Advancing Model-Based Systems Engineering (MBSE) in the Development of Systems Architecture

Exploring the Value of MBSE during Early-Stage Naval Vessel Design

Vasileios Sideris

TU Delft

STARION

Cover image source: For U.S. carrier pilots, a vexing mission: hunting down Houthi missiles and drones. Story by Courtney Kube and Dan De Luce, 5 February 2024. Retrieved from `https://www.msn.com/en-us/news/world/for-u-s-carrier-pilots-a-vexing-mission-hunting-down-houthi-missiles-and-drones/ar-BB1hM4gW`

Advancing Model-Based Systems Engineering (MBSE) in the Development of Systems Architecture

Exploring the Value of MBSE during Early-Stage Naval Vessel Design

Thesis report

by

Vasileios Sideris

to obtain the degree of Master of Science in Marine Technology
at the Delft University of Technology
to be defended publicly on July 12, 2024 at 09:00

Student number:	5842727	
Project Duration:	November, 2023 - July, 2024	
Thesis number:	MT.23/24.040.M	
Thesis committee:	Dr. Austin A. Kana	TU Delft, supervisor and committee chairman
	Eng. Sam Gerene	Starion Group, supervisor
	Dr. Alessandra Menicucci	TU Delft, committee member
	Dr.ing. Cornel H. Thill	TU Delft, committee member
	Zacharias P. Oikonomou	TU Delft, committee member

Faculty of Mechanical Engineering · Delft University of Technology

Statement: During the preparation of this work the author used ChatGPT3.5 in order to review and improve grammar, spelling and expression. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of the publication.

Acknowledgements

After dedicating considerable time and effort, I am proud to present my thesis titled " Advancing Model-Based Systems Engineering (MBSE) in the Development of Systems Architecture: Exploring the Value of MBSE During Early-Stage Naval Vessel Design". With the completion of this thesis, I conclude my journey as a Master's student in Marine Technology at Delft University of Technology. Undoubtedly, it has been a challenging yet profoundly rewarding experience.

I am grateful to my supervisors for their invaluable guidance throughout the academic journey of this thesis. Their trust in me and the freedom they granted me to implement my ideas and formulate the research topic were instrumental to its success. Support from the Delft University of Technology was given by Austin Kana. Thank you, Austin, for your inspirational meetings, constructive feedback, and excellent guidance, which were crucial in keeping me focused and progressing steadily along the right path. In addition to Austin, I would like to express my gratitude to Sam Gerene, my supervisor from Starion. Thank you, Sam, for your insightful questions, mentorship throughout the project, and encouragement to engage in Starion's projects, which significantly contributed to my thesis.

Special thanks also go to my thesis advisor from TU Delft, Zacharias Panagiotis Oikonomou, who was always eager to assist me. I would like to express my gratitude to my colleagues from Starion for their invaluable feedback and ideas during my presentations. I am deeply grateful to Belen Gonzalez Rodriguez for her guidance in my modeling efforts and assistance with numerous challenges. I also extend my gratitude to Paloma Maestro Redondo and Gwendolyn Kolfschoten for their constructive feedback. Many thanks to the rest of the team members of the DND MMBSE pilot project for their support. Furthermore, I am profoundly grateful to HELLENiQ ENERGY for awarding me a scholarship for my master's studies.

On a more personal note, I would like to express my gratitude to my amazing friends for supporting me outside of the thesis. My heartfelt thanks go to my parents and sister for their unwavering support throughout my entire academic journey, from the beginning to the end of my studies. Finally, I extend extra special thanks to my wonderful girlfriend, Eleftheria, for her love, patience, and for bringing happiness into my life.

Vasileios Sideris
Delft, June 2024

Abstract

Technological advancements are increasing system complexities, posing challenges for modern warship design, which must integrate numerous interconnected systems and cutting-edge technologies. Early-Stage Ship Design (ESSD) is crucial, involving critical decision-making with limited information, capturing requirements, and maintaining flexibility to avoid costly impacts on performance and expenses. Particularly complex for warships, ESSD entails intricate decision-making considering temporal influences, interdependencies, external factors, trade-offs, high costs, evolving requirements, and other challenges. Developing well-defined system architectures is essential for managing these complexities. Model-Based Systems Engineering (MBSE) is pivotal for overcoming precision issues, inconsistencies, and difficulties in maintaining and reusing information associated with conventional document-centric approaches in systems engineering (SE).

Despite the growing popularity of MBSE, the maritime industry continues to rely on traditional document-centric approaches, lagging behind other sectors in adopting MBSE. These challenges stem from a lack of empirical evidence demonstrating MBSE's full benefits. Moreover, confusion persists regarding MBSE implementation practices. On top of that, there is a lack of comparative studies that assess how well MBSE tools are tailored for naval warship design. Thus, this research explores the value of MBSE in the early design stage of naval vessels. Specifically, the research aims to validate and demonstrate MBSE's benefits in developing systems architecture for warships, focusing on operational, functional, logical, and physical perspectives within the context of ESSD. By analyzing industry approaches and utilizing two representative modeling tools, this thesis provides practical insights, clarifies MBSE practices, and promotes its effective implementation in future naval design projects.

The first half of the literature review examines ESSD phases and challenges, emphasizing the need for well-defined system architecture and identifying four key design criteria. It concludes by evaluating contemporary design methods and highlighting MBSE's superiority. The subsequent chapter delves into MBSE's challenges and benefits, mapping the key factors underlying these benefits to the four design requirements. This theoretical analysis confirms MBSE's capability to handle the complexities of early-stage naval ship design and identifies success factors for demonstrating MBSE's value in this context. The literature review concludes with a comparative analysis of prominent MBSE tools, methods, and languages.

To achieve the thesis objective, a structured research process comprising 10 steps with iterative cycles is formulated. It begins by defining the mission, capabilities, and requirements. For illustration purposes, a hypothetical Landing Platform Dock vessel mission is chosen, necessitating the creation of fictitious capabilities and requirements. MBSE tools Capella and CDP4-COMET are then selected for this analysis. Next, a metamodel is established for a unified understanding among stakeholders, guiding decision-making throughout the design process. Once the baseline models are constructed, the validation and verification capabilities of the tools are evaluated. In this thesis, validation ensures the models are constructed correctly and conform to the predefined metamodel, while verification confirms the modeled system meets specified design requirements. Moving on, the baseline models are modified to simulate the dynamic nature of warship design. Finally, the tools are systematically assessed based on selected key MBSE factors: consistency, traceability, flexibility, and trade-offs.

Transitioning to conclusions, the analysis reveals that both tools effectively validate anticipated benefits. By synthesizing the knowledge gained, it is concluded that MBSE can enhance and accelerate the design process during the early design phase of warships. Capella demonstrates superior performance in the early design stages, whereas CDP4-COMET excels in the latter design stages. This emphasizes the critical role of integrating MBSE tools to enhance design outcomes, leveraging their strengths across diverse phases effectively. Thus, integrating MBSE tools and establishing a unified source of truth is crucial for advancing MBSE in ship design. Further recommendations include refining ship-wide metamodels for stakeholder alignment and decision-making, conducting cost-benefit analyses for economic feasibility, exploring diverse modeling languages, and addressing scalability challenges. Finally, integrating MBSE with existing ship design software, enhancing organizational readiness through training, and simplifying interfaces will benefit the widespread adoption of MBSE in naval vessel design.

Contents

List of Figures

List of Tables

Terms and Definitions

Amphibious Force (AF) — An Amphibious Task Force (ATF) is a Navy task organization, while a landing force (LF) is a Marine Corps or Army task organization, both formed to conduct amphibious operations. An ATF and an LF, along with other forces trained for amphibious operations, combine to form an Amphibious Force (AF).

Amphibious Operations — An amphibious operation is a military operation launched from the sea by an AF embarked in ships or craft with the primary objective of conducting landing force (LF) operations within the littorals to accomplish the assigned mission. The littorals include land areas that are predominantly susceptible to engagement and influence from the sea, extending potentially far inland. Amphibious operations use maneuver principles to employ ready-to-fight combat forces from the sea to the shore to achieve a position of advantage over the enemy.

Amphibious Raid — An amphibious operation that involves a swift infiltration or temporary occupation of an objective to fulfill a specific mission, followed by a planned withdrawal. It often includes seizing territory, securing information, confusing the enemy, capturing personnel, or destroying a capability. Surprise is essential and compensates for the lack of fire support in preparing the objective area. Maximizing surprise through deception, stealth, speed, disguise, and ambiguity is crucial. The size of the amphibious raid force is typically limited to the essential number of personnel required to accomplish the mission. It should be noted that an amphibious raid is planned and executed in the same general manner as an amphibious assault, except that it also includes a provision for withdrawal of the raiding force.

Amphibious Demonstration — An amphibious operation that targets to delude or confuse the enemy, leading them to select an unfavorable course of action. This is done by creating confusion regarding the timing, location, or strength of the main effort.

Amphibious Assault — An amphibious operation that requires the rapid buildup of combat power ashore, from zero to full striking power as the attack progresses toward AF objectives. During this operation the combat power is progressively phased ashore. The complexity during this operations makes it it one of the most difficult of all military operations.

Amphibious Withdrawal — An amphibious operation which can be conducted under enemy pressure or operational urgency, in various environments, to obtain forces needed elsewhere or to remove forces whose mission is completed. It commences with defensive measures in embarkation and objective areas and ends when all forces are safely embarked on designated shipping.

Planning Phase — A phase during an amphibious operation that involves detailed coordination and decision-making among all participating forces. Commander's orders provide initial guidance, ensuring alignment with overall objectives. The planning involves the preparation of the movement plan, staging areas, littoral maneuver, and ship-to-shore movement. Moreover, the landing plan is prepared, which represents the integrated sum of detailed plans for waterborne and airborne ship-to-shore movement. It presents in detail the numbers of landing craft, helicopters, and surface craft available for use, along with the exact personnel and equipment to be loaded on each, alongside embarkation and landing times. Supporting functions such as fire support, intelligence, and logistics are vital considerations, alongside preparations.

Embarkation Phase

A phase during an amphibious operation that encompasses the orderly assembly of personnel and materiel and their subsequent loading aboard ships and/or aircraft in a sequence designed to meet the requirements of the landing force concept of operations ashore. In other words, it is the phase in which the forces, along with their equipment and supplies, are embarked in assigned ships/aircrafts.

Rehearsal Phase

A phase during an amphibious operation that refers to the period after embarkation and prior to the action phase during which the prospective operation is practiced. This phase may be conducted in parallel with other phases of the amphibious operation but usually is associated with the movement phase.

Action Phase

A phase during an amphibious operation that refers to the period of time between the arrival of the landing forces of the amphibious force in the operational area and the accomplishment of their mission. Prior to the execution of the action phase of an amphibious operation support and prelanding operations take place. Plans include actions aiming to hide the force, confuse the enemy, and reduce the enemy's sensors effectiveness, while embarking and rehearsing and during movement and the action phases. The ship-to-shore movement is that portion of the action phase of an amphibious operation that includes the deployment of the landing force from ships to designated landing areas.

Landing Helicopter Dock (LHD)

A multipurpose amphibious assault ship which has a full-length flight deck. The LHD's primary mission is AMW. Similar to the LHA in design, mission, and capabilities, the LHD's significant improvements over the LHA include increased AV-8 support capabilities, a redesigned well deck, expanded medical facilities, and upgraded command and control capabilities (C2).

Landing Helicopter Assault (LHA)

A general purpose amphibious assault ship which has a full-length flight deck and extended hanger to support aviation facilities. They combine the operational capabilities of several types of amphibious ships in a single hull. This class has the primary mission of Amphibious Warfare (AMW) and a secondary mission of power projection and a tertiary mission of sea control. It is designed to put ashore a Marine Expeditionary Unit using helicopters and MV-22B Osprey V/STOL transport aircraft, supported by AV-8B Harrier II or F-35 Lightning II V/STOL aircraft and various attack helicopters. LHAs have vertical lift aircraft and V/STOL aircraft operating facilities greater than the LHD, a well deck capacity greater than LPD and the LSD, and a substantial vehicle and cargo capacity.

Landing Platform Dock (LPD)

An amphibious warfare ship that embarks, transports, and lands elements of a landing force for expeditionary warfare missions. It supports the landing of troops, equipment, and supplies using landing craft, AAVs operating from its well deck, and helicopters and tiltrotor aircraft operating from its flight deck. The LPD is a variation of the LSD concept with increased troop and vehicle capacity, extensive ammunition and cargo stowage capabilities, and a smaller well deck. An LPD has a hangar in addition to the helicopter deck.

Landing Ship Dock (LSD)

An amphibious warfare ship with a well dock to transport and launch landing craft and amphibious vehicles. The dock landing ship (also called landing ship, dock or LSD) also facilitates a helicopter deck. LSDs support the landing of troops, equipment, and supplies using various types of landing craft and AAVs using its well deck. It has the ability to render limited docking and repair services to small boats and crafts.

Amphibious Assault Vehicles (AAVs)
Wheeled or tracked vehicles capable of operating on both land and water. Beyond transporting troops ashore and conducting inland missions, they can also move a limited amount of cargo ashore. Such an example is the Assault Amphibious Vehicle (AAVP-7A1).

Landing Craft Utility (LCU)
A highly versatile displacement craft designed to beach where hydrographic and weather conditions permit. LCUs transport wheeled and tracked vehicles, general cargo, and personnel from ship to shore, shore to ship, shore to shore, and in resupply, backload or recovery operations. Examples include the LCU 1466, 1610, and 1627 classes in the US Navy.

Landing Craft Air Cushion (LCAC)
A high-speed (35+ knots), non-displacement landing craft which is carried in the well deck of an amphibious warfare ship and can transport troops, vehicles, and cargo over-the-beach.

1
Introduction

1.1. The Need for System Architecture in Early Ship Design Phases

The technological advancements of the past few decades have drastically altered the way that ships are designed and operated (Brefort et al. 2018). Modern warships incorporate a plethora of interconnected and interoperable warfare systems. They are characterized by advanced automation, versatile mission capabilities, and the imperative to operate in challenging environments. These factors result in a lengthy and detailed acquisition process, which spans several years and can extend up to a decade or more for the most intricate naval ships (Tepper 2010). Furthermore, a modern warship typically has a service life that spans several decades. Predicting the future needs of warships for a period of up to 50 years is exceptionally challenging, primarily because of the rapid evolution of software-based technology and military capabilities, which introduce unpredictability into the design process (Tudor and Harrison 2019). Moreover, given the growing dynamic nature of a combatant ship with interconnected systems, personnel, and interactions, it becomes necessary to classify them as complex system architectures (Brefort et al. 2018). This complexity underlines the pivotal role of the systems engineering process within the ship design process, as it offers the essential methodology for effectively managing complex systems.

The systems engineering process is a top-down, iterative problem-solving approach, applied sequentially through all stages of a vessel's development. Its purpose is to provide a structured but flexible process that transforms needs into system descriptions, informs decision-makers, and provides input for the next level of development (Lightsey 2001).

During the transition from a systems engineering approach to ship design, what has been overlooked is the aspect of developing a well-defined system architecture Tepper (2010). Specifically, it is stated that ship designers often spend limited time modeling and clearly defining the system's architecture (Tepper 2010). A more recent study stresses a lack of support for traceable, robust capability definition in the early stages of naval ship acquisition projects (Morris 2019). Additionally, Tudor and Harrison (2019) underscores the necessity for a robust process in developing system architecture to describe a warship's functional, physical, and operational breakdown.

The concept of systems architecture can be subject to varying interpretations among individuals. In general, systems architecture can provide an effective way to understand, design, and manage complex systems by representing the latter as an abstraction of entities and the relationship between those entities. In this thesis, the term "system architecture" encompasses a conceptual model that underlines the structure of a system across four distinct domains. These domains align with the level views utilized to elaborate and share the architectural model within the Arcadia method (Capella n.d.). They are also referred to as layers or perspectives:

- **Operational Layer:** It defines the problem by capturing what the system users intend to accomplish. It focuses on analyzing customer needs, expected missions, and operational activities. The output of this phase consists of an "Operational Architecture (OA)", which describes and structures the need in terms of actors/users, their operational capabilities, and activities.
- **Functional Layer:** It describes what functions are needed to be accomplished for the users. It focuses on defining how the system can satisfy the operational needs and the expected behavior. The output of this phase consists of a "Functional Architecture (FA)" and it should be developed in

parallel with stakeholder requirements and the physical architecture to guarantee that the proper functions and interfaces are identified.

- **Logical Layer:** It describes how the system will work to achieve the expected functions. This perspective aims to build a coarse-grained component breakdown of the system with primary and less likely-to-be-challenged decisions of the solution. Therefore, the "Logical Architecture (LA)", the output of this phase, maintains abstract decisions from which the next phase can create concrete solutions consisting of physical components.
- **Physical Layer:** It describes how the system will be developed and built. Similar to the LA, it defines the solution architecture but concentrates on the "final" architecture for system development, implementing technical and technological constraints and choices. The output is the identification of "Physical Architecture (PA)", which includes components, formalized viewpoints, their integration in component design, as well as links with requirements and operational scenarios.

Distinguishable states can be identified during the development of a vessel. One primary phase, Early-Stage Ship Design (ESSD), stands out as a crucial step, as it is the stage during which vital decisions are made with limited knowledge available. Capturing customer requirements and transforming this information into a viable concept design can be identified as the main objective of ESSD (Andrews 2018). The complexity of this stage, with conflicting information, and the need for flexibility and full traceability to avoid locking in crucial decisions, emphasizes its critical nature (Poulis 2022). These insights highlight the pivotal role of a well-defined system architecture during ESSD in guiding decisions throughout the ship design process.

1.2. Terminology in Ship Design

While ESSD activities focus on requirements elucidation, there are several ways to achieve this goal. Hence, a clear definition of some important design effort elements used throughout this report is judged necessary. The following definitions range from overarching/social to more technical aspects (A. Kana 2023c; T. McKenney 2013).

- **Design Framework:**
  Provides a conceptual structure for developing, implementing, governing, and sustaining various architectures. It outlines a method for designing an enterprise's target state using building blocks, supported by tools, a shared vocabulary, and recommended standards for efficient implementation.
- **Design Approach:**
  Describes the initial guidance for the initiation of the design effort. Examples include sequential (or point-based) concept exploration where only a few design solutions are explored and concurrent (or set-based) concept exploration, where many solutions can be generated and assessed in parallel.
- **Design Process:**
  Provides a structure or a framework for applying the design approach. The processes may align with an approach, or an approach may need to conform to a given process. Examples include the US/NL ship acquisition processes and more relevant to this thesis, Systems Engineering.
- **Design Method:**
  Describes how the design approach is applied, enabling the analysis, and selection of design alternatives. Examples include iterative (point-based), convergent (set-based) design, and Model-Based Systems Engineering, which is a methodology for Systems Engineering that uses models to manage system complexity.
- **Design Tool:**
  Supports the design methods by providing information that enables designer decision-making. Examples include Multi-fidelity, optimization, and fuzzy tools. In the case of Model-Based Systems Engineering, it includes environments and languages.

1.3. The Need for System Modeling

The increasing complexity and need for flexibility during ESSD underscores the urgent shift from document-centric to model-centric systems engineering as a means to address inconsistencies and enhance competitiveness (Packham 2021). Indeed, computer modeling has increasingly been employed to digitally capture system details and thus track complexity, particularly in the early stages of the design process, when the cost of making design changes is relatively low (Long and Scott 2012; Odukoya et al. 2023). This is done to establish a robust system architecture and predict the emergent behavior and properties of a design (Pearce 2013). Enhanced communication, continuous requirements validation, design verification, rigorous traceability, improved decision-making, and design consistency are some of the benefits of integrating various digital models of the same system (Tepper 2010; Pearce 2013). The approach, in which computer models are used to oversee the specification, design, evaluation, and support of a system throughout its life-cycle, is referred to as Model-Based Systems Engineering (MBSE) (Pearce 2013).

The International Council of Systems Engineering (INCOSE) 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" (INCOSE 2007). Applying MBSE requires selecting the three main pillars, which include a method, a tool, and a modeling language (Delligatti, Soley, and Steiner 2014). The tool represents the software leveraged to capture the system architecture, establishing an underlying network of interconnected elements. The method is employed to systematically organize the process and elucidate the activities, techniques, and conventions involved in developing a system architecture. The language serves as a standardized communication method, capturing the grammar of a system architecture to translate practical demands into manageable units of unified information. The resulting system model will involve operational, behavioral, and structural system characteristics, along with essential data regarding system requirements, capabilities, mission needs, as well as actors and interactions between the elements involved (Odukoya et al. 2023).

Currently, ship design traditionally still relies on document-centric approaches. Even though MBSE can offer significant advantages for complex system design, its adoption in the maritime industry has been relatively limited compared to other industries (Verma and Daly 2021; Poulis 2022). A transition toward a model-centric approach would require extensive research and training to proficiently develop the system model and fully leverage the advantages of a single source of truth (Tepper 2010; Poulis 2022). As noted by Tepper (2010), the full potential of MBSE benefits was not realized in the community as of 2010. Similar concerns persist in more recent studies, with observations indicating that approximately two-thirds of the anticipated benefits of MBSE have still not been realized with measured evidence (Henderson and Salado 2021). Building upon these considerations, it can be suggested that current benefits are largely based on theory and a relatively limited number of use cases (Henderson and Salado 2021).

It is noteworthy at this stage that the current thesis is done in collaboration with Starion Group (previous RHEA Group). Starion has a trajectory of over 25 years offering systems, solutions, and services in the space sector. This company has been actively working with fully functional MBSE environments during the early phases of complex projects for several years. It supports its clients by conducting an initial evaluation of MBSE and offering continuing support at all stages of implementation and application(RHEA 2023). Moreover, Starion Group is exploring the feasibility of applying Concurrent Design (CD) practices beyond the space industry. In addition to supporting the space sector, the company is engaged in pilot projects aimed at implementing CD throughout MBSE in the maritime industry. Their projects are founded on the principle that MBSE should enable the use of the best tools and languages for every task, such as Capella and SysML, by offering all necessary translation capabilities between them. Starion uses its in-house MBSE tool, CDP4-COMET, which is an open-source collaborative tool initially developed for the space sector but now widely employed across various industries, including the naval industry (RHEA 2023). Hence, Starions's in-house expertise in the practical implementation of MBSE can support the development of a well-defined system architecture.

In conclusion, there is a clear need to explore the value of MBSE for capturing and communicating systems architecture in a structured, consistent, and coherent fashion from the perspective of a naval architect. Additional publicly available empirical evidence is required to validate the perceived benefits of investing in MBSE with the ultimate goal of tackling complexity and facilitating informed decision-making in ESSD.

1.4. Research Gap

Considering insights from both literature and industry, a clear research gap has been detected as follows:

- **Need for Validation of MBSE Benefits in Early-Stage Naval Vessel Design**
  There is a growing need to validate the expected benefits of MBSE, not only in more detail but also

on a broader scale, including the maritime industry. Publicly available empirical evidence is required to validate the expected benefits of MBSE and explain why, how, and to what extent they can add value to the ESSD of naval vessels. However, ESSD is a broad term. Nowadays, the integration of SE processes and computer-based models results in defining the ESSD process as a detailed system architecture. Subsequently, it is important to explore the value of MBSE, by specifically focusing on its impact on systems architecture development during the early-stage design of warships.

- **Absence of Comparative Analysis in MBSE Tools for Early-Stage Ship Design**
  Another notable gap in the literature is the lack of a comparative analysis of MBSE tools, particularly concerning the construction of the system architecture for warships. This highlights the potential benefits of a systematic comparison of basic MBSE tools to support engineers in making optimal tool selections during the early design phases. The lack of simple comparative analysis represents a missed opportunity to provide practical insights, alleviate the confusion surrounding MBSE practices, and promote its effective implementation in the field.

1.5. Research Objective and Research Questions

The leading objective of this research is to explore the value of MBSE by validating its benefits in the early-stage design of naval vessels. This project seeks to confirm and demonstrate that MBSE can produce valuable results that meet specific objectives within the context of ESSD. Within the broader context of ESSD, this thesis aspires to specifically concentrate on the development of systems architecture, encompassing operational, functional, logical, and physical perspectives. By studying contemporary industry approaches and gaining hands-on experience with different modeling languages, methods, and tools, this thesis aims to leverage Starions's in-house expertise and explore the implementation of MBSE in future naval design projects. The analysis of combinations of modeling tools, methods, and languages for architecting a selected system will not only deepen the understanding of MBSE's value but also will provide insights into its practices and aid in informed tool selection, all from a naval architect's perspective. It is important to clarify that the emphasis of this thesis is not on providing a generalized MBSE ship design method. Instead, the approach will involve a meticulously selected example system to systematically demonstrate and validate the benefits of MBSE. The insights from this graduation thesis can be used as valuable input for the development of a modeling strategy in follow-up projects.

Aligned with the research objective, this thesis is structured to address the following central question:

Research Objective

How can the value of MBSE, specifically its benefits, be effectively demonstrated and validated in the development of systems architecture during the early-stage naval vessel design?

To accomplish the research objective and address the central research question, the following set of sub-questions has been formulated.

Research Question 1

Why are the early design stage and the development of well-defined system architecture critical in naval design and what are the limitations of the contemporary design methods/approaches? (Chapter 2)

Research Question 2

What is MBSE, and how can it theoretically aid in managing the increased complexity of early-stage naval ship design? (Chapter 3)

Research Question 3

What are some commonly used combinations of MBSE methods, tools, and languages, and what are their strengths and weaknesses according to the literature? (Chapter 3)

Research Question 4

What critical factors play a key role in selecting a representative case study, considering a) the selected system of interest, b) the level of detail, and c) the chosen modeling tool-method-language? (Chapters 5 and 6)

Research Question 5

How can the selected MBSE tool-method-language be effectively applied to model the architecture of the chosen system, and what critical insights can be gained from this modeling process? (Chapter 6)

Research Question 6

How does the selected MBSE tool-method-language, perform in terms of consistency, traceability, flexibility, and trade-off analysis within the context of system architecture development? (Chapter 7)

Research Question 7

To what extent can MBSE become a standard working method for future naval vessel design? (Chapter 8)

1.6. Thesis Lay-out

This section describes how this thesis is structured.

- Chapter 2: Ship Design

  In this chapter, the significance of systems architecture and the uniqueness of the early naval vessel design stage are underscored. Next to that, current ship design approaches are discussed. Concluding remarks regarding the assessment of these approaches about a selected set of key method requirements are presented. To meet these requirements, the systems engineering process is investigated, leading to the introduction of the MBSE method. By the end of this chapter, Research Question 1 can be fully addressed.

- Chapter 3: Model-Based Systems Engineering

  This chapter provides a comprehensive background on MBSE. It begins with an elaboration of the three interconnected concepts that MBSE integrates and then it progresses with an elucidation of MBSE, delving into its core components, namely the modeling tool, modeling method, and modeling language. The chapter then details the broad benefits of MBSE, highlighting key factors that are behind these advantages. These factors are later linked to the ESSD method requirements established in Chapter 2. Following that, the chapter addresses challenges within MBSE. Next, a theoretical comparative analysis among well-known tools, methods, and languages is presented. By the end of this chapter, Research Questions 2 and 3 can be fully addressed.

- Chapter 4: Research Process and Implementation Steps

  This chapter outlines the systematic process undertaken to achieve the research goal of this thesis. First, the requirements necessary for the process are presented, followed by a detailed, step-by-step breakdown. This includes defining the mission, capabilities, and requirements, selecting MBSE tooling, defining a metamodel, constructing MBSE models, verifying-validating and modifying them, collecting, sharing, and analyzing modeling constructs, and finally, evaluating tools based on MBSE factors.

- Chapter 5: Background of the Case Study

  This chapter focuses on the initial phase of the proposed process of Chapter 4, centering on the preparatory steps that establish the groundwork for the example problem. It begins by outlining the scope of the case study, followed by a comprehensive examination of the mission and system of interest. The mission is then translated into a series of operational capabilities, which, in turn, are translated into a set of requirements for the vessel under consideration. By the end of this chapter, Research Question 4 is partially addressed.

- Chapter 6: Modeling

  This chapter focuses on the next phase of the proposed process of Chapter 4, centered on the steps regarding the modeling effort. First, the rationale behind the MBSE tool selection is presented. This helps fully address the remaining part of Research Question 4. Then, the construction of a design template to guide the modeling process takes place. Next, a comprehensive summary of the modeling effort in both tools is provided. Through this practical application, the Literature Research Question 5 can be fully addressed.

- Chapter 7: Assessment of Tools Based on MBSE Factors

  This chapter focuses on the final phase of the proposed process, concerning the evaluation of the selected tools based on the MBSE factors. Motivation for studying each factor, its significance in assessing the tools, and compliance methods are provided. Subsequently, detailed assessments of the selected tool are conducted through practical demonstrations and tests. The results of the comparative analysis are summarized in a comprehensive table. This examination addresses Research Question 6.

- Chapter 8: Conclusions and Recommendations

  This chapter focuses on concluding this graduation research assignment. Firstly, it restates the research gap and augments it with additional insights garnered throughout this thesis. Next, the methodology framework, its implementation, and the key findings and limitations for each project phase are discussed. The chapter proceeds to wrap up with a comprehensive analysis of the comparative study conducted among the chosen MBSE tools. After that, it presents the findings in response to the Research Questions in a concentric manner, also addressing Research Question 7. Finally, recommendations for future research are outlined, paving the way for advancing MBSE in early-stage warship design.

2 Ship Design

The chapter “Ship Design” serves as the first half of the literature background upon which this thesis will be based. Within this chapter, various aspects introduced in Chapter 1 are further elaborated. First of all, the distinct phases of the ship design process are briefly analyzed. Next, a more thorough discussion on the challenging nature of Early-Stage Ship Design is presented. The chapter then emphasizes the need for a well-defined system architecture in the early stages of naval vessel design. The unique challenges inherent in naval combatant design give rise to key requirements for typical early-stage design methods. Following this, the chapter delves into the benefits and limitations associated with various design methods. Finally, the evaluation of these methods with the key requirements brings to light the superiority of Model-Based Systems Engineering. Consequently, the Research Question 1 (*“Why are the early design stage and the development of well-defined system architecture critical in naval design and what are the limitations of the contemporary design methods/approaches?”*) can be fully addressed.

2.1. Ship Design Phases

The process of ship design consists of multiple phases, also known as stages, each with unique characteristics (Andrews 2018). Depending on the approach, these phases can vary in number. According to Habben Jansen (2020), three design stages have been identified: Concept Exploration, Concept Definition, and Detailed Design & Engineering. The phases can be divided or combined to form a different division (Chalfant 2015). This division is guided by specific drawings or documents that need to be before or at the end of each design stage, known as milestones (Scheffers 2021).

Despite variations in terminology and the number of steps found in the literature on ship design, four main consecutive design phases, which occur before production, can be commonly identified. Those are analyzed in the following paragraphs.

- **Concept Design**
  This phase is also known as the cost and feasibility study phase. The main goal is to understand the client’s specific requirements, encompassing the ship’s mission and key performance attributes. The milestones of this phase include a clear mission statement and a concept design aligning with the overall needs of the customer. This highlights the substantial role that this phase has in the overall design process. Additionally, given the conflicting nature of requirements in terms of financial and technical feasibility, this phase involves creating a cost estimate and conducting a risk assessment. The primary trade-off involves balancing cost, construction time, and the necessary ship capabilities. Generally, the capacity of key onboard components is determined in this phase (Scheffers 2021).
- **Preliminary Design**
  In this phase, both the ship costs and the performance of the vessel are determined. The trade-off studies conducted in the current phase will lead to decisive design decisions that will significantly influence the ship’s dimensions, overall configuration, performance, and costs. Key components determined include the hull shape and size, the general arrangement plant, mission-critical payload features, the crew size, and several other principal systems if those are not decided earlier, such as the propulsion plant (Scheffers 2021).
- **Contract Design**
  This is the phase in which the solution is analyzed in more detail to describe a contract and determine

a contracted price (Duchateau 2016). With the preliminary layout of systems being designed, the contracts for materials and equipment are studied in the current phase (Bakker 2021). As shipyards often have their suppliers, this more detailed design phase is predominantly carried out by the shipyard itself or in close collaboration with them, assessing producibility and cost (Duchateau 2016; Bakker 2021).

- **Detailed (Engineering) Design**
  This final phase constitutes a significant portion of the entire design process. It involves the translation of the outcomes from the Contract Design stage into a production-ready design. This is achieved through the generation of detailed engineering and production drawings. Often, this stage overlaps with the vessel's construction, which may commence before the finalization of the engineering drawings (Duchateau 2016).

The current thesis examines and analyzes the system architecture during the initial phase of the ship design process. Therefore, the first two of the aforementioned phases, the Concept Design and Preliminary Design phases, are highly relevant to this report and are further explored in the following sections. It is important to note that these two initial stages collectively constitute the Early Stages of Ship Design (ESSD).

2.2. Early Stage Ship Design (ESSD)

Now that the position of the ESSD phase is considered within the design process, it is possible to describe its nature in further detail. In general, a major difference distinguishes the early design stage from the rest of the stages during the development of any product or system. This difference lies in the fact that the early design stage, is involved in elucidating the appropriate requirements and identifying the design problem, rather than engaging in actual engineering design work (Andrews 2011). The main task during the ESSD is capturing the customer requirements and translating this set of information into a feasible concept solution. More specifically, based on the customer's requirements, a set of design solutions is proposed by the naval architect. The findings from this initial investigatory study are then used to narrow down and select one or several potential solutions, which are then explored in more detail (Duchateau 2016). However, in cases where the budget is limited, it is also possible that only a limited number or even just one design solution is explored (Bakker 2021).

ESSD is considered to be one of the most crucial phases in the vessel's design process (Charisi, Austin Kana, and J. Hopman 2022). This is explained by the fact that several important and costly performance-related decisions regarding the vessel are made and locked in early in the process. Therefore, it can be easily realized that non-informed decisions and poor designs can lead to significantly raised costs and unwanted consequences, as initial choices constrain subsequent design decisions (Poulis 2022).

One of the main challenges of this stage is that decisive design decisions have to be made with limited available knowledge of the design (Jansen et al. 2020). This challenge has previously been captured by Mavris and DeLaurentis (2000) in a well-known and often used generic design timeline. An adapted version of this timeline is presented in Figure 2.1, emphasizing the importance and influence of the early design stage. This figure depicts the ample design freedom available in the early stage. Major decision moments or transitions from one stage to the next will result in a decrease in design freedom. The challenge of understanding requirement interactions and the limited availability of detailed information becomes apparent early in the design process. Hence it is estimated that this phase contains the highest uncertainty. This is reflected early in the graph, indicating that the level of problem knowledge remains low. Nonetheless, it is during these initial stages that pivotal and defining design decisions are made. Consequently, the remaining design freedom reduces rapidly, limiting the capacity to modify the design and committing a substantial amount of cost. Therefore, making early decisions can result in significant modifications in the design, potentially leading to considerable cost overruns in later stages when adjustments are deemed necessary (Duchateau 2016).

It can be concluded that most of the performance and committed costs of the design are locked in during the early design stage. Despite this, the limited knowledge available on the problem and potential solutions poses serious challenges. Recognizing the critical role of early-stage knowledge, Wilcox (2018) underlines the necessity of introducing the "right" information earlier in the design process. To enhance decision-making and thus achieve better designs, understanding the vessel's performance is paramount (Charisi, Austin Kana, and J. Hopman 2022). Consequently, there is a vital need for effective design

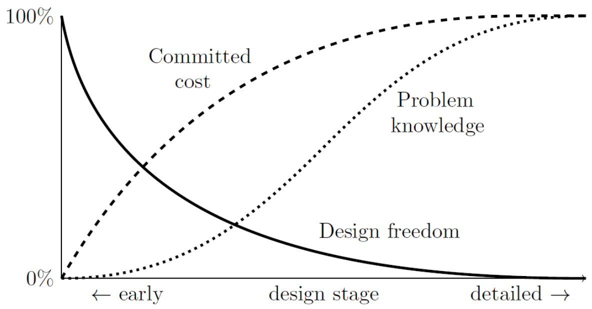


Figure 2.1: Simplified generic design timeline. Source: Mavris and DeLaurentis (2000)

methods within ESSD to empower naval architects with the necessary knowledge, methods, and tools. This supports them in making the right decisions and generating feasible concepts.

2.3. Importance of a Well-Defined System Architecture

Understanding, designing, and managing complex systems is enhanced through the significance of system architecture. Every system possesses an architecture that shapes its behavior (Tepper 2010). Engineering organizations often start modeling without recognizing that the initial problem statement may not be the best, or even right (Madni and Sievers 2018; Hein et al. 2011). Neglecting attention to system architecture and its influence on design can result in potential integration challenges downstream. Building incorrect features represents a costly form of waste in engineering development, and engineers often start modeling without recognizing potential flaws in the original problem statement (Madni and Sievers 2018). Specifically in the ship design process, it is pivotal to ensure that the system architecture is not only well-defined but also that effectively meets the stakeholder needs. The latter demands that the development of system architecture must begin at the very early stages of the ship design process (Tepper 2010).

System architecture abstracts and conveys a viewpoint. It is a formal representation of a system, designated to facilitate understanding by organizing relationships, processes, fundamental system components, constraints, and behaviors (Madni and Sievers 2018). It is a composition of diagrams or elements to reveal the solution structure through the eyes of a specific stakeholder, revealing the systems' underlying framework. Structuring a system architecture is argued to be a preferable strategy to deal with really complex systems (Maier 1996). This is due to the strong emphasis on the concept phase which is a crucial key in ensuring the coherence in the downstream design process (Maier 1996; Andrews 2018). Such an approach better meets the core task of requirements elucidation in designing complex vessels.

During the early-stage design of naval vessels, there is a need to evaluate changes in the system architecture caused by shifting requirements. A well-defined system architecture early in the design process enforces structure in stable elements while providing flexibility in elements prone to change. As demonstrated by Tepper (2010), a change in the ship's prime mover can be investigated by observing from specific diagrams which functions and which inputs and outputs may be affected. Tepper (2010) further emphasizes that a well-defined system architecture promotes better decision-making through quantifying design options, conducting precise change assessments, and enhancing requirements traceability as well as communication among stakeholders, thus reducing risk and minimizing overall costs.

Apart from the benefits in structure and decision-making, system architecture development enables several other key priorities for the Navy (Tepper 2010). First, system architecture development is pivotal for achieving modular open systems, enabling efficient module reconfiguration, fleet modernization, streamlined maintenance, and reduced production times and costs. Moreover, the development of a solution-neutral system architecture is important for maximizing the potential of Integrated Power System (IPS) technologies. The latter enables the evolution of all-electric ship designs with superior mission performance and advanced weaponry integration.

Expressing a system architecture effectively is challenging for several reasons identified by Tepper (2010), summarized here. These challenges include confusion surrounding the term “systems architecture” itself as it lacks a universally agreed-upon definition. Furthermore, the naval ship design community has not fully embraced the benefits that come with well-defined system architectures. Moreover, there is a lack of a formalized process for developing systems architecture in the overall ship design process.

In light of these challenges, establishing a robust process for developing system architectures that describe a warship's operational, functional, logical, and physical breakdown seems imperative. This becomes particularly crucial as warship designers grapple with diminished ship companies, intricate software-based control systems, limited defense budgets, and development timelines. The demand for models to manage risks in such complexities is becoming urgent, as also highlighted by Tudor and Harrison (2019).

2.4. Uniqueness of the Early-Stage Design for Naval Vessels

At this point, it is crucial to analyze the unique nature of the naval design problem and the massive decisions that need to be taken. In general, engineering design is a decision-making process, which is often iterative (ABET 2015; A. Kana, C. Shields, and D. Singer 2016). The fundamental importance of decision-making in engineering design has been known for decades (Le Masson, Dorst, and Subrahmanian 2013). Undoubtedly, designing warships involves complex decisions throughout the lifecycle (A. Kana, C. Shields, and D. Singer 2016). The ship design process is seen not as a series of tasks but as a sequence of crucial decisions. Most of these decisions are taken at the very front of the design process, highlighting its significance (Andrews 2018). However, effective decision-making is still challenging. The US Navy's Office of Naval Research notes difficulties in quantifying design impacts due to a lack of standards (ONR 2012).

Andrews' several publications (Andrews 1998; Andrews 2011; Andrews 2018) have been motivated by the belief that the design of Physically Large and Complex (PL&C) systems, such as a naval vessel, is inherently sophisticated from the early stages. Even though many practitioners view the ESSD as relatively straightforward and less complicated, recognizing this inherent sophistication is crucial for achieving a successful design (Andrews 2018).

A. Kana, C. Shields, and D. Singer (2016) sought to address the question of what makes naval design decision-making challenging. Effective decision-making and selecting the optimal product require understanding how a product attribute influences the design's behavior throughout its lifecycle. Therefore, accurate problem identification and investigation of possible solutions are deemed crucial. Despite the substantial financial investment in the development of design and analysis tools, failures in naval design still persist. There are inherent difficulties in naval design that makes guiding the development of design and analysis tools activities extremely challenging (A. Kana, C. Shields, and D. Singer 2016). This makes it essential to outline the issues associated with understanding design decision-making. Multiple factors will be discussed in the following paragraphs, underscoring the continuously evolving landscape that necessitates the ongoing development of design and analysis tools.

First, the naval design environment includes elements of unpredictability, dynamic dependencies, interconnections, and influences from external factors. These are further explained below (A. Kana, C. Shields, and D. Singer 2016).

- **Temporal Influence and Decision Relationships**
  Sequential decisions in design involve intricate inter-relationships and dependencies. Those may not be immediately transparent to decision-makers. Increased risk and hindrance of future opportunities result from failing to recognize these decision relationships (A. Kana, C. Shields, and D. Singer 2016). The substantial influence on lifecycle cost estimates and the strategic advantages highlights the necessity for early decision-making in the design process, despite limited knowledge (Seram 2013; Mavris, DeLaurentis, et al. 1998; Mavris and DeLaurentis 2000).
- **Interdependence Between Design Problems and Solutions**
  The requirement definition and solutions efforts in naval design are intertwined. This consideration had led Andrews (1998) to characterize naval ship design as a "wicked" problem. The latter implies that the problem remains unclear until a solution is formulated. Furthermore, the solutions to "wicked" problems are not categorized as right or wrong, and there are no predetermined alternative solutions. Additionally, every "wicked" problem is fundamentally novel and unique. Due to the strong link between requirements and solutions, defining the requirements is more complex than identifying the actual solution. Specifically, before jumping to the solution of a design problem, feasible design requirements need to be established. To evaluate their feasibility, these requirements must be checked with one or more physically achievable concepts, which act as solutions. This self-referential loop,

in which generating solutions necessitates requirements, while establishing requirements requires solutions, is what makes it a "wicked" design problem (Habben Jansen 2020).

- **Sensitivity to External Influences**
Naval design is influenced by external factors such as budget fluctuations or safety regulations. Depending on the sensitivity of decision-making, the inability to account for temporal externalities can significantly alter the design course, forcing designers to adjust their choices. The prediction of those requires advanced decision analysis tools. However, effective decision-making in response to externalities is challenging, needing evaluation of impacts on the design process, and temporal factors. This becomes even more challenging coupled with the interdependence of solution development and uncertainty, as explained before. Designs addressing uncertainty, like robust and flexible designs, are incapable of eliminating decision interdependencies, making a coherent decision-making strategy difficult (Hastings and McManus 2004; A. Kana, C. Shields, and D. Singer 2016).

Besides the aforementioned gap between design freedom and problem knowledge, as well as the physical factors hindering decision-making, the early stage of naval vessel design is associated with several other challenges. According to Kooij (2022), warships and their corresponding design processes are characterized by several aspects that underline the inherited complexity. Some of those aspects are listed below:

- **Interactions and Trade-Offs**
Interactions in naval vessel design pose challenges due to complex dependencies among requirements (Duchateau 2016). Historical trends mandate technological complexity for operational effectiveness in the maritime industry (Bahlman 2012). Warships demand a high level of technology and autonomy to remain competitive (Andrews 2018; L. C. Kerns, Alan Brown, and Woodward 2012; Morris 2019). This, coupled with the presence of numerous onboard systems, results in complex interactions among various elements. Nonetheless, other factors such as seaworthiness and cost are also heavily involved, giving rise to conflicting, interdependent, or non-feasible requirements. Achieving optimal designs with advanced features, self-defense, speed, and budget constraints poses a significant challenge for naval architects (Habben Jansen 2020; Andrews 2011). In other words, trade-offs are inherent in naval ship design (Andrews 2011).

- **Limited Experience and Novelty**
Naval vessel design involves numerous complexities, requiring expertise that is hindered by a lack of previous experience. The challenge arises from the production of small quantities of warships and their extended design and construction timeframes (Tepper 2010). Additionally, the complexity and novelty in design are interrelated; novel vessels, as emphasized by Charisi, Austin Kana, and J. Hopman (2022), lack previous knowledge due to their unique nature, necessitating the exploration of new solutions.

- **Acquisition Cost**
Naval acquisition projects, entail high development costs, reaching up to billions of dollars (Kooij 2022; A. Kana, C. Shields, and D. Singer 2016). Most of the costs are committed during the early design stage (Andrews 2018) and can reach up to 70% of the total lifecycle costs, as highlighted by Dierolf and Richter (1998).

- **Evolving Requirements**
Naval vessel design is connected to an evolving and unpredictable set of requirements (Habben Jansen 2020) shaped by factors like political considerations (Kooij 2022) or evolving insights. The latter could be the customer-shipyard elicitation process, which may involve defining customer needs, affordability, and shipyard constraints, or the technological progression during the project's lifespan (Dick, Hull, and K. Jackson 2017).

Several studies in the literature, including works by A. Brown and Thomas (1998), Van Oers (2011), Duchateau (2016), A. Kana, C. Shields, and D. Singer (2016), Colin Shields (2017), and Habben Jansen (2020), have identified and discussed these challenges. What is common to each of these studies is the importance the authors attribute to the critical role of ESSD in the entire design procedure, because by the end of this phase, most of the cost is incorporated in the design. In a nutshell, ESSD is about making significant decisions.

2.4.1. Key Requirements for Effective Early-Stage Design Methods in Naval Vessel Development

With that said, it would be more beneficial for this research to conclude with a manageable number of key requirements for typical early-stage design methods for a combatant vessel. Therefore, four main key requirements are identified, which will later be used to evaluate available ESSD methods. These requirements are listed as follows:

1. **Better Decision Management**
   Design methods should adeptly handle dynamic dependencies between problems and ensure transparent decision-making with limited knowledge available. It is important to recognize decision relationships early to reduce risks and avoid hindrances to future opportunities. A centralized approach that ensures comprehensive, clear, and verifiable requirements, stored in a single database, improving design integrity by eliminating inconsistency and redundancy is considered essential.
2. **Adaptability to Externalities**
   Design methods should be characterized by the capability to predict and account for temporal externalities and evolving requirements. This is pivotal for coherent decision-making, ensuring that the design process remains robust despite uncertainties and facilitating better management of trade-offs.
3. **Novelty and Innovation**
   Design methods should promote novelty and innovation without being overly constrained by prior knowledge. This is critical for naval vessel design, where naval architects need to think beyond conventional boundaries, allowing the exploration of unique and creative solutions.
4. **System Architecture Development**
   Design methods should prioritize and support the development of a well-defined system architecture that efficiently addresses the complexities of early naval vessel design. This is essential to facilitate coherence and consistency in system structure, thereby aiding decision-making processes.

2.5. Early Stage Ship Design Approaches and Methods

As far as the design approach is concerned, two primary approaches to conducting ESSD can be identified in the literature. The first is sequential (or point-based) exploration, while the second is concurrent (or set-based) exploration (Duchateau 2016). In ship design, sequential exploration adheres to the traditional design spiral, where one design is manually developed over several iterations. On the other hand, in concurrent exploration, a large number of design solutions are generated and studied in parallel.

ESSD can be tackled by using one or more (combined) design methods that describe these approaches (Scheffers 2021). In the following subsections, current design methods will be discussed. Such techniques are the classic design spiral (Evans 1959), set-based design (Doerry et al. 2014), concurrent design (Syan and Menon 1994), modular design (S. O. Erikstad 2019), system-based design (S. Erikstad and Levander 2012) and Model-Based Systems Engineering (INCOSE 2022). It is important to note that the research will remain at high levels given the comprehensive treatment of these aspects in existing research papers and theses. In this thesis, the focus lies on the existing limitations of those techniques concerning the early-stage design of naval vessels. This approach will allow for fully addressing Research Question 1.

2.5.1. Ship Design Spiral

Since its introduction the design spiral remains the traditional method for ESSD, seen by many as a straightforward representation of the process (Evans 1959; Andrews 2018). The ship design spiral is the most representative example of a point-based approach. It is a design approach in which a single solution is iteratively developed and modified until a design that meets the requirements is achieved (Duchateau 2016; T. McKenney 2013). The insights gained from each iteration guide the subsequent actions and decisions (Duchateau 2016).

Despite the design spiral dominating the maritime industry, serious concerns have been raised. The design spiral addresses the question of “how” to design and not “what” needs to be designed (Evans 1959). With more “what” questions arising in ESSD, this approach is deemed inadequate for managing the dynamic complexities (Poulis 2022). For cargo vessels, the initial parameters required can be appropriately defined through physical attributes like deck area or hold volume. However, for complex ships such as warships, capturing the mission in physical characteristics is less straightforward (Habben Jansen 2020). The relationship between input (design options and variables) and output (solutions and associated performance) introduces complexities, leading to discontinuous behavior that the design spiral struggles to capture (Duchateau 2016; Habben Jansen 2020).

The starting point is crucial for a point-based approach and thus no guarantee of convergence exists (A. Kana 2023d). During the early-stage design of naval vessels with limited prior knowledge, predicting the optimum starting point is a challenging task that also requires experience (Andrews 1998). Furthermore, the fact that the starting point is based on an existing solution poses barriers to novelty, innovation, and creativity. This limits exploration to just a few design options, leaving high-performance design areas unexplored (Duchateau 2016). Besides, the inflexible and strictly sequential nature of the design spiral requires any change to be processed through all stages again, demanding significant effort. This complexity hinders adaptability to external factors and introduces barriers to making trade-offs. Moreover, the design spiral leads to “over-the-wall” approaches where designers complete one phase and “throw” it to the next, leading to a strictly sequential and downstream process (A. Kana 2023d; Sobek, Ward, and J. K. Liker 1999). The “we design it, you build it” mindset hinders communication and collaboration between design phases, resulting in integration issues and substantial rework (Boothroyd, Dewhurst, and Knight 2002). Although several numerical approaches and tools have been employed to aid manual preliminary design, significant effort is still required (Van Oers 2011).

2.5.2. System Based Design

The starting point in the traditional ship design spiral is a set of initial requirements from ship owners, which are often unclear. Modern practices involve collaboration between ship owners and engineers to determine optimal designs, revealing a gap: designers receive requirements but not the operational concept, limiting alternative exploration and locking the design to initial assumptions (Poulis 2022). This leads the design process towards optimizing a single concept rather than exploring and evaluating alternative solutions. The identification of this problem gave birth to System Based Ship Design (SBSD), a modified version of the classic design spiral with greater emphasis on the functional analysis (S. Erikstad and Levander 2012). This approach begins with the identification of a clear mission, outlining tasks, capacity, operating area, and performance factors. The latter changes the order of the design process to prioritize defining systems and functions. As a result, the design spiral is “straightened,” and the number of loops required to reach technically feasible solutions is reduced (S. Erikstad and Levander 2012).

The SBSD method is effective for early design decisions because it serves as a tool that pinpoints the best assumptions before initiating vessel design. The new design is grounded in the most suitable basic ship, ultimately reducing the number of iterations in the upcoming design stages (Vestbøstad 2011). As a result, SBSD has proven successful in developing largely generic ships adhering to a design template with minor variations, as seen in container ships, ferries, and cruise ships (S. Erikstad and Levander 2012). Given the limited need for novelty in these vessel types, the design process becomes mainly a matter of scaling.

Similar to the design spiral, SBSD has some obvious limitations, and repeating all of them adds no value to this section. The most important one is that newly developed designs are constrained by already known designs (Bruinessen, H. Hopman, and Smulders 2013). The previously discussed unique challenges in naval vessel design demand that the development of the next generation of vessels goes beyond the capabilities of existing ones.

2.5.3. Concurrent Design

A design method that advances beyond point-based design is concurrent engineering (CE). Concurrent Design (CD) is the application of CE during the early definition and evaluation phase of a project. This systematic approach emphasizes the seamless integration of various design aspects, considering a holistic perspective (Sobey, Blake, and Shenoi 2013). The team values of cooperation, trust, and sharing are all embodied within CE in a manner where decision-making involves all perspectives in parallel (COMET™ 2023). Key elements include a structured process for parallel design, multidisciplinary team, infrastructure (facilities, software), rigorous methods and techniques for decision making, support, and understanding for the environment (Sobey, Blake, and Shenoi 2013; COMET™ 2023).

The multidisciplinary perspectives in CD contribute to a better understanding of the problem and transparent decision-making, even in the early design phases. Another major improvement, grounded in the collective principles of CE is the improved collaboration which shortens the design timelines and minimizes communication errors (David Singer, Doerry, and Buckley 2009). In ESSD, naval architects concurrently study diverse designs, establish detailed, quantitative requirements, and choose concepts that balance these requirements. This approach addresses challenges such as limited flexibility and unexplored design space. In addition to the recognized benefits in efficiency (quality, speed, cost) CD also aids in improving system awareness among experts (Piirainen, Kolfschoten, and Lukosch 2009). Consequently, CD promotes novelty and innovation without overly constraining solutions to prior knowledge.

This methodology has found application across various industrial sectors, with the aerospace sector standing out as the most prevalent (Finkel et al. 2002; Bandecchi, Melton, and Gardini 2000; Haas and Sinha 2004; Case et al. 2021). A well-established CE team is the European Space Agency's (ESA) Concurrent Design Facility (CDF), employing CE for preliminary feasibility studies on future space missions (Bandecchi, Melton, and Gardini 2000). Regarding the maritime industry, many shipbuilding companies have started to focus on CE during the last three decades (Bennett and Lamb 1996). Currently, Starion is exploring the feasibility of applying CD practices beyond the space industry, with pilot projects for the specification of new naval vessels (Group n.d.). Taking inspiration from the ESA, the COMMIT plans to adopt CD more extensively and thus is launching its own CDF. Furthermore, the adoption of CD can enable NATO to successfully deliver complex projects within both timely and budgetary constraints (Poulis 2022). Also, Feadship established three CDFs after partnering with the ESA to adapt a successful CE approach for the superyacht industry (FEADSHIP 2017). It can be concluded that considerable efforts are underway to extend the application of CD into the maritime industry, with a particular focus on naval ships.

CD proves valuable in identifying the optimal global design of complex products. Concurrency plays an increasingly crucial role due to the growing number of different domain specialists and the rising complexity of system interactions involved in the early-stage design of warships. This substantial increase in information emphasizes the need to properly capture and share this information. However, for this effective coordination to occur, support from proper digital models is essential. One way to achieve this is through MBSE (COMET™ 2023). Besides its benefits, the proper and effective application of CE has proven challenging. While some organizations have succeeded, others have faced difficulties due to substantial organizational changes during the transition to CE principles. However, the progress in contemporary design methods and tools, including Set-Based Design, MBSE and the work outlined in this thesis, hold promise in encouraging the widespread adoption and success of CE practices (T. McKenney 2013).

2.5.4. Set Based Design

The attempt to move away from the traditional design spiral and its limitations during ESSD has led to the use of Set Based Design (SBD) (Ward et al. 1995). SBD is a promising method for defining requirements and solutions that is actively pursued in the US Navy (David Singer, Doerry, and Buckley 2009) and at the Dutch Command Materiel and IT (COMMIT) agency (Oers et al. 2018). It addresses the aforementioned challenge of the early locked-in costs and the limited available knowledge (Habben Jansen 2020). Instead of seeking the optimal solution, sets of parameters are investigated to eliminate infeasible or dominated solutions (T. McKenney 2013). It should be underlined that MBSE, complements the SBD methodology since it can effectively communicate which elements of the design are stable and which are prone to change (Tepper 2010). Together, they contribute to developing a well-structured system architecture.

Separate groups of experts are involved in the design and consistently express their preferences for the studied design values within the design space (T. McKenney 2013; Habben Jansen 2020). The studied sets are kept open for an extended period to conduct a comprehensive analysis of trade-offs. As overlaps between feasible and preferred regions are better understood, the design space is narrowed to focus on these intersections. At every decision point, combinations of the design space where the solution is not likely to reside are excluded (Doerry et al. 2014). More design information is obtained, reducing uncertainty and enabling designers to make more informed decisions to reach an optimum solution. The above-mentioned process is illustrated in Figure 2.2.

The main benefits of SBD, as outlined in various studies (T. McKenney 2013; Thomas A. McKenney, Kemink, and David J. Singer 2011; Sobek 1997; J. Liker et al. 1996; Ward et al. 1995), include a comprehensive understanding of the design space, flexibility in handling uncertainties through the use of set-ranges, the power to monitor design decisions, and the ability to delay decision-making until more

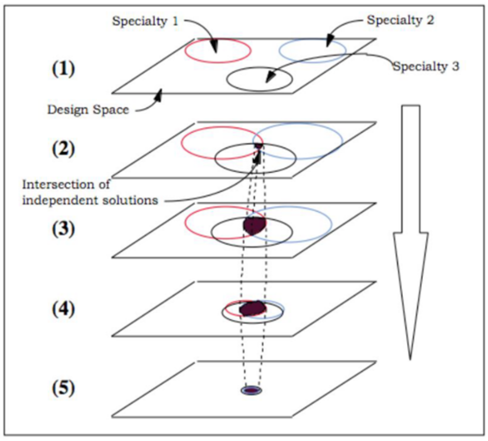


Figure 2.2: Set Based Design method. Source: Bernstein (1998)

information is known and design tradeoffs are better understood. Other studies on SBD during ESSD have underlined its value and benefits against point-based design methods (Parsons, David Singer, and Sauter 2011; David Singer, Doerry, and Buckley 2009; David Jacob Singer 2003; T. McKenney 2013). This design method holds the promise of quickly adapting to evolving requirements, as no major decisions are made during the early design stages (David Singer, Doerry, and Buckley 2009). Subsequently, the committed cost is delayed, providing increased design freedom in the later stages while acquiring problem knowledge earlier in the process (A. Kana 2023d). Moreover, similar to CD, active communication is maintained and design efforts progress concurrently throughout the process (Poulis 2022).

A major challenge in SBD is appropriately defining individual sets. SBD mandates functional groups to negotiate design spaces and to manage their convergence over time (Goodrum and M. J. Singer 2019; Habben Jansen 2020). Although extensive research, little effort has been made to understand the formation of these spaces at the beginning of the SBD process (Goodrum and M. J. Singer 2019). Another key challenge is the decision-making process for set reduction (T. A. McKenney and D. J. Singer 2012; T. McKenney 2013). Changing variable set ranges during design progression alters design relationships. The evolving temporal dynamics pose challenges in fully understanding the implications and order of modifying set ranges. Currently, determining adjustments to set ranges is a heuristic process, typically decided by a chief engineer or design manager. Unlike the extensive databases for automotive design, the U.S. Navy lacks such resources (T. McKenney 2013).

2.5.5. Modular Design

Modularity is a modern systematic approach that involves the division of the ship into blocks, sections, and modules (S. O. Erikstad 2019). From a systems view, modularity can be seen as a strategic approach to managing various forms of complexity, including structural, behavioral, contextual, perceptual, and temporal complexities (Gaspar et al. 2012). Even though modularity is understood and implemented differently across various scientific fields, some common fundamental characteristics can be identified (S. O. Erikstad 2019). Those are the division of a larger system into smaller parts, the principle of self-sufficiency of the individual parts, and the recombination of the parts into multiple end products, according to a set of “rules” given by an overall system architecture.

Different phases of the ship’s life cycle may be driven by various motivations for modularity (Poulis 2022; S. O. Erikstad 2019). For the design phase, modularity is significant for several reasons: It enables both standardization and diversification/customization using a product platform strategy. Moreover, it offers a more efficient design process through a configuration-based design process and supports innovation by exploring the design space through modular re-combinations. Furthermore, it allows for the earlier execution of basic and detailed design stages, concurrent with conceptual and preliminary stages, leading to significant reductions in lead time and resource expenditure.

Modular design enables naval architects to reuse previous designs. This helps to cope with structural complexity by managing hidden interactions within modules and simplifying representations. This simplification is crucial for holistic ship design, given the numerous subsystems and conflicting stakeholder requirements (Papanikolaou 2010). It is particularly suited for scenarios requiring design flexibility to adapt to evolving market dynamics (A. Kana 2023b). It is worth mentioning that the modular architecture employed in the modularity method proves effective primarily for designing product families. However, its efficiency diminishes when it comes to individual products, often resulting in compromised performance due to over-design (Poulis 2022; Fuchs and Golenhofen 2019).

2.5.6. Model-Based Systems Engineering

In this subsection, concepts such as Systems Engineering and Model-Based Systems Engineering (MBSE) are introduced and briefly discussed. It is important to note that a more in-depth analysis of these aspects is presented in Chapter 3. Moreover, at this point, it is crucial to emphasize the versatile role of MBSE. It can be viewed as an approach that enhances several of the aforementioned design methods. This has been highlighted in the SDB and CD subsections. In this context, it becomes evident that MBSE and SBD or CD are not mutually exclusive; rather, they complement each other. While each one has distinct characteristics, they can be used together to benefit the overall system development process. However, the author of this thesis approaches MBSE as a distinct design methodology in Systems Engineering that uses models to manage system complexity. Adopting such an approach allows for a simplified comparison with the other design methods discussed earlier.

The theoretical foundations of MBSE originate from the field of Systems Engineering (SE). SE is a theoretical framework derived from Systems Thinking, It is a generic approach that solves problems by viewing them as part of a larger overall system rather than attempting to reduce them to their individual elements and looking at them in isolation (Kossiakoff et al. 2011; INCOSE 2007).

An essential tool for a successful SE approach in complex systems is an effective system architecture and data model. This systems model serves as a repository in which important characteristics, data, and relationships are decided and documented, spanning from customer requirements to functions, system architecture, as well as validation and verification (C. Kerns 2011). MBSE is what brings together the concept of a model and framework of SE (Shevchenko 2020).

Over the past few decades, MBSE has increasingly emerged as a means to keep track of system complexity (Shevchenko 2020; Vaneman and Carlson 2020). The motivation behind the adoption of MBSE arises from the need to overcome known deficiencies and undesirable practices that negatively affect system architecture and design (Madni and Sievers 2018). Specifically in the initial stages of design, the use of MBSE is becoming more prevalent for establishing a robust system architecture and predicting the emergent behavior and properties of a design (Pearce 2013). This not only accelerates the entire design process by acquiring detailed shape and calculation information early but also facilitates the evaluation of numerous design possibilities and trade-offs with precise accuracy. Moreover, MBSE contributes to an early and better understanding of the design problem (Krasner 2015; Jorgensen 2011). This coupled with a significant decrease in requirement defects, allows to dexterously navigate dynamic dependencies between the problem and its solutions.

The benefits of MBSE in developing system architectures for naval ship design were investigated in research conducted by Tepper (2010). It was concluded that the MBSE model significantly improved communication among stakeholders. Furthermore, the decision-making process was dramatically enhanced due to the inherent traceability of the model. This enables the observation of how a small change can impact the overall design, thus allowing for more precise and robust change assessments and alternative analyses and trade-offs (Tepper 2010). Pearce (2013) ascertained that using MBSE to model a submarine subsystem architecture led to advantages that reflect the objectives of MBSE itself, namely; improved design consistency, precision, traceability, subsystem integration, and design evolution. Another study by C. Kerns (2011) found that MBSE proves to be a powerful tool for managing complexity in ship design, acting as a single-source repository for various design aspects.

2.5.7. Conclusions on Current Ship Design Approaches

The classic design spiral is simple, capable of automation, and semi-flexible due to the numerous refinements. For simple vessels, this method is ideal, but for warships with less straightforward design processes, it is deemed ineffective. Firstly, its adaptability to externalities is limited, while it sets barriers to novelty, innovation, and creativity. The starting point's significance is evident, especially for modern naval vessels, heightening the challenge and making prediction difficult. Moreover, the absence of centralized decision management, coupled with a limited understanding of the initial problem, hinders transparent decision-making. What is more, it provides minimal design product reuse and lacks convergence guarantees.

SBSD streamlines early design decisions and enhances the clarity of the problem-solution relationship by prioritizing functional analysis and reducing the number of loops needed. However, for the same reasons as the design spiral, SBSD cannot effectively account for temporal externalities and evolving requirements. Other limitations include dependence on existing solutions, constraining design within known parameters and restricting innovation.

CD enhances collaboration, accelerates decision-making, and improves system awareness among experts. Thus, it cultivates a better understanding of the problem and transparent decision-making in the early design phases. Moreover, it promotes novelty and better addresses the challenges of limited flexibility to externalities. Nonetheless, it presents difficulties in its application and requires specific collaborative tools and infrastructure.

SBD provides a comprehensive understanding of the design space, utilizes set ranges for flexibility in uncertainty management, monitors design decisions, and allows delayed decision-making. However, challenges include the appropriate definition of individual sets and the heuristic nature of set reduction decisions. Similar to CD, it excels in identifying the global optimum design, delaying decision-making until trade-offs are better understood, and managing evolving as well as conflicting requirements.

The modular design promotes standardization and versatility via a product platform strategy, enhancing design process efficiency. The modularity feature implies innovation and flexibility making the vessel adaptable to evolving markets, regulations, technologies, and customer requirements. Nevertheless, this efficiency diminishes when dealing with individual products rather than product families.

MBSE seems to be better suited for capturing the complexities in early-stage naval vessel design, by placing the model at the center of the design. Furthermore, it prioritizes establishing a well-defined system architecture. This dramatically improves communication, traceability, consistency, and system integration. As a result, a better understanding of the problem can be provided, enabling designers to handle dynamic problem-solution relationships and evolving requirements. In addition, the more precise and robust change assessments, alternative analyses, and trade-offs further improve the transparency of the decision-making process.

It is worth noting that comparing MBSE to CD or SBD might not be entirely fair, considering that MBSE can also support or be combined with such methods. However, for simply highlighting the added value of MBSE, whether combined with another aforementioned method or not, it was deemed acceptable to consider it as a distinct, generalized design ESSD method.

In conclusion, the evaluation of the six aforementioned design methods concerning the four key requirements of section 2.4, can identify MBSE as the most favorable (see Table 2.1). Therefore, MBSE will be explored in detail in the subsequent Chapter (see Chapter 3).

	Better Decision Management	Adaptability to Externalities	Novelty and Innovation	System Architecture Development
Ship Design Spiral	+	+		
System Based Design (SBD)	++	+		
Concurrent Design (CD)	++	++	+++	++
Set Based Design (SBD)	+++	++	++	+
Modular Design	+	++	++	+
Model-Based Systems Engineering (MBSE)	+++	+++	+++	+++

Table 2.1: Evaluation of six design methods with respect to the four key requirements for typical early-stage design methods for a naval vessel

3

Model-Based Systems Engineering

The chapter “Model-Based Systems Engineering” serves as the second half of the literature background upon which this thesis will be based. Within this chapter, the MBSE concepts introduced in Chapter 2 are elaborated in more detail. This chapter begins with an elaboration of the three interconnected concepts that MBSE integrates, namely Systems Thinking, Model, and Systems Engineering. The chapter progresses with an elucidation of MBSE, delving into its core components, namely the modeling tool, modeling method, and modeling language. The chapter then details the broad benefits of MBSE, highlighting key factors that are behind these advantages. These factors are later linked to the ESSD method requirements established in Chapter 2. Following that, the chapter addresses challenges within MBSE. Next, a theoretical comparative analysis among well-known tools, methods, and languages is presented. Consequently, the Research Question 2 (*“What is MBSE, and how can it theoretically aid in managing the increased complexity of early-stage naval ship design?”*), as well as Research Question 3 (*“What are some commonly used combinations of MBSE methods, tools, and languages, and what are their strengths and weaknesses according to the literature?”*) can be fully addressed.

3.1. Systems Engineering

To improve the tangibility and comprehensibility of Model-Based System Engineering (MBSE), it is considered wise to delve into the three intertwined concepts it brings together (Shevchenko 2020): Systems Thinking, Model, and Systems Engineering, as presented in the following subsections.

Systems Thinking

System Engineering derives from the fundamental approach of Systems Thinking (ST). For a broad understanding of the latter term, it is essential to consult Ackoff, Addison, and Carey (2010) which states: “System thinking is a generic approach that looks at relationships (rather than unrelated objects), connectedness, process (rather than structure), the whole (rather than just its parts), the patterns (rather than contents) of a system and context”. ST involves the capacity to identify, understand, and synthesize interactions and interdependencies within a set of components designed for a specific purpose (INCOSE 2022). Therefore, it has been defined both as a skill and as an awareness. Modifications to a systems component can lead to predictable behavior patterns for the entire system. Therefore, following an ST approach —meaning observing systems from a distance and considering boundaries, context, lifecycle, and behavior—is crucial to identifying and addressing issues.

Model

Typically, a model is the formal method of simplifying and portraying a system. It is a streamlined graphical, mathematical, or physical representation that abstracts reality to reduce complexity (Shevchenko 2020). In the world of engineering design, models bridge the conceptualization of a design solution with its realization as an actual system. They represent the system’s entities and their interrelationships, linking them to the proposed solution or existing mechanism that addresses the problem. The system model should feature reduced detail until the system’s structure and behavior are evident, ensuring manageable complexity, with models sufficiently representing the system and being confirmed by the system (Long and Scott 2012).

Systems Engineering

The discipline of SE has emerged significantly in response to the ever-increasing complexity of systems (INCOSE 2014). SE can be classified as an interdisciplinary approach, containing both management and technical processes, aiming to guide the realization of successful systems (Kossiakoff et al. 2011). It directs the creation of a total system design solution that balances cost, schedule, performance, and risk. It also includes the development and tracking of technical information for decision-making, as well as verification that technical solutions satisfy customer requirements. These key responsibilities affirm SE's value in managing complex and technologically challenging problems (Lightsey 2001).

The Systems Engineering Process is a top-down, comprehensive, iterative, and convergent problem-solving process, applied sequentially through all stages of development. It aims to transform needs and requirements into a set of system product and process descriptions, to generate information for decision-makers, and to provide input for the next level of development (Lightsey 2001). The growing complexity of modern combatants has transformed the "Ship Design Process" into a SE Process, with ship designers filling both roles and merging the two processes (Tepper 2010). Indeed, SE is a proven design approach regularly adopted in naval ship design and defense acquisition projects (Jansen et al. 2020; Morris 2019; Lightsey 2001; Andrews 2018).

Various SE approaches emerge when considering the sequence of processes and methodologies used in the execution of the design, integration, and testing of a system (Kossiakoff et al. 2011). In his research, Beery (2016) included clear, domain-neutral representations of each process model to distinguish the mechanisms within each model that most effectively address the overall goals of an SE process. This comparison was made among four widely used SE process models: the waterfall model, the spiral model, the incremental model, and the popular "Vee" model, presented in Figure 3.1

At this point, it is deemed valuable to explain a bit further the "Vee" model in terms of ESSD activities. This model best highlights that SE is at the intersection between "how" and "what" to design. This offers a view of life cycle development by illustrating explicit relationships between requirements and systems definition and the developed and validated product (Habben Jansen 2020). It describes the connections between missions at a high level of abstraction (top left of the diagram in red), extending down to the solution, which is characterized by a specific and physical nature (bottom left of the diagram in blue) (Jansen et al. 2020). The steps in this downward route are aligned with the perspectives outlined in the definition of system architecture for this thesis.

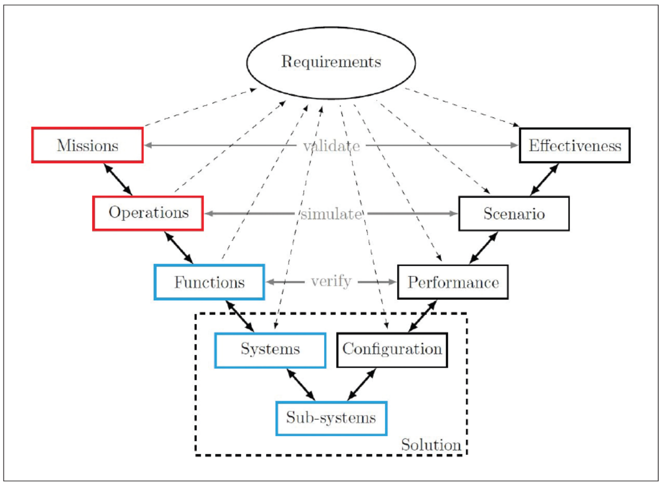


Figure 3.1: Systems Engineering "Vee" Diagram for naval ship design. Source: Jansen et al. (2020)

The term "system life cycle" typically denotes the phased development of a new system from concept to disposal. System life cycle models have evolved significantly over the past two decades. Nevertheless, no universally accepted life cycle model suits every possible situation. Fortunately, each life cycle model divides the system life into fundamental steps that separate major decision milestones. As a result, the SE life cycle model can be decomposed into three stages, illustrated in Figure 3.2. The first two encompass the developmental part, and the third involves the post-development part. Within each stage, further phases are contained (Kossiakoff et al. 2011).

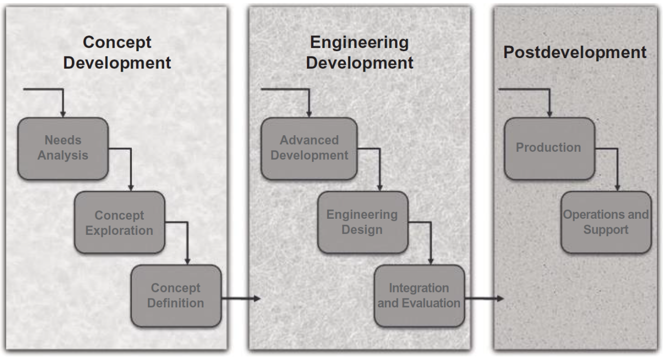


Figure 3.2: System life cycle model depicting SE stages and phases. Source: Kossiakoff et al. (2011)

The first stage is concept development and consists of three phases, which together they form ESSD in this thesis: needs analysis, concept exploration, and concept definition. The key objectives of this stage include (Kossiakoff et al. 2011):

- Establishing the need for a new system that is both technically and economically feasible
- Exploring potential system concepts, formulating and validating a set of system performance requirements
- Selecting the most attractive system concept, defining its functional characteristics, and developing a detailed plan for subsequent life cycle stages
- Developing technology required by the chosen concept and validating its capability to meet requirements

The second and third stages focus on engineering development and post-development, respectively. However, a deeper analysis of these stages does not contribute additional value to this thesis.

The future of SE is predominantly model-based (INCOSE 2022). Modern systems are inherently multidisciplinary, requiring compliance with growing customer requirements and marked by numerous external constraints. Managing the ever-increasing intricacy has become critical to the success of current and future projects, even with a rigorous application of SE processes. As a result of this rising system complexity, conventional Document-Based Systems Engineering (DBSE) is gradually being replaced by model-centric engineering. This transition aims to take advantage of the use of models, which are more expensive but less prone to ambiguity compared to traditional documents (Baron et al. 2023).

3.2. Model-Based System Engineering

The shift from DBSE to Model-Based Systems Engineering (MBSE) is a natural progression in the evolution of SE, influenced by the extensive role of computers and digital modeling in modern society (Kooij 2022; Poulis 2022). MBSE is essentially a thought process. It offers the framework that enables the SE team to be effective and consistent right from the initiation of any project. Simultaneously, it is flexible enough to allow adjustments in the thought process in response to specific constraints and circumstances present in the problem (Long and Scott 2012).

In the last two decades, there has been a growing discourse on MBSE (Weilkiens, Vinarcik, and Fischer 2023). INCOSE provides a comprehensive definition for MBSE describing it 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” (INCOSE 2007). This definition dates back to the early stages when MBSE was emerging as a viable improvement to document-intensive SE. A more recent supplementary definition, provided by Weilkiens (2022), states: “MBSE is SE with the formal application of models to make engineering information accessible to machines to support the stakeholders”. The latter definition explicitly confirms that MBSE is a part of SE rather than constituting a new and competing discipline. Additionally, it highlights the essential purpose of the models. While models provide clear representations and help work in a structured way, these objectives can also be achieved through document-based approaches.

Document-intensive SE, under the right conditions, can be effective. However, the disconnected artifacts used in SE can drift out of consistency and completeness. Critical details may be lost, documents can become outdated, communication barriers arise, and inconsistencies found late in development necessitate rework and cause delays. All of these factors result in a loss of effectiveness and momentum in the SE effort.

On the contrary, the efficiency and effectiveness achieved with model-based approaches undoubtedly surpass those of document-based methods (Weilkiens, Vinarcik, and Fischer 2023). In simpler terms, MBSE differentiates from DBSE by putting a greater emphasis on the use of computer-based models to support SE processes. It is still SE, although executed using different tools (Weilkiens, Vinarcik, and Fischer 2023). The model is the only source of truth and reflects the state of system development (Madni and Sievers 2018). While documents remain valuable, they should primarily be derived and exported from models to present perspectives of model information (Weilkiens, Vinarcik, and Fischer 2023).

A distinction between using models in SE and MBSE is important to be made at this point. In SE, traditional models often fall victim to becoming “zombie models”. This term refers to a model that is full of isolated diagrams that are domain-specific but lack interconnectedness and dependencies among their elements. This results in a high risk of misalignment and inconsistency. MBSE, on the other hand, addresses these shortcomings by emphasizing the creation of integrated models that represent the entire system (Shevchenko 2020).

The combination of computer-based models and SE processes results in defining the ship design process as a detailed system architecture. The strength of the system architecture lies in its ability to act as a single source of truth, that includes all elements contributing to the ship’s architecture. This single source of truth, or in practice a single source repository, encompasses data, guidance, design characteristics, processes, cost, risk, and effectiveness, and captures all relationships between these aspects. This attribute makes it a powerful tool in handling the complexity encountered in the design process of naval vessels (L. C. Kerns, Alan Brown, and Woodward 2012; Tepper 2010).

3.2.1. Key Principles of MBSE

There is considerable debate within the SE community regarding the key components of MBSE (Jenkins 2021). To address the question, “How do you implement MBSE and what do you need to know?” Delligatti, Soley, and Steiner (2014) introduced the popular three-pillar concept, in which MBSE can be organized into three main pillars. These pillars include a modeling language, a modeling method, and a modeling tool. The subsequent paragraphs provide a more in-depth exploration of these three pillars.

Modeling Language

Modeling involves a distinct language, separate from the one used in everyday communication. A modeling language is a semi-formal language that specifies the types of elements that can be incorporated into the model and the permissible relationships among them. In the case of a graphical modeling language, it specifies the set of notations that can be used to display the elements and relationships on diagrams (Delligatti, Soley, and Steiner 2014).

A famous language is the Systems Modeling Language (SysML). However, SysML is not the only modeling language. Depending on the engineering domain and the type of system, other modeling languages might be more appropriate. Like SysML, some of those languages are graphical modeling languages with examples like UML, UPDM, BPMN, MARTE, SoaML, and IDEFx. On the contrary, others are text modeling languages, including Verilog and Modelica. In essence, each modeling language serves as a standardized communication medium, employing defined rules to give a clear meaning to the model’s elements and relationships (Delligatti, Soley, and Steiner 2014; Kooij 2022).

Modeling Method

As previously explained, a modeling language establishes a “grammar”, containing rules that determine the model’s well-formedness. This is done without prescribing “how” or “when” to use the language to create a model. For the latter, a modeling method is required, acting as a sort of roadmap (Delligatti, Soley, and Steiner 2014).

