



**NAVAL
POSTGRADUATE
SCHOOL**

MONTEREY, CALIFORNIA

THESIS

**A MODEL-BASED SYSTEM ENGINEERING
APPROACH TO REQUIREMENTS MANAGEMENT**

by

Chun Ngee Alvin Koh

September 2023

Thesis Advisor:
Second Reader:

Eugene P. Paulo
Paul T. Beery

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.			
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE September 2023	3. REPORT TYPE AND DATES COVERED Master's thesis	
4. TITLE AND SUBTITLE A MODEL-BASED SYSTEM ENGINEERING APPROACH TO REQUIREMENTS MANAGEMENT		5. FUNDING NUMBERS	
6. AUTHOR(S) Chun Ngee Alvin Koh			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.		12b. DISTRIBUTION CODE A	
13. ABSTRACT (maximum 200 words) Requirements management is essential to the success of any project or product. Currently, requirements derived from users' needs are often document-centric and expressed in textual statements. Conventional requirements management techniques, although widely used, are known to have various shortcomings. A model-based systems engineering (MBSE) approach to requirements management, using models to capture and represent system requirements, has emerged with promise. This thesis explored the application of MBSE to requirements management with the goal of achieving improved communications, increased reusability, and better traceability. A model was created to transform a sample set of text-based requirements into a model-based representation. From this model, an evaluation of its effectiveness was conducted based on the performance metrics defined. Results showed that an MBSE approach had fulfilled the measures of effectiveness and met the goal of improving requirements management. However, for the successful implementation of MBSE on requirements management, there are hurdles to overcome and future work to be explored.			
14. SUBJECT TERMS model-based systems engineering, MBSE, requirements, requirements management, system engineering		15. NUMBER OF PAGES 109	16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

**A MODEL-BASED SYSTEM ENGINEERING APPROACH TO
REQUIREMENTS MANAGEMENT**

Chun Ngee Alvin Koh
Civilian, ST Engineering
Bachelor of Electronics Engineering, University of Sheffield, 2003

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

**NAVAL POSTGRADUATE SCHOOL
September 2023**

Approved by: Eugene P. Paulo
Advisor

Paul T. Beery
Second Reader

Oleg A. Yakimenko
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Requirements management is essential to the success of any project or product. Currently, requirements derived from users' needs are often document-centric and expressed in textual statements. Conventional requirements management techniques, although widely used, are known to have various shortcomings. A model-based systems engineering (MBSE) approach to requirements management, using models to capture and represent system requirements, has emerged with promise. This thesis explored the application of MBSE to requirements management with the goal of achieving improved communications, increased reusability, and better traceability. A model was created to transform a sample set of text-based requirements into a model-based representation. From this model, an evaluation of its effectiveness was conducted based on the performance metrics defined. Results showed that an MBSE approach had fulfilled the measures of effectiveness and met the goal of improving requirements management. However, for the successful implementation of MBSE on requirements management, there are hurdles to overcome and future work to be explored.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	BACKGROUND	1
B.	RESEARCH QUESTION AND OBJECTIVES	1
1.	Improved Communications.....	1
2.	Increased Reusability.....	2
3.	Better Traceability	2
C.	METHODOLOGY AND THESIS ORGANIZATION	2
II.	LITERATURE REVIEW	3
A.	REQUIREMENTS MANAGEMENT TECHNIQUES.....	3
B.	MODEL-BASED SYSTEM ENGINEERING	5
C.	INTEGRATED BRIDGE AND PLATFORM MANAGEMENT SYSTEM	7
III.	PROPOSED ENHANCEMENT TO REQUIREMENTS MANAGEMENT	13
A.	SYSTEMS ENGINEERING	13
B.	CURRENT SYSTEM ENGINEERING APPROACH.....	13
C.	APPLYING MODEL-BASED SYSTEMS ENGINEERING	15
D.	MEASURES OF EFFECTIVENESS.....	16
1.	Improved Communications between Stakeholders	16
2.	Reusability of Requirements.....	17
3.	Traceability of Requirements	18
E.	APPLICATION OF MBSE TO REQUIREMENTS MANAGEMENT	19
F.	DEVELOPING THE SYSTEM MODEL.....	20
1.	Innoslate.....	20
2.	Methodology for Model Development.....	24
3.	Modeling the Requirements	26
4.	Requirement Diagram	30
5.	LML Diagrams.....	32
6.	Traceability Matrix.....	35
7.	Verification and Validation.....	37
IV.	RESULTS AND OBSERVATIONS.....	41
A.	IMPROVED COMMUNICATIONS	41
1.	Multiple Views.....	41

2.	Centralized Database.....	42
B.	INCREASED REUSABILITY	44
1.	Reusable Requirement Templates.....	44
2.	Requirement Repositories	44
3.	Reusable Requirement Model.....	45
C.	BETTER TRACEABILITY	46
1.	Centralized Repository.....	46
2.	Hierarchical Structure.....	47
3.	Traceability Links.....	47
4.	Impact Analysis.....	47
V.	CONCLUSIONS AND FUTURE WORK	51
A.	SUMMARY	51
B.	CONCLUSIONS	51
C.	FUTURE WORK.....	53
	APPENDIX A. SELECTED LIST OF REQUIREMENTS	55
	APPENDIX B. COMPLETE ACTION DIAGRAMS	63
	APPENDIX C. COMPLETE REQUIREMENT DIAGRAMS	67
	APPENDIX D. SPIDER DIAGRAM	71
	APPENDIX E. HIERARCHY DIAGRAMS	73
	APPENDIX F. COMPLETE TRACEABILITY MATRIX	75
	APPENDIX G. TREE DIAGRAM	81
	LIST OF REFERENCES	83
	INITIAL DISTRIBUTION LIST	87

LIST OF FIGURES

Figure 1.	IBS typical architecture. Source: Anschutz (n.d.).....	8
Figure 2.	Bridge consoles that house IBS and IPMS workstations. Adapted from OSI Maritime Systems (n.d.).	9
Figure 3.	IPMS typical architecture. Source: L3Harris (n.d.).	10
Figure 4.	A typical functional hierarchy diagram	14
Figure 5.	A typical physical hierarchy diagram	14
Figure 6.	Innoslate Dashboard provides information for monitoring status.	21
Figure 7.	Systems engineering “V” diagram.....	23
Figure 8.	Modified “V” diagram	24
Figure 9.	Innoslate system architecture. Source: SPEC Innovations (2022).....	25
Figure 10.	Innoslate requirements document view showing the list of requirements.....	29
Figure 11.	Autopilot requirement with multiple labels	29
Figure 12.	Requirement diagram of IBS	30
Figure 13.	A requirement construct with information on the requirement	31
Figure 14.	Action diagram of IBPMS	33
Figure 15.	Spider diagram depicting the relationship of the entities.....	35
Figure 16.	Traceability matrix of WECDIS requirements vs. Actions	36
Figure 17.	Traceability matrix of WECDIS requirements vs. Assets	37
Figure 18.	SAT plan for autopilot	38
Figure 19.	Spider diagram linking autopilot test entities to requirements	39
Figure 20.	Traceability matrix of autopilot SAT test cases vs. requirements	40
Figure 21.	Traceability links between requirements and other artifacts	48
Figure 22.	Requirement diagram link to Actions	49

Figure 23.	Action diagram of IPMS (decomposed from IBPMS)	63
Figure 24.	Action diagram of IBS (decomposed from IBPMS).....	64
Figure 25.	Action diagram of “Compute Navigational Data” (decomposed from IBS)	65
Figure 26.	Action diagram of “Provide Navigation” (decomposed from IBS).....	66
Figure 27.	Requirement diagram for NRS	67
Figure 28.	Requirement diagram for autopilot system.....	68
Figure 29.	Requirement diagram for WECDIS.....	69
Figure 30.	Traceability paths between IBPMS Assets, Actions, and Input/ Output signals	71
Figure 31.	Functional hierarchy diagram of IBPMS generated from action diagram	73
Figure 32.	Physical hierarchy diagram of IBPMS generated from action diagram	74
Figure 33.	Traceability matrix of requirements vs. Actions (Part 1 of 3).....	75
Figure 34.	Traceability matrix of requirements vs. Actions (Part 2 of 3).....	76
Figure 35.	Traceability matrix of requirements vs. Actions (Part 3 of 3).....	77
Figure 36.	Traceability matrix of requirements vs. Assets (Part 1 of 2).....	78
Figure 37.	Traceability matrix of requirements vs. Assets (Part 2 of 2).....	79
Figure 38.	Tree diagram showing relationships between IBPMS requirements.....	81

LIST OF TABLES

Table 1.	Survey results – Introduce new members to system. Source: Huldt and Stenius (2019).	42
Table 2.	Survey results – Establish unified understanding of system. Source: Huldt and Stenius (2019).	43
Table 3.	List of requirements from selected systems.....	55

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AIS	automatic identification system
CCTV	closed-circuit television
CMS	combat management system
CNL	controlled natural language
ECDIS	electronic chart display identification system
FDS	fuel distribution system
GPS	global positioning system
HVAC	heating, ventilation, and air conditioning
IBPMS	integrated bridge-platform management system
IBS	integrated bridge system
IEEE	Institute of Electrical and Electronics Engineers
IMO	International Maritime Organization
IPMS	integrated platform management system
LML	Lifecycle Modeling Language
MBSE	model-based systems engineering
MOE	measures of effectiveness
NRS	navigation radar system
RTM	requirement traceability matrix
SAT	sea acceptance test
SOLAS	Safety of Life at Sea
SOW	statement of work
SysML	System Modeling Language
V&V	verification and validation
VDR	voyage data recorder
WECDIS	warship electronic chart display identification system

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Effective requirements management is a critical aspect of successful system development, ensuring that the system meets the needs and expectations of stakeholders. Traditionally, this process involved documenting requirements in textual formats, leading to potential challenges such as ambiguity, inconsistency, high error rates, and difficulty in traceability. Model-based system engineering (MBSE) has emerged as an approach with promise, to enhance the requirements management processes by utilizing models as a means of capturing, analyzing, and communicating requirements. These models provide a clear and intuitive representation of system components, their relationships, and their behavior, enabling stakeholders to better understand and communicate complex requirements.

This thesis seeks to explore the effectiveness of applying MBSE to requirements management and, in the process, identify any associated key challenges and benefits. This is achieved by identifying three research objectives with specific metrics to determine their effectiveness. The three research objectives are improved communications, increased reusability, and better traceability. This thesis focuses on the requirements from formulated requirements at the design phase. Therefore, the quality of the requirements and formulation of requirements would not be addressed.

Using Innoslate, a model was developed to represent designated requirements for the integrated bridge-platform management system (IBPMS) of a naval ship. From this model, a range of constructs for representing requirements was generated, including requirements diagrams, action diagrams, spider diagrams, hierarchy diagrams, and traceability matrices. These constructs allowed the model to be evaluated to determine the effectiveness of the MBSE methodology.

The first research objective of improved communications among stakeholders was fulfilled by MBSE's capability to generate multiple views to communicate with different stakeholders with the assurance that the underlying information is derived from the same dataset. Depending on needs, a stakeholder has the option of selecting the view(s) most

appropriate for the desired task. All these views, together with other system-related information, are “stored and configuration-managed in a central repository” (Madni and Sievers 2018, 172). Any amendments to the model will be tracked and made aware to all members. This ensures all stakeholders have access to the latest and most up-to-date information, enabling real-time collaboration and promoting transparency, thereby leading to significant improvements in communications.

The second research objective of increased reusability was achieved by creating a standardized approach to requirements development and management. Requirement templates that capture common requirements structures or patterns, including their attributes, relationships, and properties, can be created and reused across multiple projects. Furthermore, MBSE allows requirement libraries or repositories to be created for storing and maintaining the requirements templates and reusable requirements. Lastly, a key advantage of an MBSE approach is the ability for the whole requirement model to be imported from one project into another, effectively reusing the entire model or significant portions of it.

The third research objective of better traceability is achieved by leveraging MBSE elements like providing a centralized repository, establishing a hierarchical structure, enabling traceability links, and supporting impact analysis. This enhances the ability to track, manage, and validate requirements throughout the system development process, promoting consistency, transparency, and effective decision-making.

In summary, this thesis suggests requirements management using MBSE brings substantial improvements to the system development process. By harnessing visual models, improvements to traceability, communication, and reusability can be realized, leading to enhancements in the efficiency, effectiveness, and quality of requirements management, ultimately contributing to the successful delivery of complex systems.

Further refinement of MBSE’s modeling approach to requirements management can be carried out by exploring additional performance metrics and evaluating their applicability in other major systems on naval platforms. Future work can be conducted to

explore integrating requirements management with other engineering disciplines, promoting consistency, coherence, and alignment across system development activities.

