needs/Ideas Document(s)* will be mapped to *Stakeholders* and *Stakeholder Needs Analysis* respectively from *1.1, Stakeholder Needs*.
- The *Context Diagram* will be mapped to *1.2, Mission Context*.
- The *Trade Study Matrix* will be mapped to the system configurations within *2.4, System Parameters*.



ISO/IEC/IEEE 15288			
Sr No	Clause	Technical Processes (Technical)	Mapping to TRAS Model Artifacts
1	6.4.1	Business or mission analysis process	1.1.1, 1.1.2, 1.1.3, 1.2.1
2	6.4.2	Stakeholder needs and requirements definition	1.1.4, 1.1.5, 1.1.6, 1.2.2
3	6.4.3	System requirements definition process	1.3.1, 1.3.2, 1.4.1, 2.1, 3.1
4	6.4.4	Architecture definition process	2.2.1, 2.2.2, 2.2.3, 2.3.1, 3.1, 3.2, 3.3
5	6.4.5	Design definition process	4.2.1, 4.1, 4.2
6	6.4.6	System analysis process	2.4.1, 3.4
7	6.4.7	Implementation process	TBD
8	6.4.8	Integration process	2.4.2
9	6.4.9	Verification process	2.1.4, 2.4.3
10	6.4.10	Transition process	TBD
11	6.4.11	Validation process	TBD
12	6.4.12	Operation process	TBD
13	6.4.13	Maintenance process	TBD
14	6.4.14	Disposal process	TBD

Sr No	Cell	Order	Description of Artifacts/Views	SysML Diagram Type
1	1.1		Stakeholder Needs	Package Diagram
2		1.1.1	Stakeholder Needs Analysis	Requirements Table/ Diagram
3		1.1.2	Stakeholder Views and Viewoints (Task specific views)	Views and Viewpoint Diagram
4		1.1.3	Mission Objectives Use Cases	Use Case Diagram
5		1.1.4	Mission Requirements Specification (MRS)	Requirements Table/ Diagram(s)
6		1.1.5	MRS allocation to Mission structure/behavior	Requirements Matrix
7		1.1.6	Refinement relation between Needs and MRS	Requirements Matrix
8	1.2		Mission Context	Package Diagram
9		1.2.1	Mission Context external structure	Block Definition Diagram
10		1.2.2	Mission Context internal structure	Internal Block Diagram
11	1.3		Operational Scenarios	Package Diagram
12		1.3.1	Mission Thread	Use Case Diagram
13		1.3.2	Mission Scenarios (Main, Alternative, Exceptional)	Activity Diagram(s)
14	1.4		Measures of Effectiveness	Package Diagram
15		1.4.1	System MOEs	Block Definition Diagram

Figure 2.16: Coordinate the outcomes of the SE technical processes to TRAS model artifacts

As discussed in Sec. 2.1.2 an organization's tailored SE technical process can be mapped to the system architecture model artifacts. Figure 2.16 depicts the mapping of outcomes within the ISO/IEC/IEEE 15288 SE technical processes to architecture model artifacts [4, 10, 31]. Some processes that are unknown have been marked as To Be Determined (TBD) for later exploration. Hence, on account of capabilities realized in this effort, (*I1*) and (*I5*) are addressed with more manageable and complete visualization of programmatic aspects.

Chapter 3

System Architecture Model with Linkages between Requirements and Simulation Results for Verification

This chapter demonstrates the implementation of the augmented model-based systems engineering (MBSE) framework and methodology, documented in Sec. 2.2, for specifying and developing a system architecture model of a conceptual thrust reverser actuation system (TRAS), as discussed in Sec. 1.2.5. Some static and performance-based analyses undertaken to evaluate the effectiveness of the concepts are illustrated, and the idea of an integrated modeling and simulation environment is explored. Finally, resulting outcomes are consolidated to verify system requirements and how well the modeling effort addresses some pain points is documented.

3.1 Develop the TRAS Architecture Model

In this section, the modeling effort undertaken to develop a system architecture model of TRAS is described using text and graphics. This section heavily focuses on the usage of the MBSE philosophy and tools. Particular attention is dedicated to deploying a SysML modeling tool to develop a unified system architecture model representing a generic TRAS, defined using structure, behavior, and rules. For a complex system of interest, such as TRAS, object orientation techniques discussed in Sec. 1.2 are applied to make the design more flexible, modular, and understandable. Specifically, abstraction is employed to first separate the operational characteristics associated with the system's interaction from the logical characteristics associated with the system functionality, and later separate the physical characteristics associated with the implementation [9]. The abstractions are termed as *TRAS Operational*, *TRAS Logical* and *TRAS Physical*.

The key for executing a system modeling strategy effectively and rigorously begins with identifying the audience/end customer of the model. Who is the audience for this model? How voluminous is the model intended to be? The scope of details that we are trying to communicate for establishing and organizing the placeholders provides a head start to the overall modeling effort. The modeling effort focuses on creating relevant artifacts/model elements that contribute to specific systems engineering efforts.

3.1.1 Organize the Modeling Effort

An efficient system modeling effort in SysML follows a methodical organization of the model. The model is organized by creating nested packages that accurately reflects the system modeling effort as presented in Fig. 3.1. The TRAS model organization aligns with the MBSE approach discussed in chapter 2, except that the system/conceptual and subsystem/logical layers are grouped and modeled within one level. The architecture content is created under three main categories within SysML packages: TRAS Operational, TRAS Logical, and TRAS Physical. These packages are nested to capture the aspects of SysML, aligned with the architecture modeling guidance described in Sec. 2.2. Each package contains SysML diagram(s), model element(s), and their relationships that collectively articulate perspectives of the model through SysML syntax and semantics. Additional packages and diagrams can are created later as required. This practice intends to group similar kinds of model elements, all in one location. For instance, following the rule of thumb, the *Requirements* package contains the requirement diagrams/tables/matrices and the needs/requirements, their associated dependency/relations; the *Structure* package houses all the structural diagrams and model elements like blocks, part properties, value properties, and their relationships; the *Use Case* package holds the Use Case diagrams, activity diagrams, and all the Use Case Scenarios, User Roles/Actors involved including their relationships; the *Behavior* package accommodates the behavioral diagrams

As the modeling effort progresses, the SysML model becomes complex, and it is critical to ensure that such activities add value to the Systems Engineering effort. Thus, any non-value-added

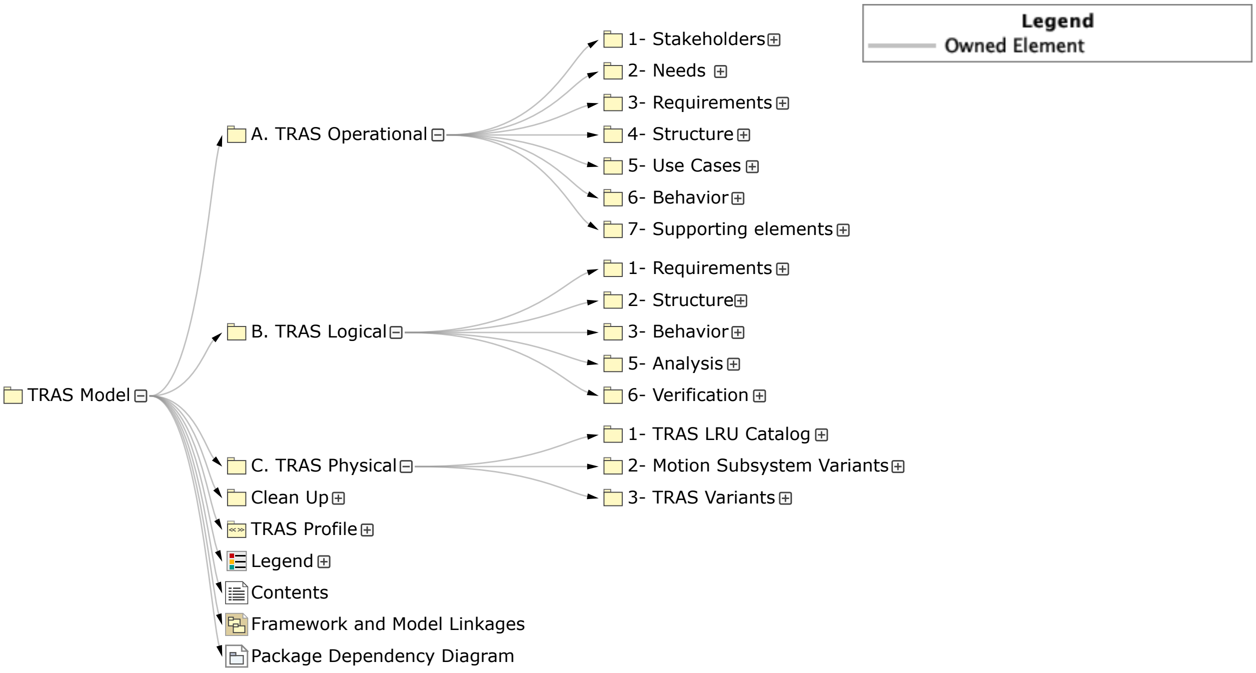


Figure 3.1: Organize the model with packages

activity is planned for a later time. Model elements and diagrams are created for measurable progress towards the specific architecture modeling goals.

In order to establish uniformity across the model, it is essential to create stereotypes typed with categories of capability that can be applied to model elements. Stereotyping is among the vital abstraction techniques practiced in system modeling. All categories of stereotypes planned for the modeling effort are created within the *TRAS Profile* package as shown in Fig. 3.2.

The model is placed under rigorous reviews, where the redundant elements are identified and eliminated from the model. Some model elements or diagrams that are unclear can be moved to the *Clean Up* package for later review to simplify the model.

In order to formulate a problem statement for TRAS system of interest, the opening idea is to investigate a solving system with an identified solution boundary, but not specific to an implementation. The system of interest is identified, and the modeling effort to develop a formal system architecture model for the Thrust Reverser Actuation System (TRAS) is discussed in this chapter. The architecture model development proceeds from abstract to detail with modeling the operational, logical and physical level of representation.

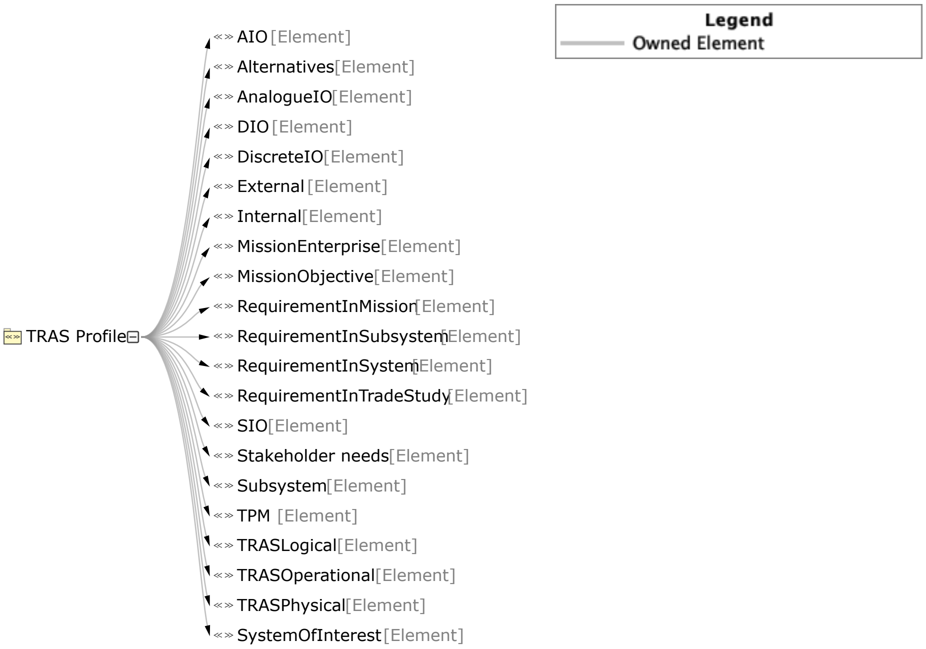

Legend
Owned Element
TRAS Profile
«» AIO [Element]
«» Alternatives[Element]
«» AnalogueIO[Element]
«» DIO [Element]
«» DiscreteIO[Element]
«» External [Element]
«» Internal[Element]
«» MissionEnterprise[Element]
«» MissionObjective[Element]
«» RequirementInMission[Element]
«» RequirementInSubsystem[Element]
«» RequirementInSystem[Element]
«» RequirementInTradeStudy[Element]
«» SIO[Element]
«» Stakeholder needs[Element]
«» Subsystem[Element]
«» TPM [Element]
«» TRASLogical[Element]
«» TRASOperational[Element]
«» TRASPhysical[Element]
«» SystemOfInterest[Element]

Figure 3.2: Set up the TRAS profile of stereotypes

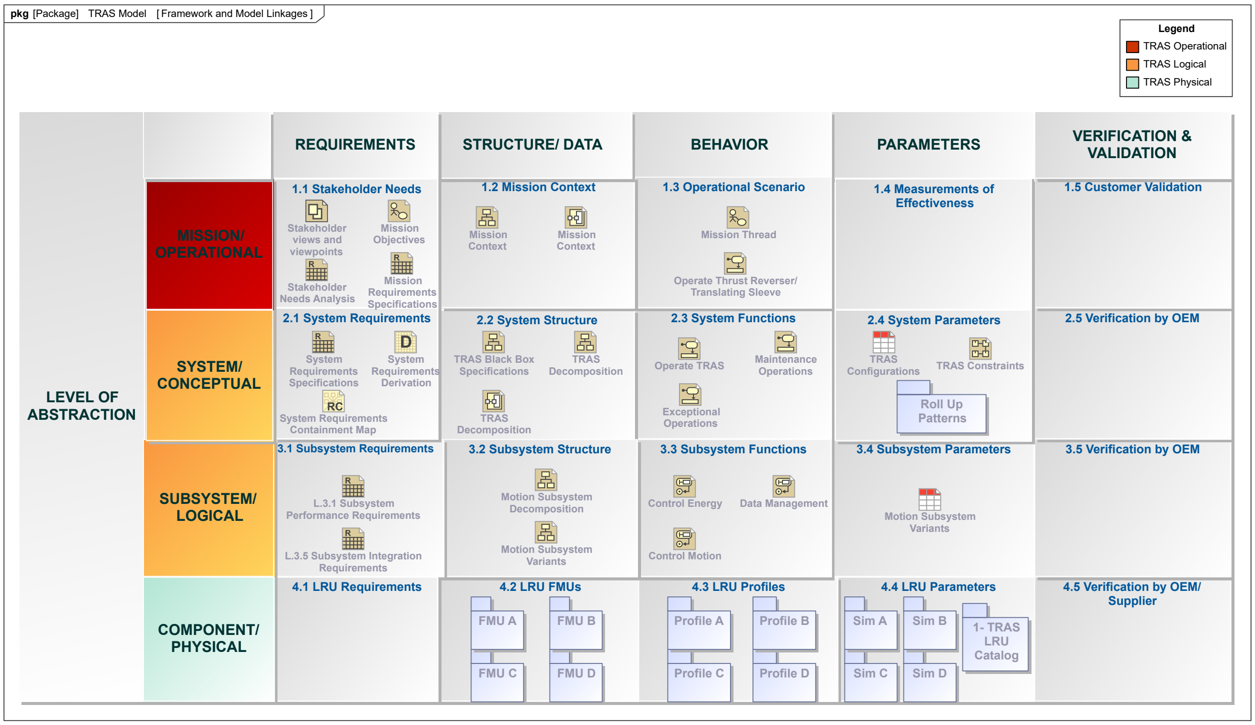


Figure 3.3: TRAS model artifacts mapped to the framework after implementation

3.1.2 Implement MBSE Framework and Methodology

1. **TRAS Operational**

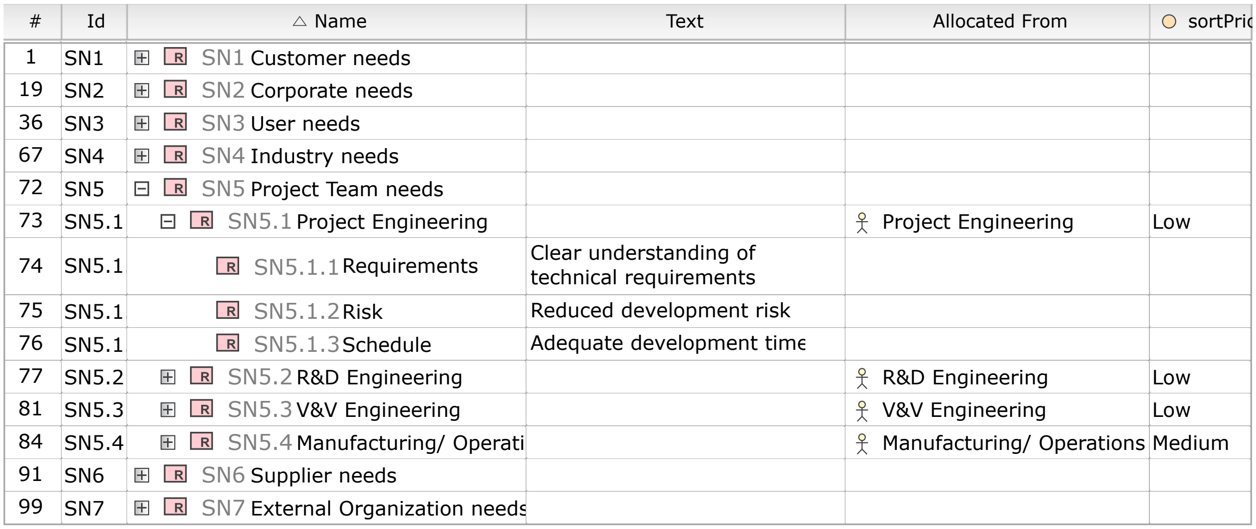
   The TRAS Mission/Operational level of abstraction represents the top-level structure of the architecture model from the standpoint of the eventual user. This is where the stakeholder dialog/voice of the customer is established.

1.1 Stakeholder Needs Introduces the key stakeholders involved in the project about who they are and what they are looking for.

1.1.1 Stakeholder Needs Analysis: This activity is performed to understand the stakeholders, their needs, expectations/concerns and their level of interest and involvement. The stakeholders involved in the project are identified and modeled as actors/user roles. The broad categories of needs are captured as customer needs, corporate needs, user needs, industry needs, project team needs, supplier needs, and external organizations needs. These generic needs are further broken down to capture the perceived stakeholder needs that are more detailed and are defined using context-relevant names and corresponding text. An appropriate numbering scheme is applied. The outcome of the stakeholder needs analysis is the stakeholder needs, represented in Fig. 3.4 as text-based requirements using a SysML requirements table. For instance, the perceived needs *SN5.1* Project Engineering of project engineering stakeholder is contained within *SN5* Project Team needs, and it specifies the needs relevant to requirements, risks and schedule as *SN5.1.1, SN5.1.2, and SN5.1.3* respectively.

The overall stakeholder needs can be reviewed using a containment map as shown in Fig. 3.5 and the respective stakeholders allocated to those needs can be reviewed using a SysML Allocation Matrix as shown in Fig. 3.6.

1.1.2 Stakeholder Views and Viewpoints: A view and viewpoint diagram is created to reflect perspectives of different stakeholders. As illustrated in Fig. 3.7, the main categories of stakeholders involved in the project/program, identified as customer, operator, manufacturer, and regulator are modeled as actors/user roles. Some of these stakeholders are further special-



#	Id	△ Name	Text	Allocated From	○ sortPric
1	SN1	⊞ R SN1 Customer needs			
19	SN2	⊞ R SN2 Corporate needs			
36	SN3	⊞ R SN3 User needs			
67	SN4	⊞ R SN4 Industry needs			
72	SN5	⊟ R SN5 Project Team needs			
73	SN5.1	⊟ R SN5.1 Project Engineering		Project Engineering	Low
74	SN5.1	R SN5.1.1 Requirements	Clear understanding of technical requirements		
75	SN5.1	R SN5.1.2 Risk	Reduced development risk		
76	SN5.1	R SN5.1.3 Schedule	Adequate development time		
77	SN5.2	⊞ R SN5.2 R&D Engineering		R&D Engineering	Low
81	SN5.3	⊞ R SN5.3 V&V Engineering		V&V Engineering	Low
84	SN5.4	⊞ R SN5.4 Manufacturing/ Operati		Manufacturing/ Operations	Medium
91	SN6	⊞ R SN6 Supplier needs			
99	SN7	⊞ R SN7 External Organization needs			

Figure 3.4: Stakeholder needs analysis

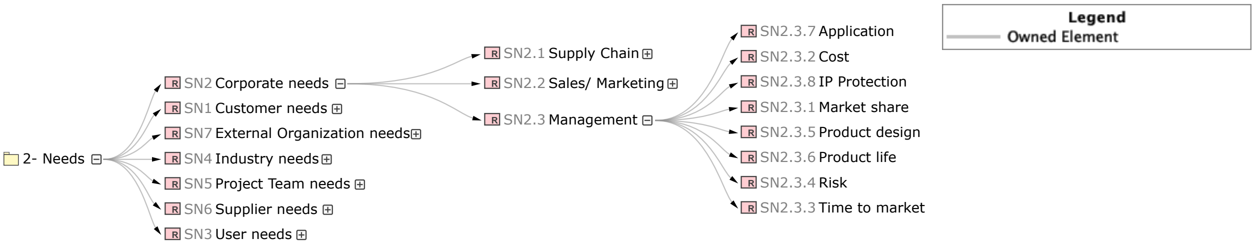
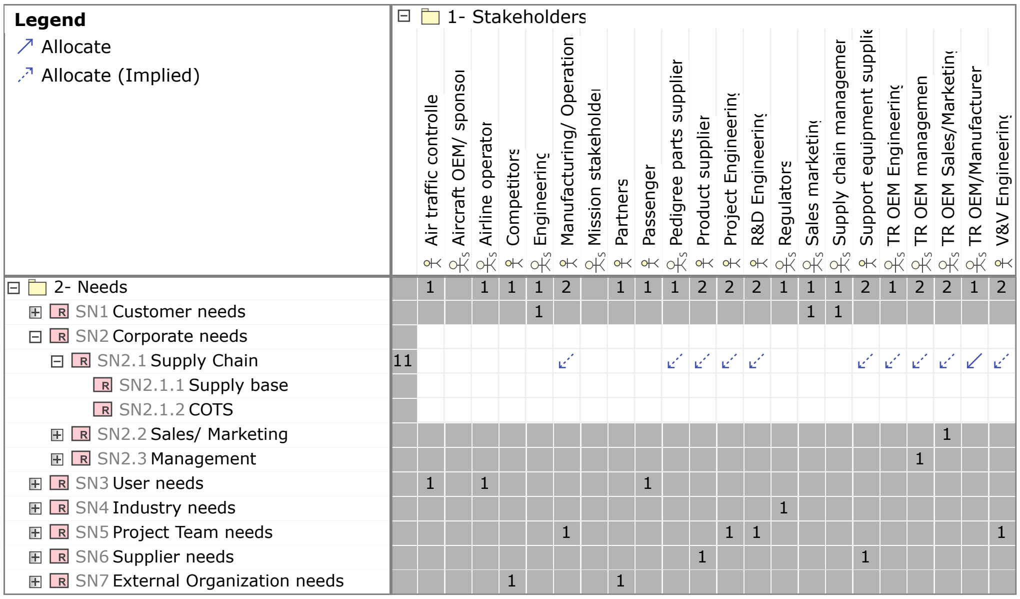


Figure 3.5: Reviewing stakeholder needs in a containment map



Legend
↗ Allocate
⇗ Allocate (Implied)

⊟ 1- Stakeholders		Air traffic controlle	Aircraft OEM/ sponsoı	Airline operator	Competitors	Engineering	Manufacturing/ Operation	Mission stakeholder	Partners	Passenger	Pedigree parts supplier	Product supplier	Project Engineering	R&D Engineering	Regulators	Sales marketing	Supply chain managemer	Support equipment supplie	TR OEM Engineering	TR OEM managemen	TR OEM Sales/Marketing	TR OEM/Manufacturer	V&V Engineering
⊟ 2- Needs		1		1	1	1	2		1	1	1	2	2	2	1	1	1	2	1	2	2	1	2
⊞ R SN1 Customer needs						1										1	1						
⊟ R SN2 Corporate needs																							
⊟ R SN2.1 Supply Chain	11						⇙				⇙	⇙	⇙	⇙				⇙	⇙	⇙	⇙	↙	⇙
R SN2.1.1 Supply base																							
R SN2.1.2 COTS																							
⊞ R SN2.2 Sales/ Marketing																					1		
⊞ R SN2.3 Management																				1			
⊞ R SN3 User needs		1		1						1													
⊞ R SN4 Industry needs															1								
⊞ R SN5 Project Team needs							1						1	1									1
⊞ R SN6 Supplier needs												1						1					
⊞ R SN7 External Organization needs					1				1														

Figure 3.6: Stakeholder needs allocation

ized into department specific teams. The purpose of capturing the stakeholder Views and Viewpoints is to support the assessment of stakeholder viewpoints during verification and validation. From the standpoint of the stakeholders, the relevant concerns are captured within the *viewpoints* and elaborated within the conforming *views* in a SysML Views and Viewpoints Diagram [32]. This diagram provides a means to focus on aspects of the system model that are of particular interest to stakeholders. A *Viewpoint* specifies a set of rules that describe how the *view* should express the information from the model to address the stakeholder concerns. A *View* is constructed from a subset of the model that conforms to a defined *viewpoint* and addresses specific stakeholder concerns. It is important to note that while a *View* is a SysML construct within a model, the artifacts generated from views belong to the external to the model environment. For instance, a table or a document generated from a view is not integrated into a SysML model, while the view itself exists in the model.

1.1.3 Mission Objectives Use Cases: The high-level mission objectives that are relevant to the stakeholder concerns are modeled as Use Cases as shown in Fig. 3.8. The association of the key stakeholders with particular objectives specifies their level of interest, either primary/secondary and expresses their direct/indirect involvement. The outcomes of this activity lead to transforming the mission objective into stakeholder/mission requirements that are allocated to them.

Note: This activity focuses on identifying what the mission objectives are rather than how that can be achieved/accomplished. However, future efforts shall elaborate the overall programmatic behavior associated with the mission-level objectives of TRAS.

1.1.4 Mission Requirements Specification (MRS): The requirements in the mission are derived from the stakeholder needs, *SN* and recognised in the TRAS model using a tag ID prefixed with "L1".

The overall TRAS requirements in the mission are grouped under broad categories of stakeholder requirements such as business, design, functional, non-functional and project. Each tier of hierarchy for requirements in a mission is elaborated in a detailed requirement dia-

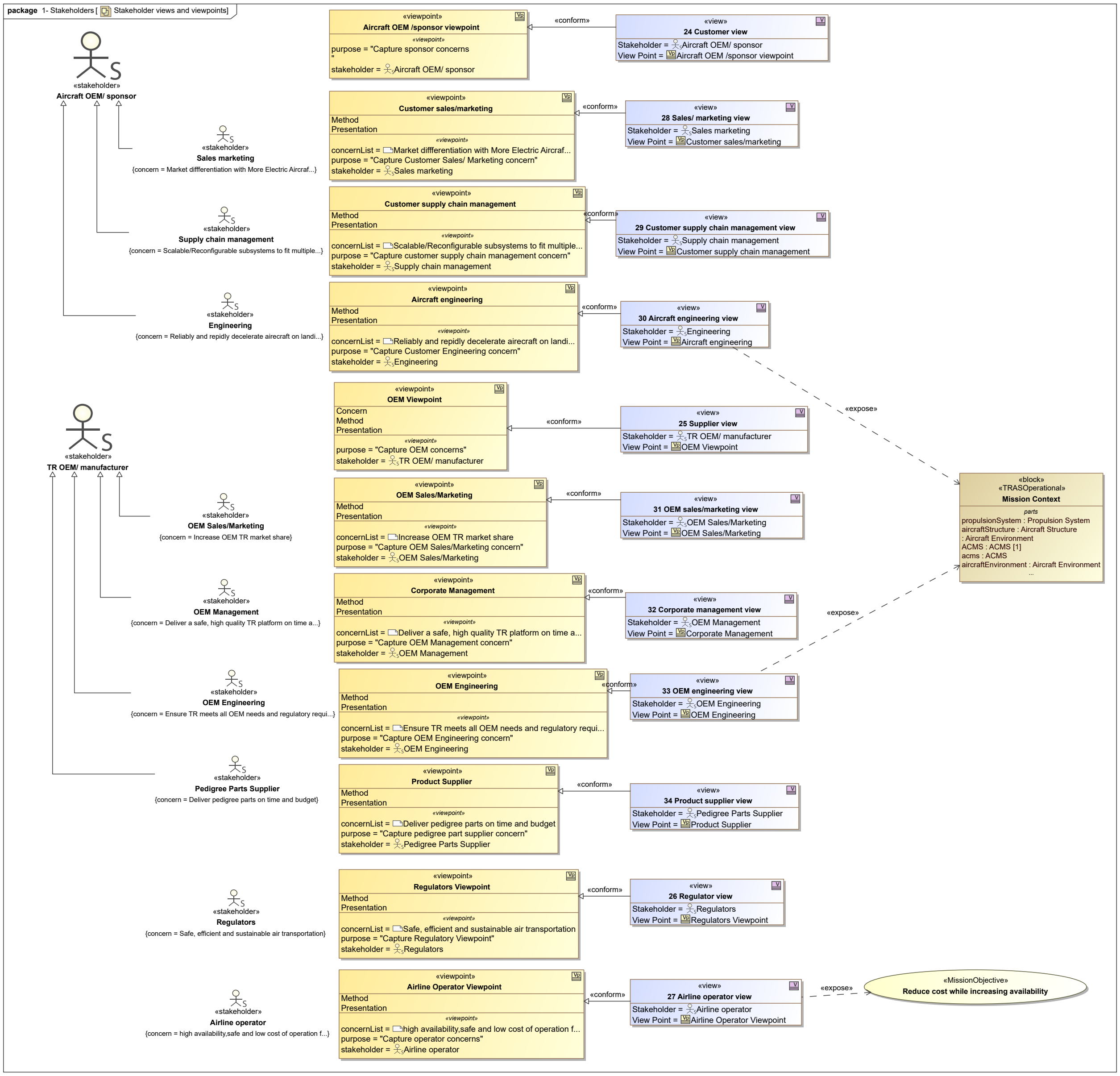

package 1- Stakeholders [Stakeholder views and viewpoints]
«stakeholder»
Aircraft OEM/ sponsor
«stakeholder»
Sales marketing
{concern = Market diffferentiation with More Electric Aircraf...}
«stakeholder»
Supply chain management
{concern = Scalable/Reconfigurable subsystems to fit multiple...}
«stakeholder»
Engineering
{concern = Reliably and repidly decelerate airecraft on landi...}
«stakeholder»
TR OEM/ manufacturer
«stakeholder»
OEM Sales/Marketing
{concern = Increase OEM TR market share}
«stakeholder»
OEM Management
{concern = Deliver a safe, high quality TR platform on time a...}
«stakeholder»
OEM Engineering
{concern = Ensure TR meets all OEM needs and regulatory requi...}
«stakeholder»
Pedigree Parts Supplier
{concern = Deliver pedigree parts on time and budget}
«stakeholder»
Regulators
{concern = Safe, efficient and sustainable air transportation}
«stakeholder»
Airline operator
{concern = high availability,safe and low cost of operation f...}
«viewpoint»
Aircraft OEM /sponsor viewpoint
«viewpoint»
purpose = "Capture sponsor concerns
"
stakeholder = Aircraft OEM/ sponsor
«viewpoint»
Customer sales/marketing
Method
Presentation
«viewpoint»
concernList = Market diffferentiation with More Electric Aircraf...
purpose = "Capture Customer Sales/ Marketing concern"
stakeholder = Sales marketing
«viewpoint»
Customer supply chain management
Method
Presentation
«viewpoint»
concernList = Scalable/Reconfigurable subsystems to fit multiple...
purpose = "Capture customer supply chain management concern"
stakeholder = Supply chain management
«viewpoint»
Aircraft engineering
Method
Presentation
«viewpoint»
concernList = Reliably and repidly decelerate airecraft on landi...
purpose = "Capture Customer Engineering concern"
stakeholder = Engineering
«viewpoint»
OEM Viewpoint
Concern
Method
Presentation
«viewpoint»
purpose = "Capture OEM concerns"
stakeholder = TR OEM/ manufacturer
«viewpoint»
OEM Sales/Marketing
Method
Presentation
«viewpoint»
concernList = Increase OEM TR market share
purpose = "Capture OEM Sales/Marketing concern"
stakeholder = OEM Sales/Marketing
«viewpoint»
Corporate Management
Method
Presentation
«viewpoint»
concernList = Deliver a safe, high quality TR platform on time a...
purpose = "Capture OEM Management concern"
stakeholder = OEM Management
«viewpoint»
OEM Engineering
Method
Presentation
«viewpoint»
concernList = Ensure TR meets all OEM needs and regulatory requi...
purpose = "Capture OEM Engineering concern"
stakeholder = OEM Engineering
«viewpoint»
Product Supplier
Method
Presentation
«viewpoint»
concernList = Deliver pedigree parts on time and budget
purpose = "Capture pedigree part supplier concern"
stakeholder = Pedigree Parts Supplier
«viewpoint»
Regulators Viewpoint
Method
Presentation
«viewpoint»
concernList = Safe, efficient and sustainable air transportation
purpose = "Capture Regulatory Viewpoint"
stakeholder = Regulators
«viewpoint»
Airline Operator Viewpoint
Method
Presentation
«viewpoint»
concernList = high availability,safe and low cost of operation f...
purpose = "Capture operator concerns"
stakeholder = Airline operator
«conform»
«view»
24 Customer view
Stakeholder = Aircraft OEM/ sponsor
View Point = Aircraft OEM /sponsor viewpoint
«view»
28 Sales/ marketing view
Stakeholder = Sales marketing
View Point = Customer sales/marketing
«view»
29 Customer supply chain management view
Stakeholder = Supply chain management
View Point = Customer supply chain management
«view»
30 Aircraft engineering view
Stakeholder = Engineering
View Point = Aircraft engineering
«view»
25 Supplier view
Stakeholder = TR OEM/ manufacturer
View Point = OEM Viewpoint
«view»
31 OEM sales/marketing view
Stakeholder = OEM Sales/Marketing
View Point = OEM Sales/Marketing
«view»
32 Corporate management view
Stakeholder = OEM Management
View Point = Corporate Management
«view»
33 OEM engineering view
Stakeholder = OEM Engineering
View Point = OEM Engineering
«view»
34 Product supplier view
Stakeholder = Pedigree Parts Supplier
View Point = Product Supplier
«view»
26 Regulator view
Stakeholder = Regulators
View Point = Regulators Viewpoint
«view»
27 Airline operator view
Stakeholder = Airline operator
View Point = Airline Operator Viewpoint
«expose»
«block»
«TRASOperational»
Mission Context
parts
propulsionSystem : Propulsion System
aircraftStructure : Aircraft Structure
: Aircraft Environment
ACMS : ACMS [1]
acms : ACMS
aircraftEnvironment : Aircraft Environment
...
«MissionObjective»
Reduce cost while increasing availability