A modeling method can be defined as a documented set of design tasks executed by a modeling team for the development of a system model. This guidance guarantees a consistent approach among team members and ensures joint efforts toward a common goal. The modeling method is necessary to prevent large variations in the scope, depth, and accuracy introduced by each team member while constructing the model (Delligatti, Soley, and Steiner 2014).

Examples of several known modeling methods include: Architecture Analysis Design Integrated Approach (Arcadia), INCOSE Object-Oriented Systems Engineering Method (OOSEM), Weilkiens System Modeling (SYSMOD) method, IBM Telelogic Harmony-SE, Vitech method, and Object-Process Method (OPM) (Delligatti, Soley, and Steiner 2014; Kooij 2022).

These modeling methods cover various stages of the SE life cycle. This means that not every step prescribed applies to every project. The selected modeling method must be tailored to meet the project's specific needs and objectives. In situations where existing methods are not suitable, then a custom modeling method can be created (Delligatti, Soley, and Steiner 2014).

Modeling Tool

Modeling tools are software-based and transfer information from documents to digital modeling environment (A. Kana 2023a). They are a special category of tools designed to comply with the rules of one or more modeling languages, thus enabling the creation of well-structured models within those languages. This model consists of elements and relationships between them, optionally accompanied by a set of diagrams that serve as views of the underlying model. Unlike diagramming tools, a modeling tool is differentiated by the presence of the model that underlies those diagrams, ensuring automatic consistency. A distinctive capability exclusive to this tooling class is that by modifying an element on one diagram within a modeling tool, this instantly updates the rest of the diagrams displaying the same element (Delligatti, Soley, and Steiner 2014).

It is worth mentioning that a modeling tool represents a vendor's interpretation of that language specification. Multiple commercial tool vendors and consortiums offer tools for several languages, which vary in cost, capability, and compliance with the modeling language specifications. Examples of modeling tool packages include CDP4-COMET, Capella, MagicDraw, Cameo Enterprise Architecture, Cameo Systems Modeler, CORE, and Rhapsody (Delligatti, Soley, and Steiner 2014; Kooij 2022).

Deciding on the appropriate tool is a critical part of the MBSE effort and should take into account the needs and cost constraints of the project. Another important factor to consider is securing compliance with other tools for data exchange. This will prevent the risk of vendor lock-in if project requirements change in the future (Delligatti, Soley, and Steiner 2014).

3.2.2. Benefits of MBSE

Several publications have explored the expected benefits of employing MBSE in the early design phases for several industries. However, in the context of ship design, the available publications are notably fewer. While certain studies in this sector remain primarily theoretical in nature (Henderson and Salado 2021; Weilkiens, Vinarcik, and Fischer 2023; Madni and Sievers 2018; Topper and Horner 2013; Hause 2011; Long and Scott 2012), others have effectively showcased practical advantages (Carroll and Malins 2016; Morris 2019; Baron et al. 2023; Kooij 2022; Tepper 2010; Poulis 2022; C. Kerns 2011; Pearce 2013; Beery 2016). In this section, the focus is on presenting the anticipated benefits identified in these publications, by giving a greater emphasis on ship design studies.

Single Source of Truth

The value of MBSE lies in the central repository where all system-related information is stored and configuration-managed (Carroll and Malins 2016). This provides consistency, implying that the system specification is in line with the modeling architecture and that the meaning of concepts is identical throughout the entire model. This authoritative single source of truth is accepted by all stakeholders, enhancing consistency. The single repository also enables the interconnection of model elements, effective information retrieval, system reasoning, requirements traceability, automatic propagation of design changes, and error identification. It can be concluded that the concept of the single source of truth forms the basis for several of the following benefits.

Attention to System Architecture

As it has been underlined, the lack of attention to system architecture and its impact on design can lead to downstream integration challenges. MBSE addresses known deficiencies and undesirable methods that negatively influence system architecture and design, thus enabling the structuring of a well-defined architecture early in the design process (Madni and Sievers 2018; Tepper 2010; Pearce 2013; C. Kerns

2011). This allows engineers to explore a much broader set of design options within the same amount of time and resources as conventional methods (Carroll and Malins 2016).

Decision-Making
Model-driven approaches and the early development of systems architecture can assist in the decision-making process. Architectural decisions are often made purely on a technical basis without a complete review of their implications for the entire system. MBSE is a way to keep track of decisions and rationale in a central repository. Traceability is created by relating elements in the model to the requirements they derive from. It allows for more accurate and rapid trade-off analysis, change assessment, and the ability to easily track changes and design decisions. The designer can study how a minor modification in one aspect of the design can drastically affect the entire system, and thus conclude in better decisions (Tepper 2010).

Model Quality
MBSE can improve model quality through requirement traceability, design integrity, and model consistency. The interconnected information guarantees rigorous traceability across information, generated data, and decisions (Madni and Sievers 2018; Kooij 2022; Beery 2016). This inherent traceability also results in requirements that are more complete, unambiguous, and verifiable. The centralized repository of model-centric approaches is significantly more effective than managing requirements using simplistic methods like checklists or ad hoc approaches such as disconnected databases (Madni and Sievers 2018).

Collaboration and Communication
MBSE helps minimize ambiguity, promotes collaboration, and ensures consistency of thought and expression across the entire program team (COMET™ 2023). The ability to communicate clearly comes from a common system design language that can bridge communication gaps among experts from different backgrounds (Carroll and Malins 2016; Hause 2011; Beery 2016). This translates to better efficiency, saving time and effort, while consistently achieving equal or superior quality outcomes. Moreover, any identified model gaps can be collaboratively addressed by the engineering team (Madni and Sievers 2018).

Risk Awareness
MBSE enhances earlier risk awareness. The rationale behind this lies in the ability to employ executable models for both early and continuous validation and verification (V&V) of systems (Tepper 2010; A. Kana 2023a). Furthermore, the model allows for rigorous trade-offs and accurate risk and cost estimations.

Incremental Generation, Scalability and Reusability
MBSE enables the incremental generation of architectures, starting with simple components, and gradually building more complex subsystems from the realized components (Topper and Horner 2013; Kooij 2022). The models contain all pertinent information, such as subsystem interrelationships, as well as tailored views and can be easily linked to assumptions, behaviors, requirements, design, and analyses in an easily reachable format (Pearce 2013). This facilitates flexibility and design evolution with effortless and consistent adaptation of the model across associated elements, even when impacting a large number of components. It also promotes both scalability and the reusability of component models within the architecture (Madni and Sievers 2018).

It is apparent that a set of distinct factors consistently emerges from the aforementioned benefits. Therefore, four key factors that form the foundation for these advantages have been identified. The interrelationship between these factors and the general benefits described above is detailed in Table 3.1.

	Consistency	Flexibility	Traceability	Trade-offs
Single source of truth	✓		✓	
Attention to System Architecture	✓	✓		✓
Decision-Making	✓		✓	✓
Model Quality	✓		✓	
Collaboration and Communication	✓			
Risk Awareness			✓	✓
Incremental Generation, Scalability and Reusability	✓	✓		✓

Table 3.1: Relation between MBSE general benefits (rows) and key factors (columns)

3.2.3. Challenges of MBSE

According to subsection 3.2.2, it can be conclusively asserted that MBSE can offer major benefits to the design of complex systems. Nonetheless, numerous limitations and obstacles are hampering its widespread adoption.

When compared to other sectors, traditionally the maritime industry, and specifically ship design, has been perceived as less open towards embracing innovation (Poulis 2022; Droste, Bedert, and Audenaert 2024; Perunovic and Vidic-Perunovic 2011). The reluctance to embrace innovative efforts results from high capital costs, as well as the complicated design regulations prevalent in the maritime industry (Poulis 2022). This establishes obstacles to the adoption of model-centric approaches. However, the resistance of the maritime industry is not the sole obstacle to the expansion of MBSE practices. Inherent challenges within MBSE itself can also present impediments, which are detailed below:

Resource Demands
Introducing and performing MBSE necessitates time and effort. A successful effort demands skilled practitioners, and MBSE efforts require proficient craftsmen in the field of modeling (Weilkiens, Vinarcik, and Fischer 2023). Model-centric strategies typically involve high initial investment for the acquisition of the tools and training of the tool users (Haskins 2011; Weilkiens, Vinarcik, and Fischer 2023).

Tool Integration
The complexity of selecting and integrating tools is stated as a major disadvantage of MBSE. Different tools can use different modeling languages and formats. Linking tools from different suppliers may be difficult in cases where standardized definitions or modeling elements are not used (Maestro Redondo, González Rodríguez, and Kolfschoten 2024). Therefore, ensuring future compatibility amongst tools is a critical aspect (Haskins 2011). There is a growing demand to invest in new tools and ensure their compatibility with existing ones for seamless integration (Fernandes 2023).

Adaptation Resistance
Adaptation resistance within the project or company is another major challenge for MBSE (Gerene 2024; A. Kana 2023a). Transitioning to MBSE requires a mindset shift from traditional document-based processes, which introduces challenges for organizations unfamiliar with modeling languages due to limited training or expertise. The non-recognition of MBSE's value by some stakeholders can also impede its acceptance (Droste, Bedert, and Audenaert 2024). Organizations should invest in training to ensure the benefits of MBSE are well understood (Fernandes 2023; Maestro Redondo, González Rodríguez, and Kolfschoten 2024).

Perceived vs Actual MBSE Implementation
Another common challenge is the confusion about what constitutes MBSE practices. This misconception arises from confusing MBSE with conventional modeling or traditional SE approaches. Engineers may think they are implementing MBSE, but in reality, they are not(Gerene 2024). They are usually involved in modeling activities while also mentally following SE steps (Droste, Bedert, and Audenaert 2024). However, it is crucial to understand that MBSE is not just about creating models and following an SE approach; it is about embracing a holistic model-centric philosophy where the models act as a single source of truth.

Modelling Tool Development
Current modeling tools are being pushed to their limits, implying that they might struggle to efficiently handle the complexity associated with MBSE practices. Consequently, slow tool development becomes a dealbreaker for the implementation of MBSE (A. Kana 2023a). Furthermore, lagging technological advancements and product immaturity serve as additional technical root causes for the low adoption of MBSE (Haskins 2011).

Research Needs for MBSE Applications
Further research is needed to explore the value of MBSE across a broader range of industries, with the maritime industry standing out as one that could benefit the most from its implementation. Literature regarding the application of MBSE practices in ship design indicates a limited presence, concentrated in a few research papers that predominantly focus on naval vessels (Poulis 2022). Tepper (2010) suggested that additional research and pilot initiatives are recommended to quantify the anticipated benefits in terms of schedule, cost, and risk.

The author of this thesis realized the need for further research from personal experience, through their active participation in the "Dutch Naval Design: Mission & MBSE" (DND-MMBSE) pilot project. This collaborative project involved key partners such as Starion, Dutch MOD, DAMEN Naval and ADSE. The main goal of the project was to gain practical experience with MBSE and its associated languages, methods, and tools within a relatively short timeframe. This project attained practical significance through its emphasis on a case study on the chilled water system of a Dutch naval frigate. Engaging in extensive discussions with naval architects and other industry experts within the Dutch maritime sector, the author witnessed the growing need for further research and empirical evidence that can prove the value of MBSE in warship design (Droste, Bedert, and Audenaert 2024).

Modelling Tool Limitations
No tool is flawless for every application. Some tools might not be flexible enough because a predefined methodology might be linked to the tool. Moreover, the capabilities of various tools may vary depending on the stage of design. For example, some tools may prove more valuable in the needs analysis phase, while others may excel in the concept exploration phase. For such tools, there is a need to develop extensions (specific add-ons) and/or to interconnect them with other tools (Baron et al. 2023; Maestro Redondo, González Rodríguez, and Kolfschoten 2024).

3.2.4. Mapping MBSE Expected Benefits to Key Requirements in Early-Stage Naval Vessel Design

To explore the added value of MBSE in early-stage naval design, a connection needs to be established between the four expected benefits (referred to as factors) identified in subsection 3.2.2 and the four key requirements for typical ESSD methods determined in subsection 2.4.1. This connection is presented in Table 3.2 and the rationale behind it is explained in the subsequent paragraphs.

	Better Decision Management	Adaptability to Externalities	Novelty and Innovation	System Architecture Development
Consistency	X			X
Flexibility		X	X	
Traceability	X			
Trade-offs		X		

Table 3.2: Mapping MBSE factors to key requirements in ESSD methods

For better decision management two factors have been recognized as the most representative. Firstly, by assuring that the system specifications are consistent throughout the entire design process, the decision-making process can become more transparent. Consistency through common modeling syntax and architecture, as well as identical values and meanings of concepts throughout the model, diminishes ambiguity and favors better decision management. Secondly, better decision management can result from maintaining traceability in the model. Traceability provides a clear understanding of the relationships between design decisions and the system's requirements. Consequently, the verification of requirements and the transparency of decision-making are increased.

The adaptability to externalities can be related to two factors. Firstly, flexibility is of significant importance. Design methods that are characterized by flexibility allow for effortless adaption to externalities and evolving requirements. Quick and cost-effective adjustments to areas prone to change enable the system to remain responsive and robust in the face of uncertainties. Secondly, obtaining a better and earlier understanding of design trade-offs is crucial for adapting to external factors. Insight into alternative analyses and trade-offs helps make adjustments to the design based on changing external factors, ensuring the inclusion of adaptability features.

The need for novelty and innovation is directly connected to the aspect of flexibility. Flexible system specifications that can be adjusted fast and with little effort support naval architects in their effort to explore unique and creative solutions beyond conventional boundaries.

One factor has been deemed relevant for prioritizing the development of a well-defined system architecture. More specifically, consistency in system specification is considered necessary. The high quality of coherence and uniformity in the system structure forms the foundation of a well-established system architecture.

Note that this mapping serves a dual purpose. First, as shown, it establishes the connection between the theoretical benefits of MBSE and the requirements of early-stage warship design methods. Secondly, it will be used later to indicate what needs to be demonstrated and why during the modeling effort, to practically prove the value of MBSE in early-stage warship design.

3.3. Comparison of MBSE tools-methods-languages

The following subsections provide a summary review of well-known modeling methods, languages, and tools. The goal of this section is to familiarize the reader with combinations of the three pillars of MBSE as applied in industry. By highlighting the advantages and disadvantages inherent in these combinations, valuable insights are gathered for thoughtful consideration both before and during the modeling effort. This will also lay the foundation for the selection of tools, methodologies, and languages in a later stage of this thesis.

3.3.1. Concurrent Design - CDP4-COMET

CDP4-COMET is a concurrent MBSE tool developed by Starion. It is designed to provide all the required integrated design and modeling capabilities to support the multidisciplinary team in their collaborative work. This empowers teams to craft customer-centric solutions, all while considering the complete life cycle perspective of the system. CDP4-COMET enables stakeholders to effortlessly adjust design parameters enabling them to verify and validate the overall design against system requirements and performance metrics (COMET™ 2023).

CDP4-COMET supports MBSE using the CD method. The underlying data model is an implementation of CD concepts agreed upon and accepted by the CD Community. These concepts are modeled and described in ECSS-E-TM-10-25, which is a technical memorandum that defines recommendations for model-based data exchange during the early phases of engineering design. This standard describes a meta-model of a system composed of elements characterized by parameters (COMET™ 2023).

A CDP4-COMET model is based on a set of requirements and a product design. The product is decomposed to the appropriate level of abstraction to enable verification and validation of the design against the requirements. The product is detailed in a product tree where elements of the system and associated parameters represent factors enabling analysis of key challenges in the design. The requirements are outlined in a requirements specification. Detailed documentation of the system's requirements and the automatic requirements verification capabilities, facilitates a rigorous comparison between the solution and its requirements, revealing early analyses of key trade-offs and challenges in system performance. Moreover, to create a product design that meets the design challenges, CDP4-COMET offers the possibility of creating multiple Options within the same project. These Options can be defined as design alternatives and can be compared to alternatives in other Options. In this way, the model serves as a single point of truth for the team.

One of the main benefits of CDP4-COMET is its inherent strength in supporting the multidisciplinary team in their collaborative work. It embodies values of cooperation, trust, and sharing in such a way that decision-making is by consensus, involving all perspectives in parallel. This MBSE tool enables all key stakeholders to access the model by defining roles (permissions) and, consequently, take ownership of information within their respective domains of expertise. With this clear sense of ownership comes responsibility within the design process. The intention is to explicitly define these responsibilities, making it easier to identify issues or ambiguities in the design work (COMET™ 2023).

Furthermore, models or individual building blocks can be saved as templates to boost reusability. Another critical aspect to highlight is that CDP4-COMET is relatively easy to navigate and learn, even for users with no prior experience in MBSE tools. The author of this thesis can attest to this based on personal experience. Last but not least, CDP4-COMET is an open-source platform that does not necessitate the purchase of commercial licenses (Maestro Redondo, González Rodríguez, and Kolfschoten 2024; COMET™ 2023; Gerene 2024).

On the other hand, the way that a system is built, using product trees instead of diagrams, may be perceived by some as a departure from conventional practices. While not necessarily a weakness, the absence of a diagram-based approach in CDP4-COMET may create the perception that the tool has fewer visualization capabilities. However, concurrent diagram editing is difficult and requires a control mechanism with predefined permissions, a challenge currently being addressed by the CDP4-COMET development team. In addition, it is worth mentioning that CDP4-COMET is a relatively recent development, and as such, it may lack maturity and broader acceptance beyond the space industry. This is also reflected in the limited availability of templates for ship systems, general ship breakdown structures, and case studies in the literature. Finally, engaging in model editing and construction, as opposed to simply observing their structure, demands high-quality large-size monitors to deal with information density (Maestro Redondo, González Rodríguez, and Kolfschoten 2024).

3.3.2. Arcadia - Capella

Capella is a comprehensive, extensible, and field-proven MBSE tool and method to support the system architecture development (Capella n.d.; Voirin 2018). This software allows the creation of different elements, including diagrams, activities, interactions, actors, and capacities, tailored to each stage of the method. It is in complete accordance with the Arcadia method and uses a unified modeling language (Baron et al. 2023; Roques 2018). Arcadia promotes a viewpoint-driven approach and emphasizes a clear distinction between need and solution (Capella n.d.).

Capella supports the definition of needs and solutions through a multi-layered structure. It begins with the elicitation of the stakeholders' needs and guides the design process until the exploration of the different technological and architectural possibilities within the solution domain is achieved (Baron et al. 2023). The views across these layers are consistently aligned with each other. Although the model can accommodate a large variety of views, their exact necessity may differ from one project to another. Consequently, these views are created throughout the projects based on their need-to-have significance. What is more, Capella offers a versatile approach adaptable to different scenarios. It primarily follows a top-down Arcadia process, beginning with operational analysis and extending to physical architecture for the establishment of a system design. On the other hand, it supports a bottom-up approach for reverse engineering when developing systems based on existing components or systems.(Baron et al. 2023; Roques 2018).

Arcadia is divided into four main levels: Operational, Functional, Logical, and Physical. These are illustrated in Figure 3.3 (Capella n.d.). Capella's strengths lie in the use of a consistent model to provide architectures on all four levels. Each level has its objectives and progressively delves deeper into the system design (Voirin 2018). It was stated in chapter 1 that the system architecture definition in this thesis involves a conceptual model, the structure of which aligns perfectly with the levels outlined in the Arcadia method. The introduction also offered a preview of the focus and objectives at each level. Therefore, the discussion of each layer will stay at a high level, incorporating examples of model diagrams.

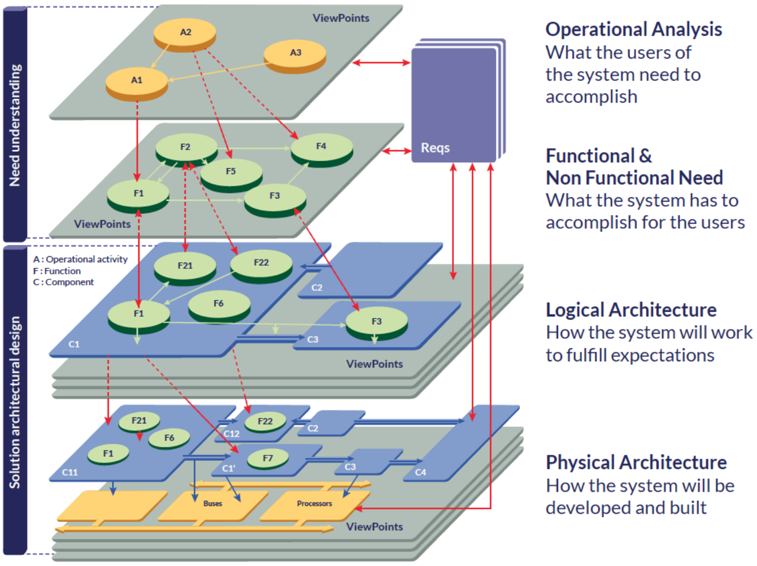


Figure 3.3: Arcadia Engineering Levels. Source: Capella (n.d.)

Operational Analysis

This perspective focuses on defining the problem by capturing and analyzing the customer needs and goals, expected missions, and activities, far beyond system requirements. Examples of diagrams in this layer include:

- Operational Architecture Diagram (OAB): shows which actions (operations) the actors want to perform, and which operations are needed by the system to support these.
- Operational Capabilities Diagram (OCB): matches the operational capabilities to the stakeholders, showing the ultimate purpose of the system.

System Analysis
This perspective focuses on the system itself, to define how it can satisfy the former operational need. Examples of diagrams in this layer include:

- System Architecture Diagram (SAB): shows (high-level) functions and can be used as an entry into the model, to see what the system does and how functions are related.
- Mission / Capability Diagram (MB/MCB): shows the mission and related desired capabilities of the system, based on the operational capabilities.
- System Functional Chain Diagram (SFCD): shows how the system capabilities are fulfilled by system functions.

Logical Architecture
This perspective focuses on building a coarse-grained component breakdown of the system carrying the most important engineering decisions. Examples of diagrams in this layer include:

- Logical Architecture Diagram (LAB): adds more details to the system which show how the system functions will be fulfilled.
- Logical Component Breakdown Diagram (LCBD): shows how the architecture of the system is created based on the topology of the system; it ensures the system functions and the “executing” components will fit in the environment.
- Logical Functional Chain Diagram (LFCD): shows how the system functions are translated to logical functions allocated to logical components within the system.

Physical Architecture
This perspective shares the same objective as constructing the logical architecture but focuses on defining the “final” architecture of the system at this level of engineering. Examples of diagrams in this layer include:

- Physical Architecture Diagram (PAB): shows the physical architecture of the realized system, with all components included in the model.
- Physical Functional Breakdown Diagram (PFBD): shows how the physical functions are defined as a further decomposition of logical functions.
- Physical Functional Chain Diagram (PFCD): shows how the system fulfills the required functions, with allocations to physical components shown in the physical architecture diagram.

A major benefit is the inherent flexibility of Capella, making it adaptable to diverse companies and projects without being tied to a specific standard. Thus, Capella becomes capable of modeling systems across various fields, including automotive, aeronautics, nuclear, space, and telecommunications. Moreover, the feature of adaptability extends to projects at various levels, from systems of systems, subsystems, low-level components, or software. The choice depends on specific project requirements, whether they involve operational analysis, functional analysis, logical architecture, and/or physical architecture (Baron et al. 2023). Another key benefit is the tool’s dedication to modeling and analyzing the functional aspects of a system. The functional chains created in Capella contribute to defining and understanding the logical flow of functions within a system, ensuring that the system aligns with its intended functionality (Maestro Redondo, González Rodríguez, and Kolfschoten 2024; Gerene 2024). Also, Capella is simple and beginner-friendly. Even though it contains multiple diagrams, those unfamiliar with the software can be easily guided by the tool’s methodology (Arcadia). Therefore, from this perspective, the embedded methodology functions as an advantage over SysML tools, which might be more open in use but less straightforward (Baron et al. 2023). An additional benefit is that Capella is open-source. The free-of-charge aspect promotes widespread use among systems engineers, encouraging the tool’s evolution and improvement with new functionalities and add-ons (Maestro Redondo, González Rodríguez, and Kolfschoten 2024). Moreover, the ability to modify the tool's source code makes it possible to adapt it to a company, team, or a specific project (Baron et al. 2023). Finally, a great benefit of Capella is visualization, which strengthens the communication of concepts. The homogeneous colors reinforce the readability and coherence of diagrams (Baron et al. 2023).

On the other hand, some argue that Capella might lack the desired flexibility because a predefined methodology is linked to the tool. Another limitation in Capella is the inability to allocate a system function directly to a physical component without going through the logical layer. Building the logical layer, which may be perceived as not adding value in certain projects, can result in unnecessary additional work (Baron et al. 2023). Furthermore, the lack of version-saving capability poses challenges for reverting in case of errors and may be problematic when reversibility of decisions is necessary (Baron et al. 2023). Finally, Capella features various add-ons, with many of them being commercial. For instance, "Team for Capella" is a collaborative commercial add-on to simply share your models on the same server. Therefore, for a company with multidisciplinary teams starting to invest in MBSE, adopting Capella may not be the most optimal option (Maestro Redondo, González Rodríguez, and Kolfschoten 2024).

3.3.3. SysML

SysML or "Systems Modeling Language" is a general-purpose modeling language for SE applications. It is designed to support engineers in specifying, designing, analyzing, verifying, and validating a broad range of systems and systems-of-systems (Friedenthal, Moore, and Steiner 2014). SysML can be defined as an extension of a subset of the Unified Modeling Language (UML), a standard for model-based software development (Baron et al. 2023; Delligatti, Soley, and Steiner 2014). SysML's development has enhanced the embrace of system modeling to all SE domains, extending beyond the initial focus on software systems (Tepper 2010). Since its inception, SysML has gained broader acceptance and is now being applied in various domains beyond aerospace and defense. As the SysML usage continues to grow, there is an ongoing development effort to create an updated version of the language, known as SysML V2, which will allow usage of the methodology across multiple sectors in a more natural way (Bui Long 2024).

SysML is a graphical language. Its vocabulary consists of graphical notations with specific meanings. The key purpose of SysML is the visualization and communication of a system's design among related stakeholders (Delligatti, Soley, and Steiner 2014; Baron et al. 2023). SysML models are based on a family of nine types of diagrams as shown in Figure 3.4. Activity Diagrams, State Machine Diagrams, Sequence Diagrams, and Use Case Diagrams are types of behavior diagrams. On the other hand, Block Definition Diagrams, Package Diagrams, and Internal Block Diagrams fall under structure diagrams. It is noted that Parametric Diagrams are a type of Internal Block Diagrams, which transitively makes them a type of Structure Diagrams. Finally, Requirements Diagrams constitute a distinct category on their own, yet they remain a useful addition to this family of SysML diagrams (Delligatti, Soley, and Steiner 2014).

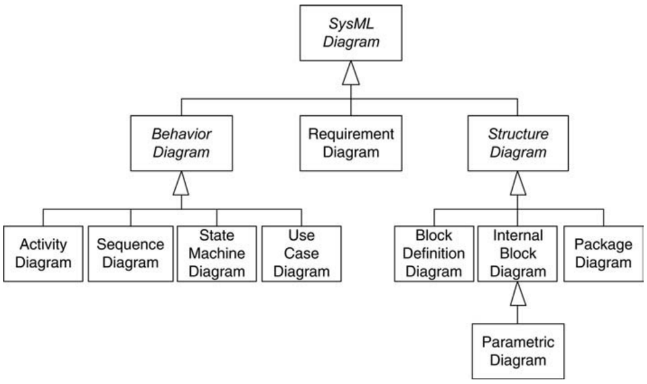


Figure 3.4: SysML diagram taxonomy. Source:Delligatti, Soley, and Steiner (2014)

SysML diagrams support the visual modeling of systems at different levels of abstraction. Logical architecture in SysML is often represented using Block Definition Diagrams and Internal Block Diagrams. Concerning functional architecture, Activity Diagrams, and State Machine Diagrams can effectively capture and portray the systems' behavior. Physical architecture in SysML is often captured using Block Definition

Diagrams and Internal Block Diagrams as well but with a focus on physical components and their interactions (Maestro Redondo, González Rodríguez, and Kolfschoten 2024). Specifically, the family of Behavioral Diagrams represents a significant advantage that enables SysML tools to excel in modeling system behavior compared to other tools. This capability allows for a focused analysis of problems, definition of activities performed, tracking the system state, specification of the flow of items between parts of the system, and identification of the responsibilities of each component (Cloutier 2007; Bui Long 2024).

It is important to note that SysML is simply a graphical modeling language. This makes it tool- and methodology-independent. It captures the system modeling information without prescribing any particular modeling method (Delligatti, Soley, and Steiner 2014). On one hand, this flexibility can be advantageous, allowing tailoring of the MBSE effort to the specific project by choosing the appropriate method and tool. On the other hand, the absence of modeling guidance may lead to ambiguity in the modeling approach. To achieve a sufficiently expressive model, one must be well-informed in selecting the appropriate tool and method. An ideal modeling environment should enable system designers to dedicate their effort to designing systems rather than learning complex tools and techniques to model systems (Alai 2019).

A few of the several tools and methods often used with SysML are discussed shortly in this study, namely Vitech MBSE methodology-CORE, Cameo Enterprise Architecture, OOSEM, and IBN Harmony for SE-Rhapsody. It is acknowledged that this study does not exhaustively evaluate every possible vendor. However, this limited but meaningful scope is deemed sufficient for this thesis.

Vitech MBSE methodology-CORE
CORE is an MBSE software tool developed for Vitech Corporation. It integrates SysML with the Vitech MBSE methodology. While SysML serves as a modeling language, Vitech CORE combines the modeling language, software tool, and methodology in one. CORE is structured around a central integrated design repository linked to four primary concurrent SE activities: Source Requirements Analysis, Functional / Behavior Analysis, Architecture Synthesis, and Design Verification & Validation. These activities are associated with four domains respectively: Source Requirements Domain, Behavior Domain, Architecture Domain, and Verification & Validation Domain (Tepper 2010).

Cameo Enterprise Architecture
Cameo Enterprise Architecture (Cameo EA) is an MBSE tool developed by No Magic Inc. It utilizes various modeling languages but the focus of this research is the SysML. The tool adheres to the pure SysML standard and supports architecture frameworks such as Unified Architecture Framework (UAF), Department of Defense Architecture Framework (DoDAF), etc. Cameo EA uses the SysML extension mechanisms for domain-specific customization through third-party vendors. It supports all nine SysML diagrams along with UML diagrams (Alai 2019). Cameo EA offers various matrices facilitating relationships between requirements and other model objects that meet the requirements. It also includes derived requirements matrices for creating relationships between various types of requirements (Alai 2019). A requirement traceability matrix eases the task of traceability management and is also identified as an important artifact by the INCOSE SE Handbook for verification activities (Walden et al. 2015).

Object-Oriented Systems Engineering Method (OOSEM)
The INCOSE Object-Oriented Systems Engineering Method (OOSEM) is an MBSE methodology, which is heavily reliant on generating SysML products. OOSEM integrates traditional SE process models with top-down, object-oriented concepts, for architecting multi-domain systems that are flexible and extensible to accommodate developing technologies and requirements (Alai 2019; Walden et al. 2015). OOSEM supports the system development process through the following activities: Stakeholder Needs Analysis, System Requirements Analysis, Logical Architecture Definition, Synthesis of Candidate Physical Architectures, Optimizing and Evaluating Alternatives, Requirements Traceability Management, System Verification and Validation. These activities align to a standard SE "Vee" model and can be implemented recursively and iteratively at various levels of the system hierarchy (Alai 2019).

IBM Harmony for SE-Rhapsody
IBM Harmony for Systems Engineering is a subset of the IBM Harmony methodology designed for software engineering. Using the "Vee" model as a basis and following a top-down development approach, IBM Harmony for SE provides a guideline for architecture modeling through three key processes: Requirements Analysis (identifying and deriving required system functions), System-level Functional Analysis (identifying associated system modes and states), and Design Synthesis (allocating identified system functions and models/states to a subsystem structure) (Hoffmann 2011; Alai 2019). This methodology relies heavily on

the creation and use of UML/SysML artifacts. Even though it was developed to be vendor-neutral, it is usually implemented using IBM's Rational Rhapsody tool (Alai 2019; Beery 2016). Rhapsody functions as a central design hub benefiting stakeholder collaboration, document generation, and reporting, all aimed at achieving coordinated and correct systems architectures (Beery 2016).

Although SysML-based tools have proven to be effective in articulating system lifecycle data using models, there are several reasons why some companies or consortiums express concerns regarding the suitability of SysML for comprehensive system modeling (Baron et al. 2023; Alai 2019; Albers and Zingel 2013). Some studies consider that SysML remains too complex or not completely adequate for systems engineers (Bonnet 2019). Being strongly based on UML, it inherits many software engineering concepts that may be challenging for engineers without a software engineering background to grasp (Alai 2019). Additionally, Alai (2019) observed in his study that Arcadia/Capella enables a tighter and easier integration of structure, facilitating functional analysis, in contrast to SysML applied with OOSEM. Notably, the SysML specification is not meant for beginners (Delligatti, Soley, and Steiner 2014). Moreover, this language faces limitations related to the ambiguity of the concept of function, as it involves both activities (actions and blocks) without clear distinctions. SysML cannot also separate structural elements from their functions (Baron et al. 2023). Finally, SysML tools are typically closed-source and often require expensive commercial licenses for usage. The findings of the literature comparison, focusing on MBSE tools, languages, and methods, are consolidated and presented in Table 3.3.

Tool	**Method**	**Language**	**Advantages**	**Limitations**
Capella	Arcadia	Domain-specific Language	Flexibility Functional modeling excellence Simplicity and beginner-friendliness Open source Visualization	Predefined methodology Version-saving Layer skippability Commercial add-ons
CDP4-COMET	Concurrent Design	ECSS-E-TM-10-25	Multidisciplinary and collaboration Comparison of alternatives Reusability Easy to lear Open source	Absence of diagrams Limited maritime adoption Requires larger monitors
CORE Cameo EA Rhapsody	Vitech MBSE Methodology OOSEM IBM Harmony for SE	SysML	Visualization Tool and methodology independent Traceability management SysML V2	Ambiguous concept of function Functional analysis Not meant for beginners Expensive commercial license

Table 3.3: Comparison of MBSE tools-methods-languages

3.4. Conclusions

At the end of this chapter the Research Question 2, which explores the nature of MBSE and its theoretical potential in handling the heightened complexity of early-stage naval ship design, is fully addressed. It can be concluded that MBSE is an approach to SE that places models at the center of design. In response to the critical question of "How do you implement MBSE and what do you need to know", this chapter elaborates the three pillars of MBSE, namely modeling language, modeling method, and modeling tool. Moreover, as explained, MBSE can theoretically offer a foundation for several benefits. Consistency, flexibility, traceability, trade-offs, and efficiency have been correlated with the four identified requirements for design methods in the early stages of warship design. This suggests that MBSE can theoretically benefit in managing the complexity of early-stage naval ship design.

Furthermore, at this stage, Research Question 3, which explores different MBSE tools-methods-languages alongside their strengths and weaknesses, can be answered. The review of popular modeling tools serves to acquaint the reader with industry applications of the three pillars of MBSE. In essence, the value of MBSE and its implementation becomes more evident. It can be concluded that there is not a universally optimal combination. Each combination is associated with features that make it more suitable for certain applications while less suitable for others. Therefore, the selection of the three pillars of MBSE must be made to achieve perfect alignment with the needs of the specific project.

4

Research Process and Implementation Steps

This chapter will elaborate on the systematic process undertaken to achieve the research goal of this thesis. First, the requirements necessary for the process are presented, followed by a detailed, step-by-breakdown. This includes defining the mission, capabilities, and requirements, selecting MBSE tooling, defining a metamodel, constructing MBSE models, verifying-validating and modifying them, collecting, sharing, and analyzing modeling constructs, and finally, evaluating tools based on MBSE factors.

4.1. Key Criteria for Process Formulation

As explained in Chapter 3, a central repository serves as the fundamental base for every MBSE tool. This centralized database enables various data representations for effective communication among stakeholders. Within this repository, system attributes are stored and maintained, enabling on-demand document creation and ensuring that everyone has the latest updates (Tepper 2010). The current thesis examines several artifacts of this central database, including:

- Stakeholders and system users
- Requirements
- Functional descriptions
- Graphical models and tree diagrams
- Operational and systems analysis
- Logical and physical architectures
- Performance attributes
- Data flows

Having identified the research gap and defined the research objective in Chapter 1, it is essential to formulate a structured process to address this. Before delving into the steps of this process, it is worth noting that its construction requires guidance based on specific criteria. Below is a list of the criteria established as part of this research.

- The process shall apply to naval problems in the ESSD
- The process shall reveal direct links between requirements and architectural artifacts
- The process shall involve practically applying MBSE to construct models demonstrating its value, with a focus on consistency, flexibility, traceability, and trade-offs
- The process shall remain straightforward and not overly complex
- The process should not be constrained by the choice of tool or system
- The process should have at least one iterative loop

4.2. Step-by-step Breakdown of the Research Process

Once the criteria for the method have been established, the formulation of the process takes place. The process consists of 10 sequential steps. These steps are presented below in chronological order.

1. Define Mission
2. Define Capabilities
3. Define Requirements
4. Select MBSE Tooling
5. Define Metamodel
6. Construct MBSE Models
7. Verify & Validate MBSE Models
8. Modify MBSE Models
9. Collect, Share, and Analyze Modeling Constructs
10. Evaluate Tools Based on the MBSE Factors

It is noted that this process includes steps from a typical MBSE process, such as requirements management and analysis in Step 3, tool selection in Step 4, system design and behavioral modeling in Step 6, and verification and validation activities in Step 7. A more detailed connection to the Vee SE model is presented later in this chapter.

Continuous iteration is key to success in the development of systems architecture (Kooij 2022). This iteration can occur at the level of individual process elements. For example, iterations are possible while constructing the MBSE model and transitioning from one architectural layer to another (Step 6). Additionally, iteration can occur across a series of consecutive elements. For instance, several iterations may take place between the model verification/validation (Step 7) and model modification (Step 8) steps.

To illustrate the practical application of this process, an example case study involving a specific mission scenario will be discussed in Chapter 5. This case study will provide a practical example of how the process is implemented in real-world settings, offering insights into the value of MBSE in naval vessel design.

The process flowchart is illustrated in Figure 4.1, while a description of each step is presented below:

1. **Define Mission**
   One of the biggest challenges in ship design is not only to design the ship correctly but also to design the right ship (Jansen et al. 2020). In that essence, the design team has to ensure that the vessel is not only designed accurately but also that it meets its operational objectives and goals. To ensure practicality in real-world naval scenarios, the system definition, architecture, and design should be driven by mission-based operational needs (Tepper 2010). As a result, the process in this thesis begins by defining the selected mission that will serve as the basis for deriving the vessel's requirements. Key tasks during this step include:

   - Stakeholder Identification: Identify the key organizations, individuals, or entities involved in the success of the mission.
   - Mission Analysis: Analyze the broader context of the mission, including risks, opportunities, and challenges.
   - Objective Definition: Define clear, measurable goals aligned with the mission and stakeholder needs.

2. **Define Capabilities**
   When designing a product, it is crucial to define its purpose and scope clearly. Without this clarity, there is the risk of investing resources into something that does not effectively address the original problem. This is a highly relevant challenge when it also comes to designing naval vessels. In this case, it is very important first to know the operational need of the vessel or the fleet to establish the early stage design requirements (Knegt 2018).

   During the ESSD, it is possible to identify the specific capabilities of a vessel by creating a capability breakdown illustrated in Figure 4.2. By assigning weights to particular branches, it becomes clear which capabilities should be given priority. It is also worth mentioning that tasks and capabilities may not be exclusive to a single mission; instead, a particular capability may be essential for more than one task (Streng 2021).

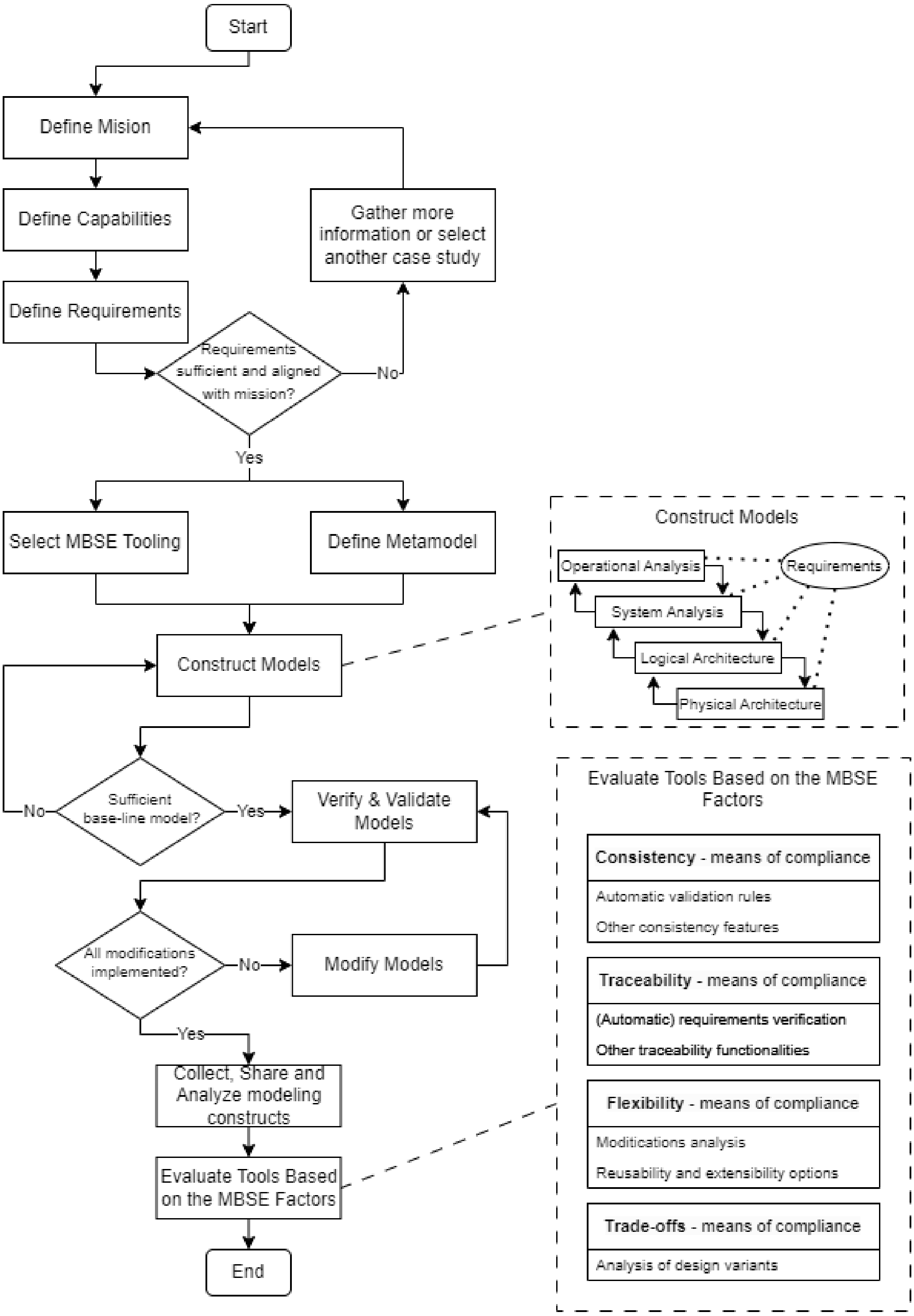

Start
Define Mision
Define Capabilities
Define Requirements
Gather more information or select another case study
Requirements sufficient and aligned with mission?
No
Yes
Select MBSE Tooling
Define Metamodel
Construct Models
Construct Models
Operational Analysis
Requirements
System Analysis
Logical Architecture
Physical Architecture
Sufficient base-line model?
No
Yes
Verify & Validate Models
All modifications implemented?
No
Modify Models
Yes
Collect, Share and Analyze modeling constructs
Evaluate Tools Based on the MBSE Factors
End
Evaluate Tools Based on the MBSE Factors
Consistency - means of compliance
Automatic validation rules
Other consistency features
Traceability - means of compliance
(Automatic) requirements verification
Other traceability functionalities
Flexibility - means of compliance
Modifications analysis
Reusability and extensibility options
Trade-offs - means of compliance
Analysis of design variants

Figure 4.1: Flowchart of the proposed research process

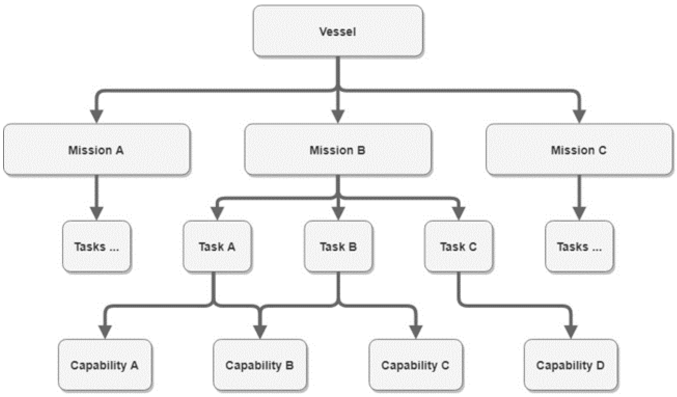


Figure 4.2: Mapping necessary capabilities to ship missions. Source: Streng (2021)

3. **Define Requirements**
   It should be noted that obtaining technical requirements from a mission is not always straightforward. For example, notable discrepancies in design between submarines and surface vessels are still anticipated, even though both can be intended for the same missions (Streng 2021).

   Through mission, task, and capability analysis, a list of systems fulfilling those capabilities can be generated. The creation of several alternative lists of systems is inherent during ESSD, particularly during the concept exploration phase. Each system impacts the vessel's effectiveness in many ways. Therefore, categorizing them helps understand interactions (Streng 2021).

   It is fully recognized that decomposing these mission requirements down to a detailed requirement level, as applied to the SoI in this thesis, falls well within the capabilities of MBSE. Thus, this process is integrated into a more comprehensive, large-scale deployment of the methodology.

   At the end of this step, it should be ensured that each requirement directly contributes to achieving the defined objectives and goals of the mission. Additionally, requirements should be validated to eliminate any that are extraneous or unrelated to mission needs. Finally, the requirements must be sufficient to demonstrate the value of MBSE, meaning that they should be selected in a way that allows for automatic verification later on.
4. **Select MBSE Tooling**
   This step involves the selection process of the MBSE tools, languages, and methods for the subsequent modeling effort. At this point, it is imperative to articulate and document the rationale behind the selected combination of tools, methods, and languages, clearly showing how they align with the project's objectives and needs.
5. **Define Metamodel**
   The success of the primary goal of MBSE, which is to encapsulate all project information within models, heavily relies on the system model's ability to effectively represent the information needs of a broad range of stakeholders (Do et al. 2012).

   A metamodel (alternatively known as a reference model, schema, or ontology) can be defined as the explicit specification of an abstraction—a simplification. To define this abstraction, the metamodel identifies a list of relevant concepts and a set of relevant relationships between these concepts (Do et al. 2012). In other words, it defines the language structure as classes of elements and the permissible relationships between them (Logan et al. 2013).

   Therefore, before modeling begins, a methodology-independent metamodel of the initial configuration of the SoI must be developed. This metamodel philosophy ensures that the various project partners have a shared view of the system under consideration. Throughout the metamodel development, careful consideration is needed to determine what should be included in the model and what should not. The metamodel serves not only to reflect the intended structure and behavior of the system but also to streamline decision-making to achieve that structure. The model should be lean, containing only essential information necessary for the current design phase to assess performance and compliance with requirements.

Implementing all previous steps, including this one, fully addresses Research Question 4 (*" What critical factors play a key role in selecting a representative case study, considering a) the selected system of interest, b) the level of detail, and c) the chosen modeling tool-method-language?"*).

6. **Construct MBSE Models**
   The baseline model should adhere to the metamodel, which outlines key elements and relationships abstractly. This high-level system decomposition provides a framework for modeling and serves as a template structure for the system architecture, independent of the tool and modeling language. The implementation of the current step is considered pivotal in answering Research Question 5 (*" How can the selected MBSE tool-method-language be effectively applied to model the architecture of the chosen system, and what critical insights can be gained from this modeling process?"*).

   As highlighted, this thesis concentrates on the development of system architecture, encompassing a conceptual model that delineates the system structure across four distinct levels: Operational, Functional, Logical, and Physical layer. Therefore, irrespective of the tool used, the model development should proceed through the following steps:

   First, the Operational Analysis stage should be performed, focusing on defining stakeholder needs and system context. The central question to be answered in this perspective is "What must the system users achieve?". The modeling workflow at this level is as follows:

   - Capture operational entities and actors
   - Capture the goals of the actors
   - Define operational activities along with their interactions
   - Define operational interactions sequence

   Then system analysis takes place, focusing on formalizing the requirements and defining the SoI. In this phase, the focus shifts to the system as a black box, its boundaries, its interactions with its environment, and its functions. The main question to be answered in this level is "What the system must achieve for the users?". The modeling workflow at this level is as follows:

   - Capture system actors
   - Transition and refine operational activities into new system functions
   - Allocate system functions to system actors
   - Create traceability links to requirements

   The System Analysis described earlier involves functionally analyzing the SoI as a "black box." The subsequent level, the Logical Architecture, begins to "open the box" by implementing the major decisions of the solution and establishing methods to meet stakeholder expectations. The central question to be answered in this perspective is "How will the system work to meet expectations?". The modeling workflow at this level is as follows:

   - Transition and refine system functions into new logical functions
   - Identify and define logical components along with their hierarchy
   - Allocate logical functions to logical components
   - Create traceability links to requirements

   Finally, based on the output of the Logical Architecture containing abstract decisions, the Physical Architecture can create concrete components that comprise the SoI. It defines the solution architecture by concentrating on the "final" architecture for system development, implementing technical and technological constraints and choices. The main question to be answered in this level is "How will the system be built?". The modeling workflow at this level is as follows:

   - Transition and refine logical functions into new physical functions
   - Identify and define physical components along with their hierarchy
   - Allocate physical functions to physical components
   - Create traceability links to requirements

The aforementioned steps should not be regarded as strictly following a chronological order. As underlined, continuous iteration is paramount to success in the development of systems architecture (Kooij 2022). The modeling process should be designed to enable revisiting previous perspectives and restructuring the architecture if it is deemed to result in a superior design. By the end of this step, the initial version of our system—a baseline model—has been established. If this model is not sufficient, iteration continues until it does. In this thesis, a sufficient baseline model should accurately represent stakeholder needs, fully capture all specified system requirements, and maintain traceability between different architectural elements. Specific benchmarks include the completeness of operational, functional, logical, and physical views as dictated by the metamodel.

7. **Verify & Validate MBSE Models**
   The primary focus of this step is to evaluate the tool's support for verification and validation activities. Verification and validation are pivotal terms in scientific research, serving as cornerstones for establishing trustworthiness in results. Verification is the process of checking a system or system element against required characteristics. This includes but is not limited to, specified requirements, design descriptions, and the system itself. Conversely, validation confirms that a system can accomplish its intended use, goals, and objectives within the intended operational environment. In simpler terms, verification ensures that the ship has been designed correctly, while validation verifies whether the correct ship has been designed (Knegt 2018).

   The model is verified within the tool to ensure that it meets its requirements and accurately represents the system being modeled. This verification process confirms whether the MBSE model effectively captures the mission requirements and design decisions specified in Step 3.

   Validation can be achieved by comparing the results obtained from either building a prototype or conducting full-scale tests with the model results. However, in the context of the current thesis, which focuses on exploring the value of MBSE in ESSD, the emphasis is on demonstrating the efficacy of the MBSE approach itself through a practical example involving a vessel design rather than directly validating specific model outputs against real-world data. Therefore, this thesis takes the form of a case study research, a methodology inherently associated with validity challenges (Morris 2019). Therefore, while validation through physical testing is a key term in research, its omission in this thesis is deliberate.

   However, the concept of "model validation" takes on a different connotation within the context of this thesis. In addition to traditional validation methods, MBSE tools incorporate built-in "validation rules" to ensure compliance with language constructs. Furthermore, custom validation rules can be added to detect deviations from the designated design strategy, even if they do not constitute language violations. Consequently, the models undergo validation through these means.

8. **Modify MBSE Models**
   Analyzing designs, determining necessary modifications, and resolving conflicts are pivotal for the success of MBSE in ESSD. The versatility of the MBSE environment is evaluated through various modifications applied to the model, impacting the system configuration and tool setup. Following each modification, the resulting design is verified and validated to ensure compliance with requirements (Step 7). Thus, the iterative loop between Steps 7 and 8 continues seamlessly.

   The following list of design modifications is proposed:

   Consumer Set Modifications:

   - Addition of new equipment
   - Removal of existing equipment

   Model modifications:

   - Addition of a new parameter

   With these components, including the mission, the capabilities, the system decomposition starting from the operational down to the physical level, as well as the design modifications and the verification of requirements, all key parts of the V model are accounted for. A mapping between the activities of this thesis methodology within the Vee SE cycle model is depicted in Figure 4.3.

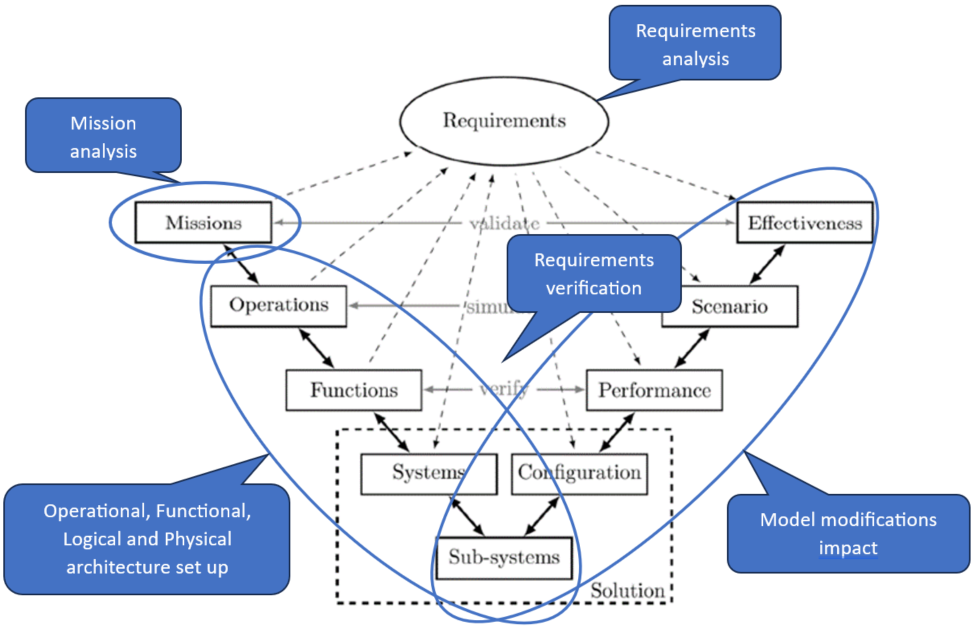


Figure 4.3: Mapping of methodology activities within the Vee SE cycle model

9. **Collect, Share and Analyze Modeling Constructs**
   Once the models are constructed, useful data can be exported and analyzed. Depending on the selected tools, these can take the form of graphical diagrams, product trees, relationship matrices, functional chains, requirements traceability matrices, reporting constructs, etc. Moreover, it is recognized that sharing functional models with all stakeholders is essential in MBSE, and for large projects with multiple stakeholders. Therefore, the sharing capabilities provided by each tool will also be explored to some extent.

10. **Evaluate Tools Based on the MBSE Factors**
    In this thesis, the exploration of the value of MBSE is achieved through the validation of its benefits. Therefore, it is crucial to clearly define the metrics and criteria used to measure this value. This includes the key MBSE factors discussed in subsection 3.2.2: consistency, flexibility, traceability, and trade-offs. The following paragraphs outline a framework that proposes means to ensure compliance in demonstrating the relevant benefits of MBSE in ship design.

    - **Consistency**
      A System Specification is consistent when it is in line with the modeling syntax & architecture (internal consistency) and the values and meaning of concepts are identical throughout the models (inter-tool consistency) (Kolfschoten et al. 2024). This project focuses on the internal consistency.

      Means of compliance:

      – Validation Rule Assessment: Evaluate consistency leveraging each tool's built-in/custom validation rules for automated checks.

      – Other consistency features: Explore additional examples of consistency functionalities offered by various MBSE tools, demonstrating how each tool supports consistency within the modeling process.

    - **Traceability**
      Traceability is created by relating elements in the model to the requirements they derive from. These traces help to create a parsimonious or lean specification; one that contains all that is needed, and nothing that is not needed to meet the requirements. Moreover, traceability helps to ensure the completeness of the model and can assist engineers in the verification of requirements (Kolfschoten et al. 2024).

  Means of compliance:

  - Traceability of requirements: Explore each tool's capability for manual/automatic verification of requirements
  - Other traceability functionalities: Explore other traceability functionalities of the different MBSE tools that are applicable to the project

- **Flexibility**
  A System Specification is flexible when it can be changed and adapted with little effort to make the change consistently throughout the associated models and to make the change when impacting a large number of elements (Kolfschoten et al. 2024).

  The assessment of flexibility concerning the adjustment to the MBSE models should focus on two key aspects:

  - Simplicity: Adjustments should allow users within an organization to initiate changes independently without relying on external parties.
  - Rapid Propagation: The speed at which adaptations can propagate throughout the entire model is also of paramount importance.

  Means of compliance:

  - Check model structure and tool features: Report reusable and non-reusable elements
  - Modification analysis: Measure the number of manual actions to incorporate changes to the models using the Design Modifications

- **Tradeoffs**
  Trade-offs refer to the compromises or choices made between different desirable attributes or features to achieve the best overall solution for a given problem (Kolfschoten et al. 2024).

  Means of compliance:

  - Comparative analysis of variants: Evaluate the capability of each tool to enhance transparency in trade-offs, allowing for a more informed comparison of design variants

As a result, Research Question 6 (*"How does the selected MBSE tool-method-language, perform in terms of consistency, traceability, flexibility, and trade-off analysis within the context of system architecture development?"*) will be addressed.

5
Background of the Case Study

This chapter focuses on the initial phase of the proposed process, centering on the preparatory steps that establish the groundwork for the example problem. It begins by outlining the scope of the case study, followed by a comprehensive examination of the mission and system of interest. The mission is then translated into a series of operational capabilities, which, in turn, are translated into a set of requirements for the vessel under consideration. Collectively, these steps form the basis for addressing Research Question 4, focusing specifically on parts a) and b): *"What critical factors play a key role in selecting a representative case study, considering a) the selected system of interest, b) the level of detail and c) the chosen modeling tool-method-language?"*

Disclaimer: The design and case study presented in this thesis are fictitious and based on generalized characteristics typical of Landing Platform Dock (LPD) vessels. They do not represent any specific vessel from the Dutch Navy or any other navy, nor are they based on proprietary or classified information. The study draws upon common features observed in LPD vessels and aims to illustrate the practices of MBSE within the context of amphibious operations.

5.1. Scope of the Case Study

The scope of the case study in this thesis is chosen to demonstrate at least the key benefits of MBSE while addressing inherent complexity at a minimum. Therefore, an amphibious warfare ship, specifically a Landing Platform Dock vessel, is designated as the system of interest (SoI). The SoI is viewed as a component within a larger system of systems (SoS), such as a naval fleet. However, it is considered wise to narrow down the level of focus to the vessel itself. Thus, for this project, interfaces with other fleet vessels are considered to provide no additional value and are therefore excluded. Conversely, interfaces with other systems, as well as users, must be taken into consideration. This includes interfaces with systems such as Amphibious Landing Crafts and Aircraft, as well as with users, such as the Amphibious Task Force and Ship Crew (comprising all other personnel onboard). All these entities are called Actors. The impacts of the environment are only addressed through the implementation of additional or adjusted requirements.

5.2. Mission and SoI

This section discusses the selected mission, representing Step 1: Defining Mission, as outlined in the methodology of Chapter 4. This will serve as the basis for deriving the capabilities and then the requirements for the SoI.

The main objective of the amphibious raid mission is to ensure the embarkation and safe transportation of the Amphibious Force (AF) and their equipment from the base to the drop-off point while maintaining ongoing support throughout the assault phase of the amphibious raid.

Description of the mission:
In this thesis, the focus will be on an amphibious raid, selected as an illustrative example from the range of amphibious operations. The example mission consists of two phases: the first occurring before the amphibious raid, and the second encompassing the amphibious raid until its completion. This simplified framework aims to enhance comprehension of the ship's operational objectives within each of these distinct phases. The phase involving the ship's return to base has been omitted as it closely resembles the initial stage, offering minimal additional value.

It is clear that the system of interest in this example case would be an amphibious assault ship. An amphibious assault ship is a type of amphibious warfare ship employed to land and support ground forces on enemy territory during an amphibious assault. In this example case, the selected type of amphibious vessel for this specific mission is selected to be a Landing Platform Dock (LPD), also known as an amphibious transport dock. This type of ship is specifically designed to support the landing of troops, equipment, and supplies using several means, including landing craft and/or Amphibious Assault Vehicles (AAVs) operating from its well deck, and helicopters and tiltrotor aircraft operating from its flight deck (United States Department of Defense 2019; Department of the Navy, Office of the Chief of Naval Operations 2007). In addition to the helicopter deck, the presence of a hangar (enclosed shelter structure surrounding the platform deck used for storage and maintenance of the aircraft) distinguishes it from a Landing Ship Dock (LSD) according to the classification of the US Navy. The LPD in this example mission should be capable of providing support during the two phases of the amphibious raid mentioned earlier.

In this example mission, the phase before the amphibious raid encompasses the embarkation and movement stages. This phase initiates with the loading of the AF onto the LPD, along with relevant equipment and amphibious vehicles, from the base, in a process known as embarkation. Therefore, the LPD vessel must be equipped with the necessary facilities and allocated spaces to facilitate the embarkation phase effectively. Also, it is assumed that the port in the base also introduces some constraints regarding the ship dimensions. After the embarkation stage is complete, everything must be swiftly and securely transferred to the Amphibious Objective Area (AOA). This period of sailing to the AOA is referred to as the movement stage. The AOA is a geographical area delineated in the order initiating the amphibious operation, and for this thesis, it can be referred to as the drop-off point. During the phase before the amphibious raid, supporting operations are carried out from the ship's vantage point. This demands not only providing propulsive power and navigation guidance to reach the specified drop-off point but also ensuring the provision of electrical resources, habitable conditions, and accommodation for the embarked marines and their equipment. This phase concludes upon the LPD reaching the designated drop-off point, initiating the embarkation of AF onto smaller ships and/or aircraft for the raid.

The next phase of this example mission is the one concerning the amphibious raid itself, also referred to as the assault or action phase. The action phase commences when sufficient AF assets are positioned in the landing area to permit the assault. This phase encompasses several distinct stages, including ship-to-shore movement, accomplishment of the assault objectives, and scheduled withdrawal, as elaborated below. The amphibious raid initiates once the AF embarks on the landing assets, along with equipment and assault vehicles. Following this is the ship-to-shore movement stage. It refers to the period during which various elements of the AF move from points of embarkation or forward-deployed locations to the objective area. The ship-to-shore movement can be surfaceborne, airborne, or a combination of both, depending on the available landing means (assets). When sea-based means of transport are used, it is termed surface movement, while in the case of aerial means, it is referred to as air movement (Department of the Navy, Office of the Chief of Naval Operations 2007). In this specific example, flexibility for the ship-to-shore movement is considered important. A landing craft, such as the Landing Craft Air Cushion (LCAC), or Landing Craft Utility (LCU), or even Amphibious Assault Vehicles (AAVs), will facilitate surface movement, while medium-sized helicopters are considered relevant for air movement. In both scenarios, the landing assets are deployed from the ship and navigated to the shoreline. Upon reaching the shore, both troops and vehicles disembark from the landing craft, with vehicles being applicable only during surface movement. This initiates the engagement phase, which refers to the period between the arrival of the AF in the operational area and the accomplishment of their mission. The actions of the AF in the operational area to secure designated objectives are considered beyond the scope of this example. During the assault phase, the ship must provide essential support functions such as ensuring safe and rapid deployment, as well as retrieval of landing assets, logistical assistance, and surveillance capabilities. Additionally, it should be equipped to offer air support to the landing operations and facilitate emergency evacuation if required. The assault phase ends with a pre-scheduled withdrawal, involving the extraction of forces by aerial or sea means from a hostile or potentially hostile shore. This stage parallels the ship-to-shore movement but in reverse, transitioning from shore to ship. The action phase concludes upon the return of the landing means to the LPD.

5.3. Translation of the Mission Needs to Operational Capabilities

This section will further examine which capabilities are necessary for the selected mission, thus representing Step 2: Define Capabilities as detailed in the methodology of Chapter 4.

In the current case study, the focus will be applied to only one major amphibious mission area - Amphibious Raid. It is worth mentioning that this approach can be scaled up to encompass more comprehensive warship designs. In a real-world LPD design project, all potential missions would be carefully evaluated and modeled accordingly. As the project unfolds, initial models would be replaced with more detailed and specific ones.

By selecting a specific mission for the LPD vessel in this case study, this method provides insight into the essential capabilities required to accomplish that mission. This process is illustrated in Figure 5.1, detailing how capabilities are mapped to the example mission.

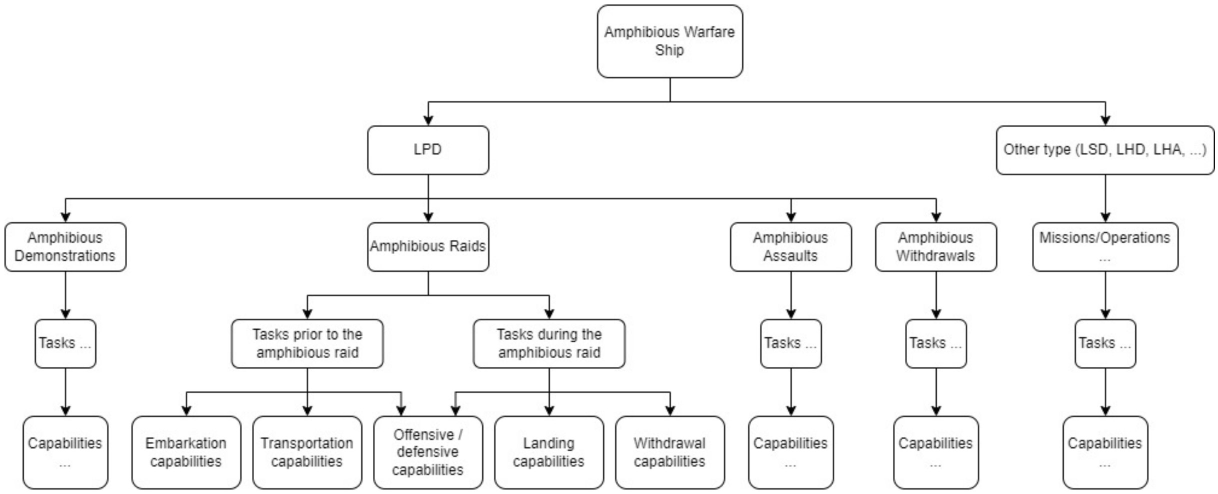


Figure 5.1: Mapping capabilities to the example mission.

5.4. Vessel Requirements

This section addresses the requirements definition, corresponding to Step 3: Define Requirements, as described in the methodology of Chapter 4.

In the context of this thesis, a simplified version of a requirements list, which reflects a small set of requirements as encountered during real ship design projects, was formulated. Alongside the requirements derived from the aforementioned mission, additional criteria from similar vessels are also incorporated. The requirements serve as guiding principles to ensure consistency in constructing models within both tools, allowing for a fair comparison between them. Furthermore, these requirements not only provide modeling content to assess the tool's capabilities and the value of MBSE in ESSD but also serve as a means to validate the constructed model.

However, at this point, it should be noted that that a comprehensive justification of these requirements falls beyond the scope of this study. Rather, this simplified analysis intends to furnish a rational foundation that facilitates the illustration of the application and significance of an MBSE approach in the preliminary stages of warship design. It is also worth noting that the requirements are formulated to reduce the level of detail, focusing on fundamental equipment and subsystems. This simplification streamlines the system architecture of the LPD by limiting the number of components (modeling elements) under consideration. Consequently, some ambiguity and conflicts may arise; however, this is considered acceptable as the objective here is to assess if and how the tools can effectively manage such challenges.

With that in mind, the examined LPD vessel must adhere to the hypothetical requirements presented in Table 5.1. Each requirement is provided with a description and an ID. The number and sequence of digits in the ID code correspond to the derivation procedure used. For example, requirement R5.1.2 was derived from requirement R5.1, which in turn was derived from requirement R5.

Landing Platform Dock Example Requirements

Req ID	Req Text	Req ID	Refinement 1	Req ID	Refinement 2
R1	The vessel shall support the deployment of troops and crew personnel	R1.1	The vessel shall accommodate a total of 300 armed troops and 100 crew-personnel		
R2	The vessel must have designated spaces for efficient embarkation, storage, and transportation of supplies, equipment, and amphibious vehicles	R2.1	The vessel shall provide 500 square meters of cargo area for general cargo		
		R2.2	The vessel shall accommodate 35 infantry fighting vehicles with a Ro-Ro area of approximately 1000 square meters and a payload weight capacity of approximately 875 metric tons for vehicles		
R3	The vessel shall ensure swift, secure, and autonomous transfer of personnel, equipment, and vehicles to the AOA	R3.1	The vessel shall maintain a minimum transit speed of 14 knots		
		R3.2	The vessel shall possess a minimum unrefueled transit range of 8000 nautical miles at a speed of 14 knots.		
		R3.3	The vessel shall be equipped with a bow thruster to enhance maneuverability during docking and low-speed operations		
R4	The vessel's dimensions shall be optimized to ensure navigability within designated ports, waterways, and operational constraints	R4.1	The vessel shall have dimensions optimized for navigability, with a maximum draft not exceeding 5.5 meters, a length overall (LOA) not exceeding 130 meters to accommodate port facilities, and a beam not exceeding 22 meters to ensure maneuverability within confined waterways and docking facilities.		
R5	The vessel shall offer amphibious raid flexibility, enabling ship-to-shore movement via landing craft and/or aircraft	R5.1	The vessel shall support ship-to-shore movement via amphibious landing craft	R5.1.1	The vessel shall offer landing craft flexibility, capable of accommodating and deploying 2 LCUAs, 1 LCU, or 9 AAVs as needed
				R5.1.2	The vessel shall feature a well deck with dimensions of L55m x W15m x H6.5m
		R5.2	The vessel shall support ship-to-shore movement via aerial means	R5.2.1	The vessel shall provide a flight deck measuring at least L40m x W20m, capable of simultaneous launch and recovery of two helicopters.
				R5.2.2	The vessel shall provide hangar space for 1 helicopter
				R5.2.3	The vessel shall feature a Helicopter Coordination Section to facilitate safe and efficient coordination of helicopter activities
R6	The vessel should facilitate the efficient loading and preparation of the landing craft or aircraft	R6.1	The vessel shall feature an overhead crane with a capacity of 20 tons for cargo handling operations		
		R6.2	The vessel shall include an aircraft elevator with a capacity of 20 tons to facilitate the vertical transportation of helicopters		
		R6.3	The vessel shall include a Ro-Ro ramp to ensure direct vehicle loading and unloading		
R7	The vessel shall provide emergency rescue capabilities utilizing aerial methods				
R8	The vessel shall be equipped with fire support systems and threat counteraction capabilities to ensure the safety and effectiveness of amphibious operations	R8.1	The vessel shall enable the detection and classification of potential threats	R8.1.1	The vessel shall be equipped with sensors
				R8.1.2	The vessel shall be equipped with automatic systems with identification protocols
		R8.2	The vessel shall be equipped with a Phalanx Close-in Weapon System (CIWS) to provide defense against anti-ship missiles and other close-range threats		
		R8.3	The vessel shall be equipped with decoy systems for self-defense against incoming threats		
R9	The vessel shall maintain habitable conditions for embarked personnel	R9.1	The vessel shall regulate internal temperature for climate control		
		R9.2	The vessel shall ensure continuous airflow for fresh air		
		R9.3	The vessel shall ensure safe and drinkable water supply		
R10	The vessel shall be equipped with an electric plant capable of meeting the vessel's electric demand under all operational conditions				
R11	The vessel shall include a hospital facility with a capacity of 10 beds, equipped to provide comprehensive medical care				
R12	The vessel shall be equipped with communications capabilities, including satellite communication, radio communication, and data transmission systems				
R13	The vessel shall be equipped with surveillance capabilities, including radar systems, sonar systems, and optical sensors, to provide comprehensive situational awareness				
R14	The vessel shall have command and coordination capabilities to facilitate effective leadership and coordination of the operation	R14.1	The vessel shall include an Operations Command Area spanning 400 square meters, capable of accommodating up to 100 military staff		

Table 5.1: Example LPD Requirements (Fictive)

5.5. Conclusions

The preparatory work in this chapter establishes the rationale for selecting the SoI and defining its level of detail:

- **System of Interest:** The study opted for an amphibious warfare ship, particularly an LPD vessel, as the SoI to highlight MBSE advantages. This choice parallels the concept exploration phase of ESSD, focusing on comprehensive ship investigation at a lower level of detail akin to early ship design stages. Focusing solely on the vessel sufficed for achieving project objectives, enabling thorough examination of interfaces with relevant systems (e.g., amphibious landing crafts, aircraft) and users (e.g., amphibious task force, ship crew) while excluding non-essential fleet vessel interfaces.
- **Level of Detail:** The level of detail in this case study intentionally focuses on essential operational aspects of the LPD vessel during an amphibious raid. Excluding non-essential interfaces with other fleet vessels allows for a detailed examination of the LPD's operational capabilities. Requirements were crafted to streamline detail in essential subsystems and components, ensuring a clear focus on operational objectives across different mission phases.

As a result, this chapter has addressed parts a) and b) of Research Question 4: *"What critical factors play a key role in selecting a representative case study, considering a) the selected system of interest, b) the level of detail and c) the chosen modeling tool-method-language?"*.

6 Modeling

This chapter focuses on the next phase of the proposed process, centered on the steps regarding the modeling effort. First, the rationale behind the MBSE tool selection is presented, followed by the construction of a design template that will guide the modeling process. Next, a comprehensive summary of the modeling effort in both tools is provided. This addresses part c) of Research Question 4 is answered: *"What critical factors play a key role in selecting a representative case study, considering a) the selected system of interest, b) the level of detail and c) the chosen modeling tool-method-language?"*. Furthermore, this practical application fully addresses Research Question 5 (*" How can the selected MBSE tool-method-language be effectively applied to model the architecture of the chosen system, and what critical insights can be gained from this modeling process?"*).

6.1. Tool-Methodology-Language Selection

This section addresses the MBSE tooling selection for the aforementioned example problem, representing Step 4: Select MBSE Tooling, as specified in the methodology of Chapter 4.

A literature comparison of contemporary MBSE tools, methods, and languages has been conducted in Section 3.3, with the results summarized in Table 3.3. Building upon these findings, the MBSE tools examined in this thesis, along with their respective languages and methods, are carefully selected. The rationale behind this selection is elaborated upon in the following paragraphs.

The first tool to be used in this project is CDP4-COMET. Given that this thesis is conducted in collaboration with Starion (previous RHEA), the company behind this MBSE tool, the author aims to make good use of the strong advantage of having direct access to experts proficient in using or even developing this tool. Moreover, the author's participation in ship design projects using CDP4-COMET addresses the tool's less common usage in the maritime industry, overcoming challenges such as the limited ship-specific templates. Furthermore, it is a tool that is not hard to learn, which is also pivotal considering the time constraints. Moreover, the benefits of reusability and comparison of alternatives are deemed highly relevant in validating the expected benefits of consistency and tradeoffs. Lastly, it is known that CDP4-COMET supports MBSE using the CD method. Thus, it facilitates a more integrated design process where trade-offs are evaluated by the customer, and design challenges are addressed in a multidisciplinary manner. This philosophy perfectly aligns with the complexities of ship design, making it an additional compelling reason for selecting CDP4-COMET for this project.

In addition to CDP4-COMET, the Capella MBSE tool will be used. The predefined methodology, Arcadia, serves as an advantage rather than a limitation in this particular thesis. This is due to its strong alignment with the author's definition of systems architecture, making it a well-suited choice. Next to that, its simplicity and user-friendliness make it a wise choice, considering the timely completion of this thesis. The guidance that this methodology inherently offers is a significant advantage for inexperienced MBSE tool users. Moreover, Capella's excellence in functional modeling is also directly linked to showcasing the advantages of traceability. Last but not least, the fact that it does not require a commercial license was also considered upon its selection.

In this project, modeling involved creating content in two MBSE tools of interest: CDP4-COMET and Capella. The decision to begin with the Capella model was based on its expected usefulness in early organization and definition of the system's ecosystem. The decision to create the CDP4-COMET model afterward was based on the suspicion that it offers an efficient solution for incorporating a large volume of parameters, particularly advantageous for the physical design aspect. It is important to note that the general methodology used in creating the models allows flexibility in the sequence of model creation, as it is tool-independent. However, the chosen order was specifically designed to facilitate the author, who lacked experience in system modeling tools.

6.2. Metamodel

This section addresses the formulation of the metamodel, representing Step 5: Define Metamodel, as specified in the methodology of Chapter 4. A metamodel helps establish a common understanding of the design challenges around complex system development for all the disciplines involved. The metamodel created to facilitate and guide the subsequent modeling activities is presented in figure 6.1.

The development of the system architecture for the SoI involves multiple levels. The first level is the operational level, which covers the questions, "Who is interested in using the system and what do they intend to accomplish?". Thus, the Operational Analysis typically begins by identifying the future system's users and any containment relationships among them. This is illustrated in the top part of the metamodel, highlighted in orange in Figure 6.1. The mission description leads to the identification of operational entities or actors, referred to and modeled as (system) actors. The actors "implement" various activities to achieve their goals. These activities are modeled as operational activities. The interactions and sequences of these activities are also modeled by including appropriate relationships offered by the tool.

Next is the System Analysis phase, which addresses the question, "What should the system do?" This phase treats the system as a black box to define how it can meet the previously identified operational needs. In this step, the operational activities identified during the Operational Analysis are transformed and refined into new modeling elements called system functions. These system functions can then be allocated to an actor or a system component using the relationships provided by the tool. With the functions identified, their interactions can be modeled as system functional exchanges. This process ultimately aims to define the SoI. Consequently, the SoI emerges and can now be modeled. The system must be able to "implement" the aforementioned system functions. The modeling elements belonging to the System Analysis level are depicted with green color in Figure 6.1.

As previously explained, the mission led to the definition of capabilities, which in turn generated a list of requirements for the example problem. All these requirements are incorporated into the models. Each requirement is characterized by a brief description and an ID, and they are represented in purple in Figure 6.1.

The requirements identified at the Operational Analysis level should be derived and formalized at this stage. This process results in the higher-level requirements, which are linked to the system—still treated as a unit (black box)—through the "satisfy" relationship. It should be noted that the high-level requirements correspond to the requirements listed in the first column of Table 5.1.

After the System Analysis, attention turns to the Logical Architecture, where significant decisions regarding the solution's construction principles are made. Essentially, this stage involves a comparison between the expressed needs, a functional analysis describing the chosen system behavior to satisfy requirements, and a structural analysis aimed at identifying the subsystems that will constitute the SoI. To achieve this, the system functions undergo detailed decomposition as they transition and refine into new logical functions. Functional exchanges between these newly modeled logical functions are also incorporated into the model. Subsequently, the functions are allocated to subsystems, which act as logical components. In essence, the previously defined system is broken down into several subsystems that collectively facilitate the execution of logical functions. The modeling elements that concern the Logical Architecture are highlighted with blue color in Figure 6.1. Furthermore, compliance with the requirements is verified by mapping the subsystems to lower-level requirements. It is important to note that lower-level requirements correspond to the refined requirements listed in the second and third columns of Table 5.1.