References

Madni, Azad M., and Michael Sievers. 2018. "Model-Based Systems Engineering: Motivation, Current Status, and Research Opportunities." *Systems Engineering* 21 (3): 172–90. <https://doi.org/10.1002/sys.21438>.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to take this opportunity to express my sincere gratitude and appreciation to all those who have played a significant role in the completion of my thesis. Their support, guidance, and encouragement have been instrumental in making this achievement possible.

First and foremost, I am immensely grateful to my thesis advisor, Professor Eugene Paulo. His expertise and mentorship have been invaluable throughout this research journey. His insightful feedback and constant motivation have helped shape my ideas and refine my work. I am deeply indebted to him for his patience, guidance, and belief in my abilities.

I would also like to extend my gratitude to my second reader, Professor Paul Beery, for his valuable input, suggestions, and critical evaluation of my thesis. I am very thankful for his effort and contribution to my work.

Additionally, I would like to express my heartfelt appreciation to my family for their unwavering support, love, and understanding throughout this journey. Their belief in my capabilities, encouragement during challenging times, and constant motivation have been the driving force behind my accomplishments. Their sacrifices and continuous encouragement have inspired me to persevere and overcome obstacles.

Furthermore, I am grateful to Singapore Technologies Engineering Marine Ltd for granting me the privilege of studying at the Naval Postgraduate School.

Finally, I would like to acknowledge the support of my friends who have been there for me, providing encouragement and offering valuable insights and discussions.

Completing this thesis would not have been possible without the collective support of my thesis advisor, second reader, family, and friends. I am profoundly grateful for their contributions and am honored to have had them in this rewarding academic endeavor.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Current approaches to requirements management involve a linear process of gathering requirements, documenting them in a requirements document, and then verifying and validating them through various testing techniques. The requirements are often text-based, expressing the needs of the user in specific statements. Although conventional requirements management techniques are extensively used, they have various shortcomings, such as inadequate traceability, inconsistency, incompleteness, and high error rates.

Model-based systems engineering (MBSE) has emerged as an approach with promise, to enhance requirements management using models that can be analyzed, simulated, and validated, enabling stakeholders to capture and represent system requirements. However, there are various factors, such as the quality of the models, cost, and receptiveness to adopt by the stakeholders, that may limit the effectiveness of MBSE in requirements management. This research seeks to gain a better understanding of the effectiveness of MBSE in requirements management, as well as identify the and benefits associated with this approach.

B. RESEARCH QUESTION AND OBJECTIVES

The goal of this research is to improve requirements management through the use of MBSE. That goal will answer the following question: Based on appropriate measures of success, are there applications of specific MBSE products that clearly result in improvements to the requirements management process? The specific research objectives that will be assessed in support of that question are:

1. Improved Communications

MBSE enables better communication among stakeholders by providing a common language and framework for expressing requirements. This helps ensure that everyone is on the same page, which can reduce misunderstandings and conflicts.

2. Increased Reusability

MBSE can increase the reusability of requirements by enabling them to be stored and reused in different contexts. This can save time and effort in future projects and reduce the risk of errors.

3. Better Traceability

MBSE provides better traceability of requirements by enabling stakeholders to trace requirements from origin to implementation. This can help ensure that all requirements are met and provide better insight into the impact of changes.

Overall, using MBSE for requirements management can improve the effectiveness and efficiency of any organization, leading to better systems and a more efficient development process.

C. METHODOLOGY AND THESIS ORGANIZATION

Requirements management is essential to the success of any project or product (Hood et al. 2008). A set of requirements for an integrated bridge-platform management system (IBPMS) is modeled to transform it from a standalone text-based form to a model-based form. This modeling process explores the benefits of using MBSE for requirements management.

The organization of this thesis is as follows. Chapter II presents the literature review, which discusses the current approaches to requirements management and their effectiveness. Chapter III details the proposed enhancement to the requirements management process with a discussion on the potential measures of effectiveness (MOE). In Chapter IV, the results and observations from the application of MBSE are discussed, and Chapter V presents the conclusion and possible future work/research areas.

This thesis focuses on the requirements from formulated requirements at the design phase. We do not address the quality of requirements and formulation of requirements.

II. LITERATURE REVIEW

Requirements management refers to the process of identifying, analyzing, documenting, validating, and controlling the requirements for a system or product. This process involves gathering requirements from stakeholders and translating them into specific, measurable, and testable criteria the system must meet. Effective requirements management is critical to the success of a project (Zhang and Zhang 2020), as it helps to ensure that the system meets the needs of the user, is developed on time and within budget, and meets any applicable regulatory or quality standards. Requirements management typically involves using tools and techniques to track changes to the requirements, manage dependencies between requirements, and ensure that the requirements are consistent, complete, and unambiguous. It is an ongoing process that requires collaboration between stakeholders throughout the development lifecycle to ensure that the system meets the needs of all stakeholders.

The foundation of this research into the improvement of requirements management involves a review of current approaches to requirements management, an overview of relevant applications of MBSE, and a summary of the system being examined to demonstrate the proposed approach's benefits. Therefore, this literature review focuses on three main areas, a) requirements management techniques, b) effectiveness of MBSE, and c) challengers of shipboard systems requirements.

A. REQUIREMENTS MANAGEMENT TECHNIQUES

Contemporary requirements management often utilizes text-based methods to document stakeholders' requirements and track changes in the requirements throughout the product lifecycle. One of the key activities of requirements management is traceability.

Improper requirements management caused by requirement traceability issues is one of the reasons for project failure based on the STANDISH Report (Satyarthi and Pandey 2017, 904). To address this issue, Satyarthi and Pandey proposed a requirements management framework comprising three phases—Planning, Execution, and

Management. These three phases combined to manage the requirements of all the various stakeholders. The key apparatus within the framework that links the three phases is the requirement traceability matrix (RTM) which links the requirements from the initial to the final stage (Satyarthi and Pandey 2017, 906). Although it is a good practice to implement requirement traceability for requirements management, the RTM is still mainly text-based, leaving it susceptible to incompleteness or inconsistencies.

Zhang and Zhang (2020) introduced an integrated requirements management platform that applies semantic analysis on template, attribute and rules databases, and knowledge database established for a particular aircraft type. There are multiple levels of requirements for an aircraft, from the stakeholder needs at the top, followed by aircraft-level requirements, system-level requirements, and sub-system requirements (Zhang and Zhang 2020). Their research proposed for different templates to be defined for each requirement level, with the templates containing all the various attributes of the requirements to form an attribute database. In addition, they proposed a rule database to automatically check for errors in the requirements for improved quality and development efficiency. They then further established a knowledge database to contain good requirements developed in the past. However, Zhang and Zhang highlighted that these good requirements are typically expressed in unstructured natural language, resulting in inconsistencies and ambiguities. A semantic analysis function is then employed to “improve the language expression quality of requirements” (5). Despite the potential improvements that the integrated requirements management platform could provide, it is still a text-based methodology. It continues to suffer from shortcomings like the inability to fully capture some requirements with text and the inability to visualize the linkages between requirements.

Kravari, Antoniou, and Bassiliades (2021) also proposed using semantics in their SENSE framework. SENSE is a framework that utilizes “standardized expressions of natural language with well-defined semantics, called boilerplates, that support a flow-down procedure for requirements management” (1). The aim was to standardize the requirements to minimize inconsistencies.

Apart from text-based methods, there are considerable recent developments in Agile requirement engineering methods (Vierlboeck and Nilchiani 2021). Agile requirement engineering is an iterative and collaborative approach to capturing and managing requirements in agile software development projects, which usually have shorter development cycles (Ramesh, Cao, and Baskerville 2010, 450). Ramesh, Cao, and Baskerville (2010) stated that Agile methods often suit rapidly changing software development as it supports the commencement of coding before requirements have been finalized. Instead, the requirements will be formed in tandem with the software development process. While this approach emphasizes flexibility, adaptability, and responsiveness to changing requirements and customer needs, it is not suited for the design of shipboard systems, where design work will only begin after requirements have been established.

Furthermore, Agile methods call for stakeholders to constantly be engaged to review and refine requirements (Schon et al. 2017). This is not feasible for the typical naval shipbuilding project, which takes years to materialize as stakeholders will change throughout the project. For this reason, possessing the capability of requirements, traceability takes precedence over the need to be flexible and adaptable to rapidly changing requirements.

B. MODEL-BASED SYSTEM ENGINEERING

In the traditional approach, requirements are often expressed in text-based form, using natural language statements to describe the needs of the user or system. However, this can lead to ambiguity and inconsistency in the requirements, as different stakeholders may interpret the text differently.

To address these issues, MBSE has emerged as a promising alternative approach to requirements management. MBSE is the “formalized application of modeling to support system requirements, design, analysis, verification, and validation, beginning in the conceptual design phase and continuing throughout development and later life cycle phases” (Friedenthal, Griego, and Sampson 2009, 6). Using models to represent the

system and its requirements, stakeholders can create a more precise, unambiguous, and consistent representation of them.

In MBSE, the requirements are captured and represented using graphical or mathematical models, which can be analyzed, simulated, and validated to ensure that they meet the needs of the user or system. This approach can help to improve the quality of the requirements, reduce the risk of errors and misunderstandings, and streamline the systems engineering process.

Subarna et al. introduced an MBSE approach to managing the requirements of an aerospace system using System Modeling Language (SysML) (Subarna et al. 2020). SysML is described by Friedenthal, Moore, and Steiner (2012) as “a general-purpose graphical modeling language that supports the analysis, specification, design, verification, and validation for complex systems” (29), making it a key enabler of MBSE. Subarna et al. applied an MBSE method to the fuel distribution system (FDS) of a conventional fixed-wing aircraft by modeling the life cycle of the FDS via a series of SysML Diagrams. These diagrams are visually associated with a validation and verification process to ensure better traceability from the eventual test results to the requirements.

Lebeaupin and Rauzy (2020) proposed a hybrid model consisting of various collections of text-based requirements and model-based descriptions. The first part of their idea was to link text in requirements with elements of models to create a more comprehensive and interconnected system specification. Treating the requirements as a hypertext specification, as suggested by their article, makes it easier to navigate through the different parts of the specification and understand the relationships between the various components. To implement this approach, requirements were linked with blocks in a SysML internal block diagram using hyperlinks between the requirements and the models (Lebeaupin and Rauzy 2020).

The second part of Lebeaupin and Rauzy’s idea is to produce parts of the requirements using controlled natural language (CNL) techniques to ensure that the requirements are clear, concise, and unambiguous (Lebeaupin and Rauzy 2020). CNL is a

subset of natural language specially designed to facilitate communication among humans. Generating parts of requirements using CNL can ensure that the requirements are structured in a way that is easier to understand and accurately convey the intended meaning. This can help to eliminate ambiguity and reduce the risk of misinterpretation or misunderstanding.

Overall, combining both parts of Lebeaupin and Rauzy's method, the aim is to use CNL to make it easier to ensure that the requirements accurately capture the intended meaning so that different stakeholders can easily understand them. Concurrently, models linked to requirements via hyperlinks seek to improve the quality of both requirements and models.

There are increasing studies on the adoption of MBSE, particularly in aerospace applications and platforms, with the objective of improving the accuracy, consistency, and traceability of the requirements. In contrast, studies on naval platforms and systems are less common.

C. INTEGRATED BRIDGE AND PLATFORM MANAGEMENT SYSTEM

The ship's integrated bridge system (IBS) is a critical component of modern marine navigation, communication, and safety systems. It is a comprehensive and integrated platform that combines various navigation, communication, and monitoring systems to provide a centralized and streamlined bridge operation. The ship IBS is designed to enhance the safety, efficiency, and accuracy of vessel operations by providing real-time information and decision support tools to the ship's crew. It enables seamless integration of various components such as radar, electronic chart display identification system (ECDIS), automatic identification system (AIS), global positioning system (GPS), voyage data recorder (VDR), and other critical navigation and communication systems, offering a single point of control for safe navigation (Figure 1). As such, the ship IBS is an essential system for modern maritime passage, and its proper installation, integration, and operation are critical to ensure the safety and efficiency of ship operations.