Figure 3.7: Stakeholder views and viewpoints

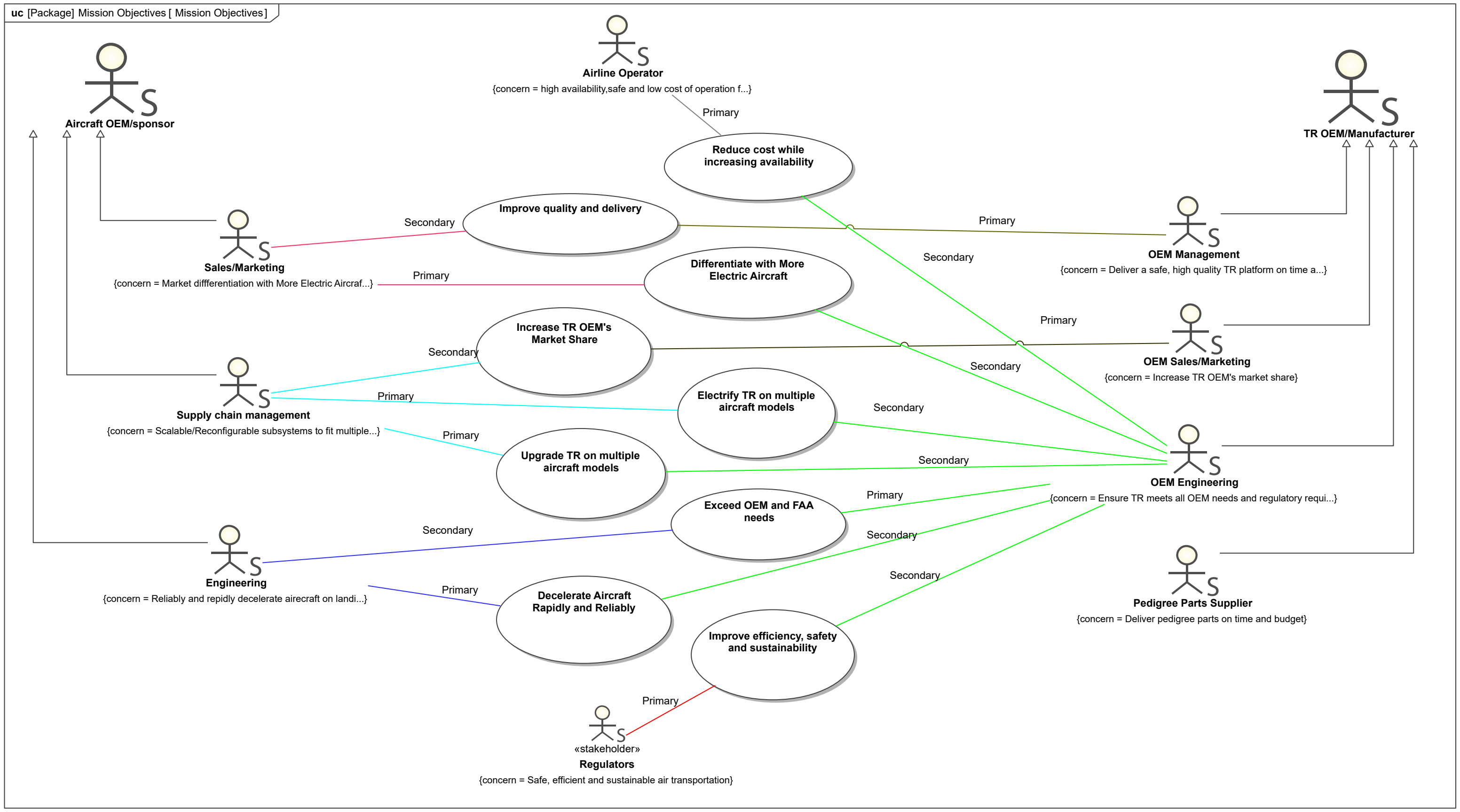
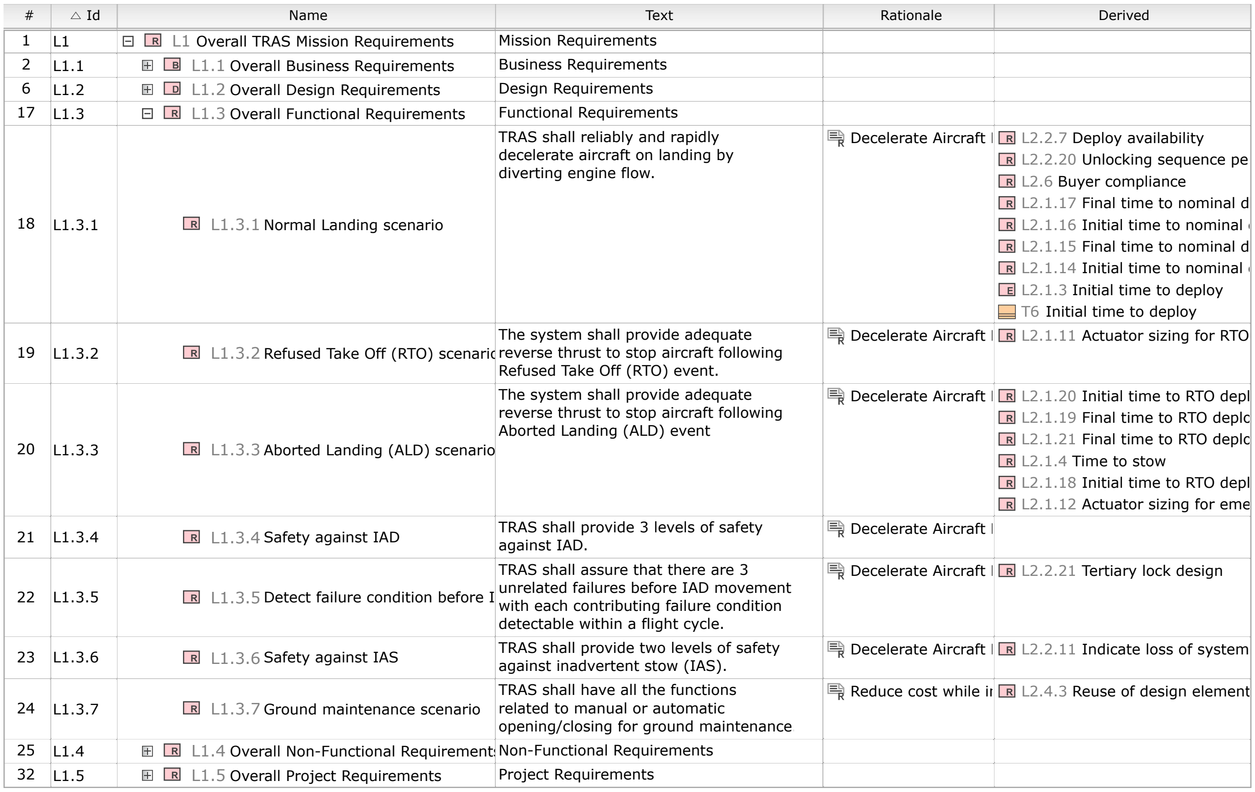

uc [Package] Mission Objectives [Mission Objectives]
Aircraft OEM/sponsor
Airline Operator
{concern = high availability,safe and low cost of operation f...}
Primary
Reduce cost while increasing availability
TR OEM/Manufacturer
Improve quality and delivery
Secondary
Sales/Marketing
{concern = Market diffferentiation with More Electric Aircraf...}
Primary
Differentiate with More Electric Aircraft
Primary
OEM Management
{concern = Deliver a safe, high quality TR platform on time a...}
Secondary
Primary
Increase TR OEM's Market Share
Secondary
OEM Sales/Marketing
{concern = Increase TR OEM's market share}
Secondary
Primary
Supply chain management
{concern = Scalable/Reconfigurable subsystems to fit multiple...}
Electrify TR on multiple aircraft models
Secondary
Primary
Upgrade TR on multiple aircraft models
Secondary
OEM Engineering
{concern = Ensure TR meets all OEM needs and regulatory requi...}
Primary
Exceed OEM and FAA needs
Secondary
Secondary
Engineering
{concern = Reliably and repidly decelerate airecraft on landi...}
Primary
Decelerate Aircraft Rapidly and Reliably
Secondary
Pedigree Parts Supplier
{concern = Deliver pedigree parts on time and budget}
Improve efficiency, safety and sustainability
Primary
«stakeholder»
Regulators
{concern = Safe, efficient and sustainable air transportation}

Figure 3.8: Mission objectives modeled as Use Cases



#	Id	Name	Text	Rationale	Derived
1	L1	L1 Overall TRAS Mission Requirements	Mission Requirements		
2	L1.1	L1.1 Overall Business Requirements	Business Requirements		
6	L1.2	L1.2 Overall Design Requirements	Design Requirements		
17	L1.3	L1.3 Overall Functional Requirements	Functional Requirements		
18	L1.3.1	L1.3.1 Normal Landing scenario	TRAS shall reliably and rapidly decelerate aircraft on landing by diverting engine flow.	Decelerate Aircraft I	L2.2.7 Deploy availability L2.2.20 Unlocking sequence pe L2.6 Buyer compliance L2.1.17 Final time to nominal d L2.1.16 Initial time to nominal L2.1.15 Final time to nominal d L2.1.14 Initial time to nominal L2.1.3 Initial time to deploy T6 Initial time to deploy
19	L1.3.2	L1.3.2 Refused Take Off (RTO) scenaric	The system shall provide adequate reverse thrust to stop aircraft following Refused Take Off (RTO) event.	Decelerate Aircraft I	L2.1.11 Actuator sizing for RTO
20	L1.3.3	L1.3.3 Aborted Landing (ALD) scenario	The system shall provide adequate reverse thrust to stop aircraft following Aborted Landing (ALD) event	Decelerate Aircraft I	L2.1.20 Initial time to RTO depl L2.1.19 Final time to RTO deplo L2.1.21 Final time to RTO deplo L2.1.4 Time to stow L2.1.18 Initial time to RTO depl L2.1.12 Actuator sizing for eme
21	L1.3.4	L1.3.4 Safety against IAD	TRAS shall provide 3 levels of safety against IAD.	Decelerate Aircraft I	
22	L1.3.5	L1.3.5 Detect failure condition before I	TRAS shall assure that there are 3 unrelated failures before IAD movement with each contributing failure condition detectable within a flight cycle.	Decelerate Aircraft I	L2.2.21 Tertiary lock design
23	L1.3.6	L1.3.6 Safety against IAS	TRAS shall provide two levels of safety against inadvertent stow (IAS).	Decelerate Aircraft I	L2.2.11 Indicate loss of system
24	L1.3.7	L1.3.7 Ground maintenance scenario	TRAS shall have all the functions related to manual or automatic opening/closing for ground maintenance	Reduce cost while ir	L2.4.3 Reuse of design element
25	L1.4	L1.4 Overall Non-Functional Requirements	Non-Functional Requirements		
32	L1.5	L1.5 Overall Project Requirements	Project Requirements		

Figure 3.9: Mission requirements specifications

gram to elicit multiple mission requirement specifications (MRS). An MRS shall statement is elicited using text with proper context-specific names. The Measure of Effectiveness (MOE) and target/expected values are specified. All requirements in mission are elaborated using appropriate rationale and linked as a model element that can be reused across a series of MRS. The text-based mission requirements are represented in a SysML requirements table as shown in Fig. 3.9. The overall requirements in mission are also represented using a containment map as shown in Fig. 3.10. Since the current knowledge about the system and MRS is limited, some requirements are missing. However, future research and collaboration efforts shall broaden the scope to yield a comprehensive set of requirements in mission.

1.1.5 Refine requirements in mission:

Note that all requirements in mission are improvised/refined by relevant mission objective Use Cases as depicted in Fig. 3.11. The outcome of this activity is a series of high-level mission requirement specifications that provide a baseline/source of information to derive

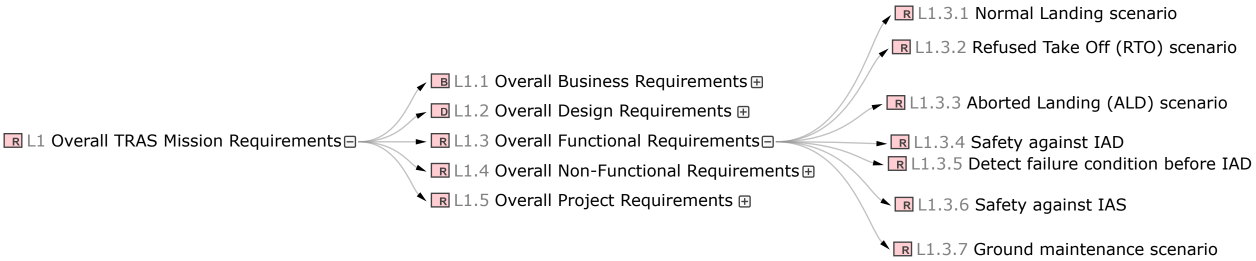
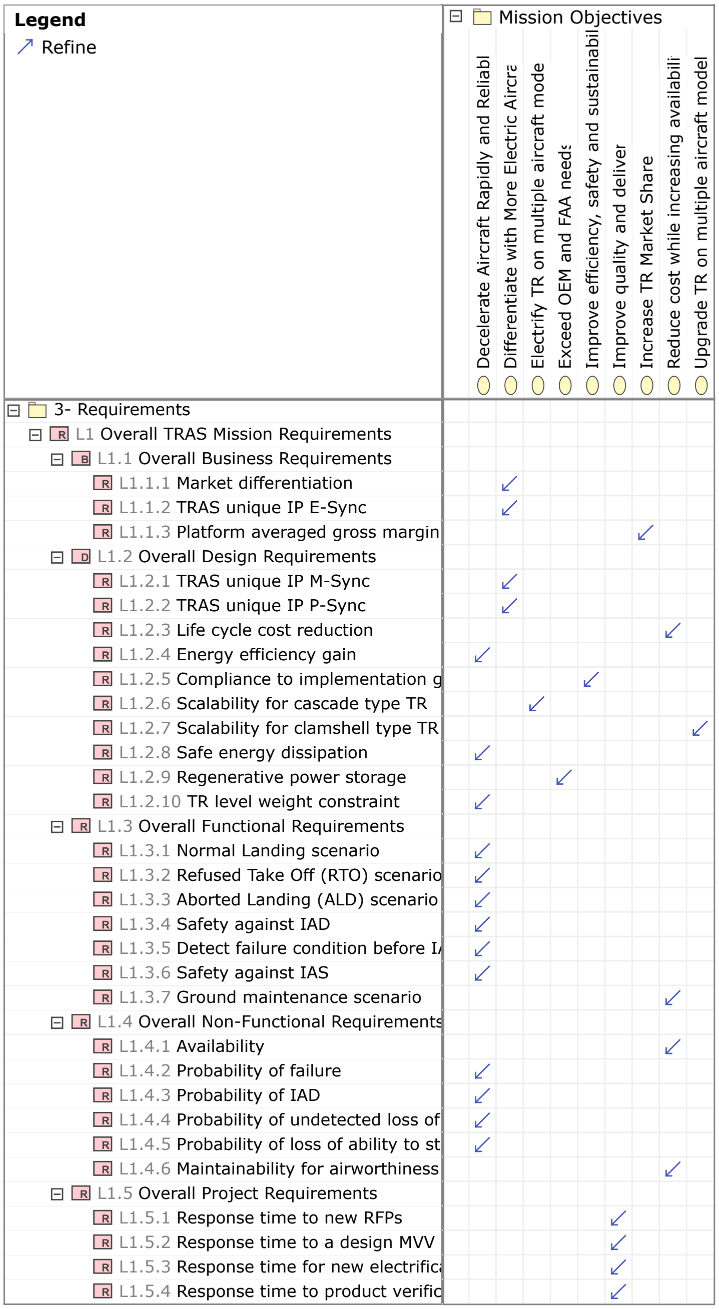

L1 Overall TRAS Mission Requirements
L1.1 Overall Business Requirements
L1.2 Overall Design Requirements
L1.3 Overall Functional Requirements
L1.4 Overall Non-Functional Requirements
L1.5 Overall Project Requirements
L1.3.1 Normal Landing scenario
L1.3.2 Refused Take Off (RTO) scenario
L1.3.3 Aborted Landing (ALD) scenario
L1.3.4 Safety against IAD
L1.3.5 Detect failure condition before IAD
L1.3.6 Safety against IAS
L1.3.7 Ground maintenance scenario

Figure 3.10: Reviewing requirements in mission using containment map



Legend
↗ Refine

Mission Objectives	Decelerate Aircraft Rapidly and Reliabl	Differentiate with More Electric Aircra	Electrify TR on multiple aircraft mode	Exceed OEM and FAA needs	Improve efficiency, safety and sustainabil	Improve quality and deliver	Increase TR Market Share	Reduce cost while increasing availabili	Upgrade TR on multiple aircraft model
3- Requirements									
L1 Overall TRAS Mission Requirements									
L1.1 Overall Business Requirements									
L1.1.1 Market differentiation		↙							
L1.1.2 TRAS unique IP E-Sync		↙							
L1.1.3 Platform averaged gross margin							↙		
L1.2 Overall Design Requirements									
L1.2.1 TRAS unique IP M-Sync		↙							
L1.2.2 TRAS unique IP P-Sync		↙							
L1.2.3 Life cycle cost reduction								↙	
L1.2.4 Energy efficiency gain	↙								
L1.2.5 Compliance to implementation g					↙				
L1.2.6 Scalability for cascade type TR			↙						
L1.2.7 Scalability for clamshell type TR									↙
L1.2.8 Safe energy dissipation	↙								
L1.2.9 Regenerative power storage				↙					
L1.2.10 TR level weight constraint	↙								
L1.3 Overall Functional Requirements									
L1.3.1 Normal Landing scenario	↙								
L1.3.2 Refused Take Off (RTO) scenario	↙								
L1.3.3 Aborted Landing (ALD) scenario	↙								
L1.3.4 Safety against IAD	↙								
L1.3.5 Detect failure condition before I/	↙								
L1.3.6 Safety against IAS	↙								
L1.3.7 Ground maintenance scenario								↙	
L1.4 Overall Non-Functional Requirements									
L1.4.1 Availability								↙	
L1.4.2 Probability of failure	↙								
L1.4.3 Probability of IAD	↙								
L1.4.4 Probability of undetected loss of	↙								
L1.4.5 Probability of loss of ability to st	↙								
L1.4.6 Maintainability for airworthiness								↙	
L1.5 Overall Project Requirements									
L1.5.1 Response time to new RFPs						↙			
L1.5.2 Response time to a design MVV						↙			
L1.5.3 Response time for new electrifica						↙			
L1.5.4 Response time to product verific						↙			

Figure 3.11: Mission objectives refine mission requirements

further lower-level requirements in system, thus contributing to setup traceability with the stakeholder objectives and concerns.

1.2 Mission Context: The primary purpose of this activity is to obtain a clear definition of the system and context boundaries for the correct definition of requirements. The system boundary is established by separating the aspects of the system that are subject to change from the stable aspects that do not belong to the system itself. (what is internal and external? What separates the system from its context and external environment?).

Based on the guidance and structuring criteria for visualizing the mission context as discussed in chapter 2, the key users and external systems participating across the system boundary and external context environment are first identified and captured in the mission context. Figures 3.12 and 3.13 explained in the following paragraphs establishes the structural aspects within the mission/operational level.

1.2.1 Mission context external structure: An external structure of the mission context that appears from the outside is captured in a SysML Block Definition Diagram (BDD) as shown in Fig. 3.12 to visualize the black-box environment. Here, system boundary is identified, the participants/user(s) associated with the system are modeled as Actors/User Roles. In contrast, the external system(s) interacting with the system of interest across the boundary are modeled as blocks composed within the mission context and described with appropriate multiplicity (number of instances that can exist at a time).

1.2.2 Mission context internal structure: The context entities previously defined as blocks and actors in a BDD are reused as parts typed by those blocks within a SysML Internal Block Diagram (IBD) as depicted in Fig. 3.13. This context diagram helps locate the system of interest (SOI) in the operational context. The diagram communicates how the SOI is interconnected with its users and external systems, the type of interconnection, and specifies/elaborates the nature of the interaction between them using interfaces (i/f) and the appropriate type of data that flows across those interfaces. (where is the SOI placed in the overall context? How is it interconnected with users and external entities? What interfaces facilitate the interaction

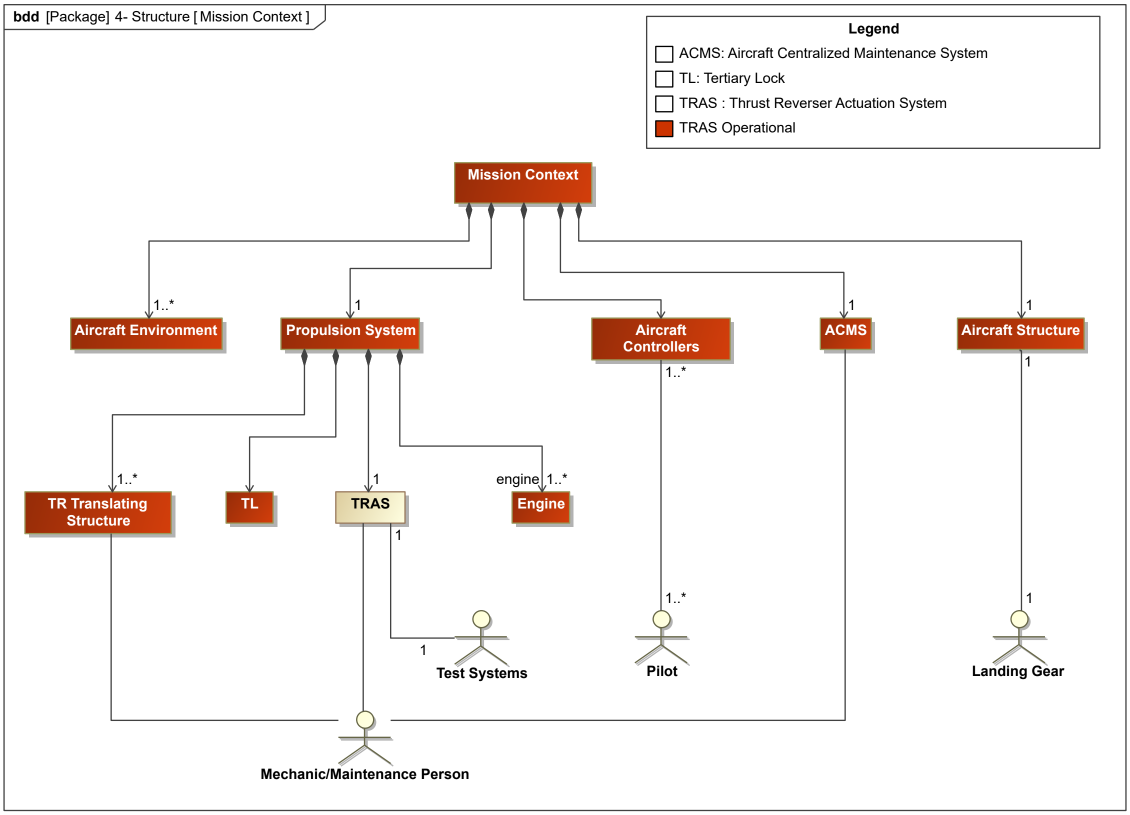


Figure 3.12: Mission context external structure

with these entities?) (This also helps clarify the idea of capturing views in separate SysML diagrams, the definition using blocks in a BDD and then its usage as parts in an IBD. As illustrated in Fig. 3.13, TRAS is composed within the Propulsion System where it interacts with the participants of the context across the system boundary. Aircraft Controllers and the Aircraft Centralized Maintenance System (ACMS) facilitates *analog signaling* through the transmission of measurable analog/discrete signals characterized by command or feedback through the Signal Input/Output (SIO) interfaces. The Engine and ACMS receive the TRAS relevant information as analog feedback from the system boundary through the SIO interfaces. The *data signaling* is facilitated by AFDx communication protocol and network at the system boundary through the Digital Input/Output (DIO) interfaces. TRAS further interacts with the Thrust Reverser (TR) Translating Structure for torque transfer, load transfer, heat transfer, and electrical bonding through power interfaces such as torque i/f, force i/f and load i/f. The *power signaling* facilitates hydraulic/electric/load transfer through interfaces such as harness, physical connections and electrical interfaces. The Tertiary Lock is considered external to the system and interacts with the Engine, TR translating structure and Aircraft Controllers. The Aircraft Structure and Aircraft Environment are external to the context but interact across the context boundary through ambient i/f. The landing gear is modeled as a User Role since it influences the purpose of TRAS, i.e., to maintain the landing distance. The Pilot interacts with the thrust lever through an HMI that triggers the logic sequence resulting in the deployment of the TRAS.

1.2.3 PLC Data Deliverable Categorization: This section discusses the significance of the data component and its aspects that benefit a modeling effort. The data aspects explored in this modeling effort refer to identifying the broad categories of information content created in the phase/gate deliverable. The system data captured in each deliverable is referred to as ’data deliverable’ in this thesis.

This data aspect channels the system information content required to be generated throughout the life cycle phases of system development. It bridges the gap between the PLC process

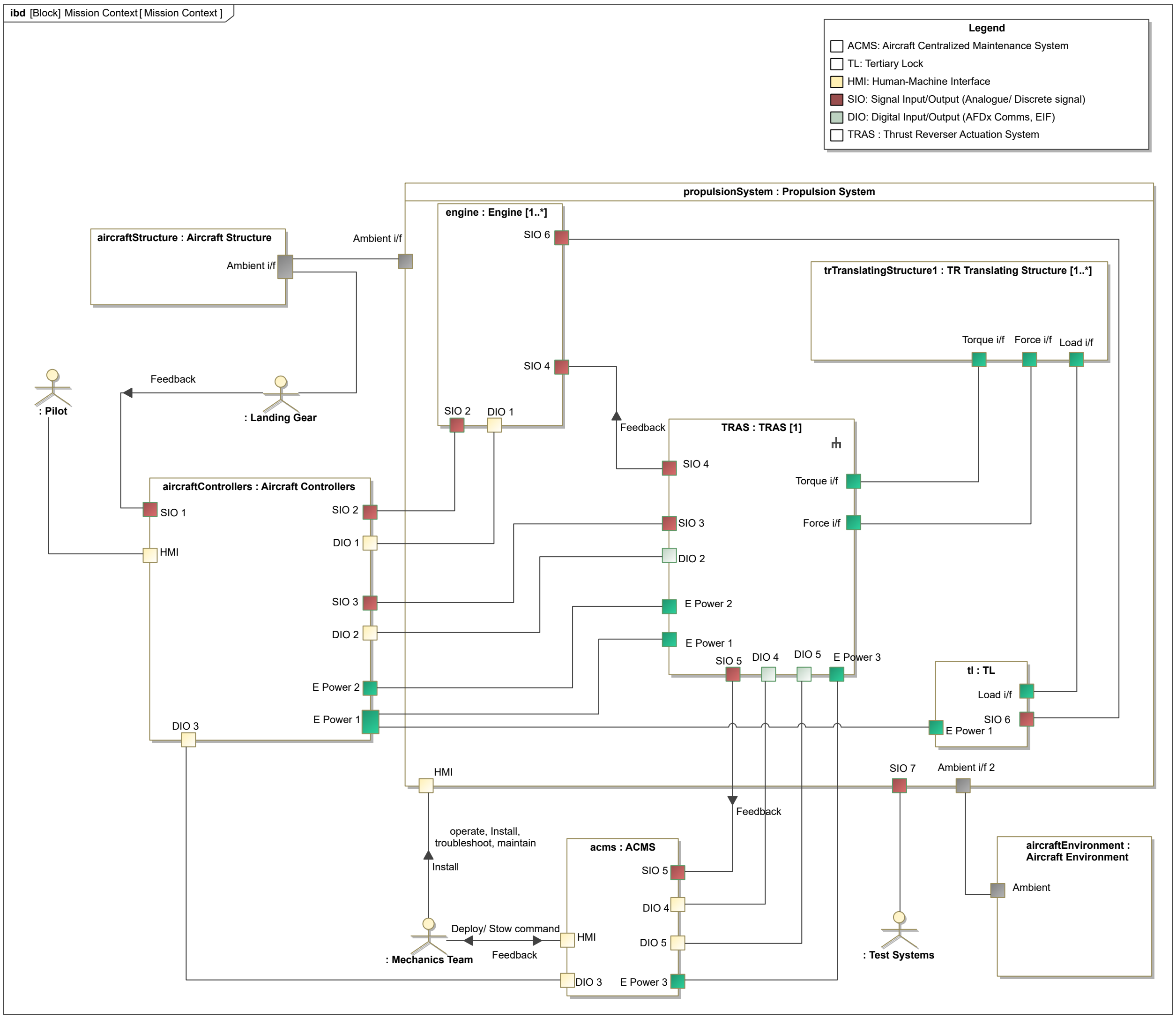
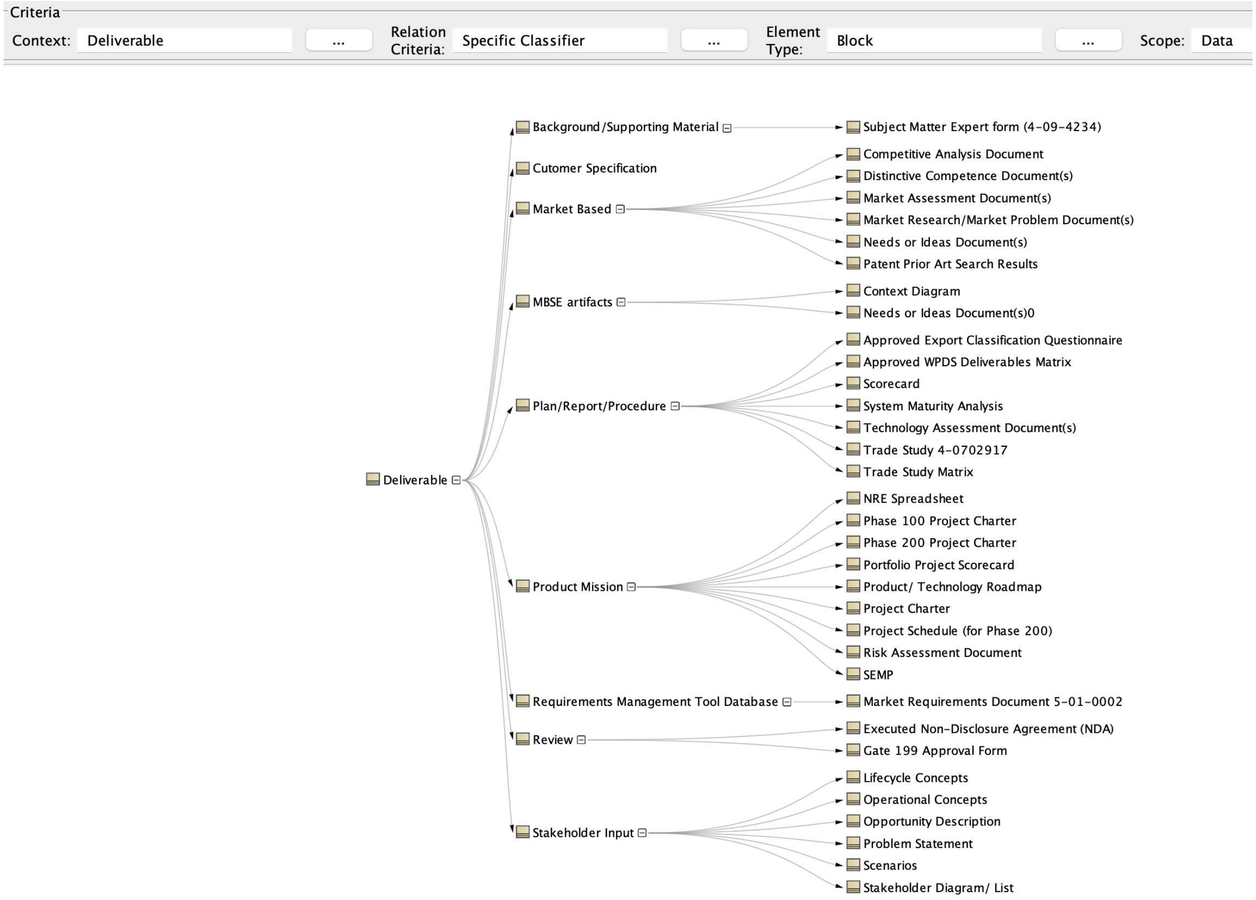


Figure 3.13: System context diagram



Criteria

Context: Deliverable ... Relation Criteria: Specific Classifier ... Element Type: Block ... Scope: Data

Deliverable
Background/Supporting Material
Subject Matter Expert form (4-09-4234)
Cutomer Specification
Market Based
Competitive Analysis Document
Distinctive Competence Document(s)
Market Assessment Document(s)
Market Research/Market Problem Document(s)
Needs or Ideas Document(s)
Patent Prior Art Search Results
MBSE artifacts
Context Diagram
Needs or Ideas Document(s)0
Plan/Report/Procedure
Approved Export Classification Questionnaire
Approved WPDS Deliverables Matrix
Scorecard
System Maturity Analysis
Technology Assessment Document(s)
Trade Study 4-0702917
Trade Study Matrix
Product Mission
NRE Spreadsheet
Phase 100 Project Charter
Phase 200 Project Charter
Portfolio Project Scorecard
Product/ Technology Roadmap
Project Charter
Project Schedule (for Phase 200)
Risk Assessment Document
SEMP
Requirements Management Tool Database
Market Requirements Document 5-01-0002
Review
Executed Non-Disclosure Agreement (NDA)
Gate 199 Approval Form
Stakeholder Input
Lifecycle Concepts
Operational Concepts
Opportunity Description
Problem Statement
Scenarios
Stakeholder Diagram/ List

Figure 3.14: Modeling categories of PLC deliverable/outcome as data objects

model and a system architecture model. Data discovery happens through all phases of the life cycle, and information content is updated in the following stages as and when more data becomes available. A data deliverable is modeled as a block and associated with the PLC process as a data object in the second level process workflow.