One important rule followed during the development of the Logical Architecture is to exclude all technological considerations or implementation choices at this level. The objective of the next level, the Physical Architecture, is to define the "real" concrete components that comprise the system.

To shift to the Physical level based on the Logical level, a transition occurs similar to those from Operational Analysis to System Analysis and from System Analysis to Logical Architecture. In this transition, logical functions are refined into even more detailed physical functions. Similarly, subsystems are further decomposed to the equipment level, defining modeling elements such as equipment that implements these Physical Functions. All the modeling elements within this layer are highlighted in yellow in Figure 6.1. Again, it is crucial to verify that the SoI meets the intended requirements. For this purpose, equipment is mapped to the lower-level requirements via the "satisfy" relationship.

During the modeling procedure, the tool's ability to parameterize the modeling elements, such as the system, subsystems, and equipment, will be evaluated. If deemed valuable, parameters like SWBS coding, mass, center of gravity, electric load, etc. will be assigned to the modeling elements.

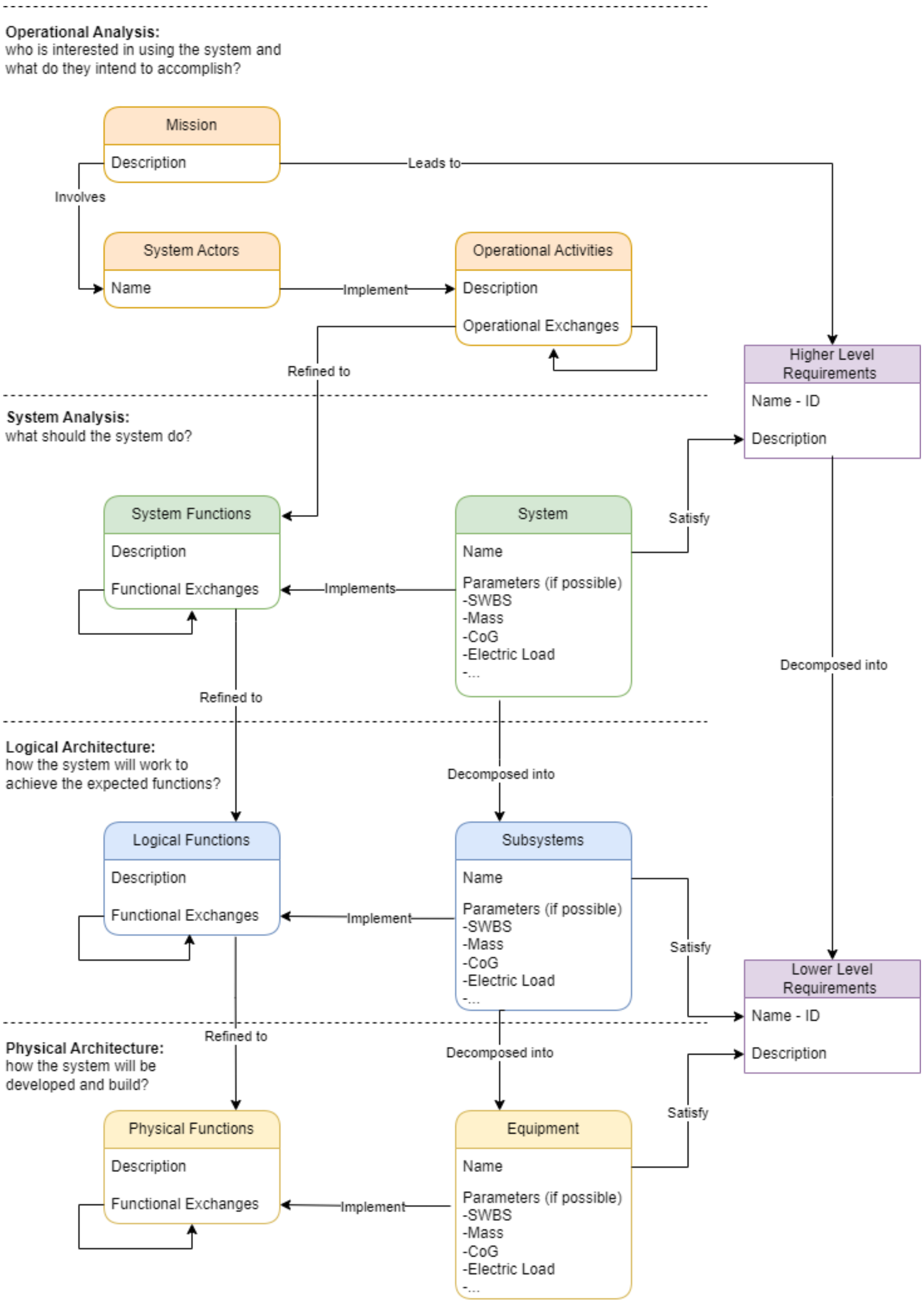


Figure 6.1: Metamodel

6.3. Modeling in Capella

6.3.1. Introduction to Capella

Before delving into the development of the system architecture in Capella, some of its key features should be outlined. Capella is a system architecture modeling tool based on the Arcadia methodology. As such, it supports the definition of needs and solutions through a multi-layered structured analysis. Visualization of the operational, as well as the functional (and non-functional), analysis offers detailed insights into the needs and objectives related to a system. Moreover, views on the logical layer and physical layer present potential solutions through architectural design. The vast majority of the views in this tool are provided in the form of architecture diagrams, breakdown diagrams, and exchange scenarios (Kolfschoten et al. 2024).

In this graduation project, Capella version 6.1 was used. To enhance its capabilities, open-source add-ons such as “Requirements Viewpoint” and “XHTML Documentation Generation”. along with sample add-ons like “Basic Mass Viewpoint”, were installed and used during the development of the architecture (Tool 2024). To gain experience with the tool and understand its potential, tutorials such as the Catapult Toy Project (Arikan and P. Jackson 2023), the UAV for Agriculture Project (Samares Engineering 2024), and the In-Flight Entertainment System (Eclipse Foundation n.d.) were carefully explored.

A detailed description of the process of creating the diagrams is considered beyond the scope of this thesis. Instead, a brief overview of the key concepts behind each diagram and its value will be provided. This approach aims to motivate the reader and offer basic knowledge for inexperienced system modelers. Details on diagram creation will be included only if they are deemed essential for achieving the objectives of this thesis. All diagrams generated using Capella are documented in the Appendix Chapter 9.

6.3.2. Getting Started

As this thesis aims to compare two MBSE tools while fulfilling its primary objective, it is deemed important to briefly outline the modeling environments of each tool. Figure 6.2 presents the user interface of Capella, focusing on four main parts labeled from A to D:

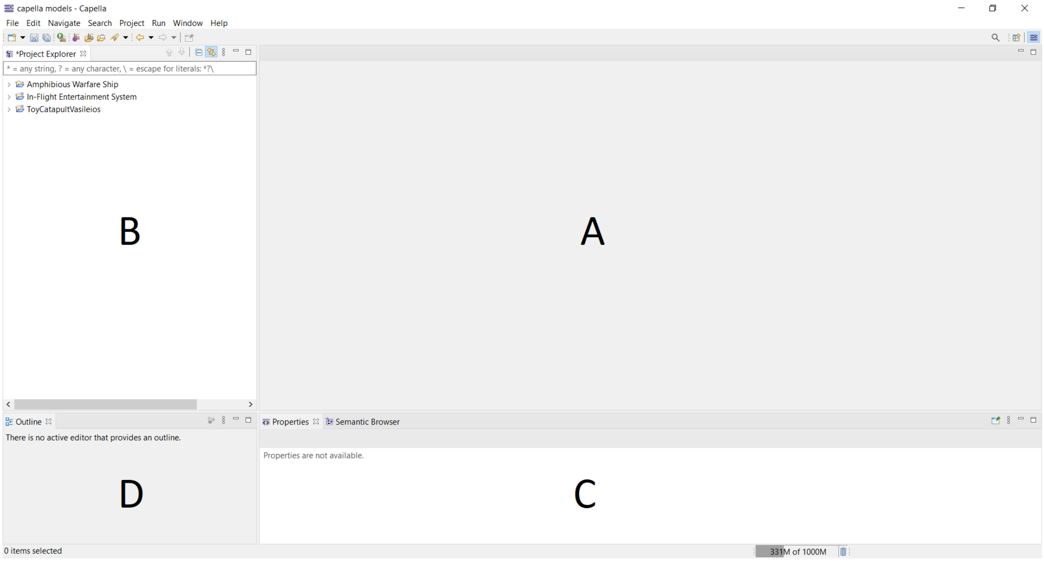
- A: The working area where diagrams and models are built
- B: The project explorer, allowing navigation of project objects
- Two tabs, each serving distinct functionalities:
  - The Properties view displays attributes or properties of the focused modeling object
  - The Semantic Browser view exhibits relations of the object in view with other modelling objects and it allows quick navigation to the related objects
- The Outline view offers a high-level view of diagrams. It is particularly valuable for managing large diagrams and allows easy navigation within them

6.3.3. Operational Analysis

This level focuses on operational users, identifying their goals, activities, constraints, and interactions. It allows the modeling of necessary high-level operational capabilities and conducting operational needs analysis without mentioning the SoI itself (Castro 2023c).

The tool’s guidance, based on the Arcadia methodology, is one of its most notable features. The Workflow window, shown on the left side of Figure 6.3, provides structured guidance for creating diagrams to define the system. Each of the five sequential design stages is represented by a button, and clicking any of these buttons navigates the user to the page for the corresponding stage effortlessly. When the Operational Analysis tab is selected, a new window appears, as illustrated on the right-hand side of Figure 6.3. This specific screenshot highlights several key sections, such as the indication of the selected Arcadia layer, the following Arcadia layer, a display of all diagrams created in this layer, and, of course, access to the specialized diagram editors. Regarding the latter, it lists the main categories of diagrams available at the Operational Analysis level, offering a clear overview of the potential diagram types that can be created during this stage.

By examining the metamodel in Figure 6.1, which will serve as a template for the modeling effort, we can identify the initial elements to be modeled, starting with the system actors. Capella excels at capturing operational entities and actors, as demonstrated in an Operational Entity Breakdown Diagram (OEBD),



* = any string, ? = any character, \ = escape for literals: *?\

There is no active editor that provides an outline.

Properties are not available.

0 items selected

Figure 6.2: User Interface in Capella

which is one of the first diagrams to be defined. For an LPD, a simplified OEBD is depicted in Figure 6.4. The Amphibious Force and the Ship Crew, representing all other personnel on board the ship, are considered operational actors. Meanwhile, the Landing Craft and the Vertical Assault Aircraft are classified as operational entities.

Before defining the operational activities, Capella makes it necessary to capture the motivations, expectations, goals, and objectives of the actors. This can be achieved by capturing Operational Capabilities (OC). This step is essentially an attempt to model the mission within the Capella tool. An Operational Capability Blank diagram (OCB) illustrates the involvement between the user's high-level goal (expressed in an OC) and the operational entities/actors. The OCBD for the LPD example is presented in Figure 6.5. The defined OC in this example is "Conduct Amphibious Raid".

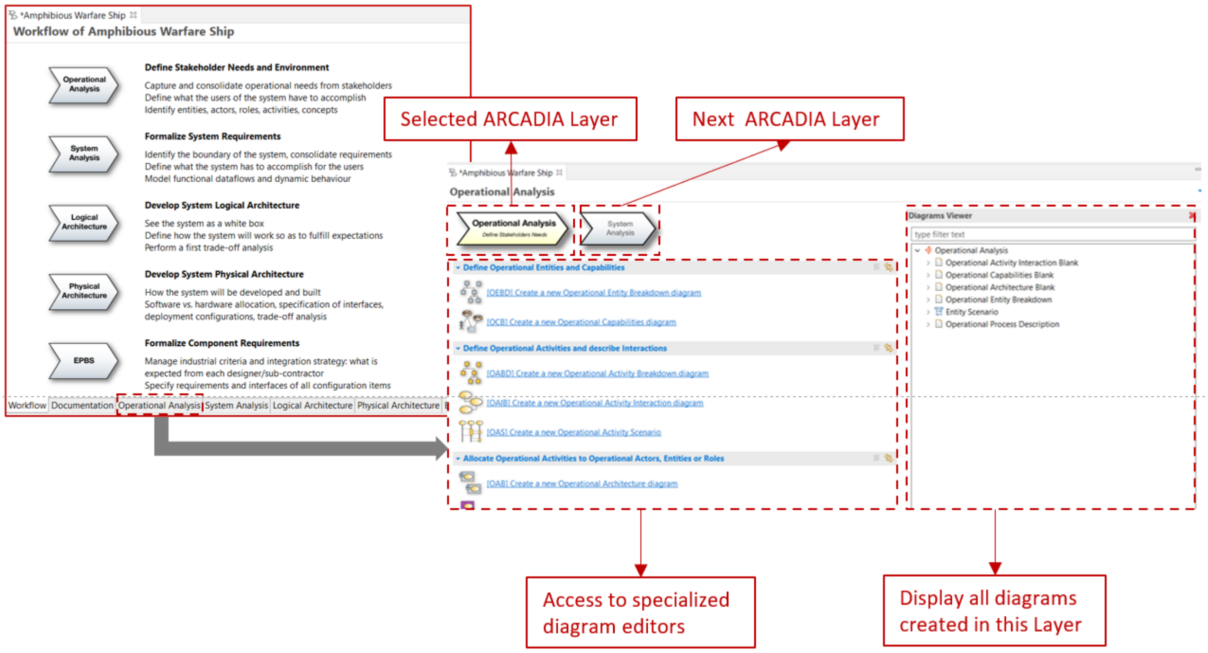
The next step, according to the design template, is defining the operational activities to be performed by the Operational entities and actors. In this case, the relevant operational activities are derived from the stakeholders' requirements, as are the operational activities interactions. Once these activities and interactions are defined, each operational activity is allocated to the actor or entity responsible for performing it.

At this point, it is important to note that in the Arcadia method, capabilities are illustrated using "scenarios," also referred to as "use cases" in other methodologies. In Capella, each capability must be associated with at least one scenario to explain it. Multiple scenarios can be included to elucidate a single capability. In this thesis, further exploration of this feature of Capella was deemed valuable. This is due to the comprehensive nature of designing a new naval ship, which necessitates investigating numerous scenarios to ensure effective performance under various conditions and threats.

In the Capella model presented in this thesis, three scenarios were formulated:

- **Surface Movement Scenario:** This scenario captures the situation in which the movement and planned withdrawal phases are conducted via the sea surface, utilizing Landing Craft.
- **Air Movement Scenario:** In this scenario, the movement and planned withdrawal phases are executed via aerial means, with the assistance of Vertical Assault Aircraft.
- **Emergency Rescue Scenario:** As the title suggests, this scenario represents a situation where an early withdrawal is needed. The movement phase in this scenario is via the Landing Craft, but emergency rescue operations are carried out with the help of Vertical Assault Aircraft.

For each scenario, different diagrams can be created to represent the operational activities, their interactions, and their allocation to the relevant actors/entities. An Operational Architecture Blank (OAB) is the most suitable diagram for including such elements. However, manually creating different OABs for



Workflow of Amphibious Warfare Ship

Define Stakeholder Needs and Environment

Capture and consolidate operational needs from stakeholders
Define what the users of the system have to accomplish
Identify entities, actors, roles, activities, concepts

Formalize System Requirements

Identify the boundary of the system, consolidate requirements
Define what the system has to accomplish for the users
Model functional dataflows and dynamic behaviour

Develop System Logical Architecture

See the system as a white box
Define how the system will work so as to fulfill expectations
Perform a first trade-off analysis

Develop System Physical Architecture

How the system will be developed and built
Software vs. hardware allocation, specification of interfaces, deployment configurations, trade-off analysis

Formalize Component Requirements

Manage industrial criteria and integration strategy: what is expected from each designer/sub-contractor
Specify requirements and interfaces of all configuration items

Define Operational Entities and Capabilities
[OEBD] Create a new Operational Entity Breakdown diagram
[OCB] Create a new Operational Capabilities diagram

Define Operational Activities and describe Interactions
[OABD] Create a new Operational Activity Breakdown diagram
[OAIB] Create a new Operational Activity Interaction diagram
[OAS] Create a new Operational Activity Scenario

Allocate Operational Activities to Operational Actors, Entities or Roles
[OAB] Create a new Operational Architecture diagram

- Operational Analysis
  - Operational Activity Interaction Blank
  - Operational Capabilities Blank
  - Operational Architecture Blank
  - Operational Entity Breakdown
  - Entity Scenario
  - Operational Process Description

Selected ARCADIA Layer

Next ARCADIA Layer

Access to specialized diagram editors

Display all diagrams created in this Layer

Figure 6.3: Workflow window - Capella

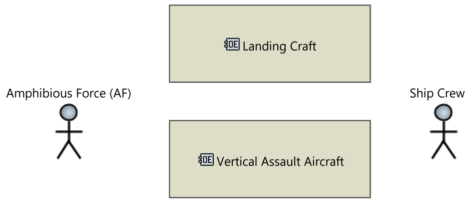


Figure 6.4: [OEBD] Operational Entity Breakdown Diagram for the LPD model in Capella

each scenario can be an intensive process in large and complex projects. Fortunately, Capella offers the "Clone Diagram" command, which significantly reduces the workload by allowing the reuse and editing of previously built diagrams.

In the example case of the LPD, one OAB diagram per scenario has been created in Capella, each illustrating the operational activities, their interactions, and sequence. It is considered that providing additional details on these diagrams does not offer any added value. However, what is deemed important is the creation of an Operational Architecture diagram that serves as a concise description of the entire operational view. This is evident in the OAB diagram of Figure 6.6, which provides a compact and complete representation of the Operational Architecture, summarizing the operational analysis described previously. As shown in this diagram, the operational activities, colored yellow, are allocated to the four main actors/entities, colored in beige. These activities are interconnected through defined interactions, depicted as input-output relationships. The vertical arrangement and arrow links between activities suggest a structured workflow.

What is more, different colored paths can be seen in Figure 6.6. This is because Capella allows the definition of operational processes from operational activities and interactions. An operational process is a sequence of linked activities. The sequence may have multiple start points but must have a single endpoint. Moreover, it cannot include cycles, meaning the same activity cannot be revisited within the same process. In this case, the operational process feature is used to show the sequence of the three different scenarios within the same OAB diagram. This defines the end-to-end activities required to achieve each specific scenario. More importantly, by declaring a group of activities as an operational process, this "chain" is saved by the tool. Consequently, new diagrams, such as the Operational Process Description (OPD) diagrams, can be automatically created to isolate each declared process. This allows for easy reuse

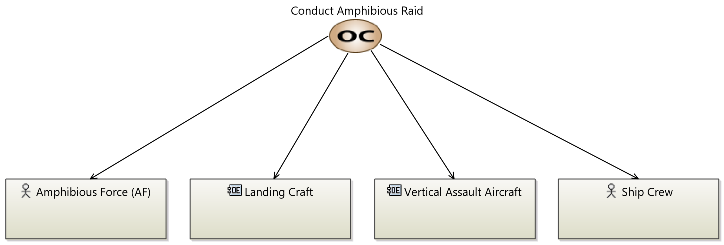


Figure 6.5: [OCB] Operational Capability Blank diagram for the LPD model in Capella

of concepts and further analysis.

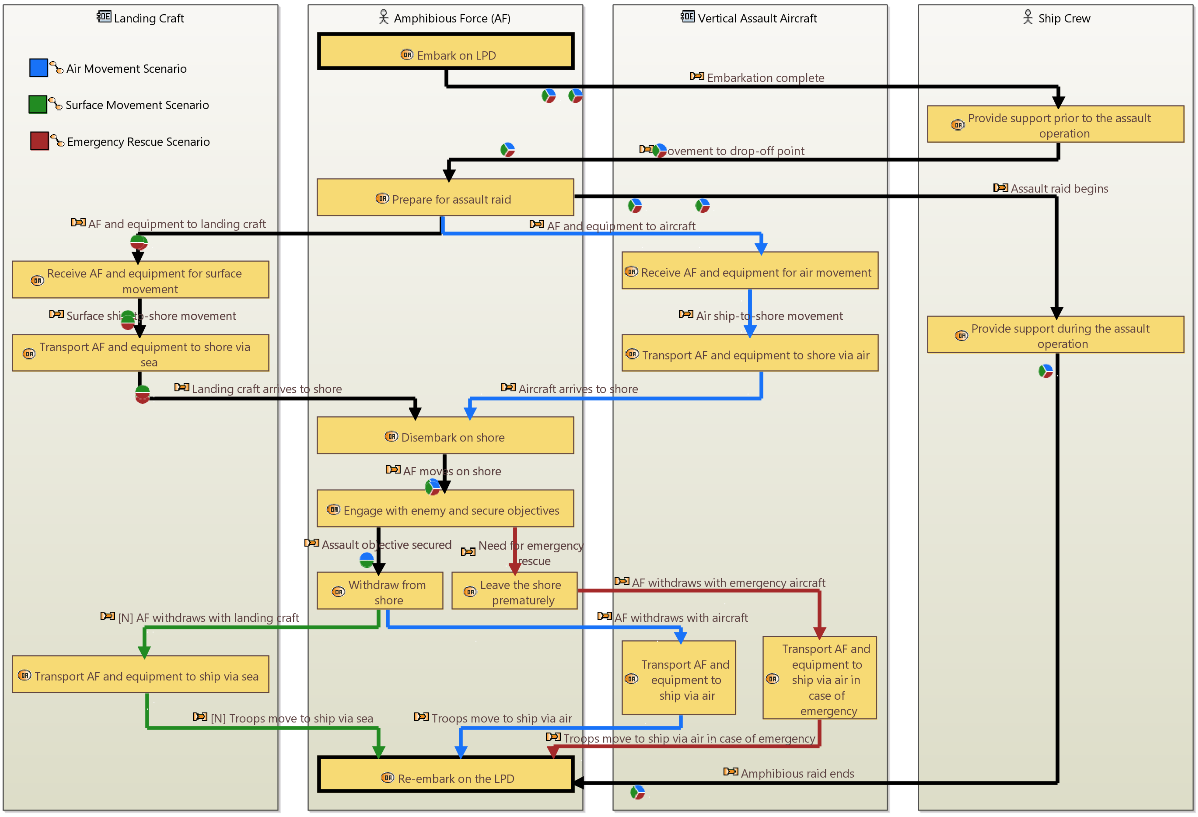

Landing Craft
Amphibious Force (AF)
Vertical Assault Aircraft
Ship Crew
🟦 Air Movement Scenario
🟩 Surface Movement Scenario
🟥 Emergency Rescue Scenario
Embark on LPD
Embarkation complete
Provide support prior to the assault operation
Movement to drop-off point
Prepare for assault raid
Assault raid begins
AF and equipment to landing craft
AF and equipment to aircraft
Receive AF and equipment for surface movement
Receive AF and equipment for air movement
Surface ship-to-shore movement
Air ship-to-shore movement
Provide support during the assault operation
Transport AF and equipment to shore via sea
Transport AF and equipment to shore via air
Landing craft arrives to shore
Aircraft arrives to shore
Disembark on shore
AF moves on shore
Engage with enemy and secure objectives
Assault objective secured
Need for emergency rescue
Withdraw from shore
Leave the shore prematurely
AF withdraws with emergency aircraft
[N] AF withdraws with landing craft
AF withdraws with aircraft
Transport AF and equipment to ship via sea
Transport AF and equipment to ship via air
Transport AF and equipment to ship via air in case of emergency
[N] Troops move to ship via sea
Troops move to ship via air
Troops move to ship via air in case of emergency
Amphibious raid ends
Re-embark on the LPD

Figure 6.6: [OAB] Operational Architecture Blank Diagram illustrating the complete operational view of the LPD model in Capella

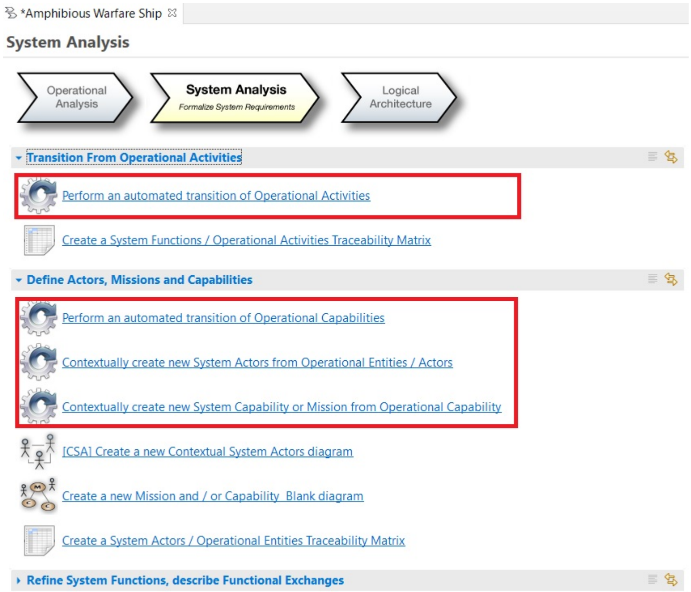
The analysis above provides only a summary of part of the modeling effort in the Operational Analysis phase. Additionally, diagrams such as the Entity Scenario (ES) and Operational Activity Interaction Blank (OAIB) were also created, although their detailed exploration is deferred to the Appendix in Chapter 9.

6.3.4. System Analysis

The System Analysis layer aims to define how the system can satisfy the former operational needs by focusing on it as a black box. It forms an external functional analysis, that is constructed based on the operational analysis from the previous layer and textual input requirements and is outlined in relation to them. The design remains open, with the analysis of how it functions reserved for later design phases (Castro 2023a).

An initial diagram generated during System Analysis is the Context System Actors (CSA). This diagram represents the SoI along with its actors. To create this diagram, it is necessary to transition all the operational entities and actors identified and documented during the Operational Analysis to the System

Analysis phase. Capella offers automated or step-by-step transitions to facilitate this swift. Figure 6.7 demonstrates the tool's capabilities in the transition dialog box used for transitioning to System Analysis, as illustrated in the example case of the LPD. It is underlined that the author observed occasional errors in the automated transitions during their modeling effort.



*Amphibious Warfare Ship

System Analysis

Operational Analysis | System Analysis *Formalize System Requirements* | Logical Architecture

Transition From Operational Activities

- Perform an automated transition of Operational Activities
- Create a System Functions / Operational Activities Traceability Matrix

Define Actors, Missions and Capabilities

- Perform an automated transition of Operational Capabilities
- Contextually create new System Actors from Operational Entities / Actors
- Contextually create new System Capability or Mission from Operational Capability
- [CSA] Create a new Contextual System Actors diagram
- Create a new Mission and / or Capability Blank diagram
- Create a System Actors / Operational Entities Traceability Matrix

Refine System Functions, describe Functional Exchanges

Figure 6.7: Transition feature in Capella for the LPD model in Capella

In the case of the LPD, the CSA diagram has been formulated and is shown in Figure 6.8. This diagram is synchronized and automatically updated with any new actors identified as the design process evolves.

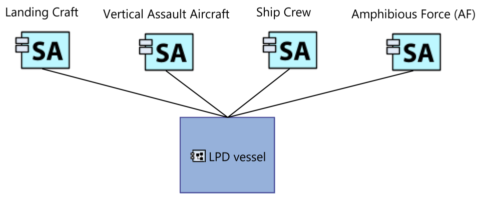


Figure 6.8: [CSA] Context System Actors for the LPD model in Capella

Similar to the actors, the operational capabilities need to be transitioned to the system phase. As soon as this is done, the mission of the system can be modeled in new diagrams, the Mission Blank and the Mission Capabilities Blank (MCB) diagrams. These diagrams have been constructed for the LPD case, with the MCB presented in Figure 6.9. It can be concluded that the operational capability identified in the Operational Analysis phase has now become the mission in the System Analysis phase. The newly defined capabilities, representing overall objectives from the user's perspective, include the following scenarios: Conduct Assault Operations from Air, Conduct Assault Operations from Sea, and Provide Emergency Rescue.

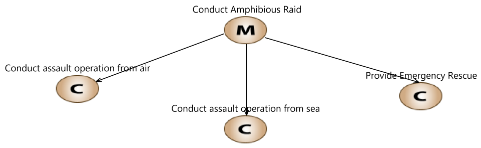

Conduct Amphibious Raid
M
Conduct assault operation from air
C
Provide Emergency Rescue
C
Conduct assault operation from sea
C

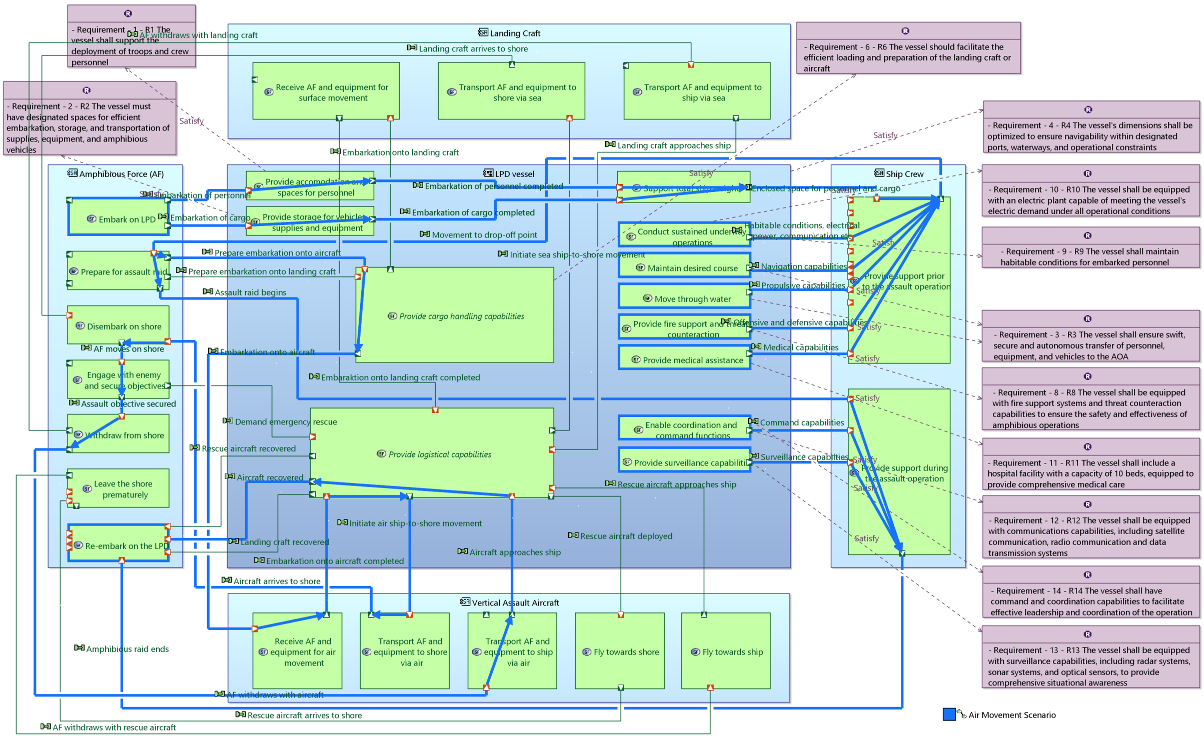
Figure 6.9: [MCB] Mission Capabilities Blank for the LPD model in Capella

As the transition to System Analysis progresses, the design template in Figure 6.1 also indicates moving from Operational Activities to System Functions. The operational activities identified during the Operational Analysis can be transitioned and refined into system functions.

Regarding the emergence of new system functions, the approach remains consistent with that used in the Operational Analysis phase. Specifically, System Architecture Blank (SAB) diagrams are utilized to define the (new or transitioned) system functions, their functional exchanges, and their allocation to system components and actors. Moreover, when transitioning operational activities to system functions, it is necessary to consider whether each activity will be realized partially, not at all, or entirely by the system. This can be formalized at the System Analysis level as follows:

- Partially: The activity must be decomposed into functions, with some allocating to the system and others to actors
- Not at all: The activity becomes a function allocated to an actor
- Entirely: The activity becomes a system function of the same name

One SAB diagram has been created to justify each one of the three selected capabilities/scenarios. Providing additional details on these diagrams is not considered beneficial. What is important is the creation of a SAB. The latter offers a concise description of the entire system view. This is exemplified in the SAB diagram in Figure 6.10, which delivers a compact and comprehensive representation of the System Architecture, summarizing the previously described system analysis.



- Requirement - 1 - R1 The vessel shall support the deployment of troops and crew personnel
- Requirement - 2 - R2 The vessel must have designated spaces for efficient embarkation, storage, and transportation of supplies, equipment, and amphibious vehicles
- Requirement - 6 - R6 The vessel should facilitate the efficient loading and preparation of the landing craft or aircraft
- Requirement - 4 - R4 The vessel's dimensions shall be optimized to ensure navigability within designated ports, waterways, and operational constraints
- Requirement - 10 - R10 The vessel shall be equipped with an electric plant capable of meeting the vessel's electric demand under all operational conditions
- Requirement - 9 - R9 The vessel shall maintain habitable conditions for embarked personnel
- Requirement - 3 - R3 The vessel shall ensure swift, secure and autonomous transfer of personnel, equipment, and vehicles to the AOA
- Requirement - 8 - R8 The vessel shall be equipped with fire support systems and threat counteraction capabilities to ensure the safety and effectiveness of amphibious operations
- Requirement - 11 - R11 The vessel shall include a hospital facility with a capacity of 10 beds, equipped to provide comprehensive medical care
- Requirement - 12 - R12 The vessel shall be equipped with communications capabilities, including satellite communication, radio communication and data transmission systems
- Requirement - 14 - R14 The vessel shall have command and coordination capabilities to facilitate effective leadership and coordination of the operation
- Requirement - 13 - R13 The vessel shall be equipped with surveillance capabilities, including radar systems, sonar systems, and optical sensors, to provide comprehensive situational awareness
Satisfy
Landing Craft
AF withdraws with landing craft
Landing craft arrives to shore
Receive AF and equipment for surface movement
Transport AF and equipment to shore via sea
Transport AF and equipment to ship via sea
Embarkation onto landing craft
Landing craft approaches ship
Amphibious Force (AF)
Embark on LPD
Embarkation of personnel
Embarkation of cargo
Prepare for assault raid
Disembark on shore
AF moves on shore
Engage with enemy and secure objectives
Assault objective secured
Withdraw from shore
Leave the shore prematurely
Re-embark on the LPD
Amphibious raid ends
LPD vessel
Provide accomodation and spaces for personnel
Provide storage for vehicles supplies and equipment
Embarkation of personnel completed
Embarkation of cargo completed
Movement to drop-off point
Prepare embarkation onto aircraft
Prepare embarkation onto landing craft
Initiate sea ship-to-shore movement
Assault raid begins
Provide cargo handling capabilities
Embarkation onto aircraft
Embaraktion onto landing craft completed
Demand emergency rescue
Rescue aircraft recovered
Aircraft recovered
Provide logistical capabilities
Initiate air ship-to-shore movement
Landing craft recovered
Embarkation onto aircraft completed
Aircraft approaches ship
Rescue aircraft deployed
Rescue aircraft approaches ship
Support total ship weight
Enclosed space for personnel and cargo
Conduct sustained underway operations
Habitable conditions, electrical power, communication etc
Maintain desired course
Navigation capabilities
Move through water
Propulsive capabilities
Provide fire support and threat counteraction
Offensive and defensive capabilities
Provide medical assistance
Medical capabilities
Enable coordination and command functions
Command capabilities
Provide surveillance capabilities
Surveillance capabilties
Ship Crew
Provide support prior to the assault operation
Provide support during the assault operation
Aircraft arrives to shore
Vertical Assault Aircraft
Receive AF and equipment for air movement
Transport AF and equipment to shore via air
Transport AF and equipment to ship via air
Fly towards shore
Fly towards ship
AF withdraws with aircraft
Rescue aircraft arrives to shore
AF withdraws with rescue aircraft
Air Movement Scenario

Figure 6.10: [SAB] System Architecture Blank Diagram illustrating the complete system view for the LPD model in Capella

To keep the analysis of Figure 6.10 straightforward, only its key aspects are discussed. The system is modeled and presented in dark blue in the center of the diagram, containing several allocated functions, known as system functions. These include leaf functions (small green rectangles) and higher-level functions (larger green rectangles). For higher-level functions, distinct Functional Dataflow Blank Diagram (SDFB) diagrams enable further analysis of their sub-functions and relevant functional exchanges.

Interactions between functions, termed system functional exchanges, are represented by green arrows. These exchanges and the dependencies between functions form the functional data flow. A functional chain, shown in dark blue, represents a specific path within this flow, useful for describing expected system behavior and guiding verification/validation tests. In Figure 6.10, only one functional chain (out of there) is represented in dark blue, specifically the Air Movement Scenario.

Additionally, in Figure 6.10, the system (depicted in darker blue) is surrounded by actors (depicted in lighter blue), each containing functions allocated to them. Notably, there are pink boxes representing high-level requirements derived from the mission description. To visually present these requirements, an add-on called "Requirements Viewpoint" was utilized. Although the requirements were imported manually, this process can be automated. The requirements were then mapped to specific functions within the system using a "satisfy" relationship.

The preceding analysis offers a concise overview of a segment of the modeling endeavor within the System Analysis phase. Additionally, diagrams like the Missions Blank (MB), System Data Flow Blank (SDFB), and System Function Breakdown were also crafted, although their detailed exploration is deferred to the Appendix in Chapter 9. With all the essential modeling elements for System Analysis, as indicated by the metamodel in Figure 6.1, now in place, the transition to the Logical phase can commence.

6.3.5. Logical Architecture

It is reminded that the system is considered a black box in the previous levels. The time to define the subsystems has arrived.

First, the capabilities must be realized in the Logical Architecture. Thus, one of the initial activities involves continuing the work performed at the System level. It should be noted that transitions of the System Capabilities can be automatically performed in Capella using the Activity Explorer.

In addition to the capabilities, the system functions should also be transitioned into new logical functions. A functional refinement process is conducted to further detail functions deemed essential. As before, a hierarchy for the new logical functions must be established. Figure 6.11 illustrates how the system functions have been refined into several logical functions or grouped into different functional groups within a Functional Breakdown diagram (LFBD). The white-colored functions represent the transitioned system functions.

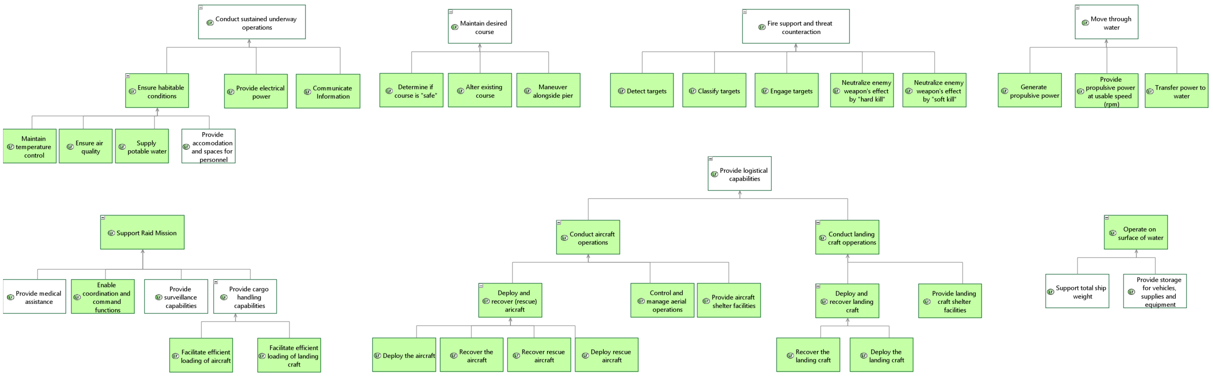

Conduct sustained underway operations
Ensure habitable conditions
Provide electrical power
Communicate Information
Maintain temperature control
Ensure air quality
Supply potable water
Provide accomodation and spaces for personnel
Maintain desired course
Determine if course is "safe"
Alter existing course
Maneuver alongside pier
Fire support and threat counteraction
Detect targets
Classify targets
Engage targets
Neutralize enemy weapon's effect by "hard kill"
Neutralize enemy weapon's effect by "soft kill"
Move through water
Generate propulsive power
Provide propulsive power at usable speed (rpm)
Transfer power to water
Provide logistical capabilities
Support Raid Mission
Provide medical assistance
Enable coordination and command functions
Provide surveillance capabilities
Provide cargo handling capabilities
Facilitate efficient loading of aircraft
Facilitate efficient loading of landing craft
Conduct aircraft operations
Deploy and recover (rescue) aricraft
Deploy the aircraft
Recover the aircraft
Recover rescue aircraft
Deploy rescue aircraft
Control and manage aerial operations
Provide aircraft shelter facilities
Conduct landing craft opperations
Deploy and recover landing craft
Recover the landing craft
Deploy the landing craft
Provide landing craft shelter facilities
Operate on surface of water
Support total ship weight
Provide storage for vehicles, supplies and equipment

Figure 6.11: [LFBD] Functional Breakdown Diagram for the LPD model in Capella

As soon as the functions are established, functional exchanges can be defined among them. It is important to mention that only "leaf" functions should have functional exchanges. In this thesis, this has been done by creating a Logical Dataflow Blank (LDFB) diagram which involves all the logical functions along with their functional exchanges.

Following that, the system must be decomposed into its subsystems so that the logical functions can be mapped to them. It is noteworthy that exchanges between the newly defined subsystems can now be modeled as component exchanges. This was done primarily in the LPD model to explore the tool's capabilities in this area. However, more detailed information on this process will not be provided in the current thesis. Additionally, it is pointed out that the processes of defining the subsystems as well as the containment relationships and refining the functions have been done in parallel. There were many back-and-forth adjustments to each diagram to ensure that all functions could be mapped to subsystems. The final structure of the system, after decomposing it into subsystems, is illustrated in the Logical

Component Breakdown diagram (LCBD) in Figure 6.12. As shown here, the system is decomposed into seven logical components, referred to as subsystems. These subsystems are further decomposed into more detailed logical components.

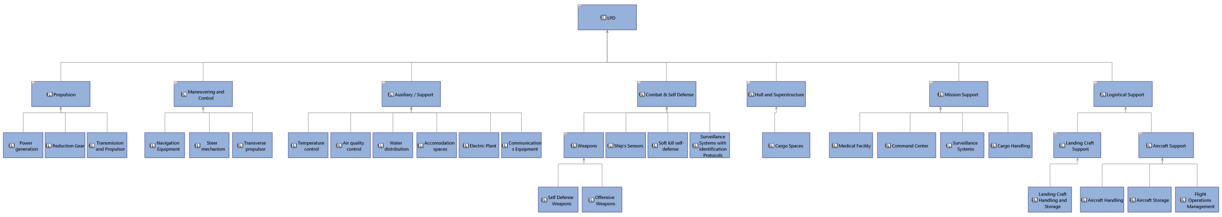


Figure 6.12: [LCBD] Logical Component Breakdown diagram for the LPD model in Capella

Next, the mapping of Logical Functions to the logical components takes place, which can be done using a Logical Architecture Blank (LAB) diagram. Three LABs were created, one for each of the three scenarios. The complete logical view is summarized in the LAB shown in Figure 6.13, representing the Logical Architecture of the LPD. It is worth noting that the mapping of requirements to the subsystems has been intentionally omitted at this layer to streamline the analysis. This decision does not impact the final structure of the systems, as a detailed requirement mapping will occur in the subsequent architectural layer.

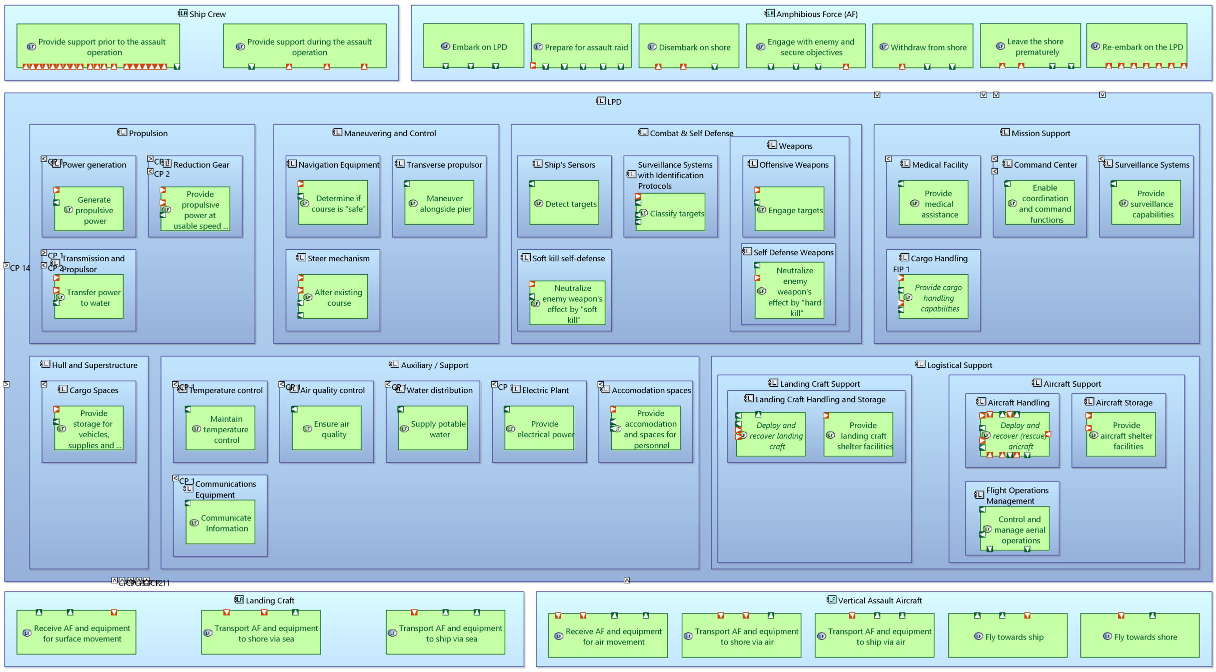

Ship Crew
Provide support prior to the assault operation
Provide support during the assault operation
Amphibious Force (AF)
Embark on LPD
Prepare for assault raid
Disembark on shore
Engage with enemy and secure objectives
Withdraw from shore
Leave the shore prematurely
Re-embark on the LPD
LPD
CP 14
Propulsion
Power generation
Generate propulsive power
Reduction Gear
CP 2
Provide propulsive power at usable speed ...
Transmission and Propulsor
Transfer power to water
Maneuvering and Control
Navigation Equipment
Determine if course is "safe"
Transverse propulsor
Maneuver alongside pier
Steer mechanism
Alter existing course
Combat & Self Defense
Ship's Sensors
Detect targets
Surveillance Systems with Identification Protocols
Classify targets
Weapons
Offensive Weapons
Engage targets
Soft kill self-defense
Neutralize enemy weapon's effect by "soft kill"
Self Defense Weapons
Neutralize enemy weapon's effect by "hard kill"
Mission Support
Medical Facility
Provide medical assistance
Command Center
Enable coordination and command functions
Surveillance Systems
Provide surveillance capabilities
Cargo Handling
FIP 1
Provide cargo handling capabilities
Hull and Superstructure
Cargo Spaces
Provide storage for vehicles, supplies and ...
Auxiliary / Support
Temperature control
Maintain temperature control
Air quality control
Ensure air quality
Water distribution
Supply potable water
Electric Plant
Provide electrical power
Accomodation spaces
Provide accomodation and spaces for personnel
Communications Equipment
Communicate Information
Logistical Support
Landing Craft Support
Landing Craft Handling and Storage
Deploy and recover landing craft
Provide landing craft shelter facilities
Aircraft Support
Aircraft Handling
Deploy and recover (rescue) aircraft
Aircraft Storage
Provide aircraft shelter facilities
Flight Operations Management
Control and manage aerial operations
Landing Craft
Receive AF and equipment for surface movement
Transport AF and equipment to shore via sea
Transport AF and equipment to ship via sea
Vertical Assault Aircraft
Receive AF and equipment for air movement
Transport AF and equipment to shore via air
Transport AF and equipment to ship via air
Fly towards ship
Fly towards shore

Figure 6.13: [LAB] Logical Architecture Blank Diagram illustrating the complete logical view of the LPD model in Capella

It is acknowledged that the model has started to appear complex, even with these sample functions. To enhance the focus on each of the subsystems, ensuring the tool's internal connectivity and consistency is pivotal. Thus, various SAB diagrams were created to offer a more detailed analysis of each subsystem. These diagrams, along with other supplementary diagrams, are available in the Appendix, which can be found in Chapter 9. With all modeling elements as dictated by the metamodel of Figure 6.1 constructed for the logical layer, we can now proceed to the Physical Architecture.

6.3.6. Physical Architecture

To formulate the Physical Architecture based on the Logical level, the tool proposes using a transition similar to those used while transitioning from Operational Analysis to System Analysis and from System

Analysis to Logical Architecture. As a result, we can create as many physical functions as logical functions while also maintaining the functional exchanges and functional chains related to them.

Once again, the capabilities must be realized in the Physical Architecture. The transition of logical capabilities can be automatically performed in Capella using the Activity Explorer. Next to this, the logical functions have to be transitioned into new physical functions. A functional refinement process is conducted to further detail functions deemed essential. As before, a functional refinement process is conducted to define the new hierarchy for the new physical functions. The Functional Breakdown diagram (PFBD) in Figure 6.14 illustrates how the logical functions have been refined into several physical functions, with white boxes representing the transitioned logical functions.

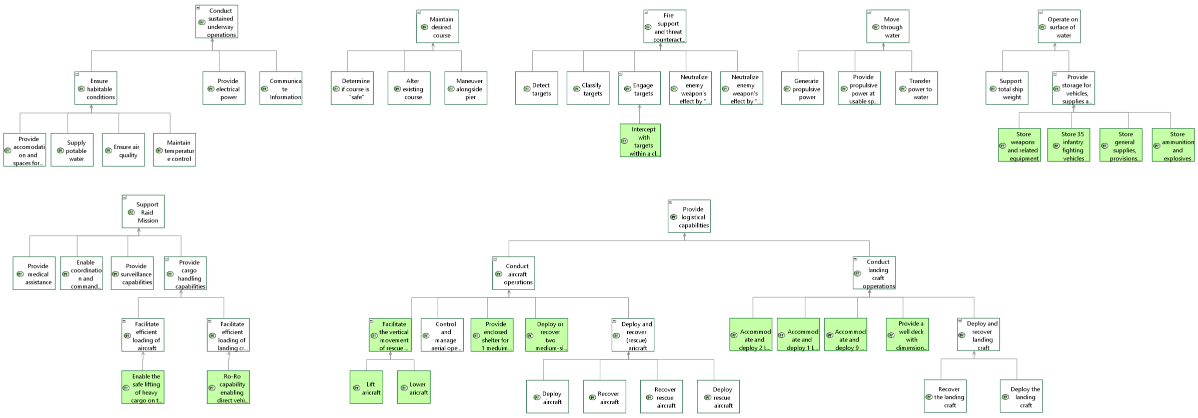

Conduct sustained underway operations
Ensure habitable conditions
Provide electrical power
Communicate Information
Provide accomodation and spaces for...
Supply potable water
Ensure air quality
Maintain temperature control
Maintain desired course
Determine if course is "safe"
Alter existing course
Maneuver alongside pier
Fire support and threat counteract...
Detect targets
Classify targets
Engage targets
Neutralize enemy weapon's effect by "...
Neutralize enemy weapon's effect by "...
Intercept with targets within a cl...
Move through water
Generate propulsive power
Provide propulsive power at usable sp...
Transfer power to water
Operate on surface of water
Support total ship weight
Provide storage for vehicles, supplies a...
Store weapons and related equipment
Store 35 infantry fighting vehicles
Store general supplies, provisions...
Store ammunition and explosives
Support Raid Mission
Provide medical assistance
Enable coordination and command...
Provide surveillance capabilities
Provide cargo handling capabilities
Facilitate efficient loading of aircraft
Facilitate efficient loading of landing cr...
Enable the safe lifting of heavy cargo on t...
Ro-Ro capability enabling direct vehi...
Provide logistical capabilities
Conduct aircraft operations
Facilitate the vertical movement of rescue ...
Control and manage aerial ope...
Provide enclosed shelter for 1 meduim...
Deploy or recover two medium-si...
Deploy and recover (rescue) aricraft
Lift aricraft
Lower aricraft
Deploy aircraft
Recover aircraft
Recover rescue aircraft
Deploy rescue aircraft
Conduct landing craft opperations
Accommodate and deploy 2 L...
Accommodate and deploy 1 L...
Accommodate and deploy 9 ...
Provide a well deck with dimension...
Deploy and recover landing craft
Recover the landing craft
Deploy the landing craft

Figure 6.14: [PFBD] Functional Breakdown Diagram for the LPD model in Capella

Similarly, as soon as the functions are established, functional exchanges, as well as new functional chains, can be defined among them. For the LPD problem, this has been done by creating a Logical Dataflow Blank (PDFB) diagram which involves all the physical functions along with their functional exchanges. It is pointed out that the author encountered difficulties adjusting the transitioned functional chains to incorporate the newly created physical functions. Consequently, new functional chains were created in the physical layer. However, this process was relatively straightforward.

Now it is time to deploy the actual concrete components that comprise the system. In Arcadia, there are two types of physical components (Castro 2023b):

1. Behaviour physical component: responsible for executing some of the assigned functions by interacting with other behavior components and external actors.
2. Node physical component: hosts a specific number of behavior components, providing them with the necessary resources to function.

Contrary to the previous architectural layer, where it was strongly advised to exclude all technological considerations, at the Physical Architecture layer, technological considerations are taken into account. By considering these factors, the components in the physical layer have been identified. The breakdown of the behaviour physical components is depicted in Figure 6.15, while the breakdown of the node physical components is depicted in Figure 6.16. To maintain consistency with the metamodel of Figure 6.1, it is important to emphasize that this thesis primarily focuses on the behavioral components. The node physical components are simply viewed as a placeholder for the latter. Therefore, it is assumed that the behaviour components represent the modeling element named equipment.

Once functions and components have been identified and documented, physical functions can be assigned to the appropriate behavioural physical components (equipment). This allocation process can be achieved by creating a Physical Architecture Blank (PAB) diagram. However, creating three different PAB diagrams, one for each scenario, was deemed to provide no further value and was omitted. Instead, a comprehensive physical view was generated in a single PAB diagram. Consequently, the final product of the physical phase, the Physical Architecture, is depicted in Figure 6.17.

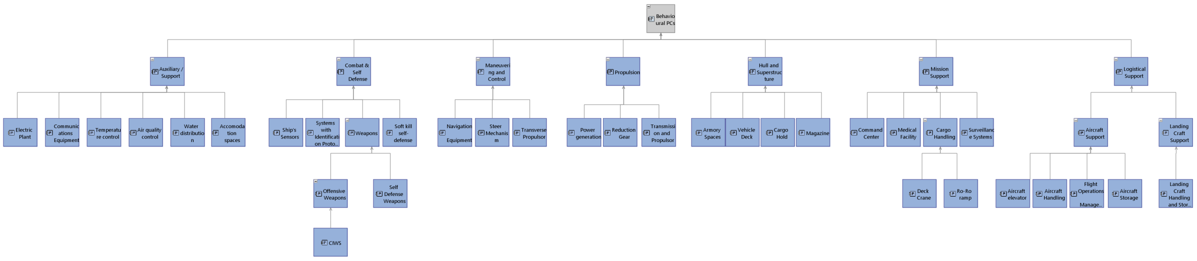


Figure 6.15: [PCBD] Physical Component Breakdown Diagram (behaviour physical components)for the LPD model in Capella

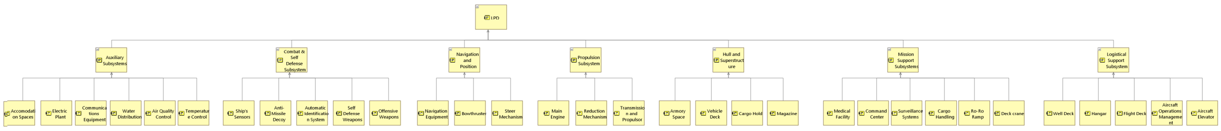


Figure 6.16: [PCBD] Physical Component Breakdown Diagram (node physical components) for the LPD model in Capella

Although the diagram in Figure 6.17 may appear complex at first glance, it is based on the same modeling logic used for SAB and LAB in the previous design phases. The system is depicted by the large white rectangle in the center, surrounded by smaller light blue rectangles representing the four main actors, each executing functions highlighted in green. Within the system, several yellow boxes represent the node physical components, clearly illustrating containment relationships. It is also worth noting that the structure of the previous layer with the seven predefined subsystems is maintained. Within these boxes are blue boxes representing the behaviour physical components, also referred to as equipment in this thesis. Each of these contains physical functions, depicted in green. The omission of functional exchanges, component exchanges, and physical exchanges was intentional to maintain straightforward analysis. Similar to the SAB in the system phase, the requirements are included in the perimeter of the diagram using the “Requirements Viewpoint” add-on. Higher-level requirements are colored in pink, while lower-level requirements are colored in gray and white, with white denoting more refined requirements. The higher-level requirements are mapped to the lower-level requirements with a “derive” relationship, while the refined lower-level requirements are linked to specific equipment with a “satisfy” relationship. It is crucial to highlight that the decomposition throughout the entire design process was conducted to ensure that at least one physical function is allocated to each piece of equipment. Since the requirements are primarily at a functional level, this ensures complete traceability between the requirements and the equipment.

6.3.7. Verification & validation of the Capella model

This section discusses model verification (Step 7: Verify & Validate MBSE Models) as detailed in Chapter 4. It is important to mention that only a summary of the Verification & Validation process followed for the LPD model is presented below. A more comprehensive exploration of the Verification & Validation step in Capella is available in Chapter 7. Although it would fit well here and strengthen the current section, it was selected to be described in detail there because model validation and verification are deemed crucial for evaluating consistency and traceability, respectively.

Verification involves confirming if the vessel model accurately reflects the example requirements listed in Table 5.1. This is achieved through the manual creation of requirements, via the Requirements Viewpoint add-on, and linking them to Capella elements. While straightforward requirements, necessitating specific equipment, can be visually verified by incorporating specific technological solutions, those involving dimensions or arithmetic comparisons are more difficult to verify in Capella. In this example, such requirements are only linked to relevant elements for visual traceability, acknowledging limitations in parametric data.

During the validation phase, the tool's built-in validation rules are utilized to ensure adherence to language constructs and identify mistakes. Capella's “Validate model” command organizes validation rules

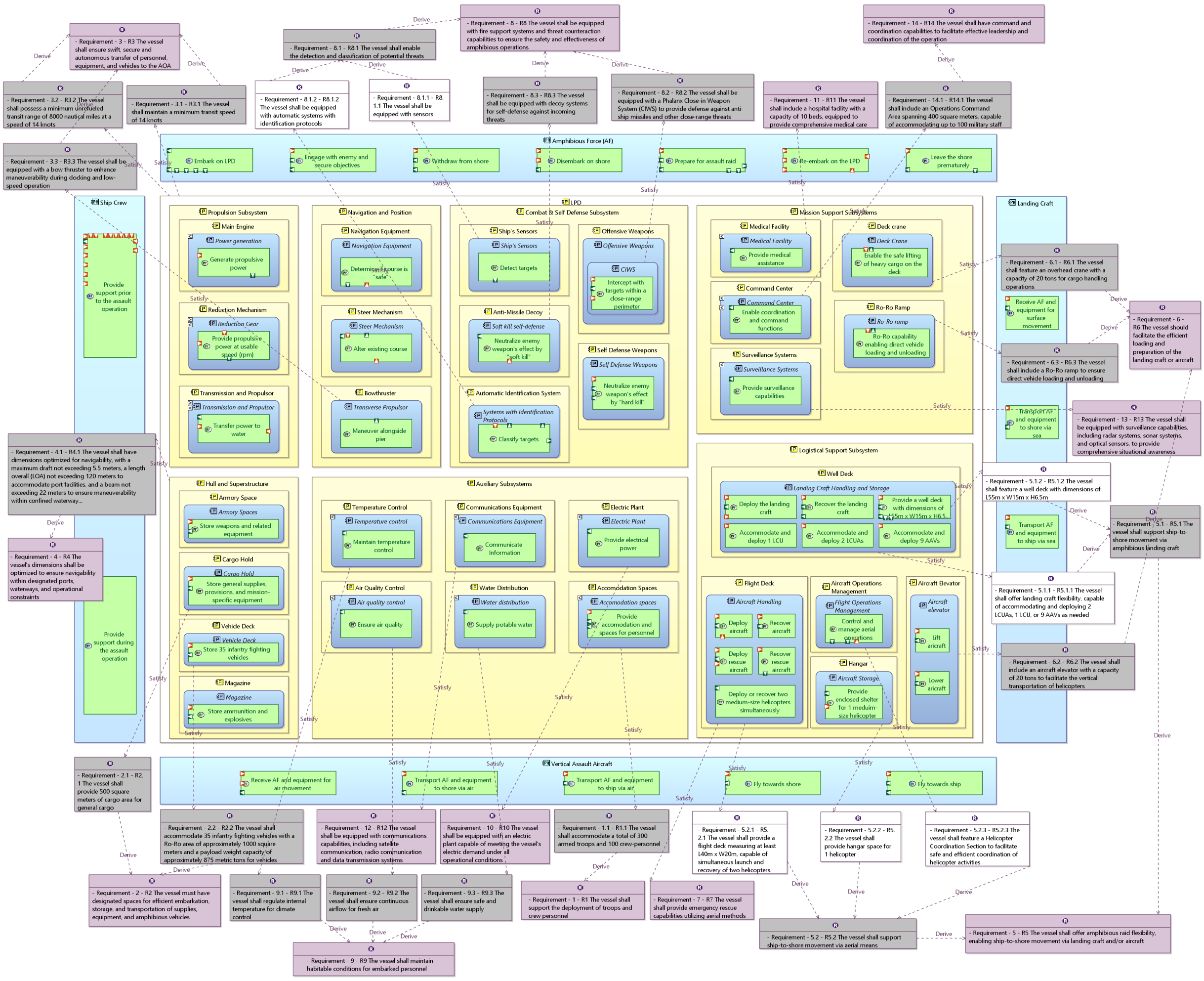

- Requirement - 3 - R3 The vessel shall ensure swift, secure and autonomous transfer of personnel, equipment, and vehicles to the AOA
- Requirement - 8.1 - R8.1 The vessel shall enable the detection and classification of potential threats
- Requirement - 8 - R8 The vessel shall be equipped with fire support systems and threat counteraction capabilities to ensure the safety and effectiveness of amphibious operations
- Requirement - 14 - R14 The vessel shall have command and coordination capabilities to facilitate effective leadership and coordination of the operation
- Requirement - 3.2 - R3.2 The vessel shall possess a minimum unrefueled transit range of 8000 nautical miles at a speed of 14 knots
- Requirement - 3.1 - R3.1 The vessel shall maintain a minimum transit speed of 14 knots
- Requirement - 8.1.2 - R8.1.2 The vessel shall be equipped with automatic systems with identification protocols
- Requirement - 8.1.1 - R8.1.1 The vessel shall be equipped with sensors
- Requirement - 8.3 - R8.3 The vessel shall be equipped with decoy systems for self-defense against incoming threats
- Requirement - 8.2 - R8.2 The vessel shall be equipped with a Phalanx Close-in Weapon System (CIWS) to provide defense against anti-ship missiles and other close-range threats
- Requirement - 11 - R11 The vessel shall include a hospital facility with a capacity of 10 beds, equipped to provide comprehensive medical care
- Requirement - 14.1 - R14.1 The vessel shall include an Operations Command Area spanning 400 square meters, capable of accommodating up to 100 military staff
- Requirement - 3.3 - R3.3 The vessel shall be equipped with a bow thruster to enhance maneuverability during docking and low-speed operation
Amphibious Force (AF)
Embark on LPD
Engage with enemy and secure objectives
Withdraw from shore
Disembark on shore
Prepare for assault raid
Re-embark on the LPD
Leave the shore prematurely
Ship Crew
Provide support prior to the assault operation
Provide support during the assault operation
LPD
Propulsion Subsystem
Main Engine
Power generation
Generate propulsive power
Reduction Mechanism
Reduction Gear
Provide propulsive power at usable speed (rpm)
Transmission and Propulsor
Transmission and Propulsor
Transfer power to water
Navigation and Position
Navigation Equipment
Navigation Equipment
Determine if course is "safe"
Steer Mechanism
Steer Mechanism
Alter existing course
Bowthruster
Transverse Propulsor
Maneuver alongside pier
Combat & Self Defense Subsystem
Ship's Sensors
Ship's Sensors
Detect targets
Anti-Missile Decoy
Soft kill self-defense
Neutralize enemy weapon's effect by "soft kill"
Automatic Identification System
Systems with Identification Protocols
Classify targets
Offensive Weapons
Offensive Weapons
CIWS
Intercept with targets within a close-range perimeter
Self Defense Weapons
Self Defense Weapons
Neutralize enemy weapon's effect by "hard kill"
Mission Support Subsystems
Medical Facility
Medical Facility
Provide medical assistance
Command Center
Command Center
Enable coordination and command functions
Surveillance Systems
Surveillance Systems
Provide surveillance capabilities
Deck crane
Deck Crane
Enable the safe lifting of heavy cargo on the deck
Ro-Ro Ramp
Ro-Ro ramp
Ro-Ro capability enabling direct vehicle loading and unloading
Landing Craft
- Requirement - 6.1 - R6.1 The vessel shall feature an overhead crane with a capacity of 20 tons for cargo handling operations
Receive AF and equipment for surface movement
- Requirement - 6 - R6 The vessel should facilitate the efficient loading and preparation of the landing craft or aircraft
- Requirement - 6.3 - R6.3 The vessel shall include a Ro-Ro ramp to ensure direct vehicle loading and unloading
Transport AF and equipment to shore via sea
- Requirement - 13 - R13 The vessel shall be equipped with surveillance capabilities, including radar systems, sonar systems, and optical sensors, to provide comprehensive situational awareness
- Requirement - 4.1 - R4.1 The vessel shall have dimensions optimized for navigability, with a maximum draft not exceeding 5.5 meters, a length overall (LOA) not exceeding 120 meters to accommodate port facilities, and a beam not exceeding 22 meters to ensure maneuverability within confined waterway...
- Requirement - 4 - R4 The vessel's dimensions shall be optimized to ensure navigability within designated ports, waterways, and operational constraints
Hull and Superstructure
Armory Space
Armory Spaces
Store weapons and related equipment
Cargo Hold
Cargo Hold
Store general supplies, provisions, and mission specific equipment
Vehicle Deck
Vehicle Deck
Store 35 infantry fighting vehicles
Magazine
Magazine
Store ammunition and explosives
Auxiliary Subsystems
Temperature Control
Temperature control
Maintain temperature control
Communications Equipment
Communications Equipment
Communicate information
Electric Plant
Electric Plant
Provide electrical power
Air Quality Control
Air quality control
Ensure air quality
Water Distribution
Water distribution
Supply potable water
Accomodation Spaces
Accomodation spaces
Provide accomodation and spaces for personnel
Logistical Support Subsystem
Well Deck
Landing Craft Handling and Storage
Deploy the landing craft
Recover the landing craft
Provide a well deck with dimensions of 155m x W15m x H6.5m
Accommodate and deploy 1 LCU
Accommodate and deploy 2 LCUAs
Accommodate and deploy 9 AAVs
Flight Deck
Aircraft Handling
Deploy aircraft
Recover aircraft
Deploy rescue aircraft
Recover rescue aircraft
Deploy or recover two medium-size helicopters simultaneously
Aircraft Operations Management
Flight Operations Management
Control and manage aerial operations
Hangar
Aircraft Storage
Provide enclosed shelter for 1 medium-size helicopter
Aircraft Elevator
Aircraft elevator
Lift aircraft
Lower aircraft
- Requirement - 5.1.2 - R5.1.2 The vessel shall feature a well deck with dimensions of L55m x W15m x H6.5m
Transport AF and equipment to ship via sea
- Requirement - 5.1 - R5.1 The vessel shall support ship-to-shore movement via amphibious landing craft
- Requirement - 5.1.1 - R5.1.1 The vessel shall offer landing craft flexibility, capable of accommodating and deploying 1 LCU, 2 LCUAs, or 9 AAVs as needed
- Requirement - 6.2 - R6.2 The vessel shall include an aircraft elevator with a capacity of 20 tons to facilitate the vertical transportation of helicopters
- Requirement - 2.1 - R2.1 The vessel shall provide 500 square meters of cargo area for general cargo
Vertical Assault Aircraft
Receive AF and equipment for air movement
Transport AF and equipment to shore via air
Transport AF and equipment to ship via air
Fly towards shore
Fly towards ship
- Requirement - 2.2 - R2.2 The vessel shall accommodate 35 infantry fighting vehicles with a Ro-Ro area of approximately 1000 square meters and a payload weight capacity of approximately 875 metric tons for vehicles
- Requirement - 12 - R12 The vessel shall be equipped with communications capabilities, including satellite communication, radio communication and data transmission systems
- Requirement - 10 - R10 The vessel shall be equipped with an electric plant capable of meeting the vessel's electric demand under all operational conditions
- Requirement - 1.1 - R1.1 The vessel shall accommodate a total of 300 armed troops and 100 crew-personnel
- Requirement - 5.2.1 - R5.2.1 The vessel shall provide a flight deck measuring at least 140m x W20m, capable of simultaneous launch and recovery of two helicopters
- Requirement - 5.2.2 - R5.2.2 The vessel shall provide hangar space for 1 helicopter
- Requirement - 5.2.3 - R5.2.3 The vessel shall feature a Helicopter Coordination Section to facilitate safe and efficient coordination of helicopter activities
- Requirement - 2 - R2 The vessel must have designated spaces for efficient embarkation, storage, and transportation of supplies, equipment, and amphibious vehicles
- Requirement - 9.1 - R9.1 The vessel shall regulate internal temperature for climate control
- Requirement - 9.2 - R9.2 The vessel shall ensure continuous airflow for fresh air
- Requirement - 9.3 - R9.3 The vessel shall ensure safe and drinkable water supply
- Requirement - 1 - R1 The vessel shall support the deployment of troops and crew personnel
- Requirement - 7 - R7 The vessel shall provide emergency rescue capabilities utilizing aerial methods
- Requirement - 5.2 - R5.2 The vessel shall support ship-to-shore movement via aerial means
- Requirement - 5 - R5 The vessel shall offer amphibious raid flexibility, enabling ship-to-shore movement via landing craft and/or aircraft
- Requirement - 9 - R9 The vessel shall maintain habitable conditions for embarked personnel
Derive
Satisfy

Figure 6.17: [PAB] Physical Architecture Blank Diagram illustrating the complete physical view of the LPD model in Capella

into categories such as completeness, integrity, and design. It was decided that detected errors (significant mistakes) must be resolved to progress to the next architectural level. Although warnings (less significant mistakes) suggest areas for improvement, addressing them is not prioritized in this project due to time constraints.

6.3.8. Modifications to the Capella model

This section addresses the model modification, representing Step 8: Modify MBSE Models, as specified in the methodology of Chapter 4. Once the model has been established, it is time to evaluate the versatility of the MBSE environment in capturing and processing evolving requirements that lead to design modifications. It is of great importance to highlight that a more detailed description of the model modification process in Capella is provided in Chapter 7 as this information is pivotal in assessing the tool's flexibility.

Consumer Set Modifications

According to the proposed process, the first group of proposed modifications corresponds to the Consumer Set Modifications. To illustrate this, we investigated both the addition of new equipment and the removal of existing equipment.

The example of adding a piece of equipment, specifically an anchor, was examined. The process of adding a new anchor to the Capella model involved several steps across different architectural layers. Beginning with the System Analysis layer, a new system function was introduced and transitioned through logical and physical layers. Throughout this process, the tool automatically updated associated diagrams and provided helpful shortcuts, like automatic function allocation, streamlining the modification process.

Additionally, the removal of existing equipment was investigated. To illustrate this, the previously added anchor was simply removed from the Physical Architecture.

Model Modifications
In Chapter 4, the addition of a new parameter was chosen to demonstrate how the tool handles model modifications. However, it is important to note that the Capella model has not been populated with parameters, for reasons that are explained later. Therefore, this modification cannot be performed in Capella.

It is vital to underscore that following each modification, the previous step (Step 7) concerning model verification and validation must be repeated. This ensures that the updated engineering solution adheres to the modeling language, satisfies the project requirements, and remains consistent with the designated design strategy.

6.3.9. Collecting, Sharing and Analyzing Modeling Constructs in Capella

This section includes the collection and export of modeling constructs, representing Step 9: Collect and Analyze Modeling Constructs, as specified in the methodology of Chapter 4.

Capella's main modeling constructs can be visualized in graphs, and exporting these graphs as image files is sufficient to achieve the objectives of this thesis. However, It is recognized that sharing functional models with all stakeholders is essential in MBSE. Thus for large projects with multiple stakeholders, more interactive exported files are required.

Team for Capella is an add-on for Capella that allows multiple team members to collaborate on editing Capella models at the same time. It aims to streamline the collaboration process and avoid conflicts that can occur from simultaneous updates. This collaborative platform provides advanced features for team-based model management, version control, and collaborative editing (Obeo n.d.). However, it requires a commercial license and was therefore deemed not worth exploring further for this project.

Instead, the XHTML add-on was investigated. This add-on enables the end-user to generate an HTML website from a Capella project. This free-of-charge solution for publishing and sharing HTML versions of models helps make models the reference for all engineering activities, supporting easier access to the model. An HTML website has been successfully created for the LPD model through this add-on.

6.3.10. Insights Gained During Capella Modeling

Some general insights and comments regarding the aforementioned modeling effort in Capella are presented below:

Every modeling element contained in the metamodel has been successfully modeled. However, the population of the system, subsystems, and equipment with parameters has been omitted. Parameterization in Capella proved to be very challenging since it had to be done manually. Nonetheless, this was not the primary reason for the omission. Even if time had been dedicated to populating the model with parameters, the Capella tool itself would not be able to perform any calculations or comparisons based on these parameters. In other words, the parameters would exist in the tool solely for completeness and would not have any other useful properties. Moreover, they are not easily visualized in the tool. One notable add-on that could be very useful in the early stages of naval vessel design is the “Basic Mass Viewpoint”. This allows for the visualization of the weight of each modeled element, as illustrated in Figure 6.18. Similar to the requirements viewpoint, each modeling element is linked to a modeling block, where both the actual and the maximum allowable mass of the part must be specified. If the actual mass is less than the maximum value, the modeling element retains its original color. However, if the actual mass meets or exceeds the maximum value, the modeling element changes to orange and red, respectively. This visual feedback enables ship designers to quickly identify and address potential weight challenges in more complex projects.

In general, interconnection with other tools, whether for calculations or other purposes, is necessary for making good use of the parameters in Capella. However, since the performance of the tool depends on add-ons and external tools, their assessment was considered outside the scope of this thesis.

As previously stated, the term “system architecture” in this thesis encompasses a conceptual model that outlines the structure of a system across four distinct domains. The final products of each design

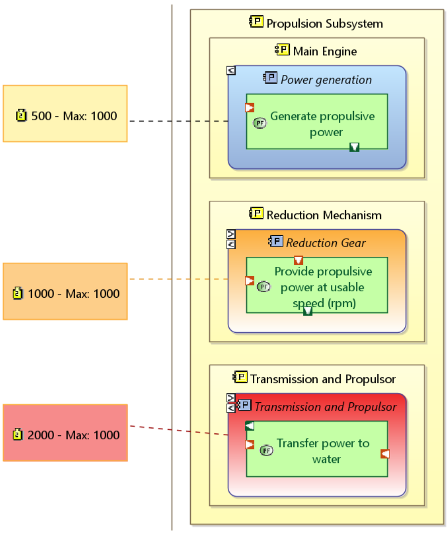


Figure 6.18: Visualization of mass with the help of the Basic Mass Viewpoint for the LPD model in Capella

phase, namely the OAB (Figure 6.6), SAB (Figure 6.10), LAB (Figure 6.13), and PAB (Figure 6.17), are fundamental parts of this conceptual model. Despite their structural similarity, each has a different purpose and level of detail. Additionally, the secondary diagrams mentioned during the modeling effort also constitute significant modeling constructs, crucial for the completeness of this conceptual model. What enhances the utility of all these diagrams is their interconnectedness, which ensures consistency across various perspectives.

Although this was a very simplified version of an LPD project, the diagrams are still quite large. This is expected as we progress to the solution layers, given the information the diagrams present: actors, components, allocated functions, and their interactions. Consequently, large diagrams should be anticipated even for simple systems.

Conversely, the consistent modeling logic throughout the process enhances modeling speed. It may be observed that the principal difference among the various architectural layers does not lie in the types of diagrams created or the way that they are developed; rather, it lies in the approach to the design problem. Even for inexperienced modelers, like the author of this thesis, the repeated use of the same concepts from layer to layer allows for quicker familiarity with both the tool and the methodology.

Finally, it is noted that adapting the transitioned functional chains to integrate the newly created Physical Functions posed challenges in usability with the tool. This implies that the tool was not as user-friendly as desired for this task. As a result, additional functional chains had to be developed at the physical layer. While this process was straightforward, it raised a few concerns regarding artifact reusability.

6.4. Modeling in COMET

6.4.1. Introduction to CDP4-COMET

CDP4-COMET supports MBSE using the CD method. It is pivotal to highlight that CDP4-COMET is a collaborative MBSE tool in which all key stakeholders have access to the model as well as ownership of the information depending on their domain of expertise. In this model, a decomposition of the system is established in which elements of the system and associated parameters represent factors that enable analysis of the key challenges in an ESSD problem.

In this graduation project, the CDP4-COMET IME (Integrated Modelling Environment - Desktop Application) version 9.5.0.0 was used. To enhance its capabilities, external calculation tools such as Excel, equipped with the CDP4-COMET Excel integration containing the CP4-COMET ribbon Viewpoint. This integration allows one to interact with COMET-CDP4 directly through Excel, enabling the import of data into Excel and export it back to the COMET-CDP4 server. This capability is particularly useful for performing complex analyses, such as creating mass budgets and power budgets. In this project, it was also used during the requirements verification setup.

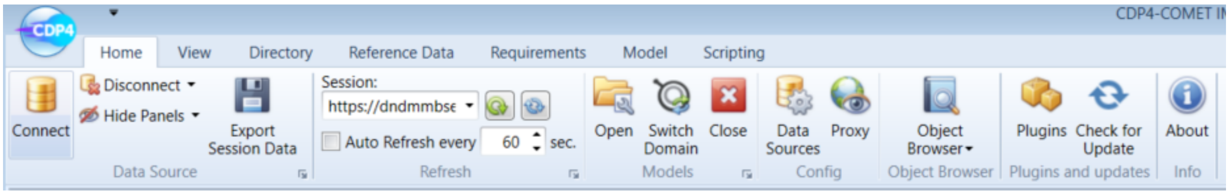
To gain proficiency with the tool and grasp its potential, tutorials such as the CDP4 video series on YouTube (RHEA Group Services 2020) were thoroughly explored. The author's familiarity with this tool was greatly enriched by his participation in the "Dutch Naval Design: Mission & MBSE" (DND-MMBSE) pilot project (Kolfschoten et al. 2024). This initiative centered around a case study on the chilled water system of a Dutch naval frigate.