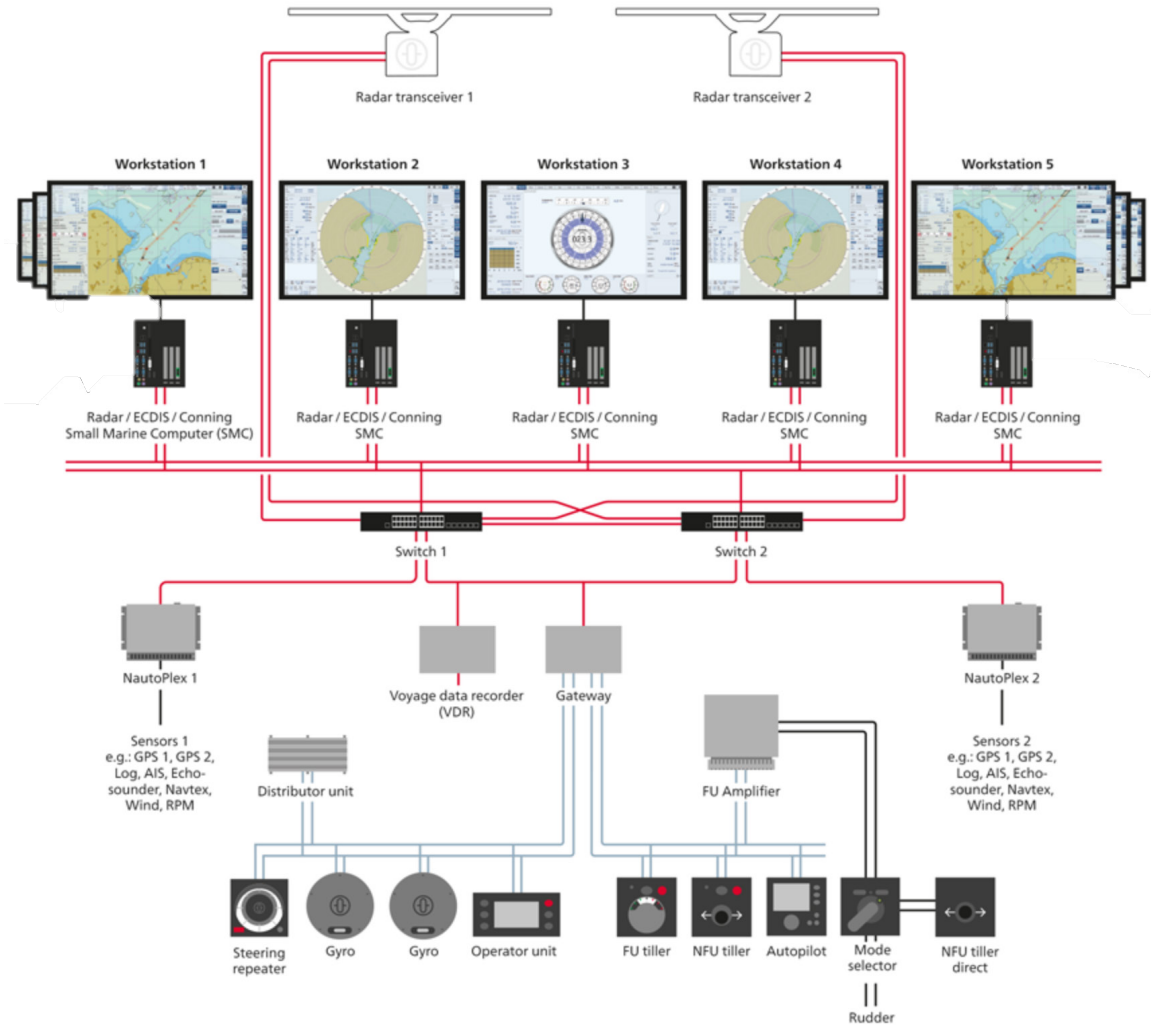


Figure 1. IBS typical architecture. Source: Anschutz (n.d.).

All IBS sub-systems can be controlled and monitored from the consoles in the bridge of the ship (Figure 2). In addition, these consoles also house the workstations of integrated platform management system (IPMS).



Figure 2. Bridge consoles that house IBS and IPMS workstations. Adapted from OSI Maritime Systems (n.d.).

The ship's integrated platform management system (IPMS) is a crucial element of modern ship automation and control systems. It is a fully integrated digital platform consolidating and streamlining shipboard functions such as propulsion, power generation, and auxiliary systems control. The ship IPMS integrates various critical ship systems, such as engine management, fuel monitoring, heating, ventilation, and air conditioning (HVAC) system, closed-circuit television (CCTV), firefighting and damage control, and ballast control systems, providing a comprehensive and centralized control system (Figure 3). By automating and streamlining these shipboard functions, the ship IPMS enhances the efficiency, reliability, and safety of ship operations while reducing crew workload. The ship IPMS also provides real-time monitoring and diagnostics capabilities that enable predictive maintenance and fault detection, reducing downtime and enhancing operational reliability. Hence, the IPMS is a vital component of the modern vessel to ensure efficient and reliable operation.

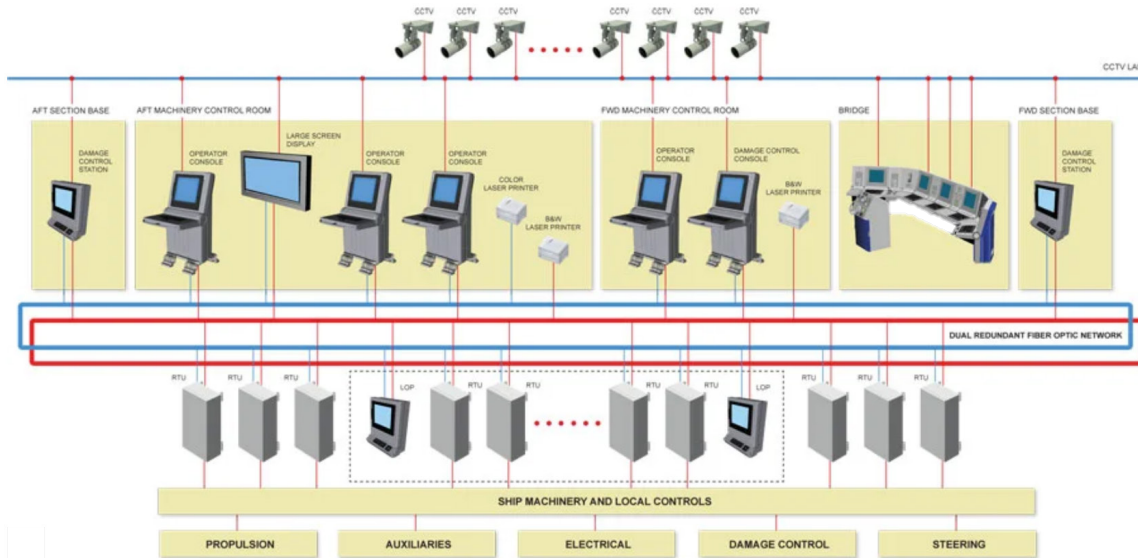


Figure 3. IPMS typical architecture. Source: L3Harris (n.d.).

There is a trend towards integrating ship’s integrated bridge systems (IBS) and ship’s integrated platform management systems (IPMS). The integration of these two systems creates a comprehensive ship automation and control platform that streamlines and centralizes shipboard functions, enhancing the efficiency and safety of ship operations.

By integrating the IBS and IPMS, ship operators can leverage the capabilities of both systems to enhance situational awareness, optimize vessel performance, and reduce operational costs. For example, integrating the navigation and propulsion systems enables the ship to maintain a more precise course, speed, and fuel consumption, resulting in improved efficiency and reduced emissions.

The need for greater efficiency and safety in maritime transportation drives the trend toward integrated IBS and IPMS systems. As shipping becomes more competitive and regulations become more stringent, ship operators seek ways to optimize their operations while maintaining high levels of safety and compliance. Integrating these two critical ship systems is one way to achieve these goals. This trend towards integrated IBS and IPMS systems is expected to continue as ship operators seek to improve the efficiency, safety, and sustainability of their operations.

The requirements of IBS and IPMS are complex and multifaceted, and managing them can present several challenges. Some of the key challenges include:

- **Complexity:** These systems are highly complex and require a deep understanding of the technology and the operational environment in which they will be used. This complexity can make it challenging to define and manage the requirements effectively.
- **Integration:** IBS and IPMS have a wide range of interconnected components, each with its specific requirements. Ensuring that all these components work together seamlessly can be a significant challenge.
- **Regulatory compliance:** The maritime industry is subject to a range of regulatory requirements, including those related to safety and environmental protection. Ensuring that the IBS and IPMS meet all these requirements can be challenging.
- **Interdisciplinary teams:** Developing and managing the IBS and IPMS typically require the involvement of interdisciplinary teams, including engineers, designers, and domain experts. Managing the requirements across these diverse teams can be challenging.
- **Changing requirements:** The requirements for IBS and IPMS can change over time as technology evolves and new operational challenges arise. Managing these changing requirements can be a significant challenge, particularly when the system is already in use.
- **Communication:** Effective communication is critical to managing the requirements of these systems. Ensuring that all stakeholders understand the requirements and are able to communicate effectively can be a challenge, particularly when stakeholders are geographically dispersed or have different areas of expertise.

Managing the requirements of a ship's IBS and IPMS requires a deep understanding of the technology, the operational environment, and the regulatory landscape, as well as effective communication and collaboration across interdisciplinary teams.

The current approach for communicating ship requirements to shipbuilders is predominantly based on document-centric methods that rely on the Statement of Work (SOW) and Ship Technical Specification documents. The SOW outlines the work to be performed by the contractor, including any necessary specifications on how it should be executed. On the other hand, the Ship Technical Specification document outlines the technical performance requirements that the ship must meet. Also included in the Ship Technical Specification are functional requirements, non-functional requirements, and regulatory requirements from major regulatory bodies like International Maritime Organization (IMO), Safety of Life at Sea (SOLAS), Institute of Electrical and Electronics Engineers (IEEE), and military standard (MIL-STD). For IBS and IBPMS, some of the sub-systems are mandated due to requirements from the regulatory bodies. Altogether, these requirements can add up to hundreds of artifacts for a typical project. The traditional way of documenting these requirements in a contract-style list resulted in a very long list that is arduous to visualize and manipulate. The primitive method of using tables to create matrices for tracing requirements led to huge traceability matrices that are equally difficult to handle (Lerche 2019). As such, requirements management software tools were developed.

Most requirements management software uses requirements traceability matrix (RTM). RTM is a tool that helps to ensure that all requirements are met. It does this by tracking the requirements throughout the development process and ensuring they are linked to the corresponding design, implementation, and test artifacts (Clark 2006).

However, this document-centric method is considered inefficient and susceptible to errors. MBSE can help the Navy and shipbuilders communicate requirements more effectively, leading to a more accurate and mutually agreed-upon contractual baseline (Tepper 2010).

III. PROPOSED ENHANCEMENT TO REQUIREMENTS MANAGEMENT

A. SYSTEMS ENGINEERING

Systems engineering is a vital discipline essential to designing, developing, and managing prevalent complex systems. It involves the integration of various engineering disciplines and other specialized fields to develop systems that meet the desired requirements, perform reliably, and are cost-effective. The application of systems engineering is widespread, and it is used in a variety of industries and domains. The benefits of systems engineering are numerous and can be observed at various stages of the system's life cycle. Systems engineering plays a vital role in ensuring the safety, effectiveness, and reliability of the systems.

B. CURRENT SYSTEM ENGINEERING APPROACH

The systems engineering discipline recognizes the limitations of the document-based approach to capture system specifications and design information. These artifacts encompass various elements such as “requirements, interface control documentation, and system architecture design descriptions” (SEBoK Contributors 2023, 22). However, there are no linkages between this information, which is contained in separate documents of different formats. Systems engineers have consistently utilized diverse types of models to enhance their work. Figure 4 and Figure 5 show the typical functional and physical hierarchy diagrams, respectively, which system engineers use to represent the functional and physical decomposition of a system. No linkage exists between these two diagrams.

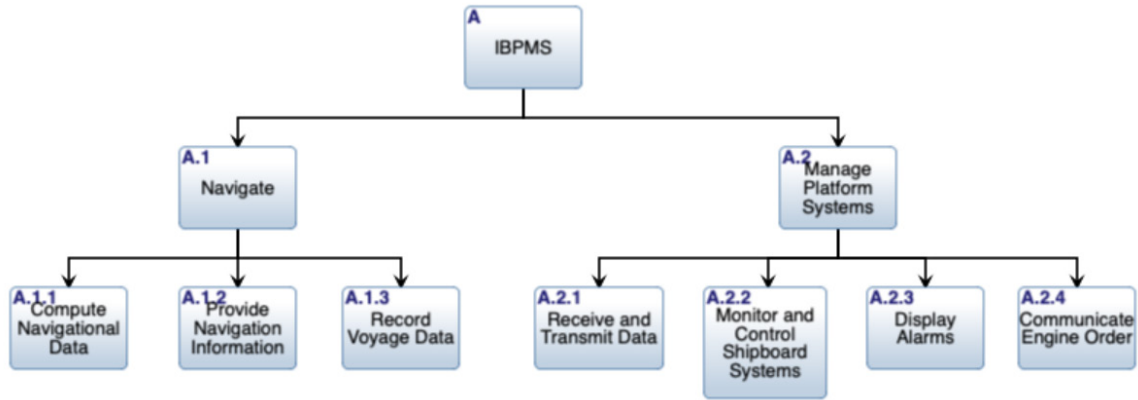


Figure 4. A typical functional hierarchy diagram

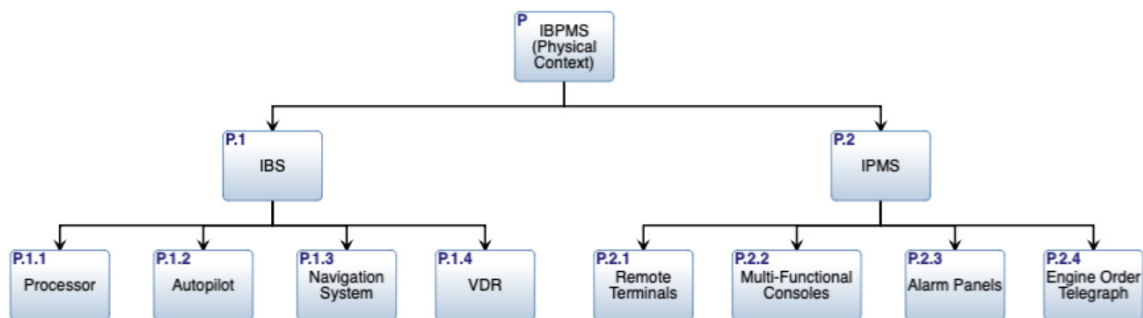


Figure 5. A typical physical hierarchy diagram

This document-centric approach in systems engineering encounters challenges such as imprecise information, inconsistencies between artifacts, and limitations in the reusability and maintainability of information (SEBoK Contributors 2023). Additionally, the proliferation of highly complex systems that are interdisciplinary in nature further accentuates the need for a more effective systems engineering method to facilitate collaborative work across various engineering disciplines and operations (Subarna et al. 2020). Furthermore, the emergence of more intelligent systems and products raises the demand for a paradigm shift towards a more integrated and collaborative approach.