Each outcome/deliverable of the PLC phase, the gate-based process is captured/stored/located in a common repository called ’Platform Project folder’. All types of data deliverable that exist in the second level process flowchart are categorized under generic (parent) deliverable types and named according to the source of data (where does the information come from?). These are modeled as blocks, and represented in a relations map as depicted in Fig. 3.14.

1.3 TRAS Operational Scenarios: The Operational Scenarios, commonly referred to as the modes of operation, are modeled as Use Cases in a SysML Use Case Diagram (UC). As depicted in Fig. 3.15, the scenarios/ Use Cases are partitioned/classified as generalized scenarios and specialized scenarios. The generalized scenarios are allocated to the system context, implying the nature of the mission operation that the system performs as a participant. The specialized scenarios imply specific modes that are realized after tailoring the generalized scenarios/modes of operation, but the logic remains the same as described in the parent scenario. All concerned users that participate in these Use Cases/scenarios are modeled as Actors/User Roles with an association relationship. As depicted in Fig. 3.15, each generalized scenario is elaborated by a primary behavior that invokes a set of system Use Case functionality marked under control functions and explained in further paragraphs.

- Control Energy–this function controls the power supply to energize and de-energize the system for satisfactory system operation.
- Control Motion–this function controls the position, velocity, acceleration, and torque to drive the synchronized motion subsystem.
- Monitor Motion–this function allows monitoring actuator position feedback using sensors during system operation.
- Control Lock–this function controls the locking/unlocking function for safe actuation of the system.
- Control Brake–This function refers to the braking/snubbing feature performed to decelerate the actuation system operation near the end of the stroke to diminish the impact.
- Control Maintenance–this function provides either manual/interactive driving of actuators during a maintenance operation.
- Data Management–this function controls the signaling and communication protocol and network during system operation.
- Anomaly Detection and Prognostics–this safety function supports fault management by detecting and responding to process and mechanical limits with the help of jam sensors and interlock switches.

In this way, the TRAS Operational Use Cases are modeled distinct from TRAS Logical Use Cases.

1.3.1 Mission Scenarios (Main, Alternative, Exceptional):

Each operational scenario/Use Case is characterized by multiple specialized scenarios where the steps in a sequence may be rearranged/tailored to suit a particular outcome/scenario.

Main Scenario: The Main Use Case refers to the most common sequence of actions intended to be performed by TRAS under the influence of the Pilot and Landing Gear to achieve/maintain a safe landing distance. The common control functions performed by TRAS during the basic operation are *Control Energy*, *Control Motion*, and *Control Lock*. The optional control functions performed by TRAS during the main operation are *Monitor Motion* and *Data Management*. The primary/basic operational scenario is elaborated by creating a SysML Activity Diagram. As illustrated in Fig. 3.17, the operational functions that are performed in the basic mission scenario are allocated to the participants (users, external systems) of the mission context. This diagram can also be referred to as a day-in-the-life of Thrust Reverser (TR). TRAS operation is further elaborated under the TR control system function in a separate activity diagram.

The three specialized scenarios modeled for the main scenario are *Normal Landing Scenario (NL), Refused Takeoff Scenario (RTO), and Aborted Landing Scenario (ALD).*

Alternative Scenario: The alternative scenario Use Case refers to an alternative sequence of actions intended to be performed on/by TRAS for the main scenario, i.e., deploy/stow translating sleeves. The common functionality of TRAS in the maintenance scenario is *Control Lock*. The optional control functions performed by TRAS in the maintenance scenario are *Control Motion, Monitor Motion, Control Brake, Control Maintenance, Data Management*, and *Anomaly Detection and Prognostics*. The maintenance operation is defined in a SysML Activity Diagram for the alternative scenario as depicted in Fig. 3.19.

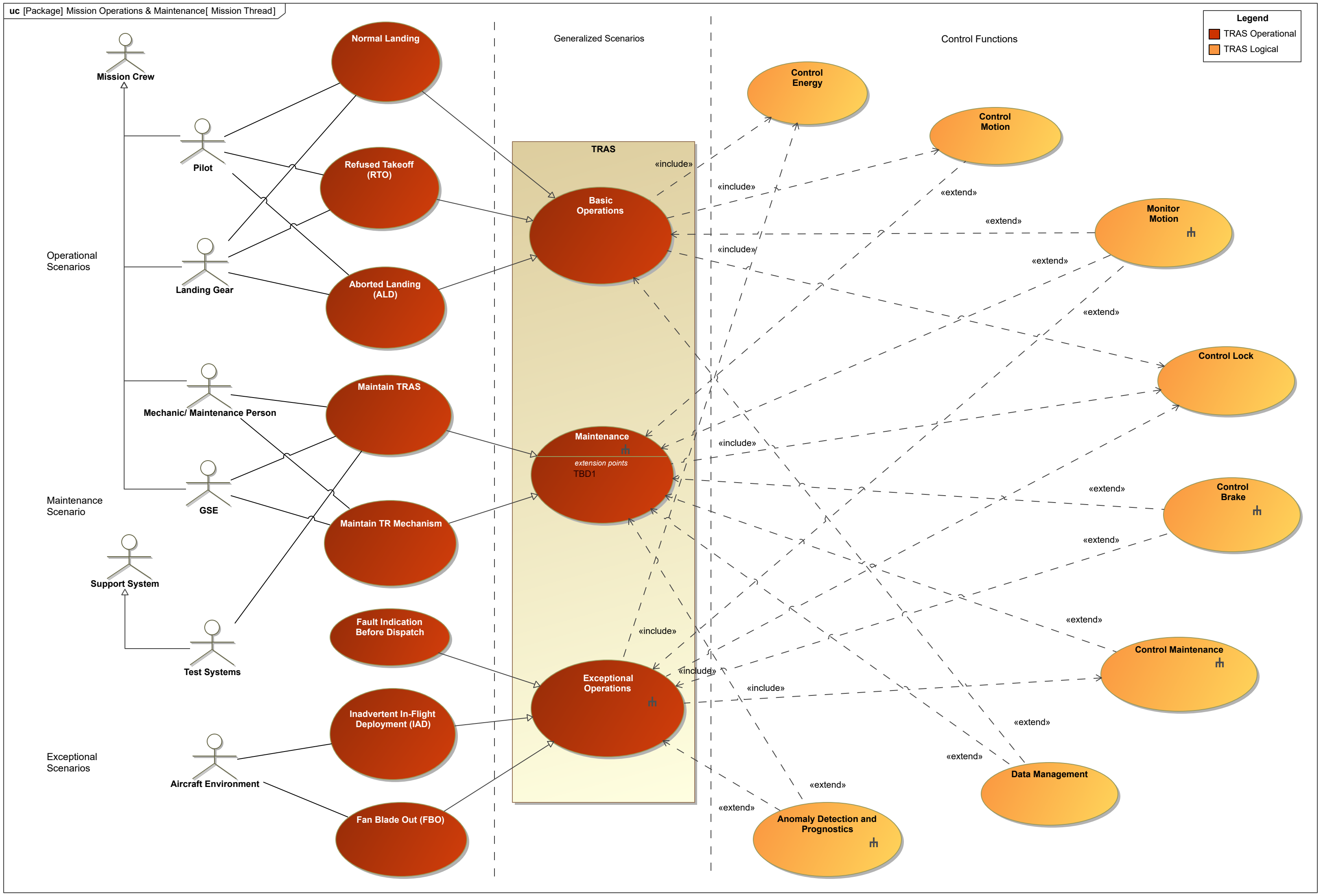
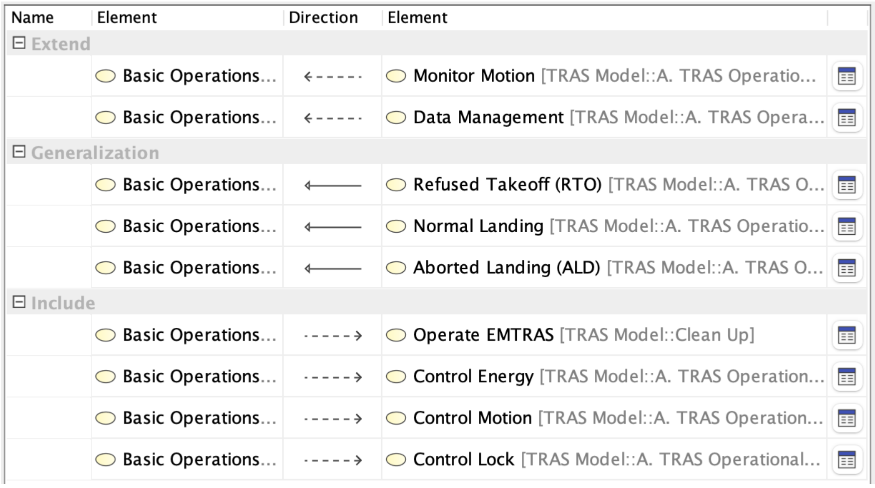


Figure 3.15: Mission/Operational Scenarios



Name	Element	Direction	Element	
⊟ Extend				
	⬭ Basic Operations...	←- - - - -	⬭ Monitor Motion [TRAS Model::A. TRAS Operatio...	▦
	⬭ Basic Operations...	←- - - - -	⬭ Data Management [TRAS Model::A. TRAS Opera...	▦
⊟ Generalization				
	⬭ Basic Operations...	←———	⬭ Refused Takeoff (RTO) [TRAS Model::A. TRAS O...	▦
	⬭ Basic Operations...	←———	⬭ Normal Landing [TRAS Model::A. TRAS Operatio...	▦
	⬭ Basic Operations...	←———	⬭ Aborted Landing (ALD) [TRAS Model::A. TRAS O...	▦
⊟ Include				
	⬭ Basic Operations...	- - - - -→	⬭ Operate EMTRAS [TRAS Model::Clean Up]	▦
	⬭ Basic Operations...	- - - - -→	⬭ Control Energy [TRAS Model::A. TRAS Operation...	▦
	⬭ Basic Operations...	- - - - -→	⬭ Control Motion [TRAS Model::A. TRAS Operation...	▦
	⬭ Basic Operations...	- - - - -→	⬭ Control Lock [TRAS Model::A. TRAS Operational...	▦

Figure 3.16: Basic operations Use Case specifications

The specialized scenarios modeled for the alternative scenario are *Maintain TRAS* scenario performed by mechanic, ground safety engineer (GSE), and OEM Test Systems and *Maintain TR Mechanism* scenario performed by mechanic and GSE.

Exceptional Scenario: The exceptional scenario Use Case refers to a sequence of actions or interactions that is unintended/accidental, leading to one or more fault conditions. The common control functions performed by TRAS in the exceptional scenario are *Control Energy, Control Lock, and Control Maintenance*. The optional control functions performed by TRAS during the exceptional scenario are *Monitor Motion, Control Brake, and Anomaly Detection and Prognostics*. A fault scenario occurs if any of the actions depicted in Fig. 3.21 remains unsuccessful.

The specialized scenarios modeled for the exceptional scenarios are *Fault Indication Before Dispatch, Inadvertent In-Flight Deployment (IAD), and Fan Blade Out (FBO)*.

Note: The detailed modeling of specialized scenarios are excluded from the scope of this thesis.

1.4 Measurements of Effectiveness: The measurements of effectiveness are specified by the quantifying parameters that gauge the overall success of the mission. MOEs are extracted from the mission requirements specification and improvises the MRS.

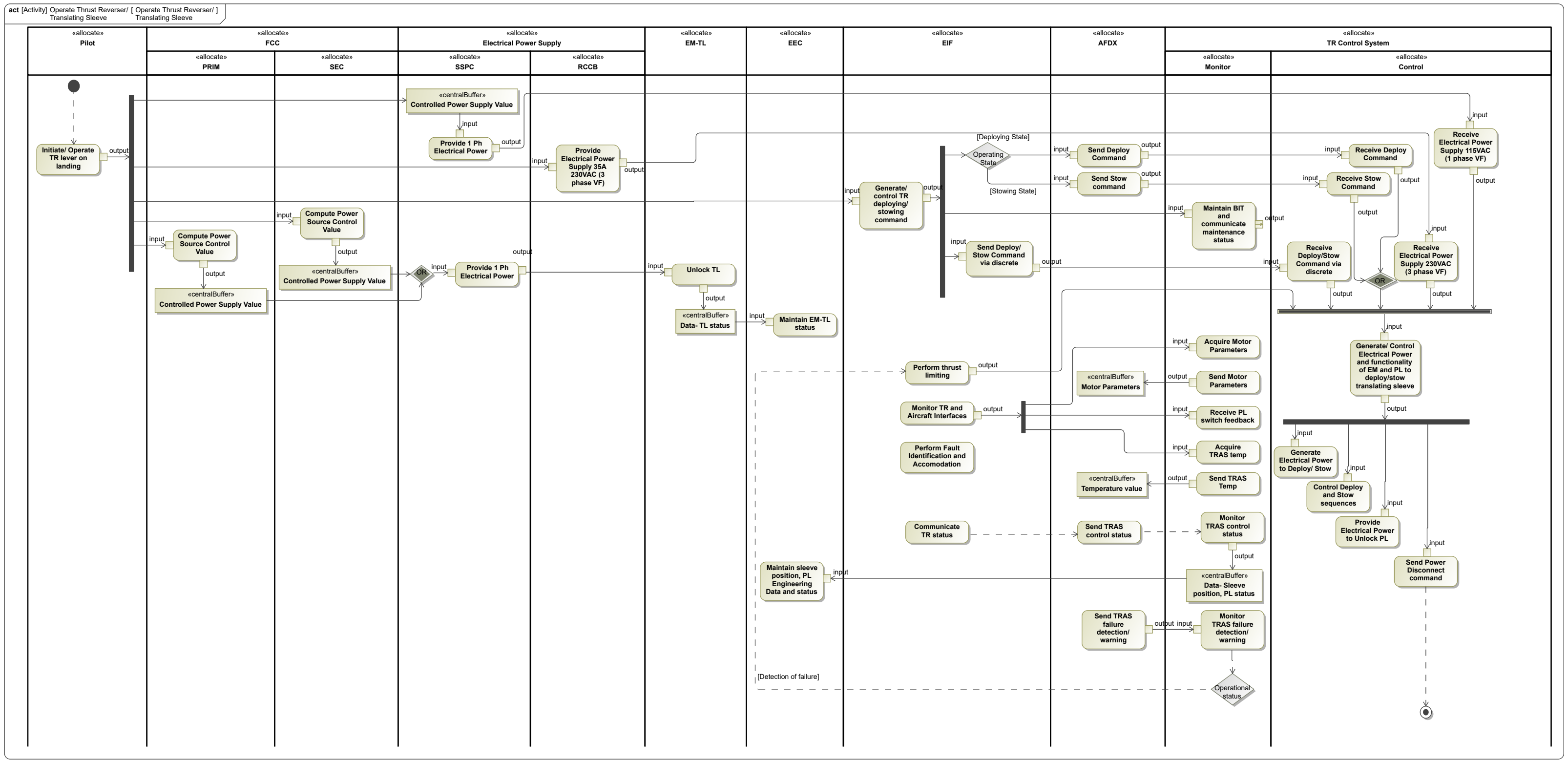
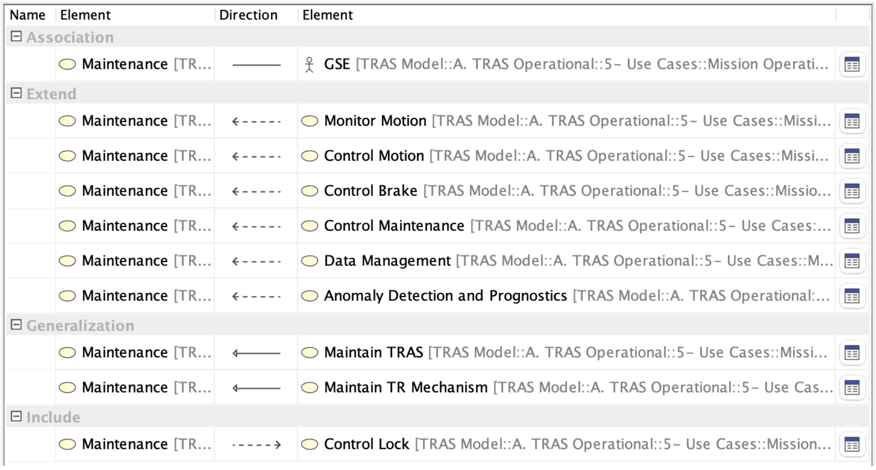

act [Activity] Operate Thrust Reverser/ Translating Sleeve [Operate Thrust Reverser/ Translating Sleeve]
«allocate»
Pilot
FCC
PRIM
SEC
Electrical Power Supply
SSPC
RCCB
EM-TL
EEC
EIF
AFDX
TR Control System
Monitor
Control
Initiate/ Operate TR lever on landing
output
input
«centralBuffer»
Controlled Power Supply Value
Provide 1 Ph Electrical Power
Provide Electrical Power Supply 35A 230VAC (3 phase VF)
Compute Power Source Control Value
OR
Unlock TL
Data- TL status
Maintain EM-TL status
Generate/ control TR deploying/ stowing command
Operating State
[Deploying State]
[Stowing State]
Send Deploy Command
Send Stow command
Send Deploy/ Stow Command via discrete
Maintain BIT and communicate maintenance status
Receive Deploy Command
Receive Stow Command
Receive Electrical Power Supply 115VAC (1 phase VF)
Receive Deploy/Stow Command via discrete
Receive Electrical Power Supply 230VAC (3 phase VF)
Generate/ Control Electrical Power and functionality of EM and PL to deploy/stow translating sleeve
Perform thrust limiting
Monitor TR and Aircraft Interfaces
Perform Fault Identification and Accomodation
Communicate TR status
Acquire Motor Parameters
Send Motor Parameters
Motor Parameters
Receive PL switch feedback
Acquire TRAS temp
Send TRAS Temp
Temperature value
Send TRAS control status
Monitor TRAS control status
Data- Sleeve position, PL status
Maintain sleeve position, PL Engineering Data and status
Send TRAS failure detection/ warning
Monitor TRAS failure detection/ warning
Operational status
[Detection of failure]
Generate Electrical Power to Deploy/ Stow
Control Deploy and Stow sequences
Provide Electrical Power to Unlock PL
Send Power Disconnect command

Figure 3.17: TRAS basic operation (a-day-in-the-life)



Name	Element	Direction	Element
⊟ Association			
	Maintenance [TR...	———	GSE [TRAS Model::A. TRAS Operational::5- Use Cases::Mission Operati...
⊟ Extend			
	Maintenance [TR...	<-----	Monitor Motion [TRAS Model::A. TRAS Operational::5- Use Cases::Missi...
	Maintenance [TR...	<-----	Control Motion [TRAS Model::A. TRAS Operational::5- Use Cases::Missi...
	Maintenance [TR...	<-----	Control Brake [TRAS Model::A. TRAS Operational::5- Use Cases::Missio...
	Maintenance [TR...	<-----	Control Maintenance [TRAS Model::A. TRAS Operational::5- Use Cases:...
	Maintenance [TR...	<-----	Data Management [TRAS Model::A. TRAS Operational::5- Use Cases::M...
	Maintenance [TR...	<-----	Anomaly Detection and Prognostics [TRAS Model::A. TRAS Operational:...
⊟ Generalization			
	Maintenance [TR...	<———	Maintain TRAS [TRAS Model::A. TRAS Operational::5- Use Cases::Missi...
	Maintenance [TR...	<———	Maintain TR Mechanism [TRAS Model::A. TRAS Operational::5- Use Cas...
⊟ Include			
	Maintenance [TR...	----->	Control Lock [TRAS Model::A. TRAS Operational::5- Use Cases::Mission...

Figure 3.18: Alternative/Maintenance operation Use Case specification

1.5 Customer Validation: The verification and validation activities are devoted to the future scope of this thesis.

2. **TRAS Logical** (System Level)

After describing the mission/operational context in which the system operates to meet the mission objectives, the system of interest is identified. The different dimensions of the system are elaborated and captured by creating diagrams and model elements in the TRAS Logical package.

2.1 System Requirements: The requirements in system are derived from the stakeholder requirements/ requirements in mission, *MRS* and recognised in the TRAS model using a tag ID prefixed with "L2". The overall TRAS requirements in system are grouped under broad categories of system aspects such as performance, reliability, maintainability and safety (RMS), physical, life cycle cost, system integration.

2.1.1 System Requirements Specification (SRS): Each tier of hierarchy for requirements in system is elaborated in a detailed requirement diagram to elicit multiple system requirement specifications (SRS). Each nested SRS is defined with a context-specific name and elicited in a text that specifies a technical performance measure (TPM), and expected values. All requirements in system are explained using appropriate rationale and linked as a model element that can be reused across a series of SRS. An exhaustive/complete list of text-based system requirements

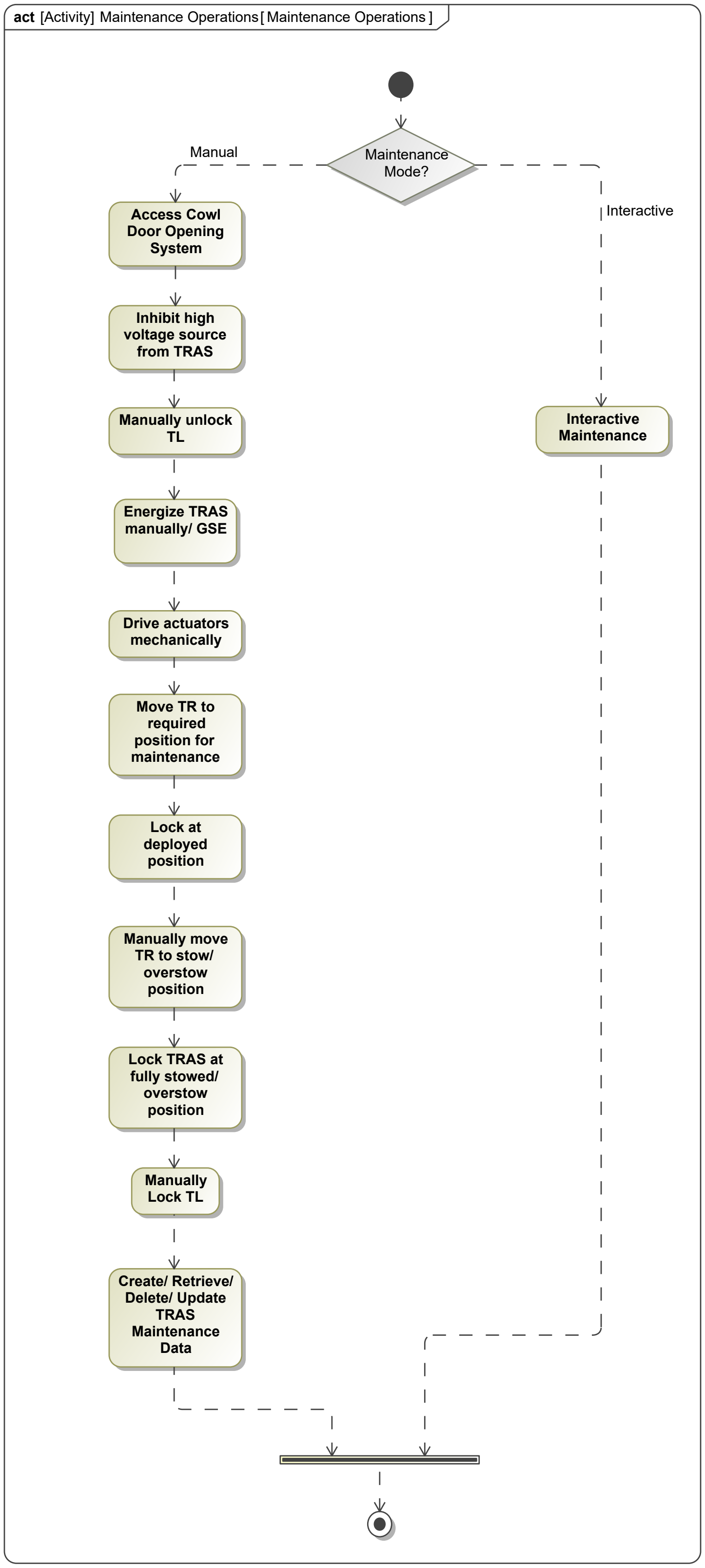
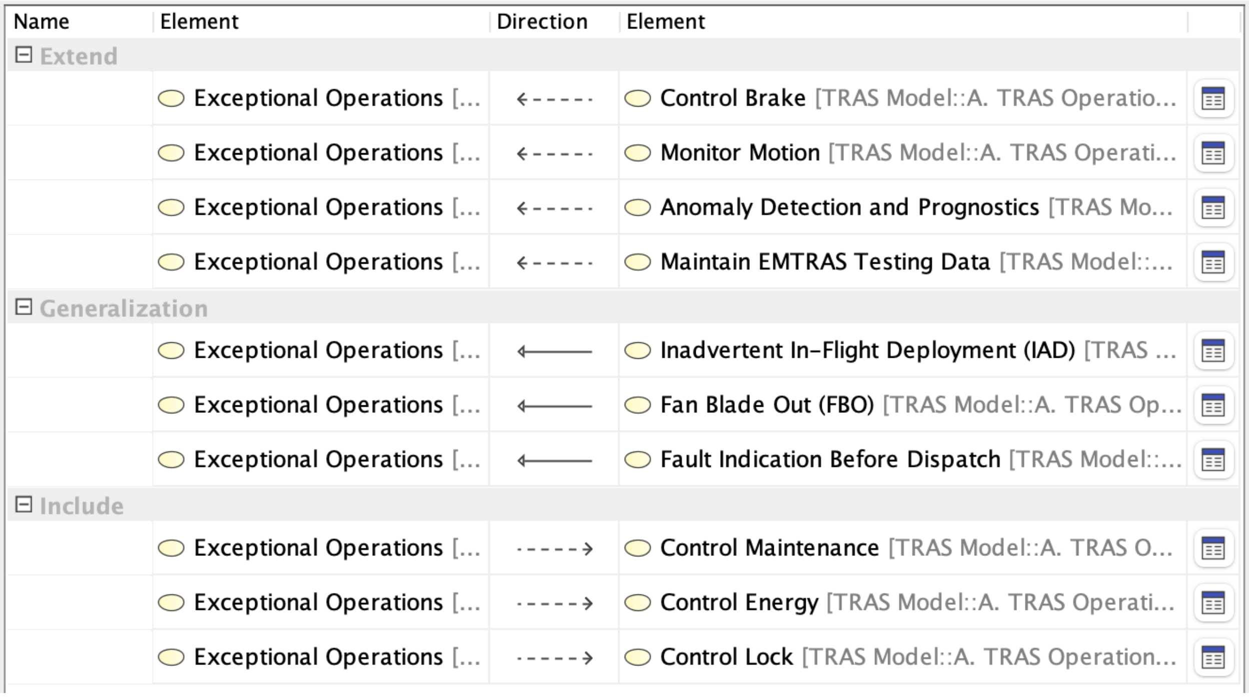

act [Activity] Maintenance Operations [Maintenance Operations]
Maintenance Mode?
Manual
Interactive
Access Cowl Door Opening System
Inhibit high voltage source from TRAS
Manually unlock TL
Energize TRAS manually/ GSE
Drive actuators mechanically
Move TR to required position for maintenance
Lock at deployed position
Manually move TR to stow/ overstow position
Lock TRAS at fully stowed/ overstow position
Manually Lock TL
Create/ Retrieve/ Delete/ Update TRAS Maintenance Data
Interactive Maintenance

Figure 3.19: Alternative/Maintenance Scenario



Name	Element	Direction	Element
⊟ Extend			
	Exceptional Operations [...	<- - - - -	Control Brake [TRAS Model::A. TRAS Operatio...
	Exceptional Operations [...	<- - - - -	Monitor Motion [TRAS Model::A. TRAS Operati...
	Exceptional Operations [...	<- - - - -	Anomaly Detection and Prognostics [TRAS Mo...
	Exceptional Operations [...	<- - - - -	Maintain EMTRAS Testing Data [TRAS Model::...
⊟ Generalization			
	Exceptional Operations [...	<———	Inadvertent In-Flight Deployment (IAD) [TRAS ...
	Exceptional Operations [...	<———	Fan Blade Out (FBO) [TRAS Model::A. TRAS Op...
	Exceptional Operations [...	<———	Fault Indication Before Dispatch [TRAS Model::...
⊟ Include			
	Exceptional Operations [...	- - - - ->	Control Maintenance [TRAS Model::A. TRAS O...
	Exceptional Operations [...	- - - - ->	Control Energy [TRAS Model::A. TRAS Operati...
	Exceptional Operations [...	- - - - ->	Control Lock [TRAS Model::A. TRAS Operation...