6.4.2. Getting Started

First, it is deemed important to briefly outline the modeling environments of the tool. The application features a ribbon interface at the top, comprising several buttons organized into tabs. Each tab provides access to specific functionalities, which are detailed below.

Home Tab

In the Home tab, depicted in Figure 6.19, one important button to use is the Connect button, which facilitates the connection to a live data source or a CDP4 server. Another commonly used button is the Open button, through which multiple engineering models from various data sources can be opened.



Session: https://dndmmbse
☐ Auto Refresh every 60 sec.

Figure 6.19: Modeling environment in CDP4-COMET: Home Tab

View Tab

In the View tab, it is possible to choose to show or hide deprecated items, display the property grid for more detailed information on selected items, and view the log panel to observe all the logging activities performed by CDP4 during interactions. The View tab is illustrated in Figure 6.20.

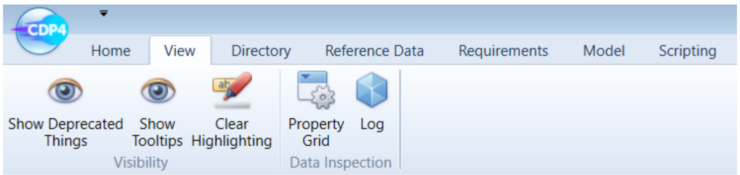


Figure 6.20: Modeling environment in CDP4-COMET: View Tab

Directory Tab

The Directory tab, shown in Figure 6.21, enables the display of a list of all available engineering models on the connected server that the user is allowed to see according to their permissions. It also gives an overview of the domains of expertise and the person roles.

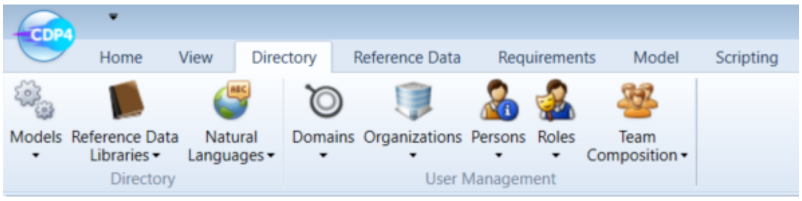


Figure 6.21: Modeling environment in CDP4-COMET: Directory Tab

Reference Data Tab

The Reference Data tab, depicted in Figure 6.22, allows for the management of reference data, which is reused across different models or projects. This includes parameter types, scales, units, and categories that add more semantics to a model.

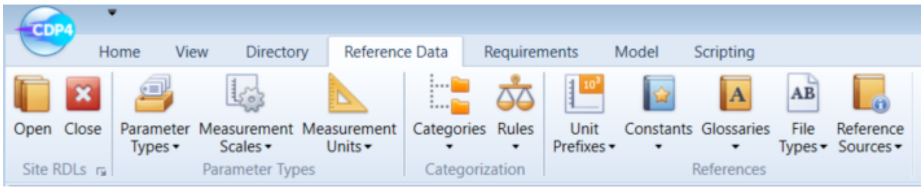


Figure 6.22: Modeling environment in CDP4-COMET: Reference Data Tab

Requirements Tab

The Requirements tab allows for the management of requirements, including the creation of requirement specifications and the linking of these requirements to the design for automated verification. Additionally, it supports the import and export of requirements. This tab is displayed in Figure 6.23.

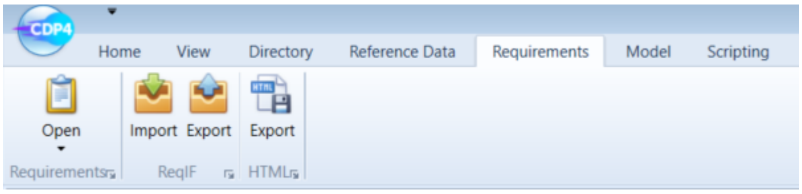


Figure 6.23: Modeling environment in CDP4-COMET: Requirements Tab

Model Tab

The Model tab, depicted in Figure 6.24 provides a comprehensive view of all building blocks of the architecture, presented in the form of product trees, along with the option browser. Other important functionalities of this tab include an overview of the relationships and relationship matrices. Moreover, this tab contains features for reporting and rules verification, both of which are used in the current thesis.

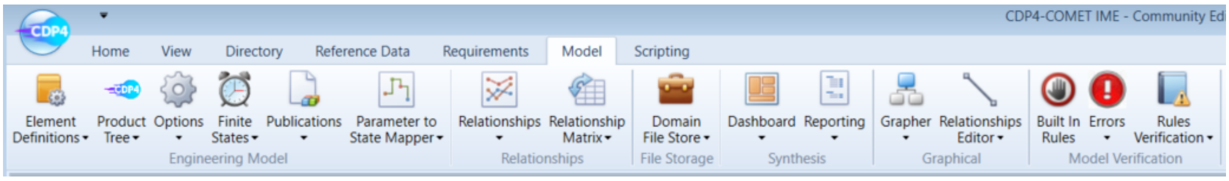


Figure 6.24: Modeling environment in CDP4-COMET: Model Tab

Scripting Tab

The Scripting tab, shown in Figure 6.25, provides functionality for interacting with the scripting engine, currently supporting Python scripting.

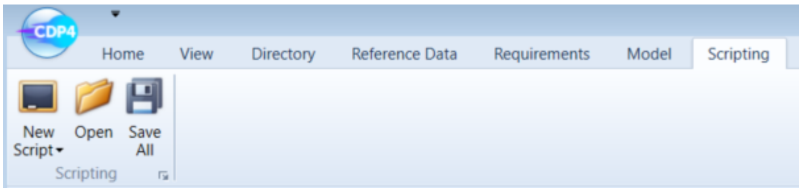


Figure 6.25: Modeling environment in CDP4-COMET: Scripting Tab

To create an engineering model, some initial decisions have to be made. Even though a new model can be based on an existing one to avoid recreating it from scratch, it was decided to create a completely new one to explore the potential of the tool. CDP4-COMET supports four types of engineering models: Study Model, Template Model, Model Catalog, and Scratch Model. The Study Model, which is used for collaborative work within a team, was selected for the model in this thesis. Additionally, models have different phases that reflect the stages of a concurrent design study, such as the Preparation Phase, Design Session Phase, Reporting Phase, and Completed Study. For this thesis, the Preparation Phase, in which the setup of the model with the core team occurs, was selected.

There is also the concept of reference data libraries. These libraries serve as a reservoir of data that can be reused across various engineering models. They include definitions of parameter types along with categories useful for labeling different building blocks within a model. This facilitates easy retrieval and allows for the addition of further semantics. Additionally, reference data libraries may contain rules pertinent to model construction and management. Throughout the project, the generic RDL was employed, with any supplementary parameters or rules being manually added to a personal library as the project progressed.

Moreover, in CDP4-COMET the system under consideration is broken down into domains of expertise. Each team member represents a domain of expertise, that is responsible for a particular aspect of the design. Thus, it becomes evident that many concepts in CDP4-COMET can be owned by a Domain of Expertise. Ownership refers to the situation where all domain experts involved manage their data in the system specification and can take responsibility for the data quality. The assumption is that ownership will ensure quality by minimizing errors that may arise from misinterpretation by individuals.

The Ownership feature can be exemplified through the Role-Based Access Control within CDP4-COMET. Some example domains of expertise that have been included in the LPD model are Armament, Marine Engineering, Marine Navigation, Naval Architecture, Propulsion, Structures, and System Engineering. It is important to note that despite incorporating multiple disciplines (domains of expertise) into the design, the author was solely responsible for the modeling procedures. As a result, the author assumed all domains of expertise, accessing the design from different perspectives by logging in with different discipline roles each time. While this approach is a simplified demonstration, in reality, it may not be feasible for a single individual to handle multiple roles in a multi-domain context. Nevertheless, it effectively showcases the capability of CDP4-COMET in distributing responsibilities and granting appropriate access to project stakeholders

Finally, a short introduction to the elements that constitute the structure will aid in the subsequent analysis. An element definition in CDP4-COMET is a building block or component that represents a specific entity in a model. For the LPD example, this can be a system, a subsystem, an equipment, a function, etc. It is defined by parameters and owned by a particular domain of expertise, indicating who is responsible for its design and management. On the other hand, an element usage denotes the inclusion of an element definition within another element definition, specifying the components or constituents of a higher-level entity. This hierarchical structure allows for the creation of complex architectures by assembling simpler building blocks into larger systems. The element definition blocks are listed in the Element Definition Browser, while the Element Usages and the architecture of the decomposition of the model are shown in the Product Tree.

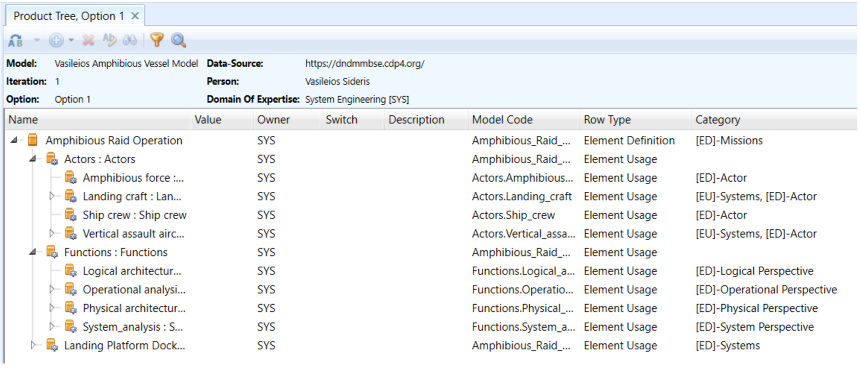
6.4.3. Operational Analysis

To effectively recreate the Operational Analysis in CDP4-COMET, it is essential to consider the metamodel depicted in Figure 6.1. The metamodel proposes to model the mission, system actors, and operational activities of the stakeholder in the operational phase.

To achieve this in CDP4-COMET, the model has been crafted with a modeling philosophy exemplified in the product tree showcased in Figure 6.26. As depicted, the Amphibious Raid Operation block, symbolizing the mission, serves as the top-level element in the model hierarchy. Notably, the inclusion of categories within this tool is fundamental, as it enables the specification of both component types and requirements. Returning to the product tree of Figure 6.26, to distinctly denote that the Amphibious Raid Operation element pertains to the mission, it has been categorized as “mission”.

Nested within the mission element are three additional elements: Actors, functions, and the Landing Platform Dock Vessel. As their names imply, they represent the actors of the system, the functions, and the system itself, respectively. Within the actor’s group, further elements, namely the Amphibious Force, the Landing Craft, the Ship Crew, and the Vertical Assault Aircraft are contained, all categorized as "actors". The "Functions" block comprises four elements, each representing lower-level function groups within one of the four architectural layers: Operational Analysis, System Analysis, Logical Architecture, and Physical Architecture. The Landing Platform Dock Vessel, categorized as a "system", serves as the SoI and encapsulates all constituent building blocks comprising the architecture. More elements will be incorporated into this structure post-Operational Analysis, once the system is fully defined.

It is reminded that the sole participant in the model is the author of this document, who assumes the roles of various domain experts to develop the system elements. Thus far, all the elements described fall under the domain of expertise of the System Engineer denoted as "SYS" in the Owner column.



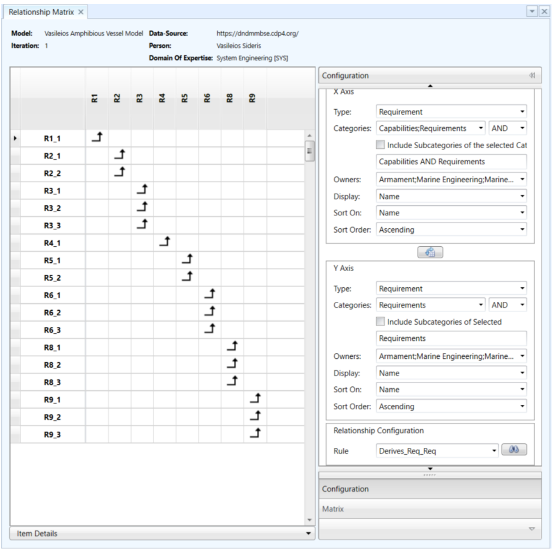
Model: Vasileios Amphibious Vessel Model **Data-Source:** https://dndmmbse.cdp4.org/
Iteration: 1 **Person:** Vasileios Sideris
Option: Option 1 **Domain Of Expertise:** System Engineering [SYS]

Name	Value	Owner	Switch	Description	Model Code	Row Type	Category
Amphibious Raid Operation		SYS			Amphibious_Raid_...	Element Definition	[ED]-Missions
Actors : Actors		SYS			Amphibious_Raid_...	Element Usage	
Amphibious force :...		SYS			Actors.Amphibious...	Element Usage	[ED]-Actor
Landing craft : Lan...		SYS			Actors.Landing_craft	Element Usage	[EU]-Systems, [ED]-Actor
Ship crew : Ship crew		SYS			Actors.Ship_crew	Element Usage	[ED]-Actor
Vertical assault airc...		SYS			Actors.Vertical_assa...	Element Usage	[EU]-Systems, [ED]-Actor
Functions : Functions		SYS			Amphibious_Raid_...	Element Usage	
Logical architectur...		SYS			Functions.Logical_a...	Element Usage	[ED]-Logical Perspective
Operational analysi...		SYS			Functions.Operatio...	Element Usage	[ED]-Operational Perspective
Physical architectur...		SYS			Functions.Physical_...	Element Usage	[ED]-Physical Perspective
System_analysis : S...		SYS			Functions.System_a...	Element Usage	[ED]-System Perspective
Landing Platform Dock...		SYS			Amphibious_Raid_...	Element Usage	[ED]-Systems

Figure 6.26: Product tree in CDP4-COMET outlining the structure of the proposed modeling solution

By establishing this template structure in CDP4-COMET, we have made a good start in the operational phase. Before proceeding, it is important to note that CDP4-COMET cannot visually construct models using diagrams. However, for the sake of consistent comparison with other tools, efforts were made to maintain coherence in the analysis. CDP4-COMET features diagramming capabilities that allow for linking elements with various types of relations. Consequently, the visualization of modeling constructs in CDP4-COMET was explored through the functionality of relationship matrix creation. It is pointed out that a small green indicator appears on each modeling element as soon as the latter is linked with at least one binary relationship. In the upcoming sections, we analyze the creation of several matrices aimed at providing a comparable visual representation to the graphs generated in Capella. Given that the modeling philosophy aligns with the approach detailed in Section **??**, the rationale behind each architectural level will be maintained at a high level to prevent redundancy.

A "derive" binary relationship was utilized to link lower-level requirements to the higher-level requirements. These relationships can be visualized in a relationship matrix. Figure 6.27 illustrates the aforementioned "derive" relationship among the modeled requirements.



Model: Vasileios Amphibious Vessel Model **Data-Source:** https://dndmmbse.cdp4.org/
Iteration: 1 **Person:** Vasileios Sideris
Domain Of Expertise: System Engineering [SYS]

	R1	R2	R3	R4	R5	R6	R8	R9
R1_1	↱							
R2_1		↱						
R2_2		↱						
R3_1			↱					
R3_2			↱					
R3_3			↱					
R4_1				↱				
R5_1					↱			
R5_2					↱			
R6_1						↱		
R6_2						↱		
R6_3						↱		
R8_1							↱	
R8_2							↱	
R8_3							↱	
R9_1								↱
R9_2								↱
R9_3								↱

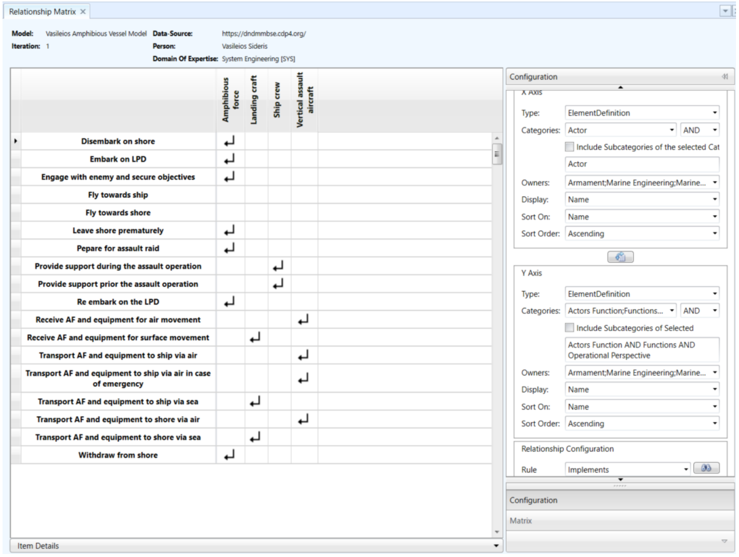
X Axis
Type: Requirement
Categories: Capabilities;Requirements AND
□ Include Subcategories of the selected Cat
Capabilities AND Requirements
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

Y Axis
Type: Requirement
Categories: Requirements AND
□ Include Subcategories of Selected
Requirements
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

Relationship Configuration
Rule Derives_Req_Req

Figure 6.27: Relationship matrix of requirements for the LPD model in CDP4-COMET

Based on the requirements and the mission description, a set of operational activities has been created in the form of several Element Definition blocks. An “implement” relationship was utilized to map the identified operational activities to the four actors. The mapping can be visualized with the help of the relationship matrix in Figure 6.28.



Model: Vasileios Amphibious Vessel Model **Data-Source:** https://dndmmbse.cdp4.org/
Iteration: 1 **Person:** Vasileios Sideris
Domain Of Expertise: System Engineering [SYS]

	Amphibious force	Landing craft	Ship crew	Vertical assault aircraft
Disembark on shore	↵			
Embark on LPD	↵			
Engage with enemy and secure objectives	↵			
Fly towards ship				
Fly towards shore				
Leave shore prematurely	↵			
Pepare for assault raid	↵			
Provide support during the assault operation			↵	
Provide support prior the assault operation			↵	
Re embark on the LPD	↵			
Receive AF and equipment for air movement				↵
Receive AF and equipment for surface movement		↵		
Transport AF and equipment to ship via air				↵
Transport AF and equipment to ship via air in case of emergency				↵
Transport AF and equipment to ship via sea		↵		
Transport AF and equipment to shore via air				↵
Transport AF and equipment to shore via sea		↵		
Withdraw from shore	↵			

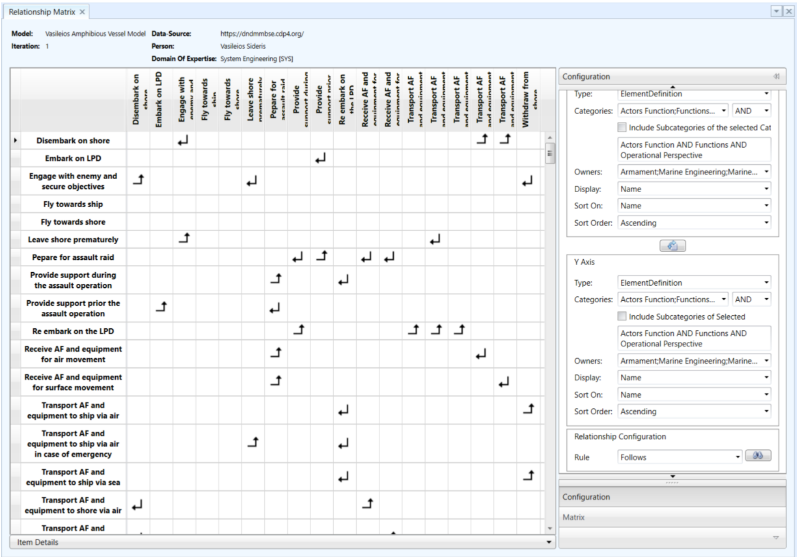
X Axis
Type: ElementDefinition
Categories: Actor AND
□ Include Subcategories of the selected Cat
Actor
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

Y Axis
Type: ElementDefinition
Categories: Actors Function;Functions... AND
□ Include Subcategories of Selected
Actors Function AND Functions AND Operational Perspective
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

Relationship Configuration
Rule Implements

Figure 6.28: Relationship matrix between actors and operational activities for the LPD model in CDP4-COMET

To complete the operational phase, the data flow among the operational activities needs to be modeled. Therefore, another binary relationship, named “follow,” is used to represent the sequence and flow of operational activities. These relationships are visualized using the relationship matrix shown in Figure 6.29.



Relationship Matrix ×

Model: Vasileios Amphibious Vessel Model **Data-Source:** https://dndmmbse.cdp4.org/
Iteration: 1 **Person:** Vasileios Sideris
Domain Of Expertise: System Engineering [SYS]

	Disembark on shore	Embark on LPD	Engage with enemy and	Fly towards ship	Fly towards shore	Leave shore prematurely	Pepare for assault raid	Provide support during	Provide support prior	Re embark on the LPD	Receive AF and equipment for	Receive AF and equipment for	Transport AF and equipment	Transport AF and equipment	Transport AF and equipment	Transport AF and equipment	Transport AF and equipment	Withdraw from shore
Disembark on shore			↵													↑	↑	
Embark on LPD									↵									
Engage with enemy and secure objectives	↑					↵												↵
Fly towards ship																		
Fly towards shore																		
Leave shore prematurely			↑											↵				
Pepare for assault raid								↵	↑		↵	↵						
Provide support during the assault operation							↑			↵								
Provide support prior the assault operation		↑					↵											
Re embark on the LPD								↑					↑	↑	↑			
Receive AF and equipment for air movement							↑									↵		
Receive AF and equipment for surface movement							↑										↵	
Transport AF and equipment to ship via air										↵								↑
Transport AF and equipment to ship via air in case of emergency						↑				↵								
Transport AF and equipment to ship via sea										↵								↑
Transport AF and equipment to shore via air	↵										↑							
Transport AF and																		

Item Details

Configuration
Type: ElementDefinition
Categories: Actors Function;Functions... AND
□ Include Subcategories of the selected Cat
Actors Function AND Functions AND Operational Perspective
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

Y Axis
Type: ElementDefinition
Categories: Actors Function;Functions... AND
□ Include Subcategories of Selected
Actors Function AND Functions AND Operational Perspective
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

Relationship Configuration
Rule Follows

Configuration
Matrix

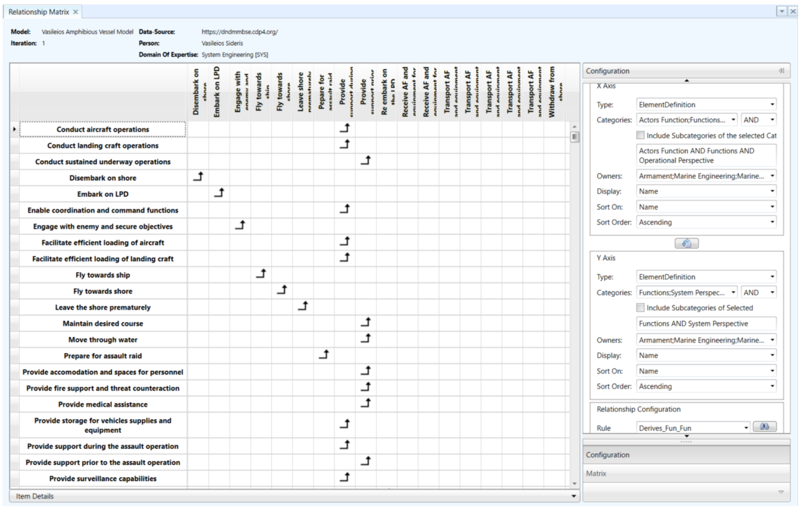
Figure 6.29: Relationship matrix depicting the flow of operational Activities for the LPD model in CDP4-COMET

The modeling structure described above, along with the visualization of the relationships among the modeling elements at the operational level, as depicted in the relationship matrices in Figures 6.27, 6.28, and 6.29, signifies the completion of the Operational Analysis phase.

6.4.4. System Analysis

Similar to the modeling procedure in Capella, proceeding to the next design phase in CDP4-COMET involves transitioning elements from the previous phase. Unlike the automatic or manual transition feature witnessed in the Arcadia methodology, CDP4-COMET does not offer such functionality. Instead, transitioning in this tool requires the manual creation of new Element Definitions and categorizing them to represent the desired modeling elements.

For the transition to the system level, the operational activities have been transformed into system functions. A “derive” relationship was used to link the elements of the two architectural layers. This transition is displayed in the relationship matrix of Figure 6.30.



Relationship Matrix ×

Model: Vasileios Amphibious Vessel Model **Data-Source:** https://dndmmbse.cdp4.org/
Iteration: 1 **Person:** Vasileios Sideris
Domain Of Expertise: System Engineering [SYS]

	Disembark on shore	Embark on LPD	Engage with enemy and	Fly towards ship	Fly towards shore	Leave shore prematurely	Pepare for assault raid	Provide support during	Provide support prior	Re embark on the LPD	Receive AF and equipment for	Receive AF and equipment for	Transport AF and equipment	Transport AF and equipment	Transport AF and equipment	Transport AF and equipment	Transport AF and equipment	Withdraw from shore
Conduct aircraft operations								↑										
Conduct landing craft operations								↑										
Conduct sustained underway operations									↑									
Disembark on shore	↑																	
Embark on LPD		↑																
Enable coordination and command functions								↑										
Engage with enemy and secure objectives			↑															
Facilitate efficient loading of aircraft								↑										
Facilitate efficient loading of landing craft								↑										
Fly towards ship				↑														
Fly towards shore					↑													
Leave the shore prematurely						↑												
Maintain desired course									↑									
Move through water									↑									
Prepare for assault raid							↑											
Provide accomodation and spaces for personnel									↑									
Provide fire support and threat counteraction									↑									
Provide medical assistance									↑									
Provide storage for vehicles supplies and equipment								↑										
Provide support during the assault operation								↑										
Provide support prior to the assault operation									↑									
Provide surveillance capabilities								↑										

Item Details

Configuration
X Axis
Type: ElementDefinition
Categories: Actors Function;Functions... AND
□ Include Subcategories of the selected Cat
Actors Function AND Functions AND Operational Perspective
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

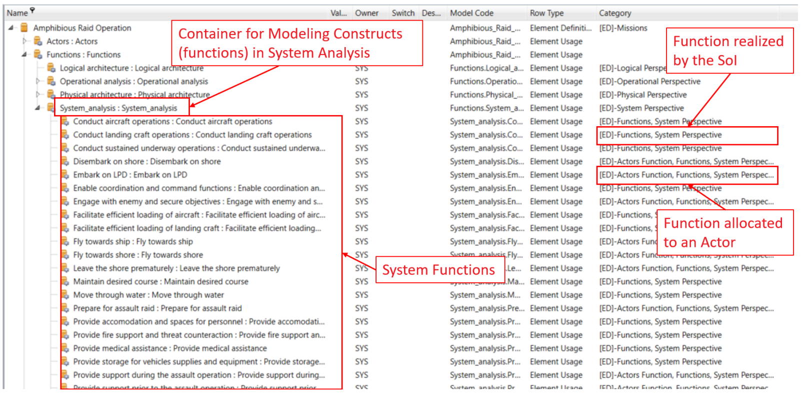
Y Axis
Type: ElementDefinition
Categories: Functions;System Perspec... AND
□ Include Subcategories of Selected
Functions AND System Perspective
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

Relationship Configuration
Rule Derives_Fun_Fun

Configuration
Matrix

Figure 6.30: Relationship matrix between operational activities and system functions for the LPD model in CDP4-COMET

Each new function's Element Definition has been included as an Element Usage within the Element Definition called "System Analysis", which serves as the container for all modeling constructs at this layer. This is evident in Figure 6.31.



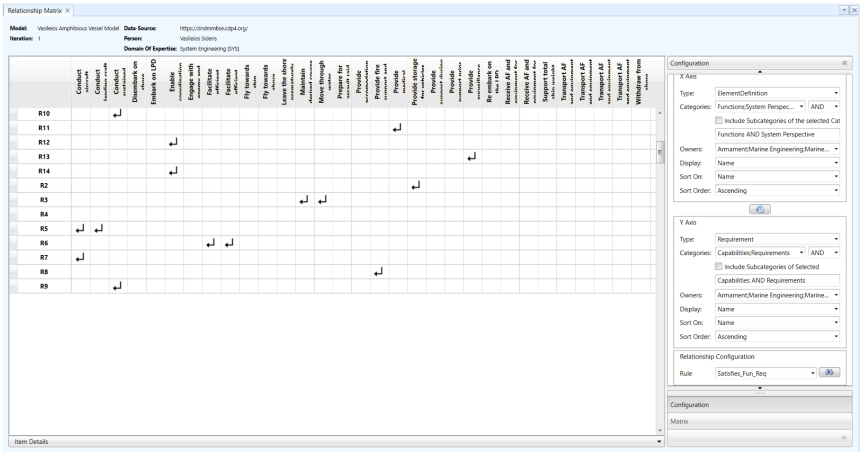
Name	Val...	Owner	Switch	Des...	Model Code	Row Type	Category
Amphibious Raid Operation					Amphibious_Raid_...	Element Definiti...	[ED]-Missions
Actors : Actors					Amphibious_Raid_...	Element Usage	
Functions : Functions					Amphibious_Raid_...	Element Usage	
Logical architecture : Logical architecture		SYS			Functions.Logical_a...	Element Usage	[ED]-Logical Perspe
Operational analysis : Operational analysis		SYS			Functions.Operatio...	Element Usage	[ED]-Operational Perspective
Physical architecture : Physical architecture		SYS			Functions.Physical_...	Element Usage	[ED]-Physical Perspective
System_analysis : System_analysis		SYS			Functions.System_a...	Element Usage	[ED]-System Perspective
Conduct aircraft operations : Conduct aircraft operations		SYS			System_analysis.Co...	Element Usage	[ED]-Functions, System Perspective
Conduct landing craft operations : Conduct landing craft operations		SYS			System_analysis.Co...	Element Usage	[ED]-Functions, System Perspective
Conduct sustained underway operations : Conduct sustained underwa...		SYS			System_analysis.Co...	Element Usage	[ED]-Functions, System Perspective
Disembark on shore : Disembark on shore		SYS			System_analysis.Dis...	Element Usage	[ED]-Actors Function, Functions, System Perspec...
Embark on LPD : Embark on LPD		SYS			System_analysis.Em...	Element Usage	[ED]-Actors Function, Functions, System Perspec...
Enable coordination and command functions : Enable coordination an...		SYS			System_analysis.En...	Element Usage	[ED]-Functions, System Perspective
Engage with enemy and secure objectives : Engage with enemy and s...		SYS			System_analysis.En...	Element Usage	[ED]-Actors Function, Functions, System Perspec...
Facilitate efficient loading of aircraft : Facilitate efficient loading of airc...		SYS			System_analysis.Fac...	Element Usage	[ED]-Functions,
Facilitate efficient loading of landing craft : Facilitate efficient loading...		SYS			System_analysis.Fac...	Element Usage	[ED]-Functions,
Fly towards ship : Fly towards ship		SYS			System_analysis.Fly...	Element Usage	[ED]-Actors Func
Fly towards shore : Fly towards shore		SYS			System_analysis.Fly...	Element Usage	[ED]-Actors Fun
Leave the shore prematurely : Leave the shore prematurely		SYS			.Le...	Element Usage	[ED]-Actors Function, Functions, System Perspec...
Maintain desired course : Maintain desired course		SYS			.Ma...	Element Usage	[ED]-Functions, System Perspective
Move through water : Move through water		SYS			System_analysis.M...	Element Usage	[ED]-Functions, System Perspective
Prepare for assault raid : Prepare for assault raid		SYS			System_analysis.Pre...	Element Usage	[ED]-Actors Function, Functions, System Perspec...
Provide accomodation and spaces for personnel : Provide accomodati...		SYS			System_analysis.Pr...	Element Usage	[ED]-Functions, System Perspective
Provide fire support and threat counteraction : Provide fire support an...		SYS			System_analysis.Pr...	Element Usage	[ED]-Functions, System Perspective
Provide medical assistance : Provide medical assistance		SYS			System_analysis.Pr...	Element Usage	[ED]-Functions, System Perspective
Provide storage for vehicles supplies and equipment : Provide storage...		SYS			System_analysis.Pr...	Element Usage	[ED]-Functions, System Perspective
Provide support during the assault operation : Provide support during...		SYS			System_analysis.Pr...	Element Usage	[ED]-Actors Function, Functions, System Perspec...
Provide support prior to the assault operation : Provide support prior...		SYS			System_analysis.Pr...	Element Usage	[ED]-Actors Function, Functions, System Perspec...

Container for Modeling Constructs (functions) in System Analysis
Function realized by the SoI
Function allocated to an Actor
System Functions

Figure 6.31: Selected system functions for the LPD Model in CDP4-COMET

At this stage, the system can also be defined. This is modeled in an Element Definition block called "Landing Platform Dock Vessel." A set of example parameters has been assigned to it, including mass, range, speed, total accommodation capacity, center of gravity in three axes, dimensions, and power consumption.

According to the metamodel, it is also important to keep traces of the modeling elements to the requirements. In this case, the system functions are mapped to the Requirements with a "satisfy" binary relationship. The result is visually presented in Figure 6.32.



Model: Vasileios Amphibious Vessel Model **Data-Source:** https://dndmmbse.cdp4.org/
Iteration: 1 **Person:** Vasileios Sideris
Domain Of Expertise: System Engineering [SYS]

	Conduct aircraft	Conduct landing craft	Conduct sustained	Disembark on shore	Embark on LPD	Enable coordination	Engage with enemy and	Facilitate efficient	Facilitate efficient	Fly towards ship	Fly towards shore	Leave the shore prematurely	Maintain desired course	Move through water	Prepare for assault raid	Provide accomodation	Provide fire support and	Provide medical	Provide storage for vehicles	Provide support during	Provide support prior	Provide surveillance	Re embark on the LPD	Receive AF and equipment for	Receive AF and equipment for	Support total ship weight	Transport AF and equipment	Transport AF and equipment	Transport AF and equipment	Transport AF and equipment	Withdraw from shore
R10			↵																												
R11																		↵													
R12						↵																									
R13																						↵									
R14						↵																									
R2																			↵												
R3													↵	↵																	
R4																															
R5	↵	↵																													
R6								↵	↵																						
R7	↵																														
R8																	↵														
R9			↵																												

Configuration

X Axis
Type: ElementDefinition
Categories: Functions;System Perspec... AND
☐ Include Subcategories of the selected Cat
Functions AND System Perspective
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

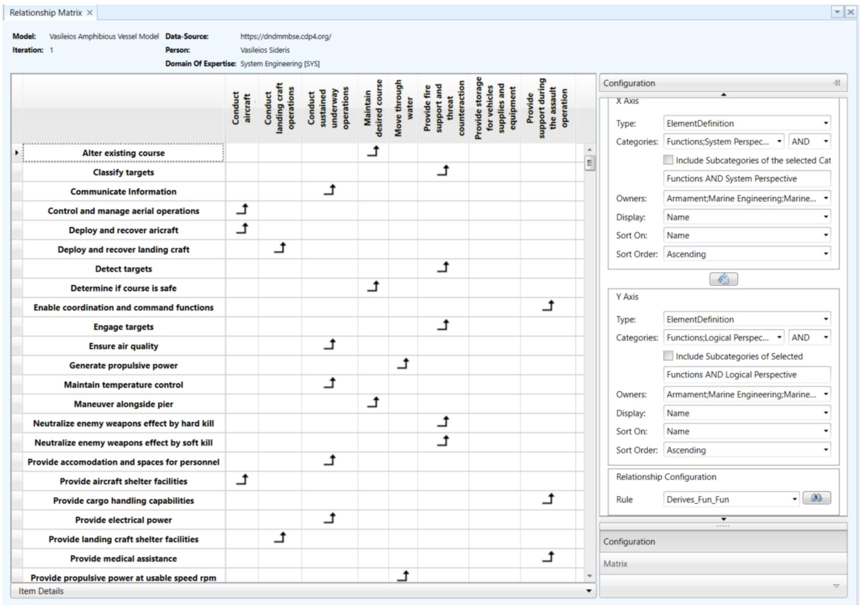
Y Axis
Type: Requirement
Categories: Capabilities;Requirements AND
☐ Include Subcategories of Selected
Capabilities AND Requirements
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

Relationship Configuration
Rule Satisfies_Fun_Req

Figure 6.32: Relationship matrix between system functions and requirements for the LPD model in CDP4-COMET

6.4.5. Logical Architecture

A similar procedure as before is followed in the Logical Architecture. New logical functions are modeled either by transitioning (copying) the system functions into new Element Definition blocks or by creating new logical functions to refine them further. The mapping of the newly created functions to the functions of the previous layer is depicted in Figure 6.33.



Relationship Matrix ×

Model: Vasileios Amphibious Vessel Model **Data-Source:** https://dndmmbse.cdp4.org/
Iteration: 1 **Person:** Vasileios Sideris
Domain Of Expertise: System Engineering [SYS]

	Conduct aircraft	Conduct landing craft operations	Conduct sustained underway operations	Maintain desired course	Move through water	Provide fire support and threat counteraction	Provide storage for vehicles supplies and equipment	Provide support during the assault operation
Alter existing course				↑				
Classify targets						↑		
Communicate Information			↑					
Control and manage aerial operations	↑							
Deploy and recover aricraft	↑							
Deploy and recover landing craft		↑						
Detect targets						↑		
Determine if course is safe				↑				
Enable coordination and command functions								↑
Engage targets						↑		
Ensure air quality			↑					
Generate propulsive power					↑			
Maintain temperature control			↑					
Maneuver alongside pier				↑				
Neutralize enemy weapons effect by hard kill						↑		
Neutralize enemy weapons effect by soft kill						↑		
Provide accomodation and spaces for personnel			↑					
Provide aircraft shelter facilities	↑							
Provide cargo handling capabilities								↑
Provide electrical power			↑					
Provide landing craft shelter facilities		↑						
Provide medical assistance								↑
Provide propulsive power at usable speed rpm					↑			

Item Details

Configuration

X Axis
Type: ElementDefinition
Categories: Functions;System Perspec... AND
□ Include Subcategories of the selected Cat
Functions AND System Perspective
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

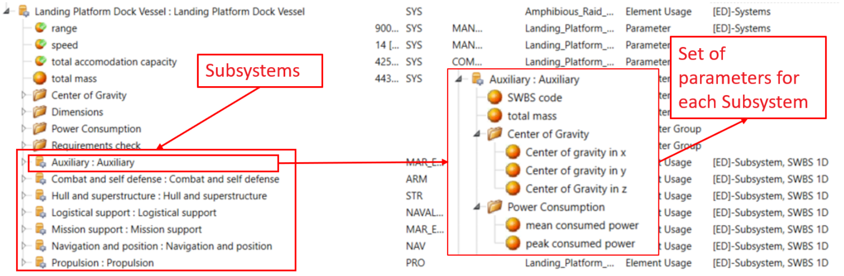
Y Axis
Type: ElementDefinition
Categories: Functions;Logical Perspec... AND
□ Include Subcategories of Selected
Functions AND Logical Perspective
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

Relationship Configuration
Rule Derives_Fun_Fun

Configuration
Matrix

Figure 6.33: Relationship matrix between system functions and logical functions for the LPD model in CDP4-COMET

At this stage, the system can be further decomposed. Several subsystems have been defined to perform the logical functions. Each one has been parameterized as shown in Figure 6.34.



```
Landing Platform Dock Vessel : Landing Platform Dock Vessel            SYS            Amphibious_Raid_...   Element Usage   [ED]-Systems
    range                                                       900...  SYS   MAN...   Landing_Platform_...  Parameter       [ED]-Systems
    speed                                                       14 [... SYS   MAN...   Landing_Platform_...  Parameter
    total accomodation capacity                                 425...  SYS   COM...   Landing_Platform_...  Parameter
    total mass                                                  443...  SYS                                  ter
    Center of Gravity                                                                                        ter
    Dimensions                                                                                               ter
    Power Consumption                                                                                        ter Group
    Requirements check                                                                                       ter Group
    Auxiliary : Auxiliary                                               MAR_E...                             Usage           [ED]-Subsystem, SWBS 1D
    Combat and self defense : Combat and self defense                   ARM                                  t Usage         [ED]-Subsystem, SWBS 1D
    Hull and superstructure : Hull and superstructure                   STR                                  t Usage         [ED]-Subsystem, SWBS 1D
    Logistical support : Logistical support                             NAVAL...                             t Usage         [ED]-Subsystem, SWBS 1D
    Mission support : Mission support                                   MAR_E...                             t Usage         [ED]-Subsystem, SWBS 1D
    Navigation and position : Navigation and position                   NAV                                  t Usage         [ED]-Subsystem, SWBS 1D
    Propulsion : Propulsion                                             PRO            Landing_Platform_...  Element Usage   [ED]-Subsystem, SWBS 1D
```

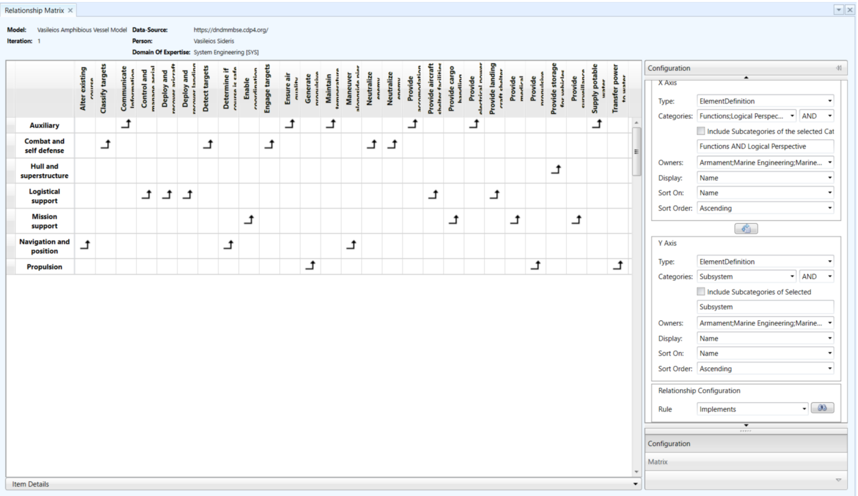
```
Auxiliary : Auxiliary
    SWBS code
    total mass
    Center of Gravity
        Center of gravity in x
        Center of gravity in y
        Center of Gravity in z
    Power Consumption
        mean consumed power
        peak consumed power
```

Figure 6.34: Identified subsystems for the LPD Model in CDP4-COMET

An “implement” binary relationship is used to map the logical functions to the identified subsystems. The result is visualized in Figure 6.35.

6.4.6. Physical Architecture

It is now time to transition to the main architectural layer in CDP4-COMET: the Physical Architecture. As with every transition, the logical functions are transferred to the physical layer. Some of these functions remain unchanged, while others are refined. After this process is finalized, the links with the functions from the previous layer are modeled. The results of this transition are depicted in Figure 6.36.



Relationship Matrix

Model: Vasileios Amphibious Vessel Model
Iteration: 1
Data-Source: https://dndmmbse.cdp4.org/
Person: Vasileios Sideris
Domain Of Expertise: System Engineering [SYS]

	Alter existing course	Classify targets	Communicate Information	Control and manage aerial	Deploy and recover aircraft	Deploy and recover landing	Detect targets	Determine if course is safe	Enable coordination	Engage targets	Ensure air quality	Generate propulsive	Maintain temperature	Maneuver alongside pier	Neutralize enemy	Neutralize enemy	Provide accomodation	Provide aircraft shelter facilities	Provide cargo handling	Provide electrical power	Provide landing craft shelter	Provide medical	Provide propulsive	Provide storage for vehicles	Provide surveillance	Supply potable water	Transfer power to water
Auxiliary			↑								↑		↑				↑			↑						↑	
Combat and self defense		↑					↑			↑					↑	↑											
Hull and superstructure																								↑			
Logistical support				↑	↑	↑												↑			↑						
Mission support									↑										↑			↑			↑		
Navigation and position	↑							↑						↑													
Propulsion												↑											↑				↑

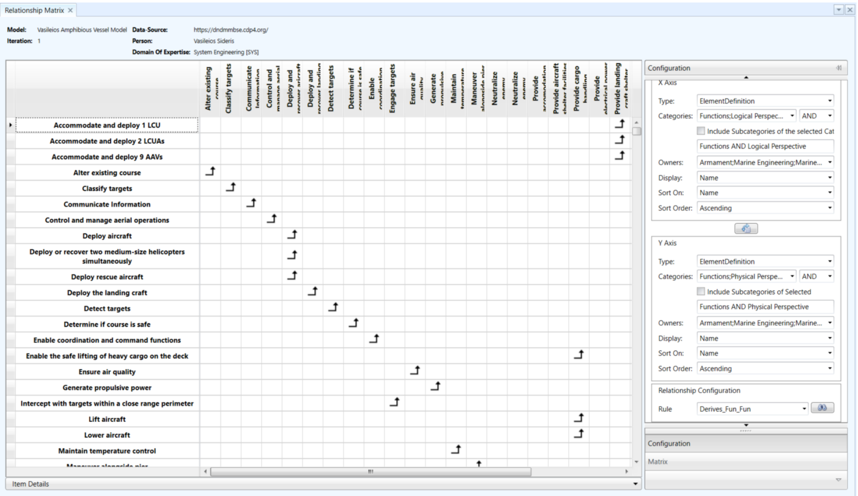
Configuration

X Axis
Type: ElementDefinition
Categories: Functions;Logical Perspec... AND
☐ Include Subcategories of the selected Cat
Functions AND Logical Perspective
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

Y Axis
Type: ElementDefinition
Categories: Subsystem AND
☐ Include Subcategories of Selected
Subsystem
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

Relationship Configuration
Rule: Implements

Figure 6.35: Relationship matrix between logical functions and subsystems for the LPD model in CDP4-COMET



Relationship Matrix

Model: Vasileios Amphibious Vessel Model
Iteration: 1
Data-Source: https://dndmmbse.cdp4.org/
Person: Vasileios Sideris
Domain Of Expertise: System Engineering [SYS]

	Alter existing course	Classify targets	Communicate Information	Control and manage aerial	Deploy and recover aircraft	Deploy and recover landing	Detect targets	Determine if course is safe	Enable coordination	Engage targets	Ensure air quality	Generate propulsive	Maintain temperature	Maneuver alongside pier	Neutralize enemy	Neutralize enemy	Provide accomodation	Provide aircraft shelter facilities	Provide cargo handling	Provide electrical power	Provide landing craft shelter
Accommodate and deploy 1 LCU																					↑
Accommodate and deploy 2 LCUAs																					↑
Accommodate and deploy 9 AAVs																					↑
Alter existing course	↑																				
Classify targets		↑																			
Communicate Information			↑																		
Control and manage aerial operations				↑																	
Deploy aircraft					↑																
Deploy or recover two medium-size helicopters simultaneously					↑																
Deploy rescue aircraft					↑																
Deploy the landing craft						↑															
Detect targets							↑														
Determine if course is safe								↑													
Enable coordination and command functions									↑												
Enable the safe lifting of heavy cargo on the deck																			↑		
Ensure air quality											↑										
Generate propulsive power												↑									
Intercept with targets within a close range perimeter										↑											
Lift aircraft																			↑		
Lower aircraft																			↑		
Maintain temperature control													↑								
Maneuver alongside pier														↑							

Configuration

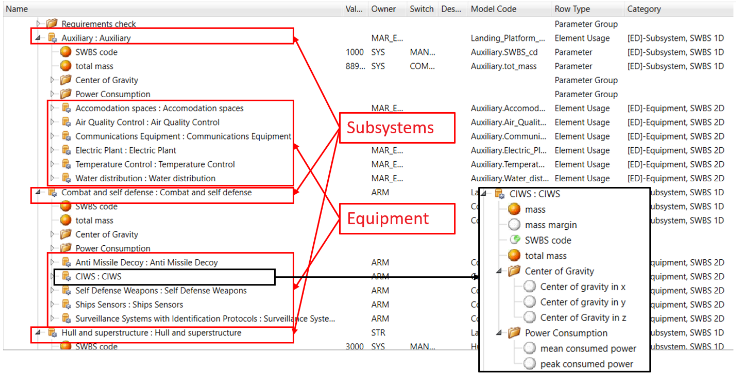
X Axis
Type: ElementDefinition
Categories: Functions;Logical Perspec... AND
☐ Include Subcategories of the selected Cat
Functions AND Logical Perspective
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

Y Axis
Type: ElementDefinition
Categories: Functions;Physical Perspe... AND
☐ Include Subcategories of Selected
Functions AND Physical Perspective
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

Relationship Configuration
Rule: Derives_Fun_Fun

Figure 6.36: Relationship matrix between logical functions and physical functions for the LPD model in CDP4-COMET

At this moment, the Subsystems can be further decomposed. Several pieces of equipment have been defined to perform the physical functions. Each one has been parameterized as indicated in Figure 6.37.



Name	Val...	Owner	Switch	Des...	Model Code	Row Type	Category
Requirements check						Parameter Group	
Auxiliary : Auxiliary		MAR_E...			Landing_Platform_...	Element Usage	[ED]-Subsystem, SWBS 1D
SWBS code	1000	SYS	MAN...		Auxiliary.SWBS_cd	Parameter	[ED]-Subsystem, SWBS 1D
total mass	889...	SYS	COM...		Auxiliary.tot_mass	Parameter	[ED]-Subsystem, SWBS 1D
Center of Gravity						Parameter Group	
Power Consumption						Parameter Group	
Accomodation spaces : Accomodation spaces		MAR_E...			Auxiliary.Accomod...	Element Usage	[ED]-Equipment, SWBS 2D
Air Quality Control : Air Quality Control					Auxiliary.Air_Qualit...	Element Usage	[ED]-Equipment, SWBS 2D
Communications Equipment : Communications Equipment					Auxiliary.Communi...	Element Usage	[ED]-Equipment, SWBS 2D
Electric Plant : Electric Plant		MAR_E...			Auxiliary.Electric_Pl...	Element Usage	[ED]-Equipment, SWBS 2D
Temperature Control : Temperature Control		MAR_E...			Auxiliary.Temperat...	Element Usage	[ED]-Equipment, SWBS 2D
Water distribution : Water distribution		MAR_E...			Auxiliary.Water_dist...	Element Usage	[ED]-Equipment, SWBS 2D
Combat and self defense : Combat and self defense		ARM			La		ubsystem, SWBS 1D
SWBS code					Co		ubsystem, SWBS 1D
total mass					Co		ubsystem, SWBS 1D
Center of Gravity							
Power Consumption							
Anti Missile Decoy : Anti Missile Decoy		ARM			Co		quipment, SWBS 2D
CIWS : CIWS		ARM			C		quipment, SWBS 2D
Self Defense Weapons : Self Defense Weapons		ARM			Co		quipment, SWBS 2D
Ships Sensors : Ships Sensors		ARM			Co		quipment, SWBS 2D
Surveillance Systems with Identification Protocols : Surveillance Syste...		ARM			Co		quipment, SWBS 2D
Hull and superstructure : Hull and superstructure		STR			La		ubsystem, SWBS 1D
SWBS code	3000	SYS	MAN...		Hu		ubsystem, SWBS 1D

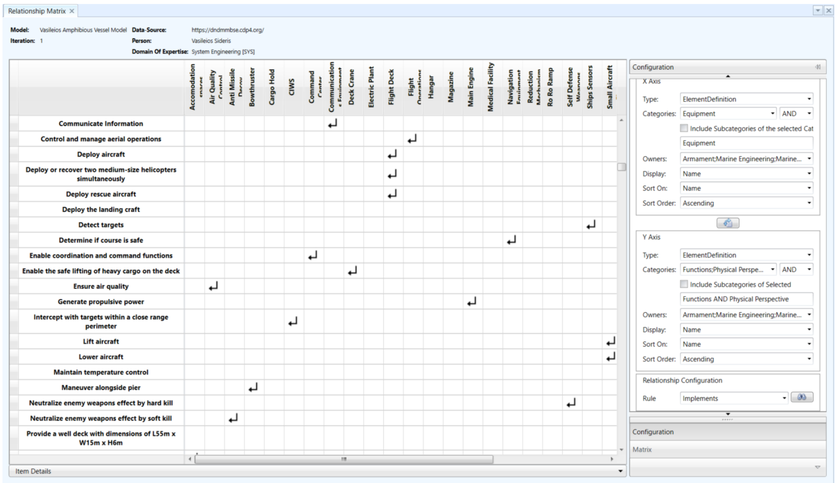
CIWS : CIWS
- mass
- mass margin
- SWBS code
- total mass
- Center of Gravity
  - Center of gravity in x
  - Center of gravity in y
  - Center of Gravity in z
- Power Consumption
  - mean consumed power
  - peak consumed power

Figure 6.37: Selected equipment for the LPD Model in CDP4-COMET

At this stage, it is suggested to focus more on the parameters. Including the same parameters depicted in Figure 6.37 for all equipment was streamlined by copying the Element Definition Blocks. Based on the system architecture and requirements specification all elements representing the equipment elements were parameterized. The specification of each element includes aspects required to calculate the mass budget and the eclectic load budget. Theoretically, each domain expert responsible for the data would be asked to add this data to the model. In this thesis, all values were then manually added to these parameters.

An AI tool, ChatGPT, was used to provide indicative values for mass and electric load. For the latter, both peak and mean consumed power were generated. Additionally, the AI tool provided random but rational values for the center of gravity of each piece of equipment, based on an axis system positioned in the middle of the ship. Furthermore, a mass margin was also provided by the AI tool following the guidelines described in ESA's margin philosophy for science assessment studies (SRE-PA & D-TEC Staff 2012). Furthermore, since many requirements would be verified according to specific parameters as described later, the inclusion of additional parameters for certain equipment was necessary. For example, besides the aforementioned parameters, the equipment called "Flight Deck" was given a Length parameter and a Width parameter for purposes related to the verification of Requirement R5.2.1. Moreover, the value of the SWBS code parameter was also added manually to provide each piece of equipment with a unique ID, enhancing identification. This parameter is also linked to the requirements verification process, as will be described in a later section.

The equipment identified must execute the designated physical functions for the system. This is achieved through the use of an "implement" binary relationship. The mapping is illustrated in Figure 6.38.



Relationship Matrix

Model: Vasileios Amphibious Vessel Model
Iteration: 1
Data-Source: https://dndmmbse.cdp4.org/
Person: Vasileios Sideris
Domain Of Expertise: System Engineering [SYS]

	Accomodation spaces	Air Quality Control	Anti Missile Decoy	Bowthruster	Cargo Hold	CIWS	Command Center	Communication Equipment	Deck Crane	Electric Plant	Flight Deck	Flight Operations	Hangar	Magazine	Main Engine	Medical Facility	Navigation Equipment	Reduction Mechanism	Ro Ro Ramp	Self Defense Weapons	Ships Sensors	Small Aircraft
Communicate Information								↵														
Control and manage aerial operations												↵										
Deploy aircraft											↵											
Deploy or recover two medium-size helicopters simultaneously											↵											
Deploy rescue aircraft											↵											
Deploy the landing craft																						
Detect targets																					↵	
Determine if course is safe																	↵					
Enable coordination and command functions							↵															
Enable the safe lifting of heavy cargo on the deck									↵													
Ensure air quality		↵																				
Generate propulsive power															↵							
Intercept with targets within a close range perimeter						↵																
Lift aircraft																						↵
Lower aircraft																						↵
Maintain temperature control																						
Maneuver alongside pier				↵																		
Neutralize enemy weapons effect by hard kill																				↵		
Neutralize enemy weapons effect by soft kill			↵																			
Provide a well deck with dimensions of L55m x W15m x H6m																						

Item Details

Configuration

X Axis
Type: ElementDefinition
Categories: Equipment AND
□ Include Subcategories of the selected Cat
Equipment
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

Y Axis
Type: ElementDefinition
Categories: Functions;Physical Perspe... AND
□ Include Subcategories of Selected
Functions AND Physical Perspective
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

Relationship Configuration
Rule Implements

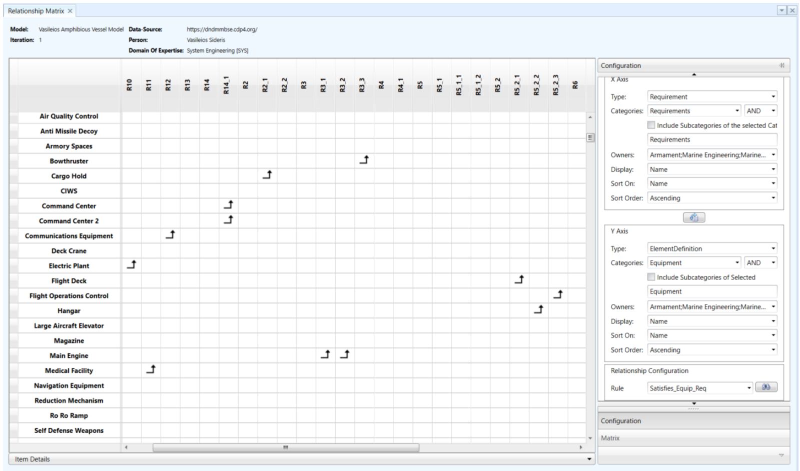
Figure 6.38: Relationship matrix between Physical Functions and Equipment for the LPD model in CDP4-COMET

Finally, to achieve complete traceability, the requirements are connected to the equipment that fulfills them through a "satisfy" linkage. The outcome of the aforementioned binary relationships can be depicted in Figure 6.39.

6.4.7. Balances in CDP4-COMET

While creating an integrated, overall balanced ship design, distinct calculations are dominating many of the integration efforts. Such examples might be the weight balance calculation or the electrical load balance. More specifically, the weight balance calculation is crucial in determining both the total weight as well as the center of gravity of the ship. This data serves as direct input for crucial performance parameters, including ship resistance and propulsion power (determined by the resulting draught of the ship), as well as a ship's intact and damaged stability. The electrical load balance can be used directly to determine the electric power production required to support the ship in all operational modes (Kolfschoten et al. 2024).

Although other balancing activities are equally important, like optimizing available onboard space, hold major significance, these two - weight balance and electrical load balance - offer a distinct advantage. They



Model: Vasileios Amphibious Vessel Model
Iteration: 1
Data-Source: https://dndmmbse.cdp4.org/
Person: Vasileios Sideris
Domain Of Expertise: System Engineering [SYS]

	R10	R11	R12	R13	R14	R14_1	R2	R2_1	R2_2	R3	R3_1	R3_2	R3_3	R4	R4_1	R5	R5_1	R5_1_1	R5_1_2	R5_2	R5_2_1	R5_2_2	R5_2_3	R6
Air Quality Control																								
Anti Missile Decoy																								
Armory Spaces																								
Bowthruster													↑											
Cargo Hold								↑																
CIWS																								
Command Center						↑																		
Command Center 2						↑																		
Communications Equipment			↑																					
Deck Crane																								
Electric Plant	↑																							
Flight Deck																					↑			
Flight Operations Control																							↑	
Hangar																						↑		
Large Aircraft Elevator																								
Magazine																								
Main Engine											↑	↑												
Medical Facility		↑																						
Navigation Equipment																								
Reduction Mechanism																								
Ro Ro Ramp																								
Self Defense Weapons																								

Configuration

X Axis
Type: Requirement
Categories: Requirements AND
□ Include Subcategories of the selected Cat
Requirements
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

Y Axis
Type: ElementDefinition
Categories: Equipment AND
□ Include Subcategories of Selected
Equipment
Owners: Armament;Marine Engineering;Marine...
Display: Name
Sort On: Name
Sort Order: Ascending

Relationship Configuration
Rule Satisfies_Equip_Req

Figure 6.39: Relationship matrix between Requirements and Equipment for the LPD model in CDP4-COMET

can be fully evaluated numerically, presenting an optimal opportunity for inclusion within the COMET model due to its data-centric approach. The calculation results provided by these balances can serve as valuable benchmarks for comparing design modifications and verifying performance parameters. More information on balances is presented in Chapter 7 due to its close relevance to the management of trade-offs.

In this thesis, the weight was derived after modifying an existing balance developed for the DND-MMBSE pilot project (Kolfschoten et al. 2024). The reporting result of the weight balance can be found in Chapter 9, specifically within the Appendix section. Furthermore, it was determined that creating the weight balance is enough to demonstrate the tool's capabilities in that area. Therefore, the creation of the electric load balance was deemed unnecessary and was omitted.

6.4.8. Verification & Validation of the CDP4-COMET Model

The following section addresses the model verification, representing Step 7: Verify & Validate MBSE Models, as specified in the methodology of Chapter 4. It is important to mention that only a summary of the Verification & Validation process followed for the LPD model is presented below. Although it would fit well here and strengthen the current section, it was selected to be described in detail there because model validation and verification are deemed crucial for evaluating consistency and traceability, respectively.

The tool enables the linking of requirements to system decomposition in the LPD vessel model, allowing detailed design refinement for performance calculation. Each of the 40 requirements was manually created and linked with design elements through associated parameters, thus enabling automated verification. Most requirements were evaluated against specific design features, using parametric constraints to assess technological solutions. This involved verifying if particular equipment was included in the model by utilizing SWBS coding. The Microsoft Excel integration was tested for complex arithmetic comparisons, demonstrating the versatility of the tool.

Validation of the CDP4-COMET model is deemed equally crucial as verification. The tool includes a set of built-in and custom validation rules aimed at ensuring the model's setup and consistency. Custom rules, such as verifying parameters like mass and center of gravity, enhance the validation process, minimizing errors in the final model. Advanced checks, including proportionality of mass to size and spatial constraints, further refine the validation process, aiding modelers in identifying complex errors that might otherwise go unnoticed in large-scale systems like a warship.

6.4.9. Modifications to the CDP4-COMET Model

Once the baseline model has been established, it is time to evaluate the versatility of the MBSE environment in capturing and processing evolving requirements that lead to design modifications.

This section will quickly discuss the modifications made to the CDP4-COMET model. It is prudent to correlate this section with the one referring to requirements verification. The modifications are intently designed to address requirements that were not fulfilled in the initial version of the model (baseline model), such as R6.2 and R14.1, ensuring their complete satisfaction.

Before delving deeper into this, the Options feature should be elaborated further. Options are used in a CDP4-COMET to model the identified promising solution directions for the design. Whenever a different engineering solution is investigated within it is possible to create a new Option in the engineering model. In this thesis, two options have been developed, with the second option initially being a replica of the first. The modifications discussed below will eventually lead to a slightly altered Option 2. In summary, the following characteristics differentiate the two Options:

- The Aircraft Elevator in Option 2 has an increased lifting capacity of 25 tons, up from 15 tons in Option 1, meeting the Requirement R6.2.
- Option 2 features a significantly larger Command Center, expanding the total area to 400 square meters, compared to the 100 square meters in Option 1, to comply with R14.1. However, it was assumed that the addition of a larger Command Center area resulted in an increased length of the ship by 10 meters. Thus, R4.1, specifying the total length of the ship, is now flagged in red. This discrepancy suggests a possible conflict between these requirements.
- The removal of the surveillance system in Option 2 led to Requirement R13, which mandates the existence of such a system, to be flagged in yellow. This discrepancy is attributed to the absence of the correct SWBS code in the system, as discussed in the requirements verification section earlier.

Consumer Set Modifications
In exploring Consumer Set Modifications in the CDP4-COMET model, both the addition of new equipment and the removal of existing ones were investigated. The addition of the “Larger Aircraft Elevator” and modification of the “Command Center” were accomplished by adjusting element definitions and setting option-dependent parameters. Removal of the “Surveillance Systems” element in Option 2 led to discrepancies in meeting certain requirements, highlighting conflicts and the need for further adjustments. Additionally, the changes in mass resulting from these modifications were considered and reflected in the mass budget for Option 2.

Model Modifications
In Chapter 4, the addition of a new parameter was chosen to demonstrate how the tool handles model modifications. This is a process that can also be easily executed in CDP4-COMET. All that is required is to locate the parameter in the parameter catalogue or create a new one. Then, by simply dragging and dropping it onto the designated element, the modification is complete. If the parameter falls within our domain of expertise, we can manually adjust its value; otherwise, we can subscribe to it to update its value.

6.4.10. Collecting, Sharing and Analysing Modeling Constructs in CDP4-COMET

The CDP4-COMET server can be used as an MBSE Hub through which SE data can be exchanged. A typical scenario for its use is to create models with the IME Desktop Application and web application. This model data is then exchangeable with various analysis tools like DOORS, MATLAB, Microsoft Excel, Python, and others. During this project, the export of model data was tested through the Microsoft Excel integration.

Additionally, the reporting mechanism of this tool is of great importance. CDP4-COMET offers users the ability to generate custom reports, providing an overview based on data created by the project team. In the case of the LPD project, this feature was demonstrated by creating and exporting the mass balance report.

Existing adapters allow users to interactively exchange data between models built with different MBSE tools such as Capella, Cameo, and Enterprise Architect (Kolfschoten et al. 2024). Although this lays the foundation for the single source of truth concept, the fundamental principle of MBSE, this possibility was not explored in this graduation project.

6.4.11. Insights Gained During CDP4-COMET Modeling

Throughout the verification of requirements, several conclusions can be drawn. The checking mechanism implemented in the CDP4-COMET Requirement Manager, which indicates the agreement of a design parameter with the conditions of the parametric constraint, proves to be an efficient method for automatically verifying design requirements. This feature can also prove valuable in identifying conflicting requirements. This was demonstrated in the section on modifying the model, where Requirement R4.1 and R14.1 appeared to be in conflict.

It should be acknowledged that the requirements can also be automatically verified based on balance calculations. In this thesis, one of the requirements could be linked to a parameter derived from these calculations. For example, a requirement could be linked to the parameter called “total mass”, which is automatically calculated through the mass balance. The verification can be performed within the tool and can also be exported as a report. This allows the team to instantly check, with each modification, whether the resulting design still meets the requirements.

Modifications were introduced during modeling to mimic a real-world design process, where both requirements and system design are adjusted under external influences. During the modifications, it was demonstrated that it is possible to either add new modeling elements or reuse existing ones to create new configurations, represented as new options. This flexibility allows for easy adaptation to evolving requirements.

CDP4-COMET provides the user with the capability to create custom reports. This offers an overview from a specific perspective of interest based on the data that has been created in the model. For this thesis, a mass balance report, also known as a mass report, has been created. This report can be reused after modifying the baseline model to provide insight into the impact of changes on the overall ship and its performance. This makes it easy to keep track of the overall development of key performance aspects throughout the design process. In general, constructing a completely new report can be difficult, and time-consuming, and requires coding knowledge, making it more suitable for advanced users. However, once a balance report is created, it can be updated with minimal effort after each modification. Moreover, it can be reused in other projects with models that follow a similar structure.

Designing warships requires extensive expertise across multiple domains, highlighting the inherent interdependency among stakeholders involved in the process. Therefore, clearly defining and managing responsibilities within an ESSD project is pivotal. Embracing a multidisciplinary approach necessitates that domain experts take ownership of aspects within their respective domains, including requirements, modeling items, and parameters. Although the LPD model did not fully demonstrate this, as only one person—the author—was involved in modeling actions, the model structure was built on solid foundations. Multiple disciplines, such as Armament, Marine Engineering, Marine Navigation, Naval Architecture, Propulsion, Structures, and Systems Engineering, were involved. Subsystems and equipment, along with their parameters, have been allocated to these disciplines. Thus, CDP4-COMET can sufficiently capture interactions between these disciplines, improving collaboration and providing an audit trail through options, iterations, and ownership.

Monitoring and logging changes are important to enable a re-assessing of an earlier design stage and to ensure changes are traceable and understood in their context. Furthermore, monitoring can support creating an overview of the progress and status of the model and its completeness. Ship systems are exceptionally complex, and the ship design process is long and intricate, sometimes extending over several years. This implies that the systems will experience multiple changes introduced by various domain experts throughout the design evolution. With automated monitoring and logging of changes, it becomes possible to provide a clear overview of the modifications over time. When domain experts modify parameter values or introduce new parameters in CDP4-COMET, the changes are indicated visually. After each iteration or session, the System Engineer reviews these changes and publishes them using the Publication Browser in the model tab, providing a transparent overview of modifications sorted by domain. These include old and new values, as well as the percentage of change.

7

Assessment of Tools Based on MBSE Factors

This chapter is dedicated to the final phase of the proposed process, focusing on the evaluation of the selected tools based on the MBSE factors. Motivation for studying each factor is provided, along with an explanation of its significance. Additionally, the means of compliance for checking each factor are outlined. Following this, the assessment of the selected tool is conducted in detail, which includes practical demonstrations and tests performed to evaluate their performance. The results of the comparative analysis are then summarized in a comprehensive table. Through this examination, the Research Question 6 (*"How does the selected MBSE tool-method-language, perform in terms of consistency, traceability, flexibility, and trade-off analysis within the context of system architecture development?"*) can be fully addressed.

7.1. Success Factor: Consistency

First, it should become clear why consistency is an important aspect to investigate in an MBSE tool. In Chapter 3, consistency was linked to two key requirements for early-stage naval vessel design: Better Decision Management and System Architecture Development.

Next, consistency in the context of the investigated tools is explained. A system specification is consistent when it aligns with the modeling syntax and architecture (internal consistency), and the values and meanings of concepts are identical throughout the models (inter-tool consistency) (Kolfschoten et al. 2024). In this thesis, only internal consistency is investigated.

Maintaining internal consistency relies on expertise and discipline by the user to adhere to the correct use of the modeling language when creating or modifying contents in the tools. As explained, the tools should include "validation rules" to enforce adherence to language constructs. Moreover, the tools should allow for the creation of custom rules to detect pattern violations that extend beyond language compliance.

Means of compliance:
To ensure that the tool meets the flexibility criteria, we use the following checks:

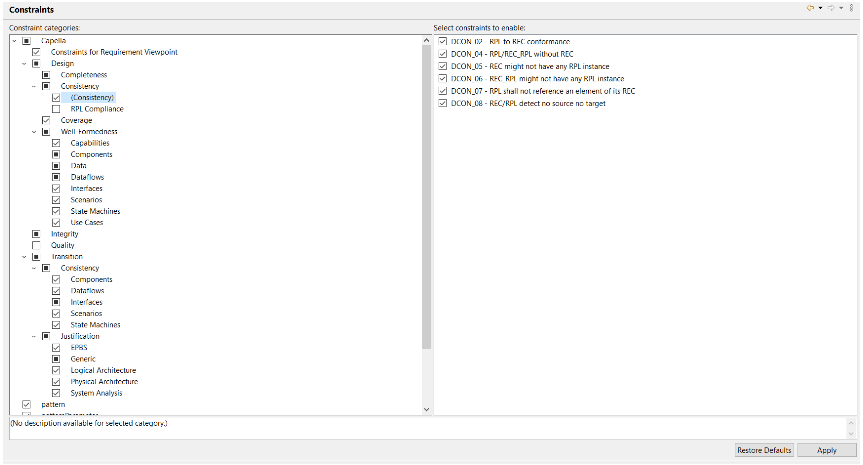
- Internal consistency in the tools is tested using each tool's built-in/custom validation rules for automated checks
- Examples of consistency functionalities offered by various MBSE tools are explored to demonstrate how each tool supports consistency within the modeling process.

7.1.1. Consistency in Capella

- **Validation Rule Assessment: Evaluate consistency leveraging each tool's built-in/custom validation rules for automated checks**

  To avoid mistakes in modeling the tool should have built-in "validation rules" to check whether language constructs are not violated. Capella offers the "Validate model" command through which it runs the selected validation rules. The tool organizes the model validation rules in various categories and subcategories. Capella also provides the option to export both active and inactive rules into a CSV or text file for a better overview.

  In Figure 7.1, the categories of active validation rules during the automatic validation assessment of the model are displayed. Concerning the results of validation through Capella, mistakes of varying severity may occur. The critical mistakes are referred to as “errors,” while the less important ones are shown as “warnings”.



Constraints

Constraint categories:

- ▣ Capella
  - ☑ Constraints for Requirement Viewpoint
  - ▣ Design
    - ▣ Completeness
    - ▣ Consistency
      - ☑ (Consistency)
      - □ RPL Compliance
    - ☑ Coverage
    - ▣ Well-Formedness
      - ☑ Capabilities
      - ▣ Components
      - ▣ Data
      - ▣ Dataflows
      - ☑ Interfaces
      - ☑ Scenarios
      - ☑ State Machines
      - ☑ Use Cases
  - ▣ Integrity
  - □ Quality
  - ▣ Transition
    - ▣ Consistency
      - ☑ Components
      - ☑ Dataflows
      - ▣ Interfaces
      - ☑ Scenarios
      - ☑ State Machines
    - ▣ Justification
      - ☑ EPBS
      - ▣ Generic
      - ☑ Logical Architecture
      - ☑ Physical Architecture
      - ☑ System Analysis
- ☑ pattern

Select constraints to enable:

- ☑ DCON_02 - RPL to REC conformance
- ☑ DCON_04 - RPL/REC_RPL without REC
- ☑ DCON_05 - REC might not have any RPL instance
- ☑ DCON_06 - REC_RPL might not have any RPL instance
- ☑ DCON_07 - RPL shall not reference an element of its REC
- ☑ DCON_08 - REC/RPL detect no source no target

(No description available for selected category.)

Figure 7.1: Categories of active validation rules for the LPD model in Capella

For the LPD model, all four architectural layers have been validated. The default validation rules were used. To proceed to the next architectural level, it was necessary to resolve any identified mistakes categorized as errors, either manually or automatically. As a result, the finished model is free of errors. The warnings are intended to further improve the quality of the model. However, due to the limited time available for this project, addressing these warnings was not prioritized.

In general, Capella relies heavily on its built-in validation rules to maintain consistency within the tool. With over 300 rules integrated into its standard functionality, these rules are categorized into Design and Transition groups, each with various subcategories, as depicted in Figure 7.1. It is noted that rules addressing consistency are available at both the Design and Transition levels. In addition to consistency, completeness, well-formedness, integrity, quality, and justification can also be automatically checked by activating their respective rules. Consequently, users can create different validation profiles to target various aspects of their models.

On the positive side, Capella not only identifies the modeling item that violates a rule but also presents the relevant rule on the side, providing a comprehensive understanding of the root cause of the issue. This feature enables users to address the problem effectively. Additionally, Capella offers quick and automatic solutions (when possible) via the “Quick Fix” command. The latter is very useful for inexperienced users. Finally, the characterization of the violations on the severity level streamlines the modeling procedure by prioritizing corrective actions. On the other hand, based on the author’s experience, they find that the tool restricts the creation of entirely new rules from scratch, limiting customization options to the activation or deactivation of predefined rules.

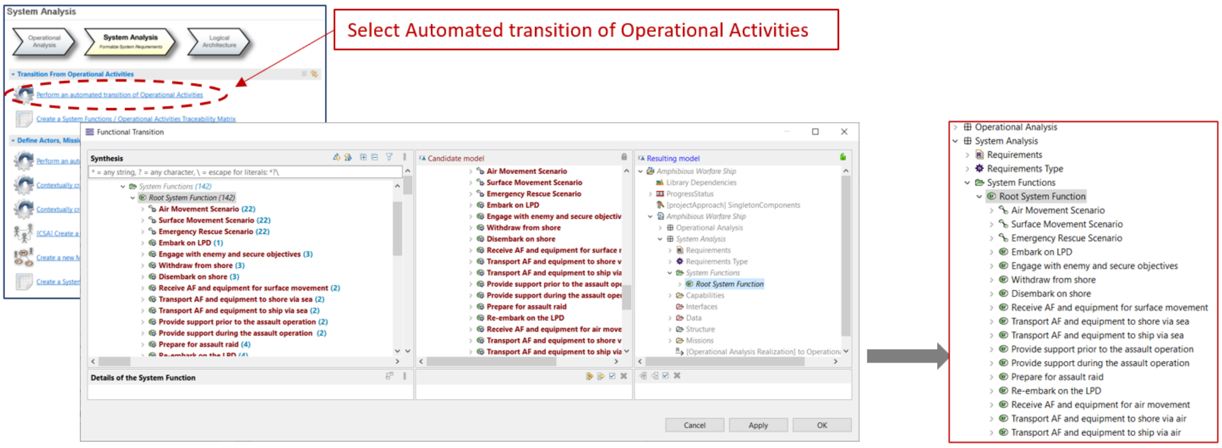
- **Other consistency features: Explore additional examples of consistency functionalities offered by various MBSE tools, demonstrating how each tool supports consistency within the modeling process.**

  A list of several features identified in Capella that ensures consistency are listed below:

  – Capella only allows “correct” use of modeling elements. The Capella tool implements the Arcadia method; a framework built around a set of concepts and templates called viewpoints, which use elements from a structured vocabulary called an ontology. This vocabulary ensures that all model parts are connected correctly through predefined rules or constraints specifying allowable connections between modeling elements. In Capella, only the elements from the ontology that are needed for a specific view are visualized in a diagram. The aforementioned ensures consistency between the model elements.

  – Capella provides automatic synchronization of different views. When changes are made to the model, they are automatically synchronized across all relevant views. This internal connectivity ensures consistency and updates across all views, eliminating the need for manual synchronization efforts. This feature is further elaborated on in a later section, correlating with flexibility.

– Capella allows automatic (or manual) transitions between the different modeling layers. To illustrate this functionality, consider the transition from operational activities to system functions in the LPD model. In this example, operational activities identified in the Operational Analysis layer are automatically transitioned to system functions within the System Analysis layer. This process is straightforward and is visually depicted in Figure 7.2. The Functional Transition Tab provides a comprehensive overview of the impact of this transition, helping in the identification of potential implications and reassuring smooth integration between the several modeling layers.



Select Automated transition of Operational Activities

System Analysis
Operational Analysis
System Analysis
Logical Architecture
Transition From Operational Activities
Perform an automated transition of Operational Activities
Create a System Functions / Operational Activities Traceability Matrix
Define Actors, Missions
Functional Transition
Synthesis
Candidate model
Resulting model
Details of the System Function
Cancel
Apply
OK
Operational Analysis
System Analysis
Requirements
Requirements Type
System Functions
Root System Function
Air Movement Scenario
Surface Movement Scenario
Emergency Rescue Scenario
Embark on LPD
Engage with enemy and secure objectives
Withdraw from shore
Disembark on shore
Receive AF and equipment for surface movement
Transport AF and equipment to shore via sea
Transport AF and equipment to ship via sea
Provide support prior to the assault operation
Provide support during the assault operation
Prepare for assault raid
Re-embark on the LPD
Receive AF and equipment for air movement
Transport AF and equipment to shore via air
Transport AF and equipment to ship via air

Figure 7.2: Transition of Operational Activities for the LPD model in Capella

At this point, it is crucial to mention that the author witnessed occasional errors in the automated transitions during their modeling experience. As a result, each transitioned element was manually checked post-transition. However, it is important to note that this additional step could significantly impact the efficiency of modeling complex systems like a warship.

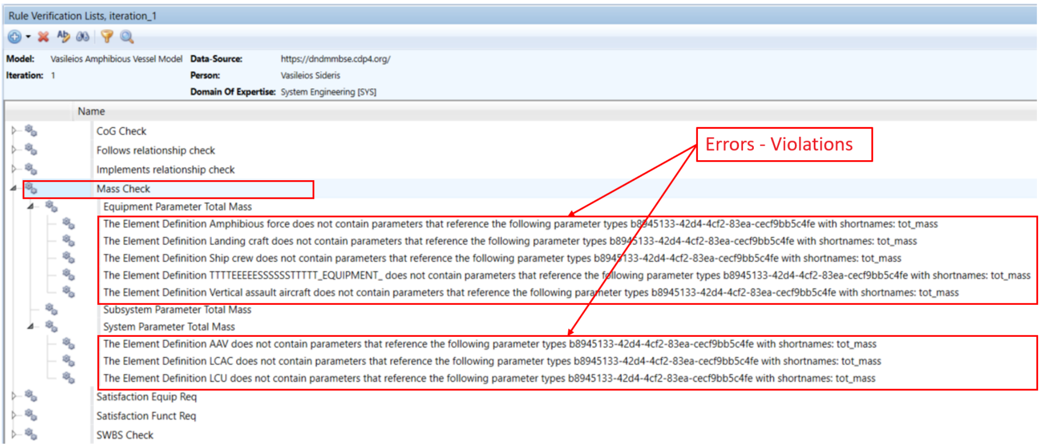
7.1.2. Consistency in CDP4-COMET

- **Validation Rule Assessment: Evaluate consistency leveraging each tool's built-in/custom validation rules for automated checks**

  The capabilities of CDP4-COMET on validation rules are considered equally robust with Capella. CDP4-COMET comes with a set of built-in validation rules, designed to enforce consistency, correctness, well-formedness, and other criteria within the modeling environment. For example, the Element Definition Short Name Rule checks the validity of the short name property for all element definition objects. However, the built-in rules in CDP4-COMET were not further investigated. Instead, the Rules Verification feature was found to be more interesting and is explored in the following paragraph.