C. APPLYING MODEL-BASED SYSTEMS ENGINEERING

MBSE is a methodology that uses models to design and manage complex systems. This approach involves creating and analyzing models that represent various aspects of a system, including its requirements, design, behavior, and functionality. MBSE is a powerful approach to systems engineering that allows engineers and other stakeholders to collaborate more effectively by providing them with an integrated conceptual framework for understanding the system.

One of the key advantages of MBSE is its ability to facilitate communication and collaboration between different engineering disciplines working on a project. By using models to represent the system, stakeholders from different disciplines can have a common understanding of the system's design and behavior. This shared understanding helps to prevent misunderstandings and reduces the risk of errors and inconsistencies. It also allows stakeholders to provide feedback and make informed decisions about the system's design, leading to a more effective and efficient system.

Another benefit of using MBSE is improved reusability. By representing the system as a model, parts of the system can be reused in future designs. This leads to cost savings and reduces the time and effort required to design new systems. Additionally, reusing parts of the design that have already been tested and validated increases the reliability of the new system.

Improved traceability is a third benefit of using MBSE. Tracing requirements, design decisions, and system behavior is easier when the system is represented as a model. This traceability helps to ensure that the system meets the desired requirements and goals. It also helps identify any issues arising during the design and implementation process, allowing for timely and effective solutions.

Furthermore, MBSE offers a holistic approach to system design and analysis that enables engineers to consider the system as a whole rather than as disconnected components. The application of MBSE is particularly beneficial for complex system development, where keeping track of various components and interdependencies can become overwhelming without an integrated approach.

In summary, MBSE is a methodology that uses models as a basis for the design, analysis, and implementation of complex systems. It has several benefits, including improved communication among stakeholders, better reusability, and enhanced traceability. These benefits help to ensure that the system meets the desired requirements and goals, leading to a more effective and efficient system. Therefore, they are used to gauge the effectiveness of the application of MBSE on requirements management.

D. MEASURES OF EFFECTIVENESS

In systems engineering, the measure of effectiveness (MOE) is a critical tool used to evaluate the effectiveness of a system in meeting its intended goals and objectives. The MOE is a quantitative measure that provides stakeholders with an objective assessment of the system's performance and helps to identify areas where improvements can be made.

In this thesis, MOE is used to define the key performance indicators that determine the effectiveness of adopting MBSE in handling the requirements of a complex system. INCOSE has presented numerous benefits of MBSE, including “improved communications,” “increased ability to manage system complexity,” “improved product quality,” improved traceability, and “enhanced knowledge capture” and reusability (Friedenthal, Griego, and Sampson 2009, 7). Out of these, three are selected as the research objectives with specific metrics to determine their effectiveness.

1. Improved Communications between Stakeholders

System modeling is a powerful method for comprehending and communicating complex systems. MBSE provides a significant advantage by facilitating clear communication among all stakeholders using a common language that is widely understood. Employing a unified system design language can overcome communication gaps between stakeholders with diverse backgrounds and levels of experience. The challenge of interpreting and clarifying requirements is frequently one of the most challenging aspects of the systems engineering process. MBSE can alleviate uncertainty and stimulate the entire program team to attain consistent thinking and communication (Tepper 2010). The ability to communicate clearly and effectively is critical to the success of any system design project. Using MBSE, engineers and stakeholders can work

together to develop a shared understanding of the system design and ensure everyone is on the same page regarding requirements, specifications, and priorities. This can help to mitigate ambiguity, promote consistency, and improve the quality of the system design, ultimately leading to a more efficient and effective system. Often, miscommunications arise between operators (i.e., the Navy and their representatives) and shipbuilders due to the ambiguity in the requirements. This issue can be mitigated with MBSE.

It is challenging to establish a quantifiable metric for communication due to its subjective nature. One potential method is to conduct a stakeholder perception survey to gauge stakeholders' views. Rating scales or structured survey questionnaires can be utilized to assign numerical values to stakeholders' responses and derive quantitative insights.

Besides a survey, other non-quantitative methods would also be helpful in the effectiveness analysis. Non-quantitative factors such as the accuracy of the information and linkage of different views would be studied to ascertain their impact on the communications among stakeholders.

2. Reusability of Requirements

INCOSE highlighted the ability of MBSE to “enhanced knowledge capture and reuse of the information by capturing information in more standardized ways” (Friedenthal, Griego, and Sampson 2009, 7). Information such as requirements can be modeled to create reusable requirement components or modules, making them more easily adaptable and reusable across different projects.

In the design of a naval ship, it is of utmost importance to fulfill regulatory requirements. Compared to functional or system requirements, regulatory requirements are updated less frequently. Regulations like IMO, SOLAS, IEEE, etc., are also applicable for a range of ships of various classes and tonnages. Therefore, the developed requirements models will be platform-independent and possible for reuse across different naval shipbuilding projects.

Although functional or system requirements are more likely to experience rapid changes with technological advancement, the fundamentals of essential systems such as navigation systems are well established, and any changes are usually evolutionary rather than revolutionary. The addition of cybersecurity capabilities to a navigation radar system will lead to additional requirements, but it will not affect the baseline functional requirements of the system.

The reusability of requirements would be examined through some of the subsystems in the IBPMS. This assessment would explore the number of reusable requirements.

3. Traceability of Requirements

Requirements traceability is an essential aspect of the MBSE process, ensuring that each requirement is accounted for throughout the system development lifecycle. Traceability allows tracking individual requirements from initial conception to the system design, implementation, testing, and final deployment. This ensures that all requirements are satisfied, and potential issues are identified and addressed early on.

Throughout the construction phase of a naval ship, numerous tests and trials are conducted at different phases and milestones as checkpoints before proceeding to the next stage. For each trial, an approved test procedure is used for conducting the test and verifying that the system under test functions as intended and meets the operational requirements of the ship. One of the most critical checkpoints is the sea acceptance trial (SAT) which is conducted on the open sea to evaluate the performance, functionality, and reliability of the ship and its associated systems. During the SAT, the ship and its systems are subjected to a range of operational conditions, including different sea states, weather conditions, and operating scenarios, to verify that they meet the required performance and safety standards.

The importance of the SAT cannot be overstated, as it is the final step in ensuring that the ship and its systems meet the required standards for safety and performance. By conducting a thorough SAT, shipbuilders, and operators can ensure that the ship and its

systems are safe and reliable and that the operational requirements of the ship and its crew are met.

However, the scope of the sea trial is typically formulated following historical trial procedures established years ago based on experience and refined throughout the years from lessons learned. With the evolution of requirements over time, these trial procedures may no longer be adequate to capture all the required tests unless proper linkages are established between the tests to the requirements. The application of MBSE to provide a linkage, allowing verification tests to be traced back to the requirements, is explored.

The effectiveness of the linkage between a requirement and its corresponding verification test is evaluated using MBSE tools within Innoslate. Tools such as traceability matrix and appropriate diagrams would map requirements to verification tests. The MOE is based on the completeness of the traceability matrix by measuring the percentage of requirements successfully linked to corresponding verification tests.

E. APPLICATION OF MBSE TO REQUIREMENTS MANAGEMENT

The design of ships has generally followed a document-based approach which entails creating textual technical specifications, design drawings, and design and interface control documents to capture all the system requirements (Tepper 2010). Given the complexity of an extensive system like a ship, the requirements and system specifications are usually documented in a contract-style document. This document is segregated into different sections, which are managed by design engineers who are the subject matter expert in the respective sections. The design engineers are responsible for managing the documents, establishing their accuracy, completeness, and consistency, and ensuring they fulfill the requirements. Typically, requirements management software tools are used for the task of managing the requirements. However, it can be simply based on the individual design engineer's prerogative and experience without a well-defined methodology or process, which can vary from project to project.

The current manual traceability process between requirements and ship design is inefficient and error-prone. The requirements database does not have any linkage to the

design documents forcing engineers to manually track requirements across multiple documents. This can be time-consuming and error-prone, as it is easy for information to be lost or misinterpreted.

In addition, the document-based approach makes it difficult to understand the overall system architecture and challenging to trace changes and assess the associated impact. This can lead to costly mistakes during the design and construction process.

In the next section, MBSE methodology will be used to examine how the requirements for the IBPMS of a naval ship could be modeled in Innoslate and explore whether the model could benefit from the advantages of MBSE.

F. DEVELOPING THE SYSTEM MODEL

Naval ships are complex machines that require advanced technological systems to operate efficiently and safely. To develop these advanced technological systems for naval ships, models can be utilized to evaluate and optimize designs before implementation and assist with design by managing requirements and the relationships between them. Traditional ship design techniques have been successful. However, with the increasing complexity of modern naval ships, a modeling approach could be adopted to alleviate the burden on the ship designer. This modeling approach is realized through Innoslate, an MBSE tool.

1. Innoslate

Innoslate is a powerful systems engineering software tool developed by SPEC Innovations. It is designed to support the entire system lifecycle, from concept development and requirements engineering to design, verification, and validation (SPEC Innovations 2022). The capabilities of Innoslate include creating and managing requirements, developing system architectures, conducting trade studies, generating test plans, tracking defects, and managing risks. This comprehensive approach to systems engineering aims to ensure that all aspects of the system are accounted for and integrated throughout the entire lifecycle, leading to better system performance and reduced risk of errors or oversights. Innoslate provides a single platform for all aspects of systems

engineering to promote collaboration and communication across teams, helping to streamline processes and increase efficiency.

A “Dashboard” will be created for each Innoslate project (Figure 6). The Innoslate Dashboard is a central component within the Innoslate platform that serves as a command center for users to access and manage their projects efficiently. It provides a visually intuitive and customizable interface that offers a comprehensive overview of the project’s key information and metrics. The Dashboard acts as a hub, allowing the monitoring of project progress, tracking milestones, and staying up to date with the latest activities. The Innoslate Dashboard enables workflow to be streamlined by providing easy access to essential project elements, such as task lists, requirements status, system architecture, test coverage, and risk assessment. By default, the “Activity Feed” widget will appear on the right of the Dashboard. It displays all recent activities with the latest updates and changes made within the project, allowing all authorized users to stay informed about ongoing activities.

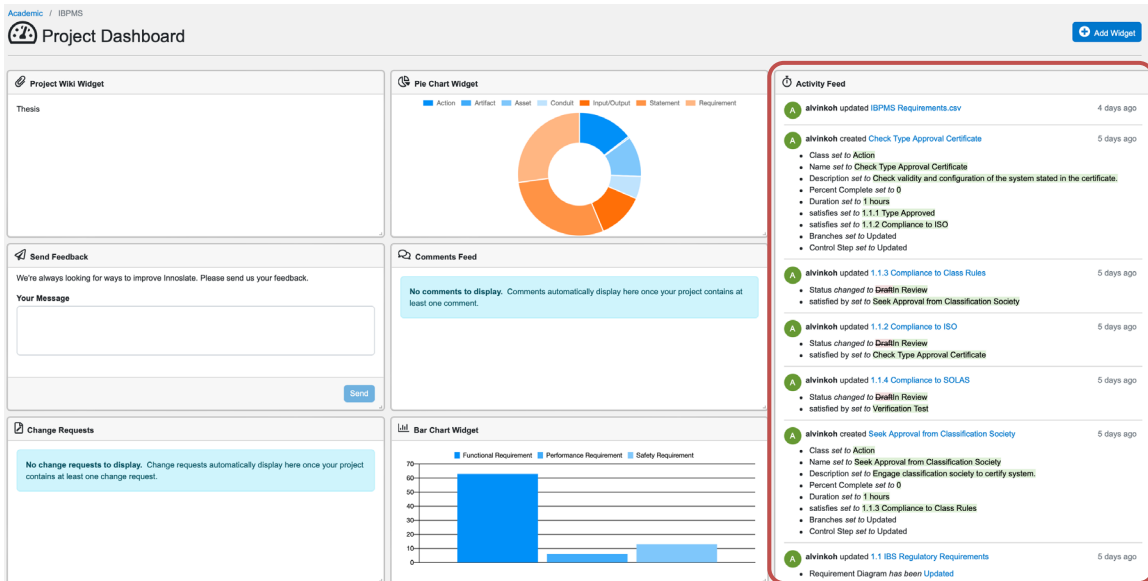


Figure 6. Innoslate Dashboard provides information for monitoring status.