Figure 3.20: Exceptional operations Use Case specifications

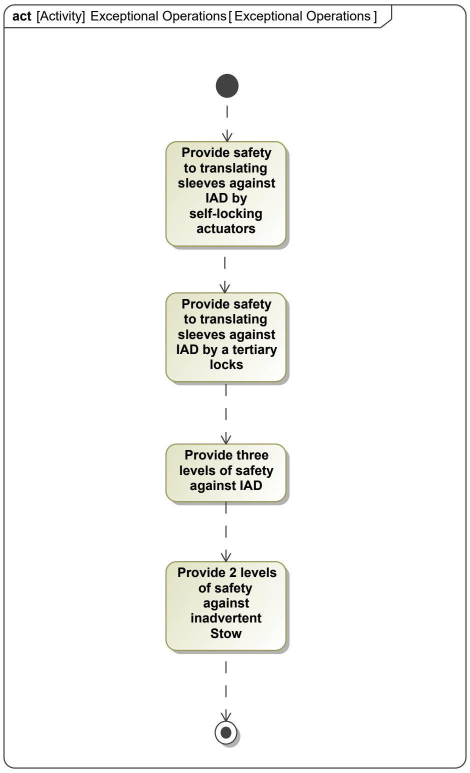
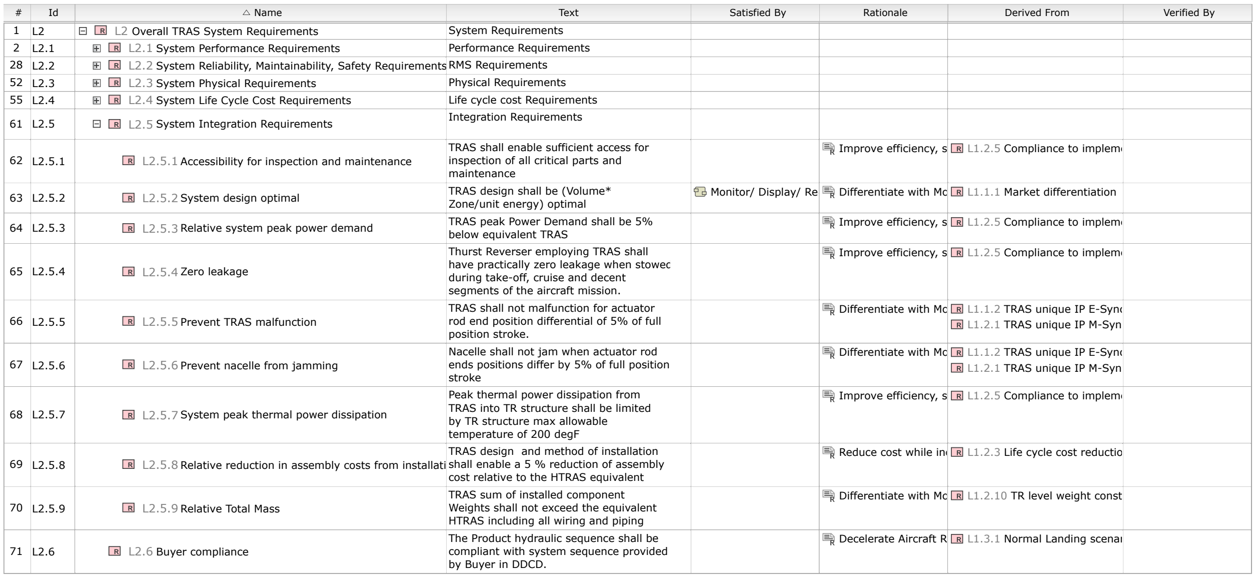

act [Activity] Exceptional Operations [Exceptional Operations]
Provide safety to translating sleeves against IAD by self-locking actuators
Provide safety to translating sleeves against IAD by a tertiary locks
Provide three levels of safety against IAD
Provide 2 levels of safety against inadvertent Stow

Figure 3.21: Exceptional/Fault Scenario



#	Id	△ Name	Text	Satisfied By	Rationale	Derived From	Verified By
1	L2	⊟ R L2 Overall TRAS System Requirements	System Requirements				
2	L2.1	⊞ R L2.1 System Performance Requirements	Performance Requirements				
28	L2.2	⊞ R L2.2 System Reliability, Maintainability, Safety Requirements	RMS Requirements				
52	L2.3	⊞ R L2.3 System Physical Requirements	Physical Requirements				
55	L2.4	⊞ R L2.4 System Life Cycle Cost Requirements	Life cycle cost Requirements				
61	L2.5	⊟ R L2.5 System Integration Requirements	Integration Requirements				
62	L2.5.1	R L2.5.1 Accessibility for inspection and maintenance	TRAS shall enable sufficient access for inspection of all critical parts and maintenance		Improve efficiency, s	R L1.2.5 Compliance to implem	
63	L2.5.2	R L2.5.2 System design optimal	TRAS design shall be (Volume* Zone/unit energy) optimal	Monitor/ Display/ Re	Differentiate with Mc	R L1.1.1 Market differentiation	
64	L2.5.3	R L2.5.3 Relative system peak power demand	TRAS peak Power Demand shall be 5% below equivalent TRAS		Improve efficiency, s	R L1.2.5 Compliance to implem	
65	L2.5.4	R L2.5.4 Zero leakage	Thurst Reverser employing TRAS shall have practically zero leakage when stowec during take-off, cruise and decent segments of the aircraft mission.		Improve efficiency, s	R L1.2.5 Compliance to implem	
66	L2.5.5	R L2.5.5 Prevent TRAS malfunction	TRAS shall not malfunction for actuator rod end position differential of 5% of full position stroke.		Differentiate with Mc	R L1.1.2 TRAS unique IP E-Sync R L1.2.1 TRAS unique IP M-Syn	
67	L2.5.6	R L2.5.6 Prevent nacelle from jamming	Nacelle shall not jam when actuator rod ends positions differ by 5% of full position stroke		Differentiate with Mc	R L1.1.2 TRAS unique IP E-Sync R L1.2.1 TRAS unique IP M-Syn	
68	L2.5.7	R L2.5.7 System peak thermal power dissipation	Peak thermal power dissipation from TRAS into TR structure shall be limited by TR structure max allowable temperature of 200 degF		Improve efficiency, s	R L1.2.5 Compliance to implem	
69	L2.5.8	R L2.5.8 Relative reduction in assembly costs from installati	TRAS design and method of installation shall enable a 5 % reduction of assembly cost relative to the HTRAS equivalent		Reduce cost while in	R L1.2.3 Life cycle cost reductio	
70	L2.5.9	R L2.5.9 Relative Total Mass	TRAS sum of installed component Weights shall not exceed the equivalent HTRAS including all wiring and piping		Differentiate with Mc	R L1.2.10 TR level weight const	
71	L2.6	R L2.6 Buyer compliance	The Product hydraulic sequence shall be compliant with system sequence provided by Buyer in DDCD.		Decelerate Aircraft R	R L1.3.1 Normal Landing scenar	

Figure 3.22: System requirements specification complete list

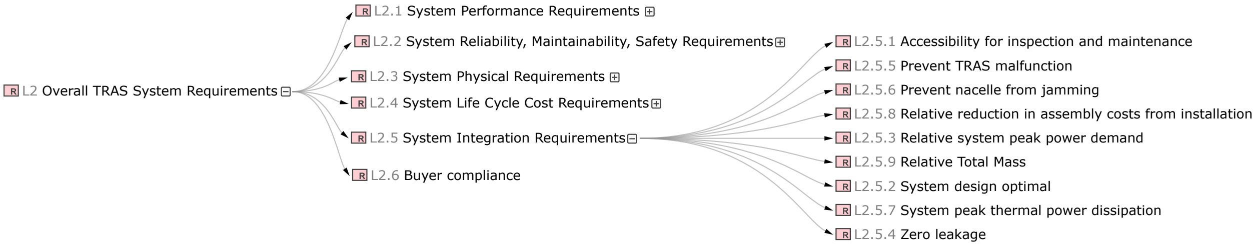

L2 Overall TRAS System Requirements
L2.1 System Performance Requirements
L2.2 System Reliability, Maintainability, Safety Requirements
L2.3 System Physical Requirements
L2.4 System Life Cycle Cost Requirements
L2.5 System Integration Requirements
L2.6 Buyer compliance
L2.5.1 Accessibility for inspection and maintenance
L2.5.5 Prevent TRAS malfunction
L2.5.6 Prevent nacelle from jamming
L2.5.8 Relative reduction in assembly costs from installation
L2.5.3 Relative system peak power demand
L2.5.9 Relative Total Mass
L2.5.2 System design optimal
L2.5.7 System peak thermal power dissipation
L2.5.4 Zero leakage

Figure 3.23: Reviewing requirements in system using containment map

is represented in a SysML requirements table as shown in Fig. 3.22. Additional columns are added to the table to capture aspects of traceability, such as the source of a requirement, whether from requirements in mission or requirements in legacy systems, the model elements that satisfy and verify them.

The overall requirements in system can also be reviewed using a containment map as shown in Fig. 3.23. With limitations on current knowledge about the system of interest, some system requirements are missing. Future collaboration efforts shall broaden the scope to yield a concise set of requirements in system.

2.1.2 SRS derivation from MRS: Fig. 3.24 specifies what SRS is derived from which MRS using a SysML derive requirements matrix.

This activity provides a basis for cascading/flow down of requirements in system from the requirements in mission level. The artifact communicates that the source of information required to elicit the SRS is provided by the MRS and helps identifies any orphan requirement in the system that does not have any parent requirement in mission.

2.1.3 SRS allocation to System structure/behavior: Each requirement is synthesized to identify the model elements that satisfy the requirement specification. A satisfy dependency is created with each model element that should be related to the system requirements to satisfy them. These satisfy dependencies are reviewed in a SysML satisfy requirements matrix as shown in Fig. 3.25. Note that this artifact helps identify the gaps in requirement specifications to ensure that missing gaps are filled with appropriate model elements so that a requirement in system is not missed.

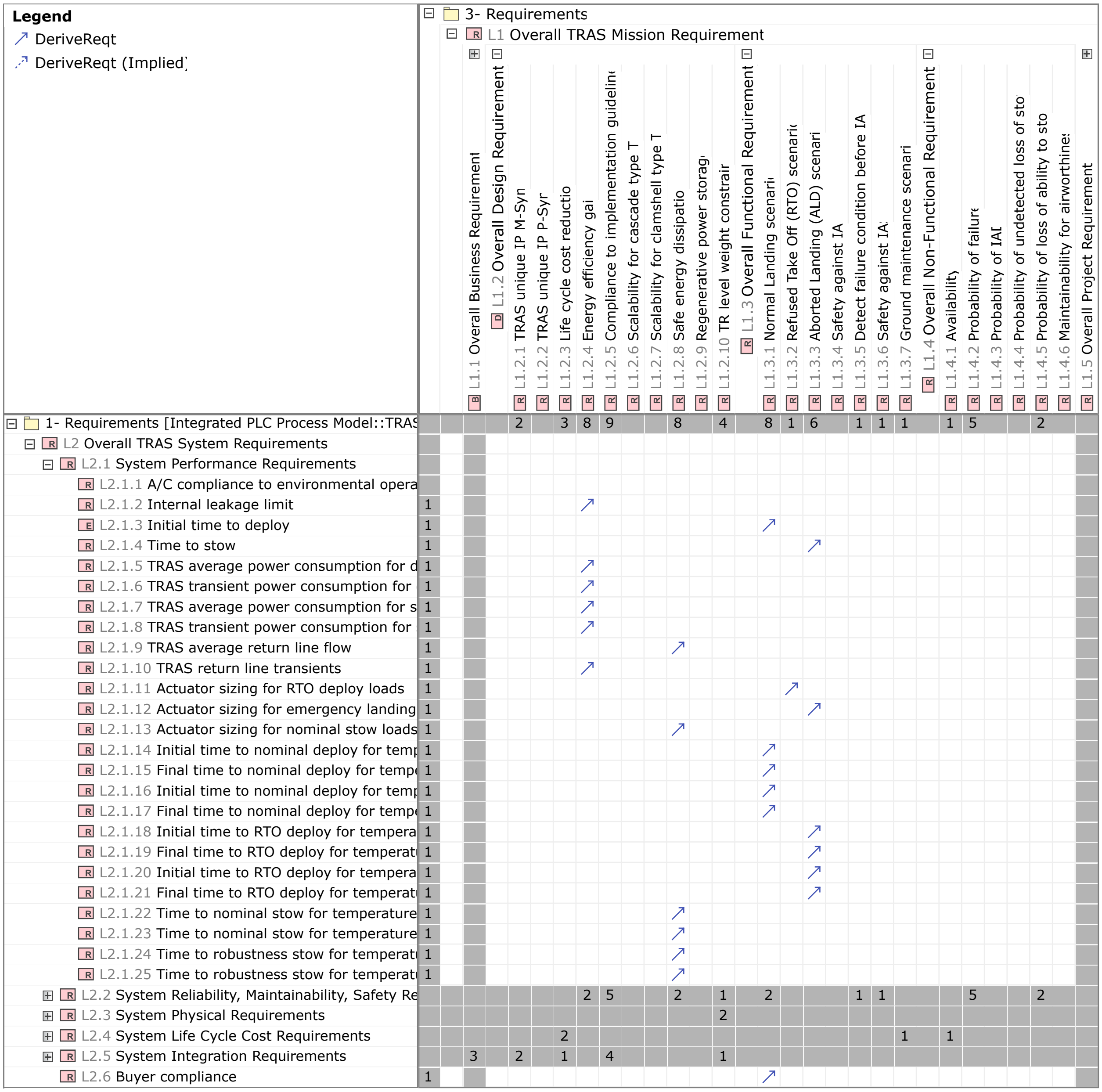
2.2 System Structure: The system of interest is identified as *TRAS* by visualizing the system under development distinct from the product sold to the end customer. The problem definition is formulated by capturing the structural and functional aspects of the system. This activity helps partition the knowledge content about the system by identifying aspects that are known from what is unknown. The technical measures of performance are captured as black-box specifications, followed by system decomposition into logical subsystems responsible for the system's functional aspects.

2.2.1 System Black Box specifications:

The performance measures that specify the system characteristics are identified (What they are? and what they do?). These characteristics are classified as technical performance measures (TPM) and grouped under the stereotype, «tpm». Each system TPM is created as a value property and specified with suitable quantity kind and its unit of measurement. These system specifications are captured in BDD as shown in Fig. 3.26.

2.2.2 System Decomposition external structure:

The system is decomposed into logical subsystems based on the generic functions that they intend to perform. The functional definition is first established to ensure generic functionality



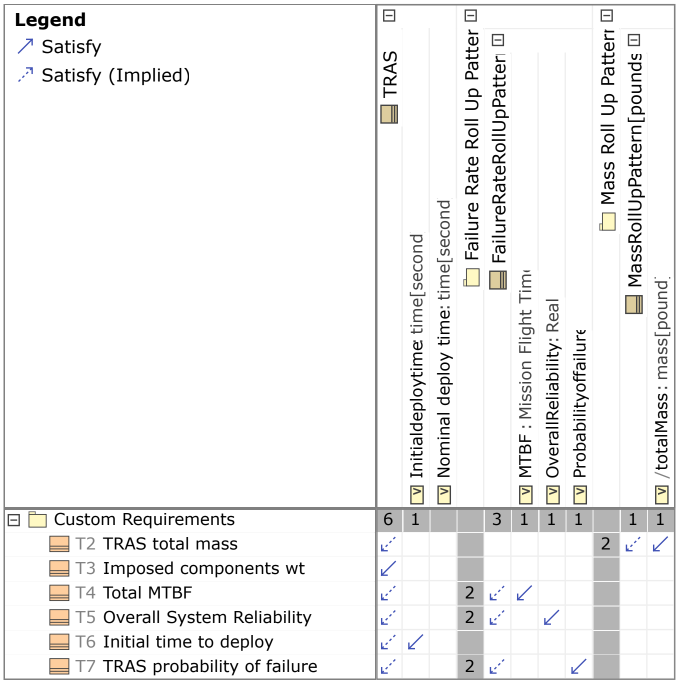
Legend

↗ DeriveReqt

⇢ DeriveReqt (Implied)

	3- Requirements	L1 Overall TRAS Mission Requirement	L1.1 Overall Business Requirement	L1.2 Overall Design Requirement	L1.2.1 TRAS unique IP M-Syn	L1.2.2 TRAS unique IP P-Syn	L1.2.3 Life cycle cost reductio	L1.2.4 Energy efficiency gai	L1.2.5 Compliance to implementation guideline	L1.2.6 Scalability for cascade type T	L1.2.7 Scalability for clamshell type T	L1.2.8 Safe energy dissipatio	L1.2.9 Regenerative power storag	L1.2.10 TR level weight constrai	L1.3 Overall Functional Requirement	L1.3.1 Normal Landing scenari	L1.3.2 Refused Take Off (RTO) scenari	L1.3.3 Aborted Landing (ALD) scenari	L1.3.4 Safety against IA	L1.3.5 Detect failure condition before IA	L1.3.6 Safety against IA	L1.3.7 Ground maintenance scenari	L1.4 Overall Non-Functional Requirement	L1.4.1 Availability	L1.4.2 Probability of failure	L1.4.3 Probability of IA	L1.4.4 Probability of undetected loss of sto	L1.4.5 Probability of loss of ability to sto	L1.4.6 Maintainability for airworthines	L1.5 Overall Project Requirement
1- Requirements [Integrated PLC Process Model::TRAS					2		3	8	9			8		4		8	1	6		1	1	1		1	5			2		
L2 Overall TRAS System Requirements																														
L2.1 System Performance Requirements																														
L2.1.1 A/C compliance to environmental opera																														
L2.1.2 Internal leakage limit	1							↗																						
L2.1.3 Initial time to deploy	1															↗														
L2.1.4 Time to stow	1																	↗												
L2.1.5 TRAS average power consumption for d	1							↗																						
L2.1.6 TRAS transient power consumption for	1							↗																						
L2.1.7 TRAS average power consumption for s	1							↗																						
L2.1.8 TRAS transient power consumption for	1							↗																						
L2.1.9 TRAS average return line flow	1											↗																		
L2.1.10 TRAS return line transients	1							↗																						
L2.1.11 Actuator sizing for RTO deploy loads	1																↗													
L2.1.12 Actuator sizing for emergency landing	1																	↗												
L2.1.13 Actuator sizing for nominal stow loads	1											↗																		
L2.1.14 Initial time to nominal deploy for temp	1															↗														
L2.1.15 Final time to nominal deploy for tempe	1															↗														
L2.1.16 Initial time to nominal deploy for temp	1															↗														
L2.1.17 Final time to nominal deploy for tempe	1															↗														
L2.1.18 Initial time to RTO deploy for tempera	1																	↗												
L2.1.19 Final time to RTO deploy for temperat	1																	↗												
L2.1.20 Initial time to RTO deploy for tempera	1																	↗												
L2.1.21 Final time to RTO deploy for temperat	1																	↗												
L2.1.22 Time to nominal stow for temperature	1											↗																		
L2.1.23 Time to nominal stow for temperature	1											↗																		
L2.1.24 Time to robustness stow for temperat	1											↗																		
L2.1.25 Time to robustness stow for temperat	1											↗																		
L2.2 System Reliability, Maintainability, Safety Re								2	5			2		1		2				1	1				5			2		
L2.3 System Physical Requirements														2																
L2.4 System Life Cycle Cost Requirements							2															1		1						
L2.5 System Integration Requirements			3		2		1		4					1																
L2.6 Buyer compliance	1															↗														

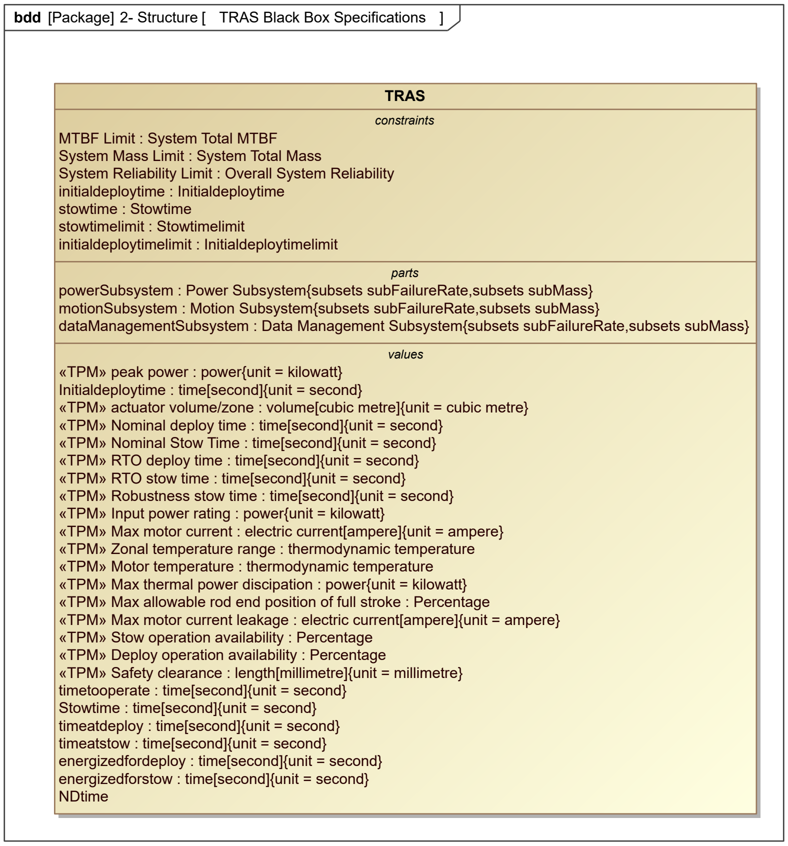
Figure 3.24: System requirements derivation from mission requirements using derive requirements matrix



Legend
↗ Satisfy
⇢ Satisfy (Implied)

	TRAS	Initialdeploytime: time[second	Nominal deploy time: time[second	Failure Rate Roll Up Patter	FailureRateRollUpPatter	MTBF : Mission Flight Tim	OverallReliability: Real	Probabilityoffailure	Mass Roll Up Patter	MassRollUpPattern[pounds	/totalMass : mass[pound
Custom Requirements	6	1			3	1	1	1		1	1
T2 TRAS total mass	⇠								2	⇠	↙
T3 Imposed components wt	↙										
T4 Total MTBF	⇠			2	⇠	↙					
T5 Overall System Reliability	⇠			2	⇠		↙				
T6 Initial time to deploy	⇠	↙									
T7 TRAS probability of failure	⇠			2	⇠			↙			

Figure 3.25: System requirements satisfied by system attributes



bdd [Package] 2- Structure [TRAS Black Box Specifications]

TRAS

constraints
MTBF Limit : System Total MTBF
System Mass Limit : System Total Mass
System Reliability Limit : Overall System Reliability
initialdeploytime : Initialdeploytime
stowtime : Stowtime
stowtimelimit : Stowtimelimit
initialdeploytimelimit : Initialdeploytimelimit

parts
powerSubsystem : Power Subsystem{subsets subFailureRate,subsets subMass}
motionSubsystem : Motion Subsystem{subsets subFailureRate,subsets subMass}
dataManagementSubsystem : Data Management Subsystem{subsets subFailureRate,subsets subMass}

values
«TPM» peak power : power{unit = kilowatt}
Initialdeploytime : time[second]{unit = second}
«TPM» actuator volume/zone : volume[cubic metre]{unit = cubic metre}
«TPM» Nominal deploy time : time[second]{unit = second}
«TPM» Nominal Stow Time : time[second]{unit = second}
«TPM» RTO deploy time : time[second]{unit = second}
«TPM» RTO stow time : time[second]{unit = second}
«TPM» Robustness stow time : time[second]{unit = second}
«TPM» Input power rating : power{unit = kilowatt}
«TPM» Max motor current : electric current[ampere]{unit = ampere}
«TPM» Zonal temperature range : thermodynamic temperature
«TPM» Motor temperature : thermodynamic temperature
«TPM» Max thermal power discipation : power{unit = kilowatt}
«TPM» Max allowable rod end position of full stroke : Percentage
«TPM» Max motor current leakage : electric current[ampere]{unit = ampere}
«TPM» Stow operation availability : Percentage
«TPM» Deploy operation availability : Percentage
«TPM» Safety clearance : length[millimetre]{unit = millimetre}
timetooperate : time[second]{unit = second}
Stowtime : time[second]{unit = second}
timeatdeploy : time[second]{unit = second}
timeatstow : time[second]{unit = second}
energizedfordeploy : time[second]{unit = second}
energizedforstow : time[second]{unit = second}
NDtime

Figure 3.26: System black-box specifications

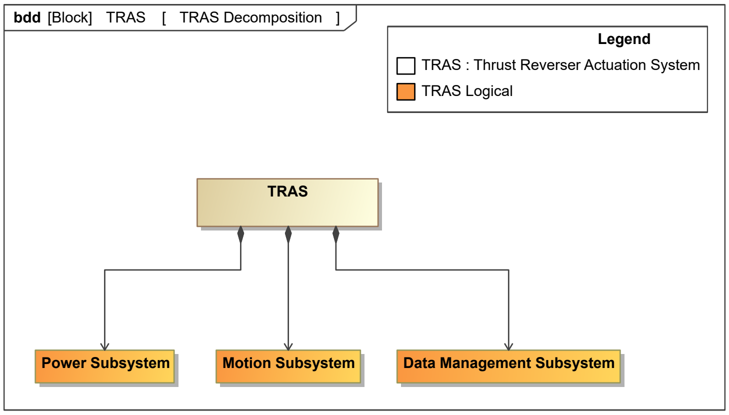

bdd [Block] TRAS [TRAS Decomposition]
Legend
TRAS : Thrust Reverser Actuation System
TRAS Logical
TRAS
Power Subsystem
Motion Subsystem
Data Management Subsystem

Figure 3.27: Decomposing system of interest to logical subsystems

that would not change with architecture alternatives. Before defining each subsystem, good thoughts are considered to ensure that the functional design remains generic irrespective of the solving system architecture. Thoughts along the lines of, 'What are the some of the objectives that the subsystem is trying to achieve?', 'What are the subsystem's functions that do not change with a specific solution/implementation?', 'How will the subsystem functions change with the architecture?' The system of interest is decomposed into *Power Subsystem*, *Data Management Subsystem* and *Motion Subsystem* as depicted in Fig. 3.27, and modeled as blocks in a BDD. This artifact presents an external view of the subsystems composed within the system without elaborating the internal structure.

The *Power Subsystem* performs the necessary functionality to provide electrical power to energize the system and aggregates, as well to de-energize the system.

Data Management Subsystem communicates data in the form of analog and digital signals necessary for the system operation while coordinating with subsystems and components that perform those functions.

Motion Subsystem performs the necessary actuation functionality and synchronizes the extension and retraction functions necessary to deploy/stow the thrust reverser (TR).

2.2.3 System Decomposition internal structure:

After defining the subsystems as blocks, the internal structure is elaborated by reusing the blocks as parts in an IBD to represent the interconnections. Specific interactions that take place through the interfaces are also defined. The interfaces are arranged in the order that follows the sequence of system behavior during operation. The system receives the discrete command and feedback signals required for intended operation as discussed in the context and interacts with the subsystems, communicating signals through SIO and DIO interfaces to perform the required functions. In response to the signals, the subsystems invoke the necessary input to support its intended functionality.

2.3 System Functions: The functional baseline is established by decomposing the operational scenario/Use Case as discussed in Sec. 1.3, for instance, Normal Landing operation to describe the system functions. The modeling effort ensures that the black-box system functions realize alternative TRAS system design concepts.

2.3.1 System Behavior:

The basic system behavior to operate TRAS is established in this effort. TRAS receives the necessary control signals/commands from the aircraft controller as defined in Fig. 3.13 and interacts with the subsystems to achieve the functionality for successful system operation. A signal and activity library is established in a modular fashion to model the system behavior. Figure 3.29 shows the send and feedback signals that are reused in SysML activity diagrams and modeled as send signal actions and accept event actions that trigger the stateful system behavior. The artifact facilitates the fundamental behavior that the system commands the respective subsystem in a set sequence to perform tasks, considering the delay, and awaits feedback that confirms that each task is performed. This modeling effort also establishes the necessary functions that analyze the duration of system operation.

2.3.2 System Functional Analysis and Allocation:

TRAS invokes the logical control functions to facilitate basic system operation. The primary system functions that are performed within the main/basic operational scenario discussed in Sec. 1.3 are further analyzed to understand the system behavior in detail. Figure 3.30 depicts the system functions modeled as nested actions in a SysML Activity Diagram. The identified system functions are further decomposed to specific send command/accept feedback actions.

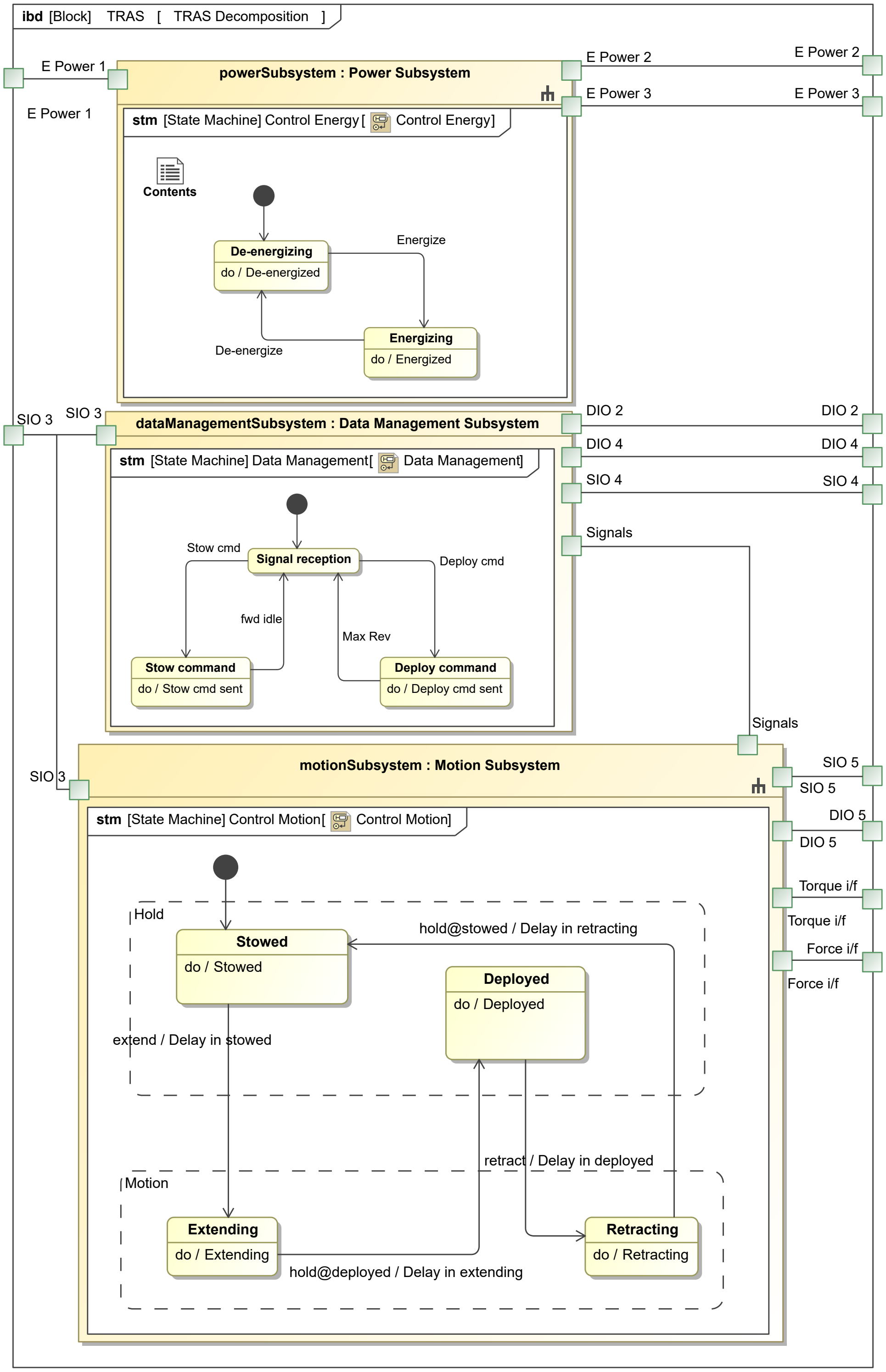


Figure 3.28: Interconnection and interaction of logical subsystems

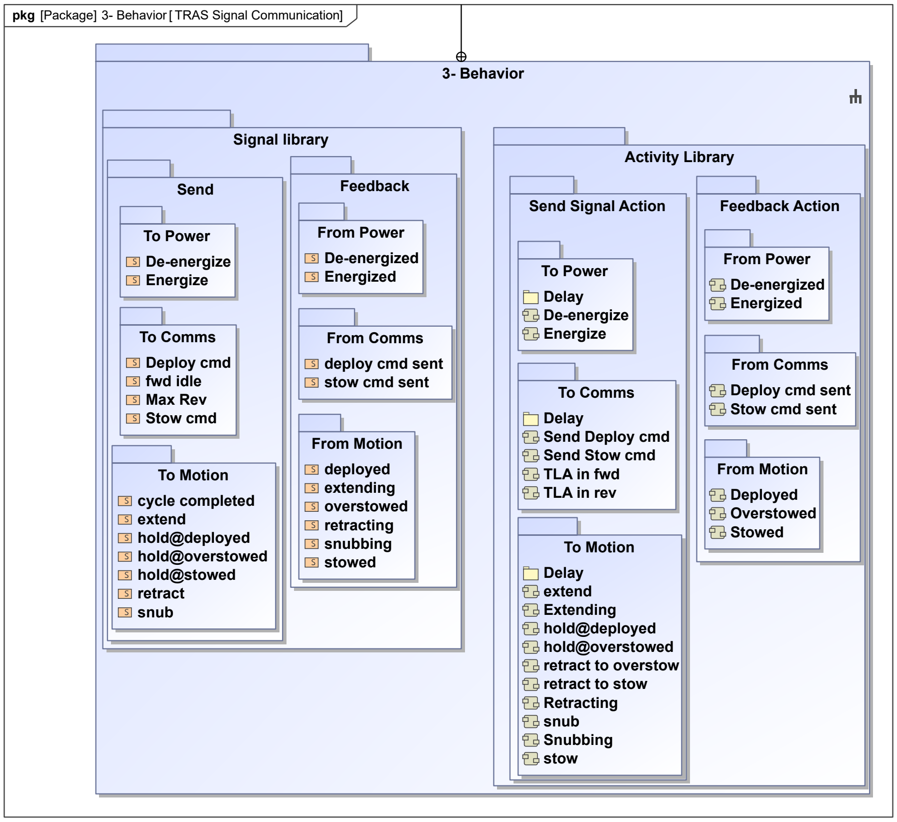


Figure 3.29: System behavior signaling architecture

These refined functions are then allocated to the subsystems that perform more such functions in a SysML Activity Diagram. As illustrated in Fig. 3.31, the functional allocation is modeled using Vertical Hierarchical Swimlanes, where the system and subsystems are placed at the top headers. The functions are allocated to respective subsystems and connected using flows to convey the sequence in which the functions are performed.