  Next to the built-in rules, custom validation rules can be created to catch violations of the design template, described by the Metamodel. These mistakes may not necessarily be language violations. The defined rules intend to inspect the model setup and consistency. In the LPD model in CDP4-COMET, the internal consistency has been automatically verified using the rules verification feature. It should be noted that these rules are referred to as "custom validation rules" in this thesis.

  As it is evident from Figure 7.3, several validation rules that can be automated have been explored. An example of a custom validation rule is the "Mass Check", which states that each system, subsystem, and equipment must have a parameter specifying its total mass (including the margin). Some element definition blocks were intentionally set to violate this rule to test the tool's error-reporting capabilities. As evident from the graph, an error message appears for each element definition that does not include this parameter. Next to the "Mass Check", another rule, the "CoG Check", was created to ensure that each system, subsystem, and equipment has parameters specifying its center of gravity along the three axes. This will ensure that the mass budget is not based on incomplete data. Moreover, a rule was established to verify that each piece of equipment includes an SWBS code for identification, as this is necessary for the verification of most requirements. Additional checks were included to verify the correct application of relationships, ensuring that the source and target elements belong to the correct categories. Apart from the intended violations, no significant errors appeared in the final version of the LPD model.



Rule Verification Lists, iteration_1

Model: Vasileios Amphibious Vessel Model
Iteration: 1
Data-Source: https://dndmmbse.cdp4.org/
Person: Vasileios Sideris
Domain Of Expertise: System Engineering [SYS]

Name
- CoG Check
- Follows relationship check
- Implements relationship check
- Mass Check
  - Equipment Parameter Total Mass
    - The Element Definition Amphibious force does not contain parameters that reference the following parameter types b8945133-42d4-4cf2-83ea-cecf9bb5c4fe with shortnames: tot_mass
    - The Element Definition Landing craft does not contain parameters that reference the following parameter types b8945133-42d4-4cf2-83ea-cecf9bb5c4fe with shortnames: tot_mass
    - The Element Definition Ship crew does not contain parameters that reference the following parameter types b8945133-42d4-4cf2-83ea-cecf9bb5c4fe with shortnames: tot_mass
    - The Element Definition TTTTEEEEESSSSSSTTTTT_EQUIPMENT_ does not contain parameters that reference the following parameter types b8945133-42d4-4cf2-83ea-cecf9bb5c4fe with shortnames: tot_mass
    - The Element Definition Vertical assault aircraft does not contain parameters that reference the following parameter types b8945133-42d4-4cf2-83ea-cecf9bb5c4fe with shortnames: tot_mass
  - Subsystem Parameter Total Mass
  - System Parameter Total Mass
    - The Element Definition AAV does not contain parameters that reference the following parameter types b8945133-42d4-4cf2-83ea-cecf9bb5c4fe with shortnames: tot_mass
    - The Element Definition LCAC does not contain parameters that reference the following parameter types b8945133-42d4-4cf2-83ea-cecf9bb5c4fe with shortnames: tot_mass
    - The Element Definition LCU does not contain parameters that reference the following parameter types b8945133-42d4-4cf2-83ea-cecf9bb5c4fe with shortnames: tot_mass
- Satisfaction Equip Req
- Satisfaction Funct Req
- SWBS Check

Errors - Violations

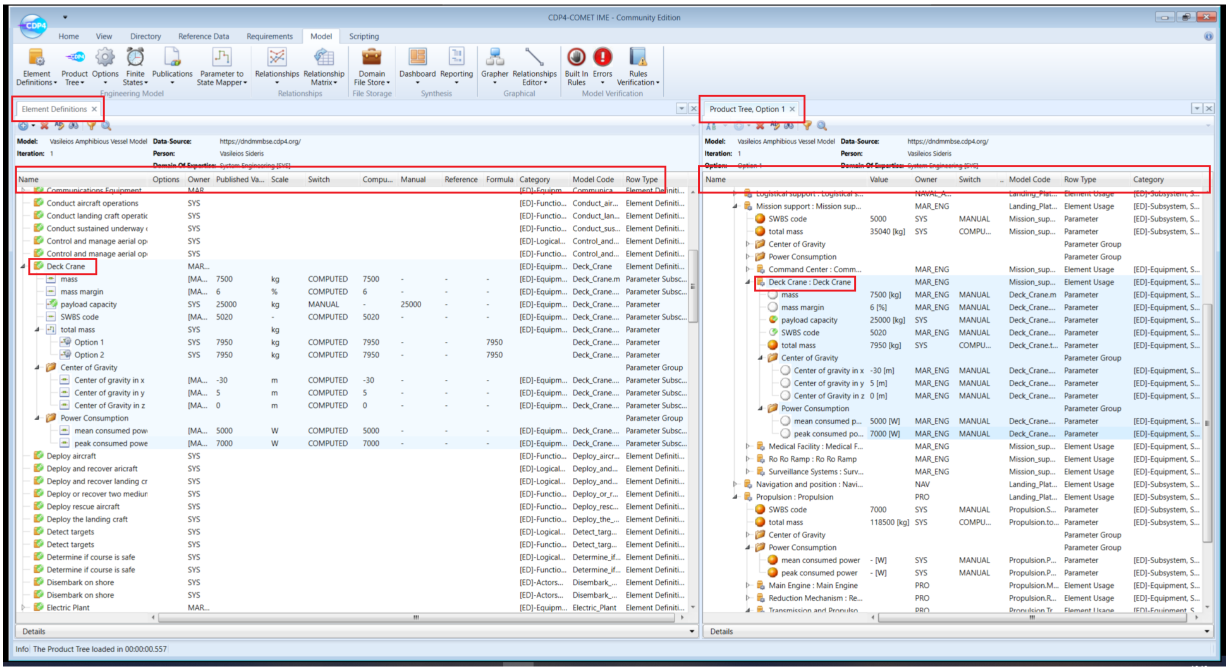
Figure 7.3: Rule Verification List for the LPD model in CDP4-COMET

We can also include more advanced checks, for instance, if the mass of equipment is within proportion to its size or whether the center of gravity of equipment is within the ship. Validation rules can also be made more mathematical, checking for instance whether the calculated volume of an Equipment fits in the calculated volume of a space. These automated validations will help modelers avoid mistakes that are difficult to detect in large systems, such as an entire naval ship.

It can be concluded that the customizability and automatic verification of rules in CDP4-COMET are among its many strengths, allowing the modeler to implement project-specific rules. However, there are a few areas that could benefit from improvement. Firstly, there is no option to add severity levels to the rules, and the tool does not support automatic error fixing. Secondly, unlike Capella, CDP4-COMET does not offer a list of predefined validation rules that can be activated or deactivated according to project needs. Finally, the error messages in CDP4-COMET can sometimes be difficult to interpret, requiring more effort from the modeler to identify and manually correct the errors.

- **Other consistency features: Explore additional examples of consistency functionalities offered by various MBSE tools, demonstrating how each tool supports consistency within the modeling process.**
  - The Element Definitions and the Product Tree browsers, the main views for the model within CDP4-COMET, are also pivotal in visually checking the consistency of the model. The Product Tree describes a specific solution direction with parameters within modeling objects, as defined by the element definitions and element usages in the Element Definitions view (or in the Product Tree directly). Figure 7.4 depicts the Element Definitions and the Product Tree views side by side. Just by looking at these comprehensive views, several things can be checked. For example, data in the Name, Owner, Published Value, Scale, Switch, Category, Model Code, and Row Type columns in the Product Tree can be inspected and modified if needed.

    It is worth noting that CDP4-COMET includes color-coded indications for better clarity. Parameters subscribed to by the active domain are indicated by a white ball. Parameters subscribed to by other domains (but not the active domain) are colored blue. Parameters not used by any domain are colored orange. Additionally, a green indication appears when an Element Definition is used as an Element Usage, and another green indication appears on a parameter when it is linked with a parametric constraint. These colored indicators help to visually identify inconsistencies.
  - Consistency in the model is further facilitated by the practice of copying element definitions. In the LPD model, a templating approach was employed to ensure consistency throughout. Predefined element blocks (Element Definitions) were created, including blocks for functions,



Element Definitions

Model: Vasileios Amphibious Vessel Model Data-Source: https://dndmmbse.cdp4.org/
Iteration: 1 Person: Vasileios Sideris
Domain Of Expertise: System Engineering [SYS]

Name	Options	Owner	Published Va...	Scale	Switch	Compu...	Manual	Reference	Formula	Category	Model Code	Row Type
Communications Equipment		MAR								[ED]-Equipm...	Communica...	Element Definiti...
Conduct aircraft operations		SYS								[ED]-Functio...	Conduct_air...	Element Definiti...
Conduct landing craft operatic		SYS								[ED]-Functio...	Conduct_lan...	Element Definiti...
Conduct sustained underway (		SYS								[ED]-Functio...	Conduct_sus...	Element Definiti...
Control and manage aerial op		SYS								[ED]-Logical...	Control_and...	Element Definiti...
Control and manage aerial op		SYS								[ED]-Functio...	Control_and...	Element Definiti...
Deck Crane		MAR...								[ED]-Equipm...	Deck_Crane	Element Definiti...
mass		[MA...	7500	kg	COMPUTED	7500	-	-	-	[ED]-Equipm...	Deck_Crane.m	Parameter Subsc...
mass margin		[MA...	6	%	COMPUTED	6	-	-	-	[ED]-Equipm...	Deck_Crane....	Parameter Subsc...
payload capacity		SYS	25000	kg	MANUAL	-	25000	-	-	[ED]-Equipm...	Deck_Crane....	Parameter
SWBS code		[MA...	5020	-	COMPUTED	5020	-	-	-	[ED]-Equipm...	Deck_Crane....	Parameter Subsc...
total mass		SYS		kg						[ED]-Equipm...	Deck_Crane....	Parameter
Option 1		SYS	7950	kg	COMPUTED	7950	-	-	7950		Deck_Crane....	Parameter
Option 2		SYS	7950	kg	COMPUTED	7950	-	-	7950		Deck_Crane....	Parameter
Center of Gravity												Parameter Group
Center of gravity in x		[MA...	-30	m	COMPUTED	-30	-	-	-	[ED]-Equipm...	Deck_Crane....	Parameter Subsc...
Center of gravity in y		[MA...	5	m	COMPUTED	5	-	-	-	[ED]-Equipm...	Deck_Crane....	Parameter Subsc...
Center of Gravity in z		[MA...	0	m	COMPUTED	0	-	-	-	[ED]-Equipm...	Deck_Crane....	Parameter Subsc...
Power Consumption												Parameter Group
mean consumed pow		[MA...	5000	W	COMPUTED	5000	-	-	-	[ED]-Equipm...	Deck_Crane....	Parameter Subsc...
peak consumed powe		[MA...	7000	W	COMPUTED	7000	-	-	-	[ED]-Equipm...	Deck_Crane....	Parameter Subsc...
Deploy aircraft		SYS								[ED]-Functio...	Deploy_aircr...	Element Definiti...
Deploy and recover aricraft		SYS								[ED]-Logical...	Deploy_and...	Element Definiti...
Deploy and recover landing cr		SYS								[ED]-Logical...	Deploy_and...	Element Definiti...
Deploy or recover two mediur		SYS								[ED]-Functio...	Deploy_or_r...	Element Definiti...
Deploy rescue aircraft		SYS								[ED]-Functio...	Deploy_resc...	Element Definiti...
Deploy the landing craft		SYS								[ED]-Functio...	Deploy_the_...	Element Definiti...
Detect targets		SYS								[ED]-Logical...	Detect_targ...	Element Definiti...
Detect targets		SYS								[ED]-Functio...	Detect_targ...	Element Definiti...
Determine if course is safe		SYS								[ED]-Logical...	Determine_if...	Element Definiti...
Determine if course is safe		SYS								[ED]-Functio...	Determine_if...	Element Definiti...
Disembark on shore		SYS								[ED]-Actors...	Disembark_...	Element Definiti...
Disembark on shore		SYS								[ED]-Actors...	Disembark_...	Element Definiti...
Electric Plant		MAR...								[ED]-Equipm...	Electric_Plant	Element Definiti...

Product Tree, Option 1

Model: Vasileios Amphibious Vessel Model Data-Source: https://dndmmbse.cdp4.org/
Iteration: 1 Person: Vasileios Sideris
Option: Option 1 Domain Of Expertise: System Engineering [SYS]

Name	Value	Owner	Switch	..	Model Code	Row Type	Category
Logistical support : Logistical s...		NAVAL_A...			Landing_Plat...	Element Usage	[ED]-Subsystem, S...
Mission support : Mission sup...		MAR_ENG			Landing_Plat...	Element Usage	[ED]-Subsystem, S...
SWBS code	5000	SYS	MANUAL		Mission_sup...	Parameter	[ED]-Subsystem, S...
total mass	35040 [kg]	SYS	COMPU...		Mission_sup...	Parameter	[ED]-Subsystem, S...
Center of Gravity						Parameter Group	
Power Consumption						Parameter Group	
Command Center : Comm...		MAR_ENG			Mission_sup...	Element Usage	[ED]-Equipment, S...
Deck Crane : Deck Crane		MAR_ENG			Mission_sup...	Element Usage	[ED]-Equipment, S...
mass	7500 [kg]	MAR_ENG	MANUAL		Deck_Crane.m	Parameter	[ED]-Equipment, S...
mass margin	6 [%]	MAR_ENG	MANUAL		Deck_Crane....	Parameter	[ED]-Equipment, S...
payload capacity	25000 [kg]	SYS	MANUAL		Deck_Crane....	Parameter	[ED]-Equipment, S...
SWBS code	5020	MAR_ENG	MANUAL		Deck_Crane....	Parameter	[ED]-Equipment, S...
total mass	7950 [kg]	SYS	COMPU...		Deck_Crane.t...	Parameter	[ED]-Equipment, S...
Center of Gravity						Parameter Group	
Center of gravity in x	-30 [m]	MAR_ENG	MANUAL		Deck_Crane....	Parameter	[ED]-Equipment, S...
Center of gravity in y	5 [m]	MAR_ENG	MANUAL		Deck_Crane....	Parameter	[ED]-Equipment, S...
Center of Gravity in z	0 [m]	MAR_ENG	MANUAL		Deck_Crane....	Parameter	[ED]-Equipment, S...
Power Consumption						Parameter Group	
mean consumed p...	5000 [W]	MAR_ENG	MANUAL		Deck_Crane....	Parameter	[ED]-Equipment, S...
peak consumed po...	7000 [W]	MAR_ENG	MANUAL		Deck_Crane....	Parameter	[ED]-Equipment, S...
Medical Facility : Medical F...		MAR_ENG			Mission_sup...	Element Usage	[ED]-Equipment, S...
Ro Ro Ramp : Ro Ro Ramp		MAR_ENG			Mission_sup...	Element Usage	[ED]-Equipment, S...
Surveillance Systems : Surv...		MAR_ENG			Mission_sup...	Element Usage	[ED]-Equipment, S...
Navigation and position : Navi...		NAV			Landing_Plat...	Element Usage	[ED]-Subsystem, S...
Propulsion : Propulsion		PRO			Landing_Plat...	Element Usage	[ED]-Subsystem, S...
SWBS code	7000	SYS	MANUAL		Propulsion.S...	Parameter	[ED]-Subsystem, S...
total mass	118500 [kg]	SYS	COMPU...		Propulsion.to...	Parameter	[ED]-Subsystem, S...
Center of Gravity						Parameter Group	
Power Consumption						Parameter Group	
mean consumed power	- [W]	SYS	MANUAL		Propulsion.P...	Parameter	[ED]-Subsystem, S...
peak consumed power	- [W]	SYS	MANUAL		Propulsion.P...	Parameter	[ED]-Subsystem, S...
Main Engine : Main Engine		PRO			Propulsion.M...	Element Usage	[ED]-Equipment, S...
Reduction Mechanism : Re...		PRO			Propulsion.R...	Element Usage	[ED]-Equipment, S...
Transmission and Propulso...		PRO			Propulsion.Tr...	Element Usage	[ED]-Equipment, S...

Info The Product Tree loaded in 00:00:00.557

Figure 7.4: Element Definitions and Product Tree views for the LPD model in CDP4-COMET

subsystems, equipment, and other components. By reusing these templates multiple times, along with their parameters and attributes, the model maintained a consistent structure. Each template was copied and adjusted as needed, but the core elements remained uniform, benefiting standardization and consistency within the model.

7.2. Success Factor: Traceability

Firstly, the importance of studying traceability in an MBSE tool should be highlighted. It is reminded that in Chapter 3, the feature of traceability was linked with one of the key requirements for early-stage naval vessel design, namely Better Decision Management.

Next, there should be a brief explanation of what traceability entails for the investigated tools. Traceability shall be ensured between requirements and between modelling items on the solution side (operational, functional, logical, and physical breakdown). These traces help to create a parsimonious or lean specification; one that contains all that is needed, and nothing that is not needed to meet the requirements (Kolfschoten et al. 2024).

Requirements traceability is at the heart of MBSE. It is established by linking elements within the model to the requirements they originate from. This ensures the model's completeness and forms the foundation for requirements verification.

Means of compliance:
To ensure that the tool meets the traceability criteria, we employ the following checks:

- Traceability of requirements: Explore each tool's capability for manual/automatic verification of requirements
- Other traceability functionalities: Explore other traceability functionalities of the different MBSE tools that are applicable to the projects

7.2.1. Traceability in Capella

- **Traceability of requirements: Explore each tool's capability for manual/automatic verification of requirements**
  The requirements verification capabilities will be explored by demonstrating them on the LPD model. This section addresses the model verification, representing Step 7: Verify & Validate MBSE Models,

as specified in the methodology of Chapter 4.

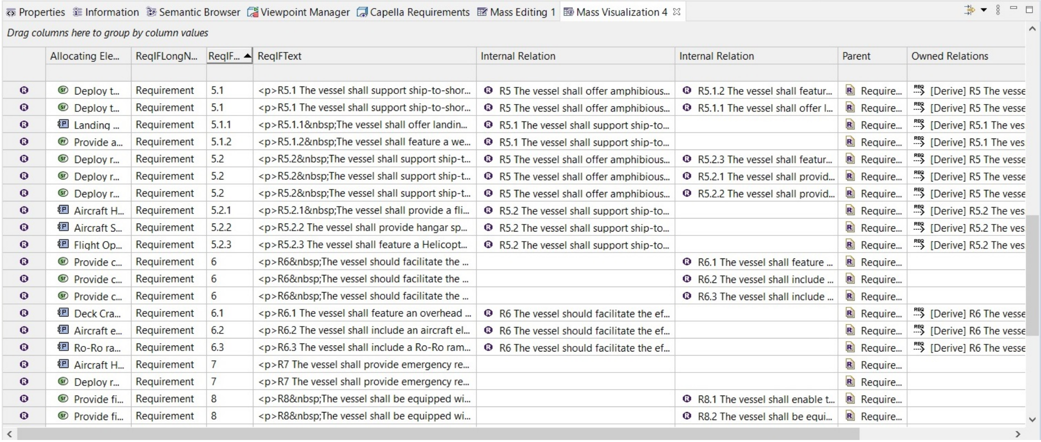
To verify the model, it is essential to ensure it meets its requirements and accurately represents the system being modeled. This is achieved by checking whether the model of the vessel accurately captures the example requirements listed in Table 5.1.

The standard version of Capella does not support requirements visualization. Thus, requirements have been modeled using the "Requirements Viewpoint" add-on. After importing or manually creating requirements, a graphical representation can be generated in most, if not all, diagrams using the Requirements palette. In the case of the LPD vessel, the 40 example requirements were manually created. Each requirement was provided with a description and an ID within the model. Links from requirements to Capella elements can also be established using a "requirement link". In this case, we have included two types of links: "derive" and "satisfy". Once a requirement is linked to a Capella element, the "requirements allocation" tab in the latter will also be populated with the link, completing traceability.

Figure 6.17 confirms that all 40 requirements have been allocated to individual Capella elements. Most requirements refer to general capabilities and are typically phrased to specify a general capability that necessitates particular equipment. By incorporating specific technological solutions, such as particular equipment within the model, we can visually verify the fulfillment of such simple requirements. For instance, according to R8.2, the vessel shall be equipped with a Phalanx Close-in Weapon System (CIWS) to provide defense against anti-ship missiles and other close-range threats. This requirement can be met by including a CIWS system on the vessel.

However, some requirements involve dimensions and arithmetic comparisons during the ESSD. For example, R5.2.1 dictates the minimum dimensions of the flight deck. Such requirements cannot be verified with a model that lacks parametric data. To partially address this issue, these requirements have been linked to the relevant elements to ensure they are at least visually traceable to the correct equipment. For instance, R5.2.1 has been linked to the element representing the flight deck. If the model were parameterized, verification could be completed by manually checking the corresponding parameter values.

Editing or viewing several requirements in a table is also possible. For example, in the Mass Visualization View shown in Figure 7.5, requirements can be sorted and grouped to meet a modeler's needs. This feature is useful for reviewing requirement traceability and analyzing large sets of requirements efficiently.



Drag columns here to group by column values

	Allocating Ele...	ReqIFLongN...	ReqIF... ▲	ReqIFText	Internal Relation	Internal Relation	Parent	Owned Relations
R	Deploy t...	Requirement	5.1	<p>R5.1 The vessel shall support ship-to-shor...	R5 The vessel shall offer amphibious...	R5.1.2 The vessel shall featur...	Require...	[Derive] R5 The vesse
R	Deploy t...	Requirement	5.1	<p>R5.1 The vessel shall support ship-to-shor...	R5 The vessel shall offer amphibious...	R5.1.1 The vessel shall offer l...	Require...	[Derive] R5 The vesse
R	Landing ...	Requirement	5.1.1	<p>R5.1.1 The vessel shall offer landin...	R5.1 The vessel shall support ship-to...		Require...	[Derive] R5.1 The ves
R	Provide a...	Requirement	5.1.2	<p>R5.1.2 The vessel shall feature a we...	R5.1 The vessel shall support ship-to...		Require...	[Derive] R5.1 The ves
R	Deploy r...	Requirement	5.2	<p>R5.2 The vessel shall support ship-t...	R5 The vessel shall offer amphibious...	R5.2.3 The vessel shall featur...	Require...	[Derive] R5 The vesse
R	Deploy r...	Requirement	5.2	<p>R5.2 The vessel shall support ship-t...	R5 The vessel shall offer amphibious...	R5.2.1 The vessel shall provid...	Require...	[Derive] R5 The vesse
R	Deploy r...	Requirement	5.2	<p>R5.2 The vessel shall support ship-t...	R5 The vessel shall offer amphibious...	R5.2.2 The vessel shall provid...	Require...	[Derive] R5 The vesse
R	Aircraft H...	Requirement	5.2.1	<p>R5.2.1 The vessel shall provide a fli...	R5.2 The vessel shall support ship-to...		Require...	[Derive] R5.2 The ves
R	Aircraft S...	Requirement	5.2.2	<p>R5.2.2 The vessel shall provide hangar sp...	R5.2 The vessel shall support ship-to...		Require...	[Derive] R5.2 The ves
R	Flight Op...	Requirement	5.2.3	<p>R5.2.3 The vessel shall feature a Helicopt...	R5.2 The vessel shall support ship-to...		Require...	[Derive] R5.2 The ves
R	Provide c...	Requirement	6	<p>R6 The vessel should facilitate the ...		R6.1 The vessel shall feature ...	Require...	
R	Provide c...	Requirement	6	<p>R6 The vessel should facilitate the ...		R6.2 The vessel shall include ...	Require...	
R	Provide c...	Requirement	6	<p>R6 The vessel should facilitate the ...		R6.3 The vessel shall include ...	Require...	
R	Deck Cra...	Requirement	6.1	<p>R6.1 The vessel shall feature an overhead ...	R6 The vessel should facilitate the ef...		Require...	[Derive] R6 The vesse
R	Aircraft e...	Requirement	6.2	<p>R6.2 The vessel shall include an aircraft el...	R6 The vessel should facilitate the ef...		Require...	[Derive] R6 The vesse
R	Ro-Ro ra...	Requirement	6.3	<p>R6.3 The vessel shall include a Ro-Ro ram...	R6 The vessel should facilitate the ef...		Require...	[Derive] R6 The vesse
R	Aircraft H...	Requirement	7	<p>R7 The vessel shall provide emergency re...			Require...	
R	Deploy r...	Requirement	7	<p>R7 The vessel shall provide emergency re...			Require...	
R	Provide fi...	Requirement	8	<p>R8 The vessel shall be equipped wi...		R8.1 The vessel shall enable t...	Require...	
R	Provide fi...	Requirement	8	<p>R8 The vessel shall be equipped wi...		R8.2 The vessel shall be equi...	Require...	

Figure 7.5: Grouping requirements in Mass Visualization View for the LPD model in Capella

It should be noted that once requirements are linked to an element, they can quickly be displayed in a different view (diagram) or on a different element of another view with just one click using the "All linked Requirements" tool. However, this functionality is only available within the same architectural level where the link was created.

In general, automatic verification of requirements was not found to be possible in the tool, at least in the standard version. However, the Requirements Viewpoint add-on was used. It is a useful add-on that allows for a visual display of the requirements. For the specific case of the LPD, the verification of requirements was approached such that every requirement displayed and successfully linked to an element with a satisfaction relationship was considered fulfilled. A limitation of this approach is that it only applies to functional requirements and cannot fully address requirements that involve arithmetic calculations and comparisons.

- **Other traceability functionalities: Explore other traceability functionalities of the different MBSE tools that are applicable to the projects**

  – A functional chain is an ordered set of references towards functions and functional exchanges. Functional chains are utilized to describe the system behavior in specific usage contexts by delineating one or more system capabilities. Creating a functional chain enhances traceability by establishing clear connections between the selected functions and their interactions. Once a functional chain is created in Capella, it can be easily viewed in other diagrams, aiding in tracking function interactions. Traces generated via functional chains enable analysis of change impacts on the system by visualizing dependencies between functions. By identifying affected or critical functions, one can assess broader implications across the system.

    For the LPD model, the use of function chains allowed us to define how the functions interact in certain use-case scenarios of the system, including Surface Movement, Air Movement, and Emergency Rescue scenarios. The aforementioned chains are visualized in the Physical Data Flow Diagram shown in Figure 7.6. Using this part of the diagram as an example, let's assume that the vessel has sustained damage, rendering it incapable of executing the "lift aircraft" or "lower aircraft" functions, located on the lower left-hand side of the figure. By examining the visualized functional chains, we can readily conclude that such a failure or malfunction of the aircraft lift system would only impact the Emergency Rescue scenario, represented by the blue functional chain. Thus, leveraging functional chains enables the identification of critical functions and facilitates vulnerability analysis within the system, among other benefits.

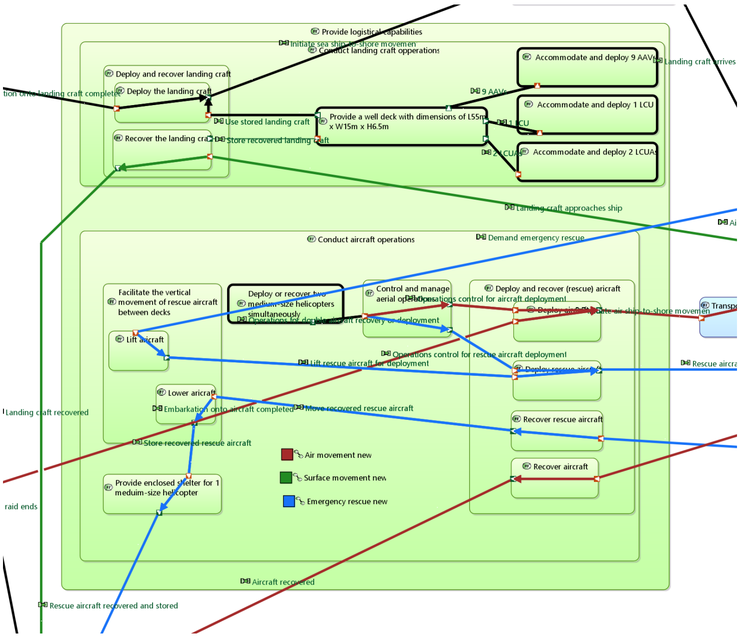
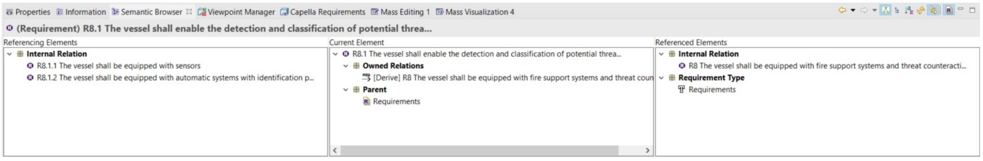

Provide logistical capabilities
Initiate sea ship-to-shore movemen
Conduct landing craft opperations
Deploy and recover landing craft
Deploy the landing craft
tion onto landing craft complete
Recover the landing cra
Use stored landing craft
Store recovered landing craft
Provide a well deck with dimensions of L55m x W15m x H6.5m
Accommodate and deploy 9 AAVs
Accommodate and deploy 1 LCU
Accommodate and deploy 2 LCUAs
9 AAVs
1 LCU
2 LCUAs
Landing craft arrives
Landing craft approaches ship
Ai
Conduct aircraft operations
Demand emergency rescue
Facilitate the vertical movement of rescue aircraft between decks
Lift aircraft
Lower aricraft
Deploy or recover two medium-size helicopters simultaneously
Operations for double aircraft recovery or deployment
Control and manage aerial operations
Operations control for aircraft deployment
Deploy and recover (rescue) aricraft
Deploy aircraft
Initiate air ship-to-shore movemen
Transp
Operations control for rescue aircraft deployment
Lift rescue aircraft for deployment
Deploy rescue aircraft
Rescue aircra
Embarkation onto aircraft completed
Move recovered rescue aircraft
Recover rescue aircraft
Recover aircraft
Store recovered rescue aircraft
Landing craft recovered
raid ends
Provide enclosed shelter for 1 meduim-size helicopter
🟥 Air movement new
🟩 Surface movement new
🟦 Emergency rescue new
Aircraft recovered
Rescue aircraft recovered and stored

Figure 7.6: [PDFD] Physical Data Flow Diagram (provide logistical capabilities) for the LPD model in Capella

  – Capella excels in linking diagrams together. The semantic browser aids in tracing parent/child relationships between any type of modeling element. Note also that the semantic browser allows us to navigate quickly to objects related to the focused object with a single click. For example, the requirement R8.1 and its attributes can be seen in the project explorer and semantic browser, as shown in Figure 7.7.



(Requirement) R8.1 The vessel shall enable the detection and classification of potential threa...

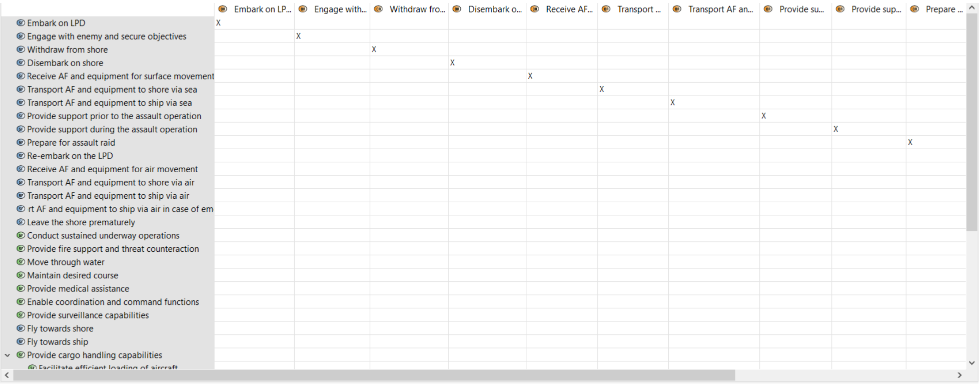
Referencing Elements
- Internal Relation
  - R8.1.1 The vessel shall be equipped with sensors
  - R8.1.2 The vessel shall be equipped with automatic systems with identification p...

Current Element
- R8.1 The vessel shall enable the detection and classification of potential threa...
  - Owned Relations
    - [Derive] R8 The vessel shall be equipped with fire support systems and threat coun
  - Parent
    - Requirements

Referenced Elements
- Internal Relation
  - R8 The vessel shall be equipped with fire support systems and threat counteracti...
- Requirement Type
  - Requirements

Figure 7.7: Requirements in Semantic Browser for the LPD model in Capella

- The transition functionalities of Capella from one architectural layer to the next are crucial not only for consistency but also for traceability. Automatic or manual transitions in Capella allow for the redesign of parts while maintaining traceability to the required functions. This ensures that changes at one layer are accurately reflected and linked to corresponding elements in the following layers.
- Another important tool that enhances traceability in the model is the Traceability Matrices. These matrices enable users to not only inspect but also create traces between elements from different architectural levels. For example, the traceability matrix for operational activities and system functions, depicted in Figure 7.8, illustrates this capability.



	Embark on LP...	Engage with...	Withdraw fro...	Disembark o...	Receive AF...	Transport ...	Transport AF an...	Provide su...	Provide sup...	Prepare ...
Embark on LPD	X									
Engage with enemy and secure objectives		X								
Withdraw from shore			X							
Disembark on shore				X						
Receive AF and equipment for surface movement					X					
Transport AF and equipment to shore via sea						X				
Transport AF and equipment to ship via sea							X			
Provide support prior to the assault operation								X		
Provide support during the assault operation									X	
Prepare for assault raid										X
Re-embark on the LPD										
Receive AF and equipment for air movement										
Transport AF and equipment to shore via air										
Transport AF and equipment to ship via air										
rt AF and equipment to ship via air in case of em										
Leave the shore prematurely										
Conduct sustained underway operations										
Provide fire support and threat counteraction										
Move through water										
Maintain desired course										
Provide medical assistance										
Enable coordination and command functions										
Provide surveillance capabilities										
Fly towards shore										
Fly towards ship										
Provide cargo handling capabilities										
Facilitate efficient loading of aircraft										

Figure 7.8: Traceability Matrix linking operational activities to system functions for the LPD model in Capella

7.2.2. Traceability in CDP4-COMET

- **Traceability of requirements: Explore each tool's capability for manual/automatic verification of requirements**

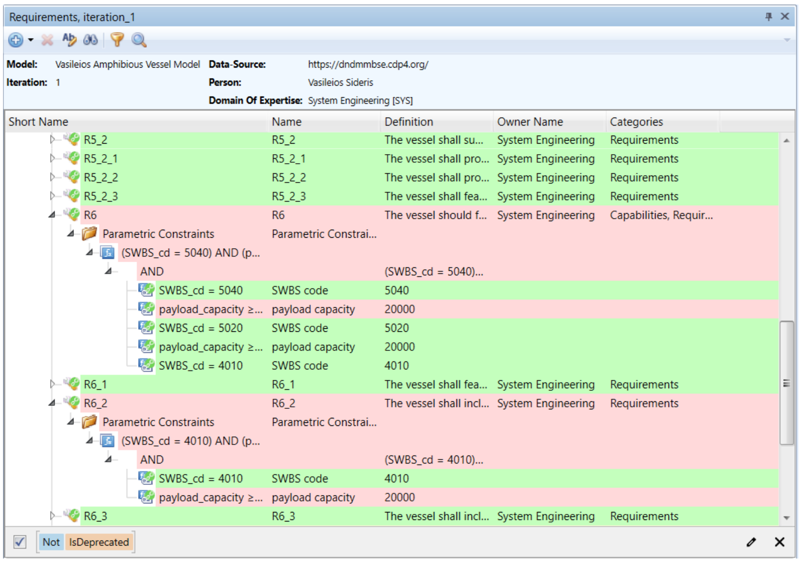
  The requirements verification capabilities will be explored by demonstrating them on the LPD model. The following section addresses the model verification, representing Step 7: Verify & Validate MBSE Models, as specified in the methodology of Chapter 4.

  The tool can be used to link the requirements to the system decomposition and detail the latter to a point where we can calculate the performance of the design. In the case of the LPD vessel, the 40 example requirements were manually created. Each requirement was provided with a description and an ID code within the model. CDP4-COMET enables the user to capture requirements and establish direct links between these requirements and design elements through their parameters. The latter allows automated verification of requirements throughout the design process. Thus, all requirements have been linked with at least one parameter in CDP4-COMET.

  It is reminded that most requirements refer to general capabilities and are typically phrased to specify a general capability that includes particular equipment. For instance, according to requirement R8.2, the vessel shall be equipped with a Phalanx Close-in Weapon System (CIWS) to provide defense against anti-ship missiles and other close-range threats. This requirement can be fulfilled by including a CIWS system on the vessel. Here, the SWBS coding proves useful. The tool checks if the specific SWBS code representing a CIWS is included in the model. If such a code is present, the requirement will be verified and marked green. If the requirement is not met, it will be marked red. If the parameter is not found in the model, the requirement will be marked in yellow. In cases where the inclusion of existing equipment is under investigation, the presence of a yellow mark indicating the absence of this equipment will signify that the requirement is not met.

The integration with Microsoft Excel was also tested. This integration allows for importing data into Excel, interacting with it, and exporting it back to the CDP4-COMET server. In this thesis, the potential of this integration was evaluated by using it to verify a set of requirements, specifically requirements R5 through R5.2.3. The verification process for this set of requirements required several arithmetic comparisons, which were performed in Excel. The results were then exported back to the tool as a boolean parameter. Although this method is unconventional for verifying requirements, it provided an opportunity to explore the potential of the Excel add-in.

After automatically verifying the requirements, it appears that R14.1 and R6.2 are not met, as they are highlighted in red. These requirements stem from R14 and R6, respectively, so it is logical that the latter are also not met. Taking a closer look at each one, starting with R6.2, this requirement states: "The vessel shall include an aircraft elevator with a capacity of 20 tons to facilitate the vertical transportation of helicopters". This necessitates the inclusion of a SWBS code representing the aircraft elevator in the model. Additionally, the aircraft elevator must have a lifting capacity of 20 tons or more. Therefore, an AND expression is used to represent the appropriate parametric constraint. Despite the presence of the aircraft elevator in the vessel (verified by the presence of the correct SWBS code), its capacity falls short at 15 tons, failing to meet both criteria. As a result, the requirement is flagged in red. This is illustrated in Figure 7.9. It is important to note that the initial ship configuration is being verified here. As will be discussed later, CDP4-COMET provides the ability to compare design alternatives defined in different configurations, referred to as Options in the tool.


Requirements, iteration_1

Model: Vasileios Amphibious Vessel Model **Data-Source:** https://dndmmbse.cdp4.org/
Iteration: 1 **Person:** Vasileios Sideris
Domain Of Expertise: System Engineering [SYS]

Short Name	Name	Definition	Owner Name	Categories
R5_2	R5_2	The vessel shall su...	System Engineering	Requirements
R5_2_1	R5_2_1	The vessel shall pro...	System Engineering	Requirements
R5_2_2	R5_2_2	The vessel shall pro...	System Engineering	Requirements
R5_2_3	R5_2_3	The vessel shall fea...	System Engineering	Requirements
R6	R6	The vessel should f...	System Engineering	Capabilities, Requir...
Parametric Constraints	Parametric Constrai...			
(SWBS_cd = 5040) AND (p...				
AND		(SWBS_cd = 5040)...		
SWBS_cd = 5040	SWBS code	5040		
payload_capacity ≥...	payload capacity	20000		
SWBS_cd = 5020	SWBS code	5020		
payload_capacity ≥...	payload capacity	20000		
SWBS_cd = 4010	SWBS code	4010		
R6_1	R6_1	The vessel shall fea...	System Engineering	Requirements
R6_2	R6_2	The vessel shall incl...	System Engineering	Requirements
Parametric Constraints	Parametric Constrai...			
(SWBS_cd = 4010) AND (p...				
AND		(SWBS_cd = 4010)...		
SWBS_cd = 4010	SWBS code	4010		
payload_capacity ≥...	payload capacity	20000		
R6_3	R6_3	The vessel shall incl...	System Engineering	Requirements

☑ Not IsDeprecated

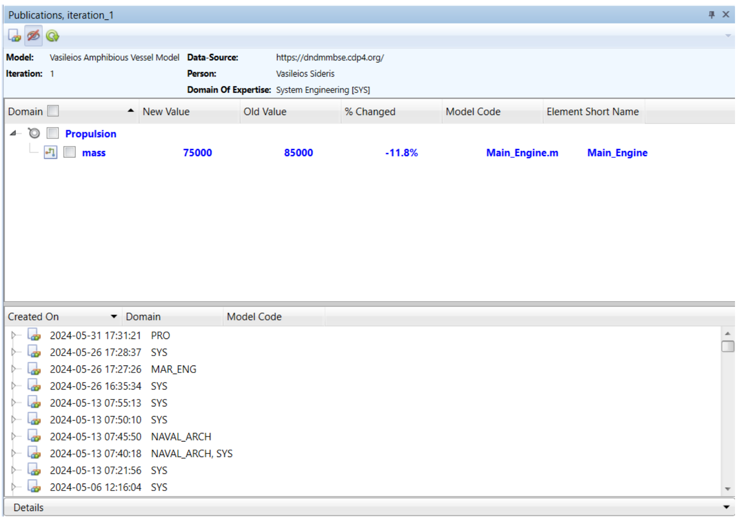
Figure 7.9: Requirements verification (Option 1) for the LPD model in CDP4-COMET

For R14.1, specifying: "The vessel must incorporate an Operations Command Area spanning 400 square meters, capable of accommodating up to 100 military staff", a parameter concerning the area has been incorporated into the element representing the Operations Command Area. This same parameter has been linked to R14.1. Upon checking the value of this parameter, it is found to be only 100 square meters. Consequently, during requirement verification, R14.1 is flagged in red.

In general, CDP4-COMET excels in automatic requirements verification, with the Microsoft Excel integration further enhancing this capability. CDP4-COMET can verify requirements of every kind, from simple functional to complex quantitative design requirements. However, the manual creation and direct linking of new parameters solely for verification, along with establishing parametric constraints for each requirement, can be a time-consuming and labor-intensive process.

- **Other traceability functionalities: Explore other traceability functionalities of the different MBSE tools that are applicable to the projects**

  - In CDP4-COMET, traceability is enabled through defined binary relationships. These relationships include "Derives", "Follows", "Implements", and "Satisfies". Traceability within this tool can be effectively demonstrated through matrices and tables, as showcased in Chapter 6. These matrices and tables display the established binary relationships among model elements. However, while the visualization capabilities in tables within the tool were explored, it was found that the functionality of such relationships is somewhat limited to visual purposes only.
  - Traceability can also be promoted by enabling monitoring and logging changes throughout the design process. Monitoring and logging of changes is important for re-assessing an earlier design stage and ensuring that changes are documented. With automated monitoring and logging of changes, it becomes possible to provide a clear overview of the model status and its modifications over time. When domain experts modify parameter values or introduce new parameters in CDP4-COMET, the changes are indicated visually. These changes are reviewed and published using the Publication Browser in the model tab, as depicted in Figure 7.10. The browser provides a transparent overview of the changes organized by domain expertise. The old and new values, as well as the percentage of change, are also displayed. The bottom part of the browser contains the history of Publications.



Publications, iteration_1

Model: Vasileios Amphibious Vessel Model **Data-Source:** https://dndmmbse.cdp4.org/
Iteration: 1 **Person:** Vasileios Sideris
Domain Of Expertise: System Engineering [SYS]

Domain □	New Value	Old Value	% Changed	Model Code	Element Short Name
□ Propulsion					
□ mass	75000	85000	-11.8%	Main_Engine.m	Main_Engine

Created On	Domain	Model Code
2024-05-31 17:31:21	PRO	
2024-05-26 17:28:37	SYS	
2024-05-26 17:27:26	MAR_ENG	
2024-05-26 16:35:34	SYS	
2024-05-13 07:55:13	SYS	
2024-05-13 07:50:10	SYS	
2024-05-13 07:45:50	NAVAL_ARCH	
2024-05-13 07:40:18	NAVAL_ARCH, SYS	
2024-05-13 07:21:56	SYS	
2024-05-06 12:16:04	SYS	

Details

Figure 7.10: Publications browser for the LPD model in CDP4-COMET

7.3. Success Factor: Flexibility

First, the significance of investigating flexibility in an MBSE tool should be established. As discussed in Chapter 3, the aspect of flexibility was linked with two key requirements for early-stage naval vessel design: Adaptability to Externalities and Novelty and Innovation.

Following that, a concise elucidation of flexibility in the context of the investigated tools is required. A system specification is regarded as flexible when it can be easily modified and adapted to ensure consistent changes throughout the associated models. Moreover, flexibility comprises the ability to swiftly and efficiently implement changes that impact a large number of elements (Kolfschoten et al. 2024).

Flexibility is inherent in MBSE, ensuring that the modeling process remains adaptable and responsive to evolving requirements throughout the development process. The objective of flexibility assessment in this project is to demonstrate the model's integration into the real-world ship design process, where both requirements and system design are varied under external influences while designing.

Means of compliance:
To ensure that the tool meets the flexibility criteria, we employ the following checks:

- Modification analysis: Measure the number of manual actions to incorporate changes to the models using the Design Modifications
- Check model structure and tool features: Report reusable and non-reusable elements

7.3.1. Flexibility in Capella

- **Modification analysis: Measure the number of manual actions to incorporate changes to the models using the Design Modifications**

  Consumer Set Modifications
  To practically explore the tool's flexibility, it was more effective to perform design modifications. The example of adding an anchor is investigated below. Since a large part of the Capella model extends beyond the Physical Architecture, we chose to start by adding a new function in the System Analysis. The necessary work in each layer to modify the vessel so that this change propagates all the way up to the Physical Architecture is presented below.

  1. A new system function, "Secure position of the vessel during the assault", is added to the SAB diagram along with a functional exchange. This function is then transferred to the next layer using the transition option.
  2. We create a new logical component named "Anchor" in the LAB diagram. The transitioned function is then allocated to this component using the "Allocate Function" button. As before, this logical component is subsequently transitioned to the Physical Architecture.
  3. We create a node physical component, called "Anchor" within the Navigation Subsystem. Within this newly created element, we deploy the transitioned behaviour physical component. Finally, we click on "Allocated Functions" to visually present the transitioned physical function "Secure position of the vessel during the assault".

  It is concluded that although many modeling elements were created, the process was straightforward and well-guided by the tool. The entire process can be divided into the three aforementioned steps and took approximately seven minutes. It is important to note that during the aforementioned steps, the rest of the diagrams were automatically updated. Additionally, the tool accelerated the process by offering smart shortcuts like the "Allocated Functions" command in the Palette, which instantly shows the suggested functions that can be used. These aspects highlight the power of having a single source of truth to make changes consistently throughout the associated models with minimal effort and in no time. Illustrative screenshots from the modeling environment during the aforementioned steps are aggregated in Figure 7.11.

  Now the removal of existing equipment will be investigated. To illustrate this, the previously added anchor will be removed from the Physical Architecture. The process is so straightforward that it does not necessitate a list of steps. All we have to do is select the Node Physical Component, the Behavioral Physical Component, as well as the System Function of the anchor, and click "delete from the model". The interesting aspect of this simple process is the window that appears, analyzing the impact of the deletion, as depicted in Figure 7.12. In this window, Capella indicates the elements that will be deleted, along with a view of the referencing elements. This provides a clear overview of how the modification impacts the resulting model.

  Overall, Capella makes it very easy to add, remove, or modify elements visually. However, parameter modification is expected to be a challenging and time-consuming task. It is important to note that this aspect was not practically explored in this project due to time constraints, as the required effort would not correspond to the scientific contribution. Consequently, the flexibility factor in Capella is assessed without considering parameter modifications.

- **Check model structure and tool features: Report reusable and non-reusable elements**
  - It should be clear that it is not necessary to create new elements in each diagram; instead, we can organize the existing ones into a new diagram. Moreover, elements from the previous layer (capabilities, functions, functional exchanges, requirements, logical components, functional chains, etc.) can be transitioned and readily used again. Since the transition aspect has already been covered, no further explanation is needed.

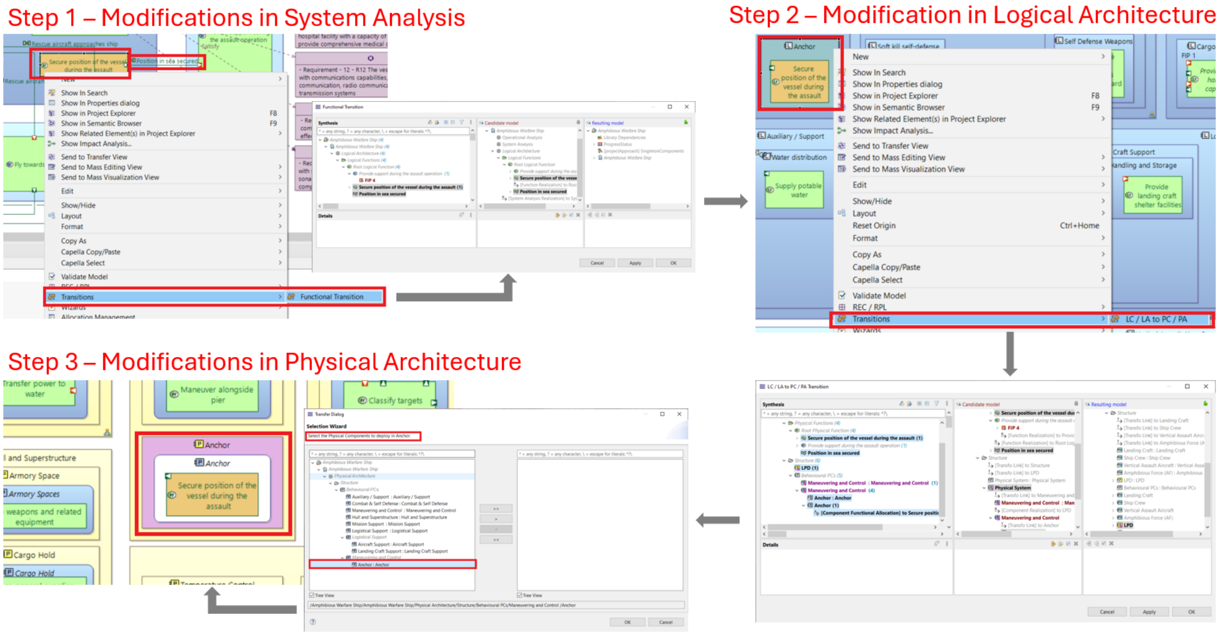
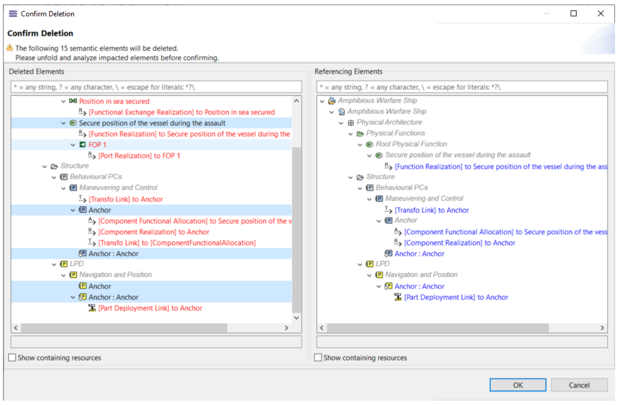

Step 1 – Modifications in System Analysis
Step 2 – Modification in Logical Architecture
Step 3 – Modifications in Physical Architecture
Transitions
Functional Transition
LC / LA to PC / PA

Figure 7.11: Steps followed during the addition of new equipment for the LPD model in Capella


Confirm Deletion

Confirm Deletion

The following 15 semantic elements will be deleted.
Please unfold and analyze impacted elements before confirming.

Deleted Elements

```
* = any string, ? = any character, \ = escape for literals: *?\
      Position in sea secured
         [Functional Exchange Realization] to Position in sea secured
      Secure position of the vessel during the assault
         [Function Realization] to Secure position of the vessel during the
         FOP 1
            [Port Realization] to FOP 1
  Structure
    Behavioural PCs
      Maneuvering and Control
         [Transfo Link] to Anchor
         Anchor
            [Component Functional Allocation] to Secure position of the v
            [Component Realization] to Anchor
            [Transfo Link] to [ComponentFunctionalAllocation]
         Anchor : Anchor
    LPD
      Navigation and Position
         Anchor
         Anchor : Anchor
            [Part Deployment Link] to Anchor
```

□ Show containing resources

Referencing Elements

```
* = any string, ? = any character, \ = escape for literals: *?\
Amphibious Warfare Ship
  Amphibious Warfare Ship
    Physical Architecture
      Physical Functions
        Root Physical Function
          Secure position of the vessel during the assault
             [Function Realization] to Secure position of the vessel during the ass
      Structure
        Behavioural PCs
          Maneuvering and Control
             [Transfo Link] to Anchor
             Anchor
                [Component Functional Allocation] to Secure position of the vess
                [Component Realization] to Anchor
             Anchor : Anchor
        LPD
          Navigation and Position
             Anchor : Anchor
                [Part Deployment Link] to Anchor
```

□ Show containing resources

OK Cancel

Figure 7.12: Delete confirmation tab appeared during the removal of existing equipment for the LPD model in Capella

– Cloning diagrams significantly reduces workload. In the LPD model, the three scenarios shared similar diagrams. Thus, by cloning one, we avoided modeling everything from scratch for the remaining two.

7.3.2. Flexibility in CDP4-COMET

- **Modification analysis: Measure the number of manual actions to incorporate changes to the models using the Design Modifications**

Consumer Set Modifications

To practically explore the tool's flexibility, it was more effective to perform design modifications. Similar to the work done in the Capella, to demonstrate Consumer Set Modifications, we investigate both the addition of new equipment and the removal of existing equipment.

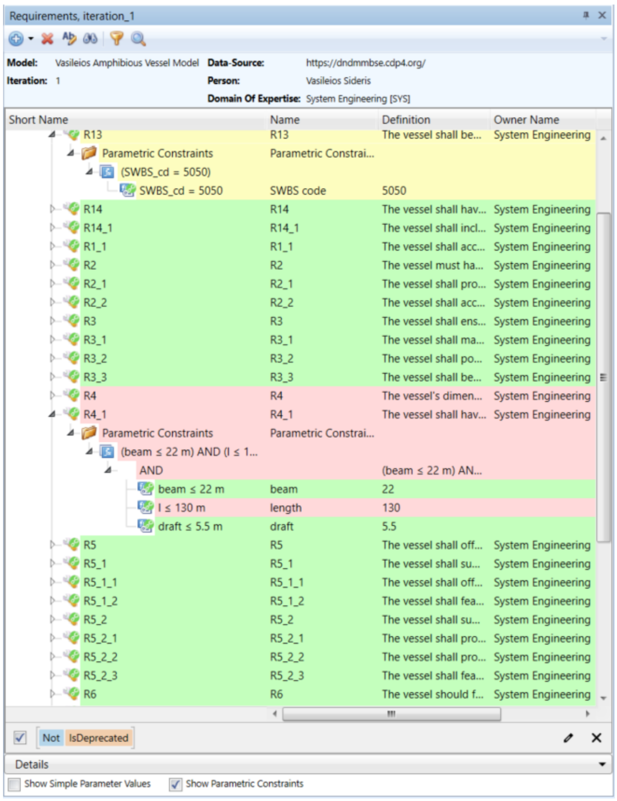
Since a large part of the CDP4-COMET model belongs to the Physical Architecture, we chose to simply add a new Equipment. This modification concerns only the physical layer, with the functional part considered out of the scope of this demo. The new equipment is called "Larger Aircraft Elevator". Its creation was achieved in an extremely straightforward manner by copying the "Smaller Aircraft Elevator" element definition block and then modifying the name and parameters of the new block.

Finally, the newly created block can be easily dragged and dropped into the designated branch of the architectural breakdown.

Next to this, to illustrate the Options principle, we have chosen to modify an existing piece of equipment from the baseline model (Option 1) in the design alternative solution (Option 2). Specifically, the Command Center underwent alterations for this purpose. To achieve this, we used the existing element definition. However, unlike the original configuration, certain parameters, such as mass and area, are now designated as option-dependent. This strategic approach avoids the creation of entirely new modeling elements from scratch with distinct characteristics. Consequently, these parameters are capable of accommodating different values for every Option, offering enhanced flexibility and adaptability within the model.

Now, the removal of existing equipment will be investigated by completely eliminating it from Option 2. To illustrate this, the modeling element "Surveillance Systems" is set to be option-dependent and is selected to be present only in Option 1.

It is reminded that these adjustments aim to theoretically meet the requirements of R6.2 and R14.1. The results of the automatic requirements verification discussed above for the final version of Option 2 are depicted in Figure 7.13.



Requirements, iteration_1

Model: Vasileios Amphibious Vessel Model
Iteration: 1
Data-Source: https://dndmmbse.cdp4.org/
Person: Vasileios Sideris
Domain Of Expertise: System Engineering [SYS]

Short Name	Name	Definition	Owner Name
R13	R13	The vessel shall be...	System Engineering
Parametric Constraints	Parametric Constrai...		
(SWBS_cd = 5050)			
SWBS_cd = 5050	SWBS code	5050	
R14	R14	The vessel shall hav...	System Engineering
R14_1	R14_1	The vessel shall incl...	System Engineering
R1_1	R1_1	The vessel shall acc...	System Engineering
R2	R2	The vessel must ha...	System Engineering
R2_1	R2_1	The vessel shall pro...	System Engineering
R2_2	R2_2	The vessel shall acc...	System Engineering
R3	R3	The vessel shall ens...	System Engineering
R3_1	R3_1	The vessel shall ma...	System Engineering
R3_2	R3_2	The vessel shall po...	System Engineering
R3_3	R3_3	The vessel shall be...	System Engineering
R4	R4	The vessel's dimen...	System Engineering
R4_1	R4_1	The vessel shall hav...	System Engineering
Parametric Constraints	Parametric Constrai...		
(beam ≤ 22 m) AND (l ≤ 1...			
AND		(beam ≤ 22 m) AN...	
beam ≤ 22 m	beam	22	
l ≤ 130 m	length	130	
draft ≤ 5.5 m	draft	5.5	
R5	R5	The vessel shall off...	System Engineering
R5_1	R5_1	The vessel shall su...	System Engineering
R5_1_1	R5_1_1	The vessel shall off...	System Engineering
R5_1_2	R5_1_2	The vessel shall fea...	System Engineering
R5_2	R5_2	The vessel shall su...	System Engineering
R5_2_1	R5_2_1	The vessel shall pro...	System Engineering
R5_2_2	R5_2_2	The vessel shall pro...	System Engineering
R5_2_3	R5_2_3	The vessel shall fea...	System Engineering
R6	R6	The vessel should f...	System Engineering

☑ Not IsDeprecated

Details

☐ Show Simple Parameter Values ☑ Show Parametric Constraints

Figure 7.13: Requirements verification (Option 2) for the LPD model in CDP4-COMET

Of course, the new elements vary in mass, which will consequently affect the total mass and the center of gravity. This change is reflected in the mass budget of Option 2, as detailed in the Appendix in Chapter 9.

Model Modifications It was discussed that adding new parameters to the model, chosen as a representative check in the Model Modifications, can be effortlessly executed in CDP4-COMET. The process involves either locating the parameter in the parameter catalogue or creating a new one. Next, the parameter can be easily added by simply dragging and dropping it to the designated element, completing the modification with minimal effort and time.

In general, modifications in CDP4-COMET are generally quick and straightforward due to its modeling logic with building blocks and the drag-and-drop technique. However, the ease of making modifications is hindered, especially in very complex projects, by the lack of an impact analysis view for proposed changes. Additionally, the absence of an undo or redo feature complicates the handling of modeling mistakes, making error correction more challenging.

- **Check model structure and tool features: Report reusable and non-reusable elements**

  Flexibility in CDP4-COMET was demonstrated with the following means:

  - Use of a cloning technique for the modifications: The “Smaller Aircraft Elevator” Element Definition, including its parameters, has been used to create the “Larger Aircraft Elevator” Element Definition.
  - Use of a templating technique for the model structure: Predefined element blocks (Element Definitions) were created, including blocks for functions, subsystems, equipment, and other components. These templates were then reused multiple times to quickly generate modeling elements within the model by copying the templates and making any necessary adjustments.
  - Reuse of reports and balances: Once a report is generated, it can be utilized multiple times within the same project or for other projects with similar structural models. In this thesis, the mass balance report from a previous project was employed both prior to and following modifications, allowing for more informed decisions at many critical points throughout the project.
  - Reuse of existing parameters: Many of the necessary parameters already exist in CDP4-COMET. The modelers only need to drag and drop these parameters into the appropriate element definitions.
  - Reuse of existing element definitions in Options: Existing element definitions can be reused within options by setting them, or some of their parameters, as option-dependent, as demonstrated earlier during the model modifications.
  - Reuse of verification and validation rules: Both the validation and verification rules for the requirements did not need modification after updating the model.

7.4. Success Factor: Trade-offs

The rationale for studying the management of trade-offs in an MBSE tool should be made clear first. According to the analysis done in Chapter 3, the feature of managing trade-offs was linked with one of the key requirements for early-stage naval vessel design, namely Adaptability to Externalities.

Following this, it should briefly detail what managing trade-offs entail in the context of the investigated tools. A trade-off represents the strategic compromise or decision-making process when selecting from multiple desirable attributes or features to achieve the best overall solution for a given problem (Kolfschoten et al. 2024).

A crucial element of MBSE’s effectiveness lies in its ability to provide insights into design trade-offs. Such trade-offs are now often an end-point decision, but there is value in understanding these trade-offs better and earlier in the design process. The inherent traceability in a systems model assists in precise and robust change assessments, alternative analysis, and trade-offs. This enables the designer to understand how a minor adjustment in one aspect can have a drastic impact on the entire system.

Means of compliance:
To ensure that the tool meets the trade-off criteria, we employ the following check:

- Comparative analysis of variants: Evaluate the capability of each tool to enhance transparency in trade-offs, allowing for a more informed comparison of design variants

7.4.1. Trade-offs in Capella

- **Comparative analysis of variants: Evaluate the capability of each tool to enhance transparency in trade-offs, allowing for a more informed comparison of design variants**

  To study if and how Capella can be used for conducting trade-off analysis, it is important to establish key parameters. These parameters might include mass, electric load, center of gravity, and cost, among others. However, visualizing such quantitative performance parameters in Capella is challenging. As previously noticed, the current model of the LPD in Capella does not incorporate any quantitative parameters. While there are useful add-ons, such as the Basic Mass Viewpoint, which allows for the visualization of an element's mass, as demonstrated in Chapter 6, the core functionality of Capella lacks direct support for such quantitative performance parameters. Thus, it can be concluded that the tool's effectiveness in managing trade-offs heavily relies on these add-ons. Since the assessment of these add-ons is considered beyond the scope of this thesis, it can be concluded that Capella performs inadequately in managing trade-offs related to quantitative performance parameters without additional extensions.

On the other hand, Capella can qualitatively support trade-off studies. In the Arcadia method, capabilities are illustrated using "scenarios", referred to as "use cases" in other methodologies. At least one scenario is needed to explain each capability while it is possible to have many scenarios to explain a single capability. Since these scenarios can be compared on different levels, the conventional method of visual inspection of diagrams in Capella can be used for tracking changes and comparing design variants.

In this thesis, three scenarios were used to conduct the Amphibious Raid mission. For each scenario, diagrams depicting the different situations were constructed. Side-by-side comparisons of these diagrams, representing the system structure for each scenario, have allowed for the study of different interpretations of the system's architecture. For instance, a side-by-side comparison of the Emergency Rescue and Air Movement scenarios at the system level, as shown in Figure 7.14, highlights the visual differences in functions between them. These variations can eventually lead to different technological solutions in subsequent layers. However, it is mentioned that side-by-side comparisons become challenging, especially as model complexity increases.

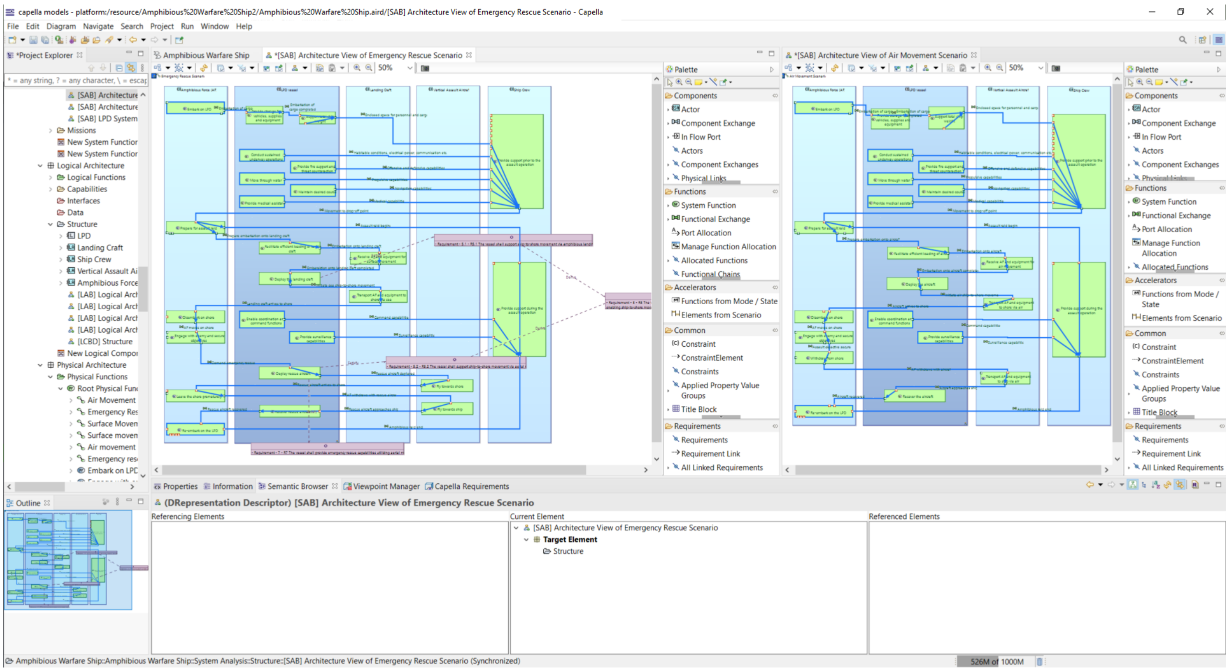


Figure 7.14: Side-by-side comparison of scenarios in System Analysis for the LPD model in Capella

Generally speaking, Capella is not well-suited for comparing variants within the model itself, indicating a need for further exploration to identify more effective methods.

7.4.2. Trade-offs in CDP4-COMET

- **Comparative analysis of variants: Evaluate the capability of each tool to enhance transparency in trade-offs, allowing for a more informed comparison of design variants**

In CDP4-COMET, the model typically includes requirements, solution architecture, and multiple key parameters for a comprehensive evaluation of overall system performance. Thus, balances, also known as budgets, are crucial for tracking changes and exploring trade-offs. It is important to remember that a budget provides an overview of the system's performance across various aspects, including mass, power, costs, data, technology readiness, and more. The reporting feature in CDP4-COMET provides users with the capability to create custom reports, allowing them to focus on specific key parameters to evaluate the overall performance of the system.

Moreover, during the exploration of different solution directions, users can create a new CDP4-COMET Product Tree using various modeling concepts. These concepts involve actions such as including or excluding element definitions and differentiating parameter values through setting option dependencies or using overrides. These Product Trees are provided in the CDP4-COMET Workbook after rebuilding, representing the subset of the engineering model that is relevant for and used by a domain. All these different design alternatives and model configurations can be documented and saved in different Options.

To practically assess the tool's capabilities in capturing and analyzing trade-offs, factors such as ease of use, time, and reusability were considered:

- In terms of ease of use, constructing a balance report is typically considered an activity for more advanced users due to the requirement of coding knowledge. However, using an existing report as a base allows for simple modifications to adapt it to models with similar structures. In this thesis, for example, a mass balance from a similar project was modified to be compatible with the LPD model (Kolfschoten et al. 2024).
- Considering the time factor, creating a mass balance from scratch would require a significant investment of time, contingent upon one's coding skills. The estimated duration to acquire fundamental knowledge of CDP4-COMET's reporting mechanism and to craft a report similar to the mass balance report is approximately one week. However, adapting an existing mass balance can be accomplished relatively quickly. In the context of this thesis, these adjustments were completed within a few hours. It is noteworthy that the contributions of the author's colleagues in performing the adjustments were crucial.
- Once the report is generated, it can be reused to provide an instant overview of the system's performance. In this thesis, calculations such as determining the total mass and center of gravity for each design option, as well as updating model parameters, were performed quickly, within a few seconds. This not only saves time but also emphasizes the aspect of reusability.

Overall, it can be concluded that CDP4-COMET excels in managing trade-offs. Its reporting capability enables the synthesis of information and facilitates comparisons between design alternatives, which can be saved and easily accessed across different options. Consequently, transparency is enhanced, leading to more informed decision-making.

7.5. Conclusions

In the forthcoming sections, a comprehensive review of the factors examined within each tool will be presented. Each tool will be evaluated based on these factors using a five-step scale, ranging from "poor" as the lowest score to "excellent" as the highest, with intermediate evaluations including "fair", "average", and "good".

Consistency in Capella

In summary, Capella supports automatic checks using built-in rules, offering over 300 integrated rules to validate models across different architectural layers. The tool identifies modeling items violating rules and categorizes violations as errors or warnings, providing quick fixes where possible. However, it restricts the creation of new rules, allowing only activation or deactivation of predefined ones. To further enhance consistency, Capella has features like the correct use of modeling elements based on the Arcadia method, automatic view synchronization, and seamless transitions between modeling layers. Consequently, it earns an overall rating of "Good", reflecting minor areas for improvement.

Consistency in CDP4-COMET

CDP4-COMET offers both built-in and customizable rules to sustain internal consistency within the model. It automatically detects violations and issues error messages accordingly. Its standout feature lies in the capability to create custom checks. However, there is room for enhancement, particularly in integrating severity levels into rules and providing clearer guidance for resolving identified errors. Additionally, consistency is maintained through views like Element Definitions and Product Tree, which allow users to inspect/ modify data and track parameters visually. The templating approach of copying element definitions ensures uniformity throughout the model. Despite these strengths, CDP4-COMET's consistency was evaluated as "Average," indicating some areas for potential improvement.

Traceability in Capella
In Capella's basic version without extensions, automatic verification of requirements is not possible. However, the Requirements Viewpoint add-on aids in requirement visualization, allowing for a visual display. Verification in the LPD case relied on linking each requirement to an element with a satisfaction relationship, limited to functional requirements and not covering arithmetic calculations and comparisons. Moreover, other traceability functionalities in the tool include functional chains for illustrating system behavior and the semantic browser for tracing parent/child relationships and enabling quick navigation. Finally, it offers transition functionalities for traceability between architectural layers, and traceability matrices for inspecting and creating traces between elements from different levels. Consequently, its overall traceability score is assessed as "Average".

Traceability in CDP4-COMET
CDP4-COMET stands out for its strong automatic requirements verification, strengthened by the Microsoft Excel integration. It efficiently verifies various requirements, from basic functional to intricate quantitative design requirements. Nevertheless, manual creation and direct linking of new parameters solely for verification, together with establishing parametric constraints for each requirement, can be time-consuming and labor-intensive. Next to that, CDP4-COMET enables traceability through defined binary relationships and supports monitoring and logging changes throughout the design, guaranteeing transparent documentation and overview of model modifications over time. Consequently, it earns an overall rating of "Good".

Flexibility in Capella
Capella streamlines the visual addition, removal, or modification of elements. It also helps make these changes consistent across the model and provides automatic diagram updates. Nevertheless, modifying parameters is expected to be more challenging and time-consuming, although this aspect has not been extensively investigated, resulting in a neutral assessment. Despite this, the tool offers smart shortcuts and a clear overview of impacted elements, enhancing guidance and flexibility. Moreover, the automatic transition feature encourages reusability by transitioning and reusing elements from previous layers, while the diagram cloning feature significantly reduces workload. Therefore, it earns an overall rating of "Excellent".

Flexibility in CDP4-COMET
The modeling logic of building blocks and the drag-and-drop technique allow for quick and straightforward modifications in CDP4-COMET. Nonetheless, the lack of an impact analysis view for proposed changes and the absence of undo or redo features complicate error handling and hinder ease in making modifications, particularly in complex projects. On top of that, functionalities like cloning, templating, reusing reports, existing parameters, and verification/validation rules enhance overall flexibility. Consequently, its overall flexibility score is assessed as "Good".

Trade-offs in Capella
Capella's ability to enhance transparency in trade-offs is hindered by the lack of direct support for quantitative performance parameters without additional extensions. While qualitative trade-off studies are supported through scenarios and visual inspection of diagrams, side-by-side comparisons become challenging with increasing model complexity. Therefore, Capella's inadequacy in comparing design variants is reflected in its assessment, rated as "Fair".

Flexibility in CDP4-COMET
CDP4-COMET comprises requirements, solution architecture, and key parameters for system performance evaluation. Balances and reporting mechanisms are crucial for tracking changes and studying trade-offs. The tool allows users to create custom reports focusing on specific parameters, while Product Trees facilitate the exploration of solution directions, documented and saved as options. Although creating reports from scratch demands time and coding skills, adapting existing ones can be done relatively quickly. Once constructed, reports can be reused repeatedly, offering instant updates and feeding the model with computed values for the performance parameters. This exceptional capability of CDP4-COMET to provide transparency in trade-offs is reflected in its assessment, rated as "Excellent".

		Capella features			CDP4-COMET features		
Factor	**Means of compliance**	**Positive aspects**	**Negative aspects**	**Overall assessment**	**Positive aspects**	**Negative aspects**	**Overall assessment**
Consistency	Automated checks using built-in/custom validation rules	More than 300 incorporated rules Identification of modeling items violating rules Rule alongside for better understanding Automatic solutions via ”Quick Fix” option Severity indication streamlines corrective actions	Customization limited to rule activation/deactivation Automatic transition malfunctions	Good	Custom validation rules allowing tailored checks Detection of violations and error message Possibility for more advanced checks	No severity levels for rules Lack of predefined validation rules Complex error message interpretation	Average
	Investigation of other consistency functionalities	”Correct” use of modeling elements (Automatic) Transitions between layers Automatic synchronisation of different views	-		Element Definitions and Product Tree views Color-coded parameters and other visuals Copying element definitions (templating technique)	-	
Traceability	Evaluation of requirements verification capabilities	Graphical requirement representation (add-on) Requirement Links Mass Visualization View for editing or viewing	Automatic verification not possible in basic version Visual verification lacks arithmetic comparison capabilities	Average	Automatic verification of requirements Color-coded indicators for visual clarity	Creation of new verification parameters Time-consuming parametric constraint establishment	Good
	Investigation of other traceability functionalities	Functional chains Semantic Browser Transition functionalities Traceability Matrices	-		Binary relationship rules and Traceability Matrices Monitoring and logging changes	-	
Flexibility	Analysis of modification process	Very easy to add/remove/modify elements visually Consistent changes in model and diagram updates Well-guided process and smart shortcuts Impact analysis window before deletion	Adjusting parameters is expected to be challenging	Excellent	Very easy to add elements by drag-and-drop Options feature simplifies modifications	Difficult to analyze the impact of a modification No undo or redo feature included	Good
	Evaluation of reusability and extensibility options	No need for new elements; reuse existing ones Transitioned elements easily available for reuse Cloning diagrams reduces workload	-		Cloning technique for modifications Templating technique for creating new elements Reuse of balances for similar models Reuse of existing parameters Reuse of verification and validation rules	-	
Trade-offs	Comparative analysis of design variants	Scenario-based comparison Qualitative analysis by visual inspection	Quantitative analysis is difficult Visual comparison is difficult for complex models	Fair	Custom reports for focused performance evaluation Options for solution direction exploration Reuse reports across multiple projects Instant updates for quick performance overview	Time and coding skills required for new reports	Excellent

Table 7.1: Assessment of Capella and CDP4-COMET in terms of consistency, traceability, flexibility and trade-offs

8 Conclusions and Recommendations

This chapter concludes the graduation research assignment by first revisiting the research gap and augmenting it with additional insights garnered throughout this thesis. Next, the methodology framework, its implementation, and the key findings and limitations for each project phase are discussed. The chapter then proceeds with a comprehensive analysis of the comparative study conducted among the selected MBSE tools, followed by presenting findings in response to the Research Questions in a structured manner. Furthermore, synthesizing answers from previous questions and lessons extracted from this endeavor aims to address Research Question 7 (*"To what extent can MBSE become a standard working method for future naval vessel design?"*). Finally, recommendations for future research are outlined, paving the way for advancing MBSE in early-stage warship design.

8.1. General Discussion

MBSE Benefits and Adoption Challenges in Early-Stage Warship Design

The transition from document-centric to model-centric approaches is paramount for addressing inconsistencies and enhancing competitiveness in complex system design. This holds particular importance in the context of early-stage warship design. Enhanced communication, continuous verification and validation, rigorous traceability, improved decision-making, and design consistency are some of the expected benefits of employing MBSE. However, despite its potential advantages, MBSE adoption in the maritime industry remains limited, with significant benefits yet to be fully realized by the community. The hindrance to widespread MBSE adoption can be attributed to the following challenges identified in the literature and from the author's overall experience with industry experts and participation in similar projects:

- One significant limitation in the adoption of MBSE within the maritime industry is the lack of substantial evidence demonstrating its benefits. The full potential of MBSE has not been realized. This is one of the main reasons why it has not been embraced on a larger scale in ship design. Pilot projects aim to assess its effectiveness and lay the ground for broader adoption in the coming years. Furthermore, the existing benefits of MBSE in ESSD for naval ships are primarily theoretical, with limited research focusing mainly on functional and operational architectures. All of that confirms the continued need for additional research in this area.
- Another challenge identified is the confusion surrounding MBSE practices. Engineers may assume they are implementing MBSE, yet, in actuality, they are often involved in regular modeling activities and/or mentally following SE steps. Thus, highlighting the theoretical benefits of MBSE is not enough; there is also an important need for further research on the proper implementation and execution of MBSE techniques. Moreover, the literature review has not revealed any research specifically exploring the value and suitability of MBSE for architecting systems of a modern warship. This is a critical aspect requiring attention, given that one of the major obstacles in MBSE lies in the inherent limitations of the tools, methods, and languages themselves. As emphasized, no tool is flawless for every application, and thus selecting the appropriate tool becomes an important aspect. These facts highlight the potential benefits of systematically comparing basic MBSE tools to help engineers select the best options during early design phases. The absence of such analysis is a missed opportunity to provide practical insights, reduce confusion around MBSE practices, and promote effective implementation.