Innoslate offers a robust project-sharing function that enables seamless collaboration and communication among project stakeholders. The project-sharing

function provides various capabilities and controls to manage project access and permissions. Users can specify the level of access granted to each user, such as read-only, edit, or administrator rights. This ensures that sensitive or critical project information is protected while allowing collaborators to contribute effectively. Users granted access to a shared project can view and edit project artifacts, make comments, and track project changes in real-time. Innoslate offers version control to track and manage different project iterations, allowing users to revert to previous versions if needed. Furthermore, Innoslate supports concurrent collaboration, allowing multiple users to work on the same project simultaneously. This enables teams to collaborate efficiently, avoiding conflicts and ensuring that everyone is working on the most up-to-date version of the project.

Innoslate utilizes two modeling languages, “the Lifecycle Modeling Language (LML) and Systems Modeling Language (SysML), to describe its concepts and diagrams. LML was developed to simplify the elements, relationships, attributes, and diagrams used in systems engineering and project management” (SPEC Innovations 2022). This facilitates the understanding and communication of complex systems engineering concepts and processes. SysML is capable of providing a standardized way to visualize diagrams. By using these modeling languages, Innoslate can represent and communicate system information and behavior in a consistent and standardized way.

The “V” diagram in Figure 7 represents the systems engineering process. Although MBSE can be utilized to replicate this model to represent a system’s entire lifecycle, this thesis only focuses on the system-level and sub-system requirements.

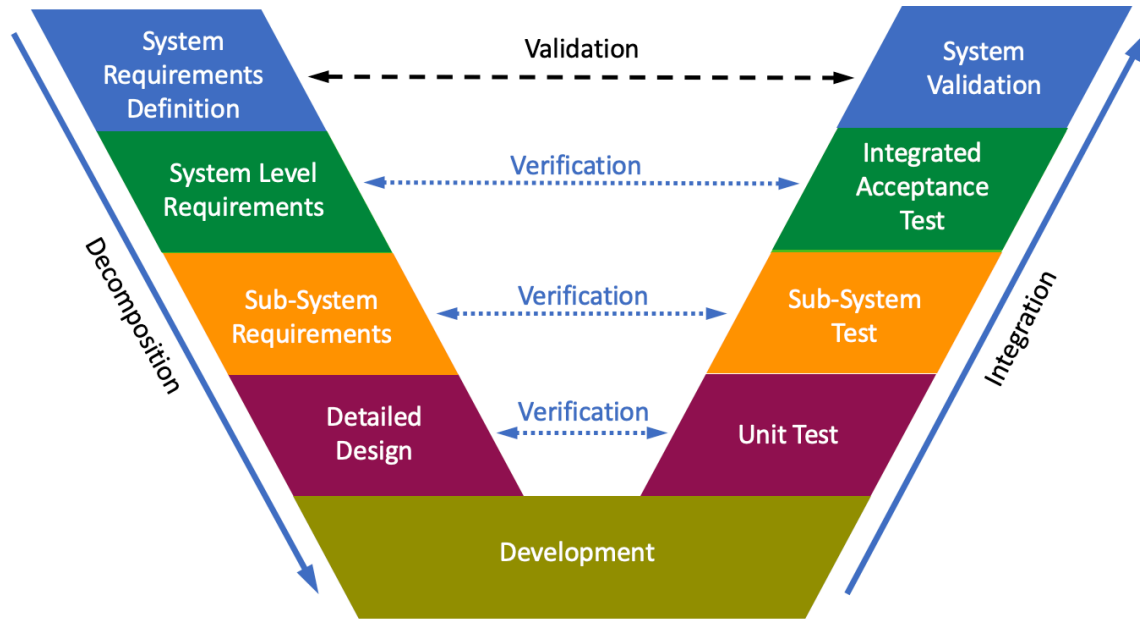


Figure 7. Systems engineering “V” diagram

A modified “V” diagram, shown in Figure 8, illustrates the scope of the model to be developed. From the system-level and sub-system requirements in the initial system engineering “V” diagram, the modified “V” diagram focuses only on requirements and expands to include project and regulatory requirements.

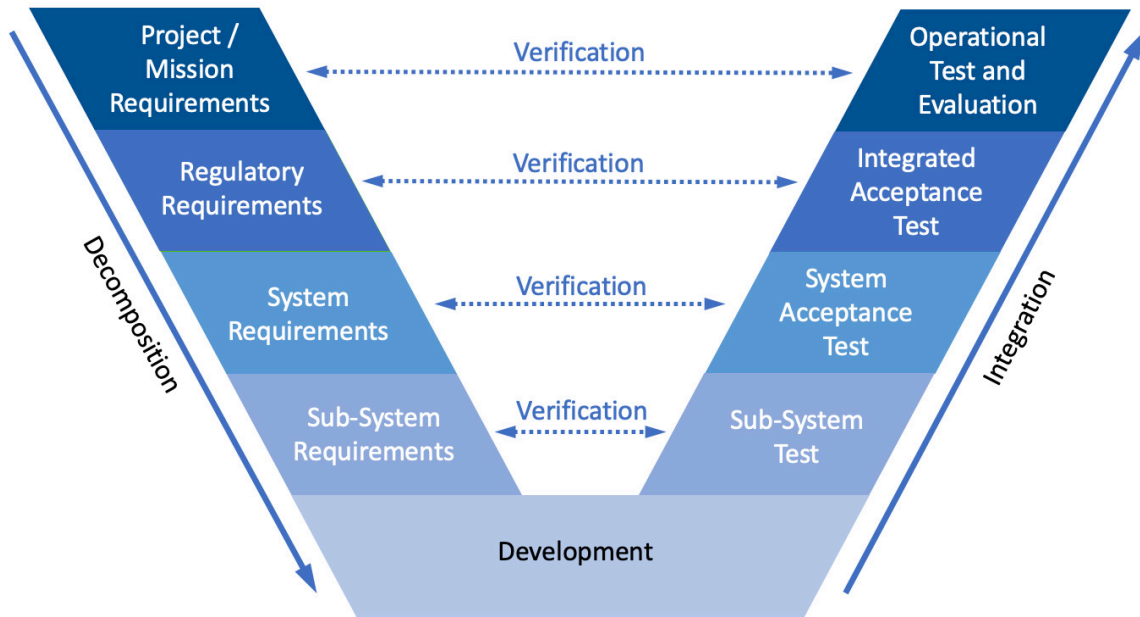


Figure 8. Modified “V” diagram

2. Methodology for Model Development

A full fledge model of the IBPMS extends beyond the scope of this thesis. Instead, a simplified model with the majority of IBPMS sub-systems, which covers most of the critical functions, is developed to explore the feasibility of MBSE in meeting the MOEs.

An LML model is first created in Innoslate within the project’s database. This LML model is constructed in the background and is accessible to each view within Innoslate, which can interpret and display the information in various formats. All diagrams, documents, and other components created or edited are captured and updated into the LML model. The LML model being generated consists mainly of seven categories of entities (Figure 9) that establish the foundation of any system design.

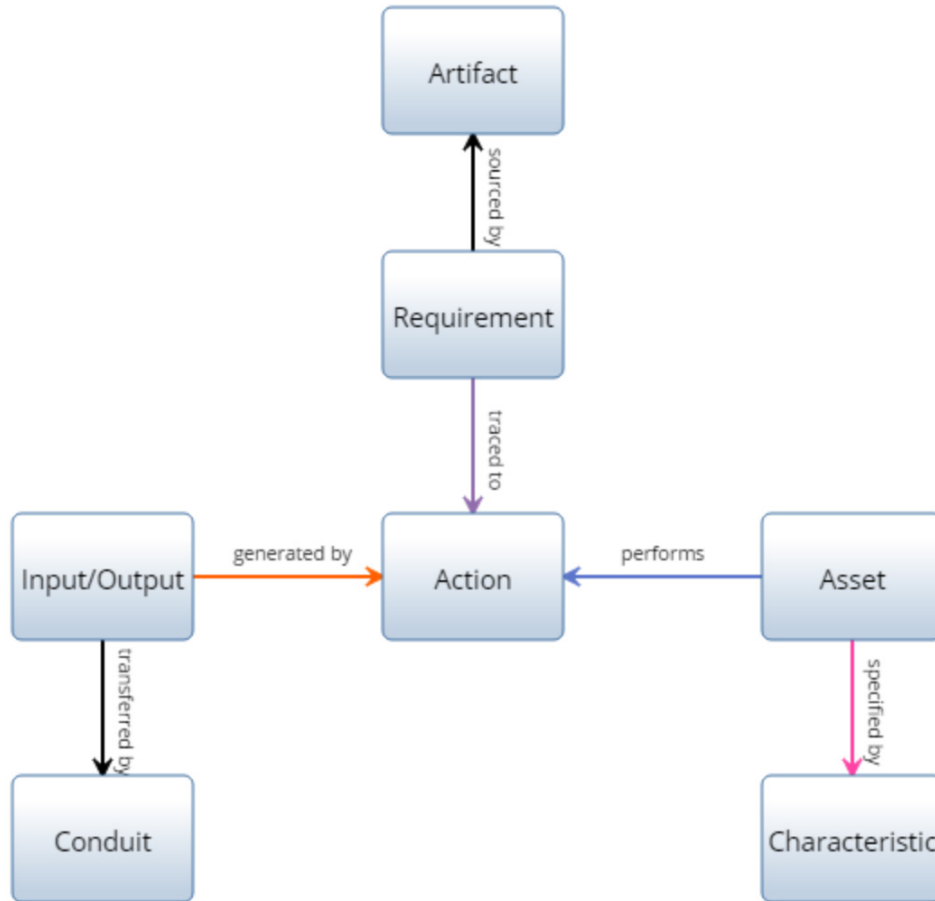


Figure 9. Innoslate system architecture. Source: SPEC Innovations (2022).

The element definitions from the LML specification are as follows (SPEC Innovations 2022):

- A "Requirement" entity captures the desired capabilities, functionalities, constraints, and performance criteria the system or its components must satisfy. Requirements define what the system should do or how it should behave.
- An "Artifact" entity represents any tangible or intangible item associated with the system. It can include documents, drawings, models, software, or any other deliverable or work product created during the system's development or operation.

- An "Action" entity represents the behaviors or actions performed by the system or its components. It includes processes, operations, algorithms, or any other form of activity that contributes to the system's functionality.
- An "Asset" entity specifies the physical or virtual resources utilized by the system to perform an action. It can include equipment, materials, data, information, or any other resource that has value within the context of the system.
- An "Input/Output" entity captures the inputs and outputs of the system or its components. It represents the data, signals, or information the system receives as input and the results, outputs, or responses produced by the system.
- A "Conduit" entity specifies the channels, paths, or mediums through which information, energy, or material flows within the system. It includes communication channels, physical connections, interfaces, or any other means of transfer or exchange.
- A "Characteristic" entity describes the qualities, attributes, or properties associated with the system or its components. It can include both qualitative and quantitative characteristics, such as reliability, performance, safety, usability, or any other relevant attribute.

The critical view to be developed in Innoslate is the requirement diagram to illustrate how requirements can be modeled. Accompanying the requirement diagram are "hierarchy chart," action diagram, and "traceability matrix."

3. Modeling the Requirements

The SysML specification defines a requirement as a statement that "specifies a capability or condition that must be satisfied, a function that a system must perform, or a performance condition a system must achieve" (Friedenthal, Moore, and Steiner 2012, 317). Requirements can also specify performance conditions that a system must achieve.

In the context of systems engineering, requirements play a critical role in ensuring that the final product or system meets the needs and expectations of stakeholders. Requirements can be elicited from various sources, including customer needs, regulations, standards, and design constraints.

Requirements are typically captured in a requirements document, which may include a list of high-level system requirements and more detailed requirements that specify the behavior and performance of individual system components. These requirements serve as a basis for system design, testing, and verification, and they are used throughout the system development lifecycle to ensure that the system meets its intended objectives.