2.4 System Parameters: The system parameters are characterized using constraints to specify the expected values of system measures and expressed as equations. The technical performance measures (TPMs) that define the metrics for successful system operation are quantified with rules and expressions. Some constraints are extracted from the system requirements (SRS) as discussed in Sec. 2.1. These constraint blocks previously defined in a BDD are now reused as constraint properties in a parametric diagram (PAR), as depicted in Fig. 3.32 and tied to the system TPM values.

2.5 Verification by OEM: The technical evaluation processes to verify that TRAS design meet the needs are planned for future effort.

3. **TRAS Logical** (Subsystem Level) In this level of abstraction, the logical subsystems are identified and its functions are explicitly specified to explore alternatives. This is where the subsystem alternatives and range of functional components are identified.

3.1 Subsystem Requirements: Future research and collaboration efforts shall yield a comprehensive list of requirements in subsystem.

3.2 Subsystem Structure: The Motion Subsystem is identified to provide the required functionality to deploy/stow TRAS during system operation.

3.2.1 Subsystem Alternative Configurations:

The logical motion subsystem configurations corresponding to the motion synchronization mechanism are assembled by making selection of suitable LRU variants. The three alternative candidates of technology considered for the motion subsystem are modeled as blocks in a

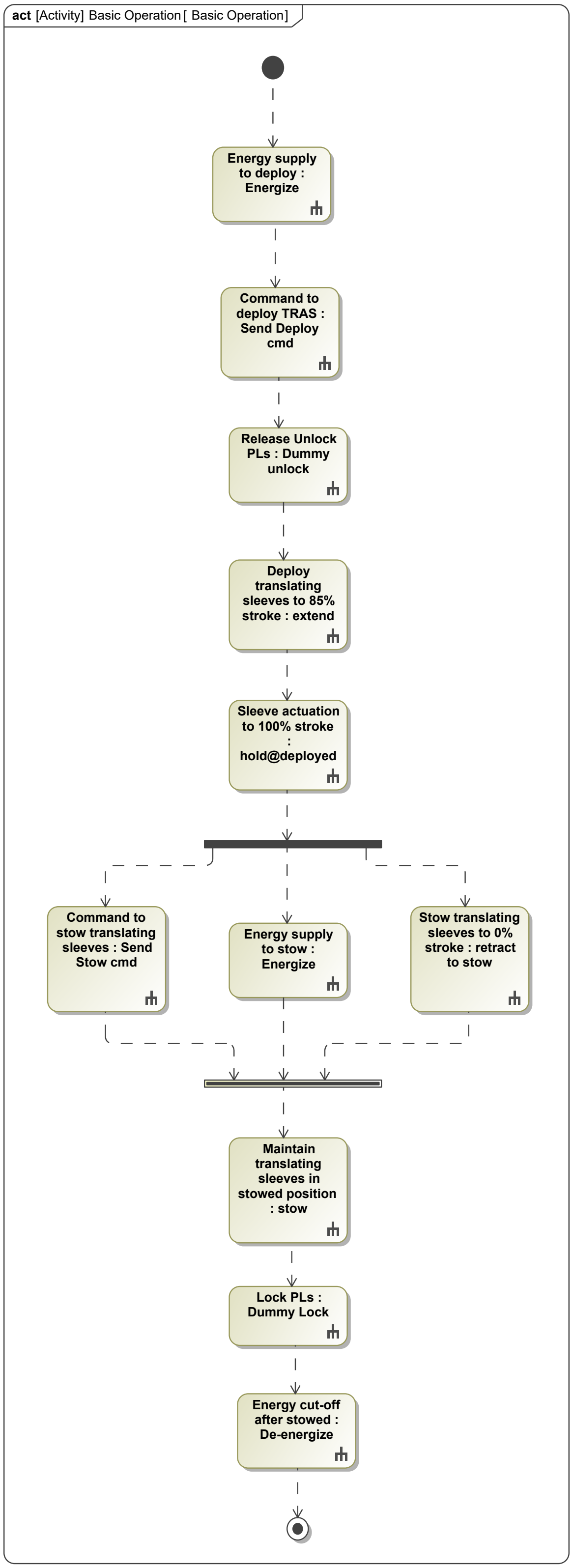
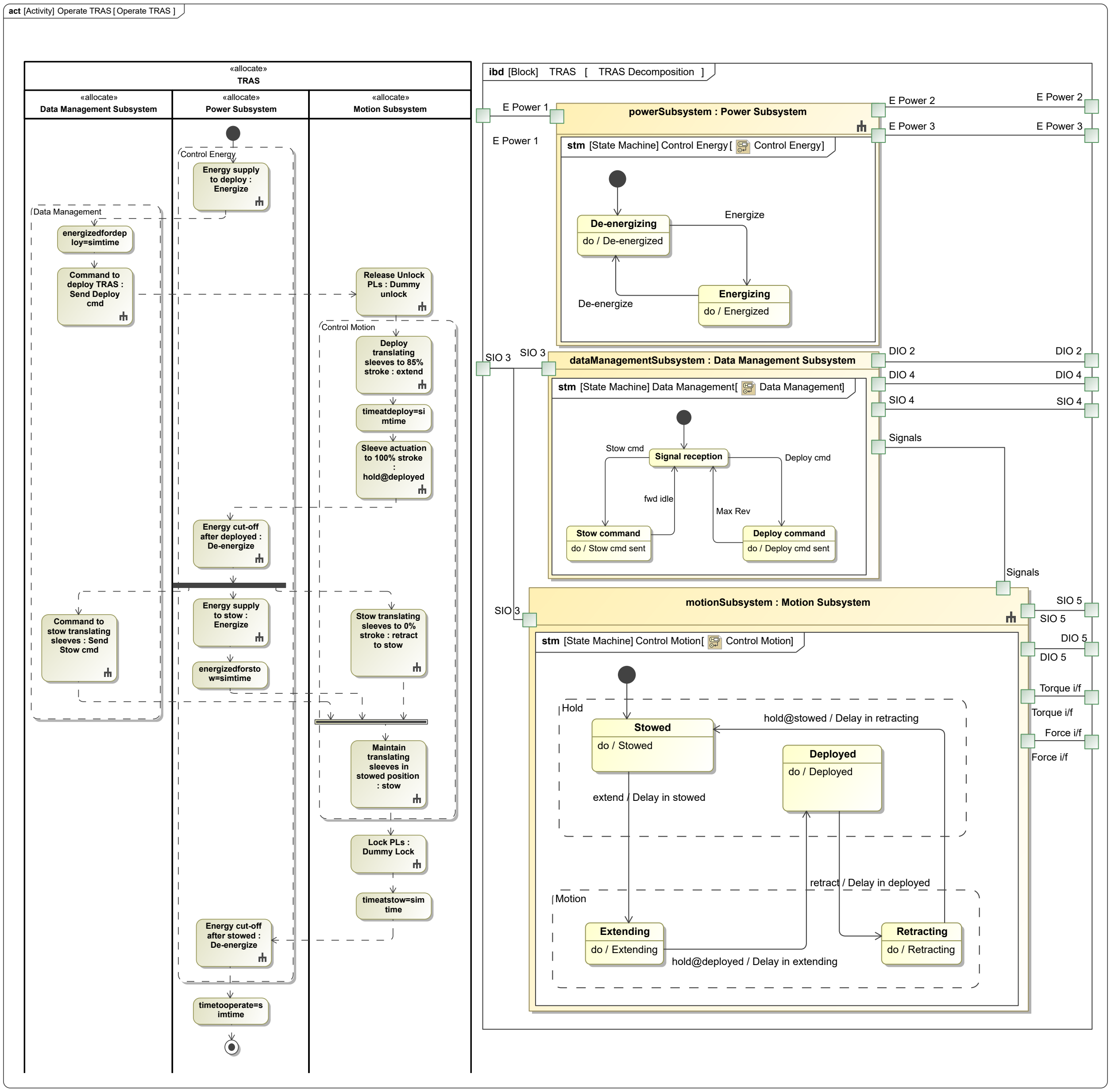

act [Activity] Basic Operation [Basic Operation]
Energy supply to deploy : Energize
Command to deploy TRAS : Send Deploy cmd
Release Unlock PLs : Dummy unlock
Deploy translating sleeves to 85% stroke : extend
Sleeve actuation to 100% stroke : hold@deployed
Command to stow translating sleeves : Send Stow cmd
Energy supply to stow : Energize
Stow translating sleeves to 0% stroke : retract to stow
Maintain translating sleeves in stowed position : stow
Lock PLs : Dummy Lock
Energy cut-off after stowed : De-energize

Figure 3.30: TRAS functional analysis



act [Activity] Operate TRAS [Operate TRAS]
«allocate»
TRAS
«allocate»
Data Management Subsystem
«allocate»
Power Subsystem
«allocate»
Motion Subsystem
Control Energy
Energy supply to deploy : Energize
Data Management
energizedfordeploy=simtime
Command to deploy TRAS : Send Deploy cmd
Release Unlock PLs : Dummy unlock
Control Motion
Deploy translating sleeves to 85% stroke : extend
timeatdeploy=simtime
Sleeve actuation to 100% stroke : hold@deployed
Energy cut-off after deployed : De-energize
Command to stow translating sleeves : Send Stow cmd
Energy supply to stow : Energize
Stow translating sleeves to 0% stroke : retract to stow
energizedforstow=simtime
Maintain translating sleeves in stowed position : stow
Lock PLs : Dummy Lock
timeatstow=simtime
Energy cut-off after stowed : De-energize
timetooperate=simtime
ibd [Block] TRAS [TRAS Decomposition]
E Power 1
E Power 1
powerSubsystem : Power Subsystem
E Power 2
E Power 2
E Power 3
E Power 3
stm [State Machine] Control Energy[Control Energy]
De-energizing
do / De-energized
Energize
Energizing
do / Energized
De-energize
SIO 3
SIO 3
dataManagementSubsystem : Data Management Subsystem
DIO 2
DIO 2
DIO 4
DIO 4
SIO 4
SIO 4
Signals
stm [State Machine] Data Management[Data Management]
Stow cmd
Signal reception
Deploy cmd
fwd idle
Max Rev
Stow command
do / Stow cmd sent
Deploy command
do / Deploy cmd sent
Signals
SIO 3
motionSubsystem : Motion Subsystem
SIO 5
SIO 5
DIO 5
DIO 5
Torque i/f
Torque i/f
Force i/f
Force i/f
stm [State Machine] Control Motion[Control Motion]
Hold
Stowed
do / Stowed
hold@stowed / Delay in retracting
Deployed
do / Deployed
extend / Delay in stowed
retract / Delay in deployed
Motion
Extending
do / Extending
hold@deployed / Delay in extending
Retracting
do / Retracting

Figure 3.31: Decomposing operational Use Case under normal landing scenario to system functions

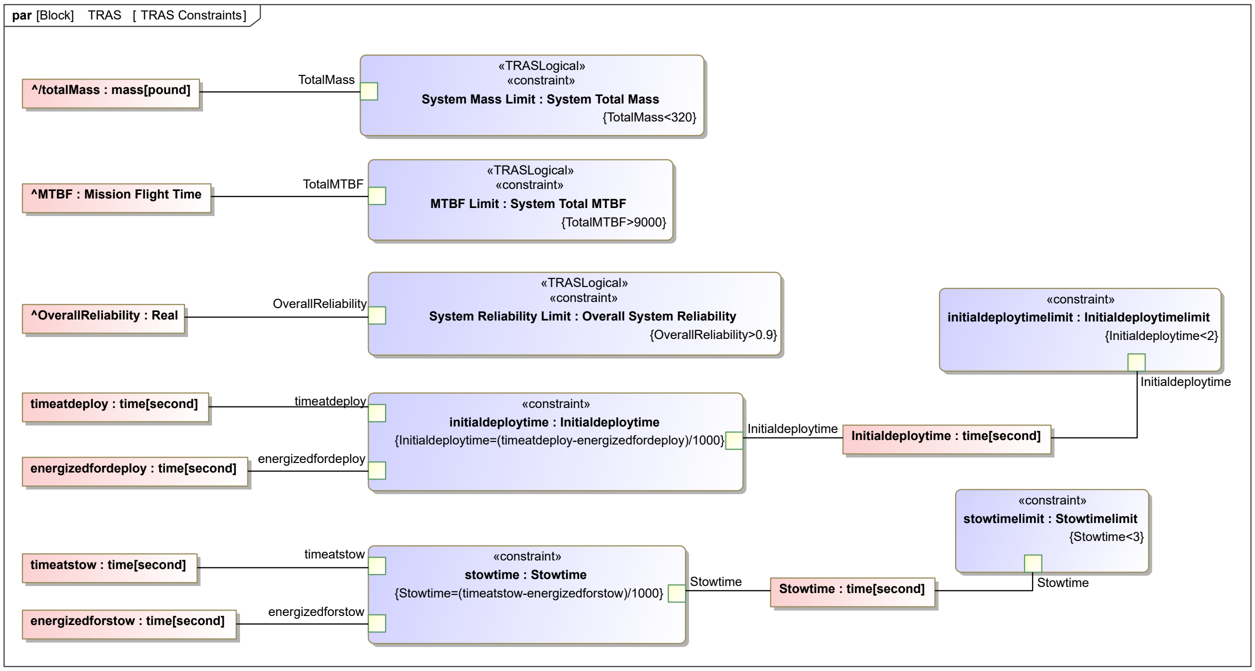
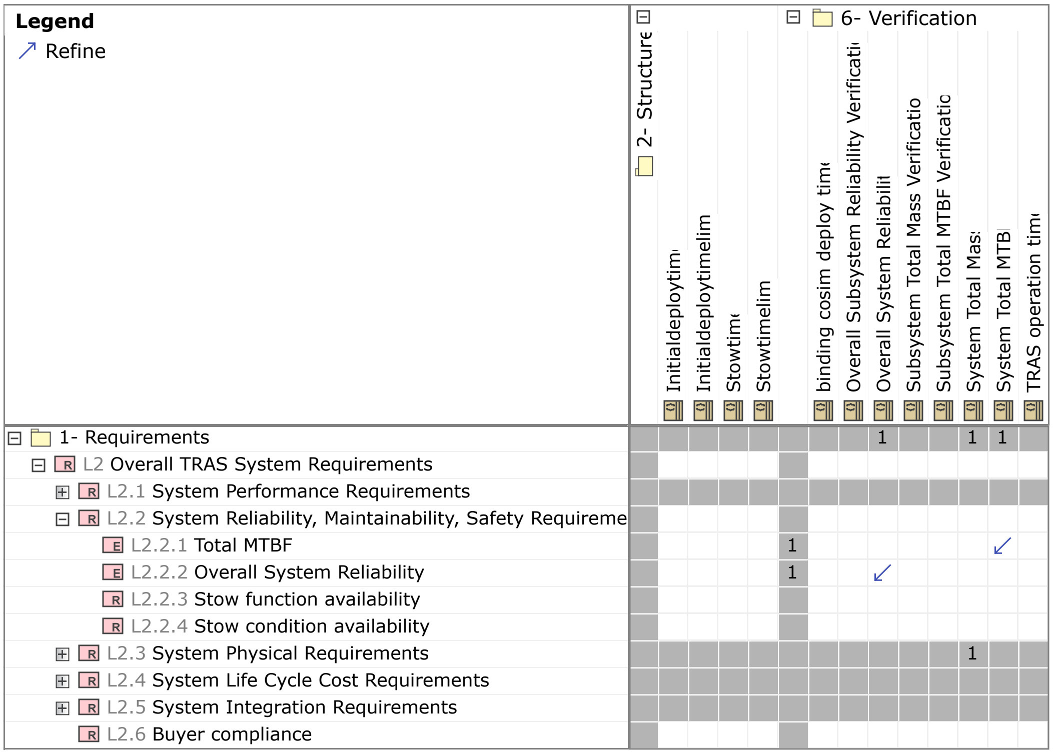

par [Block] TRAS [TRAS Constraints]
^/totalMass : mass[pound]
TotalMass
«TRASLogical»
«constraint»
System Mass Limit : System Total Mass
{TotalMass<320}
^MTBF : Mission Flight Time
TotalMTBF
«TRASLogical»
«constraint»
MTBF Limit : System Total MTBF
{TotalMTBF>9000}
^OverallReliability : Real
OverallReliability
«TRASLogical»
«constraint»
System Reliability Limit : Overall System Reliability
{OverallReliability>0.9}
«constraint»
initialdeploytimelimit : Initialdeploytimelimit
{Initialdeploytime<2}
Initialdeploytime
timeatdeploy : time[second]
timeatdeploy
«constraint»
initialdeploytime : Initialdeploytime
{Initialdeploytime=(timeatdeploy-energizedfordeploy)/1000}
Initialdeploytime
Initialdeploytime : time[second]
energizedfordeploy : time[second]
energizedfordeploy
«constraint»
stowtimelimit : Stowtimelimit
{Stowtime<3}
timeatstow : time[second]
timeatstow
«constraint»
stowtime : Stowtime
{Stowtime=(timeatstow-energizedforstow)/1000}
Stowtime
Stowtime : time[second]
Stowtime
energizedforstow : time[second]
energizedforstow

Figure 3.32: Applying constraints to system performance parameters



Legend ↗ Refine	⊟ 2- Structure	Initialdeploytim	Initialdeploytimelim	Stowtim	Stowtimelim	⊟ 6- Verification	binding cosim deploy tim	Overall Subsystem Reliability Verificati	Overall System Reliabilit	Subsystem Total Mass Verificatio	Subsystem Total MTBF Verificatio	System Total Mas	System Total MTB	TRAS operation tim
⊟ 1- Requirements									1			1	1	
⊟ R L2 Overall TRAS System Requirements														
⊞ R L2.1 System Performance Requirements														
⊟ R L2.2 System Reliability, Maintainability, Safety Requireme														
E L2.2.1 Total MTBF						1							↙	
E L2.2.2 Overall System Reliability						1			↙					
R L2.2.3 Stow function availability														
R L2.2.4 Stow condition availability														
⊞ R L2.3 System Physical Requirements												1		
⊞ R L2.4 System Life Cycle Cost Requirements														
⊞ R L2.5 System Integration Requirements														
R L2.6 Buyer compliance														

Figure 3.33: System constraints refining requirements in system

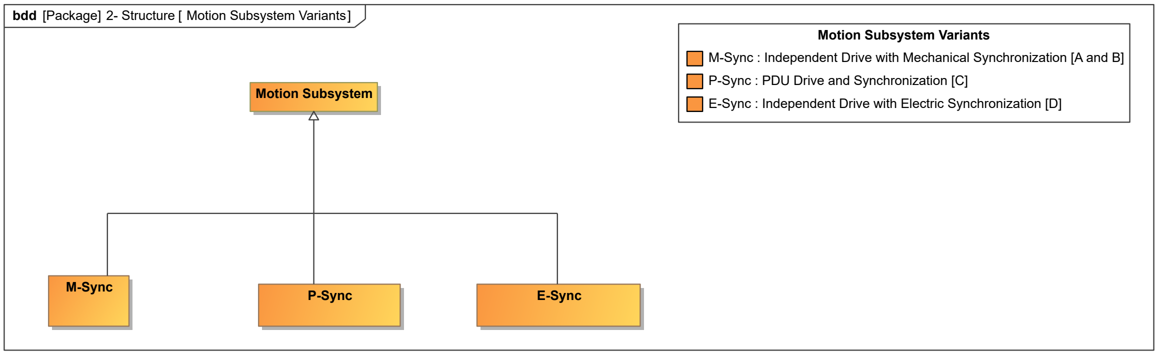

bdd [Package] 2- Structure [Motion Subsystem Variants]
Motion Subsystem Variants
🟧 M-Sync : Independent Drive with Mechanical Synchronization [A and B]
🟧 P-Sync : PDU Drive and Synchronization [C]
🟧 E-Sync : Independent Drive with Electric Synchronization [D]
Motion Subsystem
M-Sync
P-Sync
E-Sync

Figure 3.34: Alternative configurations of the synchronized motion subsystem

BDD as shown in Fig. 3.34. Each alternative represents a sub-assembly of the synchronized motion subsystem.

Add a note or details around the variants of the configurations to understand what the architecture is (addressing the types) corresponding to synchronization.

3.2.2 Subsystem Decomposition external structure: Attention is steered to focus on the synchronization mechanism, and the motion subsystem is then decomposed to modular, physical components, referred to as Line Replaceable Units (LRUs). A separate package is created to establish the LRU catalog used as a separate model. The LRUs can be accessed to constitute different configurations in the TRAS model. This modeling practice demonstrates an excellent example of model reuse. As shown in Fig. 3.35, all possible LRUs are composed within the motion subsystem, and the multiplicity (number of instances that could exist at a time) is defined for each LRU. Each LRU is designed to a set specification and may differ in application pertinent to a configuration of the synchronized mechanism within the motion subsystem. These are categorized among variants as per the application. A said configuration of the Motion Subsystem is realized after integrating and assembling a set combination of LRU variants.

3.3 Subsystem Functions: The subsystem functions are specified with stateful behaviors to depict the different conditions in which the subsystems exist at an instance of time..

3.3.1 Subsystem Stateful Behavior

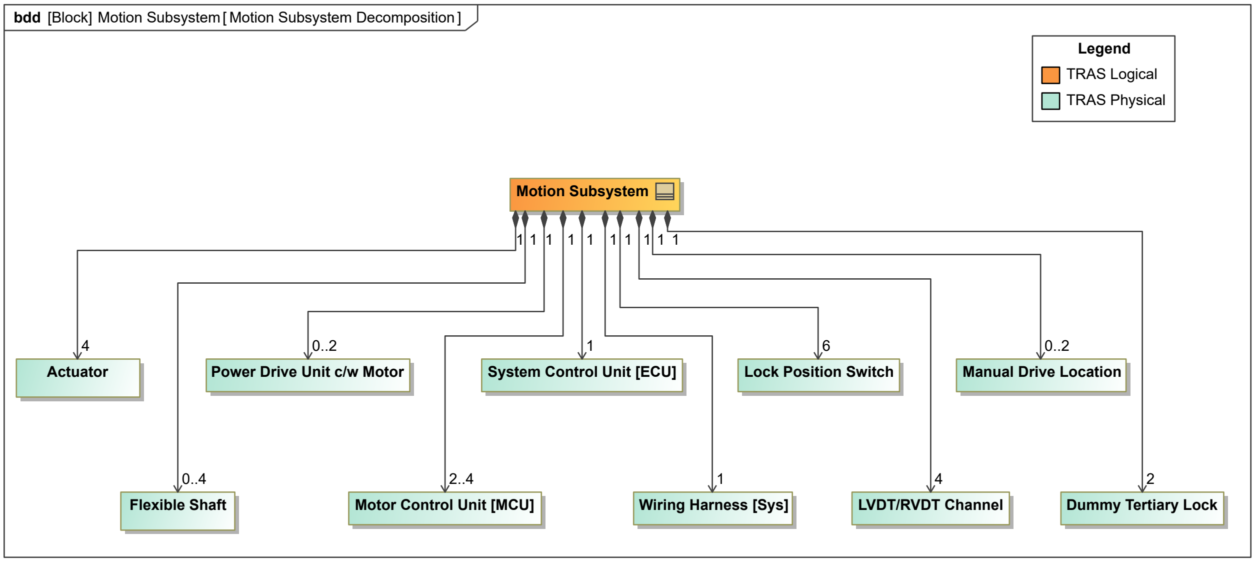


Figure 3.35: Decomposing the logical motion subsystem to physical components

A state depicts a significant condition of an entity at a particular instance of time. A stateful behavior is modeled to elaborate the different states in which the subsystem exists at a particular instance of time. While in a state, a subsystem can perform certain activities until interrupted by a disturbance/trigger that causes a transition to switch from one state to another state. A stateful behavior is specified as the main behavior for each of the three subsystems using SysML State Machine Diagram (STM). The three subsystem state machines created within this modeling activity are:

1. Control Energy/Power States: The Power Subsystem begins its operation in the state de-energizing by default, where it performs an activity to de-energize the system. After receiving a trigger signal that commands "energize", the power subsystem transitions from the current state to the energizing state, where it performs an activity to energize the system. The power subsystem switches between states several times to control the power supply for the system's successful operation.

2. Control Data Management/Signaling States: The Data Management Subsystem enters the state of signal reception by default, where it receives command/feedback from the system without interruption in the form of discrete analog signals. On receiving the deploy/stow command from the system, it transitions to *Deploy Command* state or *Stow Command* state

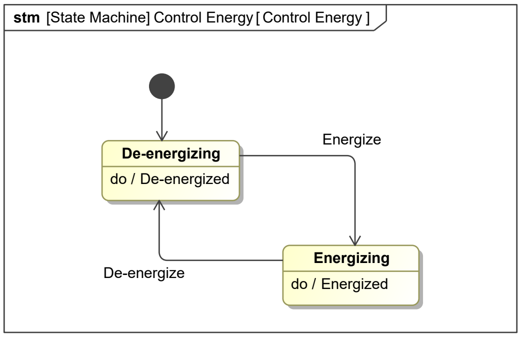


Figure 3.36: Power subsystem states while performing control energy function

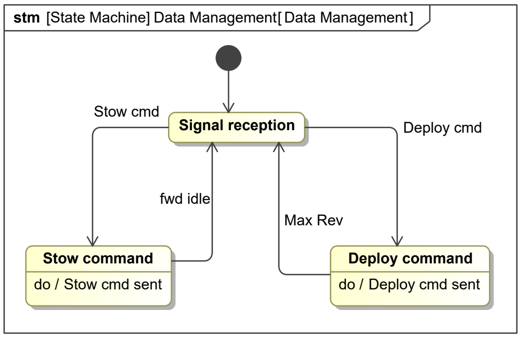


Figure 3.37: Data management states

where a feedback signal action is executed that confirms that the deploy/stow command is sent. Only when the signals indicate the Max Rev/Fwd Idle position, it transitions back to the *Signal Reception* state.

3. Control Motion/Motion States: During system operation, the *Motion Subsystem* remains either in the ’hold’ condition or ’actuating’ condition. Four descriptive states are created to model the stateful behavior of the motion subsystem, namely *stowed*, *extending*, *deployed* and *retracting*. As illustrated in Fig. 3.38, the motion subsystem is initially assumed to enter the stowed state, where it sends a feedback signal confirming that it is stowed. On receiving the extend signal command, it transitions to the extending state, where the system begins actuating and sends extending feedback.

Another possible operating condition that the system can exist at an instance is the ’jammed’ state. However, the jammed state is excluded from the current scope but future modeling and collaboration efforts shall address the additional states and transitions.

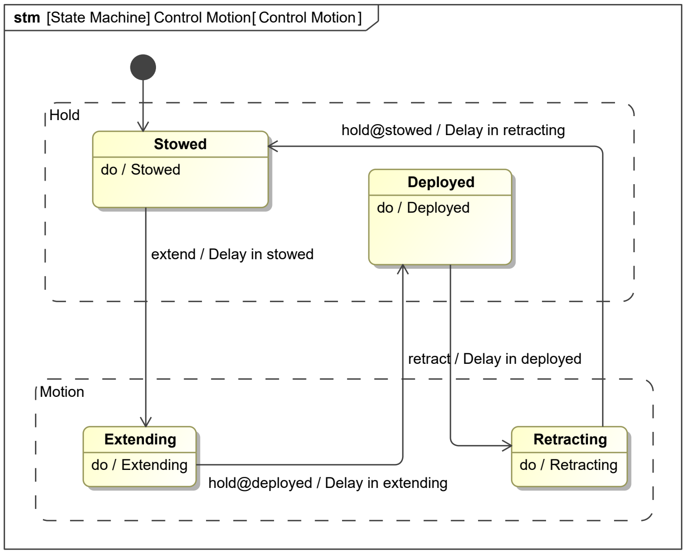

stm [State Machine] Control Motion[Control Motion]
Hold
Stowed
do / Stowed
hold@stowed / Delay in retracting
Deployed
do / Deployed
extend / Delay in stowed
retract / Delay in deployed
Motion
Extending
do / Extending
hold@deployed / Delay in extending
Retracting
do / Retracting

Figure 3.38: Motion subsystem states

3.4 Subsystem Parameters: The subsystems parameters are characterized using constraints to specify the limiting values of subsystem measures for mass, MTBF and probability of failure.

3.5 Verification by OEM: Verification methods at the component level, such as inspection, analysis, test, and demonstration. 1-D analysis, stress analysis like finite element analysis (FEA), or an endurance test are planned for execution in future scope.

4. **TRAS Physical** The TRAS Physical level of abstraction captures the physical characteristics associated with implementation. This is where the component catalog and point design is established.

4.1 LRU Requirement: The component requirement are not developed in this modeling effort.

4.2 LRU Structure: The detailed specification of the component catalog is planned in future effort.

4.2.1 LRU variants in catalog: An LRU catalog is established to capture the tiers of component variants as applicable to the subsystem configurations.

4.3 LRU Functions: The component behavior and functions of specific LRU are modeled in an external engineering analysis tool and integrated within the system model environment.