Methodology

The methodology outlined in Chapter 4 provided a systematic framework, consisting of 10 steps, for addressing the research objective. We began by defining the mission of the SoI (Step 1). For this thesis, this was an Amphibious Raid, and the SoI was an Amphibious Vessel, specifically an LPD. This drove us to define the mission capabilities of the naval vessels under study (Step 2), and thus the requirements to consider for modeling this system (Step 3). This laid the foundation for our subsequent modeling efforts. The selection of appropriate MBSE tools (Step 4) and the formulation of a metamodel (Step 5) guided us to consistently construct MBSE models that captured the system architecture at various levels of abstraction (Step 6). Through the iterative process of verification, validation (Step 7), and modifications (Step 8) of the models, we ensured their correctness, completeness, accuracy, and relevance to the research objectives, while also simulating the ever-evolving nature of warship design. Finally, we conducted the collection and analysis of modeling constructs (Step 9) and evaluated the tools based on key MBSE factors, including consistency, traceability, flexibility, and trade-offs (Step 10).

It can be concluded that the described process adequately meets the success criteria established in Section 4.1.

- It addresses naval challenges in the early design phase, encompassing critical aspects like mission definition, capability analysis, requirement exploitation, and systematic system development from stakeholder needs to a tangible physical concept.
- It prioritizes establishing direct links between requirements and architectural artifacts, ensuring alignment with mission objectives and user needs by tracing requirements through system architecture levels
- It centers on the practical application of MBSE, constructing models to showcase its value, and subsequently assesses tool capabilities in improving consistency, flexibility, traceability, and trade-offs.
- It ensures simplicity and accessibility through clear step-by-step guidance. Nevertheless, achieving full implementation, particularly in the context of warship complexities, may require some level of technical expertise and familiarity with MBSE tools.
- The methodology is tool-independent, allowing for implementation with various MBSE tools, and adaptable to diverse systems within the early design phases
- It acknowledges the importance of iteration, especially in the development of system architecture. It emphasizes continuous iteration at both the level of individual process elements and across consecutive steps.

We acknowledge several limitations in the process, but in our discussion, we have chosen to focus solely on those stemming from utilizing the methodology concurrently across multiple tools, as observed in the explored case study.

- When multiple tools are used for comparative analysis, the methodology lacks clear assessment criteria to precisely rate the capabilities of each tool. It offers only generalized means of compliance that are binary in nature—either met or not met. Consequently, the rating process becomes subjective, lacking qualitative assessment standards.
- There is a need for stringent validation criteria to ensure that the comparative models accurately showcase the full capabilities of the tools. This involves assessing whether the models are constructed in a manner that does not inadvertently limit the tools' capabilities based on the explored aspects or the modelers' experience. In essence, the quality of the model directly impacts the perceived effectiveness of the tool.
- The sequence in which the models were constructed in each tool could significantly influence the overall outcome of subsequent models. Although the Metamodel was there to guide the modeling effort, it did not ensure immunity to potential biases from the capabilities of the initial tool. In the author's experience, the initial selection of Capella for architecture introduced bias into the subsequent modeling process in the second tool (CDP4-COMET). Decisions made in Capella strongly influenced the structure of the second model, leading to replication. This limited exploration of alternative system configurations

Case Study and Modeling Regarding the case study selection, the scope was determined to ensure it can effectively demonstrate the key benefits of MBSE in a relatively straightforward manner. However, developing an overly simplified model risks failing to simulate the challenges and complexities inherent in large and complex systems, such as a naval ship.

Moreover, the decision to study the ship as a whole, rather than focusing on a specific system within it, was made consciously. This approach is believed to better represent the challenges encountered during the early stages of ship design. To simulate a realistic problem, and given the Dutch Navy's plans to replace its Landing Platform Docks (LPDs) with a new class of amphibious transport ships, this case study focuses on the design considerations of a hypothetical LPD vessel (News 2024).

To ensure practicality in real-world scenarios, the system definition, architecture, and design should be driven by mission-based operational needs. Consequently, an Amphibious Raid mission was selected for this purpose, guiding the capability breakdown for the LPD vessel. A comprehensive list of requirements was also compiled to drive the design process and to evaluate the modeling tool's capabilities in verifying these requirements and managing potential conflicts.

Some key limitations can be identified in the aforementioned activities:

- A naval vessel, particularly an LPD, is one of the most complex human-made systems with a broad range of CONOPs. Thus, a limitation of this thesis is the narrow focus on a single mission area, which may not fully represent the LPD's capabilities and complexities in real-world scenarios. Despite efforts to encompass the entire vessel in the modeling process, the final model may still represent only a fraction of the complete system's capabilities and complexity.
- The simplified requirements list used to illustrate MBSE application, while suitable for demonstrating MBSE, may lack the depth and granularity required for comprehensive ship design. Both quantitative and qualitative requirements were included, yet essential types of requirements could have been overlooked. For example, no requirement regarding subsystem interactions was included. Additionally, assuming a one-to-one direct mapping between requirements and components oversimplifies the design process. Such oversights may lead to a false sense of confidence in the tools' capabilities.

The Metamodel played a crucial role in establishing a shared understanding of design challenges across various disciplines. While the author may be the sole participant in modeling activities, the Metamodel served as a foundational framework for guiding these activities. In this thesis, it ensured a common ground for modeling across different tools, thereby supporting consistency and coherence.

Regarding the modeling activities, the development of the system architecture has been documented extensively to provide readers with an opportunity to explore MBSE in practice and understand the capabilities of the selected tools. It is worth noting that the utilization of CDP4-COMET in this project diverges from its conventional approach, aiming to facilitate a more subjective comparison with the Arcadia methodology. This adjustment in the modeling process, particularly in resembling the Arcadia methodology, enhances the consistency between the two models, thereby enabling a more subjective comparison. However, it is essential to acknowledge that this departure from the conventional usage of CDP4-COMET carries the inherent risk of underutilizing the tool's full potential or not fully exploring its primary strengths. Despite this non-conventional approach, considerable effort was devoted to ensuring that the tool's main strengths were effectively demonstrated. This approach aimed to ensure that the resulting models serve as comprehensive representations of MBSE tools in general, showcasing their capabilities even when used in non-traditional ways.

To maintain alignment with the thesis timeline, several simplifications and shortcuts were implemented during the modeling effort, leading to certain limitations. Some of them are presented below.

- Limitations in Capella
  - The physical layer omits the three scenarios for simplification purposes.
  - Exchanges between logical and physical modeling elements are only modeled in a few cases, leading to missing traceability links.
  - Parameterization is omitted due to the perceived imbalance between time investment and benefits.
  - Requirements verification was conducted manually and visually, leading to incomplete coverage.
  - Sequence Diagrams are considered only in the initial architectural layers.

  - Modes and States diagrams are not explored at all.
- Limitations in CDP4-COMET
  - The three scenarios are not modeled at all, despite the possibility of using the Options functionality for that purpose.
  - The CDP4-COMET modeling process is aligned with Arcadia methodology for consistent comparison, deviating from its conventional approach.
  - Key collaborative features of CDP4-COMET, such as Ownership, Parameter Subscriptions, Domains of Expertise, Roles, and Reporting, are only partially explored due to the single-person participation in the project.
  - Other essential functionalities like Model Iterations, Constraints, Finite States, Grapher, Relationship Editing, Export and Import of Requirements, and Scripting capabilities are not explored at all.

8.2. Concluding the Capella vs CDP4-COMET Comparison

One of the major lessons of this thesis is that every tool is unique, thus the selection must be made according to the project's needs. The early design exploration of an LPD vessel in both Capella and CDP4-COMET across operational, functional, logical, and physical layers provided valuable insights into their strengths and weaknesses.

The operational and functional analysis capabilities of Capella make it an ideal tool for capturing stakeholders, understanding their intentions, and defining the system to meet their needs. Consequently, it proves invaluable for early-stage organization and delineation of the system's ecosystem. Capella's strength also lies in its ability to seamlessly transition to the logical layer, allowing for a comprehensive exploration of potential solutions.

However, as the level of detail increases and the focus shifts towards the practical implementation of the system, Capella's effectiveness diminishes. The physical architecture is where CDP4-COMET excels. CDP4-COMET shines in the later stages of design, leveraging a wealth of data as input. It allows for integrating a vast array of parameters into the design process. Moreover, CDP4-COMET offers a direct linkage of requirements to design elements and parameters, enabling automated verification of requirements. Its robust reporting and balancing capabilities are particularly noteworthy, allowing for quick assessment of design variants and, consequently, effective management of trade-offs.

Finally, substantial differences can be identified regarding design cycle management. In CDP4-COMET, roles are precisely defined, and mechanisms like gates for data flow control, baselining, and fine-grained Role-Based Access Control are implemented. Key features, such as the Parameter Subscription Switch and Publications, enable domain experts to control input values and regulate parameter propagation to the team. Iterations facilitate baselining by freezing previous iterations. In contrast, the standard version of Capella does not support collaborative efforts. The only relevant add-on identified by the author was the "XHTML Documentation Generation" add-on, enabling the sharing of full models among stakeholders without requiring the software to be installed, albeit without allowing modifications. Capella does offer commercial add-ons like "Teams for Capella" for collaborative work, though this aspect was not further explored in this project.

It becomes evident that a potential collaboration between both tools, leveraging each for its respective strengths, will lead to superior designs. This underscores the critical necessity of seamlessly integrating various tools. The comparison between Capella and CDP4-COMET highlights the specific areas where each tool excels, as depicted in Figure 8.1.

8.3. Answers to Research Questions

In this thesis, the aforementioned existing gaps were addressed by exploring the value of MBSE and enhancing the maturity of MBSE tools, methodologies, and languages, specifically in the context of early-stage naval design. This was achieved by validating a set of expected benefits critical to early-stage warship design. To demonstrate MBSE's efficacy, the thesis specifically concentrated on crafting systems architecture, encompassing operational, functional, logical, and physical perspectives. By focusing on a

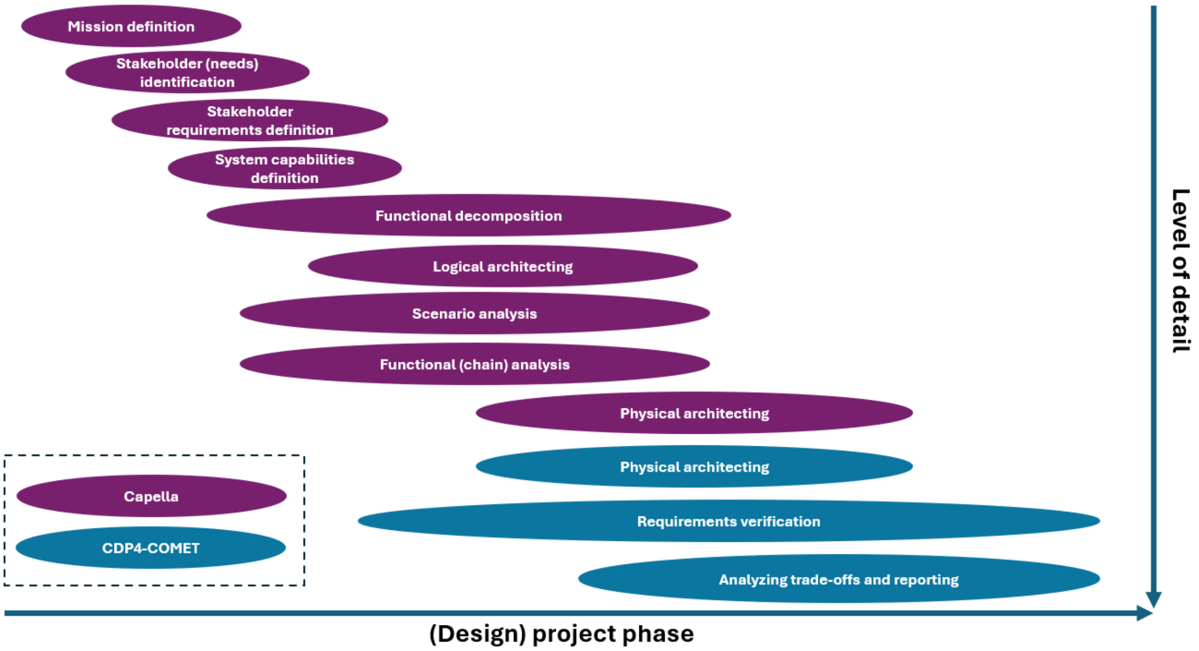

Mission definition
Stakeholder (needs) identification
Stakeholder requirements definition
System capabilities definition
Functional decomposition
Logical architecting
Scenario analysis
Functional (chain) analysis
Physical architecting
Physical architecting
Requirements verification
Analyzing trade-offs and reporting
Capella
CDP4-COMET
Level of detail
(Design) project phase

Figure 8.1: Visualizing Tool Proposals Across Project Phases and Detail Levels (Adapted from (Kolfschoten et al. 2024), and modified)

specific example system, a simplified Landing Platform Dock vessel, the study leveraged modeling expertise from Starion and industry approaches. The comparative analysis of the two MBSE tools, each with its method and language, deepened the understanding of MBSE's value, provided insights into MBSE practices, and aided in informed tool selection, all from a naval architect's perspective. This objective was reflected in the establishment of the following central research question:

How can the value of MBSE, specifically its benefits, be effectively demonstrated and validated in the development of systems architecture during the early-stage naval vessel design?

To systematically address this question, it has been broken down into seven sub-questions, referred to as Research Questions. These questions and their answers are provided below.

Research Question 1

Why are the early design stage and the development of well-defined system architecture critical in naval design and what are the limitations of the contemporary design methods/approaches?

This question was answered in n Chapter 2, in which the fundamental principles of ship design relevant to the thesis were outlined. After distinguishing several design phases in Section 2.1, Section 2.2 focused on ESSD, concluding that it represented an initial phase during vessel development where vital decisions are made with limited knowledge. When considering naval vessels specifically, Section 2.4 delved into the uniqueness of early-stage warship design, characterized by complex decision-making, interdependencies among requirements, and sensitivity to external influences. These challenges, including trade-offs, limited experience, high acquisition costs, and evolving requirements, emphasized the critical role of ESSD in naval design, as it involves making significant decisions that heavily impact the vessel's performance and overall lifecycle costs.

Section 2.3 discussed the importance of developing a well-defined system architecture during ESSD as an effective way to deal with the complexity of warships. It was discovered that the development of a system architecture facilitates better decision-making, enhanced requirements traceability, and reduced costs in complex vessel design by providing structure to stable elements while allowing flexibility for those prone to change. Considering these challenges, four key requirements for typical early-stage design methods were determined.

The four key requirements were used in Section 2.5 to evaluate contemporary design methods/approaches like the Ship Design Spiral, System Based Design, Concurrent Design, Set Based Design, and MBSE. Despite their advantages, limitations such as lack of adaptability, restricted innovation, and challenges in managing evolving requirements and design relationships were identified. The assessment, summarized in Table 2.1, suggested MBSE as superior due to its ability to address these limitations effectively.

Research Question 2

What is MBSE, and how can it theoretically aid in managing the increased complexity of early-stage naval ship design?

This question was fully addressed in Chapter 3. To improve the understanding of what MBSE is, Section 3.1 explored the three intertwined concepts it entails, namely Systems Thinking, Model, and Systems Engineering. From Section 3.2, it can be concluded that MBSE is a modern approach to Systems Engineering that places models at the center of the design 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. To complete the analysis of what MBSE is, its three main pillars were examined: modeling language, modeling method, and modeling tool.

Section 3.2 also outlined that MBSE theoretically offers numerous benefits, including a single source of truth, a well-structured system architecture, improved decision-making, enhanced design quality, better collaboration and communication, reduced development risk, and the scalability and reusability of modeling constructs. To streamline the analysis, a set of factors forming the foundation of these benefits, referred to as MBSE factors, has been identified These were consistency, traceability, flexibility, and trade-offs. By revealing the correlations between these MBSE factors and the four identified requirements for design methods in the early stages of warship design (Table 3.1), it was suggested that MBSE can theoretically aid in managing the complexity of early-stage naval ship design.

Research Question 3

What are some commonly used combinations of MBSE methods, tools, and languages, and what are their strengths and weaknesses according to the literature?

This question was covered in Chapter 3. To address this, a detailed review of some well-known modeling methods, languages, and tools has been explored in Section 3.3. Among those are CDP4-COMET (tool) and Capella (tool), as well as SysML tools like CORE, Cameo EA, Rhapsody, etc. This analysis served to acquaint the reader with industry applications of the three pillars of MBSE and made the value of MBSE and its implementation more evident. A comprehensive summary of the strengths and weaknesses of each one was presented in Table 3.3. It can be concluded that there is no universally perfect combination. Each combination has its advantages and disadvantages, making it more suitable for certain applications. Therefore, selecting the three pillars of MBSE should align closely with the project's specific needs.

Research Question 4

What critical factors play a key role in selecting a representative case study, considering a) the selected system of interest, b) the level of detail, and c) the chosen modeling tool-method-language?

This question is addressed in Chapters 5 and 6.

In Chapter 5, the rationale behind selecting the system of interest and defining its level of detail was outlined. Section 5.1 introduced the scope of the case study, highlighting the choice of a Landing Platform

Dock vessel as the system of interest to demonstrate the key benefits of MBSE within a manageable complexity. This choice parallels the concept exploration phase of ESSD, focusing on comprehensive ship investigation at a lower level of detail akin to early ship design stages. While recognizing its integration within a larger fleet system, focusing solely on the vessel was deemed sufficient for achieving the project's objectives. This approach enabled a thorough examination of interfaces with relevant systems and users while excluding non-essential fleet vessel interfaces.

The subsequent Sections 5.2, 5.3, and 5.4 explored the specific mission focus, the mission-related capabilities, and the detailed vessel requirements, respectively, thus forming the backbone of the example problem. The level of detail in this case study intentionally focuses on essential operational aspects of the LPD vessel during an amphibious raid. Requirements were crafted to streamline detail in essential subsystems and components, ensuring a clear focus on operational objectives across different mission phases. Moreover, the Metamodel (Section 6.2) also played a crucial role in scoping, decision-making, and determining the level of detail for the modeling problem. Serving as a simplified model with elements for simulating the current design challenge, it established tool-independent modeling guidelines, clarified the examined modeling rules and conventions, and defined the scope of the investigation.

The rationale for tooling selection was elaborated in Chapter 6 and specifically in Section 6.1. Building upon the findings of the literature comparison of contemporary MBSE tools, methods, and languages in Section 3.3, the tools implemented in this thesis, along with their respective languages and methods, were carefully selected. This research leveraged two MBSE tools:

- CDP4-COMET was primarily selected due to the author's collaboration with Starion, the developers of the tool, granting direct access to experts for guidance and support. This choice was also influenced by its user-friendly interface, reusability, and robust capabilities for comparing alternatives. Moreover, the alignment with the CD method, tailored to the complexities of ship design, also played a key role in its selection.
- Capella was selected due to its alignment with the Arcadia methodology, which closely resonated with the systems architecture defined in this thesis. This alignment not only provided guidance but also compensated for the author's limited experience in MBSE, ensuring the timely completion of the study. Furthermore, Capella's proficiency in functional modeling and open-source nature added to its appeal.

Research Question 5

How can the selected MBSE tool-method-language be effectively applied to model the architecture of the chosen system, and what critical insights can be gained from this modeling process?

This question was addressed in Chapter 6, in which the modeling effort unfolded. The Sections 6.3 and 6.4 provided comprehensive summaries of the modeling efforts in Capella and CDP4-COMET, respectively. These sections covered the development of the vessel's system architecture, including the operational, functional, logical, and physical layers, and depicted the main modeling constructs. It was important to note that in both processes the Metamodel was guiding the process dictating the needed modeling elements for each architectural layer, thus ensuring consistency among the developed models in both tools. Once the baseline models were constructed, they underwent validation to ensure compliance with both the language constructs and the Metamodel, and verification to confirm that the developed system met its requirements. Additionally, the versatility of the MBSE environment was tested by simulating evolving requirements during the naval vessel design through simple modifications. After each modification, the resulting design was re-verified to ensure ongoing compliance with the requirements. The modeling effort concluded with the collection and export of the modeling constructs.

The modeling insights gained for both Capella and CDP4-COMET models were presented in Subsections 6.3.10 and 6.4.11 respectively. Below are some key takeaways for each tool:

- General Insights and Comments on the Capella Modeling Effort:
  - All Metamodel elements were successfully modeled, but parameter population was omitted due to manual challenges and limited utility in Capella

  - Effective parameter utilization in Capella required integration with other tools, which was outside the thesis scope
  - The “Basic Mass Viewpoint” add-on was useful for early design stages, providing weight visualization for each element
  - The OAB, SAB, LAB, and PAB diagrams were fundamental parts of the system architecture, each with a different purpose and level of detail
  - Secondary diagrams were crucial for the conceptual model’s completeness
  - Despite the simplicity of the vessel, the diagrams were large due to the detailed information they present
  - Maintaining traceability among elements, diagrams, and architectural layers was easy
  - Verification of requirements was performed visually as automation was not feasible
  - Consistent modeling logic enhanced modeling speed and allowed quicker familiarity with the tool and methodology
  - Capella’s exceptional functional modeling capabilities make it well-suited for identifying problems, establishing functional structures, and searching for solution principles during the very early stages of ship design

- General Insights and Comments on the CDP4-COMET Modeling Effort:
  - CDP4-COMET’s Requirement Manager efficiently verified design requirements, aiding in identifying conflicts
  - Requirements could be automatically verified based on balance calculations ensuring compliance and allowing instant checks with each modification
  - Flexibility in model modification allowed easy adaptation to evolving requirements
  - Custom reports, like the mass balance report, provided insights into changes’ impact on ship performance
  - CDP4-COMET supported collaboration among multiple disciplines, enhancing interaction and ownership
  - Although the LPD model only involved the author in modeling actions, it was structured on solid foundations with involvement from multiple disciplines
  - CDP4-COMET captured interactions between disciplines, boosting collaboration and promoting responsibility
  - CDP4-COMET’s robust development of concept variants, preliminary solution layouts, refinement, and evaluation of technical and economic criteria make it more suitable for the advanced stages of ESSD

Research Question 6

How does the selected MBSE tool-method-language, perform in terms of consistency, traceability, flexibility, and trade-off analysis within the context of system architecture development?

This question was addressed in Chapter 7, where the selected tools underwent evaluation based on MBSE factors; consistency, traceability, flexibility, and trade-offs. The assessment process was systematically documented, including the means of compliance for checking each factor. Detailed evaluations of the tools were conducted in Sections 7.1, 7.2, 7.3 and 7.4 incorporating practical demonstrations, observations, and tests to validate the factors. The comparative analysis results were thoroughly analyzed in Section 7.5. Each tool received an evaluation on a five-step scale ranging from “poor” to “excellent”, with the summarized results presented in Table 7.1.

- Consistency:
  Capella demonstrated slightly stronger consistency, earning a “Good” rating, in contrast to the “Average” rating received by the CDP4-COMET, suggesting there is room for improvement in both.

- Traceability:
  Capella's traceability performance was deemed inferior to CDP4-COMET, resulting in an overall "Average" rating compared to CDP4-COMET's "Good" rating, suggesting the potential for enhancement in both tools.
- Flexibility:
  Flexibility emerged as a significant strength of Capella, reflected in its "Excellent" rating, surpassing the notable but not flawless assessment of CDP4-COMET, which received an overall score of "Good"
- Trade-offs:
  Despite Capella's management of trade-offs being deemed subpar, its impact was still evident, resulting in a relatively low rating of "Fair" compared to the flawless performance of CDP4-COMET, which earned a "Perfect" rating in managing trade-offs.

Research Question 7

To what extent can MBSE become a standard working method for future naval vessel design?

The insights gained from addressing the preceding questions can now be synthesized to tackle this question.

Several factors contribute to the potential adoption of MBSE in naval vessel design. It is believed that MBSE can accelerate the design process, reduce errors, facilitate the early involvement of industry players, and ensure better alignment between design and requirements. In this thesis, the considerations for the early-stage design included better decision management, adaptability to externalities, novelty and innovation, and the development of a well-structured system architecture. To ensure MBSE has the potential to handle these aspects, it was deemed sufficient to validate the four underlying factors: consistency, traceability, flexibility, and trade-offs.

The assessment indicated that all four anticipated benefits had been effectively validated for both investigated tools. Each tool exhibits distinct characteristics, which account for variations in the extent to which these benefits are fulfilled. Regarding the Capella tool, it proved to be valuable, with good to excellent capabilities in ensuring consistency, traceability, and flexibility, respectively. However, its management of trade-offs was deemed subpar, resulting in a relatively low rating. Nonetheless, this still validates the benefit to some extent. Regarding the CDP4-COMET tool, although there is potential for improvement in terms of consistency, the tool has demonstrated commendable performance in realizing the expected benefits of traceability and flexibility. Moreover, its capability to manage trade-offs stands out as particularly robust.

Thus, the assessment of both Capella and CDP4-COMET demonstrated that MBSE tools, in general, can validate the four anticipated benefits. While each tool exhibits unique characteristics and varying degrees of fulfillment for these benefits, the overall findings suggest that MBSE tools can guarantee these fundamental aspects. This overarching conclusion is bolstered by the recognition that both Capella and CDP4-COMET represent prominent and proven MBSE tools within the market. What further supports this assertion is the intensive effort dedicated to ensuring that the constructed models effectively showcase the full capabilities of these tools. Hence, the modeling process was accorded significant attention throughout this thesis, emerging as one of its central components. The author devoted over four months to mastering the usage of these tools and acquiring substantial experience to ensure the models were constructed with sufficient precision. Regular model reviews were conducted with experienced modelers to ensure adherence to standards. These efforts aimed to enhance model quality, recognizing its potential to directly influence the perceived effectiveness of the tools and, by extension, the field of MBSE as a whole.

Therefore, this graduation research has demonstrated that, through monitoring efforts and results against predefined success criteria, MBSE has the capacity to enhance and accelerate the design process during the early design phase of warships. In conclusion, MBSE has the potential to become a standard working method for future naval vessel design.

However, this partially addresses the question. To delve deeper into the extent to which MBSE can become a standard working method for future naval vessel design, it is crucial to consider a spectrum of factors. This includes examining current MBSE capabilities and achievements, which are pivotal in bridging theoretical considerations with practical insights gained from the author's thesis.

Current MBSE Capabilities and Insights

- **Technological Factors**
  - **Establishing Singular Truth and Traceability through MBSE Tool Integration:** Each MBSE tool has unique characteristics that make it suitable for certain scenarios and less appropriate for others. Since a perfect tool has not yet been developed, integrating several tools to leverage their combined strengths is key. The interconnection of models will provide a level of back-and-forth synchronization functionality in all directions, establishing a single source of truth and fostering traceability. This capability was demonstrated in a pilot project, in which the author participated. Adapters facilitated synchronization, albeit with limited automation that future advancements aim to enhance. Among the tools evaluated—Capella, Cameo, and CDP4-COMET—the latter proved to be the most effective as a central MBSE data hub.
  - **Collaborative Tool Integration**: Achieving a single model that spans the entire lifecycle is likely unfeasible. Rather, our focus should be on developing a central data information model that integrates multiple interconnected models. This approach enables collaboration among stakeholders using a variety of toolsets. Both Capella and CDP4-COMET possess the capability to integrate with various scientific tools across configuration, programming, mathematical computation, thermal analysis, mission analysis, aerothermodynamics, and more.
- **Cost-Benefit Analysis**
  - **Investments vs Benefits:** In contrast to a document-based SE approach, MBSE demands a higher level of rigor and completeness, preventing the team from taking shortcuts. However, once the baseline model is complete, changes and adjustments can be implemented faster and verified for every change. By leveraging reusable components and modeling structures from prior projects, alongside enforcing rigorous requirements definition and automated verification, MBSE significantly reduces manual efforts and errors. This thesis has delved into the automatic verification capabilities and identified reusable elements in Capella and CDP4-COMET. While the scalability of the model has not been thoroughly investigated, the suggested modifications indicate that it aligns with MBSE capabilities.
- **Organizational Factors**
  - **Organizational Readiness:** Effective data management and collaboration are crucial for MBSE implementation in warship design. A structured approach ensures stakeholders can work together towards project goals, with regular updates and model synchronization reducing errors and misalignments. Organizational roles, particularly in data ownership, must be clearly defined. For simple systems, like the one in this thesis, a single individual may manage data ownership. On the contrary, complex systems require a team under a Project Manager. Tools like CDP4-COMET provide principles such as ownership and publication to enable stakeholders to take responsibility and maintain the integrity of the data, ensuring that all changes are tracked, verified, and approved according to protocols.
- **Educational and Training Factors**
  - **Training Programs and Pilot Initiatives:** MBSE necessitates significant investment in training. An increasing number of projects in the industry are focused on elevating the maturity level of applying MBSE practices and exploring their feasibility for future vessels. By investing in training initiatives, team members are equipped with the requisite skills and knowledge. The author has identified several educational initiatives to support this endeavor. For example, Dutch Naval Design and the Dutch Ministry of Defense are actively prioritizing MBSE methodologies through pilot projects aimed at providing training and practical insights.

In conclusion, while it is challenging to predict the exact extent to which MBSE will become a standard working method for future naval vessel design, the evidence suggests that the tools studied—representative examples of MBSE tools—are well-positioned to address or accommodate the requirements across various factors discussed. Therefore, it is reasonable to anticipate a significant transition towards widespread adoption of MBSE in naval vessel design, marking a transformative shift in the industry's approach to system engineering and design.

8.4. Future Directions for Advancing MBSE in Ship Design

Lessons from this thesis can inform and guide future projects. While this research demonstrated that MBSE tools meet the requirements for early-stage naval design and can enhance the design process of future warships, it does not necessarily imply that MBSE will become the standard working method. Future research must investigate additional parameters to gain deeper insights into the value of MBSE and promote its establishment as a standard in early-stage ship design. These include:

- **Preparatory Steps and Advanced Ship-Wide Metamodels:**
  Future research should delve deeper into the preparatory steps preceding the use of MBSE methods and tools, which are essential for establishing a common understanding among stakeholders. Specifically, there should be a focus on aligning modeling rules and conventions, including determining optimal abstraction levels and selecting or building appropriate metamodels and design patterns for effective design. The metamodel developed within this thesis plays a pivotal role in ensuring successful MBSE implementation. Despite requiring significant initial time investment, it has proven critical in scoping and decision-making during the design process. Future research should explore further extensions or the creation of more sophisticated, ship-wide metamodels for use in future naval vessel projects.
- **Evaluating the Costs and Benefits of MBSE Implementation in Ship Design:**
  While this thesis mainly focused on the benefits of MBSE, it is equally important to delve into the associated costs. Upfront investments are necessary to reap the long-term benefits of MBSE. Further study is necessary to ascertain the resources required for implementing MBSE in the ship design industry, encompassing human resources, time commitments, infrastructure needs, and software requirements. Furthermore, it is essential to investigate the requirements of the initial training process to document its intensity and the level of mental and physical effort it demands. For example, employing questionnaires to assess end-users' perceived usefulness and ease of use can facilitate this investigation. Thorough documentation of these aspects will enable a comprehensive cost estimation, revealing how MBSE reduces manpower costs through time savings and quantifying these savings in ship acquisition. Comparing defined costs and benefits will ultimately unveil the overall value proposition of implementing MBSE in ship design.
- **Comparative Analysis of More Modeling Languages:**
  Both Capella and CDP4-COMET are proven MBSE tools and their features provide a general sense of MBSE capabilities However, there are many other MBSE tools available. Investigating a broader range of tools can help document the expected benefits required in early-stage naval ship design. This, in turn, allows for a more confident generalization of MBSE's value. SysML tools such as Cameo Systems Modeler, MagicDraw, Enterprise Architect, and Rational Rhapsody are recommended for this investigation.
- **Addressing Scalability Challenges:**
  Scalability is another fundamental factor, and its importance is discussed separately. The current thesis focused on defining the system requirements for a specific part of a specific mission, intentionally limiting the model to only 40-50 pieces of equipment. However, a realistic model of an LPD vessel could involve thousands of pieces of equipment in full scale. The number of data points and model parameters to be managed will increase exponentially for larger systems, either due to more components or increased interactions/interfaces between components. Consequently, scalability depends on the capability of the IT infrastructure. Studying the tools' abilities to handle such large amounts of data is paramount.
- **Pursuing a Single Source of Truth by Integrating Multiple MBSE Tools:**
  Both the literature review and modeling efforts showed that no single MBSE tool excels in all design phases or offers all necessary collaborative features for the lengthy warship design process. Therefore, embracing the concept of a "Single Source of Truth" through collaborative MBSE applications is crucial, particularly in the foreseeable future. To realize this vision, adapters for tool synchronization are essential. Consequently, further research in this area is deemed essential to fully assess the value of MBSE. While pre-existing adaptors for both Capella and COMET were identified, it became evident that utilizing them may result in potential information loss. Thus, future research should also prioritize further software development and optimization of tool adaptors to ensure seamless data exchange and synchronization across MBSE platforms.
- **Integration of MBSE Tools with Existing Ship Design Tools:**

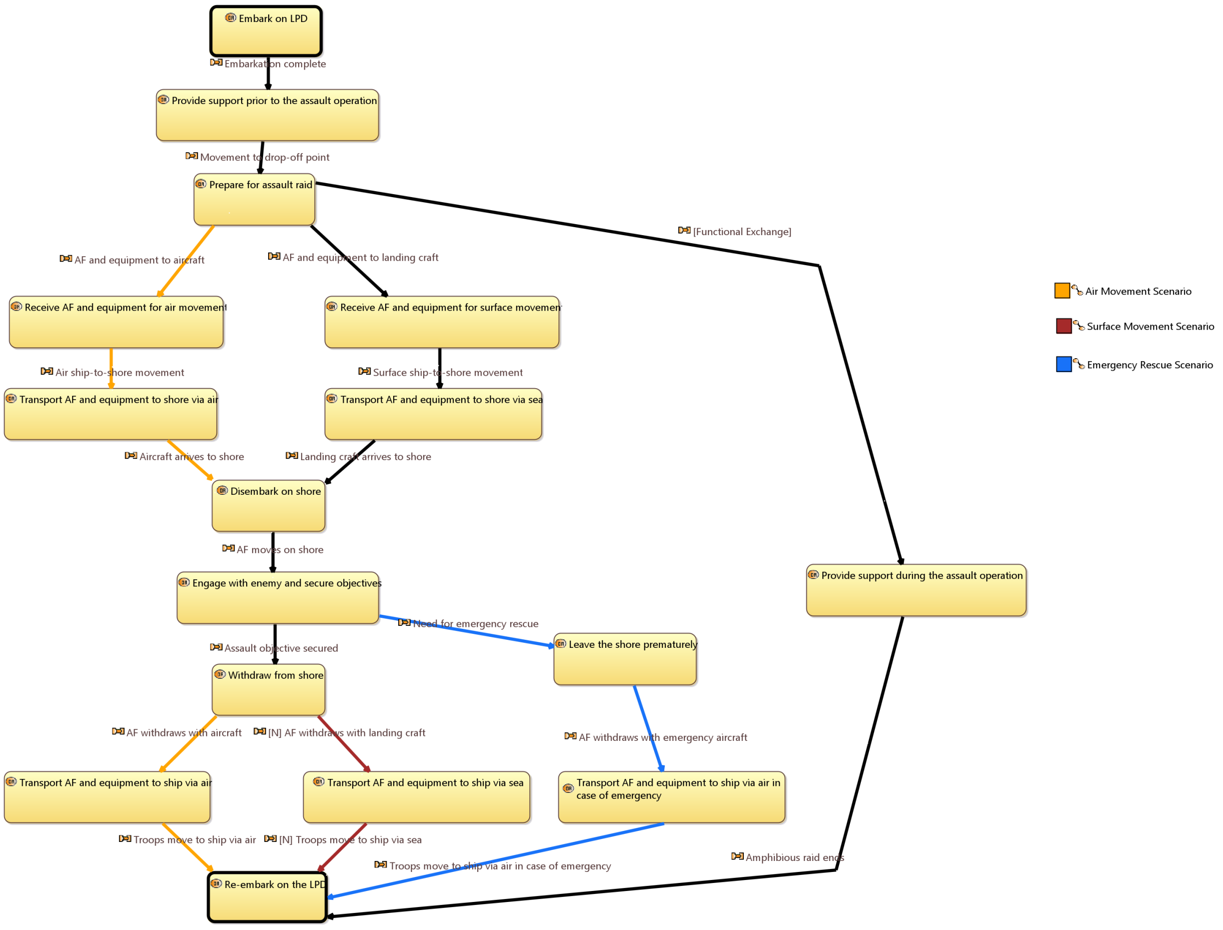
The emphasis should extend beyond integrating MBSE tools with each other to encompass integration with other essential tools in ship design. To conduct concept exploration and concept definition studies, ship designers invested in a set of early-stage ship design tools. The toolset includes performance prediction tools, costing models, operational models, data visualization, and ship synthesis tools. While future MBSE tools hold promise in seamlessly integrating these elements, the integration of the latter remains challenging. Within the Dutch Command Materiel and IT (COMMIT) agency, formerly known as the Dutch Defence Materiel Organization (DMO), the Maritime Systems Division, responsible for warship procurement projects, employs ship synthesis tools at various stages. In the conceptual exploration phase, tools like PACKING are utilized, while in the concept definition phase, tools like WARGEAR and FIDES (Functional Integrated Design Exploration of Ships) come into play (Oers et al. 2018). Often, blocks are visualized within the 3D-CAD software like Rhinoceros. Developers should prioritize integrating MBSE tools with such essential components, thereby emphasizing the integration with 3D CAD software.

- **Enhancing Organizational Readiness:**
  Assessing organizational readiness is another critical parameter for the successful implementation of MBSE practices. Through interactions with industry experts during this thesis and insights gleaned from the literature study, it is evident that the maritime industry exhibits a notable reluctance to embrace innovation compared to other sectors such as Transportation and Mobility, Aerospace, and Energy. This underscores the importance of investigating strategies to evaluate organizational culture, IT infrastructure capabilities, and readiness for adopting new methodologies. Proactively addressing these challenges and overcoming barriers will be crucial. Implementing effective training programs and pilot initiatives can benefit organizational readiness.
- **Promoting Reusability and Leveraging Existing Designs:**
  While innovation is crucial in naval vessel design, it is uncommon for naval vessels to be entirely developed from scratch without leveraging existing designs (Droste, Bedert, and Audenaert 2024). Drawing upon proven architectures and configurations allows designers to optimize specific aspects, enhancing reliability and cost-effectiveness. Therefore, an approach such as MBSE, which can also start deliberately with existing designs, would significantly benefit the industry. Thus, the reusability of MBSE artifacts is vital and should be enhanced even more in future projects. More ship-related reference data have to be Incorporated into MBSE tools to expedite development and improve efficiency. This data needs to be defined only once and can be reused across different vessel types. For example, CDP4-COMET provides the capabilities and foundation to achieve this by offering access to multiple data sources. It includes options and predefined data libraries that can be reused across engineering models, as well as catalogues intended to facilitate the reuse of element definitions. However, reference data and existing ship models for reuse are scarce, with the majority found in the software or aerospace industries. Efforts are urgently required to tailor these tools for naval applications and establish foundational reference data crucial for warship design.
- **Enhancing Communication and Simplified Interfaces to Streamline Collaboration:**
  It is pivotal to ensure that the results can be effectively communicated across diverse stakeholders in the ship design industry through various means such as shared diagrams, online documents, and platforms. Moreover, recognizing that not every stakeholder involved in the ship design process can be expected to acquire expertise in MBSE, developing an interface that does not demand extensive MBSE expertise is crucial for future development. Such an interface could facilitate collaboration with stakeholders beyond the MBSE community. For example, providing a simplified interface for cost or material engineers to input cost or material parameters without directly interacting with MBSE tools can streamline the design process and improve cross-functional communication.

9 Appendix

Diagrams created in Capella

Diagrams of operational analysis



Embark on LPD
Embarkation complete
Provide support prior to the assault operation
Movement to drop-off point
Prepare for assault raid
[Functional Exchange]
AF and equipment to aircraft
AF and equipment to landing craft
Receive AF and equipment for air movement
Receive AF and equipment for surface movement
Air ship-to-shore movement
Surface ship-to-shore movement
Transport AF and equipment to shore via air
Transport AF and equipment to shore via sea
Aircraft arrives to shore
Landing craft arrives to shore
Disembark on shore
AF moves on shore
Engage with enemy and secure objectives
Need for emergency rescue
Provide support during the assault operation
Leave the shore prematurely
Assault objective secured
Withdraw from shore
AF withdraws with aircraft
[N] AF withdraws with landing craft
AF withdraws with emergency aircraft
Transport AF and equipment to ship via air
Transport AF and equipment to ship via sea
Transport AF and equipment to ship via air in case of emergency
Troops move to ship via air
[N] Troops move to ship via sea
Troops move to ship via air in case of emergency
Amphibious raid ends
Re-embark on the LPD
🟧 Air Movement Scenario
🟥 Surface Movement Scenario
🟦 Emergency Rescue Scenario

Figure 9.1: [OAIB] Root Operational Activity

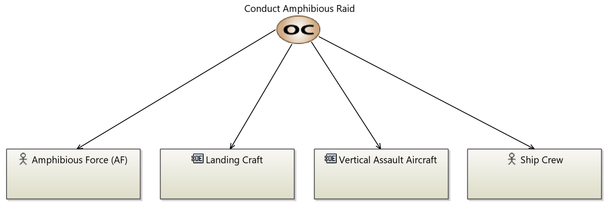


Figure 9.2: [OCB] Operational Capabilities AWS

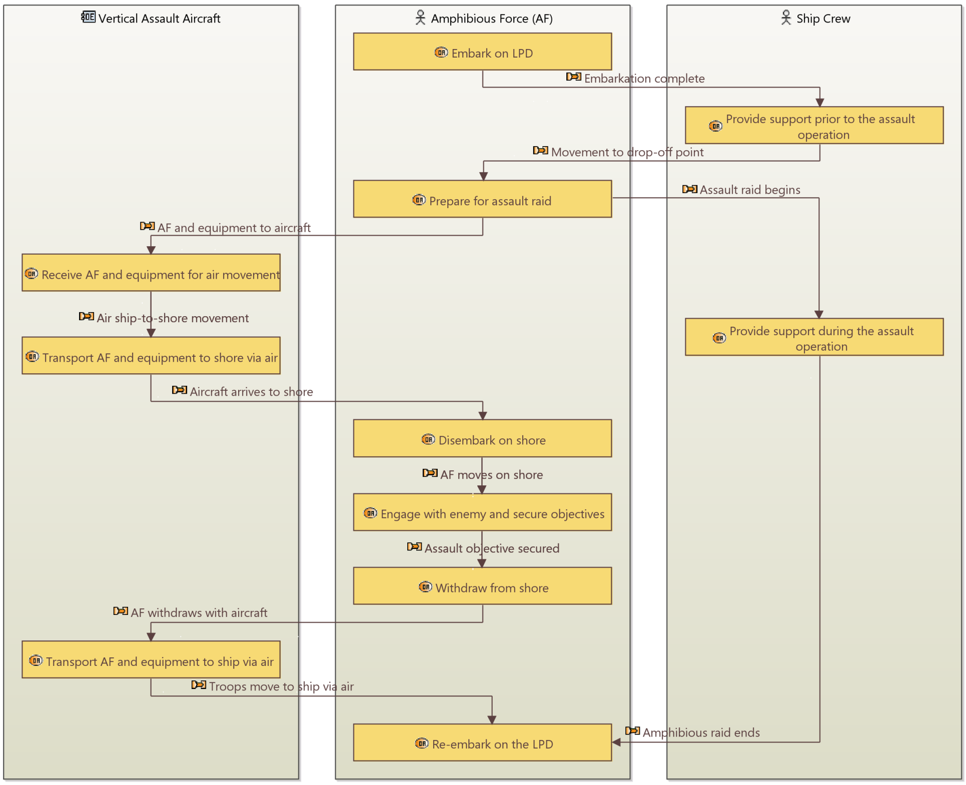

Vertical Assault Aircraft
Amphibious Force (AF)
Ship Crew
Embark on LPD
Embarkation complete
Provide support prior to the assault operation
Movement to drop-off point
Prepare for assault raid
Assault raid begins
AF and equipment to aircraft
Receive AF and equipment for air movement
Air ship-to-shore movement
Transport AF and equipment to shore via air
Provide support during the assault operation
Aircraft arrives to shore
Disembark on shore
AF moves on shore
Engage with enemy and secure objectives
Assault objective secured
Withdraw from shore
AF withdraws with aircraft
Transport AF and equipment to ship via air
Troops move to ship via air
Amphibious raid ends
Re-embark on the LPD

Figure 9.3: [OAB] Activities for Air Movement Scenario

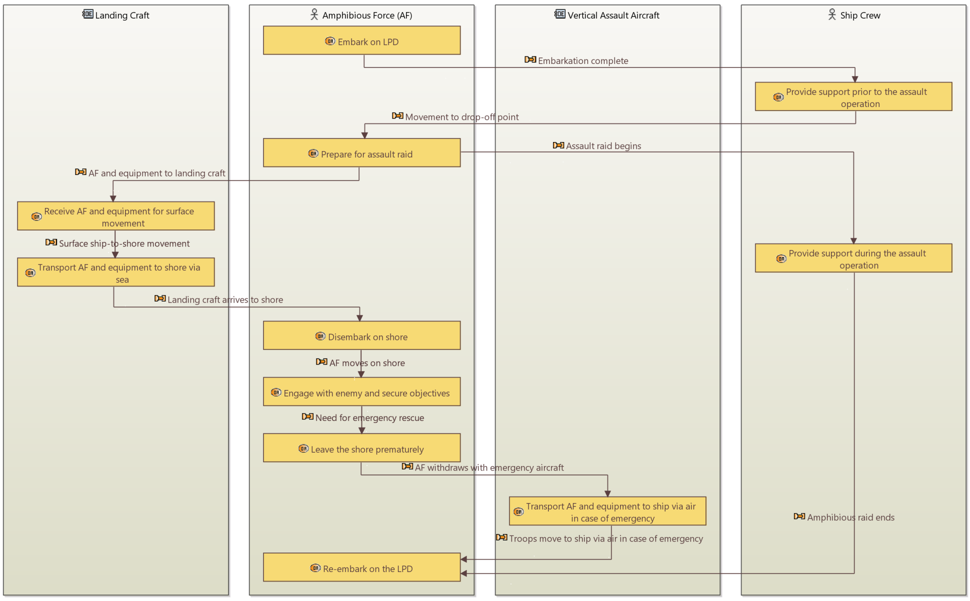

Landing Craft
Amphibious Force (AF)
Vertical Assault Aircraft
Ship Crew
Embark on LPD
Embarkation complete
Provide support prior to the assault operation
Movement to drop-off point
Prepare for assault raid
Assault raid begins
AF and equipment to landing craft
Receive AF and equipment for surface movement
Surface ship-to-shore movement
Transport AF and equipment to shore via sea
Provide support during the assault operation
Landing craft arrives to shore
Disembark on shore
AF moves on shore
Engage with enemy and secure objectives
Need for emergency rescue
Leave the shore prematurely
AF withdraws with emergency aircraft
Transport AF and equipment to ship via air in case of emergency
Amphibious raid ends
Troops move to ship via air in case of emergency
Re-embark on the LPD

Figure 9.4: [OAB] Activities for Emergency Rescue Scenario

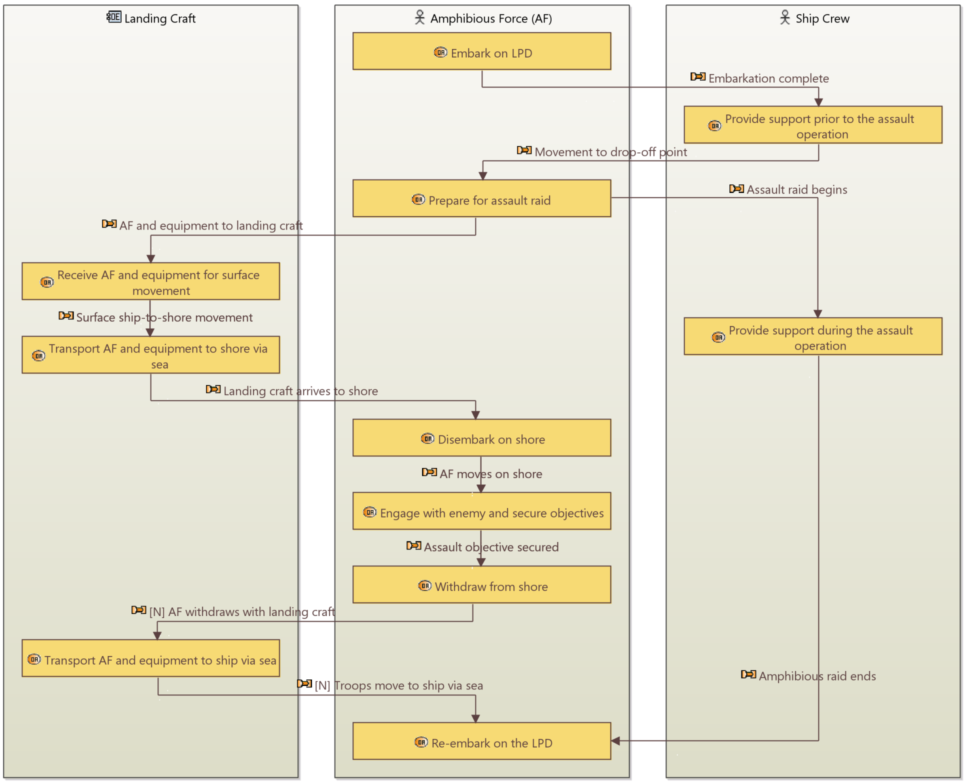

Landing Craft
Amphibious Force (AF)
Ship Crew
Embark on LPD
Embarkation complete
Provide support prior to the assault operation
Movement to drop-off point
Prepare for assault raid
Assault raid begins
AF and equipment to landing craft
Receive AF and equipment for surface movement
Surface ship-to-shore movement
Transport AF and equipment to shore via sea
Provide support during the assault operation
Landing craft arrives to shore
Disembark on shore
AF moves on shore
Engage with enemy and secure objectives
Assault objective secured
Withdraw from shore
[N] AF withdraws with landing craft
Transport AF and equipment to ship via sea
Amphibious raid ends
[N] Troops move to ship via sea
Re-embark on the LPD

Figure 9.5: [OAB] Activities for Surface Movement Scenario

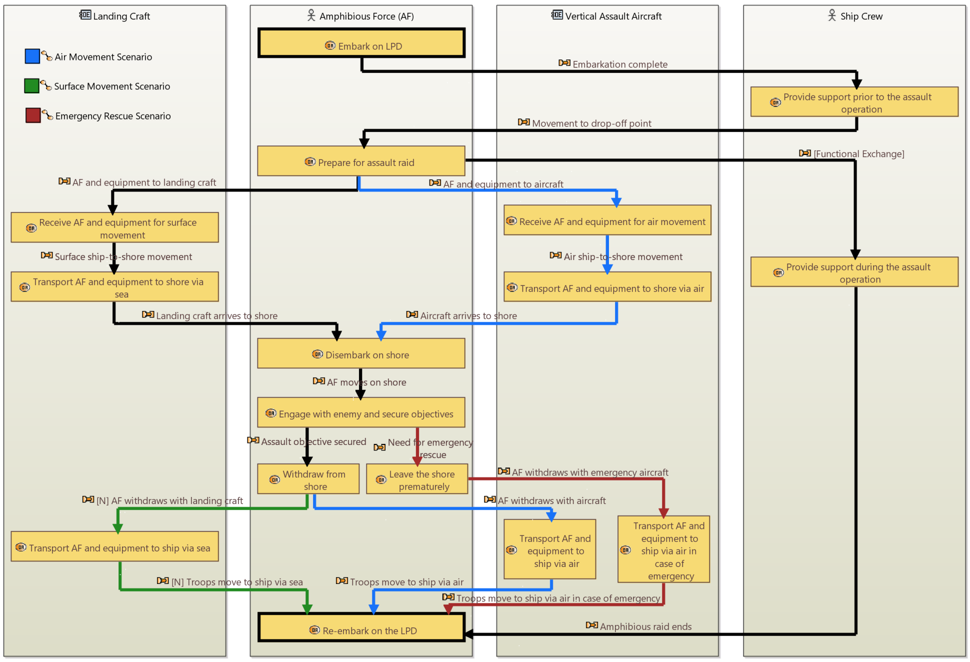

Landing Craft
Amphibious Force (AF)
Vertical Assault Aircraft
Ship Crew
🟦 Air Movement Scenario
🟩 Surface Movement Scenario
🟥 Emergency Rescue Scenario
Embark on LPD
Embarkation complete
Provide support prior to the assault operation
Movement to drop-off point
Prepare for assault raid
[Functional Exchange]
AF and equipment to landing craft
AF and equipment to aircraft
Receive AF and equipment for surface movement
Receive AF and equipment for air movement
Surface ship-to-shore movement
Air ship-to-shore movement
Provide support during the assault operation
Transport AF and equipment to shore via sea
Transport AF and equipment to shore via air
Landing craft arrives to shore
Aircraft arrives to shore
Disembark on shore
AF moves on shore
Engage with enemy and secure objectives
Assault objective secured
Need for emergency rescue
Withdraw from shore
Leave the shore prematurely
AF withdraws with emergency aircraft
[N] AF withdraws with landing craft
AF withdraws with aircraft
Transport AF and equipment to ship via sea
Transport AF and equipment to ship via air
Transport AF and equipment to ship via air in case of emergency
[N] Troops move to ship via sea
Troops move to ship via air
Troops move to ship via air in case of emergency
Amphibious raid ends
Re-embark on the LPD

Figure 9.6: [OAB] Operational Architecture

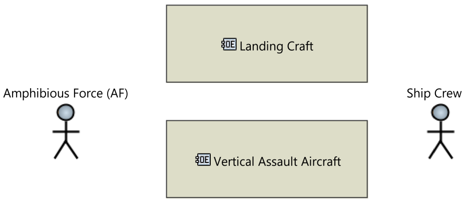


Figure 9.7: [OEBD] Operational Entities AWS

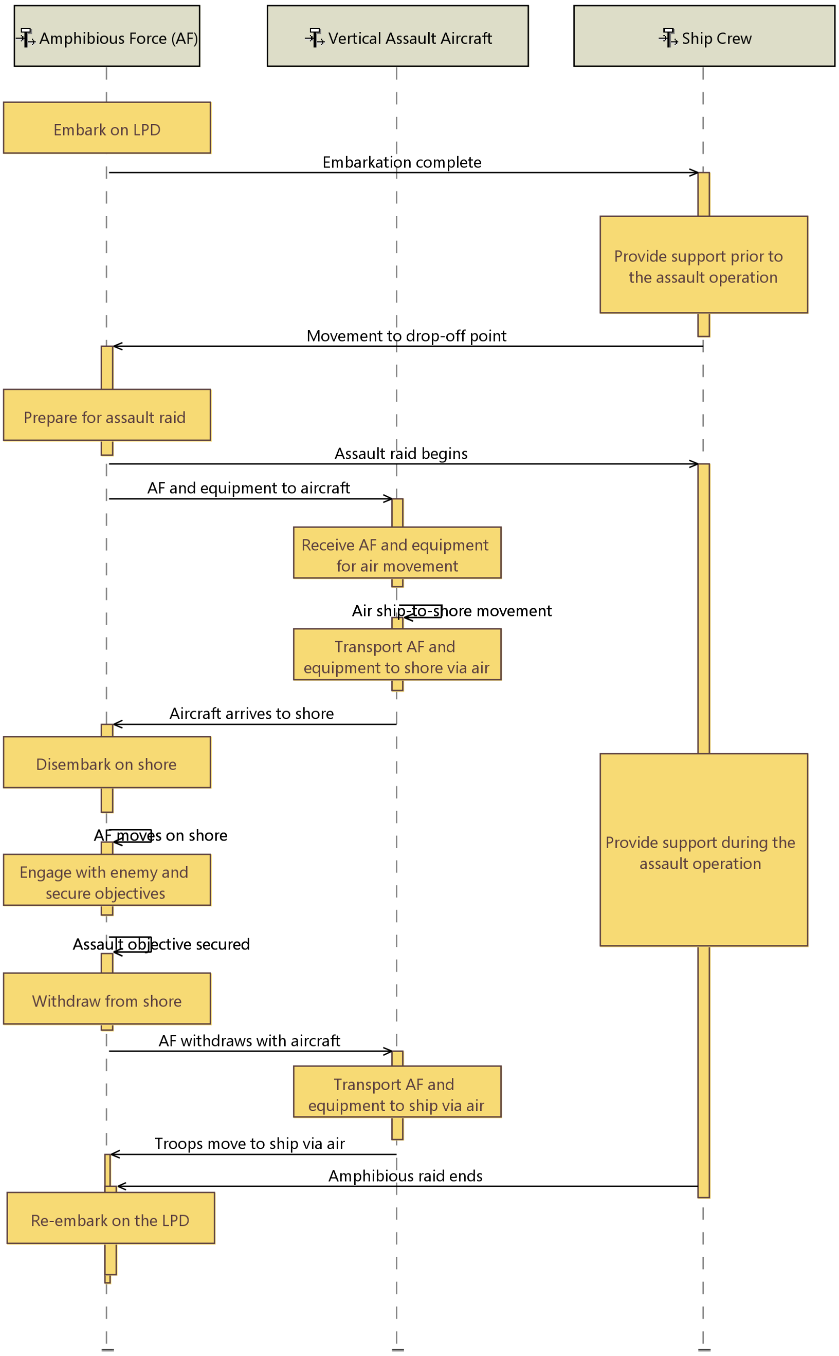

Amphibious Force (AF)
Vertical Assault Aircraft
Ship Crew
Embark on LPD
Embarkation complete
Provide support prior to the assault operation
Movement to drop-off point
Prepare for assault raid
Assault raid begins
AF and equipment to aircraft
Receive AF and equipment for air movement
Air ship-to-shore movement
Transport AF and equipment to shore via air
Aircraft arrives to shore
Disembark on shore
Provide support during the assault operation
AF moves on shore
Engage with enemy and secure objectives
Assault objective secured
Withdraw from shore
AF withdraws with aircraft
Transport AF and equipment to ship via air
Troops move to ship via air
Amphibious raid ends
Re-embark on the LPD

Figure 9.8: [OES] Air Movement Scenario

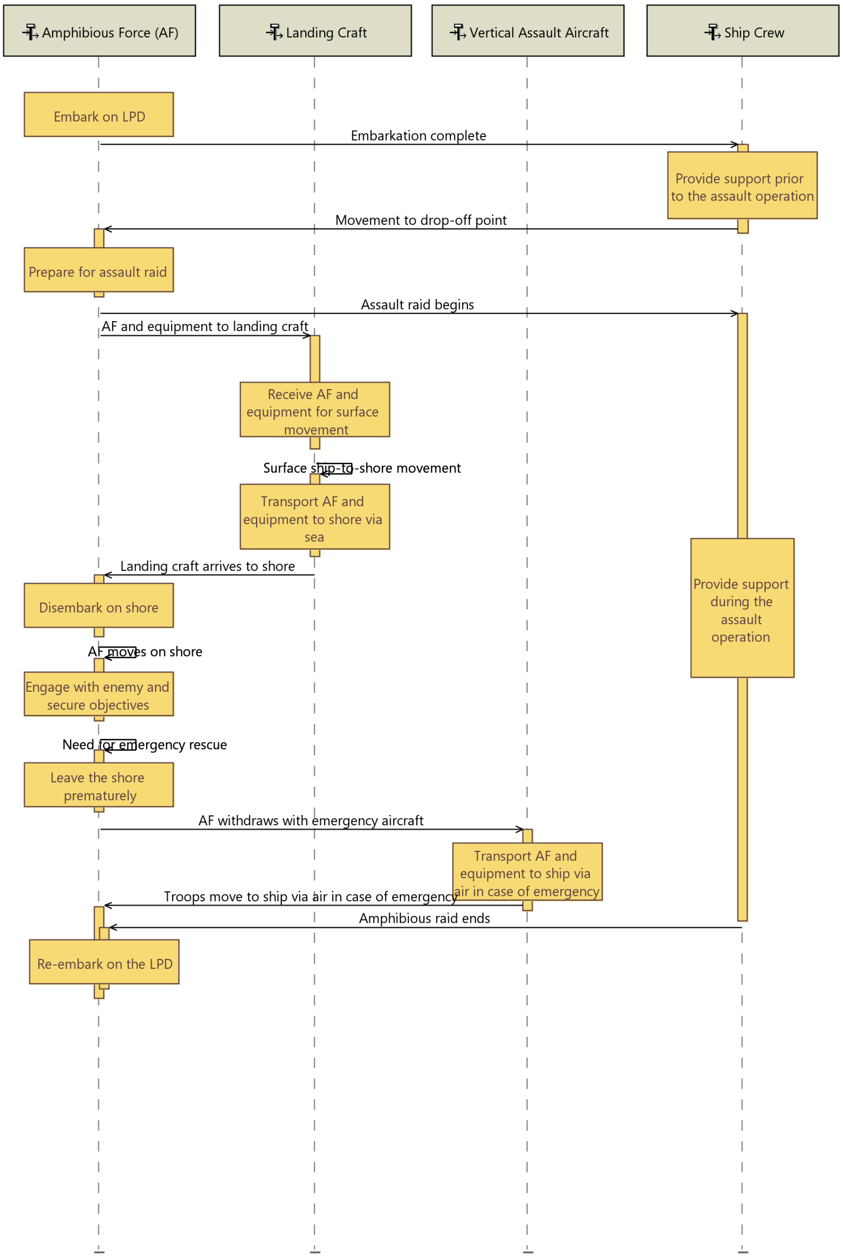

Amphibious Force (AF)
Landing Craft
Vertical Assault Aircraft
Ship Crew
Embark on LPD
Embarkation complete
Provide support prior to the assault operation
Movement to drop-off point
Prepare for assault raid
Assault raid begins
AF and equipment to landing craft
Receive AF and equipment for surface movement
Surface ship-to-shore movement
Transport AF and equipment to shore via sea
Provide support during the assault operation
Landing craft arrives to shore
Disembark on shore
AF moves on shore
Engage with enemy and secure objectives
Need for emergency rescue
Leave the shore prematurely
AF withdraws with emergency aircraft
Transport AF and equipment to ship via air in case of emergency
Troops move to ship via air in case of emergency
Amphibious raid ends
Re-embark on the LPD

Figure 9.9: [OES] Emergency Rescue Scenario

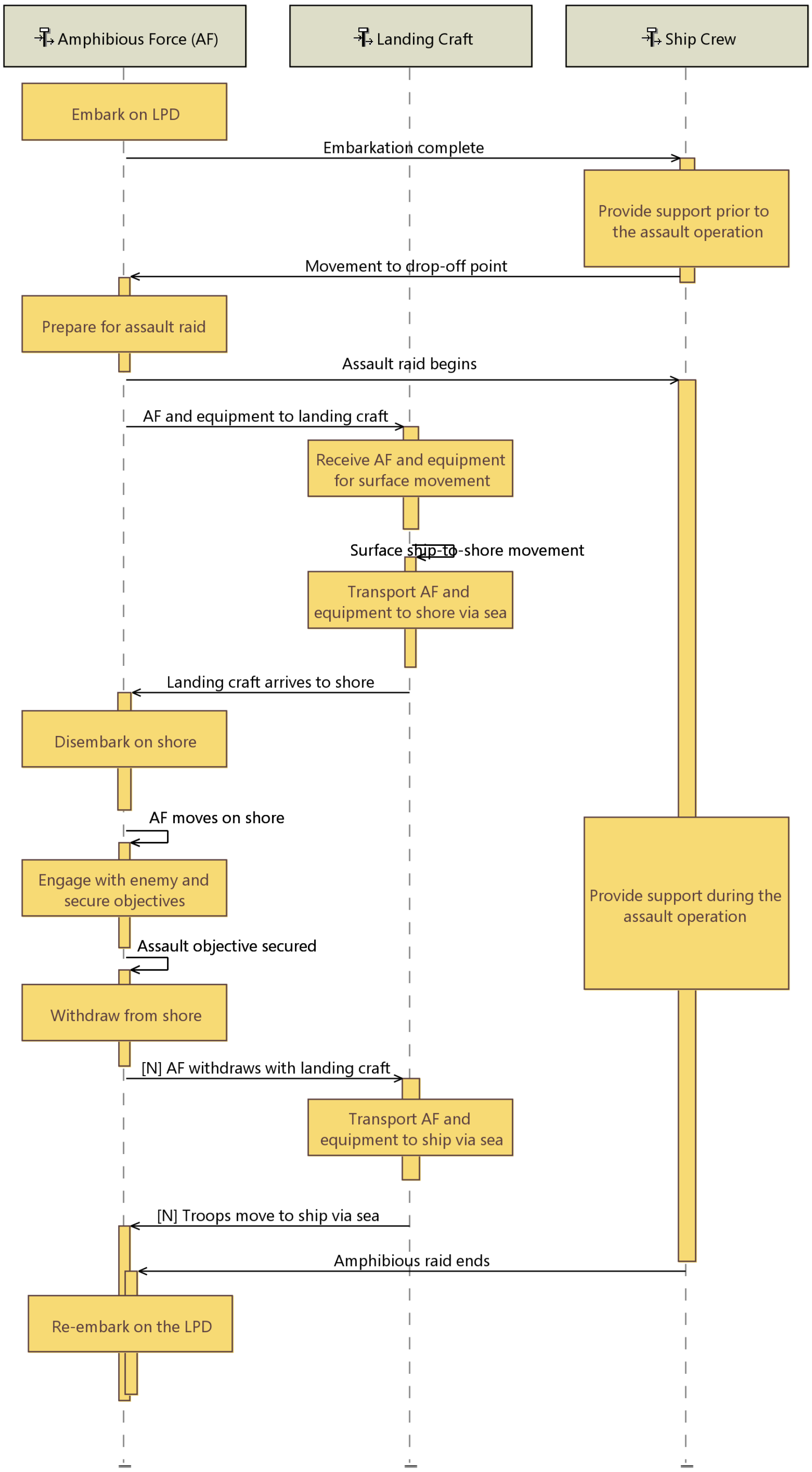

Amphibious Force (AF)
Landing Craft
Ship Crew
Embark on LPD
Embarkation complete
Provide support prior to the assault operation
Movement to drop-off point
Prepare for assault raid
Assault raid begins
AF and equipment to landing craft
Receive AF and equipment for surface movement
Surface ship-to-shore movement
Transport AF and equipment to shore via sea
Landing craft arrives to shore
Disembark on shore
AF moves on shore
Engage with enemy and secure objectives
Provide support during the assault operation
Assault objective secured
Withdraw from shore
[N] AF withdraws with landing craft
Transport AF and equipment to ship via sea
[N] Troops move to ship via sea
Amphibious raid ends
Re-embark on the LPD

Figure 9.10: [OES] Surface Movement Scenario

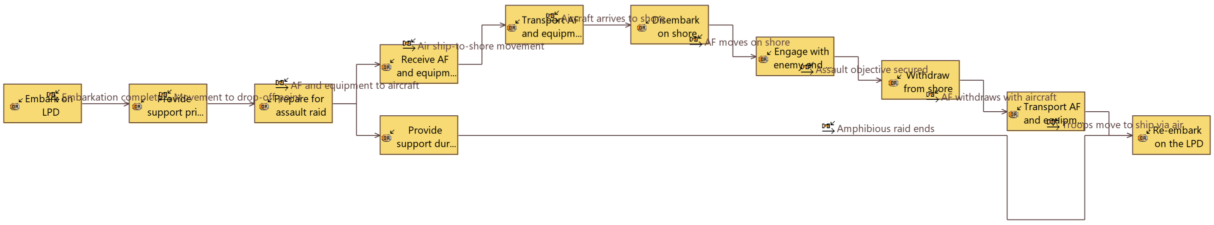


Figure 9.11: [OPD] Air Movement Scenario

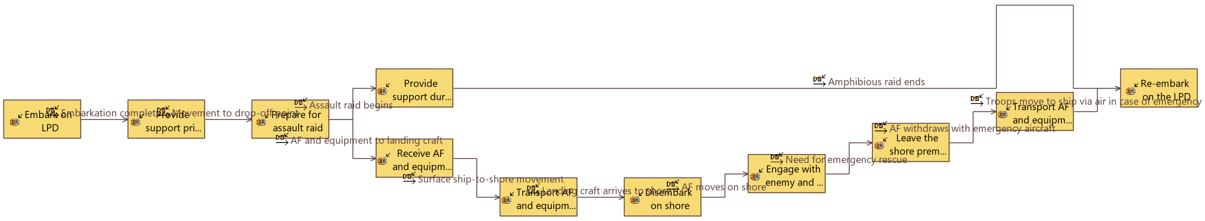

Embark on LPD
Embarkation complete
Provide support pri...
Movement to drop-off point
Prepare for assault raid
Assault raid begins
AF and equipment to landing craft
Provide support dur...
Amphibious raid ends
Receive AF and equipm...
Surface ship-to-shore movement
Transport AF and equipm...
Landing craft arrives to shore
Disembark on shore
AF moves on shore
Engage with enemy and ...
Need for emergency rescue
Leave the shore prem...
AF withdraws with emergency aircraft
Transport AF and equipm...
Troops move to ship via air in case of emergency
Re-embark on the LPD

Figure 9.12: [OPD] Emergency Rescue Scenario

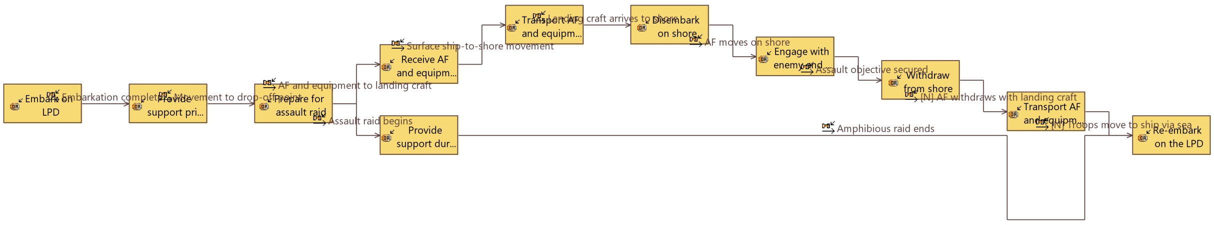

Embark on LPD
Embarkation complete
Provide support pri...
Movement to drop-off point
Prepare for assault raid
AF and equipment to landing craft
Assault raid begins
Receive AF and equipm...
Surface ship-to-shore movement
Provide support dur...
Transport AF and equipm...
Landing craft arrives to shore
Disembark on shore
AF moves on shore
Engage with enemy and...
Assault objective secured
Withdraw from shore
[N] AF withdraws with landing craft
Transport AF and equipm...
[N] Troops move to ship via sea
Amphibious raid ends
Re-embark on the LPD

Figure 9.13: [OPD] Surface Movement Scenario

Diagrams of System Analysis

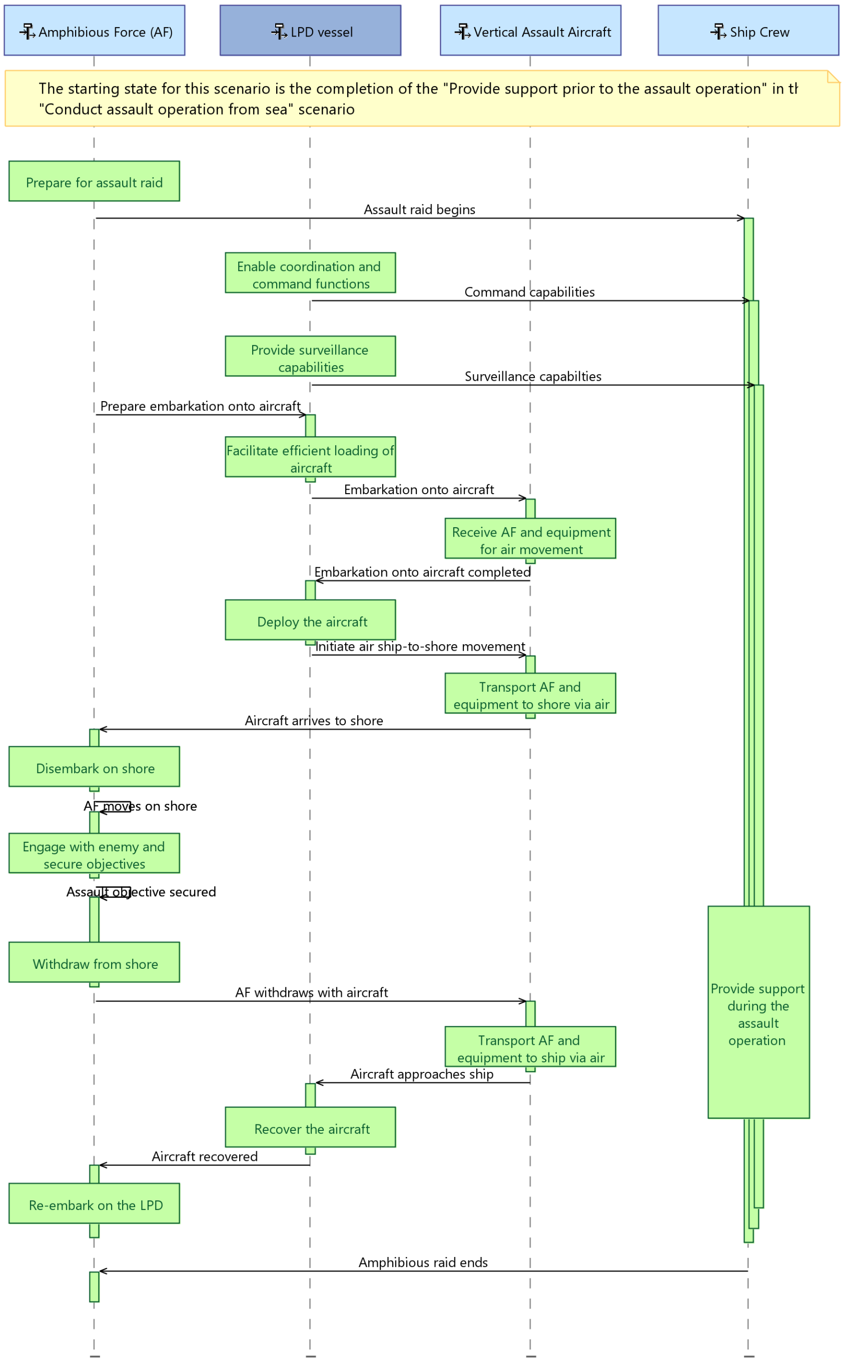
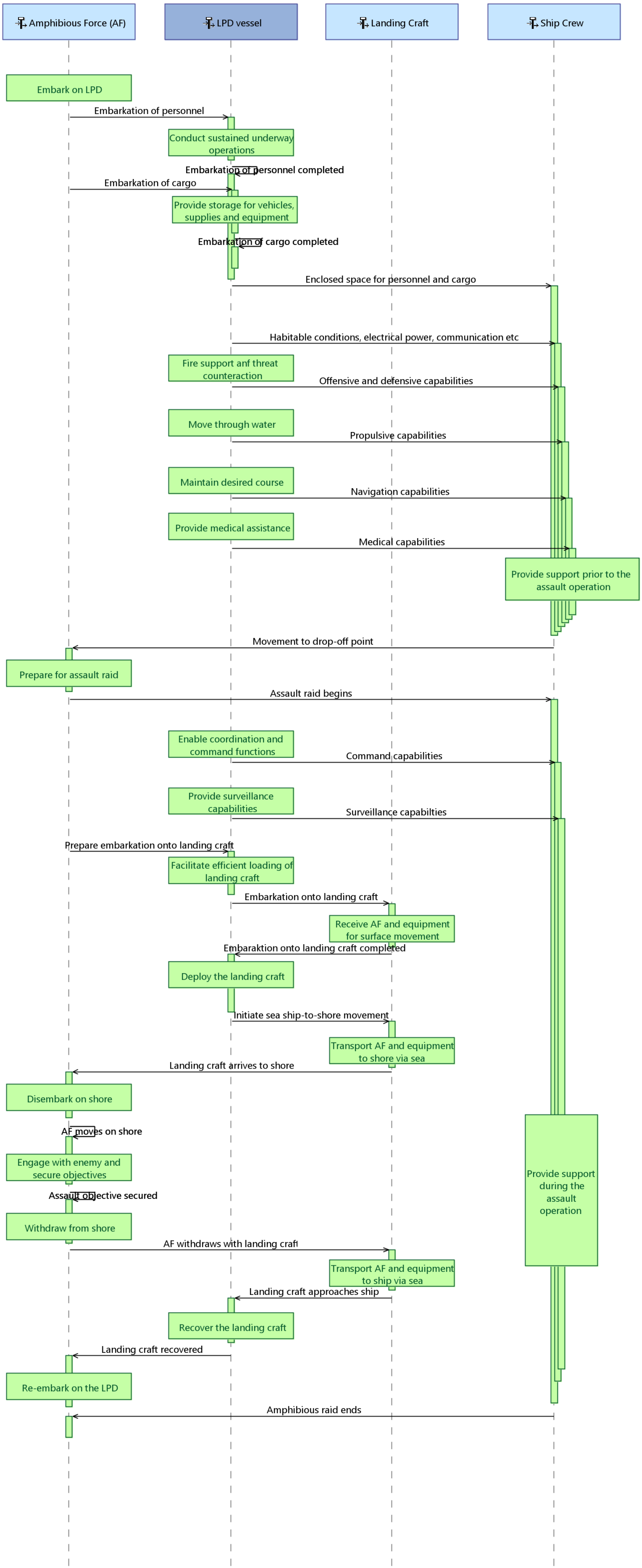

Amphibious Force (AF)
LPD vessel
Vertical Assault Aircraft
Ship Crew
The starting state for this scenario is the completion of the "Provide support prior to the assault operation" in tł
"Conduct assault operation from sea" scenario
Prepare for assault raid
Assault raid begins
Enable coordination and command functions
Command capabilities
Provide surveillance capabilities
Surveillance capabilties
Prepare embarkation onto aircraft
Facilitate efficient loading of aircraft
Embarkation onto aircraft
Receive AF and equipment for air movement
Embarkation onto aircraft completed
Deploy the aircraft
Initiate air ship-to-shore movement
Transport AF and equipment to shore via air
Aircraft arrives to shore
Disembark on shore
AF moves on shore
Engage with enemy and secure objectives
Assault objective secured
Withdraw from shore
AF withdraws with aircraft
Transport AF and equipment to ship via air
Aircraft approaches ship
Recover the aircraft
Aircraft recovered
Re-embark on the LPD
Provide support during the assault operation
Amphibious raid ends

Figure 9.14: [ES] Conduct assault operation from air


Amphibious Force (AF)
LPD vessel
Landing Craft
Ship Crew
Embark on LPD
Embarkation of personnel
Conduct sustained underway operations
Embarkation of personnel completed
Embarkation of cargo
Provide storage for vehicles, supplies and equipment
Embarkation of cargo completed
Enclosed space for personnel and cargo
Habitable conditions, electrical power, communication etc
Fire support anf threat counteraction
Offensive and defensive capabilities
Move through water
Propulsive capabilities
Maintain desired course
Navigation capabilities
Provide medical assistance
Medical capabilities
Provide support prior to the assault operation
Movement to drop-off point
Prepare for assault raid
Assault raid begins
Enable coordination and command functions
Command capabilities
Provide surveillance capabilities
Surveillance capabilties
Prepare embarkation onto landing craft
Facilitate efficient loading of landing craft
Embarkation onto landing craft
Receive AF and equipment for surface movement
Embaraktion onto landing craft completed
Deploy the landing craft
Initiate sea ship-to-shore movement
Transport AF and equipment to shore via sea
Landing craft arrives to shore
Disembark on shore
AF moves on shore
Engage with enemy and secure objectives
Assault objective secured
Withdraw from shore
Provide support during the assault operation
AF withdraws with landing craft
Transport AF and equipment to ship via sea
Landing craft approaches ship
Recover the landing craft
Landing craft recovered
Re-embark on the LPD
Amphibious raid ends