Innoslate provides a range of constructs for representing requirements, including requirements diagrams, requirement documents, and requirement relationships. These constructs are designed to provide a comprehensive representation of a system's specifications and constraints. The requirements modeling capability in Innoslate allows requirements to be modeled and visualized in a structured and systematic way, making it easier to manage complex requirements and ensure that they are consistent, complete, and traceable throughout the system development lifecycle. Requirements can be linked to specific system elements and components, allowing the impact of changes to be analyzed and traced back to their source.

Using Innoslate's capability to manage the requirements and their configurations ensure that the requirements are always aligned to the system model. Changes to the requirements can be tracked and managed, and the impact of those changes on the system model can be analyzed and managed accordingly. This helps to ensure that the system development process remains on track and that the final system meets its intended objectives and requirements.

To capture all the requirements for IBPMS would be too enormous for this thesis. Instead, three sub-systems within the IBS were selected, and their requirements were imported into the database of Innoslate. The three selected sub-systems are the warship electronic chart display identification system (WECDIS), autopilot, and navigation radar

system (NRS). Also included are the high-level regulatory standards the IBS must conform to. These imported requirements can be displayed in the document view, as shown in Figure 10. The organizational structure within Innoslate utilizes labels to arrange the requirement models. Each model can be tagged to multiple labels. The regulatory requirements are labeled under the category of Safety Requirement as the regulations are generally associated with safety-related standards. Other requirements can be labeled as Environmental Requirement, Functional Requirement, Interface Requirement, Performance Requirement, Reliability Requirement, Safety Requirement, Verification Requirement, or a combination of them. Figure 11 shows one of the requirements for “Heading Keeping Mode,” which describes the autopilot system and has two labels. For a complex system, such as a ship, with thousands of requirements, having a highly organized structure is very important during model development.





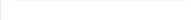
Entity	Quality Score	Labels	Rationale
1 Integrated Bridge System	N/A	No labels to display.	N/A
1.1 IBS Regulatory Requirements	N/A	No labels to display.	N/A
1.1.1 Type Approved The IBS shall be type approved.	S 75% 	Safety Requirement	
1.1.2 Compliance to ISO The IBS shall conform with the updated nautical safety standards as laid down in the International Standard (ISO8468).	S 75% 	Safety Requirement	
1.1.3 Compliance to Class Rules The IBS shall conform with the nautical safety standards as laid down in the class rules.	S 75% 	Safety Requirement	
1.1.4 Compliance to SOLAS The IBS shall conform with the nautical safety standards as laid down in SOLAS.	S 75% 	Safety Requirement	
1.1.5 Compliance to IMO The IBS shall conform with the nautical safety standards as laid down in the IMO.	S 75% 	Safety Requirement	

Figure 10. Innoslate requirements document view showing the list of requirements


Entity	Quality Score	Labels
1.3.6 Heading Keeping Mode The autopilot system shall be capable of maintaining the Vessel at a constant heading at up to Sea State 5 with Heading deviation < 3° for 90% of the time.	63% 	Functional Requirement Performance Requirement

Figure 11. Autopilot requirement with multiple labels

While the document view would be useful for adding labels and conducting quality checks to the requirements (outside this thesis's scope), other views graphically depict the requirements in various diagrams. These diagrams can illustrate the requirements and their relationship to one another.

4. Requirement Diagram

The parent-child relationship is an important aspect of requirements management as it organizes and structures requirements hierarchically (Dam 2023). Higher-level requirements act as parents for more specific and detailed child requirements in this relationship. Innoslate creates this relationship automatically by generating the requirement diagram using the imported requirements already residing in the Innoslate database. Figure 12 shows the requirement diagram of IBS as the parent requirement with four child requirements. The requirement diagram provides a visual representation to facilitate understanding and analysis of the relationships between requirements within a system. It provides a graphical view of how requirements are interconnected and how they contribute to the overall system functionality.

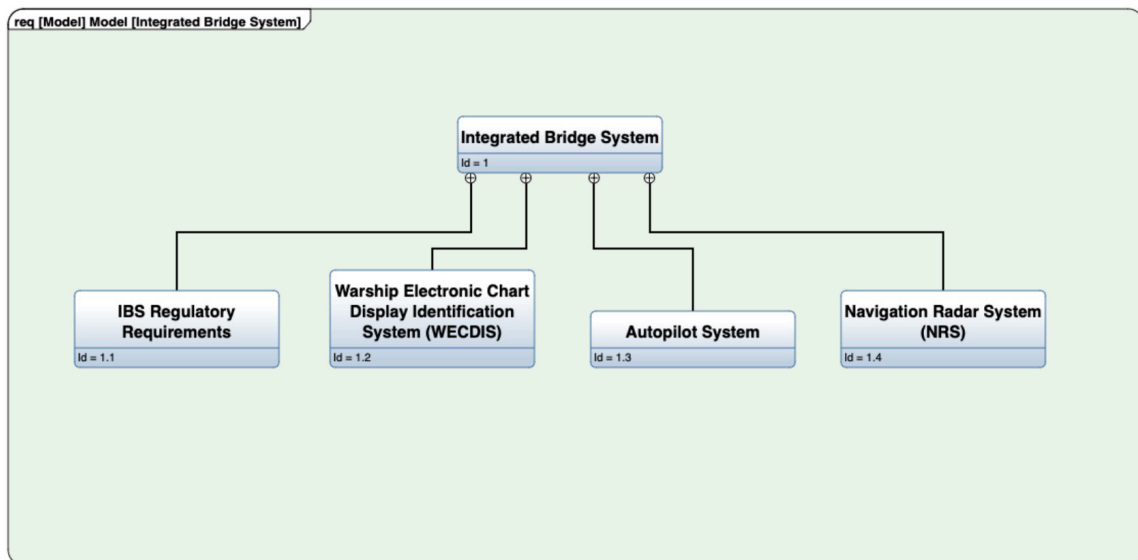


Figure 12. Requirement diagram of IBS

In the requirement diagram, the “requirement construct” refers to representing individual requirements as elements within the diagram. Each requirement construct represents a specific requirement within the system being modeled. The requirement construct (Figure 13) consists of a block that contains information related to the requirement. The name of the requirement is shown in the top section, and the identifier, text, and rationale associated with the requirement are shown in the bottom section (SPEC Innovations 2022).

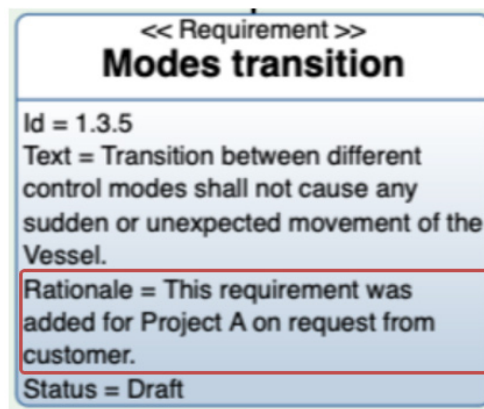


Figure 13. A requirement construct with information on the requirement

The identifier, text, and rationale in the Requirement Construct are essential components that provide valuable information and serve specific purposes. The identifier is a unique identifier assigned to each requirement. It helps in uniquely identifying and referencing a specific requirement within the system. It also shows the relationship to its parent or child requirement, making it easier for stakeholders to communicate and track requirements across different diagrams and documents. The text component of the requirement construct represents the actual requirement statement or description. The rationale provides the reasoning or justification behind the requirement. The rationale helps stakeholders understand the purpose and significance of the requirement, giving context and guiding decision-making processes.

Moreover, it provides a place for the reasoning and consideration to be documented whenever there are amendments to requirements. This is useful for tracking

changes, especially when the requirements are being reused for subsequent projects. It can also lead to better communications among stakeholders as any new team member can view and understand based on the rationale provided.

5. LML Diagrams

The IBPMS was modeled using LML diagrams to realize the benefits of MBSE. One of the LML diagrams is the action diagram, which depicts the functional elements of the system. It provides a visual representation of the sequence of actions and activities within a system. The action diagram can analyze the flow of actions, their dependencies, and the overall behavior of the system. Figure 14 shows the action diagram of IBPMS with Action and Input/Output (I/O) entities represented by blocks and green parallelograms, respectively. The action diagram also allows the actions to be assigned to the assets that perform the actions. At the highest level, the IBPMS model consists of IBS and IPMS, the two assets that perform the “Navigate” and “Manage Platform” actions. Both actions are decomposed further to provide more resolution to the system. The decomposed action diagrams are shown in Appendix B.

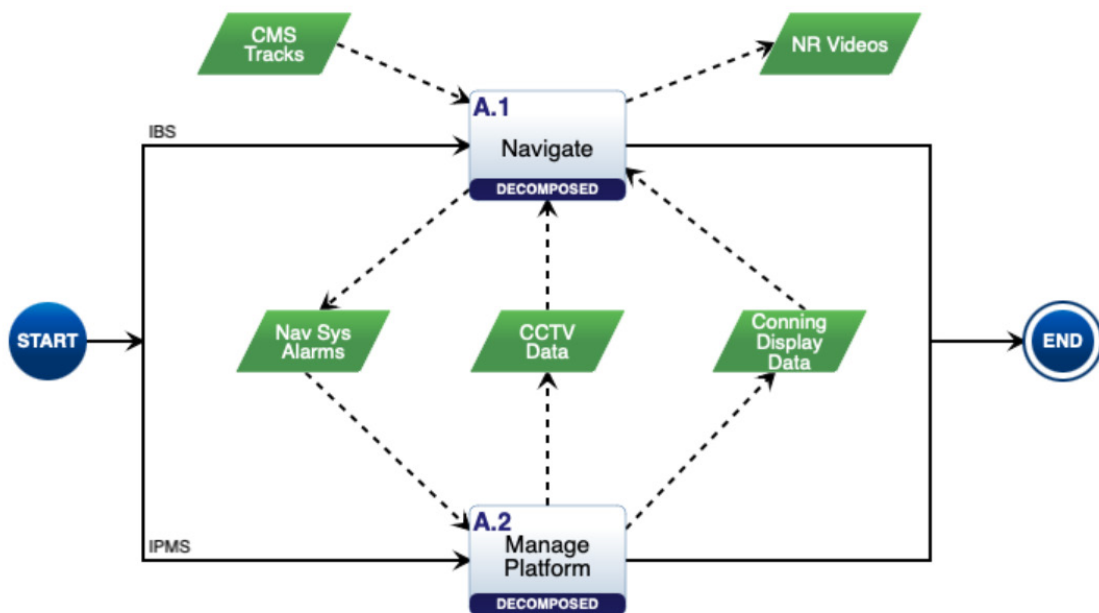


Figure 14. Action diagram of IBPMS

There are three main groups of data exchanges between IBS and IPMS, as shown by the three I/O entities between them. The dotted lines with arrows represent the data flow from one Action block to the other. Various navigation systems within the IBS (Figure 1) integrate to perform the Navigate action. Each navigation system will output an alarm when a fault occurs and send this alarm to IPMS to be recorded in the error log. This is an essential data transfer from IBS to IPMS, and it would be used to demonstrate the traceability feature of MBSE. Both CCTV data and conning display data are transferred from IPMS to IBS. The combat management system (CMS) tracks, and NR videos refer to data exchanged between the CMS and IBS. CMS is not modeled as it is an external system, and modeling the input and output data to IBS is sufficient to capture the major functions. The CMS tracks refer to targets acquired by the combat management system, which are sent to IBS for display on the radar picture display at the bridge. NR videos are navigation radar pictures that are sent to CMS.

Another LML diagram is the “spider” diagram (Figure 15), which is a type of hierarchical organizational chart used for visualizing traceability. The blocks in this

diagram represent the various entities, such as Actions, physical Assets, and Input/Output. The relationships between the blocks are represented by the lines with the color denoting the relation type. All the relationships depicted in Figure 14 are derived from the database and shown in Figure 15 (blue and purple lines). The blocks in green are I/O entities that exactly correspond to the parallelograms in the action diagram, with the blue and purple lines representing the relationship between the I/O entities and Action entities. The Action entities are denoted by their ID with the prefix A. The spider diagram also shows the physical Assets (yellow boxes) that perform the actions. Apart from showing the relationships across various entities, the spider diagram shows the decomposition of each entity type from the top to the lowest level. The decomposition of Action entities from IBPMS to the next two levels is shown in Figure 15. The spider diagram can be further expanded to the lowest level to display all entities and their linkages, as shown in Appendix D. This diagram facilitates traceability by allowing the ease of identifying the connection among the entities. It also improves impact assessment when changes to any entities arise. While the spider diagram unveils the underlying relationship among all the entities, another tool displays the link between the requirements and the entities.