4.4 LRU Parameters: The component parameter specifications are limited to mass and failure rate values corresponding to the piece-part values obtained from field data. These are typed as de-

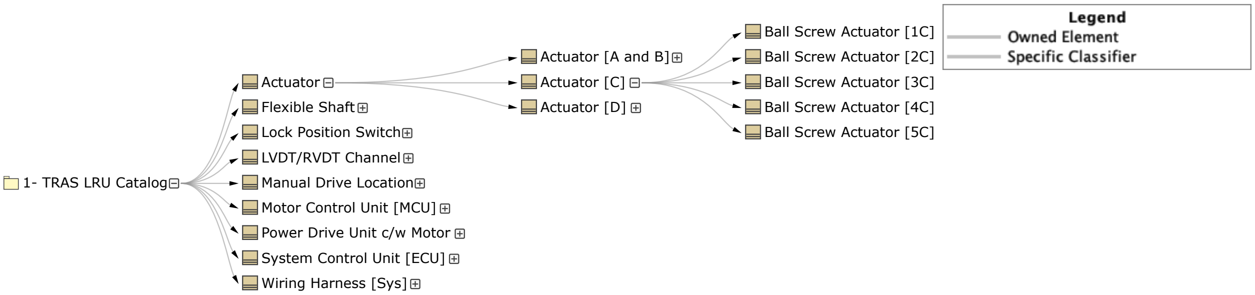


Figure 3.39: Modeling a catalog of line replaceable units

fault/initial parametric values is specified for each LRU within an instance table. The physical characteristics of components are modeled in an external engineering analysis tool and planned for integration within the system model in future effort.

4.5 Verification by OEM/Supplier: The technical evaluation of components with vendors is planned during the future effort.

In conclusion, the MBSE methodology is implemented to develop a unified repository of system information, artifacts and SE products that address (*I2*), (*I3*). Moreover, a common understanding is created among teams of multi-disciplinary stakeholders. Implementing this, the MBSE tools support SE effort to better understand the system and communicate a holistic picture. Therefore, addressing (*I5*).

3.2 Analyze Architecture Alternatives to Setup a Trade-Study

The result obtained on pursuing an architecture development methodology is an abstract system architecture model. The model is then subject to analysis to support the SE effort, such as performing a trade study, managing requirements traceability, etc. The following paragraphs discuss the analysis techniques performed in this modeling effort.

3.2.1 Apply Roll-Up Analysis Patterns for the Key Performance Measures

For a complex system such as TRAS, an early analysis carried out to realize some system-level specifications even before implementation is a valuable verification activity. It provides vital

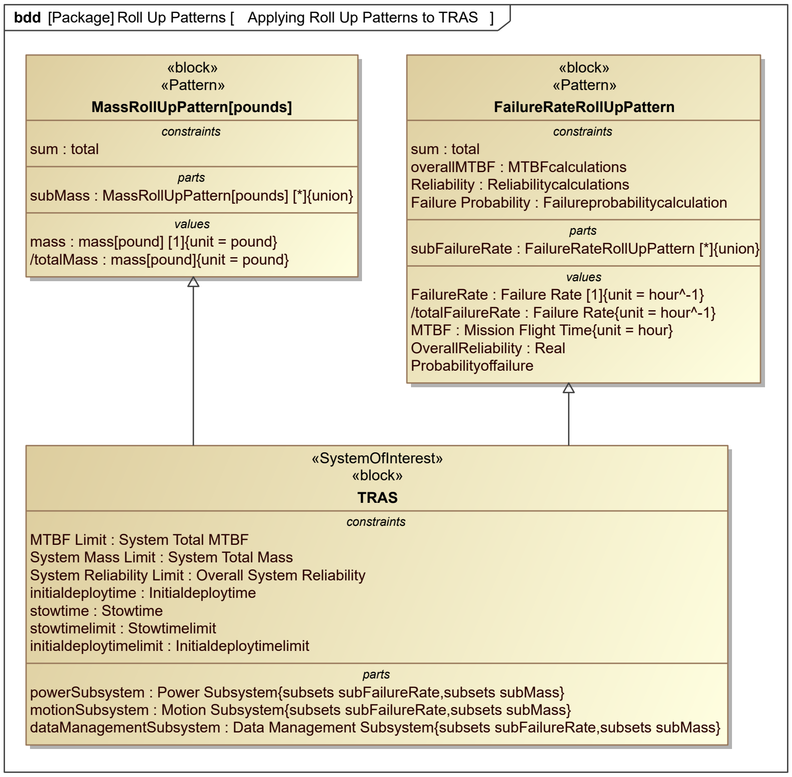

bdd [Package] Roll Up Patterns [Applying Roll Up Patterns to TRAS]
«block»
«Pattern»
MassRollUpPattern[pounds]
constraints
sum : total
parts
subMass : MassRollUpPattern[pounds] [*]{union}
values
mass : mass[pound] [1]{unit = pound}
/totalMass : mass[pound]{unit = pound}
«block»
«Pattern»
FailureRateRollUpPattern
constraints
sum : total
overallMTBF : MTBFcalculations
Reliability : Reliabilitycalculations
Failure Probability : Failureprobabilitycalculation
parts
subFailureRate : FailureRateRollUpPattern [*]{union}
values
FailureRate : Failure Rate [1]{unit = hour^-1}
/totalFailureRate : Failure Rate{unit = hour^-1}
MTBF : Mission Flight Time{unit = hour}
OverallReliability : Real
Probabilityoffailure
«SystemOfInterest»
«block»
TRAS
constraints
MTBF Limit : System Total MTBF
System Mass Limit : System Total Mass
System Reliability Limit : Overall System Reliability
initialdeploytime : Initialdeploytime
stowtime : Stowtime
stowtimelimit : Stowtimelimit
initialdeploytimelimit : Initialdeploytimelimit
parts
powerSubsystem : Power Subsystem{subsets subFailureRate,subsets subMass}
motionSubsystem : Motion Subsystem{subsets subFailureRate,subsets subMass}
dataManagementSubsystem : Data Management Subsystem{subsets subFailureRate,subsets subMass}

Figure 3.40: Applying roll-up analysis patterns to the system's structural hierarchy

support to the SE efforts. Some system specifications are evaluated against the requirements, which will otherwise be known later when integrating different subsystems and components. Performing such calculations manually for the integrated system is laborious, time-consuming, and prone to errors.

The capability of system modeling tool is utilized. The system model is executed to perform automated calculations recursively and repetitively while avoiding errors. The roll-up analysis pattern is one such capability that can facilitate the reuse of a set of equations to perform calculations recursively on the system structural hierarchy and provide the effective/total value of the system parameter. Hence, a static, top-down Roll-Up Analysis Pattern (provided by the tool) is applied to the system structure for evaluating the system specification based on the piece-part values that make up the system. The calculated values are compared against the expected values from associated system requirements to check whether the requirements are satisfied.

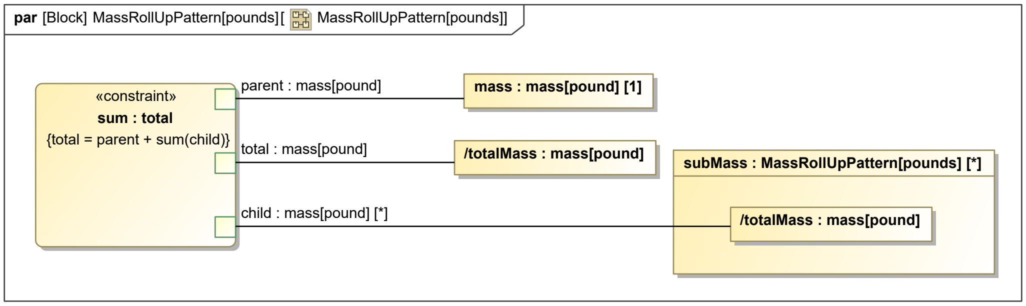


Figure 3.41: Roll-up analysis pattern to evaluate total system mass

As shown in Fig. 3.40, the roll-up analysis pattern is applied for some key technical parameters like mass and failure rate. The patterns are simulated to recursively evaluate the effective/resulting system-level values such as total mass, total failure rate/MTBF, and overall probability of failure based on the piece-part values that collectively make up the system.

Figure 3.41 depicts the mathematical basis for a plain mass roll-up analysis pattern. In this pattern, the mass value for a definite number of child parts is evaluated to provide the cumulative total mass, tagged as a parent mass. The parent value is later considered as child values recursively at the next level of the system decomposition hierarchy. The pattern evaluates the combinations repetitively until it culminates to the system block where it was applied.

Figure 3.42 depicts the mathematical basis for evaluating the total failure rate by a customized roll-up analysis pattern. Due to the safety and reliability analysis limitations within the tool environment, the roll-up pattern is customized to accommodate reliability analysis. In this pattern, other constraint expressions have been added to calculate the effective Mean Time Between Failure (MTBF) and the overall probability of failure. Like a plain roll-up pattern, the total failure rate is calculated from the piece-part failure rates (obtained from field data) and provided as input for all the parts composed within the system. The roll-up pattern considers the assumption that components are connected in series without any redundancy and that if any component fails, the system fails.

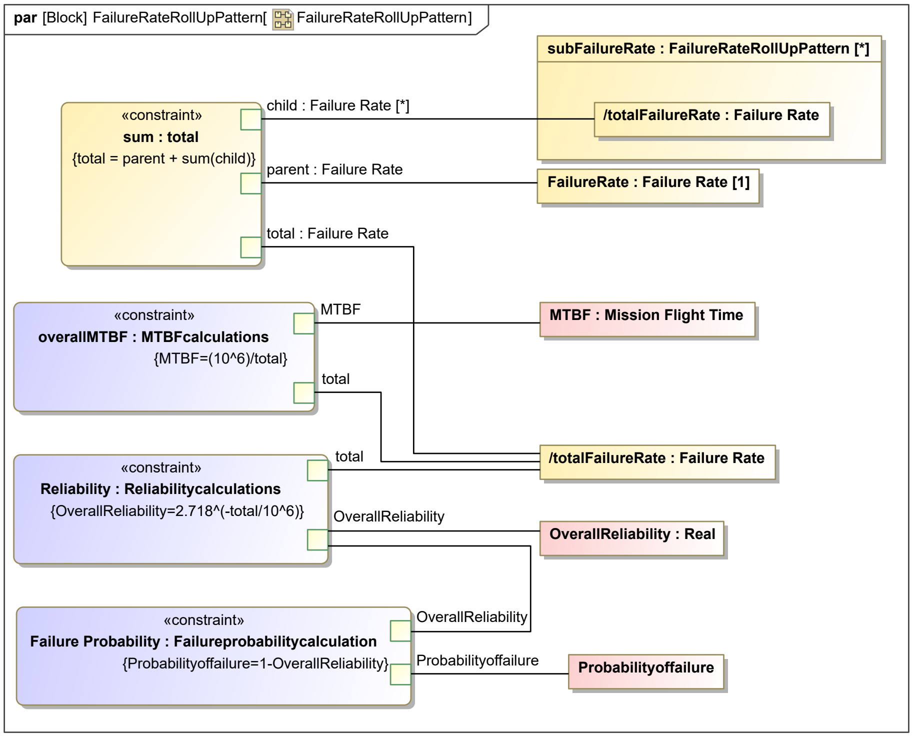
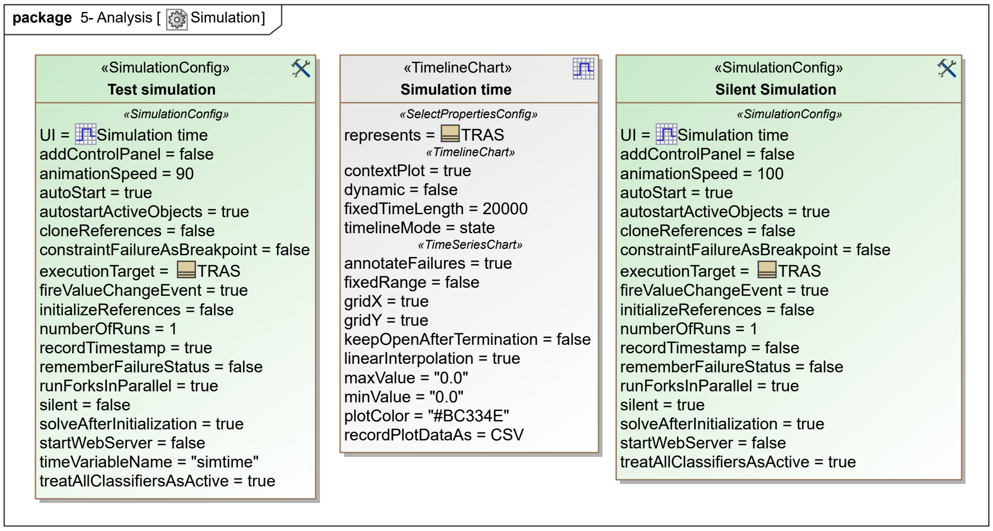

par [Block] FailureRateRollUpPattern[FailureRateRollUpPattern]
subFailureRate : FailureRateRollUpPattern [*]
/totalFailureRate : Failure Rate
«constraint»
sum : total
{total = parent + sum(child)}
child : Failure Rate [*]
parent : Failure Rate
FailureRate : Failure Rate [1]
total : Failure Rate
«constraint»
overallMTBF : MTBFcalculations
{MTBF=(10^6)/total}
MTBF
MTBF : Mission Flight Time
total
total
/totalFailureRate : Failure Rate
«constraint»
Reliability : Reliabilitycalculations
{OverallReliability=2.718^(-total/10^6)}
OverallReliability
OverallReliability : Real
«constraint»
Failure Probability : Failureprobabilitycalculation
{Probabilityoffailure=1-OverallReliability}
OverallReliability
Probabilityoffailure
Probabilityoffailure

Figure 3.42: Roll-up analysis pattern to evaluate total failure rate/MTBF and overall probability of failure based on piece part failure rate

3.2.2 Analyze Time to Operate TRAS

The executable system behavior, discussed in Sec. 2.3 is modified with addition of opaque actions typed by *simtime*. These actions capture and store the simulation time during the simulation of the main behavior. The system performance measure, *time to operate TRAS* specified for TRAS block, is typed by *simtime*. As shown in Fig. 3.43, a simulation configuration is employed. The execution target is typed by the system block that has a main behavior, *operate TRAS*. A timeline chart is specified to represent the active states that are triggered during the simulation of series of activities by the simulation configuration. The behavior is executed, and the timeline chart is plotted to record the operational times. The resulting timeline chart is not presented in this thesis. However, further collaboration will improve the knowledge about timing parameters to plot estimated times. The simulation result is tied to a system requirement, *time to operate TRAS* to analyze whether the system specification meets the requirement. The instances of time to operate TRAS is exported and represented in a separate instance table.



package 5- Analysis [Simulation]

«SimulationConfig»
Test simulation
«SimulationConfig»
UI = Simulation time
addControlPanel = false
animationSpeed = 90
autoStart = true
autostartActiveObjects = true
cloneReferences = false
constraintFailureAsBreakpoint = false
executionTarget = TRAS
fireValueChangeEvent = true
initializeReferences = false
numberOfRuns = 1
recordTimestamp = true
rememberFailureStatus = false
runForksInParallel = true
silent = false
solveAfterInitialization = true
startWebServer = false
timeVariableName = "simtime"
treatAllClassifiersAsActive = true

«TimelineChart»
Simulation time
«SelectPropertiesConfig»
represents = TRAS
«TimelineChart»
contextPlot = true
dynamic = false
fixedTimeLength = 20000
timelineMode = state
«TimeSeriesChart»
annotateFailures = true
fixedRange = false
gridX = true
gridY = true
keepOpenAfterTermination = false
linearInterpolation = true
maxValue = "0.0"
minValue = "0.0"
plotColor = "#BC334E"
recordPlotDataAs = CSV

«SimulationConfig»
Silent Simulation
«SimulationConfig»
UI = Simulation time
addControlPanel = false
animationSpeed = 100
autoStart = true
autostartActiveObjects = true
cloneReferences = false
constraintFailureAsBreakpoint = false
executionTarget = TRAS
fireValueChangeEvent = true
initializeReferences = false
numberOfRuns = 1
recordTimestamp = false
rememberFailureStatus = false
runForksInParallel = true
silent = true
solveAfterInitialization = true
startWebServer = false
treatAllClassifiersAsActive = true

Figure 3.43: Simulation configuration and timeline chart for timing simulation

3.2.3 Comparison of Subsystem Architecture Alternatives

The alternative configurations of the motion subsystem discussed in Sec. 3.2 are instantiated to replicate its implementation considering the components that are composed within each configuration. A said configuration is assembled by making the selection of instances of LRU variants as applicable. An additional dummy configuration is assembled with an inferior specification to visualize the distinction with a master specification. The effective values for total mass, MTBF, and Probability of Failure are evaluated by simulating the roll-up analysis patterns. The resulting values for each motion subsystem alternative configuration obtained as a result of integrating the piece-part values defined in Sec. 4.2 are plotted in an instance table as shown in Fig. 3.46 for comparison.

3.2.4 Comparison of System Implementation Alternatives

An implementation of a particular system-level configuration is realized by integrating the subsystems. Owing to the limitation in knowledge about the subsystems, the scope of this modeling effort, as discussed in Sec. 3.2, limits capturing the data pertinent to the motion subsystem, but placeholders for power and data management subsystems are provided. Similar to the comparison demonstrated in Sec. 3.2.3, now the implementation of system-level configurations are instantiated

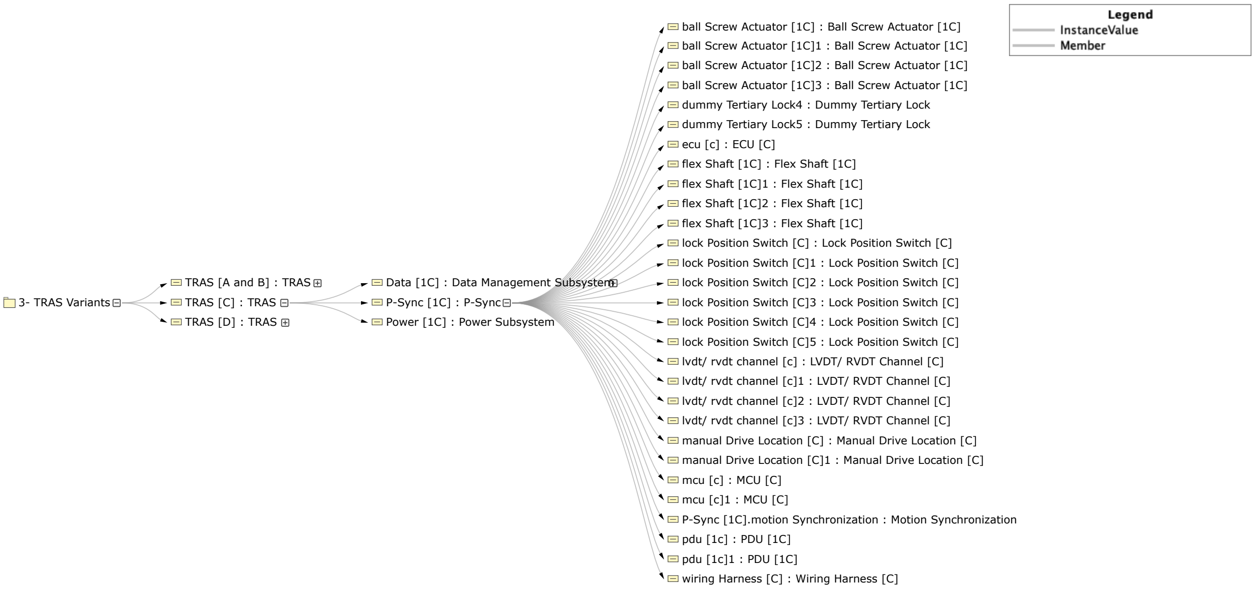

Legend
InstanceValue
Member
3- TRAS Variants
TRAS [A and B] : TRAS
TRAS [C] : TRAS
TRAS [D] : TRAS
Data [1C] : Data Management Subsystem
P-Sync [1C] : P-Sync
Power [1C] : Power Subsystem
ball Screw Actuator [1C] : Ball Screw Actuator [1C]
ball Screw Actuator [1C]1 : Ball Screw Actuator [1C]
ball Screw Actuator [1C]2 : Ball Screw Actuator [1C]
ball Screw Actuator [1C]3 : Ball Screw Actuator [1C]
dummy Tertiary Lock4 : Dummy Tertiary Lock
dummy Tertiary Lock5 : Dummy Tertiary Lock
ecu [c] : ECU [C]
flex Shaft [1C] : Flex Shaft [1C]
flex Shaft [1C]1 : Flex Shaft [1C]
flex Shaft [1C]2 : Flex Shaft [1C]
flex Shaft [1C]3 : Flex Shaft [1C]
lock Position Switch [C] : Lock Position Switch [C]
lock Position Switch [C]1 : Lock Position Switch [C]
lock Position Switch [C]2 : Lock Position Switch [C]
lock Position Switch [C]3 : Lock Position Switch [C]
lock Position Switch [C]4 : Lock Position Switch [C]
lock Position Switch [C]5 : Lock Position Switch [C]
lvdt/ rvdt channel [c] : LVDT/ RVDT Channel [C]
lvdt/ rvdt channel [c]1 : LVDT/ RVDT Channel [C]
lvdt/ rvdt channel [c]2 : LVDT/ RVDT Channel [C]
lvdt/ rvdt channel [c]3 : LVDT/ RVDT Channel [C]
manual Drive Location [C] : Manual Drive Location [C]
manual Drive Location [C]1 : Manual Drive Location [C]
mcu [c] : MCU [C]
mcu [c]1 : MCU [C]
P-Sync [1C].motion Synchronization : Motion Synchronization
pdu [1c] : PDU [1C]
pdu [1c]1 : PDU [1C]
wiring Harness [C] : Wiring Harness [C]

Figure 3.44: Visualizing an instance of TRAS configuration implementation

to evaluate the subsystems and their alternatives. An instance of a system configuration is assembled by making the selection of an instance of each subsystem. The effective system values for total mass, MTBF, and probability of failure are evaluated by executing the system block. The resulting instances of a system-level implementation are represented within an instance table as demonstrated in Fig. 3.47. The overall implementation of system configurations is visualized in an instance map as illustrated in Fig. 3.44.

Consequently, this analysis points to *(I3)* by providing necessary decision support capability.

3.3 Manage and Trace System Requirements

This section elaborates the modeling effort undertaken to establish traceability relationships between requirements and model elements to manage and track the requirements in a trade study. Future efforts shall build upon current progress to establish such relationships across the series of requirements to ensure that a requirement is complete and managed.

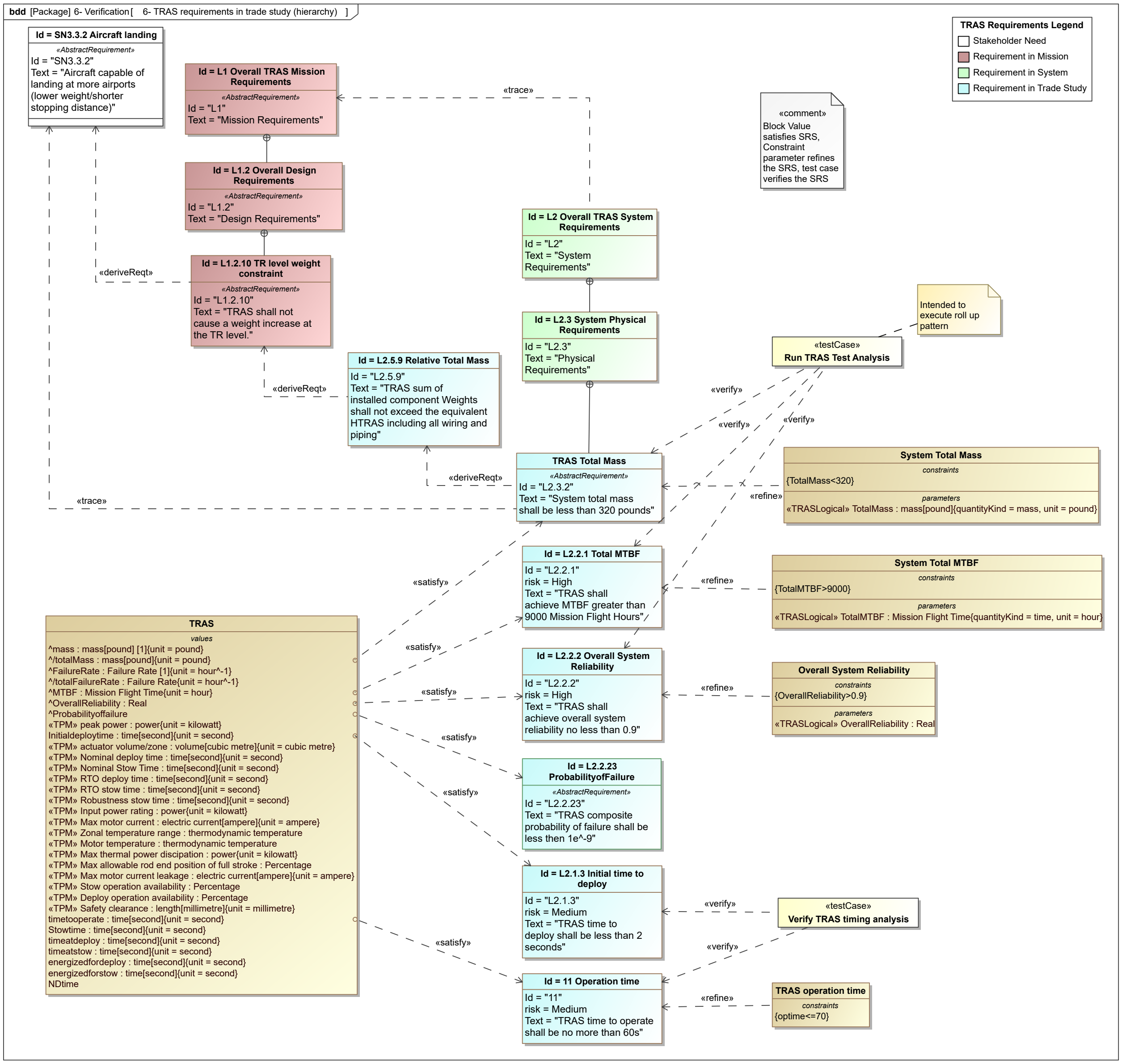

bdd [Package] 6- Verification [6- TRAS requirements in trade study (hierarchy)]
TRAS Requirements Legend
□ Stakeholder Need
🟥 Requirement in Mission
🟩 Requirement in System
🟦 Requirement in Trade Study
Id = SN3.3.2 Aircraft landing
«AbstractRequirement»
Id = "SN3.3.2"
Text = "Aircraft capable of landing at more airports (lower weight/shorter stopping distance)"
Id = L1 Overall TRAS Mission Requirements
«AbstractRequirement»
Id = "L1"
Text = "Mission Requirements"
«trace»
Id = L1.2 Overall Design Requirements
«AbstractRequirement»
Id = "L1.2"
Text = "Design Requirements"
«comment»
Block Value satisfies SRS, Constraint parameter refines the SRS, test case verifies the SRS
Id = L2 Overall TRAS System Requirements
Id = "L2"
Text = "System Requirements"
«deriveReqt»
Id = L1.2.10 TR level weight constraint
«AbstractRequirement»
Id = "L1.2.10"
Text = "TRAS shall not cause a weight increase at the TR level."
Intended to execute roll up pattern
Id = L2.3 System Physical Requirements
Id = "L2.3"
Text = "Physical Requirements"
«testCase»
Run TRAS Test Analysis
Id = L2.5.9 Relative Total Mass
Id = "L2.5.9"
Text = "TRAS sum of installed component Weights shall not exceed the equivalent HTRAS including all wiring and piping"
«deriveReqt»
«verify»
«verify»
«verify»
TRAS Total Mass
«AbstractRequirement»
Id = "L2.3.2"
Text = "System total mass shall be less than 320 pounds"
«deriveReqt»
«trace»
System Total Mass
constraints
{TotalMass<320}
parameters
«TRASLogical» TotalMass : mass[pound]{quantityKind = mass, unit = pound}
«refine»
Id = L2.2.1 Total MTBF
Id = "L2.2.1"
risk = High
Text = "TRAS shall achieve MTBF greater than 9000 Mission Flight Hours"
«satisfy»
«refine»
System Total MTBF
constraints
{TotalMTBF>9000}
parameters
«TRASLogical» TotalMTBF : Mission Flight Time{quantityKind = time, unit = hour}
TRAS
values
^mass : mass[pound] [1]{unit = pound}
^/totalMass : mass[pound]{unit = pound}
^FailureRate : Failure Rate [1]{unit = hour^-1}
^/totalFailureRate : Failure Rate{unit = hour^-1}
^MTBF : Mission Flight Time{unit = hour}
^OverallReliability : Real
^Probabilityoffailure
«TPM» peak power : power{unit = kilowatt}
Initialdeploytime : time[second]{unit = second}
«TPM» actuator volume/zone : volume[cubic metre]{unit = cubic metre}
«TPM» Nominal deploy time : time[second]{unit = second}
«TPM» Nominal Stow Time : time[second]{unit = second}
«TPM» RTO deploy time : time[second]{unit = second}
«TPM» RTO stow time : time[second]{unit = second}
«TPM» Robustness stow time : time[second]{unit = second}
«TPM» Input power rating : power{unit = kilowatt}
«TPM» Max motor current : electric current[ampere]{unit = ampere}
«TPM» Zonal temperature range : thermodynamic temperature
«TPM» Motor temperature : thermodynamic temperature
«TPM» Max thermal power discipation : power{unit = kilowatt}
«TPM» Max allowable rod end position of full stroke : Percentage
«TPM» Max motor current leakage : electric current[ampere]{unit = ampere}
«TPM» Stow operation availability : Percentage
«TPM» Deploy operation availability : Percentage
«TPM» Safety clearance : length[millimetre]{unit = millimetre}
timetooperate : time[second]{unit = second}
Stowtime : time[second]{unit = second}
timeatdeploy : time[second]{unit = second}
timeatstow : time[second]{unit = second}
energizedfordeploy : time[second]{unit = second}
energizedforstow : time[second]{unit = second}
NDtime
«satisfy»
Id = L2.2.2 Overall System Reliability
Id = "L2.2.2"
risk = High
Text = "TRAS shall achieve overall system reliability no less than 0.9"
«satisfy»
«refine»
Overall System Reliability
constraints
{OverallReliability>0.9}
parameters
«TRASLogical» OverallReliability : Real
«satisfy»
Id = L2.2.23 ProbabilityofFailure
«AbstractRequirement»
Id = "L2.2.23"
Text = "TRAS composite probability of failure shall be less then 1e^-9"
«satisfy»
Id = L2.1.3 Initial time to deploy
Id = "L2.1.3"
risk = Medium
Text = "TRAS time to deploy shall be less than 2 seconds"
«verify»
«testCase»
Verify TRAS timing analysis
«verify»
«satisfy»
Id = 11 Operation time
Id = "11"
risk = Medium
Text = "TRAS time to operate shall be no more than 60s"
«refine»
TRAS operation time
constraints
{optime<=70}

Figure 3.45: Establish relationship of model elements with requirements in trade study

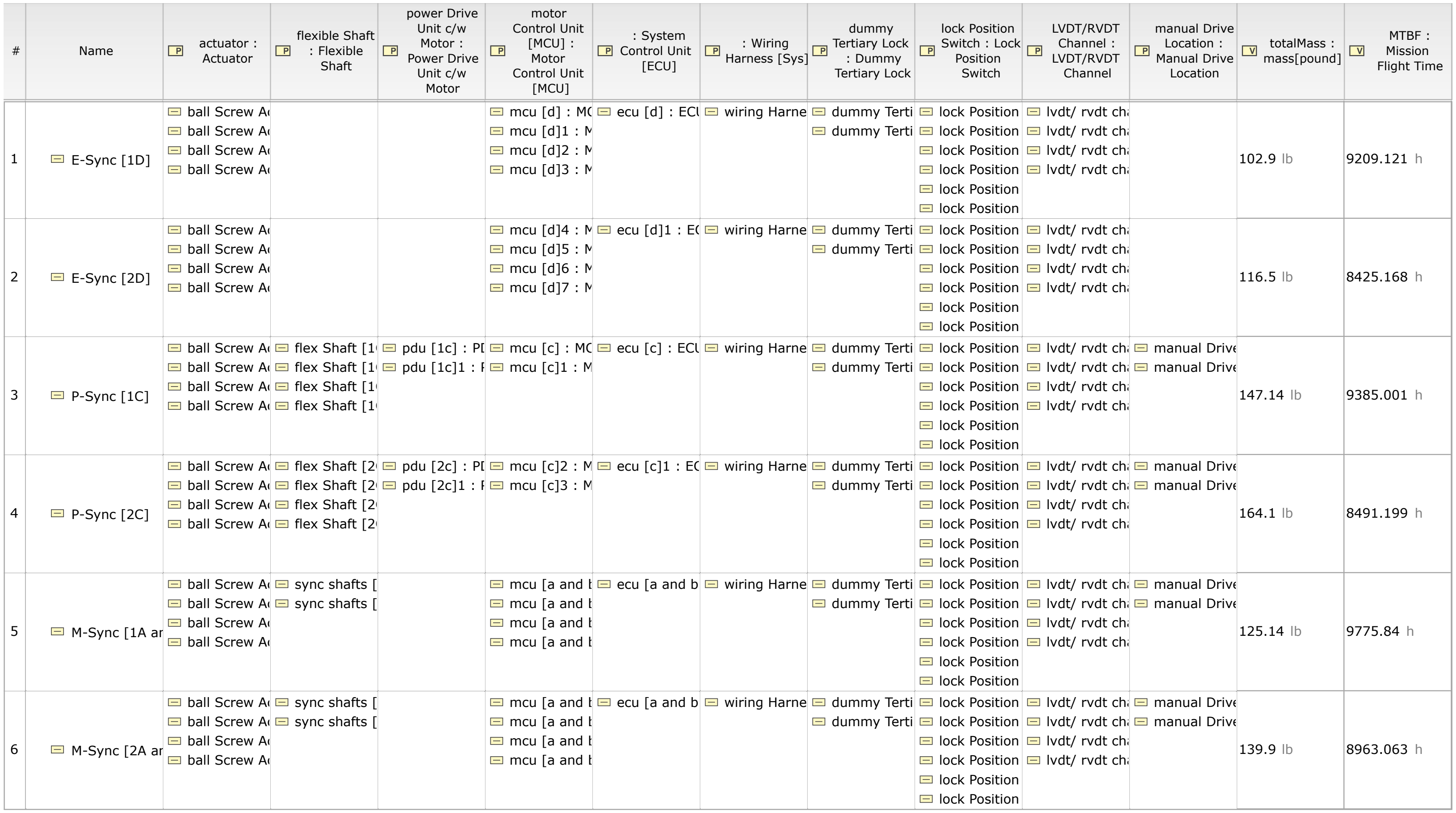
3.3.1 Trace Relationship between Model Elements and Requirements in the Trade Study

Several types of relationships (requirement dependencies) are created with requirements, architecture/design, analysis results, and verification methods for this purpose. As discussed in Secs. 1.1, 1.1.4, and 2.1, the information pertinent to the source of a requirements is captured using the "derive" dependencies between a higher level, 'parent requirement' and a lower level, 'child requirement'. A requirements decomposition hierarchy for the requirement in a trade study is created in a BDD as illustrated in Fig. 3.45 to visualize the upstream and downstream relationships between requirements. For instance, *TRAS Total Mass* requirement in trade study is derived from *Relative Total Mass* requirement in the system, which is obtained from a requirement in mission and from the Stakeholder Needs, and cascaded to requirements in system.