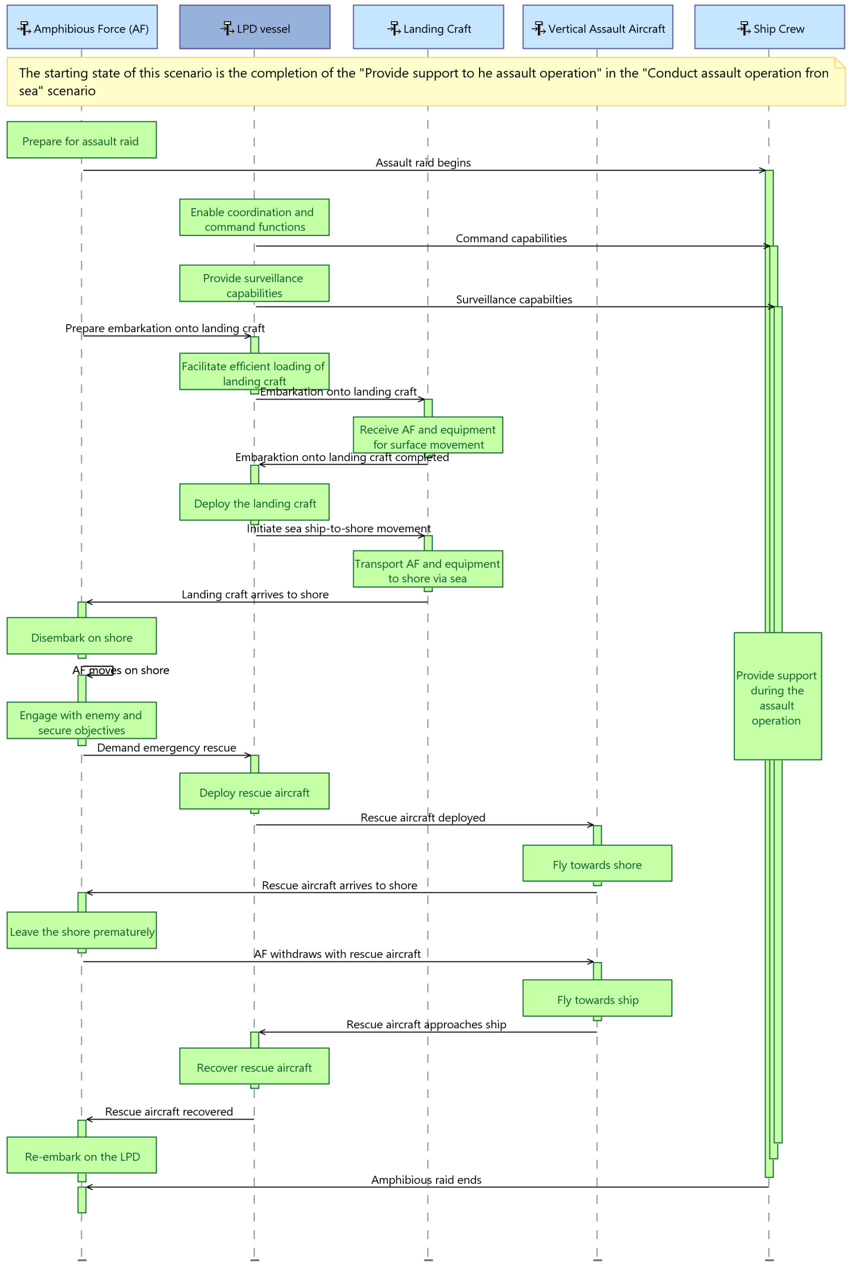

Amphibious Force (AF)
LPD vessel
Landing Craft
Vertical Assault Aircraft
Ship Crew
The starting state of this scenario is the completion of the "Provide support to he assault operation" in the "Conduct assault operation fron sea" scenario
Prepare for assault raid
Assault raid begins
Enable coordination and command functions
Command capabilities
Provide surveillance capabilities
Surveillance capabilties
Prepare embarkation onto landing craft
Facilitate efficient loading of landing craft
Embarkation onto landing craft
Receive AF and equipment for surface movement
Embaraktion onto landing craft completed
Deploy the landing craft
Initiate sea ship-to-shore movement
Transport AF and equipment to shore via sea
Landing craft arrives to shore
Disembark on shore
AF moves on shore
Engage with enemy and secure objectives
Demand emergency rescue
Deploy rescue aircraft
Rescue aircraft deployed
Fly towards shore
Rescue aircraft arrives to shore
Leave the shore prematurely
AF withdraws with rescue aircraft
Fly towards ship
Rescue aircraft approaches ship
Recover rescue aircraft
Rescue aircraft recovered
Re-embark on the LPD
Amphibious raid ends
Provide support during the assault operation

Figure 9.16: [ES] Provide Emergency Rescue

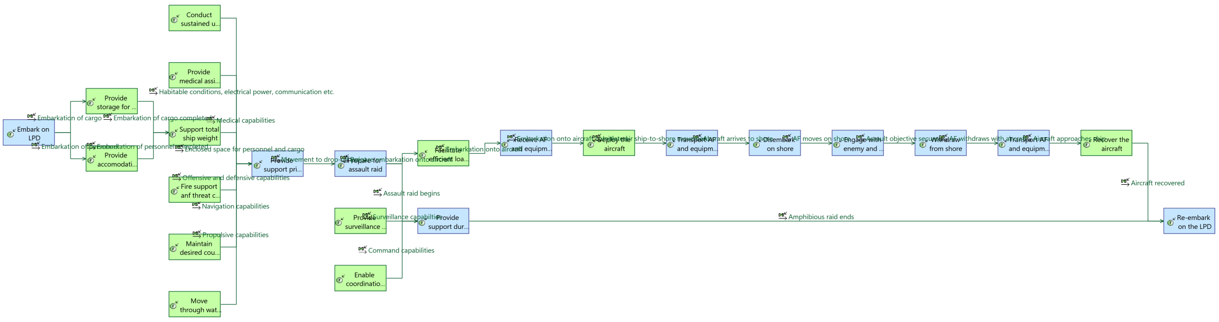


Figure 9.17: [SFCD] Air Movement Scenario

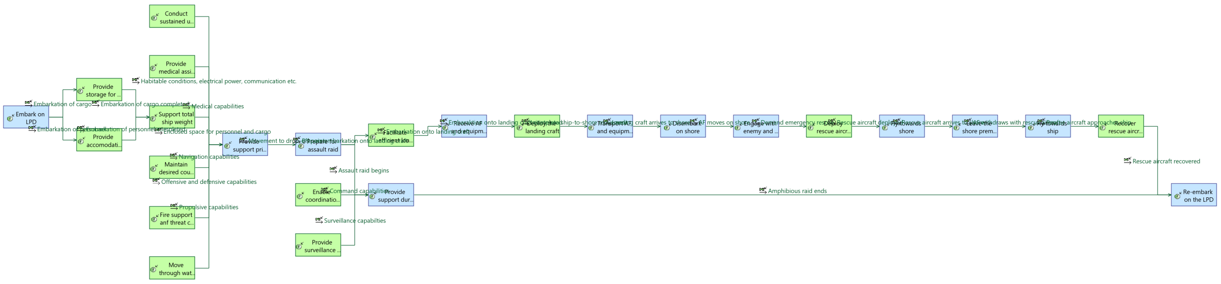


Figure 9.18: [SFCD] Emergency Rescue Scenario

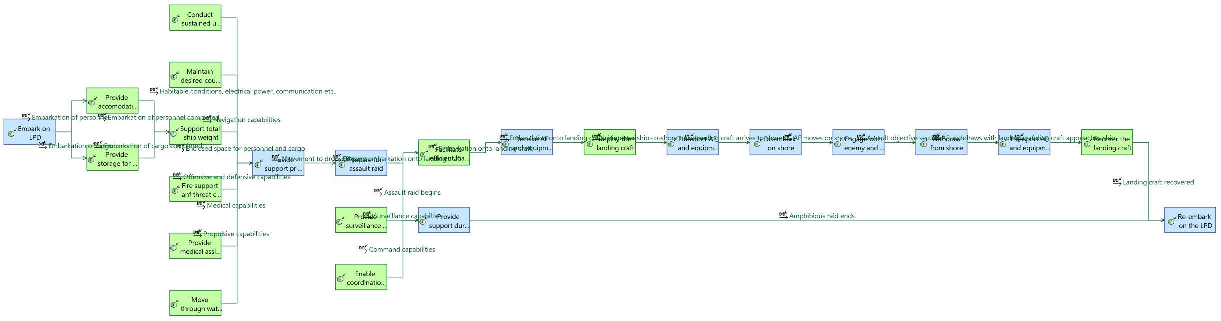

Conduct sustained u...
Maintain desired cou...
Habitable conditions, electrical power, communication etc.
Provide accomodati...
Embarkation of personnel
Embarkation of personnel completed
Navigation capabilities
Embark on LPD
Support total ship weight
Embarkation of cargo
Embarkation of cargo completed
Enclosed space for personnel and cargo
Provide storage for ...
Provide support pri...
Movement to drop off point
Prepare for assault raid
Facilitate efficient loa...
Embarkation onto landing craft
Offensive and defensive capabilities
Fire support anf threat c...
Medical capabilities
Assault raid begins
Provide surveillance ...
Surveillance capabilities
Provide support dur...
Propulsive capabilities
Provide medical assi...
Command capabilities
Enable coordinatio...
Move through wat...
Receive AF and equipm...
Deploy the landing craft
Transport AF and equipm...
Disembark on shore
Engage with enemy and ...
Withdraw from shore
Transport AF and equipm...
Recover the landing craft
Landing craft arrives to shore
AF moves on shore
Assault objective secured
AF withdraws with landing craft
Landing craft approaches ship
Landing craft recovered
Amphibious raid ends
Re-embark on the LPD

Figure 9.19: [SFCD] Surface Movement Scenario

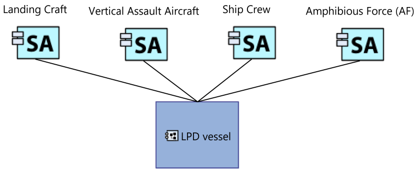


Figure 9.20: [CSA] LPD vessel

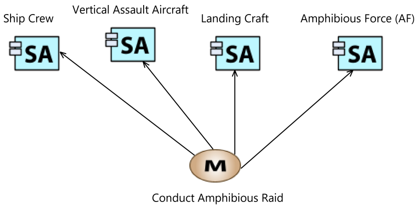


Figure 9.21: [MB] Missions

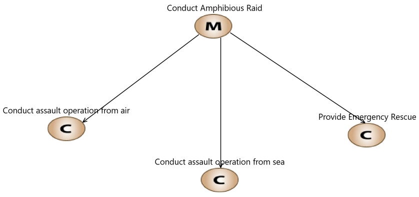

Conduct Amphibious Raid
M
Conduct assault operation from air
C
Provide Emergency Rescue
C
Conduct assault operation from sea
C

Figure 9.22: [MCB] Capabilities

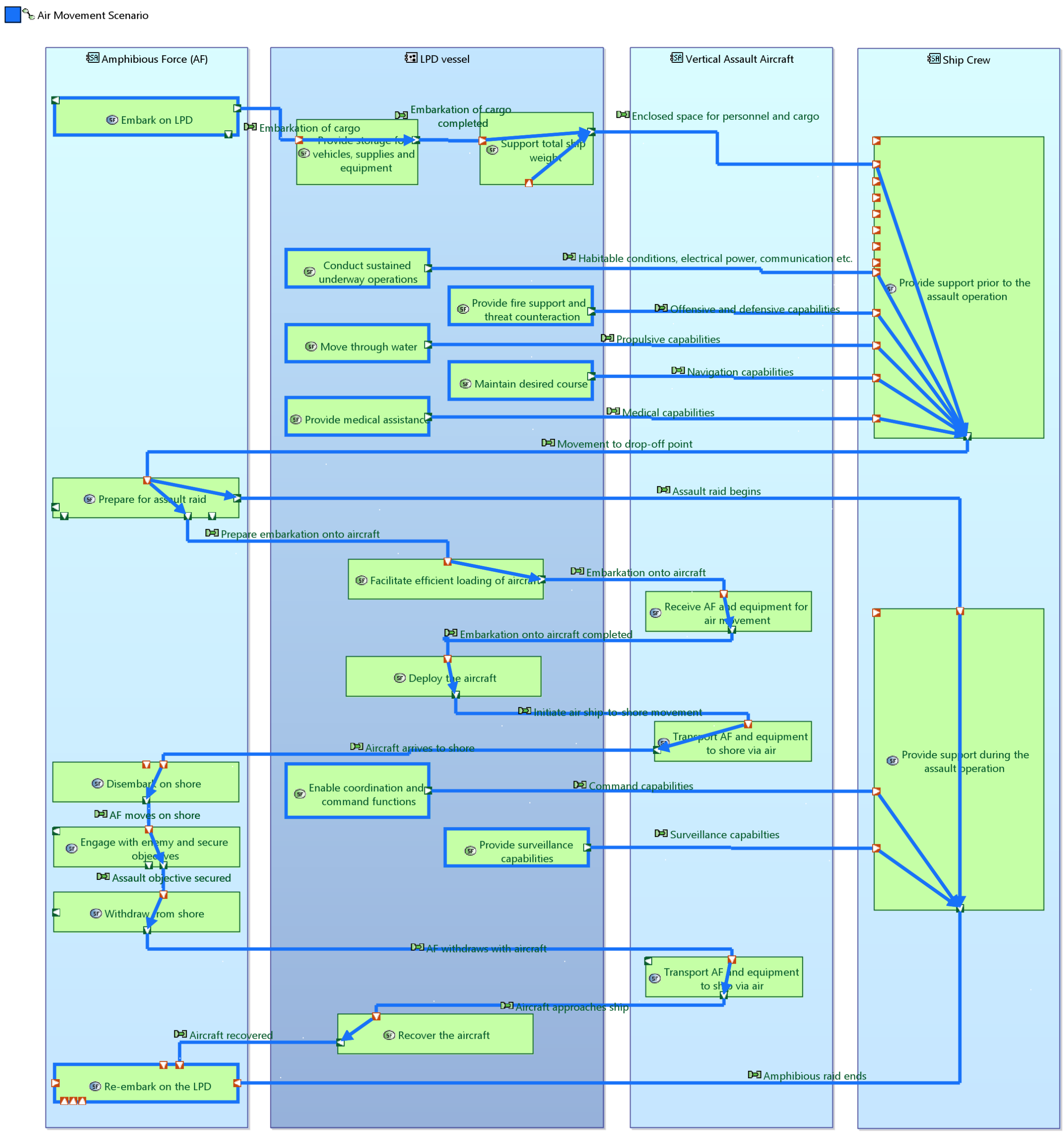

Air Movement Scenario
Amphibious Force (AF)
LPD vessel
Vertical Assault Aircraft
Ship Crew
Embark on LPD
Embarkation of cargo
Provide storage for vehicles, supplies and equipment
Embarkation of cargo completed
Support total ship weight
Enclosed space for personnel and cargo
Conduct sustained underway operations
Habitable conditions, electrical power, communication etc.
Provide support prior to the assault operation
Provide fire support and threat counteraction
Offensive and defensive capabilities
Move through water
Propulsive capabilities
Maintain desired course
Navigation capabilities
Provide medical assistance
Medical capabilities
Movement to drop-off point
Prepare for assault raid
Assault raid begins
Prepare embarkation onto aircraft
Facilitate efficient loading of aircraft
Embarkation onto aircraft
Receive AF and equipment for air movement
Embarkation onto aircraft completed
Deploy the aircraft
Initiate air ship-to-shore movement
Transport AF and equipment to shore via air
Aircraft arrives to shore
Provide support during the assault operation
Disembark on shore
Enable coordination and command functions
Command capabilities
AF moves on shore
Engage with enemy and secure objectives
Provide surveillance capabilities
Surveillance capabilties
Assault objective secured
Withdraw from shore
AF withdraws with aircraft
Transport AF and equipment to ship via air
Aircraft approaches ship
Recover the aircraft
Aircraft recovered
Re-embark on the LPD
Amphibious raid ends

Figure 9.23: [SAB] Architecture View of Air Movement Scenario

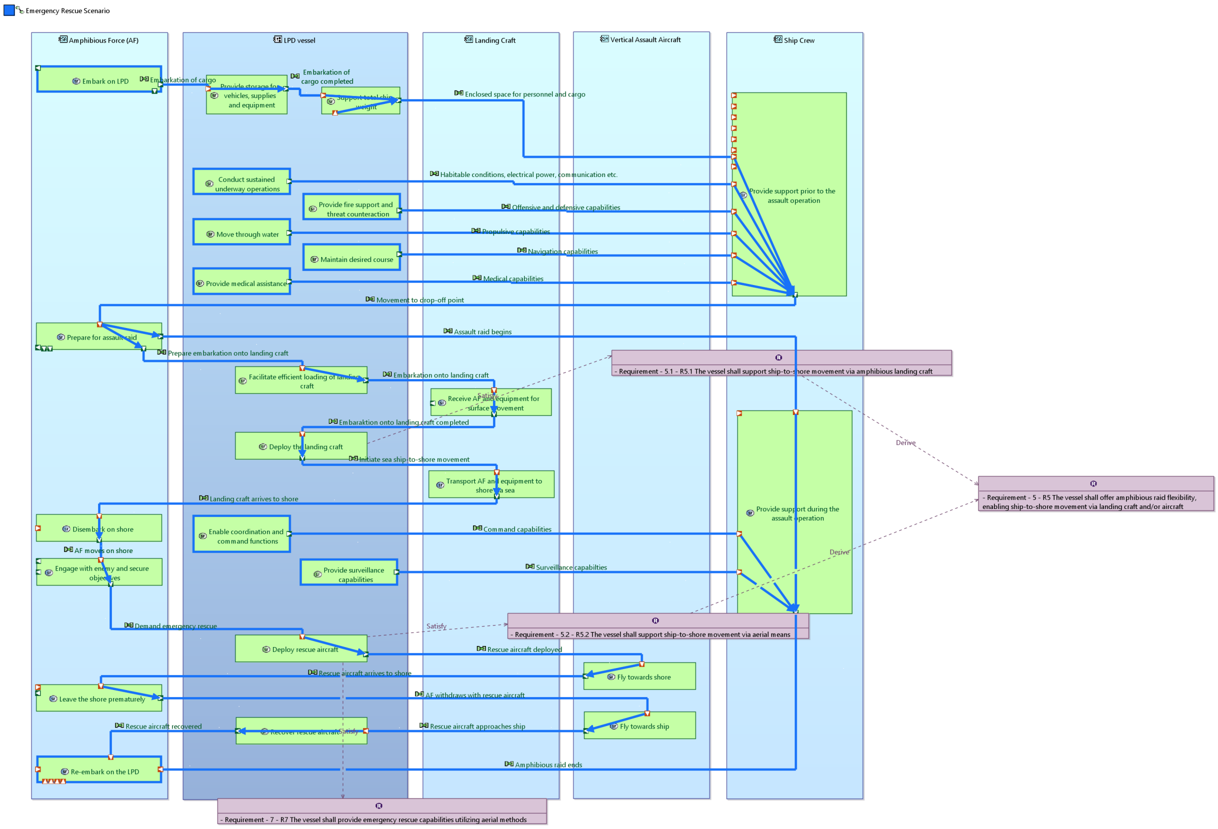

Emergency Rescue Scenario
Amphibious Force (AF)
LPD vessel
Landing Craft
Vertical Assault Aircraft
Ship Crew
Embark on LPD
Embarkation of cargo
Provide storage for vehicles, supplies and equipment
Embarkation of cargo completed
Support total ship weight
Enclosed space for personnel and cargo
Conduct sustained underway operations
Habitable conditions, electrical power, communication etc.
Provide fire support and threat counteraction
Offensive and defensive capabilities
Move through water
Propulsive capabilities
Maintain desired course
Navigation capabilities
Provide medical assistance
Medical capabilities
Provide support prior to the assault operation
Movement to drop-off point
Prepare for assault raid
Assault raid begins
Prepare embarkation onto landing craft
Facilitate efficient loading of landing craft
Embarkation onto landing craft
Receive AF and equipment for surface movement
Satisfy
- Requirement - 5.1 - R5.1 The vessel shall support ship-to-shore movement via amphibious landing craft
Embaraktion onto landing craft completed
Deploy the landing craft
Initiate sea ship-to-shore movement
Transport AF and equipment to shore via sea
Derive
- Requirement - 5 - R5 The vessel shall offer amphibious raid flexibility, enabling ship-to-shore movement via landing craft and/or aircraft
Landing craft arrives to shore
Disembark on shore
Enable coordination and command functions
Command capabilities
Provide support during the assault operation
AF moves on shore
Engage with enemy and secure objectives
Provide surveillance capabilities
Surveillance capabilties
Derive
Demand emergency rescue
Satisfy
- Requirement - 5.2 - R5.2 The vessel shall support ship-to-shore movement via aerial means
Deploy rescue aircraft
Rescue aircraft deployed
Fly towards shore
Rescue aircraft arrives to shore
Leave the shore prematurely
AF withdraws with rescue aircraft
Fly towards ship
Rescue aircraft recovered
Recover rescue aircraft
Satisfy
Rescue aircraft approaches ship
Re-embark on the LPD
Amphibious raid ends
- Requirement - 7 - R7 The vessel shall provide emergency rescue capabilities utilizing aerial methods

Figure 9.24: [SAB] Architecture View of Emergency Rescue Scenario

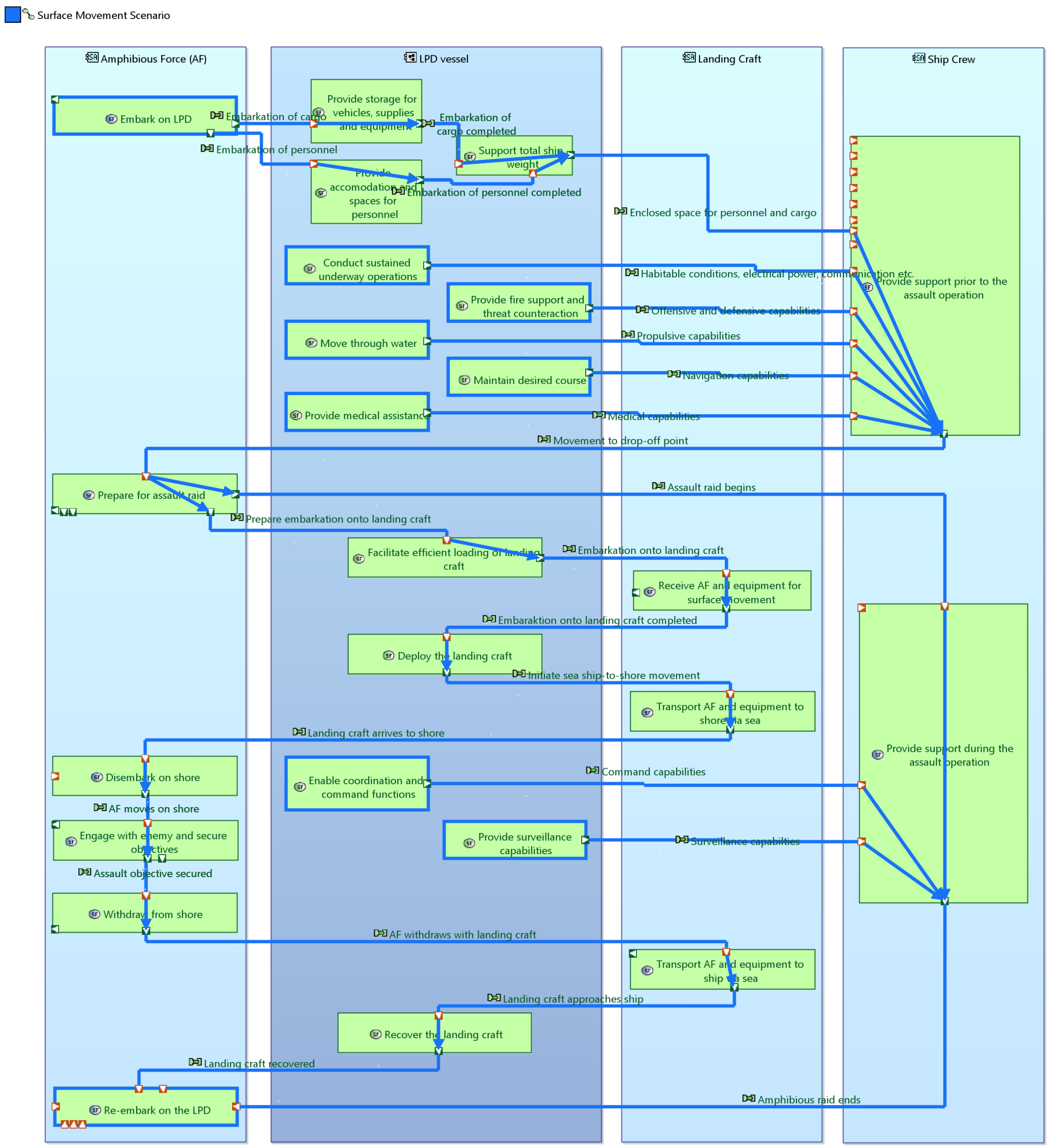

Surface Movement Scenario
Amphibious Force (AF)
LPD vessel
Landing Craft
Ship Crew
Embark on LPD
Embarkation of cargo
Provide storage for vehicles, supplies and equipment
Embarkation of cargo completed
Embarkation of personnel
Support total ship weight
Provide accomodation and spaces for personnel
Embarkation of personnel completed
Enclosed space for personnel and cargo
Conduct sustained underway operations
Habitable conditions, electrical power, communication etc.
Provide support prior to the assault operation
Provide fire support and threat counteraction
Offensive and defensive capabilities
Move through water
Propulsive capabilities
Maintain desired course
Navigation capabilities
Provide medical assistance
Medical capabilities
Movement to drop-off point
Prepare for assault raid
Assault raid begins
Prepare embarkation onto landing craft
Facilitate efficient loading of landing craft
Embarkation onto landing craft
Receive AF and equipment for surface movement
Embaraktion onto landing craft completed
Deploy the landing craft
Initiate sea ship-to-shore movement
Transport AF and equipment to shore via sea
Landing craft arrives to shore
Provide support during the assault operation
Disembark on shore
Enable coordination and command functions
Command capabilities
AF moves on shore
Engage with enemy and secure objectives
Provide surveillance capabilities
Surveillance capabilities
Assault objective secured
Withdraw from shore
AF withdraws with landing craft
Transport AF and equipment to ship via sea
Landing craft approaches ship
Recover the landing craft
Landing craft recovered
Amphibious raid ends
Re-embark on the LPD

Figure 9.25: [SAB] Architecture View of Surface Movement Scenario

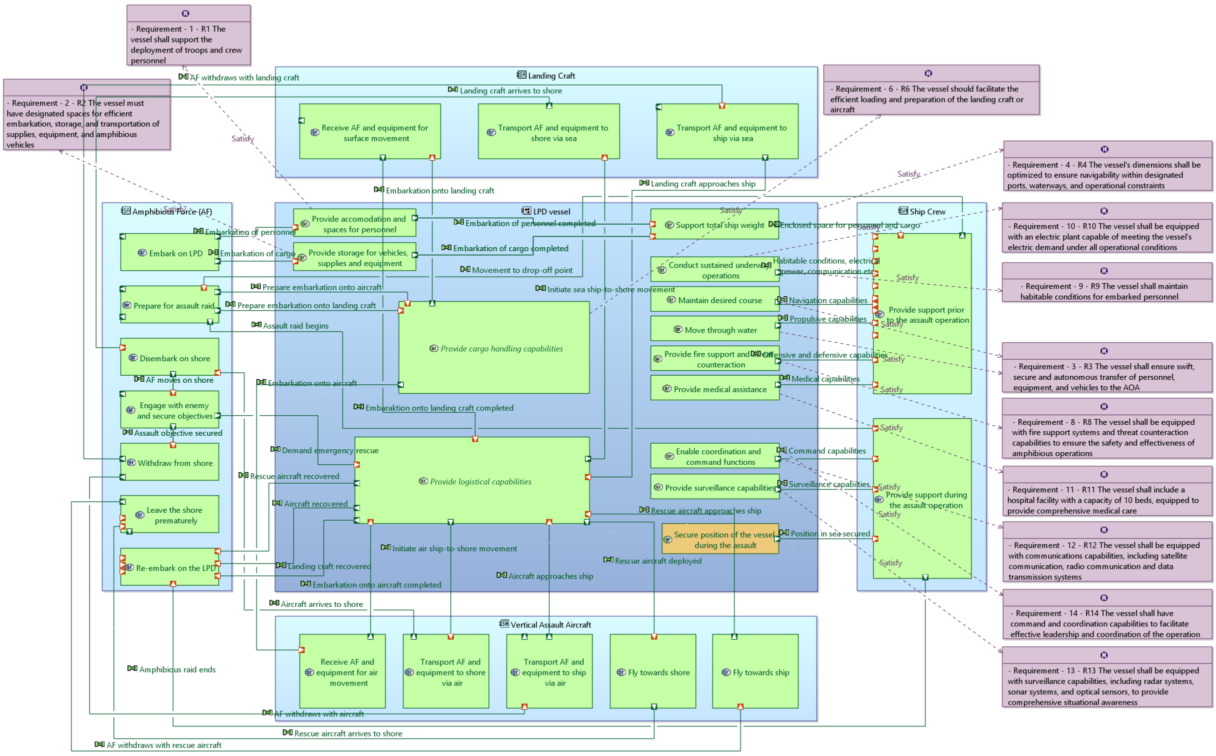

- Requirement - 1 - R1 The vessel shall support the deployment of troops and crew personnel
- Requirement - 2 - R2 The vessel must have designated spaces for efficient embarkation, storage, and transportation of supplies, equipment, and amphibious vehicles
- Requirement - 6 - R6 The vessel should facilitate the efficient loading and preparation of the landing craft or aircraft
- Requirement - 4 - R4 The vessel's dimensions shall be optimized to ensure navigability within designated ports, waterways, and operational constraints
- Requirement - 10 - R10 The vessel shall be equipped with an electric plant capable of meeting the vessel's electric demand under all operational conditions
- Requirement - 9 - R9 The vessel shall maintain habitable conditions for embarked personnel
- Requirement - 3 - R3 The vessel shall ensure swift, secure and autonomous transfer of personnel, equipment, and vehicles to the AOA
- Requirement - 8 - R8 The vessel shall be equipped with fire support systems and threat counteraction capabilities to ensure the safety and effectiveness of amphibious operations
- Requirement - 11 - R11 The vessel shall include a hospital facility with a capacity of 10 beds, equipped to provide comprehensive medical care
- Requirement - 12 - R12 The vessel shall be equipped with communications capabilities, including satellite communication, radio communication and data transmission systems
- Requirement - 14 - R14 The vessel shall have command and coordination capabilities to facilitate effective leadership and coordination of the operation
- Requirement - 13 - R13 The vessel shall be equipped with surveillance capabilities, including radar systems, sonar systems, and optical sensors, to provide comprehensive situational awareness
Satisfy
Landing Craft
AF withdraws with landing craft
Landing craft arrives to shore
Receive AF and equipment for surface movement
Transport AF and equipment to shore via sea
Transport AF and equipment to ship via sea
Embarkation onto landing craft
Landing craft approaches ship
Amphibious Force (AF)
Embark on LPD
Embarkation of personnel
Embarkation of cargo
Prepare for assault raid
Prepare embarkation onto aircraft
Prepare embarkation onto landing craft
Assault raid begins
Disembark on shore
AF moves on shore
Engage with enemy and secure objectives
Assault objective secured
Withdraw from shore
Leave the shore prematurely
Re-embark on the LPD
LPD vessel
Provide accomodation and spaces for personnel
Provide storage for vehicles, supplies and equipment
Embarkation of personnel completed
Embarkation of cargo completed
Movement to drop-off point
Initiate sea ship-to-shore movement
Provide cargo handling capabilities
Embarkation onto aircraft
Embaraktion onto landing craft completed
Demand emergency rescue
Rescue aircraft recovered
Aircraft recovered
Provide logistical capabilities
Initiate air ship-to-shore movement
Landing craft recovered
Embarkation onto aircraft completed
Aircraft approaches ship
Rescue aircraft deployed
Rescue aircraft approaches ship
Support total ship weight
Enclosed space for personnel and cargo
Conduct sustained underway operations
Habitable conditions, electrical power, communication etc
Maintain desired course
Navigation capabilities
Move through water
Propulsive capabilities
Provide fire support and threat counteraction
Offensive and defensive capabilities
Provide medical assistance
Medical capabilities
Enable coordination and command functions
Command capabilities
Provide surveillance capabilities
Surveillance capabilities
Secure position of the vessel during the assault
Position in sea secured
Ship Crew
Provide support prior to the assault operation
Provide support during the assault operation
Aircraft arrives to shore
Vertical Assault Aircraft
Receive AF and equipment for air movement
Transport AF and equipment to shore via air
Transport AF and equipment to ship via air
Fly towards shore
Fly towards ship
Amphibious raid ends
AF withdraws with aircraft
Rescue aircraft arrives to shore
AF withdraws with rescue aircraft

Figure 9.26: [SAB] LPD System

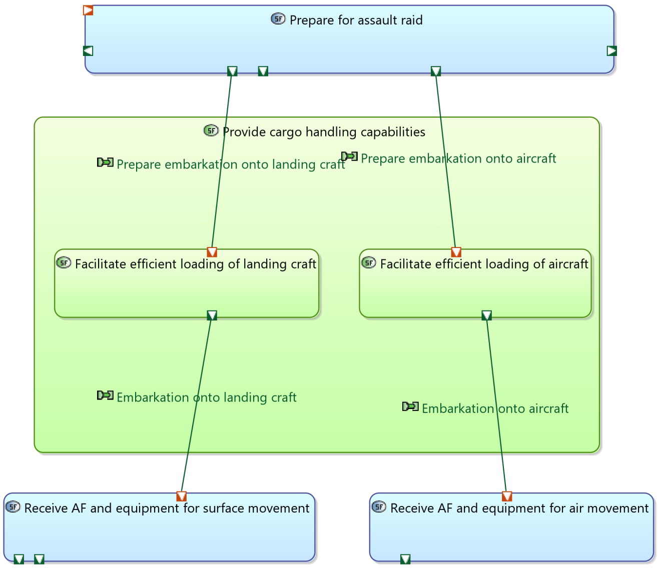

Prepare for assault raid
Provide cargo handling capabilities
Prepare embarkation onto landing craft
Prepare embarkation onto aircraft
Facilitate efficient loading of landing craft
Facilitate efficient loading of aircraft
Embarkation onto landing craft
Embarkation onto aircraft
Receive AF and equipment for surface movement
Receive AF and equipment for air movement

Figure 9.27: [SDFB] Provide cargo handling capabilities

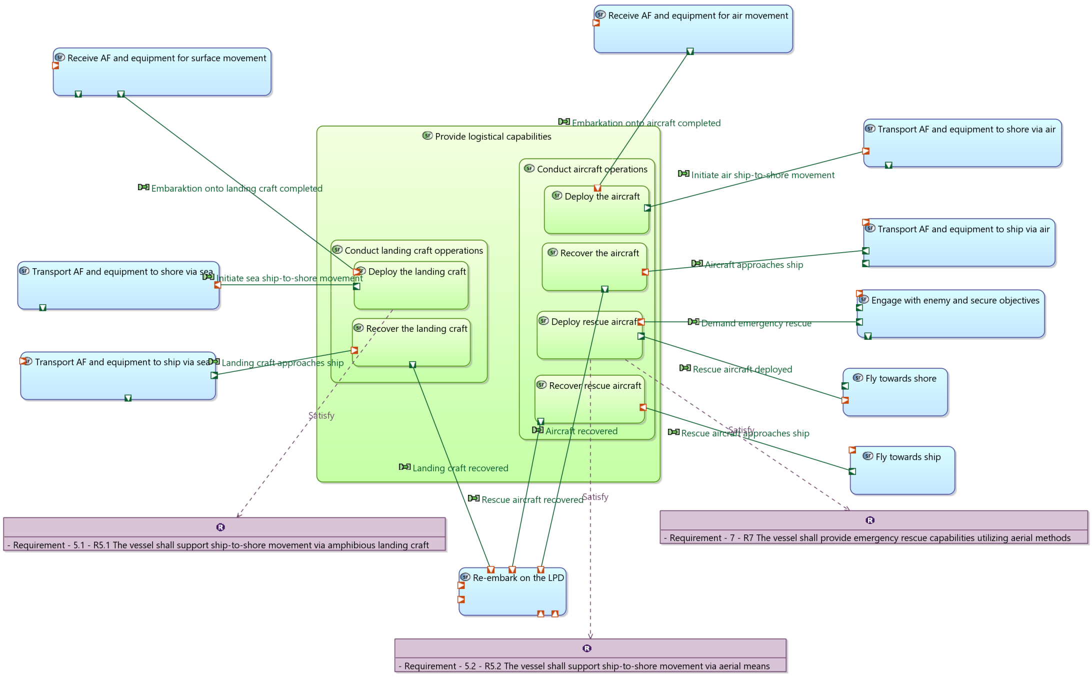

Receive AF and equipment for air movement
Receive AF and equipment for surface movement
Embarkation onto aircraft completed
Provide logistical capabilities
Transport AF and equipment to shore via air
Conduct aircraft operations
Initiate air ship-to-shore movement
Embaraktion onto landing craft completed
Deploy the aircraft
Transport AF and equipment to ship via air
Recover the aircraft
Conduct landing craft opperations
Aircraft approaches ship
Transport AF and equipment to shore via sea
Deploy the landing craft
Initiate sea ship-to-shore movement
Engage with enemy and secure objectives
Recover the landing craft
Deploy rescue aircraft
Demand emergency rescue
Transport AF and equipment to ship via sea
Landing craft approaches ship
Rescue aircraft deployed
Fly towards shore
Satisfy
Recover rescue aircraft
Aircraft recovered
Rescue aircraft approaches ship
Satisfy
Fly towards ship
Landing craft recovered
Rescue aircraft recovered
Satisfy
R
- Requirement - 5.1 - R5.1 The vessel shall support ship-to-shore movement via amphibious landing craft
R
- Requirement - 7 - R7 The vessel shall provide emergency rescue capabilities utilizing aerial methods
Re-embark on the LPD
R
- Requirement - 5.2 - R5.2 The vessel shall support ship-to-shore movement via aerial means

Figure 9.28: [SDFB] Provide logistical capabilities

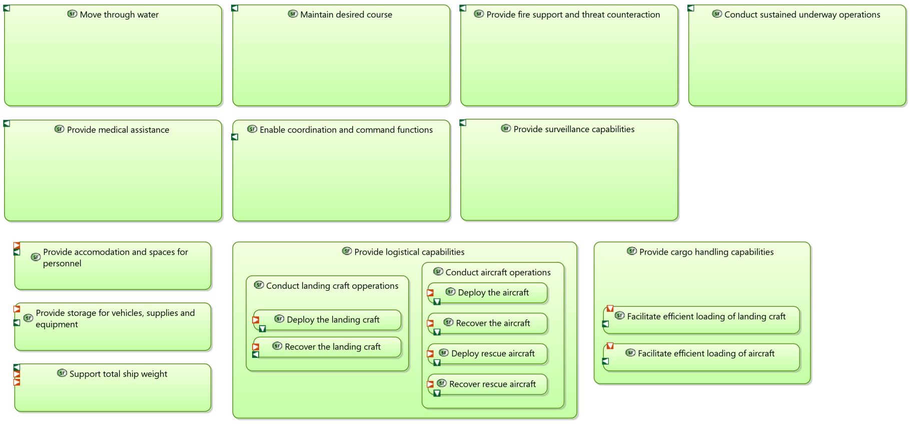

Move through water
Maintain desired course
Provide fire support and threat counteraction
Conduct sustained underway operations
Provide medical assistance
Enable coordination and command functions
Provide surveillance capabilities
Provide accomodation and spaces for personnel
Provide logistical capabilities
Provide cargo handling capabilities
Conduct aircraft operations
Deploy the aircraft
Conduct landing craft opperations
Recover the aircraft
Facilitate efficient loading of landing craft
Provide storage for vehicles, supplies and equipment
Deploy the landing craft
Recover the landing craft
Deploy rescue aircraft
Facilitate efficient loading of aircraft
Support total ship weight
Recover rescue aircraft

Figure 9.29: [SDFB] Root System Function

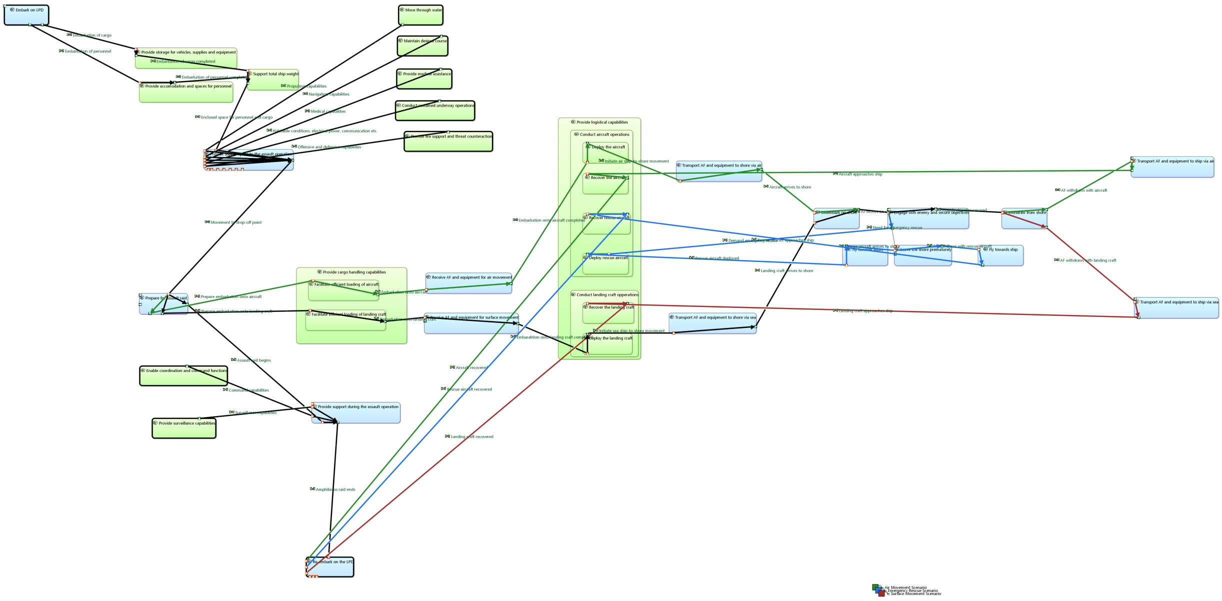

Embark on LPD
Provide storage for vehicles, supplies and equipment
Provide accommodation and spaces for personnel
Support total ship weight
Move through water
Maintain desired course
Provide medical assistance
Conduct sustained underway operations
Provide fire support and threat counteraction
Provide logistical capabilities
Conduct aircraft operations
Deploy the aircraft
Recover the aircraft
Deploy rescue aircraft
Conduct landing craft operations
Recover the landing craft
Deploy the landing craft
Provide cargo handling capabilities
Facilitate efficient loading of aircraft
Facilitate efficient loading of landing craft
Receive AF and equipment for air movement
Receive AF and equipment for surface movement
Transport AF and equipment to shore via air
Transport AF and equipment to ship via air
Transport AF and equipment to shore via sea
Transport AF and equipment to ship via sea
Fly towards ship
Leave the shore prematurely
Enable coordination and command functions
Provide surveillance capabilities
Provide support during the assault operation
Re-embark on the LPD
Air Movement Scenario
Emergency Rescue Scenario
Surface Movement Scenario

Figure 9.30: [SDFB] Root System Function - Chain

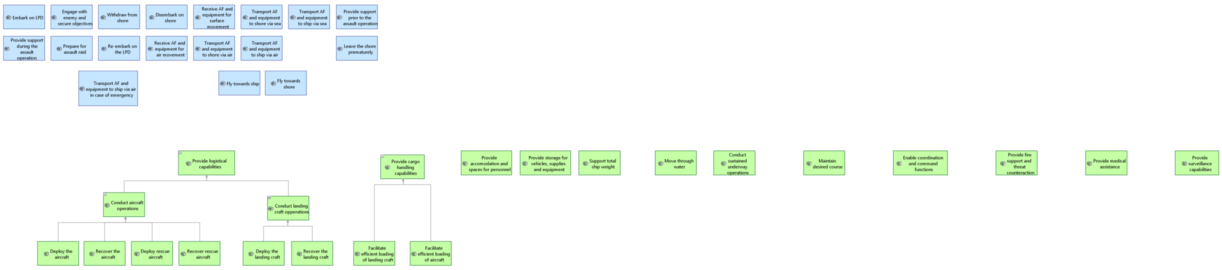

Embark on LPD
Engage with enemy and secure objectives
Withdraw from shore
Disembark on shore
Receive AF and equipment for surface movement
Transport AF and equipment to shore via sea
Transport AF and equipment to ship via sea
Provide support prior to the assault operation
Provide support during the assault operation
Prepare for assault raid
Re-embark on the LPD
Receive AF and equipment for air movement
Transport AF and equipment to shore via air
Transport AF and equipment to ship via air
Leave the shore prematurely
Transport AF and equipment to ship via air in case of emergency
Fly towards ship
Fly towards shore
Provide logistical capabilities
Conduct aircraft operations
Conduct landing craft opperations
Deploy the aircraft
Recover the aircraft
Deploy rescue aircraft
Recover rescue aircraft
Deploy the landing craft
Recover the landing craft
Provide cargo handling capabilities
Facilitate efficient loading of landing craft
Facilitate efficient loading of aircraft
Provide accomodation and spaces for personnel
Provide storage for vehicles, supplies and equipment
Support total ship weight
Move through water
Conduct sustained underway operations
Maintain desired course
Enable coordination and command functions
Provide fire support and threat counteraction
Provide medical assistance
Provide surveillance capabilities

Figure 9.31: [SFBD] Root System Function

Diagrams of Logical Architecture

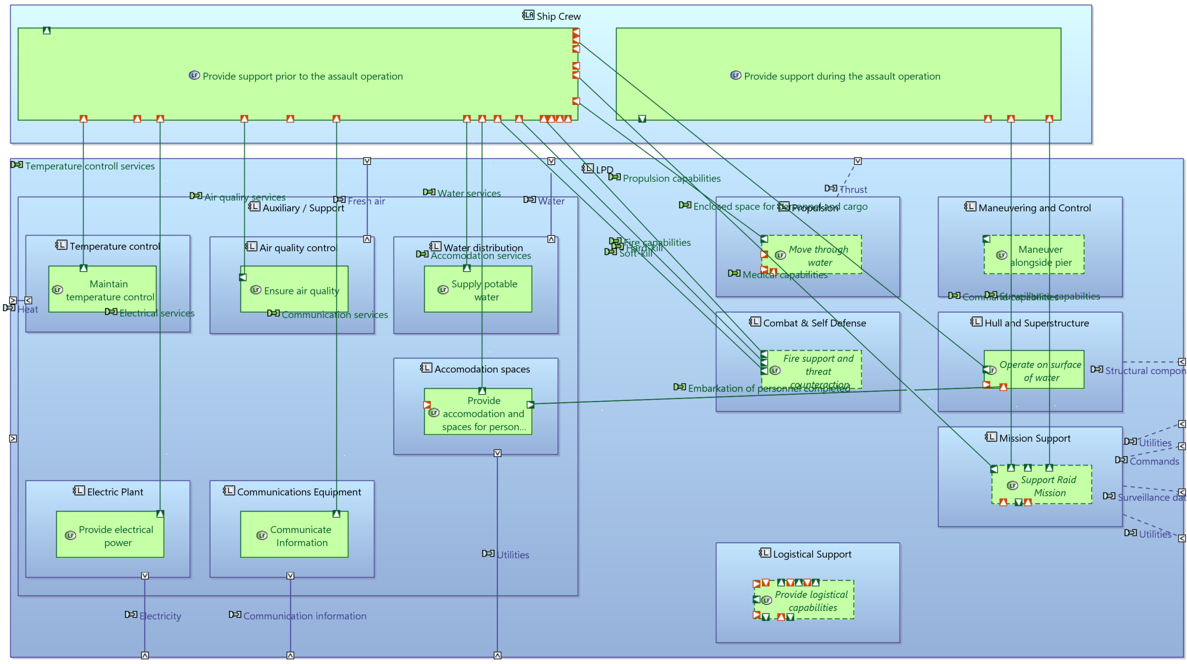


Figure 9.32: [LAB] Auxiliary / Support

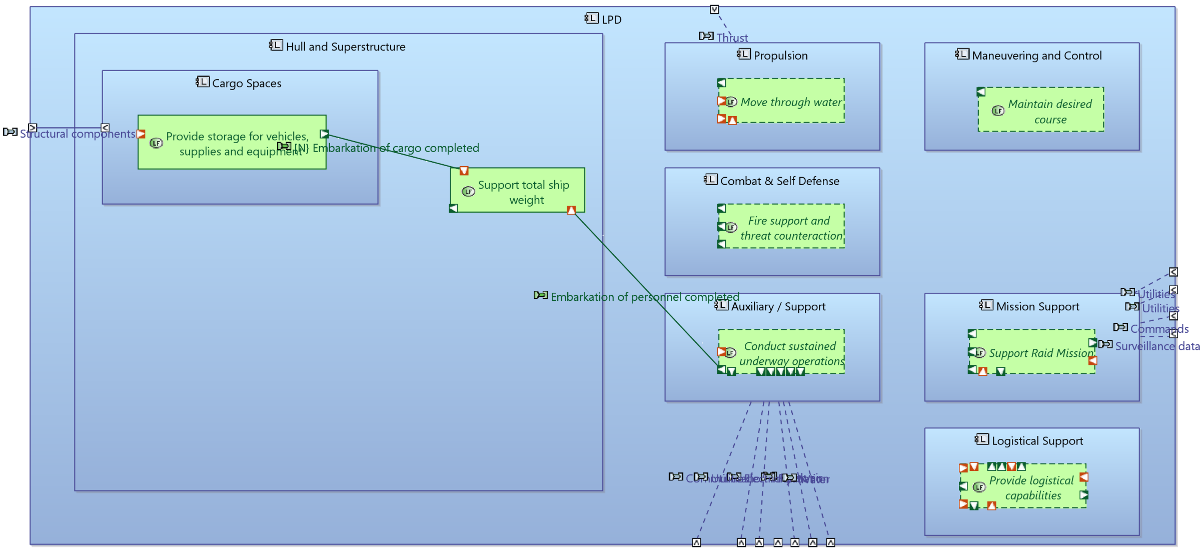


Figure 9.33: [LAB] Hull and Superstructure

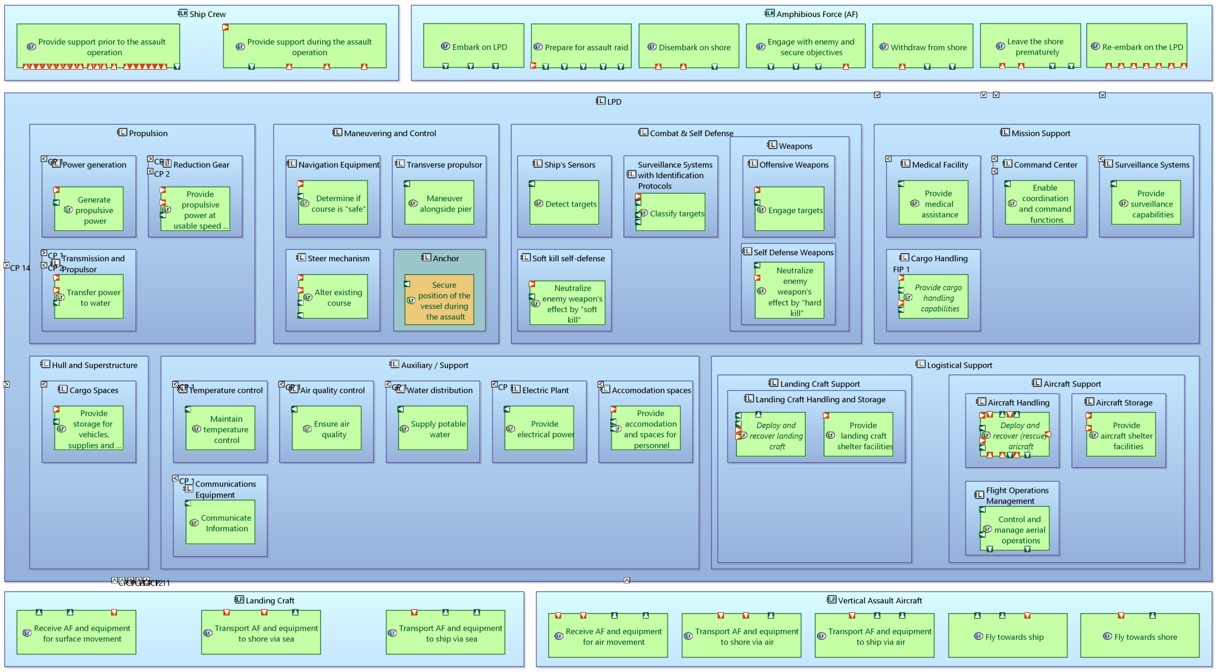

Ship Crew
Provide support prior to the assault operation
Provide support during the assault operation
Amphibious Force (AF)
Embark on LPD
Prepare for assault raid
Disembark on shore
Engage with enemy and secure objectives
Withdraw from shore
Leave the shore prematurely
Re-embark on the LPD
LPD
Propulsion
Power generation
Generate propulsive power
Reduction Gear
Provide propulsive power at usable speed ...
Transmission and Propulsor
Transfer power to water
Maneuvering and Control
Navigation Equipment
Determine if course is "safe"
Transverse propulsor
Maneuver alongside pier
Steer mechanism
Alter existing course
Anchor
Secure position of the vessel during the assault
Combat & Self Defense
Ship's Sensors
Detect targets
Surveillance Systems with Identification Protocols
Classify targets
Weapons
Offensive Weapons
Engage targets
Self Defense Weapons
Neutralize enemy weapon's effect by "hard kill"
Soft kill self-defense
Neutralize enemy weapon's effect by "soft kill"
Mission Support
Medical Facility
Provide medical assistance
Command Center
Enable coordination and command functions
Surveillance Systems
Provide surveillance capabilities
Cargo Handling
Provide cargo handling capabilities
Hull and Superstructure
Cargo Spaces
Provide storage for vehicles, supplies and ...
Auxiliary / Support
Temperature control
Maintain temperature control
Air quality control
Ensure air quality
Water distribution
Supply potable water
Electric Plant
Provide electrical power
Accomodation spaces
Provide accomodation and spaces for personnel
Communications Equipment
Communicate Information
Logistical Support
Landing Craft Support
Landing Craft Handling and Storage
Deploy and recover landing craft
Provide landing craft shelter facilities
Aircraft Support
Aircraft Handling
Deploy and recover (rescue) aircraft
Aircraft Storage
Provide aircraft shelter facilities
Flight Operations Management
Control and manage aerial operations
CP 14
CP 1
CP 2
FIP 1
Landing Craft
Receive AF and equipment for surface movement
Transport AF and equipment to shore via sea
Transport AF and equipment to ship via sea
Vertical Assault Aircraft
Receive AF and equipment for air movement
Transport AF and equipment to shore via air
Transport AF and equipment to ship via air
Fly towards ship
Fly towards shore

Figure 9.34: [LAB] Logical Architecture Final

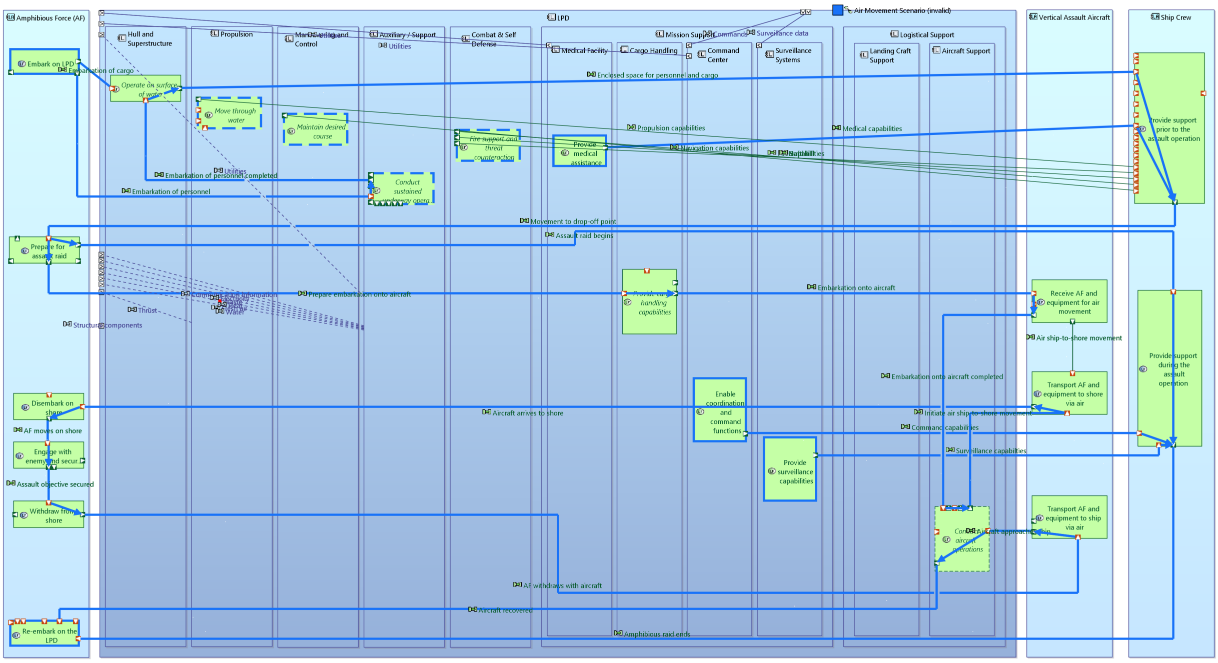

Air Movement Scenario (invalid)
Amphibious Force (AF)
LPD
Vertical Assault Aircraft
Ship Crew
Hull and Superstructure
Propulsion
Maneuvering and Control
Auxiliary / Support
Utilities
Combat & Self Defense
Mission Support
Commands
Surveillance data
Medical Facility
Cargo Handling
Command Center
Surveillance Systems
Logistical Support
Landing Craft Support
Aircraft Support
Embark on LPD
Embarkation of cargo
Operate on surface of water
Enclosed space for personnel and cargo
Move through water
Maintain desired course
Fire support and threat counteraction
Provide medical assistance
Propulsion capabilities
Medical capabilities
Navigation capabilities
Provide support prior to the assault operation
Embarkation of personnel completed
Embarkation of personnel
Conduct sustained underway opera...
Movement to drop-off point
Assault raid begins
Prepare for assault raid
Communication information
Electricity
Waste
Heat
Fresh air
Water
Thrust
Structural components
Prepare embarkation onto aircraft
Provide cargo handling capabilities
Embarkation onto aircraft
Receive AF and equipment for air movement
Air ship-to-shore movement
Embarkation onto aircraft completed
Transport AF and equipment to shore via air
Provide support during the assault operation
Disembark on shore
Aircraft arrives to shore
Enable coordination and command functions
Initiate air ship-to-shore movement
Command capabilities
AF moves on shore
Engage with enemy and secur...
Provide surveillance capabilities
Surveillance capabilities
Assault objective secured
Withdraw from shore
Control aircraft operations
Aircraft approach ship
Transport AF and equipment to ship via air
AF withdraws with aircraft
Aircraft recovered
Re-embark on the LPD
Amphibious raid ends

Figure 9.35: [LAB] Logical Architecture View of Air Movement Scenario

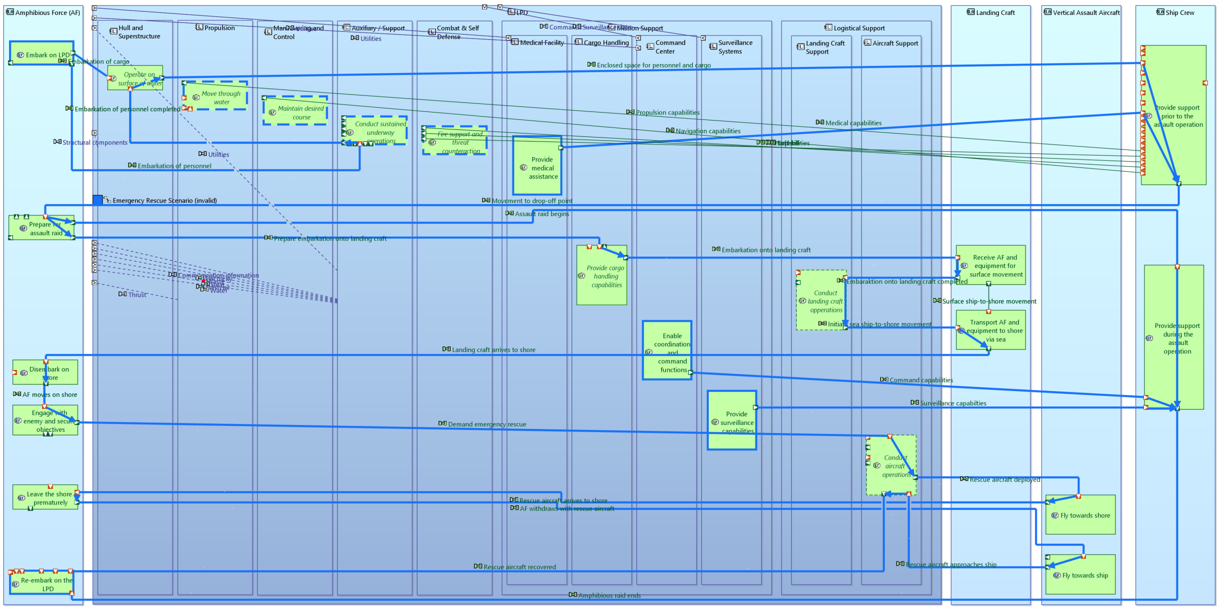

Amphibious Force (AF)
LPD
Hull and Superstructure
Propulsion
Maneuvering and Control
Auxiliary / Support
Combat & Self Defense
Mission Support
Medical Facility
Cargo Handling
Command Center
Surveillance Systems
Logistical Support
Landing Craft Support
Aircraft Support
Landing Craft
Vertical Assault Aircraft
Ship Crew
Embark on LPD
Embarkation of cargo
Operate on surface of water
Move through water
Maintain desired course
Conduct sustained underway operations
Fire support and threat counteraction
Provide medical assistance
Embarkation of personnel completed
Structural components
Utilities
Embarkation of personnel
Enclosed space for personnel and cargo
Propulsion capabilities
Medical capabilities
Navigation capabilities
Provide support prior to the assault operation
Emergency Rescue Scenario (invalid)
Movement to drop-off point
Assault raid begins
Prepare for assault raid
Prepare embarkation onto landing craft
Provide cargo handling capabilities
Embarkation onto landing craft
Receive AF and equipment for surface movement
Embaraktion onto landing craft completed
Conduct landing craft opperations
Surface ship-to-shore movement
Initiate sea ship-to-shore movement
Transport AF and equipment to shore via sea
Provide support during the assault operation
Communication information
Electricity
Waste
Heat
Fresh air
Water
Thrust
Landing craft arrives to shore
Enable coordination and command functions
Disembark on shore
AF moves on shore
Engage with enemy and secure objectives
Command capabilities
Surveillance capabilties
Provide surveillance capabilities
Demand emergency rescue
Conduct aircraft operations
Rescue aircraft deployed
Leave the shore prematurely
Rescue aircraft arrives to shore
AF withdraws with rescue aircraft
Fly towards shore
Rescue aircraft recovered
Rescue aircraft approaches ship
Fly towards ship
Re-embark on the LPD
Amphibious raid ends

Figure 9.36: [LAB] Logical Architecture View of Emergency Rescue Scenario

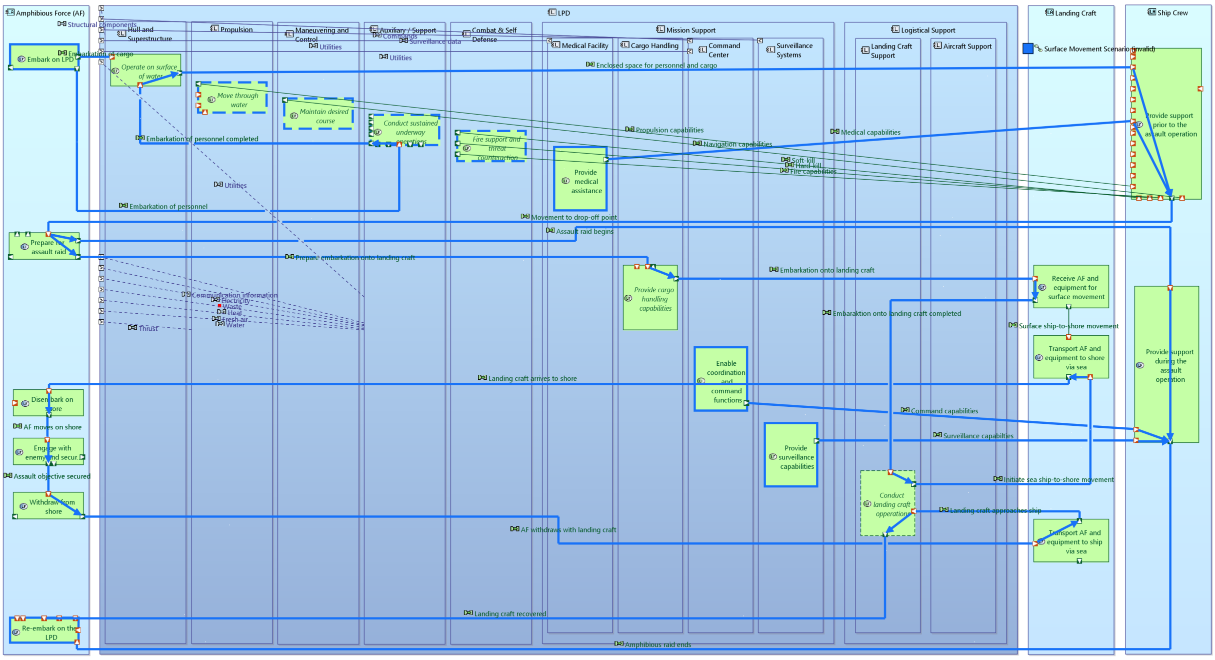

Amphibious Force (AF)
Structural components
LPD
Hull and Superstructure
Propulsion
Maneuvering and Control
Auxiliary / Support
Commands
Surveillance data
Combat & Self Defense
Utilities
Mission Support
Medical Facility
Cargo Handling
Command Center
Surveillance Systems
Logistical Support
Landing Craft Support
Aircraft Support
Landing Craft
Ship Crew
Surface Movement Scenario (invalid)
Embark on LPD
Embarkation of cargo
Operate on surface of water
Move through water
Maintain desired course
Conduct sustained underway operations
Fire support and threat counteraction
Provide medical assistance
Enclosed space for personnel and cargo
Propulsion capabilities
Navigation capabilities
Medical capabilities
Soft-kill
Hard-kill
Fire capabilities
Provide support prior to the assault operation
Embarkation of personnel completed
Utilities
Embarkation of personnel
Movement to drop-off point
Assault raid begins
Prepare for assault raid
Prepare embarkation onto landing craft
Embarkation onto landing craft
Provide cargo handling capabilities
Receive AF and equipment for surface movement
Embaraktion onto landing craft completed
Surface ship-to-shore movement
Communication information
Electricity
Waste
Heat
Fresh air
Water
Thrust
Transport AF and equipment to shore via sea
Provide support during the assault operation
Landing craft arrives to shore
Enable coordination and command functions
Disembark on shore
AF moves on shore
Command capabilities
Surveillance capabilties
Engage with enemy and secur...
Provide surveillance capabilities
Assault objective secured
Initiate sea ship-to-shore movement
Withdraw from shore
Conduct landing craft opperations
Landing craft approaches ship
AF withdraws with landing craft
Transport AF and equipment to ship via sea
Landing craft recovered
Re-embark on the LPD
Amphibious raid ends

Figure 9.37: [LAB] Logical Architecture View of Surface Movement Scenario

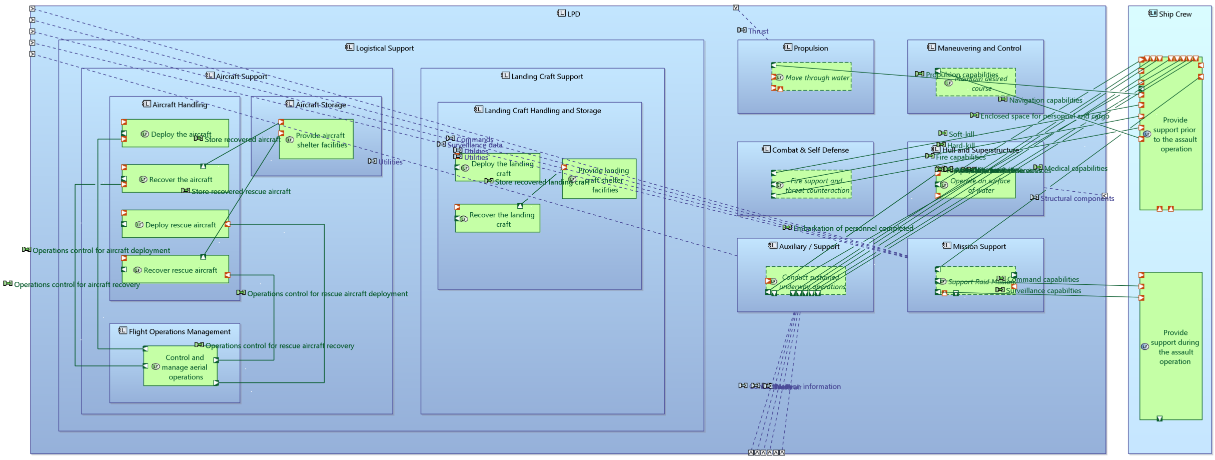
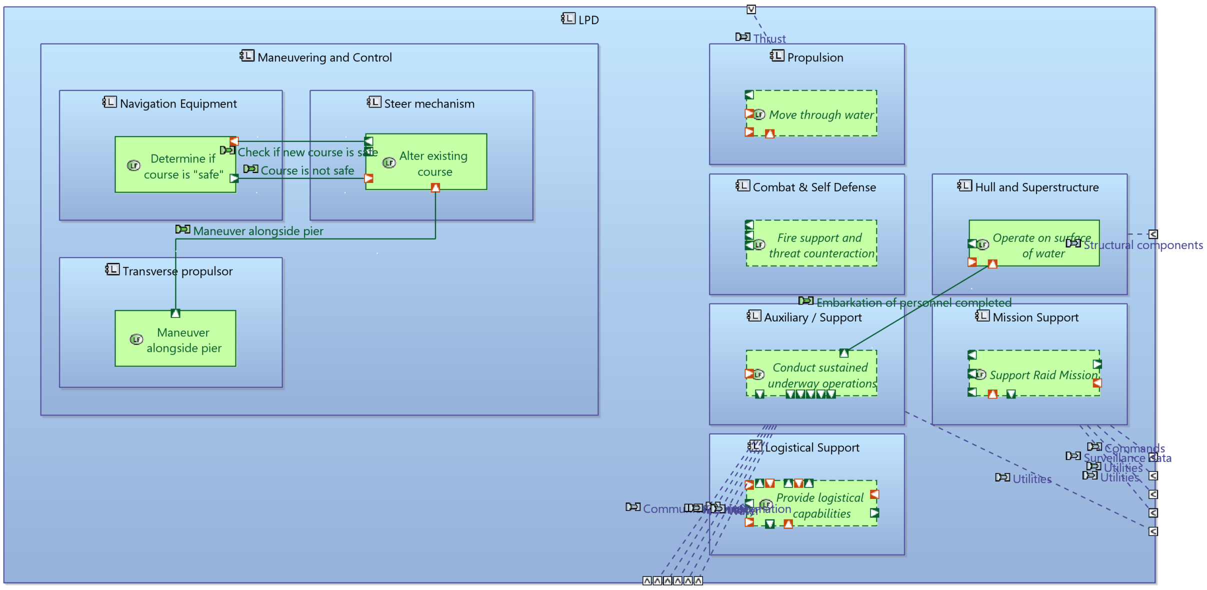

LPD
Logistical Support
Aircraft Support
Aircraft Handling
Aircraft Storage
Deploy the aircraft
Store recovered aircraft
Provide aircraft shelter facilities
Recover the aircraft
Store recovered rescue aircraft
Deploy rescue aircraft
Recover rescue aircraft
Operations control for aircraft deployment
Operations control for aircraft recovery
Operations control for rescue aircraft deployment
Flight Operations Management
Operations control for rescue aircraft recovery
Control and manage aerial operations
Landing Craft Support
Landing Craft Handling and Storage
Commands
Surveillance data
Utilities
Deploy the landing craft
Store recovered landing craft
Provide landing craft shelter facilities
Recover the landing craft
Thrust
Propulsion
Move through water
Maneuvering and Control
Propulsion capabilities
Maintain desired course
Navigation capabilities
Enclosed space for personnel and cargo
Combat & Self Defense
Soft-kill
Hard-kill
Hull and Superstructure
Fire capabilities
Fire support and threat counteraction
Operate on surface of water
Medical capabilities
Structural components
Embarkation of personnel completed
Auxiliary / Support
Mission Support
Conduct sustained underway operations
Support Raid Mission
Command capabilities
Surveillance capabilities
Mission information
Ship Crew
Provide support prior to the assault operation
Provide support during the assault operation

Figure 9.38: [LAB] Logistical Support Systems



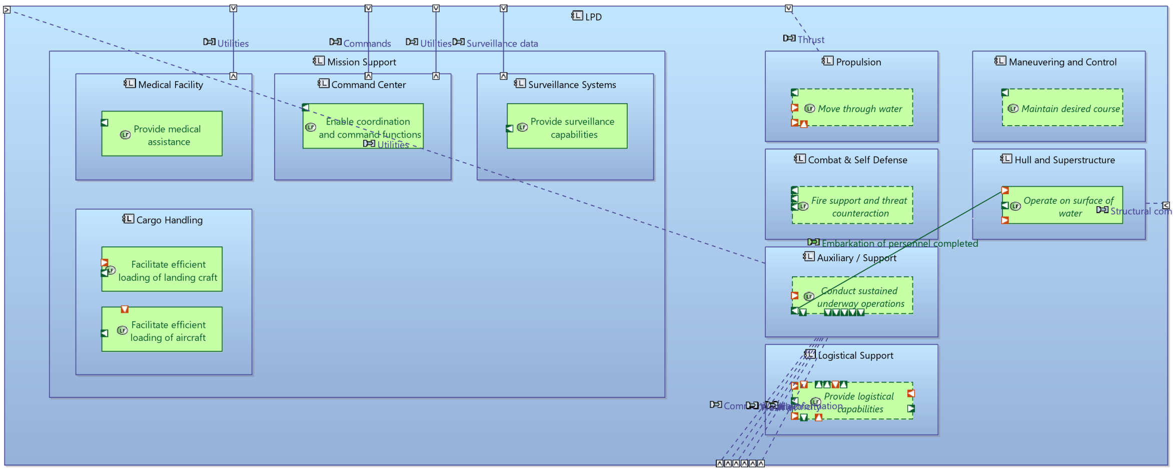

LPD
Utilities
Commands
Utilities
Surveillance data
Mission Support
Medical Facility
Provide medical assistance
Command Center
Enable coordination and command functions
Utilities
Surveillance Systems
Provide surveillance capabilities
Cargo Handling
Facilitate efficient loading of landing craft
Facilitate efficient loading of aircraft
Thrust
Propulsion
Move through water
Maneuvering and Control
Maintain desired course
Combat & Self Defense
Fire support and threat counteraction
Hull and Superstructure
Operate on surface of water
Structural components
Embarkation of personnel completed
Auxiliary / Support
Conduct sustained underway operations
Logistical Support
Provide logistical capabilities
Communication
Information