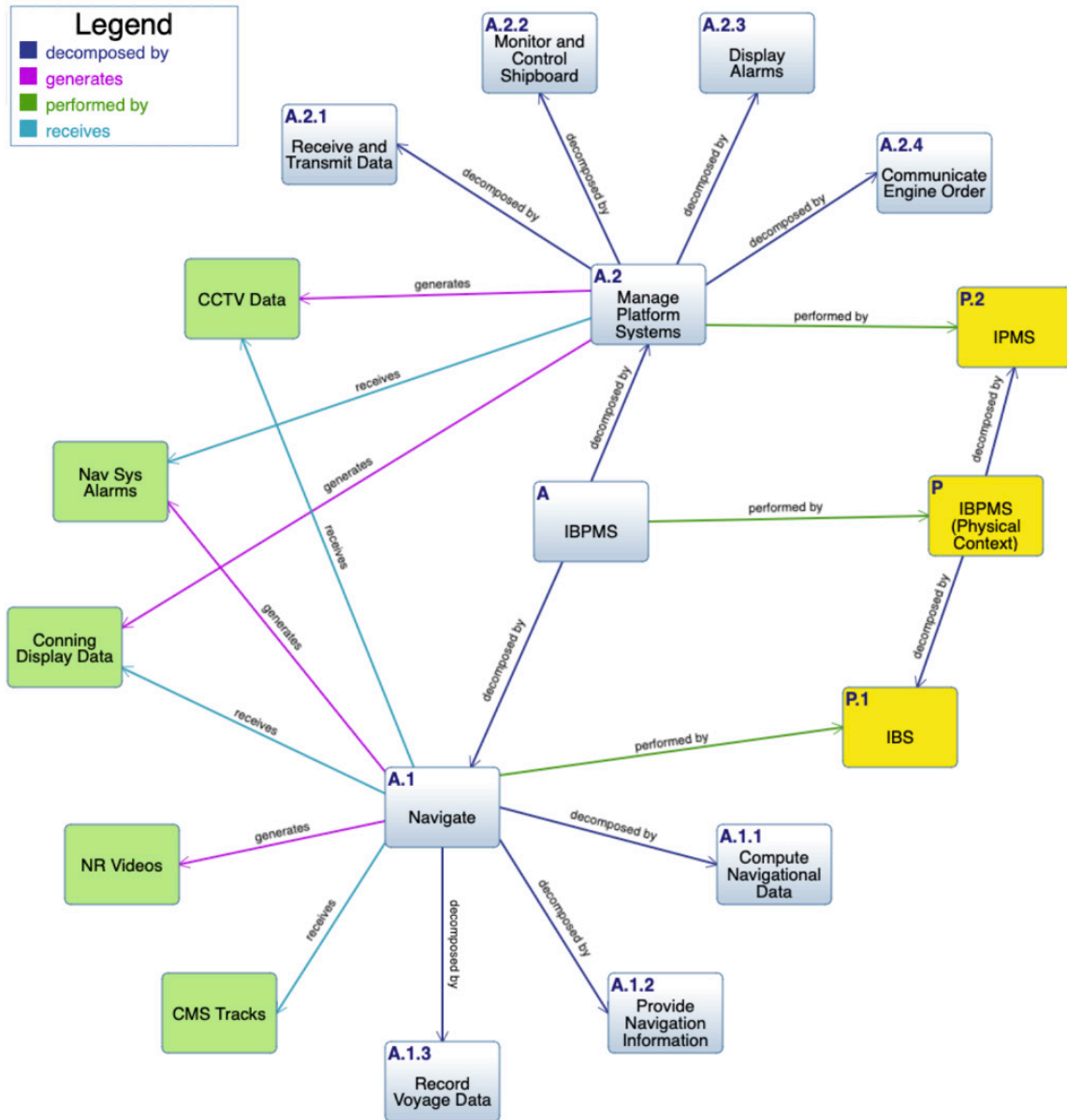


Figure 15. Spider diagram depicting the relationship of the entities

6. Traceability Matrix

The traceability matrix is used for visualizing the relationships and traceability between the different elements of the system model. It provides a structured overview of how various artifacts, such as requirements, actions, assets, and characteristics, are linked to one another. It is particularly useful for establishing traceability between requirements and other entities.

The traceability matrix consists of a grid-like structure, with the rows representing the source artifacts and the columns representing the target artifacts. Each cell in the matrix indicates the traceability relationship between a specific source and target artifact. Figure 16 shows a traceability matrix, tracing the link between Requirements in the rows and Actions in the columns. The crosses indicate the Actions “traced to” the corresponding Requirements.

	A.1 Navigate	A.1.1 Compute Navigational	A.1.1.1 Control Ship Automa	A.1.1.2 Provide Digital Map	A.1.1.3 Ensure Crew Vigilanc	A.1.2 Display Navigation In	A.1.2.1 Receive Navigation In	A.1.2.2 Compute Satellite Sigr	A.1.2.3 Compute Ship Positio	A.1.2.4 Provide Time	A.1.2.5 Compute Ship Speed	A.1.2.6 Compute Depth	A.1.2.7 Compute Depth	A.1.2.8 Detect Object	A.1.2.9 Acquire Target	A.1.2.10 Track Target	A.1.2.11 Display Radar Pictur	A.1.2.12 Transmit & Receive
1.2 Warship Electronic Chart Dis...																		
1.2.1 Display charts				X														
1.2.2 Conform to international sta...																		
1.2.3 Chart standards																		
1.2.4 Type approved																		
1.2.5 Presentation standards																		
1.2.6 Content and display aspects				X														
1.2.7 Overlay targets								X					X	X				
1.2.8 Manual fixing																		
1.2.9 Route Planning																		
1.2.10 Projection and Dead Reck...								X										
1.2.11 History Path																		
1.2.12 Chart Correction																		
1.2.13 Measurement Vector																		
1.2.14 Planting of Temporary Ref...																		
1.2.15 De-cluttering Features																		
1.2.16 Display alarms				X														
1.2.17 User Profiles and Layers																		
1.2.18 Software update																		
1.2.19 Integration with Maritime A...																X		
1.2.20 Integration with Autopilot																		

Figure 16. Traceability matrix of WECDIS requirements vs. Actions

Apart from tracing the relationship to Action, a traceability matrix can also be generated to trace the requirements to Assets in the model. Figure 17 shows the

traceability matrix for WECDIS requirements against Assets. The complete traceability matrixes are shown in Appendix F.

	P.1 IBS	P.1.1 Processor	P.1.1.1 Autopilot	P.1.1.2 WECDIS	P.1.1.3 BNWAS	P.1.1.4 Conning Display	P.1.3 Navigation System	P.1.3.1 DONS	P.1.3.2 Speed Log	P.1.3.4 Echo Sounder	P.1.3.5 WAIS	P.1.3.6 INS	P.1.3.7 Meteorological Sensor	P.1.4 VDR	P.2 IPMS	P.2.1 Remote Terminals	P.2.2 Multi-Functional Console	P.2.3 Alarm Panels	P.2.4 Engine Order Telegraph
1.2 Warship Electronic Chart Dis...	X	X																	
1.2.1 Display charts	X	X	X																
1.2.2 Conform to international sta...			X																
1.2.3 Chart standards	X																		
1.2.4 Type approved			X																
1.2.5 Presentation standards			X																
1.2.6 Content and display aspects	X	X	X	X															
1.2.7 Overlay targets	X	X	X																
1.2.8 Manual fixing			X																
1.2.9 Route Planning			X																
1.2.10 Projection and Dead Reck...			X				X												
1.2.11 History Path			X																
1.2.12 Chart Correction	X	X	X																
1.2.13 Measurement Vector			X																
1.2.14 Planting of Temporary Ref...			X																
1.2.15 De-cluttering Features	X	X	X																
1.2.16 Display alarms			X	X													X		
1.2.17 User Profiles and Layers			X																
1.2.18 Software update			X																
1.2.19 Integration with Maritime A...			X																
1.2.20 Integration with Autopilot		X	X																

Figure 17. Traceability matrix of WECDIS requirements vs. Assets

7. Verification and Validation

The Test Center is a dedicated module within the Innoslate MBSE platform that supports the management and execution of testing activities for verification and validation (V&V) purposes. Test planning and management are conducted to create and manage test plans and define the scope, objectives, and strategies for testing. Test cases, test procedures, and test scripts are specified and organized into test suites for efficient test execution. The Test Center provides a centralized location to document and track all testing-related information. Figure 18. shows a view of the Test Center with the SAT plan

for autopilot. Each row represents a test case for the verification of the system requirements.

Entity	Expected Result	Actual Result	Status
T1 SAT - Verify Wheel over alarm Approach a waypoint on a route to below the alert limit.	Alert		Not Run
T2 SAT - Verify course change alarm Approach a waypoint on a route to below the alert limit.	Alert		Not Run
T3 SAT - Verify cross track error alarm Reduce the cross track error alarm limit to a value below the current value.	Alert		Not Run
T4 SAT - Change of operational modes test Once the autopilot is engaged and an active route is available, operate the autopilot operational keyboard to change from heading control, course control and track control.	The change of operational modes (i.e. between heading mode, course mode and track mode) works as intended.		Not Run
T5 SAT - Override function test Operate the change-over mechanism to engage manual steering while in heading control mode, course control mode and track control mode.	Manual steering.		Not Run
T6 SAT - Verify Heading Control Mode Travel the defined heading at 20 knots, or at a speed to be mutually agreed during the test, depending on the weather conditions. Observe the heading deviation on the conning display, and for each leg, record the heading setpoint. At the end of the leg, extract the logfile for the actual heading for later analysis, if required.	Ship heading with < 3 deg deviation, 90% of the time.		Not Run
T7 SAT - Verify Course Control Mode Travel the defined course at 20 knots, or at a speed to be mutually agreed during the test, depending on the weather conditions. Observe the cross-track deviation on the conning display. At the end of the leg, extract the logfile for the actual cross-track for later analysis, if required.	A cross-track error of less than 20 m, 99% of the time at sea state 3 A cross-track error of less than 20 m, 90% of the time at up to sea state 5.		Not Run
T8 SAT - Verify Track Control Mode Travel the defined route at 20 knots, or at a speed to be mutually agreed during the test, depending on the weather conditions. Observe the cross-track deviation on the conning display. At completion of the route, extract the logfile for the actual cross-track, for later analysis.	Cross Track Error of less than 20m, 99% of the time at sea state 3 Cross Track Error of less than 20m, 90% of the time at up to sea state 5.		Not Run

Figure 18. SAT plan for autopilot

The Test Center supports traceability between test cases and requirements, ensuring that each requirement is adequately tested. Traceability links are established between test cases and the corresponding requirements, allowing the coverage of requirements to be tracked through testing. The traceability links are demonstrated by the relationships between the entities shown in the spider diagram (Figure 19). Alternatively, the linkage between test cases and requirements can also be depicted in the form of a traceability matrix (Figure 20). These links help ensure all requirements are tested and validated during the testing phase.