A TRAS Requirements legend indicates the stakeholder need, mission requirement, system requirement, and trade study requirement types and is specified with color codes. The diagram is easy to follow. Further traceability relationships are established for the top five requirements in the trade study, which is a specialised system requirement. These requirements specifications are related to the system performance measures that satisfy them, constraint specifications that refine them, and test case specifications that verify them.

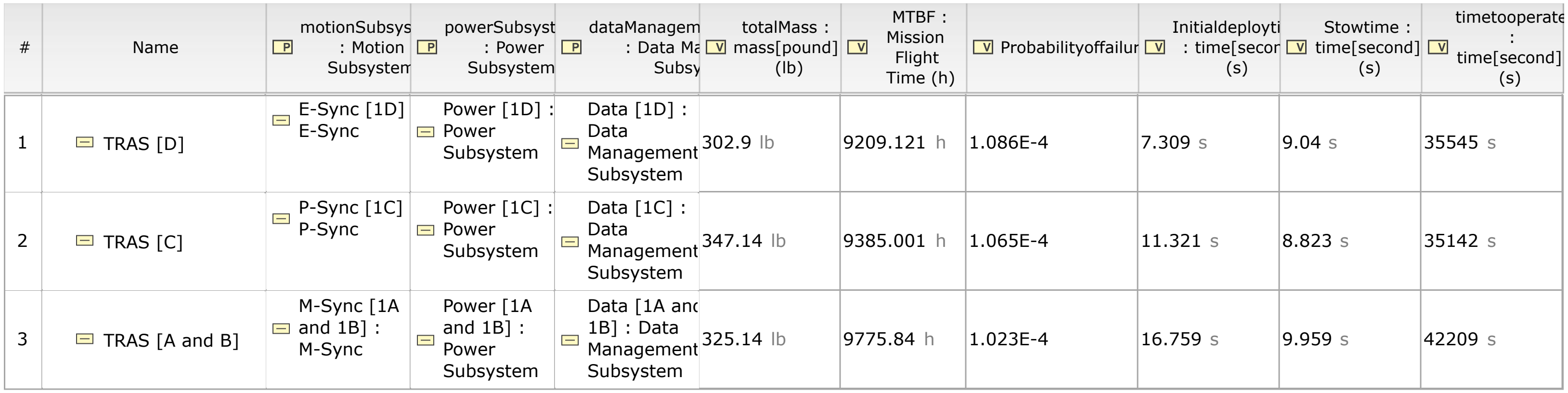
3.3.2 Verify System Requirements with Simulation Results from External Performance Models

In order to verify that the design meets the requirements, a physics-based performance model is developed and utilized in a co-simulation environment. The subsystem design configurations are synthesized by assembling physical components in an analysis tool. The performance model is subjected to specific operational scenarios, and the resulting behavior is plotted as performance curves (e.g., load and speed profiles). Physical components are parameterized by providing input values defined with reasonable assumptions. The behavior is simulated to evaluate the performance measures, and the results are captured through interfaces and exported as Functional Mock-Up



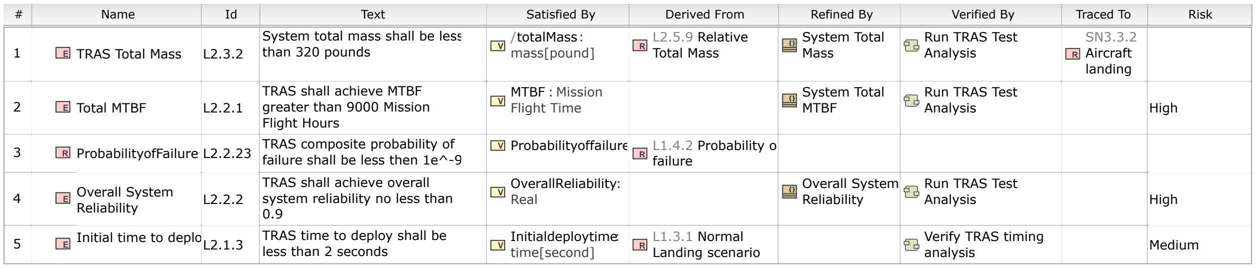
#	Name	P actuator : Actuator	P flexible Shaft : Flexible Shaft	P power Drive Unit c/w Motor : Power Drive Unit c/w Motor	P motor Control Unit [MCU] : Motor Control Unit [MCU]	P : System Control Unit [ECU]	P : Wiring Harness [Sys]	P dummy Tertiary Lock : Dummy Tertiary Lock	P lock Position Switch : Lock Position Switch	P LVDT/RVDT Channel : LVDT/RVDT Channel	P manual Drive Location : Manual Drive Location	V totalMass : mass[pound]	V MTBF : Mission Flight Time
1	E-Sync [1D]	ball Screw Ac ball Screw Ac ball Screw Ac ball Screw Ac			mcu [d] : MC mcu [d]1 : M mcu [d]2 : M mcu [d]3 : M	ecu [d] : ECU	wiring Harne	dummy Terti dummy Terti	lock Position lock Position lock Position lock Position lock Position lock Position	lvdt/ rvdt cha lvdt/ rvdt cha lvdt/ rvdt cha lvdt/ rvdt cha		102.9 lb	9209.121 h
2	E-Sync [2D]	ball Screw Ac ball Screw Ac ball Screw Ac ball Screw Ac			mcu [d]4 : M mcu [d]5 : M mcu [d]6 : M mcu [d]7 : M	ecu [d]1 : EC	wiring Harne	dummy Terti dummy Terti	lock Position lock Position lock Position lock Position lock Position lock Position	lvdt/ rvdt cha lvdt/ rvdt cha lvdt/ rvdt cha lvdt/ rvdt cha		116.5 lb	8425.168 h
3	P-Sync [1C]	ball Screw Ac ball Screw Ac ball Screw Ac ball Screw Ac	flex Shaft [1 flex Shaft [1 flex Shaft [1 flex Shaft [1	pdu [1c] : PD pdu [1c]1 : P	mcu [c] : MC mcu [c]1 : M	ecu [c] : ECU	wiring Harne	dummy Terti dummy Terti	lock Position lock Position lock Position lock Position lock Position lock Position	lvdt/ rvdt cha lvdt/ rvdt cha lvdt/ rvdt cha lvdt/ rvdt cha	manual Drive manual Drive	147.14 lb	9385.001 h
4	P-Sync [2C]	ball Screw Ac ball Screw Ac ball Screw Ac ball Screw Ac	flex Shaft [2 flex Shaft [2 flex Shaft [2 flex Shaft [2	pdu [2c] : PD pdu [2c]1 : P	mcu [c]2 : M mcu [c]3 : M	ecu [c]1 : EC	wiring Harne	dummy Terti dummy Terti	lock Position lock Position lock Position lock Position lock Position lock Position	lvdt/ rvdt cha lvdt/ rvdt cha lvdt/ rvdt cha lvdt/ rvdt cha	manual Drive manual Drive	164.1 lb	8491.199 h
5	M-Sync [1A ar	ball Screw Ac ball Screw Ac ball Screw Ac ball Screw Ac	sync shafts [sync shafts [		mcu [a and b mcu [a and b mcu [a and b mcu [a and b	ecu [a and b	wiring Harne	dummy Terti dummy Terti	lock Position lock Position lock Position lock Position lock Position lock Position	lvdt/ rvdt cha lvdt/ rvdt cha lvdt/ rvdt cha lvdt/ rvdt cha	manual Drive manual Drive	125.14 lb	9775.84 h
6	M-Sync [2A ar	ball Screw Ac ball Screw Ac ball Screw Ac ball Screw Ac	sync shafts [sync shafts [		mcu [a and b mcu [a and b mcu [a and b mcu [a and b	ecu [a and b	wiring Harne	dummy Terti dummy Terti	lock Position lock Position lock Position lock Position lock Position lock Position	lvdt/ rvdt cha lvdt/ rvdt cha lvdt/ rvdt cha lvdt/ rvdt cha	manual Drive manual Drive	139.9 lb	8963.063 h

Figure 3.46: Instances of motion subsystem configurations



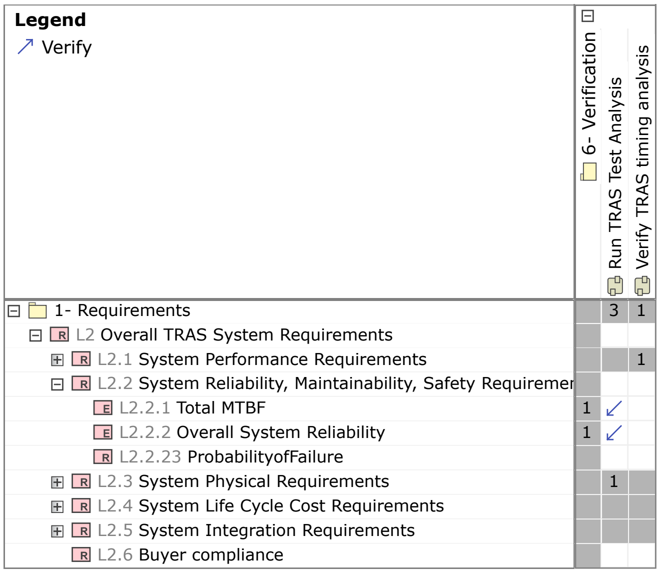
#	Name	motionSubsys : Motion Subsystem	powerSubsyst : Power Subsystem	dataManagem : Data Ma Subsy	totalMass : mass[pound] (lb)	MTBF : Mission Flight Time (h)	Probabilityoffailur	Initialdeployti : time[secor (s)	Stowtime : time[second] (s)	timetooperate : time[second] (s)
1	TRAS [D]	E-Sync [1D] : E-Sync	Power [1D] : Power Subsystem	Data [1D] : Data Management Subsystem	302.9 lb	9209.121 h	1.086E-4	7.309 s	9.04 s	35545 s
2	TRAS [C]	P-Sync [1C] : P-Sync	Power [1C] : Power Subsystem	Data [1C] : Data Management Subsystem	347.14 lb	9385.001 h	1.065E-4	11.321 s	8.823 s	35142 s
3	TRAS [A and B]	M-Sync [1A and 1B] : M-Sync	Power [1A and 1B] : Power Subsystem	Data [1A anc 1B] : Data Management Subsystem	325.14 lb	9775.84 h	1.023E-4	16.759 s	9.959 s	42209 s

Figure 3.47: Instances of TRAS configurations



#	Name	Id	Text	Satisfied By	Derived From	Refined By	Verified By	Traced To	Risk
1	TRAS Total Mass	L2.3.2	System total mass shall be less than 320 pounds	/totalMass : mass[pound]	L2.5.9 Relative Total Mass	System Total Mass	Run TRAS Test Analysis	SN3.3.2 Aircraft landing	
2	Total MTBF	L2.2.1	TRAS shall achieve MTBF greater than 9000 Mission Flight Hours	MTBF : Mission Flight Time		System Total MTBF	Run TRAS Test Analysis		High
3	ProbabilityofFailure	L2.2.23	TRAS composite probability of failure shall be less then 1e^-9	Probabilityoffailure	L1.4.2 Probability o failure				
4	Overall System Reliability	L2.2.2	TRAS shall achieve overall system reliability no less than 0.9	OverallReliability: Real		Overall System Reliability	Run TRAS Test Analysis		High
5	Initial time to deplo	L2.1.3	TRAS time to deploy shall be less than 2 seconds	Initialdeploytime: time[second]	L1.3.1 Normal Landing scenario		Verify TRAS timing analysis		Medium

Figure 3.48: Manage traceability of requirements in trade study



Legend
↗ Verify

	6- Verification	Run TRAS Test Analysis	Verify TRAS timing analysis
1- Requirements		3	1
L2 Overall TRAS System Requirements			
L2.1 System Performance Requirements			1
L2.2 System Reliability, Maintainability, Safety Requiremer			
L2.2.1 Total MTBF	1	↙	
L2.2.2 Overall System Reliability	1	↙	
L2.2.23 ProbabilityofFailure			
L2.3 System Physical Requirements		1	
L2.4 System Life Cycle Cost Requirements			
L2.5 System Integration Requirements			
L2.6 Buyer compliance			

Figure 3.49: System requirements verified by test case depicted in a verify requirements matrix

Units (FMUs). The intended outcome of this activity is requirements verification in the system model by linking the simulation results obtained from external engineering analysis tools.

3.3.3 Manage Requirements in Trade Study

After establishing appropriate relationships between requirements and model elements, the relationships are summarised in an interactive table. This helps to visualize the gaps that need to be filled to manage the requirements well. Figure 3.48 presents such a summary of requirements in a trade study, specifying the source of a requirement, design elements that fulfil the requirement, test cases that verify them, and the priority and degree of risks involved. Hence, this is supporting the requirements management effort.

Other artifacts that help visualize the requirement dependencies are created using the requirements matrix. For instance, as shown in Fig. 3.49, a verify requirements matrix is created to summarize the requirements that are verified by the test cases.

Conclusively, this effort addresses *(I1)* and *(I3)* with demonstrated requirements that are traceable and managed within the architecture model.

Chapter 4

Some Model-Based Improvement in Requirements Development and Management Activities

In this chapter, progress is made in addressing the identified issues and challenges with the development and management of text-based requirements. Some improvements to current methods for requirement development and management discussed in Sec. 3.3 are suggested and tried out with customized requirements to justify the improvement and how this approach can lead to the formation of well-formed requirements is postulated. Some advanced tool usage and techniques are demonstrated with good examples transforming static, text-based requirements to attribute-based, more interactive requirements.

Regarding the basics of requirements engineering discussed in Sec. 1.2, "A well-formed requirement" can be described as [33]:

- A statement of system functionality/capability that is necessary, can be achieved and validated,
- A specification possessed by a system that must be met to solve a problem or achieve an objective, and
- Is qualified by measurable conditions and bounded by constraints.

It is observed that stakeholder requirements captured initially are often crude, incomplete, and require some tweaking for improvement. The requirements are manually refined by identifying and filling the missing gaps for definition, management, and traceability.

The fundamental characteristics of a well-formed requirement are summarized as below [34]:

1. Necessary: Requirements are expensive since resources are allocated to manage each requirement specification. Only requirements that are necessary to be specified should be captured, and any redundant requirements should be eliminated.

2. Abstract: Each requirement should be implementation-independent. A requirement should specify the problem description (what is to be done/achieved?) rather than describing the solution/implementation (how it is to be done/achieved?)
3. Unambiguous: Each requirement should be stated in a way that can be interpreted in only one way.
4. Traceable: Each requirement should have a well-documented source/parent requirement.
5. Validatable/Testable: Each requirement should have the means to prove that the model element (i.e., piece of the system) satisfies the requirements.

An attribute-centric approach of modeling requirements is investigated to support Requirements Analysis for better understanding and managing requirements. An attribute is an inherent property/characteristic that can be analyzed qualitatively or quantitatively by a human or automated means. An attribute can be distinguished in two types: a permanent characteristic existing inherently within an entity; and an assigned characteristic of a product, process, or system. In this approach, requirement properties are customized with more descriptive attributes to establish completeness based on definition, traceability, and management aspects [19]. A requirement statement can be assembled by specifying the necessary attributes typed with model elements pertinent to a requirement [35]. Therefore, the said approach for eliciting a typical functional requirement statement/specification shall look like the following [36]:

The [**Who**] shall do [**What**] constrained by [**How Well**] subject to [**Condition**]

4.1 Develop Well-Structured Requirements

A major problem identified in the requirements development effort is increased cost and risk associated with complex systems development due to the poor definition of requirements during early stages. This effort aims to develop well-defined requirements using structured attributes in the form of model elements/set of measurable quality properties while creating a requirement specification [37]. As shown in Fig. 4.1, a new stereotype for the *OEM Requirement* is established in a BDD by inheriting the basic capabilities and properties of a requirement from the SysML *Require-*

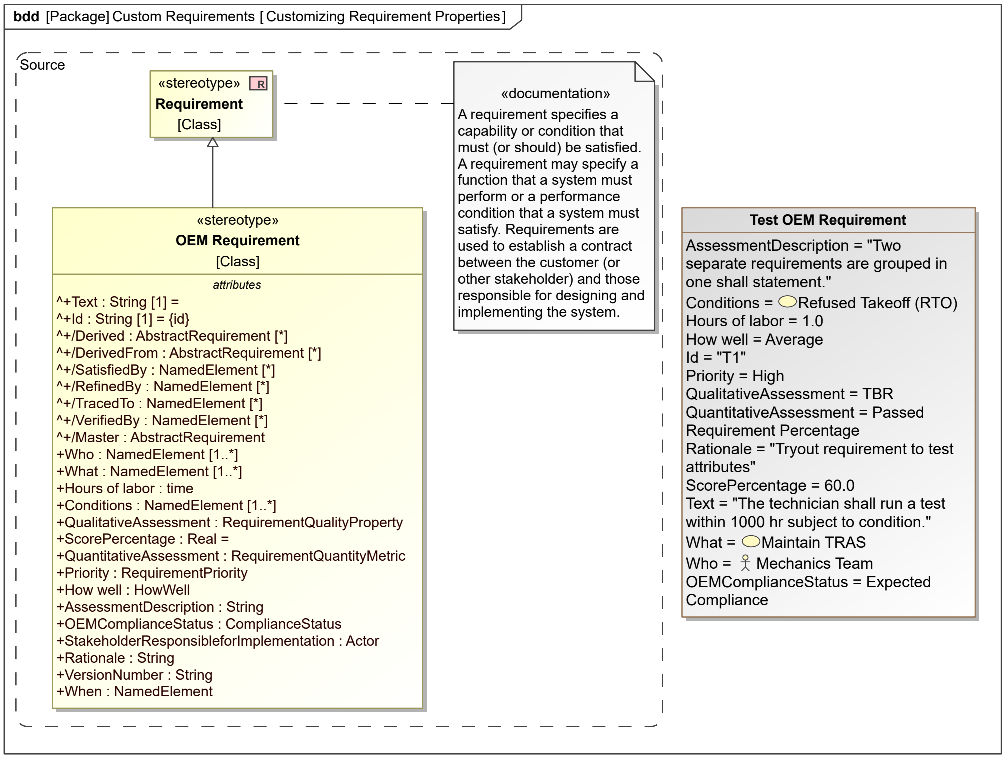

bdd [Package] Custom Requirements [Customizing Requirement Properties]
Source
«stereotype»
Requirement
[Class]
«documentation»
A requirement specifies a capability or condition that must (or should) be satisfied. A requirement may specify a function that a system must perform or a performance condition that a system must satisfy. Requirements are used to establish a contract between the customer (or other stakeholder) and those responsible for designing and implementing the system.
«stereotype»
OEM Requirement
[Class]
attributes
^+Text : String [1] =
^+Id : String [1] = {id}
^+/Derived : AbstractRequirement [*]
^+/DerivedFrom : AbstractRequirement [*]
^+/SatisfiedBy : NamedElement [*]
^+/RefinedBy : NamedElement [*]
^+/TracedTo : NamedElement [*]
^+/VerifiedBy : NamedElement [*]
^+/Master : AbstractRequirement
+Who : NamedElement [1..*]
+What : NamedElement [1..*]
+Hours of labor : time
+Conditions : NamedElement [1..*]
+QualitativeAssessment : RequirementQualityProperty
+ScorePercentage : Real =
+QuantitativeAssessment : RequirementQuantityMetric
+Priority : RequirementPriority
+How well : HowWell
+AssessmentDescription : String
+OEMComplianceStatus : ComplianceStatus
+StakeholderResponsibleforImplementation : Actor
+Rationale : String
+VersionNumber : String
+When : NamedElement
Test OEM Requirement
AssessmentDescription = "Two separate requirements are grouped in one shall statement."
Conditions = Refused Takeoff (RTO)
Hours of labor = 1.0
How well = Average
Id = "T1"
Priority = High
QualitativeAssessment = TBR
QuantitativeAssessment = Passed Requirement Percentage
Rationale = "Tryout requirement to test attributes"
ScorePercentage = 60.0
Text = "The technician shall run a test within 1000 hr subject to condition."
What = Maintain TRAS
Who = Mechanics Team
OEMComplianceStatus = Expected Compliance

Figure 4.1: Customize requirement properties with attributes for definition, traceability, and management

ment stereotype. Thus, the new stereotype obtained is specialized by adding attributes pertinent to definition, traceability, and management aspects that should be specified for completeness of a requirement. The following paragraphs elaborate on the requirement attributes considered in this effort.

4.1.1 Definition Attributes

The attributes that support requirements analysis and help understand the requirements.

1. *Who* (subject): This attribute specifies the subject model element that performs the action or acts upon the capability. Example: Actor/User Role/any subject model element
2. *What* (capability): This attribute describes the expected capability/function in terms of an observable action or outcome. Example: Activity/demonstration/design characteristics
3. *How Well* (constraint): This attribute describes how well the system performs its functions. The attributes could be specified as imposing rules for expected compliance of a measurable

(qualitative/quantitative) characteristic. Constraints restrict a solution/implementation of the system. Example: Performance criteria/qualifying threshold.

4. *Condition* (scenario): This attribute describes the intended operational scenario/condition of use under which the system operates. Example: Operational scenario/state of operation/default condition/subject to duration.
5. *Rationale*: This attribute describes the underlying reason/assumption to specify the intent why a requirement exists. Example: Why is the requirement needed?

4.1.2 Traceability and Management Attributes

The attributes that help uniquely identify, manage and track requirements [19].

1. *Owner/Author*: This attribute describes the stakeholder who creates a requirement and is responsible for managing the requirement.
2. *Version Number*: This attribute describes the serial/version of a requirement.
3. *Qualitative Assessment*: This attribute describes the assessment criteria to measure the quality of a requirement. It could be an enumerated list of quality properties assessed by a human based on a scheme that follows a well-defined set of rules/guidance. Example: Missing/Incomplete/Complete/to be reviewed (TBR).
4. *Quantitative Assessment*: This attribute captures the measure of coverage in terms of overall quantity or percentage of passed/failed requirements.
5. *Score Percentage*: This attribute describes an overall score of how well the requirement is elicited, performed manually.
6. *Assessment Description*: This attribute describes a textual summary of assessment/review stating any improvement remarks.
7. *Priority*: This attribute describes the scheme to prioritize a requirement based on the degree of interest/involvement. Priority is specified later while managing requirements. Example: High/Medium/Low.

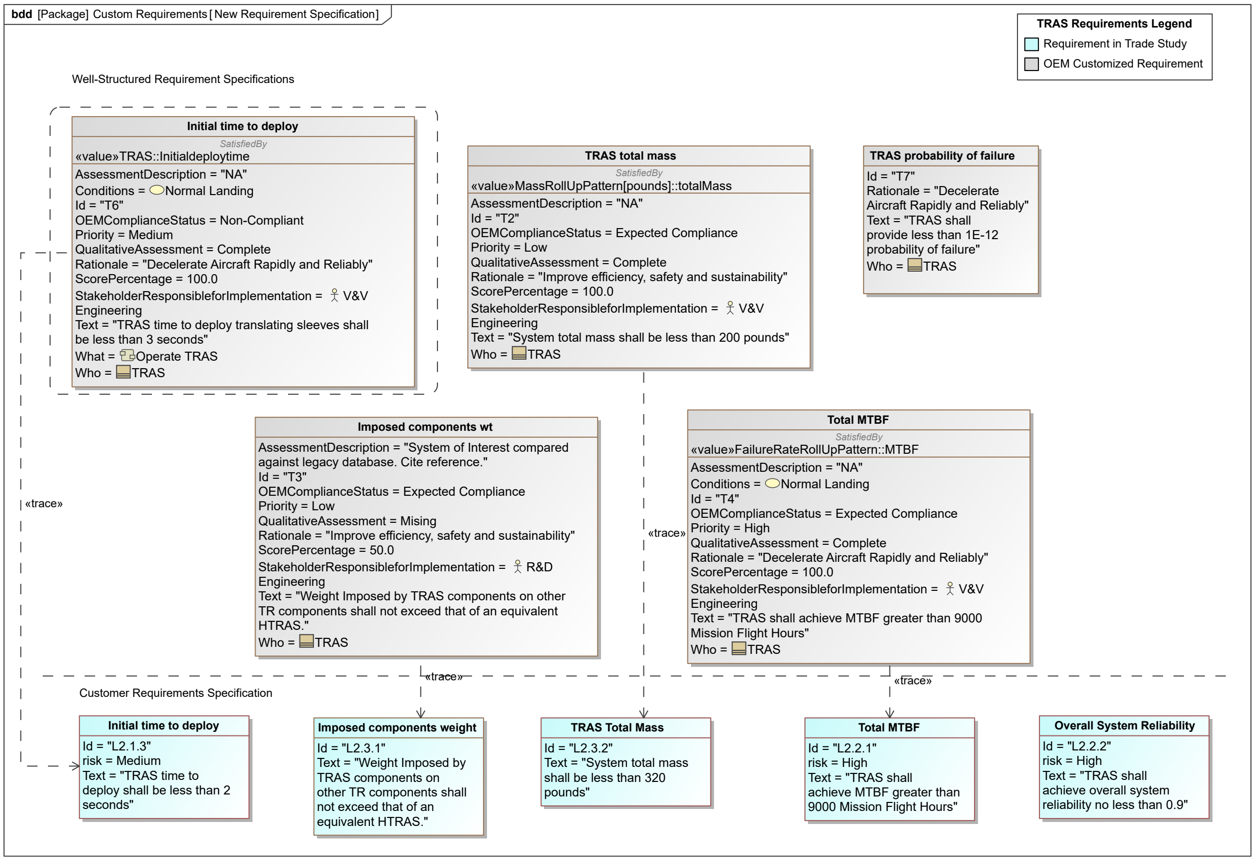
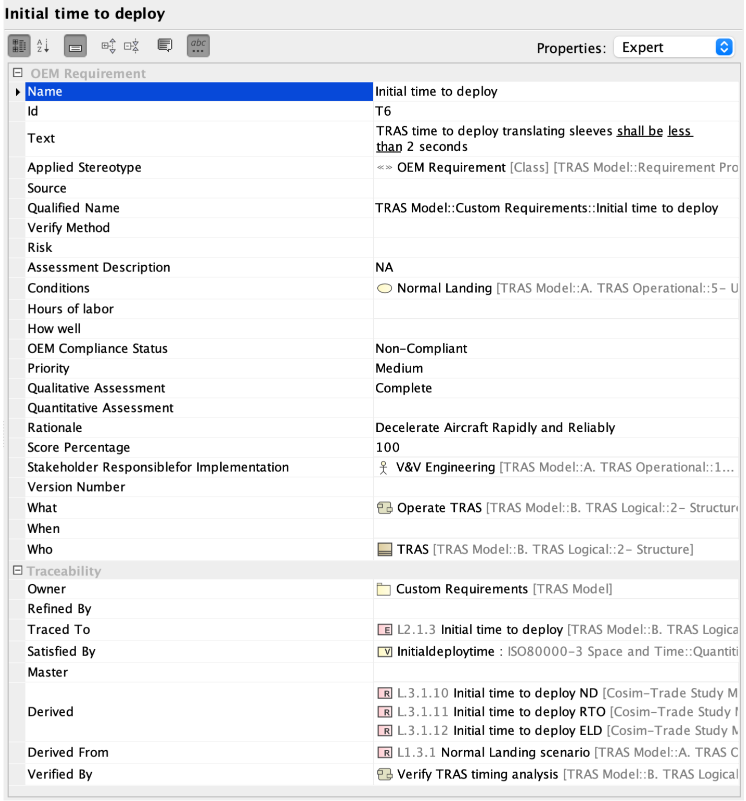

bdd [Package] Custom Requirements [New Requirement Specification]
TRAS Requirements Legend
Requirement in Trade Study
OEM Customized Requirement
Well-Structured Requirement Specifications
Initial time to deploy
SatisfiedBy
«value»TRAS::Initialdeploytime
AssessmentDescription = "NA"
Conditions = Normal Landing
Id = "T6"
OEMComplianceStatus = Non-Compliant
Priority = Medium
QualitativeAssessment = Complete
Rationale = "Decelerate Aircraft Rapidly and Reliably"
ScorePercentage = 100.0
StakeholderResponsibleforImplementation = V&V Engineering
Text = "TRAS time to deploy translating sleeves shall be less than 3 seconds"
What = Operate TRAS
Who = TRAS
TRAS total mass
SatisfiedBy
«value»MassRollUpPattern[pounds]::totalMass
AssessmentDescription = "NA"
Id = "T2"
OEMComplianceStatus = Expected Compliance
Priority = Low
QualitativeAssessment = Complete
Rationale = "Improve efficiency, safety and sustainability"
ScorePercentage = 100.0
StakeholderResponsibleforImplementation = V&V Engineering
Text = "System total mass shall be less than 200 pounds"
Who = TRAS
TRAS probability of failure
Id = "T7"
Rationale = "Decelerate Aircraft Rapidly and Reliably"
Text = "TRAS shall provide less than 1E-12 probability of failure"
Who = TRAS
Imposed components wt
AssessmentDescription = "System of Interest compared against legacy database. Cite reference."
Id = "T3"
OEMComplianceStatus = Expected Compliance
Priority = Low
QualitativeAssessment = Mising
Rationale = "Improve efficiency, safety and sustainability"
ScorePercentage = 50.0
StakeholderResponsibleforImplementation = R&D Engineering
Text = "Weight Imposed by TRAS components on other TR components shall not exceed that of an equivalent HTRAS."
Who = TRAS
Total MTBF
SatisfiedBy
«value»FailureRateRollUpPattern::MTBF
AssessmentDescription = "NA"
Conditions = Normal Landing
Id = "T4"
OEMComplianceStatus = Expected Compliance
Priority = High
QualitativeAssessment = Complete
Rationale = "Decelerate Aircraft Rapidly and Reliably"
ScorePercentage = 100.0
StakeholderResponsibleforImplementation = V&V Engineering
Text = "TRAS shall achieve MTBF greater than 9000 Mission Flight Hours"
Who = TRAS
«trace»
«trace»
«trace»
«trace»
Customer Requirements Specification
Initial time to deploy
Id = "L2.1.3"
risk = Medium
Text = "TRAS time to deploy shall be less than 2 seconds"
Imposed components weight
Id = "L2.3.1"
Text = "Weight Imposed by TRAS components on other TR components shall not exceed that of an equivalent HTRAS."
TRAS Total Mass
Id = "L2.3.2"
Text = "System total mass shall be less than 320 pounds"
Total MTBF
Id = "L2.2.1"
risk = High
Text = "TRAS shall achieve MTBF greater than 9000 Mission Flight Hours"
Overall System Reliability
Id = "L2.2.2"
risk = High
Text = "TRAS shall achieve overall system reliability no less than 0.9"

Figure 4.2: Create customized requirement specifications

8. *OEM Compliance Status*: This attribute describes the status of a requirement's compliance to rules/regulations to track the compliance and maintain traceability between the requirements and respective verification methods. Example: Compliant/Expected Compliance/Non-compliant/TBR.
9. *Responsible Stakeholder*: This attribute specifies the team member responsible for the implementation of a particular requirement.