Figure 9.39: [LAB] Mission Systems

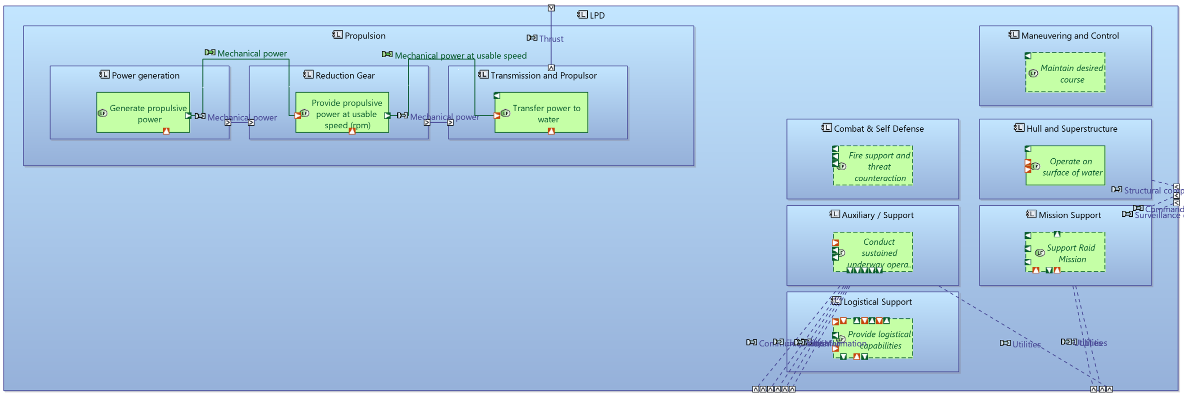


Figure 9.40: [LAB] Propulsion

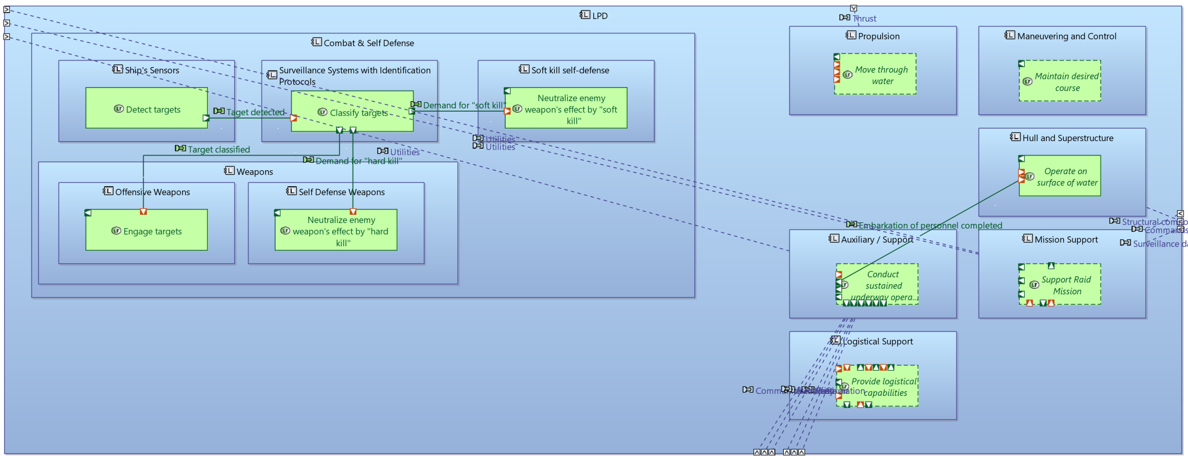


Figure 9.41: [LAB] Weapons

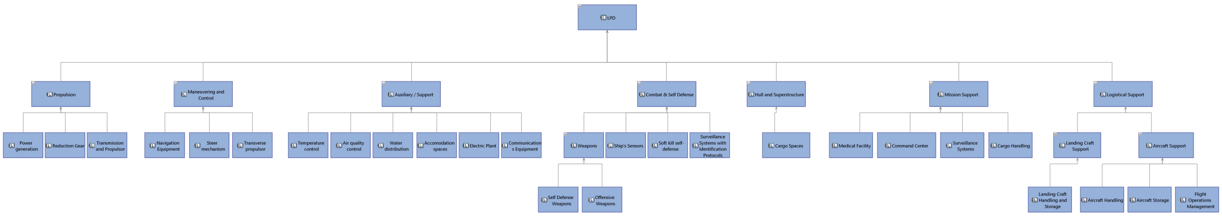


Figure 9.42: [LCBD] Structure

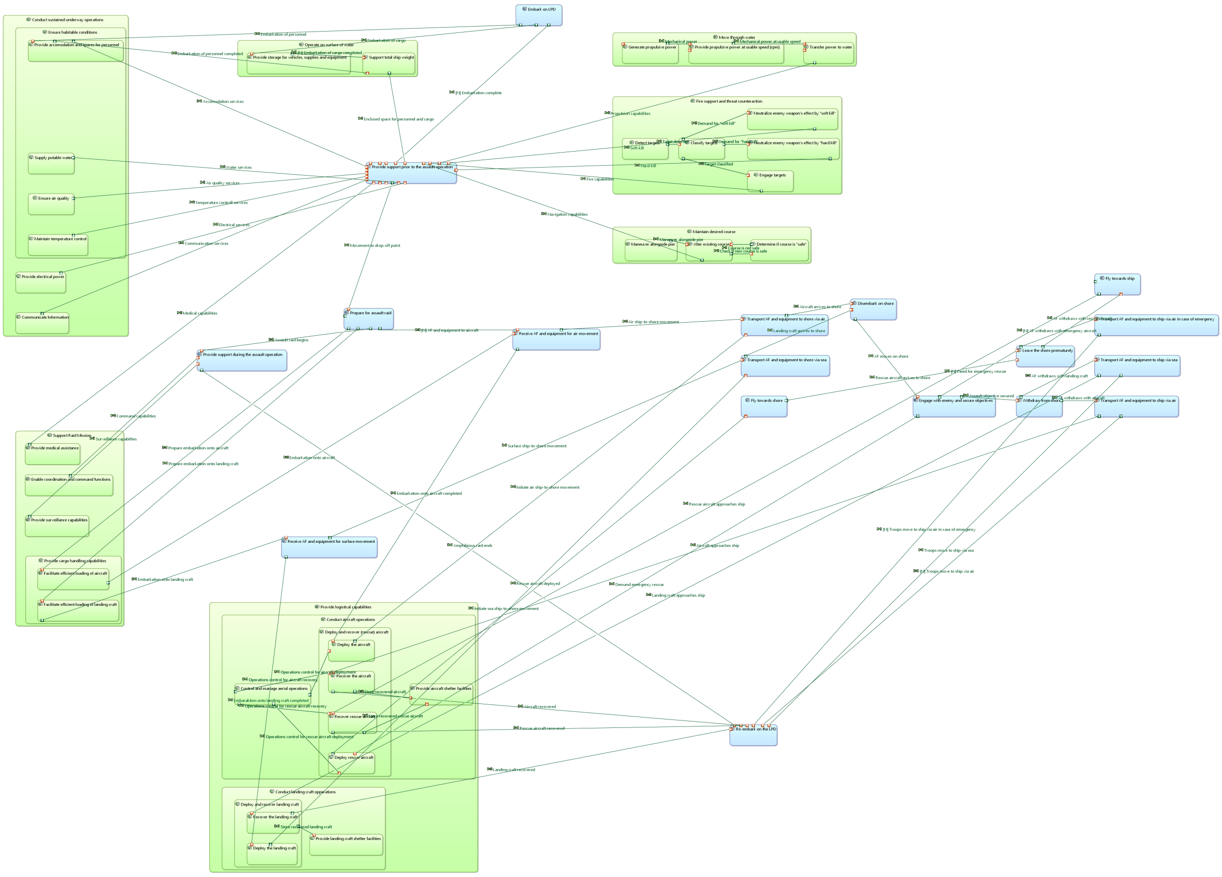


Figure 9.43: [LDFB] Root Logical Function

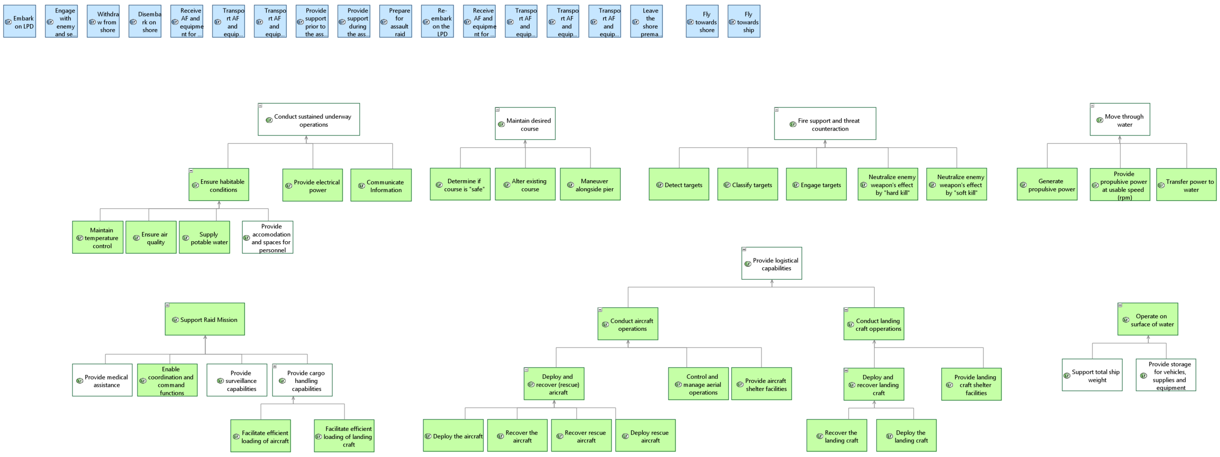

Embark on LPD
Engage with enemy and se...
Withdraw from shore
Disembark on shore
Receive AF and equipment for...
Transport AF and equip...
Transport AF and equip...
Provide support prior to the ass...
Provide support during the ass...
Prepare for assault raid
Re-embark on the LPD
Receive AF and equipment for...
Transport AF and equip...
Transport AF and equip...
Transport AF and equip...
Leave the shore prema...
Fly towards shore
Fly towards ship
Conduct sustained underway operations
Ensure habitable conditions
Provide electrical power
Communicate Information
Maintain temperature control
Ensure air quality
Supply potable water
Provide accomodation and spaces for personnel
Maintain desired course
Determine if course is "safe"
Alter existing course
Maneuver alongside pier
Fire support and threat counteraction
Detect targets
Classify targets
Engage targets
Neutralize enemy weapon's effect by "hard kill"
Neutralize enemy weapon's effect by "soft kill"
Move through water
Generate propulsive power
Provide propulsive power at usable speed (rpm)
Transfer power to water
Provide logistical capabilities
Support Raid Mission
Provide medical assistance
Enable coordination and command functions
Provide surveillance capabilities
Provide cargo handling capabilities
Facilitate efficient loading of aircraft
Facilitate efficient loading of landing craft
Conduct aircraft operations
Deploy and recover (rescue) aricraft
Control and manage aerial operations
Provide aircraft shelter facilities
Deploy the aircraft
Recover the aircraft
Recover rescue aircraft
Deploy rescue aircraft
Conduct landing craft operations
Deploy and recover landing craft
Provide landing craft shelter facilities
Recover the landing craft
Deploy the landing craft
Operate on surface of water
Support total ship weight
Provide storage for vehicles, supplies and equipment

Figure 9.44: [LFBD] Root Logical Function

Diagrams of Physical Architecture

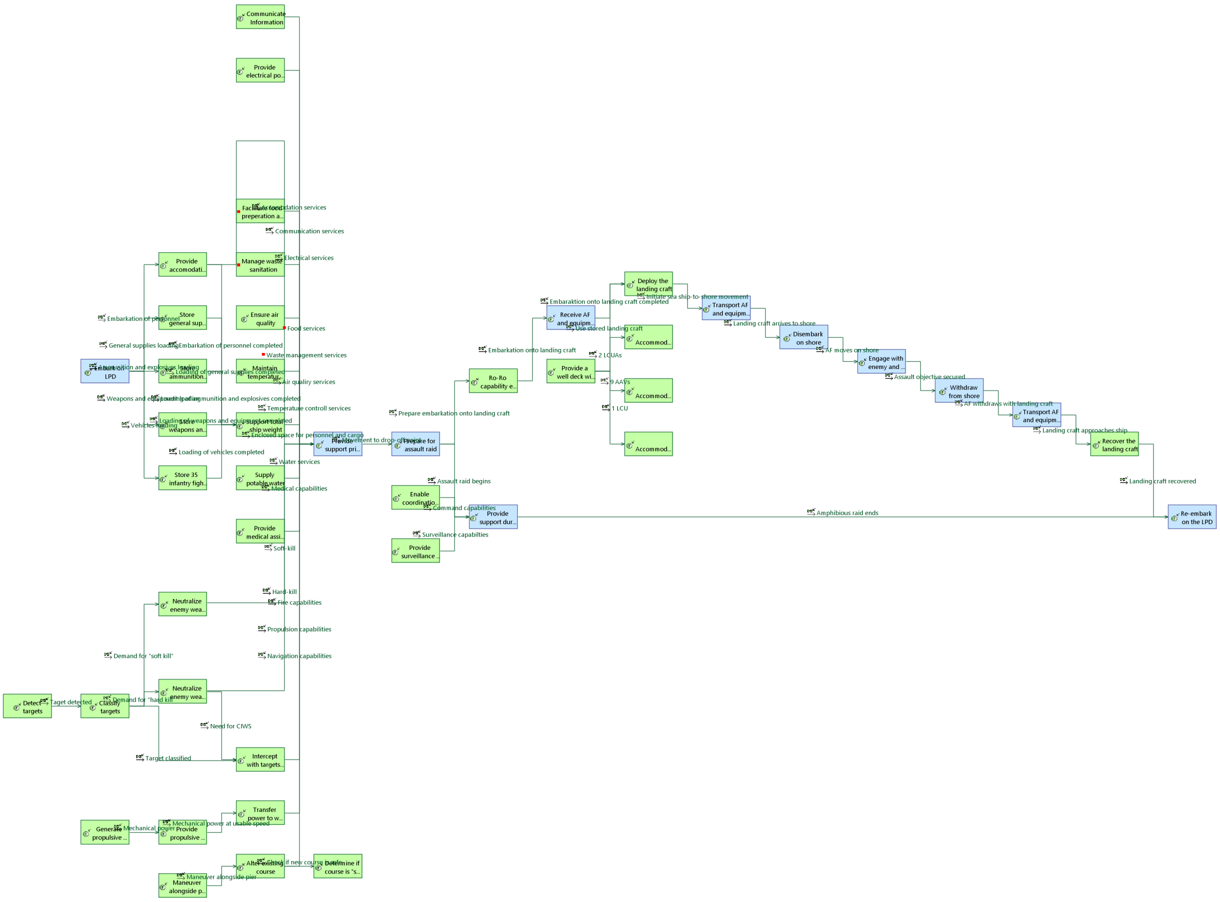


Figure 9.45: [PFCD] Surface movement new

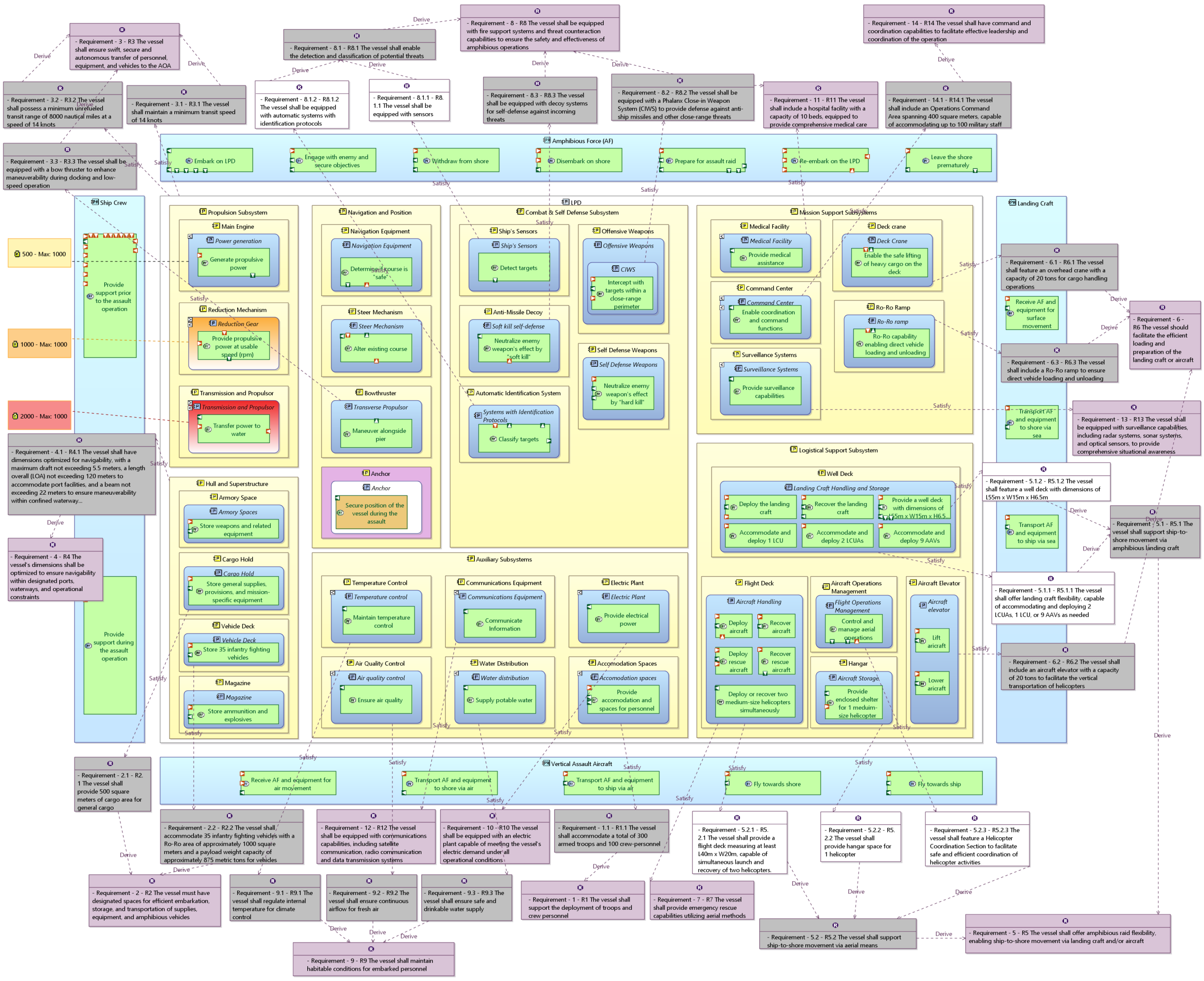

- Requirement - 3 - R3 The vessel shall ensure swift, secure and autonomous transfer of personnel, equipment, and vehicles to the AOA
- Requirement - 8.1 - R8.1 The vessel shall enable the detection and classification of potential threats
- Requirement - 8 - R8 The vessel shall be equipped with fire support systems and threat counteraction capabilities to ensure the safety and effectiveness of amphibious operations
- Requirement - 14 - R14 The vessel shall have command and coordination capabilities to facilitate effective leadership and coordination of the operation
- Requirement - 3.2 - R3.2 The vessel shall possess a minimum unrefueled transit range of 8000 nautical miles at a speed of 14 knots
- Requirement - 3.1 - R3.1 The vessel shall maintain a minimum transit speed of 14 knots
- Requirement - 8.1.2 - R8.1.2 The vessel shall be equipped with automatic systems with identification protocols
- Requirement - 8.1.1 - R8.1.1 The vessel shall be equipped with sensors
- Requirement - 8.3 - R8.3 The vessel shall be equipped with decoy systems for self-defense against incoming threats
- Requirement - 8.2 - R8.2 The vessel shall be equipped with a Phalanx Close-in Weapon System (CIWS) to provide defense against anti-ship missiles and other close-range threats
- Requirement - 11 - R11 The vessel shall include a hospital facility with a capacity of 10 beds, equipped to provide comprehensive medical care
- Requirement - 14.1 - R14.1 The vessel shall include an Operations Command Area spanning 400 square meters, capable of accommodating up to 100 military staff
- Requirement - 3.3 - R3.3 The vessel shall be equipped with a bow thruster to enhance maneuverability during docking and low-speed operation
Derive
Satisfy
Amphibious Force (AF)
Embark on LPD
Engage with enemy and secure objectives
Withdraw from shore
Disembark on shore
Prepare for assault raid
Re-embark on the LPD
Leave the shore prematurely
Ship Crew
Provide support prior to the assault operation
Provide support during the assault operation
500 - Max: 1000
1000 - Max: 1000
2000 - Max: 1000
LPD
Propulsion Subsystem
Main Engine
Power generation
Generate propulsive power
Reduction Mechanism
Reduction Gear
Provide propulsive power at usable speed (rpm)
Transmission and Propulsor
Transmission and Propulsor
Transfer power to water
Navigation and Position
Navigation Equipment
Navigation Equipment
Determine if course is "safe"
Steer Mechanism
Steer Mechanism
Alter existing course
Bowthruster
Transverse Propulsor
Maneuver alongside pier
Anchor
Anchor
Secure position of the vessel during the assault
Combat & Self Defense Subsystem
Ship's Sensors
Ship's Sensors
Detect targets
Offensive Weapons
Offensive Weapons
CIWS
Intercept with targets within a close-range perimeter
Anti-Missile Decoy
Soft kill self-defense
Neutralize enemy weapon's effect by "soft kill"
Self Defense Weapons
Self Defense Weapons
Neutralize enemy weapon's effect by "hard kill"
Automatic Identification System
Systems with Identification Protocols
Classify targets
Mission Support Subsystems
Medical Facility
Medical Facility
Provide medical assistance
Deck crane
Deck Crane
Enable the safe lifting of heavy cargo on the deck
Command Center
Command Center
Enable coordination and command functions
Ro-Ro Ramp
Ro-Ro ramp
Ro-Ro capability enabling direct vehicle loading and unloading
Surveillance Systems
Surveillance Systems
Provide surveillance capabilities
Hull and Superstructure
Armory Space
Armory Spaces
Store weapons and related equipment
Cargo Hold
Cargo Hold
Store general supplies, provisions, and mission-specific equipment
Vehicle Deck
Vehicle Deck
Store 35 infantry fighting vehicles
Magazine
Magazine
Store ammunition and explosives
Logistical Support Subsystem
Well Deck
Landing Craft Handling and Storage
Deploy the landing craft
Recover the landing craft
Provide a well deck with dimensions of L55m x W15m x H6.5m
Accommodate and deploy 1 LCU
Accommodate and deploy 2 LCUAs
Accommodate and deploy 9 AAVs
Flight Deck
Aircraft Handling
Deploy aircraft
Recover aircraft
Deploy rescue aircraft
Recover rescue aircraft
Deploy or recover two medium-size helicopters simultaneously
Aircraft Operations Management
Flight Operations Management
Control and manage aerial operations
Hangar
Aircraft Storage
Provide enclosed shelter for 1 meduim-size helicopter
Aircraft Elevator
Aircraft elevator
Lift aircraft
Lower aircraft
Auxiliary Subsystems
Temperature Control
Temperature control
Maintain temperature control
Communications Equipment
Communications Equipment
Communicate Information
Electric Plant
Electric Plant
Provide electrical power
Air Quality Control
Air quality control
Ensure air quality
Water Distribution
Water distribution
Supply potable water
Accomodation Spaces
Accomodation spaces
Provide accomodation and spaces for personnel
Landing Craft
- Requirement - 6.1 - R6.1 The vessel shall feature an overhead crane with a capacity of 20 tons for cargo handling operations
Receive AF and equipment for surface movement
- Requirement - 6 - R6 The vessel should facilitate the efficient loading and preparation of the landing craft or aircraft
- Requirement - 6.3 - R6.3 The vessel shall include a Ro-Ro ramp to ensure direct vehicle loading and unloading
Transport AF and equipment to shore via sea
- Requirement - 13 - R13 The vessel shall be equipped with surveillance capabilities, including radar systems, sonar systems, and optical sensors, to provide comprehensive situational awareness
- Requirement - 5.1.2 - R5.1.2 The vessel shall feature a well deck with dimensions of L55m x W15m x H6.5m
Transport AF and equipment to ship via sea
- Requirement - 5.1 - R5.1 The vessel shall support ship-to-shore movement via amphibious landing craft
- Requirement - 5.1.1 - R5.1.1 The vessel shall offer landing craft flexibility, capable of accommodating and deploying 2 LCUAs, 1 LCU, or 9 AAVs as needed
- Requirement - 6.2 - R6.2 The vessel shall include an aircraft elevator with a capacity of 20 tons to facilitate the vertical transportation of helicopters
- Requirement - 4.1 - R4.1 The vessel shall have dimensions optimized for navigability, with a maximum draft not exceeding 5.5 meters, a length overall (LOA) not exceeding 120 meters to accommodate port facilities, and a beam not exceeding 22 meters to ensure maneuverability within confined waterway...
- Requirement - 4 - R4 The vessel's dimensions shall be optimized to ensure navigability within designated ports, waterways, and operational constraints
Vertical Assault Aircraft
Receive AF and equipment for air movement
Transport AF and equipment to shore via air
Transport AF and equipment to ship via air
Fly towards shore
Fly towards ship
- Requirement - 2.1 - R2.1 The vessel shall provide 500 square meters of cargo area for general cargo
- Requirement - 2.2 - R2.2 The vessel shall accommodate 35 infantry fighting vehicles with a Ro-Ro area of approximately 1000 square meters and a payload weight capacity of approximately 875 metric tons for vehicles
- Requirement - 12 - R12 The vessel shall be equipped with communications capabilities, including satellite communication, radio communication and data transmission systems
- Requirement - 10 - R10 The vessel shall be equipped with an electric plant capable of meeting the vessel's electric demand under all operational conditions
- Requirement - 1.1 - R1.1 The vessel shall accommodate a total of 300 armed troops and 100 crew-personnel
- Requirement - 5.2.1 - R5.2.1 The vessel shall provide a flight deck measuring at least L40m x W20m, capable of simultaneous launch and recovery of two helicopters.
- Requirement - 5.2.2 - R5.2.2 The vessel shall provide hangar space for 1 helicopter
- Requirement - 5.2.3 - R5.2.3 The vessel shall feature a Helicopter Coordination Section to facilitate safe and efficient coordination of helicopter activities
- Requirement - 2 - R2 The vessel must have designated spaces for efficient embarkation, storage, and transportation of supplies, equipment, and amphibious vehicles
- Requirement - 9.1 - R9.1 The vessel shall regulate internal temperature for climate control
- Requirement - 9.2 - R9.2 The vessel shall ensure continuous airflow for fresh air
- Requirement - 9.3 - R9.3 The vessel shall ensure safe and drinkable water supply
- Requirement - 1 - R1 The vessel shall support the deployment of troops and crew personnel
- Requirement - 7 - R7 The vessel shall provide emergency rescue capabilities utilizing aerial methods
- Requirement - 5.2 - R5.2 The vessel shall support ship-to-shore movement via aerial means
- Requirement - 5 - R5 The vessel shall offer amphibious raid flexibility, enabling ship-to-shore movement via landing craft and/or aircraft
- Requirement - 9 - R9 The vessel shall maintain habitable conditions for embarked personnel

Figure 9.46: [PAB] Physical Architecture Final

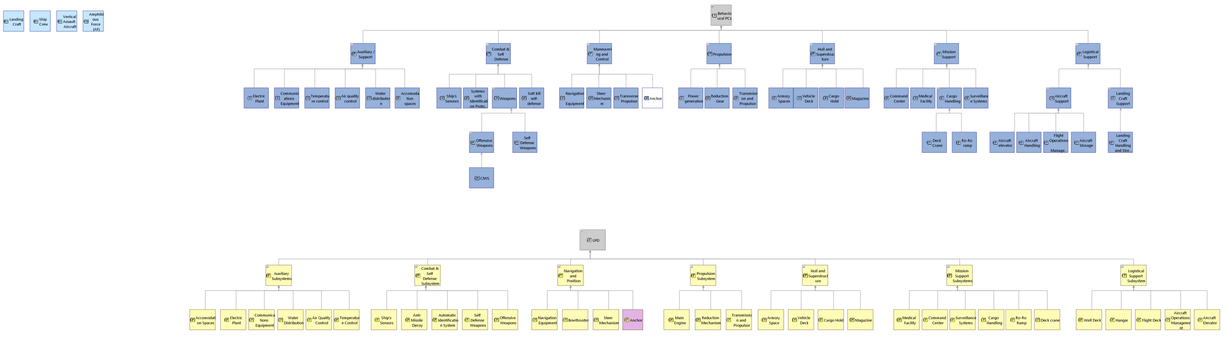


Figure 9.47: [PCBD] Structure

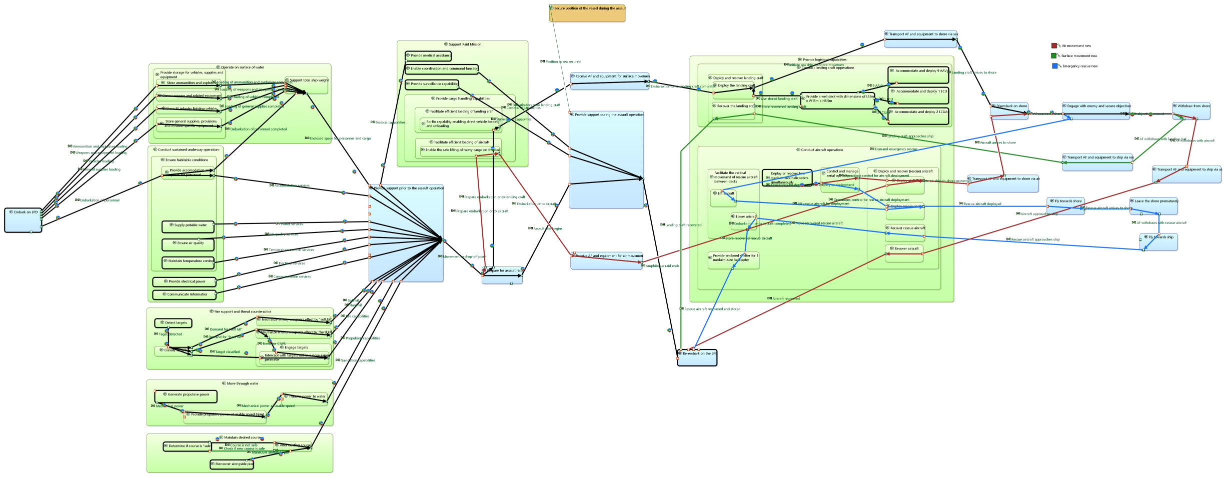


Figure 9.48: [PDFB] Root Physical Function

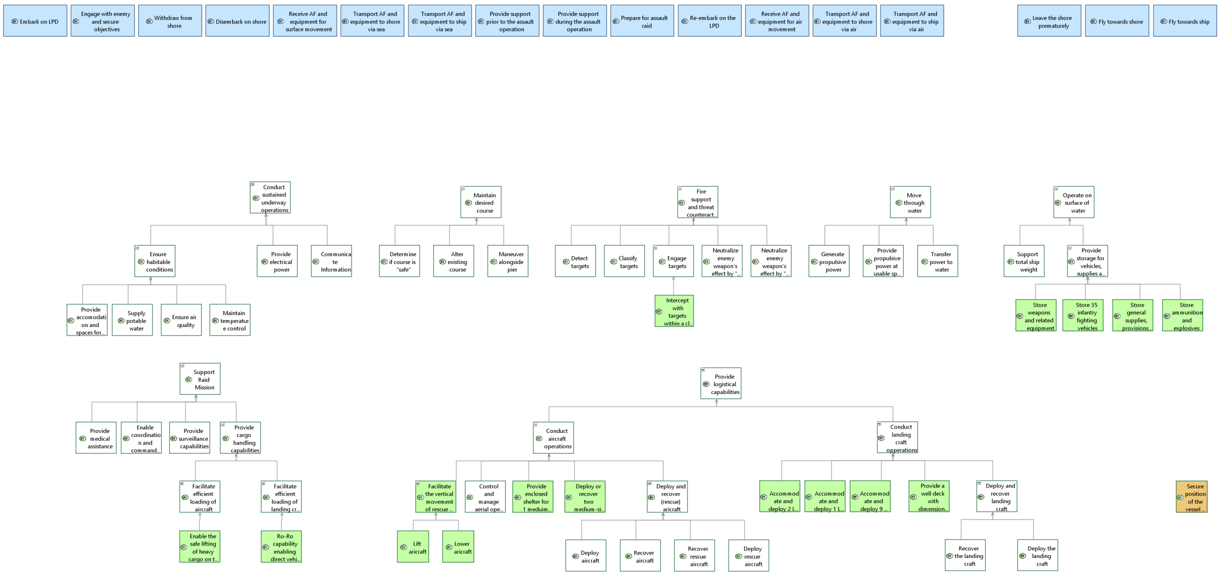
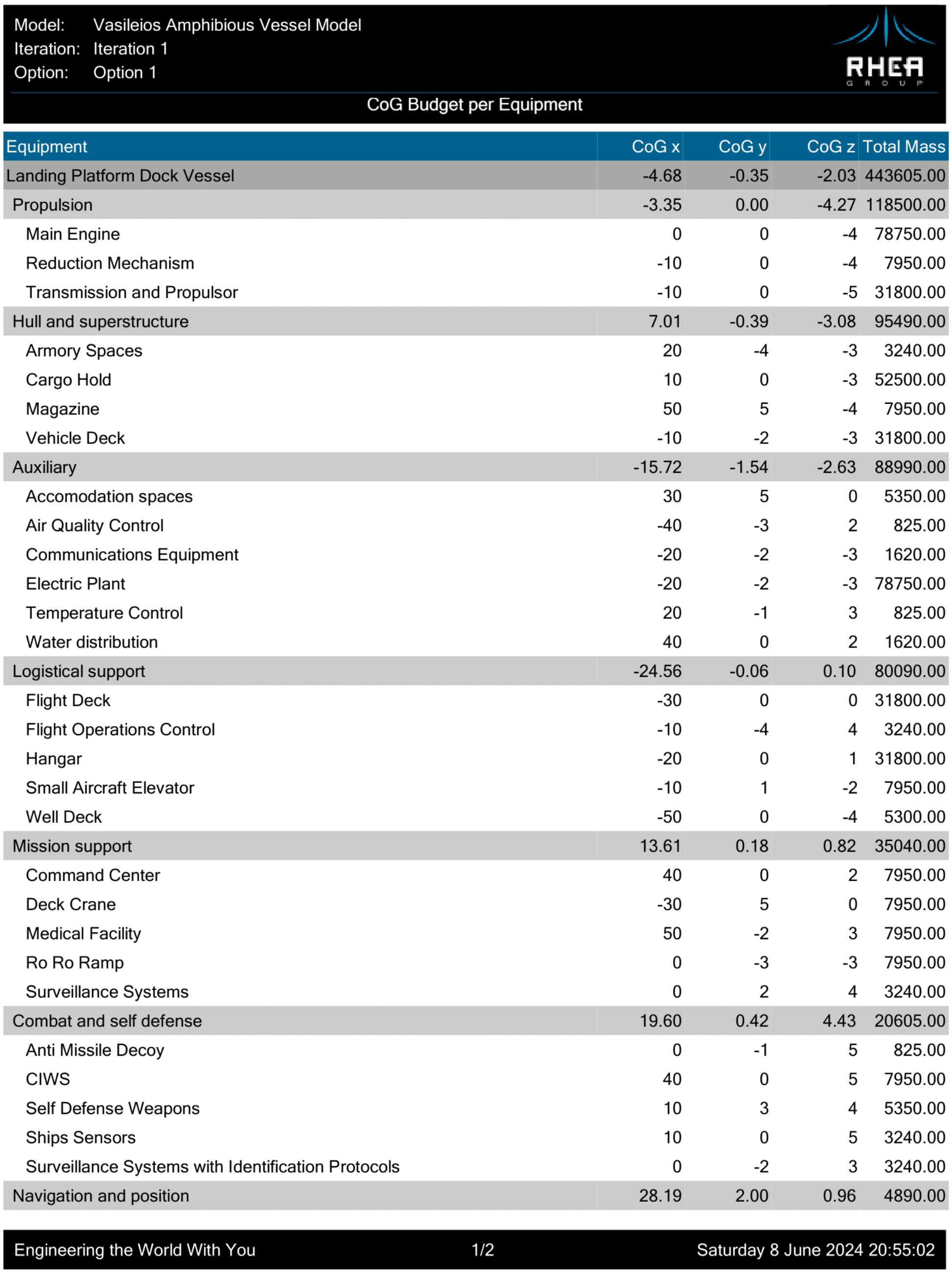

Embark on LPD
Engage with enemy and secure objectives
Withdraw from shore
Disembark on shore
Receive AF and equipment for surface movement
Transport AF and equipment to shore via sea
Transport AF and equipment to ship via sea
Provide support prior to the assault operation
Provide support during the assault operation
Prepare for assault raid
Re-embark on the LPD
Receive AF and equipment for air movement
Transport AF and equipment to shore via air
Transport AF and equipment to ship via air
Leave the shore prematurely
Fly towards shore
Fly towards ship
Conduct sustained underway operations
Ensure habitable conditions
Provide electrical power
Communicate Information
Provide accomodation and spaces for..
Supply potable water
Ensure air quality
Maintain temperature control
Maintain desired course
Determine if course is "safe"
Alter existing course
Maneuver alongside pier
Fire support and threat counteract..
Detect targets
Classify targets
Engage targets
Neutralize enemy weapon's effect by ..
Neutralize enemy weapon's effect by ..
Intercept with targets within a cl..
Move through water
Generate propulsive power
Provide propulsive power at usable sp..
Transfer power to water
Operate on surface of water
Support total ship weight
Provide storage for vehicles, supplies a..
Store weapons and related equipment
Store 35 infantry fighting vehicles
Store general supplies, provisions..
Store ammunition and explosives
Support Raid Mission
Provide medical assistance
Enable coordination and command
Provide surveillance capabilities
Provide cargo handling capabilities
Facilitate efficient loading of aircraft
Facilitate efficient loading of landing cr..
Enable the safe lifting of heavy cargo on t..
Ro-Ro capability enabling direct vehi..
Provide logistical capabilities
Conduct aircraft operations
Conduct landing craft operations
Facilitate the vertical movement of rescue ..
Control and manage aerial ope..
Provide enclosed shelter for 1 medium..
Deploy or recover two medium-si..
Deploy and recover (rescue) aircraft
Lift aircraft
Lower aircraft
Deploy aircraft
Recover aircraft
Recover rescue aircraft
Deploy rescue aircraft
Accommodate and deploy 2 L..
Accommodate and deploy 1 L..
Accommodate and deploy 9 ..
Provide a well deck with dimension..
Deploy and recover landing craft
Recover the landing craft
Deploy the landing craft
Secure position of the vessel

Figure 9.49: [PFBD] Root Physical Function

Mass Balance Reports created in CDP4-COMET

Mass Balance Report for Option 1



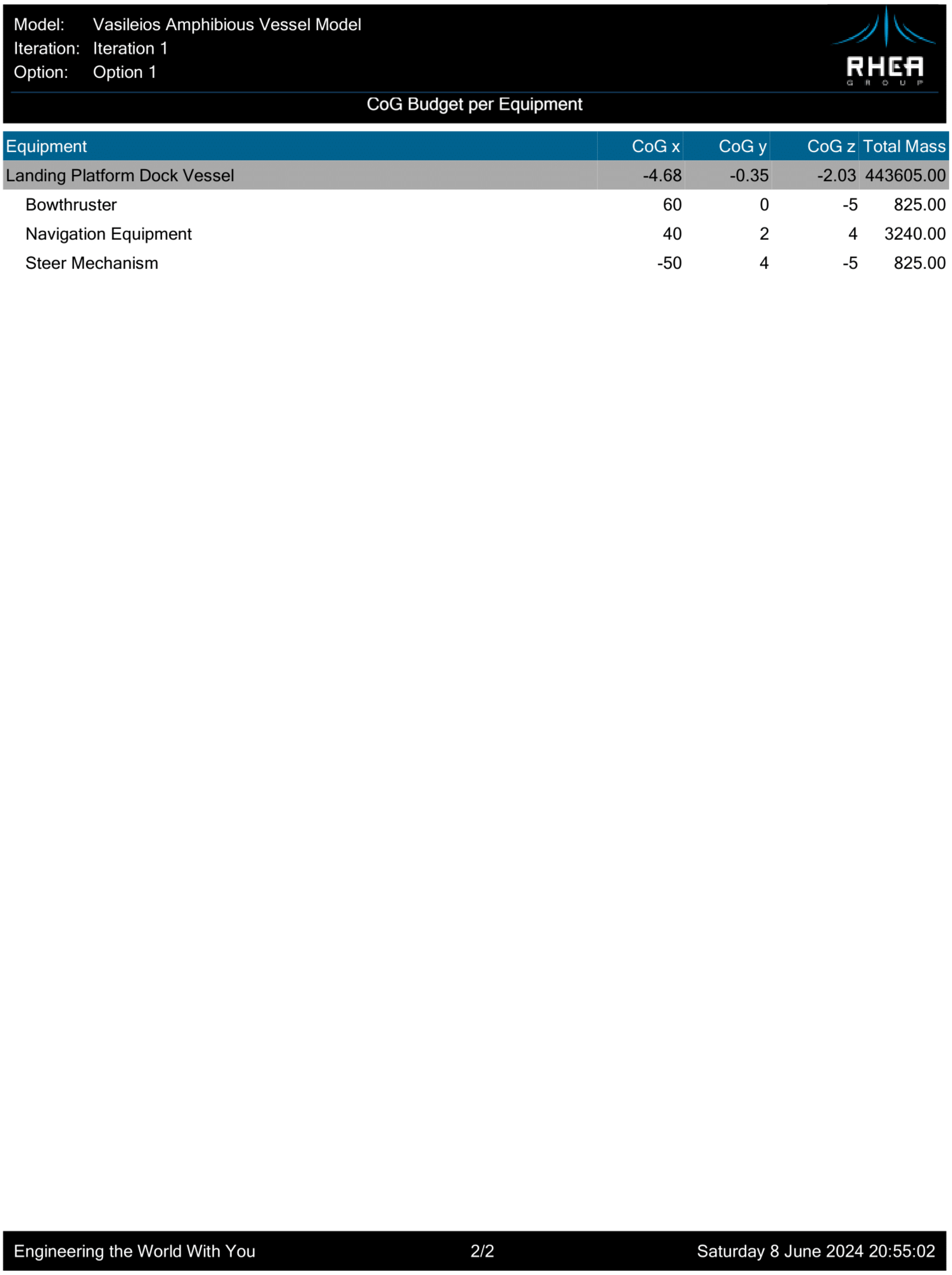
Model: Vasileios Amphibious Vessel Model
Iteration: Iteration 1
Option: Option 1

CoG Budget per Equipment

Equipment	CoG x	CoG y	CoG z	Total Mass
Landing Platform Dock Vessel	-4.68	-0.35	-2.03	443605.00
Propulsion	-3.35	0.00	-4.27	118500.00
Main Engine	0	0	-4	78750.00
Reduction Mechanism	-10	0	-4	7950.00
Transmission and Propulsor	-10	0	-5	31800.00
Hull and superstructure	7.01	-0.39	-3.08	95490.00
Armory Spaces	20	-4	-3	3240.00
Cargo Hold	10	0	-3	52500.00
Magazine	50	5	-4	7950.00
Vehicle Deck	-10	-2	-3	31800.00
Auxiliary	-15.72	-1.54	-2.63	88990.00
Accomodation spaces	30	5	0	5350.00
Air Quality Control	-40	-3	2	825.00
Communications Equipment	-20	-2	-3	1620.00
Electric Plant	-20	-2	-3	78750.00
Temperature Control	20	-1	3	825.00
Water distribution	40	0	2	1620.00
Logistical support	-24.56	-0.06	0.10	80090.00
Flight Deck	-30	0	0	31800.00
Flight Operations Control	-10	-4	4	3240.00
Hangar	-20	0	1	31800.00
Small Aircraft Elevator	-10	1	-2	7950.00
Well Deck	-50	0	-4	5300.00
Mission support	13.61	0.18	0.82	35040.00
Command Center	40	0	2	7950.00
Deck Crane	-30	5	0	7950.00
Medical Facility	50	-2	3	7950.00
Ro Ro Ramp	0	-3	-3	7950.00
Surveillance Systems	0	2	4	3240.00
Combat and self defense	19.60	0.42	4.43	20605.00
Anti Missile Decoy	0	-1	5	825.00
CIWS	40	0	5	7950.00
Self Defense Weapons	10	3	4	5350.00
Ships Sensors	10	0	5	3240.00
Surveillance Systems with Identification Protocols	0	-2	3	3240.00
Navigation and position	28.19	2.00	0.96	4890.00

Engineering the World With You 1/2 Saturday 8 June 2024 20:55:02

Figure 9.50: Mass Balance Report for Option 1 (Part I)



Model: Vasileios Amphibious Vessel Model
Iteration: Iteration 1
Option: Option 1

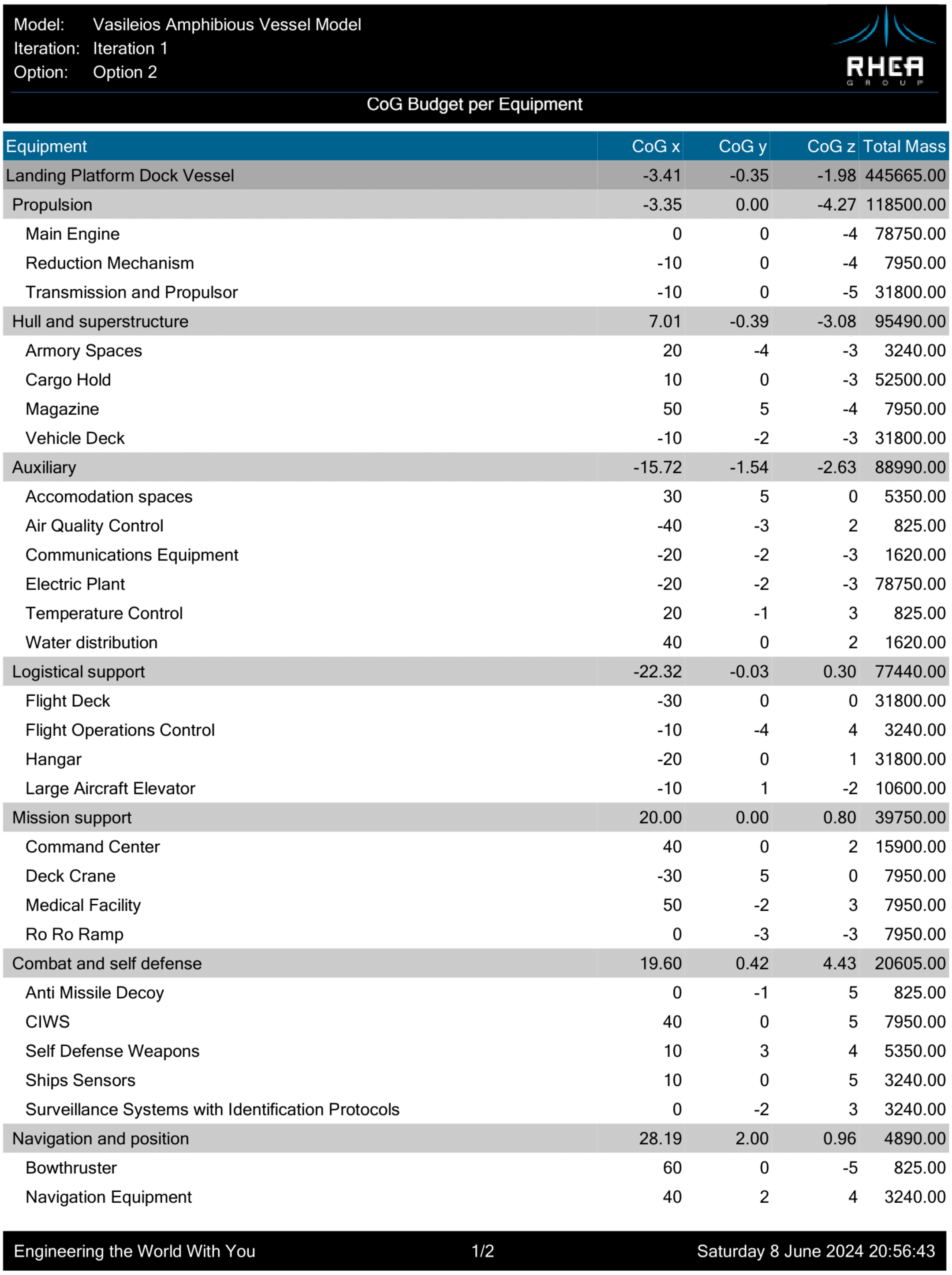
CoG Budget per Equipment

Equipment	CoG x	CoG y	CoG z	Total Mass
Landing Platform Dock Vessel	-4.68	-0.35	-2.03	443605.00
Bowthruster	60	0	-5	825.00
Navigation Equipment	40	2	4	3240.00
Steer Mechanism	-50	4	-5	825.00

Engineering the World With You   2/2   Saturday 8 June 2024 20:55:02

Figure 9.51: Mass Balance Report for Option 1 (Part II)

Mass Balance Report for Option 2



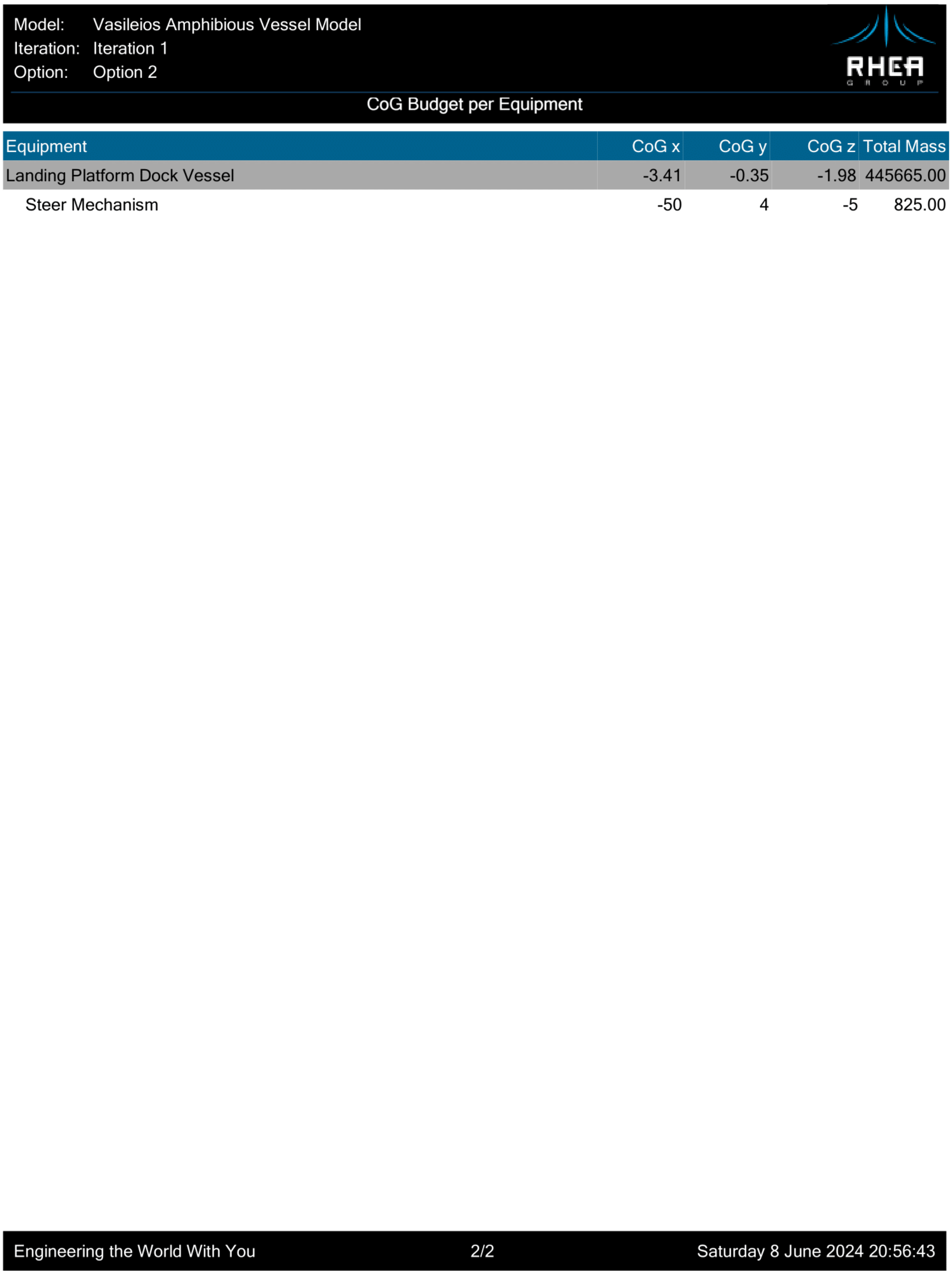
Model: Vasileios Amphibious Vessel Model
Iteration: Iteration 1
Option: Option 2

CoG Budget per Equipment

Equipment	CoG x	CoG y	CoG z	Total Mass
Landing Platform Dock Vessel	-3.41	-0.35	-1.98	445665.00
Propulsion	-3.35	0.00	-4.27	118500.00
Main Engine	0	0	-4	78750.00
Reduction Mechanism	-10	0	-4	7950.00
Transmission and Propulsor	-10	0	-5	31800.00
Hull and superstructure	7.01	-0.39	-3.08	95490.00
Armory Spaces	20	-4	-3	3240.00
Cargo Hold	10	0	-3	52500.00
Magazine	50	5	-4	7950.00
Vehicle Deck	-10	-2	-3	31800.00
Auxiliary	-15.72	-1.54	-2.63	88990.00
Accomodation spaces	30	5	0	5350.00
Air Quality Control	-40	-3	2	825.00
Communications Equipment	-20	-2	-3	1620.00
Electric Plant	-20	-2	-3	78750.00
Temperature Control	20	-1	3	825.00
Water distribution	40	0	2	1620.00
Logistical support	-22.32	-0.03	0.30	77440.00
Flight Deck	-30	0	0	31800.00
Flight Operations Control	-10	-4	4	3240.00
Hangar	-20	0	1	31800.00
Large Aircraft Elevator	-10	1	-2	10600.00
Mission support	20.00	0.00	0.80	39750.00
Command Center	40	0	2	15900.00
Deck Crane	-30	5	0	7950.00
Medical Facility	50	-2	3	7950.00
Ro Ro Ramp	0	-3	-3	7950.00
Combat and self defense	19.60	0.42	4.43	20605.00
Anti Missile Decoy	0	-1	5	825.00
CIWS	40	0	5	7950.00
Self Defense Weapons	10	3	4	5350.00
Ships Sensors	10	0	5	3240.00
Surveillance Systems with Identification Protocols	0	-2	3	3240.00
Navigation and position	28.19	2.00	0.96	4890.00
Bowthruster	60	0	-5	825.00
Navigation Equipment	40	2	4	3240.00

Engineering the World With You 1/2 Saturday 8 June 2024 20:56:43

Figure 9.52: Mass Balance Report for Option 2 (Part I)



Model: Vasileios Amphibious Vessel Model
Iteration: Iteration 1
Option: Option 2

CoG Budget per Equipment

Equipment	CoG x	CoG y	CoG z	Total Mass
Landing Platform Dock Vessel	-3.41	-0.35	-1.98	445665.00
Steer Mechanism	-50	4	-5	825.00

Engineering the World With You 2/2 Saturday 8 June 2024 20:56:43

Figure 9.53: Mass Balance Report for Option 2 (Part II)

References

ABET (2015). “Criteria for Accrediting Engineering Programs”. In: Baltimore, MD 21201.

Ackoff, Russell Lincoln et al. (2010). *Systems Thinking for Curious Managers: With 40 New Management F-Laws.* Triarchy Press Ltd, p. 96.

Alai, Shashank P. (Aug. 2019). “Evaluating ARCADIA/Capella vs. OOSEM/SysML for System Architecture Development”. In: DOI: `10.25394/PGS.8301050.v1`. URL: `https://hammer.purdue.edu/articles/thesis/Evaluating_ARCADIA_Capella_vs_OOSEM_SysML_for_System_Architecture_Development/8301050`.

Albers, Albert et al. (Jan. 2013). “Challenges of Model-Based Systems Engineering: A Study towards Unified Term Understanding and the State of Usage of SysML”. In: DOI: `10.1007/978-3-642-30817-8_9`.

Andrews, David (Jan. 1998). “A Comprehensive Methodology for the Design of Ships (and Other Complex Systems)”. In: *Proceedings: Mathematical, Physical and Engineering Sciences* 454.

— (Jan. 2011). “Marine Requirements Elucidation and the nature of Preliminary Ship Design”. In: *The International Journal of Maritime Engineering* 153, A23–A39. DOI: `10.3940/rina.ijme.2011.a1.202`.

— (Dec. 2018). “The Sophistication of Early Stage Design for Complex Vessels”. In: *International Journal of Maritime Engineering* Vol 160. DOI: `10.3940/rina.ijme.2018.SE.472`.

Arikan, Ugur et al. (2023). *A Capella Tutorial: Toy Catapult Project.* Accessed: 2024-01-05. URL: `https://gettingdesignright.com/GDR-Educate/Capella_Tutorial_v6_0/index.html`.

Bahlman, Jillian E. (2012). *A Model-Based Architecture Approach to Ship Design Linking Capability Needs to System Solutions.* Master thesis. Available at: `https://apps.dtic.mil/sti/citations/ADA562905`.

Bakker, Michael (2021). *A Reference-based Design Approach: in Preliminary Ship Design.* Master thesis. Available at TU Delft Library. `http://resolver.tudelft.nl/uuid:3b4f713f-13f2-45c8-9fed-f3ab7e1ac8f7`.

Bandecchi, Massimo et al. (Jan. 2000). “The ESA/ESTEC Concurrent Design Facility”. In.

Baron, Claude et al. (2023). “Using the ARCADIA/Capella Systems Engineering Method and Tool to Design Manufacturing Systemsmdash;Case Study and Industrial Feedback”. In: *Systems* 11.8. DOI: `10.3390/systems11080429`. URL: `https://www.mdpi.com/2079-8954/11/8/429`.

Beery, Paul T. (2016). *A Model-Based Systems Engineering Methodology for Employing Architecture in System Analysis: Developing Simulation Models Using Systems Modeling Language Products to Link Architecture and Analysis.* Master’s thesis. URL: `https://apps.dtic.mil/sti/tr/pdf/AD1026108.pdf`.

Bennett, James G. et al. (May 1996). “Concurrent Engineering – Application and Implementation for U.S. Shipbuilding”. In: *Journal of Ship Production* 12.2. Presented at the Ship Production Symposium, Seattle, Washington, January 25-27, 1995. Includes discussion., pp. 107–125. URL: `https://onepetro.org/JSPD/issue/25/01`.

Bernstein, J. (1998). *Design Methods in the Aerospace Industry: Looking for Evidence of Set-Based.* MSc thesis. Available at: `https://core.ac.uk/reader/18321770`.

Bonnet, S. (2019). *The Spirit of Arcadia and Capella in 7 Minutes.* Accessed 20 Jan 2024. URL: `https://www.youtube.com/watch?v=BtzhlZUaWA8`.

Boothroyd, G. et al. (2002). *Product Design for Manufacture and Assembly, 2nd Edition Revised and Expanded.* Marcel Dekker.

Brefort, Dorian et al. (2018). “An architectural framework for distributed naval ship systems”. In: *Ocean Engineering* 147, pp. 375–385. DOI: 10.1016/j.oceaneng.2017.10.028. URL: https://www.sciencedirect.com/science/article/pii/S0029801817306303.

Brown, A. et al. (1998). “Reengineering the Naval Ship Concept Design Process”. In: *From Research to Reality in Ship Systems Engineering Symposium*.

Bruinessen, Ties et al. (June 2013). “Towards a Different View on Ship Design: The Development of Ships Observed Through a Social-Technological Perspective”. In: vol. 1. DOI: 10.1115/OMAE2013-11585.

Bui Long, Anh Toan (2024). *Personal Communication with RHEA Group experts.* in person.

Capella, MBSE (n.d.). *A comprehensive methodological and tool-supported model-based engineering guidance.* Accessed on 19 01 2024. URL: https://mbse-capella.org/arcadia.html.

Carroll, Edward Ralph et al. (Mar. 2016). “Systematic Literature Review: How is Model-Based Systems Engineering Justified?” In: DOI: 10.2172/1561164. URL: https://www.osti.gov/biblio/1561164.

Case, Kelley et al. (Mar. 2021). “The Evolution of Team-X: 25 Years of Concurrent Engineering Design Experience”. In: pp. 1–6. DOI: 10.1109/AERO50100.2021.9438521.

Castro, Helder (2023a). *MBSE with Arcadia Method: Step by Step Operational Analysis.* https://www.linkedin.com/pulse/mbse-arcadia-method-step-by-step-system-analysis-helder-castro/?trackingId=1Q7VP1NrTFCXlNRNOOh7Uw%3D%3D. Accessed: 2024-01-21.

— (2023b). *MBSE with Arcadia Method: Step by Step Physical Architecture.* https://www.linkedin.com/pulse/mbse-arcadia-method-step-by-step-physical-helder-castro/?trackingId=1Q7VP1NrTFCXlNRNOOh7Uw%3D%3D. Accessed: 2024-01-21.

— (2023c). *MBSE with Arcadia Method: Step by Step System Analysis.* https://www.linkedin.com/pulse/mbse-arcadia-method-step-by-step-helder-castro/?trackingId=1Q7VP1NrTFCXlNRNOOh7Uw%3D%3D. Accessed: 2024-01-21.

Chalfant, Julie (Aug. 2015). “Early-Stage Design for Electric Ship”. In: *Proceedings of the IEEE* 103, pp. 1–15. DOI: 10.1109/JPROC.2015.2459672.

Charisi, Nikoleta et al. (June 2022). “Early-Stage Design of Novel Vessels: How can we Take a Step Forward?” In: DOI: 10.5957/IMDC-2022-239.

Cloutier, R. (2007). “The Use of Behavioral Diagrams in SysML”. In: *2007 IEEE Long Island Systems, Applications and Technology Conference*.

COMET™, Development Team (2023). *COMET Integrated Modelling Environment – Version 9.5.0 User Manual.* RHEA Group.

Delligatti, Lenny et al. (2014). *SysML distilled: A brief guide to the systems modeling language.* Addison-Wesley.

Department of the Navy, Office of the Chief of Naval Operations (May 2007). *NTTP 3-02.1M/MCWP 3-31.5, Ship-to-Shore Movement.* Retrieved from https://navytribe.files.wordpress.com/2015/11/nttp-3-02-1m.pdf. (Visited on 03/18/2024).

Dick, Jeremy et al. (2017). *Requirements Engineering (4th ed.)* Springer.

Dierolf, D. et al. (1998). “Computer-Aided Group Problem Solving for Unified Life Cycle Engineering (ULCE)”. In: *IDA Paper P-2149*.

Do, Quoc et al. (Dec. 2012). “Requirements for a Metamodel to Facilitate Knowledge Sharing between Project Stakeholders”. In: *Procedia Computer Science* 8, pp. 285–292. DOI: 10.1016/j.procs.2012.01.059.

Doerry, Norbert et al. (Sept. 2014). “Using Set-Based Design in Concept Exploration”. In.

Droste, Koen et al. (2024). *Personal Communication with DAMEN engineers during DND-MMBSE project.* in person. Discussions with Droste Koen, Bedert Simon and Audenaert Koen.

Duchateau, E.A.E. (2016). *Interactive evolutionary concept exploration in preliminary ship design*. Doctoral thesis. Available at TU Delft Library. https://doi.org/10.4233/uuid:27ff1635-2626-4958-bcdb-8aee282865c8.

Eclipse Foundation (n.d.). *Capella - Samples - In-Flight Entertainment System*. https://wiki.eclipse.org/Capella/Samples/IFE. Accessed: 2024-01-05.

Erikstad, Stein et al. (Jan. 2012). “System Based Design of Offshore Support Vessels”. In.

Erikstad, Stein Ove (2019). “Design for Modularity”. In: *A Holistic Approach to Ship Design*. Springer, pp. 329–356.

Evans, J. Harvey (1959). “Basic Design Concepts”. In: *Journal of the American Society for Naval Engineers*, pp. 671–678.

FEADSHIP (2017). *FEADSHIP Launches Concurrent Design*. URL: https://www.feadship.nl/news/feadship-launches-concurrent-design-feadship-cd-f.

Fernandes, Fabio M. (2023). *A Model-Based Systems Engineering Framework for developing Knowledge Based Engineering Applications*. Master thesis. Available at TU Delft Library. http://resolver.tudelft.nl/uuid:c9b8e9ad-3410-4d93-8a4c-b09b3313f0ad.

Finkel, S. et al. (2002). “Design Centers—Transferring Experience from Astronautics to Aeronautics”. In: *Proceedings of the 12th Annual Symposium of INCOSE (International Council on Systems Engineering)*. Las Vegas.

Friedenthal, Sanford et al. (2014). *A Practical Guide to SysML: The Systems Modeling Language*. Burlington, MA, USA: Morgan Kaufmann.

Fuchs, Christoph et al. (Dec. 2019). “Modular Design and Platforms”. In: *Mastering Disruption and Innovation in Product Management*. Management for Professionals. Springer. Chap. 7, pp. 123–145. DOI: 10.1007/978-3-319-93512-6. URL: https://ideas.repec.org/h/spr/mgmchp/978-3-319-93512-6_7.html.

Gaspar, H. et al. (2012). “Addressing Complexity Aspects in Conceptual Ship Design - A Systems Engineering Approach”. In: *Journal of Ship Production and Design* 28.4, pp. 1–15.

Gerene, Sam (2024). *Personal Communication as supervisor from RHEA Group*. in person.

Goodrum, Taylor Dean et al. (2019). “Understanding Initial Design Spaces: Set-Based Design Using Networks and Information Theory”. In: *Advances in Human Factors and Ergonomics*. Ed. by Gavriel Salvendy. CRC Press. Chap. 33, pp. 421–435. URL: https://www.taylorfrancis.com/chapters/edit/10.1201/9780429440533-33/understanding-initial-design-spaces-set-based-design-using-networks-information-theory-goodrum-taylordean-singer.

Group, Rhea (n.d.). *Concurrent Engineering in Support of Defence at the Dutch MOD*.

Haas, Roland et al. (Jan. 2004). “Concurrent Engineering at Airbus - A Case Study”. In: *IJMTM* 6, pp. 241–253. DOI: 10.1504/IJMTM.2004.005388.

Habben Jansen, A.C (2020). *A Markov-based vulnerability assessment of distributed ship systems in the early design stage*. Doctoral thesis. Available at TU Delft Library. https://doi.org/10.4233/uuid:f636539f-64a5-4985-b77f-4a0b8c3990f4.

Haskins, Cecilia (June 2011). “4.6.1 A historical perspective of MBSE with a view to the future”. In: *INCOSE International Symposium* 21, pp. 493–509. DOI: 10.1002/j.2334-5837.2011.tb01220.x.

Hastings, D. et al. (2004). “A Framework for Understanding Uncertainty and its Mitigation and Exploitation in Complex Systems”. In: *2004 Engineering Systems Symposium*. Cambridge, MA, USA, MIT, pp. 1–19.

Hause, Matthew (2011). “4.6.2 “Are we there yet?” Assessing Quality in Model Based Systems Engineering”. In: *INCOSE International Symposium* 21.1, pp. 510–522. DOI: https://doi.org/10.1002/j.2334-5837.2011.tb01221.x. eprint: https://incose.onlinelibrary.wiley.com/doi/pdf/10.1002/j.2334-5837.2011.tb01221.x. URL: https://incose.onlinelibrary.wiley.com/doi/abs/10.1002/j.2334-5837.2011.tb01221.x.

Hein, Andreas M. et al. (Jan. 2011). *Cookbook for MBSE with SysML.* Technical Report. INCOSE - SE2 MBSE Telescop Challenge Team. DOI: 10.13140/2.1.4291.2324.

Henderson, Kaitlin et al. (2021). "Value and benefits of model-based systems engineering (MBSE): Evidence from the literature". In: *Systems Engineering* 24.1, pp. 51–66. DOI: https://doi.org/10.1002/sys.21566. URL: https://incose.onlinelibrary.wiley.com/doi/abs/10.1002/sys.21566.

Hoffmann, H.-P. (2011). "Systems engineering best practices with the rational solution for systems and software engineering". In: *IBM Software Group* 4.2.

INCOSE (2007). *Systems Engineering Vision 2020.* INCOSE Technical Operations.

— (2014). *Systems Engineering Vision 2025.* URL: http://www.incose.org/docs/default-source/aboutse/se-vision-2025.pdf?sfvrsn=4.

— (2022). *Systems Engineering Vision 2035: Engineering Solutions for a Better World.* URL: https://www.incose.org/about-systems-engineering/se-vision-2035.

Jansen, Agnieta et al. (June 2020). "A Framework for Vulnerability Reduction in Early Stage Design of Naval Ship Systems". In: *Naval Engineers Journal* 132, p. 119.

Jenkins, Z I. (2021). "A Project-Oriented Model-Based Systems Engineering (MBSE) Approach for Naval Decision Support". PhD thesis. George Washington University.

Jorgensen, R. W. (2011). "Defining Operational Concepts using SysML: System Definition from the Human Perspective". In: *INCOSE International Symposium.* Ed. by I. Rockwell Collins. Denver, CO.

Kana, A. (2023a). "Digital Engineering - Design of Complex Specials - Lecture Notes".

— (2023b). "Modularity - Design of Complex Specials - Lecture Notes".

— (2023c). "Putting it all together - Design of Complex Specials - Lecture Notes".

— (2023d). "Set Based Design - Design of Complex Specials - Lecture Notes".

Kana, A. et al. (June 2016). "Why is Naval Design Decision-Making so Difficult?" In.

Kerns, Corey (2011). *Naval Ship Design and Synthesis Model Architecture Using a Model-Based Systems Engineering Approach.* Master thesis. Blacksburg, Virginia.

Kerns, Lt Corey et al. (2012). "Application of a DoDAF Total-Ship System Architecture in Building Naval Ship Operational Effectiveness Models". In: URL: https://api.semanticscholar.org/CorpusID:14590380.

Knegt, Sander (2018). *Winning at sea: Developing a method to provide insight in early stage naval fleet design requirements.* Master's thesis. Available at TU Delft Library. http://resolver.tudelft.nl/uuid:f5de21cd-dc49-42dc-a17e-2b6ec9a6bca8.

Kolfschoten, Gwendolyn et al. (2024). *Dutch Naval Design: Mission MBSE Pilot 1.*

Kooij, Egbert (2022). *Systems Modeling in the Naval Domain: Relating Stakeholders with Requirements in the Early Stages of Naval Ship Design.* Master's thesis. Available at TU Delft Library. https://doi.org/10.4233/uuid:12345678.

Kossiakoff, Alexander et al. (2011). *Systems Engineering Principles and Practice.* Vol. 83. John Wiley & Sons.

Krasner, Jerry (Oct. 2015). "How Product Development Organizations can Achieve Long-Term Cost Savings Using Model-Based Systems Engineering (MBSE)". In: *American Technology International, Inc.* Available at: https://docplayer.net/18566603-How-product-development-organizations-can-achieve-long-term-cost-savings-using-model-based-systems-engineering-mbse.html.

Le Masson, Pascal et al. (2013). "Design Theory: History, State of the Art, and Advancements". In: *Research in Engineering Design* 24.2, pp. 97–103. DOI: 10.1007/s00163-013-0154-4.

Lightsey, Bob (2001). *Systems Engineering Fundamentals.* Defense Acquisition University, p. 223.

Liker, Jeffrey et al. (1996). “Involving suppliers in product development in the United States and Japan: Evidence for set-based concurrent engineering”. In: *Engineering Management, IEEE Transactions on* 43, pp. 165–178. DOI: 10.1109/17.509982.

Logan, P. et al. (2013). “Model-based systems engineering metamodel: roadmap for effective systems engineering process”. In: *Systems Engineering/Test and Evaluation Conference [Proceedings CD-ROM].*

Long, David et al. (2012). *A Primer for Model-Based Systems Engineering (2nd ed.)* Vitech.

Madni, Azad et al. (May 2018). “Model-based systems engineering: Motivation, current status, and research opportunities”. In: *Systems Engineering* 21. DOI: 10.1002/sys.21438.

Maestro Redondo, Paloma et al. (2024). *Personal Communication with RHEA Group experts.* in person. Discussions with Maestro Redondo Paloma and González Rodríguez Belén and Kolfschoten Gwendolyn.

Maier, M.W. (1996). “Systems architecting: an emergent discipline?” In: *1996 IEEE Aerospace Applications Conference. Proceedings.* Vol. 3, 231–245 vol.3. DOI: 10.1109/AERO.1996.496066.

Mavris, Dimitri, Daniel DeLaurentis, et al. (1998). “A Stochastic Approach to Multi-disciplinary Aircraft Analysis and Design”. In: DOI: 10.2514/6.1998-912.

Mavris, Dimitri et al. (2000). “Methodology for Examining the Simultaneous Impact of Requirements, Vehicle Characteristics, and Technologies on Military Aircraft Design”. In.

McKenney, T. A. et al. (Feb. 2012). “Determining the Influence of Variables for Functional Design Groups in the Set-Based Design Process”. In: *ASNE Day 2012.* Arlington, VA.

McKenney, Thomas (2013). *An Early-Stage Set-Based Design Reduction Decision Support Framework Utilizing Design Space Mapping and a Graph Theoretic Markov Decision Process Formulation.* Doctoral thesis. Available at: https://hdl.handle.net/2027.42/102464.

McKenney, Thomas A. et al. (2011). “Adapting to Changes in Design Requirements Using Set-Based Design”. In: *Naval Engineers Journal* 123, pp. 66–77. DOI: 10.1111/j.1559-3584.2011.00331.x.

Morris, Brett Anthony (2019). “A Model-Based Systems Engineering Methodology to Support Early Phase Australian Off-the-Shelf Naval Ship Acquisitions”. Available at Adelaide Research Scholarship. https://hdl.handle.net/2440/119895. Doctoral thesis. University of Adelaide.

News, Naval (2024). *Dutch Navy to Replace OPV and LPD with a Single Class of Ships.* URL: https://www.navalnews.com/naval-news/2024/03/dutch-navy-to-replace-opv-and-lpd-with-a-single-class-of-ships/ (visited on 06/06/2024).

Obeo (n.d.). *Team for Capella.* https://www.obeosoft.com/en/team-for-capella. Accessed: 2024-03-27.

Odukoya, Kofoworola et al. (Oct. 2023). “Towards a semantic approach to knowledge exchange in Architectural Framework”. In: DOI: 10.13140/RG.2.2.18415.20645.

Oers, Bart van et al. (June 2018). “Warship Concept Exploration and Definition at The Netherlands Defence Materiel Organisation”. In: *Naval Engineers Journal* 130, p. 61.

ONR (2012). *ONR BAA 11-022 Assessing Total Ownership Costs.* Tech. rep. U.S. Department of Navy Office of Naval Research.

Packham, Karen (2021). *Time to say goodbye to documents and hello to MBSE.* URL: https://www.rheagroup.com/time-to-say-goodbye-to-documents-and-hello-to-mbse/ (visited on 12/05/2023).

Papanikolaou, Apostolos (Nov. 2010). “Holistic ship design optimization”. In: *Computer-Aided Design* 42, pp. 1028–1044. DOI: 10.1016/j.cad.2009.07.002.

Parsons, Michael et al. (Oct. 2011). “A Hybrid Agent Approach for Set-Based Conceptual Ship Design”. In.

Pearce, Paul (2013). “A Practical Approach For Modelling Submarine Subsystem Architecture In SysML”. In: URL: https://api.semanticscholar.org/CorpusID:9435926.

Perunovic, Zoran et al. (2011). “Innovation in the maritime industry”. In: URL: https://api.semanticscholar.org/CorpusID:54817259.

Piirainen, Kalle et al. (2009). "Unraveling Challenges in Collaborative Design: A Literature Study". In: *Groupware: Design, Implementation, and Use*. Ed. by Luís Carriço et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 247–261.

Poulis, Ioannis (2022). *Application of Model Based System Engineering (MBSE) with Ship Design Arrangement Tool of advanced zero emissions Power, Propulsion and Energy Systems in Maritime Technology*. Master's thesis. Available at TU Delft Library. URL: `http://resolver.tudelft.nl/uuid:7faf4cc0-c493-4efb-ad3b-5046ca208288`.

RHEA (June 2023). URL: `https://www.rheagroup.com/services-solutions/system-engineering/concurrent-design/download-cdp4-comet/`.

RHEA Group Services (2020). *CDP4 Integrated Modelling Environment Video Series*. Accessed: 2024-02-11. URL: `https://www.youtube.com/playlist?list=PL_4pFoUvQUgaZWOdXAS9xvOdz9D_qn5PU`.

Roques, Pascal (2018). *Systems architecture modeling with the Arcadia Method: A practical guide to capella*. ISTE Press Ltd.

Samares Engineering (2024). *Capella Use Case: UAV for Agriculture*. URL: `https://www.samares-engineering.com/en/category/non-classe/`.

Scheffers, Evelien (2021). *An improved approach for on-board distribution system robustness estimation in early-stage ship design*. Master thesis. Available at TU Delft Library. `http://resolver.tudelft.nl/uuid:b4e00788-990d-4531-b009-fe64e215bce6`.

Seram, N. (2013). "Decision Making in Product Development – A Review of the Literature". In: *International Journal of Engineering and Applied Sciences*, pp. 1–11.

Shevchenko, Nataliya (2020). *An Introduction to Model-Based Systems Engineering (MBSE)*. Carnegie Mellon University's Software Engineering Institute Blog. URL: `http://insights.sei.cmu.edu/blog/introduction-model-based-systems-engineering-mbse/`.

Shields, Colin (2017). *Investigating Emergent Design Failures Using a Knowledge-Action-Decision Framework*. Doctoral thesis. Available at: `https://hdl.handle.net/2027.42/140874`.

Singer, David et al. (Oct. 2009). "What Is Set-Based Design?" In: *Naval Engineers Journal* 121, pp. 31–43. DOI: `10.1111/j.1559-3584.2009.00226.x`.

Singer, David Jacob (2003). *A Hybrid Agent Approach for Set-Based Conceptual Ship Design Through the Use of a Fuzzy Logic Agent to Facilitate Communications and Negotiation*. Doctoral thesis. Available at: `https://hdl.handle.net/2027.42/123704`.

Sobek, D. K. (1997). *Principles that Shape Product Development Systems: A Toyota-Chrysler Comparison*. Doctoral thesis. Available at: `https://hdl.handle.net/2027.42/130365`.

Sobek, D. K. et al. (1999). "Toyota's Principles of Set-Based Concurrent Engineering". In: *Sloan Management Review* 40.2, pp. 67–83.

Sobey, Adam et al. (June 2013). "Implementation of a Generic Concurrent Engineering Environment Framework for Boatbuilding". In: *Journal of Marine Science and Technology* 18. DOI: `10.1007/s00773-012-0205-y`.

SRE-PA & D-TEC Staff (June 2012). *Margin Philosophy for Science Assessment Studies*. TN SRE-PA/2011.097/. Version 1.3. Status: N/A. Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands: European Space Research and Technology Centre. URL: `http://www.esa.int`.

Streng, Jurjen (2021). *Alternative Energy Carriers in Naval Vessels: Design Options and Implications for RNLN Large Surface Vessels*. Master's thesis. Available at TU Delft Library. URL: `http://resolver.tudelft.nl/uuid:47e02b82-5a0f-4eba-8092-f2e10b5c6845`.

Syan, C. S. et al. (1994). *Concurrent Engineering: Concepts, Implementation, and Practice*. 1st. Retrieved from `http://books.google.com/books?id=K-FTAAAAMAAJ`. Chapman & Hall.

Tepper, Nadia (2010). *Exploring the use of Model-Based Systems Engineering (MBSE) to develop systems architectures in naval ship design.* Master's thesis. Available at MIT Libraries. `http://hdl.handle.net/1721.1/61910`.

Tool, Capella MBSE (2024). *Capella Add-Ons.* Accessed: 2024-02-11. URL: `https://mbse-capella.org/addons.html`.

Topper, J. et al. (June 2013). "Model-Based Systems Engineering in Support of Complex Systems Development". In: *Johns Hopkins Apl Technical Digest* 32, pp. 419–432.

Tudor, W et al. (July 2019). "Virtual integration: managing complex warship design through model based engineering". In: DOI: `10.24868/issn.2515-8171.2019.009`.

United States Department of Defense (Jan. 2019). *Joint Publication 3-02: Amphibious Operations.* Validated on January 21, 2021. URL: `https://www.jcs.mil/Portals/36/Documents/Doctrine/pubs/jp3_02.pdf` (visited on 03/18/2024).

Van Oers, B.J. (2011). *A Packing Approach for the Early Stage Design of Service Vessels.* Doctoral thesis. Available at TU Delft Library. `http://resolver.tudelft.nl/uuid:6be7582c-63b1-477e-b836-87430bcfb43f`.

Vaneman, Warren K. et al. (2020). "Evaluating Current Systems Engineering Models for Applicability to Model-Based Systems Engineering Technical Reviews". In: *Naval Engineers Journal* 132.2, pp. 51–58.

Verma, Dinesh et al. (2021). *MBSE Applied.* URL: `https://esi.nl/events/2021/160421` (visited on 12/05/2023).

Vestbøstad, Øyvind (2011). *System Based Ship Design for Offshore Vessels.* MSc Thesis. Available at NTNU Open: `http://hdl.handle.net/11250/265905`.

Voirin, Jean-Luc (2018). *Conception Architecturale des systèmes basée sur les modèles avec la méthode Arcadia.* ISTE Editions Ltd.

Walden, David D. et al. (2015). *Systems engineering handbook: A guide for system life cycle processes and activities.* Wiley.

Ward, A. C. et al. (1995). "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster". In: *Sloan Management Review* 36.2, pp. 43–61.

Weilkiens, Tim (2022). *Definition of MBSE - Revised.* Accessed: 2024-01-14. URL: `https://mbse4u.com/2022/01/11/definition-of-mbse-revised`.

Weilkiens, Tim et al. (2023). *The Craft of MBSE.* MBSE4U. URL: `http://leanpub.com/craft-of-mbse`.

Wilcox, K. (Sept. 2018). *Fusing Models and Data to Achieve Efficient Design, Optimization, and Uncertainty Quantification.*