4.1.3 Create New Requirements and Specify Customized Attributes

As depicted in Fig. 4.2, new requirements are created in a BDD and stereotyped with *OEM Requirement*. Each new requirement traces back to the corresponding initial text-based requirement specification that is to be refined. The trace dependency implies that the source/reference requirement is the basis for defining a new requirement.



Initial time to deploy

Properties: Expert

⊟ OEM Requirement	
Name	Initial time to deploy
Id	T6
Text	TRAS time to deploy translating sleeves shall be less than 2 seconds
Applied Stereotype	«» OEM Requirement [Class] [TRAS Model::Requirement Pro
Source	
Qualified Name	TRAS Model::Custom Requirements::Initial time to deploy
Verify Method	
Risk	
Assessment Description	NA
Conditions	Normal Landing [TRAS Model::A. TRAS Operational::5– U
Hours of labor	
How well	
OEM Compliance Status	Non–Compliant
Priority	Medium
Qualitative Assessment	Complete
Quantitative Assessment	
Rationale	Decelerate Aircraft Rapidly and Reliably
Score Percentage	100
Stakeholder Responsiblefor Implementation	V&V Engineering [TRAS Model::A. TRAS Operational::1...
Version Number	
What	Operate TRAS [TRAS Model::B. TRAS Logical::2– Structur
When	
Who	TRAS [TRAS Model::B. TRAS Logical::2– Structure]
⊟ Traceability	
Owner	Custom Requirements [TRAS Model]
Refined By	
Traced To	E L2.1.3 Initial time to deploy [TRAS Model::B. TRAS Logica
Satisfied By	V Initialdeploytime : ISO80000–3 Space and Time::Quantiti
Master	
Derived	R L.3.1.10 Initial time to deploy ND [Cosim–Trade Study M R L.3.1.11 Initial time to deploy RTO [Cosim–Trade Study I R L.3.1.12 Initial time to deploy ELD [Cosim–Trade Study M
Derived From	R L1.3.1 Normal Landing scenario [TRAS Model::A. TRAS C
Verified By	Verify TRAS timing analysis [TRAS Model::B. TRAS Logical

Figure 4.3: Example specification for a customized requirement *Initial time to deploy*

Definition Attributes:

Who = TRAS

What = Operate TRAS

How Well = < 2 seconds

Condition = Normal Landing Scenario

Traceability and Management Attributes:

Parent Requirement/Derived From = Requirement in Mission

Satisfied By = System performance measure for deploy time

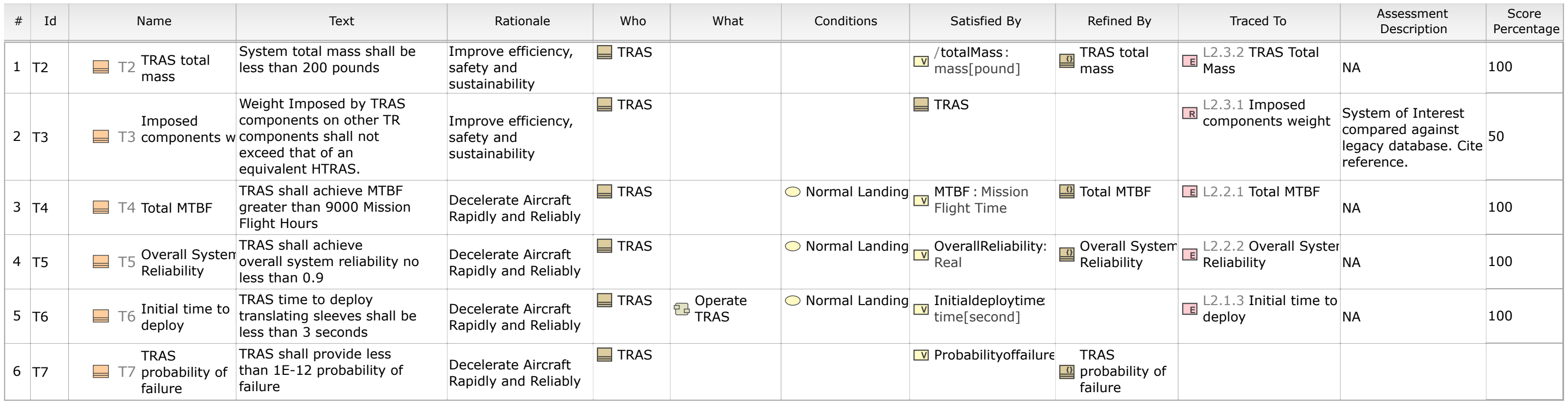
Verified By = Timing analysis test case

Traced To= Requirement in System

Compliance Status= Non-Compliant

Risk= Medium

Priority= High



#	Id	Name	Text	Rationale	Who	What	Conditions	Satisfied By	Refined By	Traced To	Assessment Description	Score Percentage
1	T2	T2 TRAS total mass	System total mass shall be less than 200 pounds	Improve efficiency, safety and sustainability	TRAS			/totalMass : mass[pound]	TRAS total mass	L2.3.2 TRAS Total Mass	NA	100
2	T3	T3 Imposed components w	Weight Imposed by TRAS components on other TR components shall not exceed that of an equivalent HTRAS.	Improve efficiency, safety and sustainability	TRAS			TRAS		L2.3.1 Imposed components weight	System of Interest compared against legacy database. Cite reference.	50
3	T4	T4 Total MTBF	TRAS shall achieve MTBF greater than 9000 Mission Flight Hours	Decelerate Aircraft Rapidly and Reliably	TRAS		Normal Landing	MTBF : Mission Flight Time	Total MTBF	L2.2.1 Total MTBF	NA	100
4	T5	T5 Overall System Reliability	TRAS shall achieve overall system reliability no less than 0.9	Decelerate Aircraft Rapidly and Reliably	TRAS		Normal Landing	OverallReliability: Real	Overall System Reliability	L2.2.2 Overall Syster Reliability	NA	100
5	T6	T6 Initial time to deploy	TRAS time to deploy translating sleeves shall be less than 3 seconds	Decelerate Aircraft Rapidly and Reliably	TRAS	Operate TRAS	Normal Landing	Initialdeploytime time[second]		L2.1.3 Initial time to deploy	NA	100
6	T7	T7 TRAS probability of failure	TRAS shall provide less than 1E-12 probability of failure	Decelerate Aircraft Rapidly and Reliably	TRAS			Probabilityoffailure	TRAS probability of failure			

Figure 4.4: Well-structured requirements with customized attributes

Figure 4.3 depicts an example of a well-defined, structured requirement specification created in the tool environment. The same approach is practiced to develop other requirement specifications corresponding to the initial requirements in the trade study. The customized attributes of definition, traceability, and management are specified for each requirement and represented in a requirements table as shown in Fig. 4.4. The well-structured requirements specification thus obtained are interactive.

Following the proposed model-based approach for requirements elicitation, it is evident that specifying attributes tailored to the requirement development and management effort enable requirements to relate with model elements and other requirements within a model. This practice further ensures that the requirement specifications are complete and unambiguous, consistent, and not duplicated. Also, requirements elicitation following this approach demands specific selection from a predefined list of model elements and could be used to restrict the selection to particular model elements only. In contrast to the static, text-based requirements, the attribute-centric requirements discussed in this chapter are more interactive in nature. Therefore, the task of analyzing requirements is now simplified, and it is easier to address questions like: *Is the requirement even necessary? What model elements are specified for the requirement tied to attributes, and where else in the model are these model elements related? What are the missing gaps in a requirement specification? If there is a change in the requirement specification, what model elements are impacted and further propagation?*

These well-structured requirements can also be reviewed/assessed by customized matrices to visualize a query of information related to the requirements. For instance, visualizing a series of requirements by categorizing the information content, such as:

- Using *ID/Tag* to show all the requirements identified in trade study;
- Using a scale or scheme to gauge the *priority* of say, all requirements in a trade study;
- Show all requirements in system are either Vital/Essential/Desirable depending on the *criticality*;
- Show all the requirement in system that are *feasible and agreed upon*;

- Assessing requirements based on consequences or degree of risk avoidance—show all the requirements in a trade study that are indicated with *High Risk*;
- Show the *Source* (parent requirement) of all requirement in trade study that should be consulted;
- Depending on the Requirement *Type*, show all the performance, functional, non-functional, regulatory, etc. requirements in the system.

In conclusion, a model-based approach to develop well-structured requirements has enhanced the robustness and improved the overall requirement development and management effort. Future efforts shall contribute to establishing a catalog of stereotypes for specifying requirement attributes that can be applied to a series of requirements. Specific attributes can be added simply by applying the relevant stereotype(s) to requirements. For instance, a *Performance Requirement* could be specified with definition, traceability, and management attributes simply by applying the relevant stereotypes.

4.2 Analyze Interactively Requirement Specification Changes

A significant challenge involved in the SE effort is to systematically identifying, perceiving, and analyzing changes proposed for implementation and manage changes that emerge as the system evolves through the life cycle. Some typical sources of changes that can be introduced in a system development project and strategies to address such changes are now discussed. Although changes are necessary for the system development process, it is undoubtedly disruptive and problematic in complex system design projects. Primarily, changes that emerge later in the life cycle can significantly impact the budget and schedule of the project, leading to undesirable ramifications. Managing a change to accommodate development projects within the budget and schedule constraints is challenging. A change, once identified, can be exposed to multiple disciplines and stakeholders collaborating on a project. Thus, providing visibility into assessing change adds value in understanding the impact and efficiently managing those changes.

Current approaches reflect manual transfer of data to manage requirements with a siloed approach that might be uncoordinated and can remain disconnected, leading to a gap between requirements and planning activities. In contrast to text-based requirements, the interactive model-based requirements discussed in this section demonstrates the ability to trace the change in model elements through upstream and downstream relationships, including higher-order relationships. This approach enables a collective understanding of system aspects that have been affected, thus improving communication and reducing the time and effort taken to report the status of change impact.

Some possible sources of a change are identified as design errors, results obtained from trials or tryouts and testing, reviews/certifications, programmatic change requests, and updates/revisions. Some strategies to analyze and trace the impact of such changes can be summarized as below:

- Comparing model versions to visualize change, i.e., old vs. new;
- Reviewing upstream and downstream relationships to analyze impact on model elements;
- Utilizing MBSE tool capabilities such as suspect links.

4.2.1 Review the Requirements Attributes to Analyze and Capture Change

A major contribution of the attribute-based requirements to the SE effort is productivity, measured in terms of savings in hours of labor to perform a requirement change impact assessment. For instance, consider a change introduced to the customized requirement, *T6, Initial time to deploy* from *TRAS time to deploy shall be less than 2 seconds* to *TRAS time to deploy shall be less than 3 seconds*. The model elements linked to attributes of the requirement *T6, Initial time to deploy* are reviewed from Fig. 4.4. The *Constraint* attribute for the deploy time is modified from < *2 seconds* to < *3 seconds* to match with the requirement specification. The *What* attribute specified by the capability, *Operate TRAS* is reviewed for the overall TRAS time to deploy.

Conclusively, a change in requirement specification can be analyzed and accommodated by modifying the relevant attributes of the requirement. Any missing gap/redundancy is identified to improve the requirement specification further. This also helps to identify and validate whether a requirement is necessary or redundant. Note that requirements developed with a model-based approach are more interactive and intuitive than static, text-based requirements. The assessment of propagation and impact of change on other requirements and model elements is elaborated in the next section.

4.2.2 Navigate the Suspicious Propagation of Change and Analyze the Impact on Related Model Elements

As a result of modifying the attributes typed by model elements within a requirement specification to accommodate a change, both upstream and downstream model elements related to the modified requirement and might have been impacted are analyzed. The impact on additional model elements related to the modification is assessed by reviewing and analyzing the suspicions marked on propagated elements that might have changed. The suspicions are raised by employing a tool capability, called *Suspect Links*, where any model elements related to the missing/modified/deprecated model elements are identified to address a suspicion. As illustrated in Fig. 4.5, after introducing a change to the requirement specification *T6, Initial time to deploy*, as discussed in Sec. 4.2.1, the suspicions of change are automatically highlighted by the tool (yellowish color). Following upstream and downstream suspicions are reviewed, and required adjustments are made:

- System specification for a performance measure, *Initialdeploytime* that satisfies the requirement.
- Stakeholder requirement/ customer specification *L2.1.3* that provides a reference/source of information to specify the requirement;
- The parent requirement in mission *L1.3.1* from where it is derived;
- The lower level, child requirements *L3.1.10, L3.1.11 and L3.1.12* that are derived from this requirement;
- The timing constraint *LimitingDeployTime* that refines the requirement;
- The test case *Verify TRAS timing analysis* that verifies the requirement.

The current scope of change impact analysis is limited to identifying related requirements and model elements. However, as the modeling effort progresses and more relations are established, a comprehensive assessment can be performed. Each time a change is introduced, efforts are made to ensure that necessary placeholders are accommodated for effective and efficient management of requirements.

In conclusion, the existing text-based approach to develop and manage requirements is improved by attribute-based techniques that allows to perform quality more interactively and ensures that the gaps are filled for the completeness of requirements. Thus, the effort clearly addresses (*I1*), (*I4*), and (*I5*).

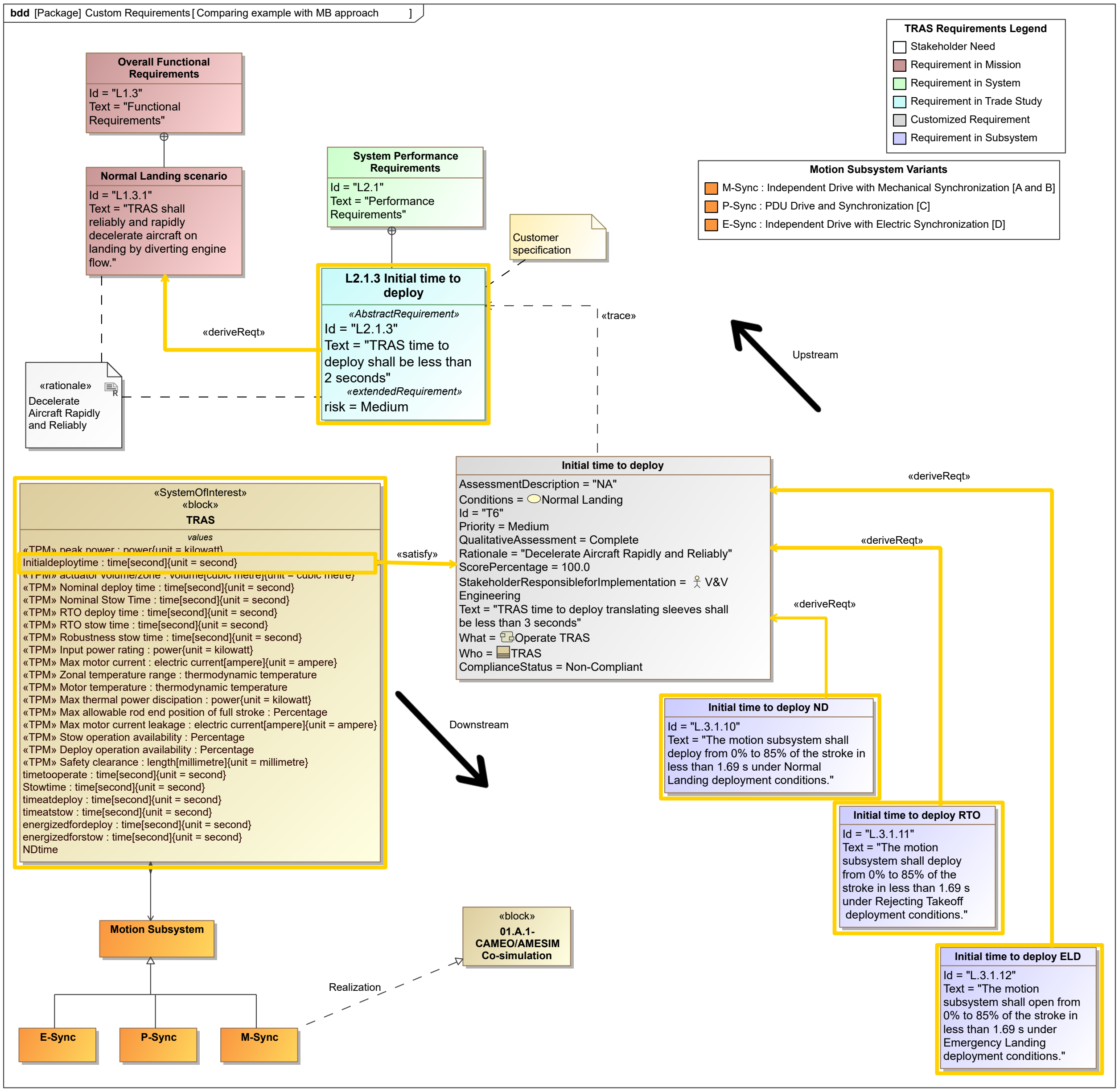

bdd [Package] Custom Requirements [Comparing example with MB approach]
TRAS Requirements Legend
Stakeholder Need
Requirement in Mission
Requirement in System
Requirement in Trade Study
Customized Requirement
Requirement in Subsystem
Motion Subsystem Variants
M-Sync : Independent Drive with Mechanical Synchronization [A and B]
P-Sync : PDU Drive and Synchronization [C]
E-Sync : Independent Drive with Electric Synchronization [D]
Overall Functional Requirements
Id = "L1.3"
Text = "Functional Requirements"
Normal Landing scenario
Id = "L1.3.1"
Text = "TRAS shall reliably and rapidly decelerate aircraft on landing by diverting engine flow."
System Performance Requirements
Id = "L2.1"
Text = "Performance Requirements"
Customer specification
L2.1.3 Initial time to deploy
«AbstractRequirement»
Id = "L2.1.3"
Text = "TRAS time to deploy shall be less than 2 seconds"
«extendedRequirement»
risk = Medium
«deriveReqt»
«trace»
Upstream
«rationale»
Decelerate Aircraft Rapidly and Reliably
Initial time to deploy
AssessmentDescription = "NA"
Conditions = Normal Landing
Id = "T6"
Priority = Medium
QualitativeAssessment = Complete
Rationale = "Decelerate Aircraft Rapidly and Reliably"
ScorePercentage = 100.0
StakeholderResponsibleforImplementation = V&V Engineering
Text = "TRAS time to deploy translating sleeves shall be less than 3 seconds"
What = Operate TRAS
Who = TRAS
ComplianceStatus = Non-Compliant
«deriveReqt»
«deriveReqt»
«deriveReqt»
«SystemOfInterest»
«block»
TRAS
values
«TPM» peak power : power{unit = kilowatt}
Initialdeploytime : time[second]{unit = second}
«TPM» actuator volume/zone : volume[cubic metre]{unit = cubic metre}
«TPM» Nominal deploy time : time[second]{unit = second}
«TPM» Nominal Stow Time : time[second]{unit = second}
«TPM» RTO deploy time : time[second]{unit = second}
«TPM» RTO stow time : time[second]{unit = second}
«TPM» Robustness stow time : time[second]{unit = second}
«TPM» Input power rating : power{unit = kilowatt}
«TPM» Max motor current : electric current[ampere]{unit = ampere}
«TPM» Zonal temperature range : thermodynamic temperature
«TPM» Motor temperature : thermodynamic temperature
«TPM» Max thermal power discipation : power{unit = kilowatt}
«TPM» Max allowable rod end position of full stroke : Percentage
«TPM» Max motor current leakage : electric current[ampere]{unit = ampere}
«TPM» Stow operation availability : Percentage
«TPM» Deploy operation availability : Percentage
«TPM» Safety clearance : length[millimetre]{unit = millimetre}
timetooperate : time[second]{unit = second}
Stowtime : time[second]{unit = second}
timeatdeploy : time[second]{unit = second}
timeatstow : time[second]{unit = second}
energizedfordeploy : time[second]{unit = second}
energizedforstow : time[second]{unit = second}
NDtime
«satisfy»
Downstream
Initial time to deploy ND
Id = "L.3.1.10"
Text = "The motion subsystem shall deploy from 0% to 85% of the stroke in less than 1.69 s under Normal Landing deployment conditions."
Initial time to deploy RTO
Id = "L.3.1.11"
Text = "The motion subsystem shall deploy from 0% to 85% of the stroke in less than 1.69 s under Rejecting Takeoff deployment conditions."
Initial time to deploy ELD
Id = "L.3.1.12"
Text = "The motion subsystem shall open from 0% to 85% of the stroke in less than 1.69 s under Emergency Landing deployment conditions."
Motion Subsystem
«block»
01.A.1-CAMEO/AMESIM Co-simulation
Realization
E-Sync
P-Sync
M-Sync

Figure 4.5: Reviewing upstream and downstream requirements for change impact assessment

Chapter 5

Conclusions and Future Work

5.1 Conclusions

While much of the literature on model-based solutions address the challenges of a broader SE effort from the perspective of generic users and tool developers, fewer point to the challenges in the system development effort within technology/platform projects faced by organizations. The modeling and research effort demonstrated in this thesis investigated specific challenges of the system development effort for platform development projects within a SE group led by an industry partner. A system architecture modeling framework and methodology were augmented to create an abstract representation of a Thrust Reverser Actuation System (TRAS). The TRAS model was developed by specifying the system aspects and transforming information content from abstract to detailed within the operational, logical, and physical levels of abstraction.

It is critical to understand that a model-centric approach adds value to support SE efforts. This thesis demonstrates an approach to implement MBSE for supporting the system development effort in technology/platform projects. Throughout this work, numerous clear and good examples of MBSE artifacts were shown. The feasibility evaluation process was translated into a process model to elaborate the lower-level activities and perceive the deliverable/tangible work products needed to progress. This effort led to an improved understanding of the interfaces between processes and how they might affect the work products.

The proposed framework filled the gap for channeling data between an organization's PLC process and a unified system model through data modeling resulting in the coordination of the PLC phase process deliverables/outcomes with model-derived artifacts. The tailored architecture development framework also exposed the aspects of system verification and validation (V&V). The introduction of the V&V column pointed to the technical evaluation and planning activities necessary to obtain agreement among stakeholders. The TRAS model was synthesized to analyze architecture alternatives. Cross-cutting relationships were established to capture traceability. Requirements and their dependencies with model elements were analyzed to assess traceability. Deficiencies in current methods of requirement development and management were identified, and an attribute-based approach was augmented to transform poorly written text-based requirements to well-structured requirements that could be assessed more interactively.

Overall, this thesis demonstrates how model-based implementation can be targeted to identify capabilities and develop them to address a span of challenges and issues faced in today's complex system development. MBSE promises a cultural transformation with a philosophical shift from document-centric effort to model-centric SE effort. It is evident that the model-centric approach supports identifying risks and errors early in the life cycle, even before the implementation, as compared to the classical SE document-centric approach that observes verification after resources are committed, and a solution/decision is implemented, hence, saving huge on resource constraints.

The noticeable benefits a central repository realized in the modeling and research effort points to some qualities such as:

- Consistency in the language to communicate system aspects;
- Implementation is more scalable starting from a proof of concept that can be evolved to a full-scale model;
- Models are fairly maintainable, version control can be managed with more convenience;

However, to note, the quantitative improvements are difficult to measure and perceived from assessments/audits performed over a period of time after an implementation.

5.2 Future Work

Although the modeling and research effort documented in this thesis demonstrates some clear benefits and is aligned with the value addition to the overall SE effort, it only partially demonstrates the complete implementation. Future efforts, summarized below, will attempt to overcome the implementation deficiencies and accomplish the true model-based paradigm for modern system development.

1. Further, modeling the phase, gate-based activities, mapping the deliverable within the PLC Phase 200, 300, 400 to system model and artifacts, and automating the process model for gate reviews.

2. More comprehensive and automated model-based requirements to help guide requirements development as well as identify problems with requirements (e.g., quality and completeness attributes). This might include the creation of custom rules to assess/test the quality of shall statements. It may also include generating natural language requirement statements automatically from the model attributes.

3. Full integration of simulation models with architecture model for engineering analysis and requirements verification. This would include detailed interface modeling and its connectivity, safety and reliability analysis using RAAML extension to SysML [38], and integrating external testbed models to verify system requirements using test scenarios.

4. Model-based verification and validation for virtual prototyping (fifth column): Future collaboration shall introduce verification planning capabilities with the development of custom stereotypes. Each stereotype will provide information content that characterizes the structure of a verification plan and elaborates the planning activity involved with the different verification methods used to verify requirements.

5. Additionally, a survey could be conducted to learn more from the current practitioners and industry partners by requesting feedback on some structured questions. This might include statements and opinions regarding their role in the effort, the teams involved, different areas of application, their intentions to achieve which perceived value points (specifically pointing to some quantifiable metrics such as the labor hours that are reduced), and metrics on how the overall effort is benefiting the organization.

Bibliography

[1] J. T. Karam, *Managing Systems Development 101: A Guide to Designing Effective Commercial Products & Systems for Engineers & Their Bosses/CEOs*. ASME, Jan. 2007.

[2] E. R. Carroll and R. J. Malins, "Systematic literature review: How is model-based systems engineering justified?" no. SAND2016-2607, Mar. 2016. [Online]. Available: https://www.osti.gov/biblio/1561164

[3] B. S. Blanchard and W. J. Fabrycky, *Systems Engineering and Analysis*. Boston: Prentice Hall, 2011.

[4] "ISO/IEC/IEEE international standard – systems and software engineering – system life cycle processes," *ISO/IEC/IEEE 15288 First edition 2015-05-15*, pp. 1–118, 2015.

[5] *INCOSE Systems Engineering Handbook*, 2000, vol. 2.0.

[6] A. Kossiakoff and W. N. Sweet, *Systems Engineering Principles and Practice*. John Wiley & Sons, Inc., Nov. 2002.

[7] A. Aleksandraviciene and A. Morkevicius, *MagicGrid Book of Knowledge*.

[8] D. M. Buede and W. D. Miller, *The Engineering Design of Systems: Models and Methods, 3rd Edition*. John Wiley and Sons, Feb. 2016.

[9] J. M. Borky and T. H. Bradley, *Effective Model-Based Systems Engineering*. Springer International Publishing, 2019.

[10] T. Weilkiens, *SYSMOD - The Systems Modeling Toolbox: Pragmatic MBSE with SysML*. MBSE4U, 2020. [Online]. Available: https://books.google.com/books?id=-j6_zQEACAAJ

[11] L. Wheatcraft, M. Ryan, and C. Svensson, "Integrated data as the foundation of systems engineering," in *INCOSE International Symposium*, vol. 27, Jul. 2017, pp. 1423–1437.

[12] "Systems engineering vision 2020," *INCOSE*, vol. 26, p. 2019, 2007.

[13] R. Karban, L. Andolfato, P. Bristow, G. Chiozzi, M. Esselborn, M. Schilling, C. Schmid, H. Sommer, and M. Zamparelli, "Model based systems engineering for astronomical projects," vol. 9150, Aug. 2014, p. 91500L.

[14] D. Harvey, P. Logan, M. Waite, and T. Liddy, "Document the model, don't model the document," 2012.

[15] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML.* Elsevier, 2015.

[16] "Cameo systems modeler." [Online]. Available: https://www.3ds.com/products-services/catia/products/no-magic/cameo-systems-modeler/

[17] *OMG Systems Modeling Language*, Object Management Group Std., 2019. [Online]. Available: https://www.omg.org/spec/SysML/1.6/About-SysML/

[18] L. Delligatti, *SysML Distilled: A Brief Guide to the Systems Modeling Language*, ser. YBP Print DDA. Upper Saddle River, NJ: Addison-Wesley, 2014.

[19] "ISO/IEC/IEEE international standard - systems and software engineering – life cycle processes – requirements engineering," *ISO/IEC/IEEE 29148:2018(E)*, pp. 1–104, 2018.

[20] K. E. Wiegers and J. Beatty, *Software Requirements.* USA: Microsoft Press, 2013.

[21] K. Pohl, *Requirements Engineering: Fundamentals, Principles, and Techniques*, 1st ed. Springer, 2010.

[22] C. Hood, S. Wiedemann, S. Fichtinger, and U. Pautz, *Requirements Management: The Interface Between Requirements Development and All Other Systems Engineering Processes.* Berlin, Heidelberg: Springer Berlin / Heidelberg, 2007.

[23] J. A. Yetter, "Why do airlines want and use thrust reversers? a compilation of airline industry responses to a survey regarding the use of thrust reversers on commercial transport airplanes," Jan. 1995.

[24] C. A. Scott and A. Jeffrey, “Static performance of six innovative thrust reverser concepts for subsonic transport applications: Summary of the NASA Langley innovative thrust reverser test program,” Jul. 2000.

[25] “TRAS system,” Woodward, Inc. [Online]. Available: https://www.woodward.com/en/applications/aircraft-controls/tras-system

[26] M. Aurélio, L. Porto, J. E. B. D. Santos, and A. C. Batista, “Overview on thrust reverser design,” in *International Congress of Mechanical Engineering*, Nov. 2005.

[27] J.-C. Maré, *Aerospace Actuators 3: European Commercial Aircraft and Tiltrotor Aircraft*. John Wiley & Sons, Inc., Jan. 2018.

[28] R. K. Tsui, J. M. Borky, and T. H. Bradley, “Applying model-based systems architecture processes (MBSAP) methodology for diversified MBSE projects with efficient systems of systems accomplishments,” in *INCOSE International Symposium*, vol. 30, no. 1, Jul. 2020, pp. 1568–1580.

[29] A. Gerber, P. le Roux, C. Kearney, and A. van der Merwe, “The Zachman framework for enterprise architecture: An explanatory IS theory,” in *Responsible Design, Implementation and Use of Information and Communication Technology*, M. Hattingh, M. Matthee, H. Smuts, I. Pappas, Y. K. Dwivedi, and M. Mäntymäki, Eds. Springer International Publishing, 2020, pp. 383–396.

[30] J. A. Zachman, “A framework for information systems architecture,” *IBM Systems Journal*, vol. 26, no. 3, pp. 276–292, 1987.

[31] P. Pearce and M. Hause, “ISO-15288, OOSEM and model-based submarine design.” [Online]. Available: https://www.omgsysml.org/Pearce_Hause_ISO-15288_OOSEM_and_Model-Based_Submarine_Design_SETE_APCOSE_20121.pdf

[32] S. Friedenthal and C. Oster, *Architecting Spacecraft with SysML: A Model-based Systems Engineering Approach*.

[33] "IEEE guide for developing system requirements specifications," *IEEE Std 1233, 1998 Edition*, pp. 1–36, 1998.

[34] INCOSE Requirements Working Group, *Guide for Writing Requirements*, 3rd ed., INCOSE, Jul. 2019. [Online]. Available: https://connect.incose.org/Pages/Product-Details.aspx?ProductCode=TechGuideWR2019Soft

[35] L. Wheatcraft, M. Ryan, and J. Dick, "On the use of attributes to manage requirements," *Systems Engineering*, vol. 19, Nov. 2016.

[36] R. Carson, "Using architecture and MBSE to develop validated requirements," in *International Council on Systems Engineering 2020 Western States Regional*, Sep. 2020.

[37] R. S. Carson, E. Aslaksen, G. Caple, P. Davies, R. Gonzales, R. Kohl, and A.-E.-K. Sahraoui, "Requirements completeness," in *INCOSE International Symposium*, vol. 14, no. 1, Jun. 2004, pp. 930–944.

[38] *OMG Risk Analysis and Assessment Modeling Language*, Object Management Group Std., 2020. [Online]. Available: https://www.omg.org/spec/RAAML/