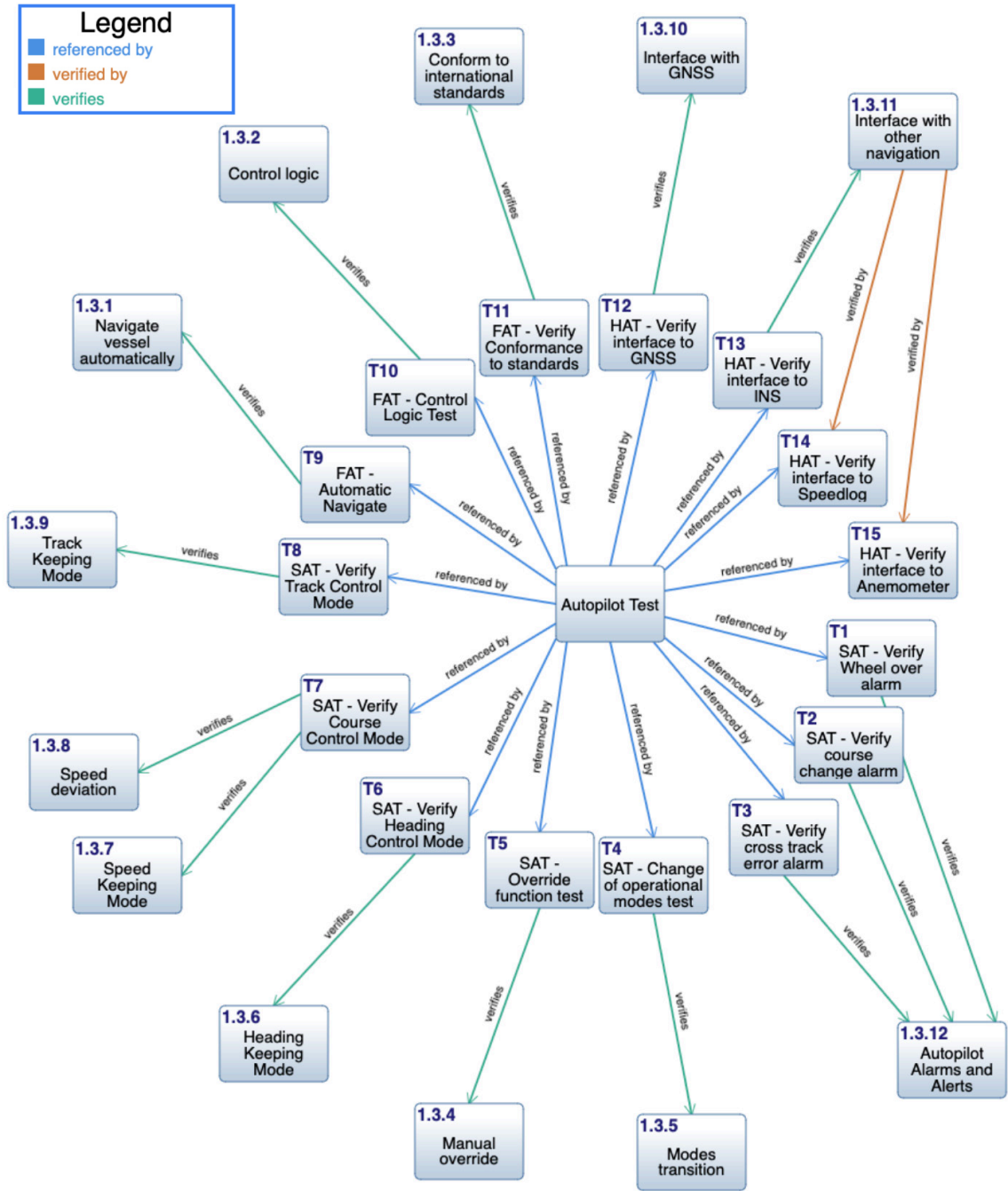


Figure 19. Spider diagram linking autopilot test entities to requirements

	1.3 Autopilot System	1.3.1 Navigate vessel automa	1.3.2 Control logic	1.3.3 Conform to internation	1.3.4 Manual override	1.3.5 Modes transition	1.3.6 Heading Keeping Mode	1.3.7 Speed Keeping Mode	1.3.8 Speed Keeping Mode	1.3.9 Track Keeping Mode	1.3.10 Interface with GNSS	1.3.11 Interface with other na	1.3.12 Autopilot Alarms and	1.4 Navigation Radar System
T1 SAT - Verify Wheel over alarm													X	
T2 SAT - Verify course change ala...													X	
T3 SAT - Verify cross track error ...													X	
T4 SAT - Change of operational m...					X									
T5 SAT - Override function test			X											
T6 SAT - Verify Heading Control ...						X								
T7 SAT - Verify Course Control M...							X	X						
T8 SAT - Verify Track Control Mode									X					
T9 FAT - Automatic Navigate Ves...	X													
T10 FAT - Control Logic Test		X												
T11 FAT - Verify Conformance to ...			X											
T12 HAT - Verify interface to GNSS									X					
T13 HAT - Verify interface to INS											X			
T14 HAT - Verify interface to Spee...											X			
T15 HAT - Verify interface to Ane...											X			

Figure 20. Traceability matrix of autopilot SAT test cases vs. requirements

IV. RESULTS AND OBSERVATIONS

This thesis explores using MBSE to improve requirements management. Three research objectives that support this goal are as follows:

- Improved communications
- Increased reusability
- Better traceability

Each research objective was evaluated based on its respective MOE to determine whether the goal had been met.

A. IMPROVED COMMUNICATIONS

There are several ways the MBSE approach improves communications among stakeholders compared to a document-centric approach. Two methods are discussed in the following paragraphs.

1. Multiple Views

A model-based approach allows a more effective understanding of a complex system than a document-centric one. The main reason is that a model consists of “multiple complementary, mutually compatible perspectives” needed to address the concerns of various stakeholders (Madni and Sievers 2018, 172).

MBSE has the capability to generate multiple views to communicate with different stakeholders with the assurance that the underlying information is derived from the same dataset. A stakeholder has the option to view the requirements in the form of a requirement diagram, spider diagram, or “tree” diagram (Appendix G.), depending on the needs. When an amendment is made to a requirement in any of the diagrams, the change will be updated accordingly in the other diagrams, ensuring that the information is accurate and up-to-date regardless of the selected view. The typical shipbuilding project takes years to complete. Inevitably, there will be personnel change. Instead of dealing

with several documents in silos, a model will facilitate the new project member to visualize the requirements and communicate better with other members.

Four organizations consisting of industry, government, defense, and research were interviewed in a survey conducted by Huldt and Stenius (2019) to evaluate the state of practice of MBSE. Analysis of their survey results indicates “that a model-based approach increases the capability to communicate” between different stakeholders (Huldt and Stenius 2019, 141). The majority of the respondents agreed (Table 1) that using an MBSE approach resulted in improvements when introducing new members to the project (142).

Table 1. Survey results – Introduce new members to system. Source: Huldt and Stenius (2019).

	Big improvement	Some improvement	No change	Some impairment	Big impairment	Not applicable
Introduce new members in the respect of system design	18%	44%	18%	2%	3%	15%
Introduce new members in the respect of system behavior	26%	33%	15%	8%	3%	15%
Introduce new members in the respect of system characteristics	20%	32%	26%	5%	2%	15%

2. Centralized Database

MBSE brings significant value by storing and managing all system-related information in a central repository. A central repository for system-related information significantly improves communications by providing a shared platform for stakeholders to access and contribute to the information.

All stakeholders have access to the latest and most up-to-date information. Required information can be retrieved anytime, regardless of physical location or time zone. This ensures all members work with the same information set, reducing misunderstandings and discrepancies. As shown in Innoslate, the Dashboard tracks the changes made by every authorized user allowing all members to be made aware of all

amendments and the person initiating the change. This contributes to improving the communications among stakeholders as the model will make them all known.

The central repository enables real-time collaboration among team members. Multiple stakeholders can simultaneously work on the same information, making updates, adding comments, or proposing changes. This promotes a collaborative environment where ideas can be shared, discussed, and refined promptly.

A central repository also promotes transparency by providing visibility into the system-related information. Stakeholders can easily review and understand the current state of the system, including its requirements, design, characteristics, and documentation. This transparency fosters better communication and shared understanding among team members. In the same survey conducted by Huldt and Stenius (2019), most respondents agreed (Table 2) that the MBSE approach had improved the ability of the stakeholders to gain a unified understanding of the system (142).

Table 2. Survey results – Establish unified understanding of system.
Source: Huldt and Stenius (2019).

	Big improvement	Some improvement	No change	Some impairment	Big impairment	Not applicable
Establish a unified understanding of system behavior	35%	37%	10%	2%	2%	14%
Establish a unified understanding of system design	24%	46%	11%	0%	3%	16%
Establish a unified understanding of system characteristics	25%	43%	18%	0%	2%	12%

Furthermore, there are mechanisms to provide notifications and alerts to keep stakeholders informed about updates or changes to the system requirements. This ensures that stakeholders stay informed and can participate in relevant discussions or decision-making processes. These relevant discussions within the central repository are recorded and documented, becoming part of the system’s history, and can be referenced later (as shown in Figure 10 and Figure 13).

A stakeholder perception survey would have been useful in verifying the enhancements in communication between stakeholders. However, the survey is beyond the intended scope of this thesis, and considering the time limitations, it is not undertaken. Nevertheless, it is believed that if a survey were conducted, the results would be similar to those obtained by Huldts and Stenius (2019).

B. INCREASED REUSABILITY

MBSE can improve reusability through the modeling of requirements by creating a standardized approach to requirements development and management. By using models to represent system requirements, designers and engineers can create a set of standardized requirements that can be easily adapted and reused across different projects (Subarna et al. 2020, 7).

1. Reusable Requirement Templates

One of the primary ways that MBSE improves reusability through the modeling of requirements is by facilitating the creation of requirement templates that capture common requirements structures or patterns, including their attributes, relationships, and properties. These templates and patterns can be reused across multiple projects or systems, providing a consistent approach to requirement development, and fostering requirements reusability.

This is illustrated by the requirements diagram and requirement construct created in Innoslate (Figure 12 and Figure 13), where attributes such as labels, rationale, ID, etc., can be carried over to a new project together with the requirement. This applies to the parent-child relationship of the requirement as well. The relationship will remain if a parent requirement is reused with partial or all its child requirements.

2. Requirement Repositories

MBSE allows requirement libraries or repositories to be created and maintained. The requirements templates and reusable requirements can be stored in these centralized repositories. By organizing and categorizing requirements in the repositories, the

requirements can be easily accessed and incorporated into the requirements diagrams in different projects or system models.

For the initial implementation of MBSE, the requirements of the first project can serve as the baseline for the repository to be reused in the next project. Requirements and templates can be created in subsequent projects and added to the repository.

3. Reusable Requirement Model

The degree and extent of requirements' reusability are examined using the sample list of requirements for WECDIS, autopilot, NRS, and IBS regulatory requirements in Appendix A. In total, the list consists of 69 requirements that are made up of safety, functional, and performance requirements. Of the 69 requirements, 57 were assessed as fully reusable, and 12 could be reused with minimum modifications. The 57 reusable requirements are predominately safety and functional requirements that apply to most surface naval ships. Safety requirements that address nautical safety standards typically do not undergo frequent changes. This stability in safety requirements is primarily due to the critical nature of safety in the maritime industry and the need for consistent standards to ensure the well-being of vessels, crew, and passengers. The naval sector operates under well-established international regulations and conventions, such as those developed by IMO. Changes to these regulations often require thorough consideration, research, and consultation among stakeholders. Therefore, the reusability for safety requirements is very high.

The requirements that need modifications are mainly performance and interface requirements. These two groups of requirements tend to encounter more frequent changes. This is primarily driven by the rapid advancements in navigation technologies and systems. Since navigation systems are tightly integrated with other ship systems, such as combat systems, communications, and automation systems, as these systems undergo upgrades or changes, performance requirements for navigation systems may need to be adjusted to ensure seamless integration, interoperability, and compatibility with the updated ship systems. Conversely, some customers will trade off higher

performance for cost and/or reliability. Moreover, evolving naval operations and mission profiles also cause performance and interface requirements to be revised.

What has been discussed in the previous paragraphs are textual modifications to the statements representing the requirements. While it is important to consider them, another critical factor must be considered when assessing the reusability of requirements in the context of MBSE. The other key consideration is the ability of the model to be reused for future projects. Instead of retrieving requirements templates and reusable requirements stored in the centralized repositories, the whole requirement model can be imported from one project into another, effectively reusing the entire model or specific portions of it. This facilitates the reuse of requirements across different projects or systems, reducing duplication and promoting consistency.

By leveraging requirement repository, templates, and import/export functionality, MBSE promotes the reusability of requirements. This enhancement in reusability led to reduced duplication, improved consistency, and increased efficiency in managing requirements across different projects or systems.

C. BETTER TRACEABILITY

Innoslate has been utilized to demonstrate how MBSE principles and practices enhance the traceability of requirements. MBSE provides graphical representations, such as requirement diagrams, spider diagrams, and traceability graphs, which visually depict the traceability paths and relationships. The following paragraphs detail how traceability can be enhanced.

1. Centralized Repository

Besides improving communications and increasing usability, the centralized repository leads to better traceability. This repository serves as a single source of truth for the system model, ensuring that requirements are consistently captured, organized, and accessible. The centralized repository enables easy navigation and retrieval of requirements, enhancing traceability.

2. Hierarchical Structure

A hierarchical structure can be established for requirements, defining parent-child relationships. This hierarchical structure provides a clear and organized representation of requirements, enabling traceability from high-level system requirements to detailed sub-requirements. The parent-child relationships establish traceability links and help maintain consistency and alignment throughout the requirement hierarchy. These hierarchical relationships are best illustrated by the requirement diagrams (Figure 12), hierarchy diagrams (Appendix E), and tree diagram (Appendix G).

3. Traceability Links

Innoslate supports the creation of traceability links between requirements and other system entities, such as design elements, Actions, and test cases. Bidirectional traceability links can be established to capture dependencies, relationships, and impacts between requirements and different entities. This is shown in Figure 21, where the autopilot requirements are linked to their corresponding test cases, Actions, and Assets. In addition, the type of relationship between the entities is also stated in the diagram. These traceability links facilitate the tracking of requirements and enable the impact of changes on the entire system to be assessed.

Figure 22 shows a requirement diagram with three Action entities added to connect the activities required to satisfy the requirements. It demonstrated that a model could leverage the benefits of traceability links and hierarchical structure together. Entities can be used in the requirement diagram to derive the relationship between the entity and the requirements and maintain the hierarchical structure.

4. Impact Analysis

Impact analysis based on the established traceability links can be conducted. The spider diagram helps users assess the impact of changes or updates to the system. By visualizing the relationships between entities, elements that may be affected by a change can be quickly identified, enabling informed decisions to be made and better management of potential risks.

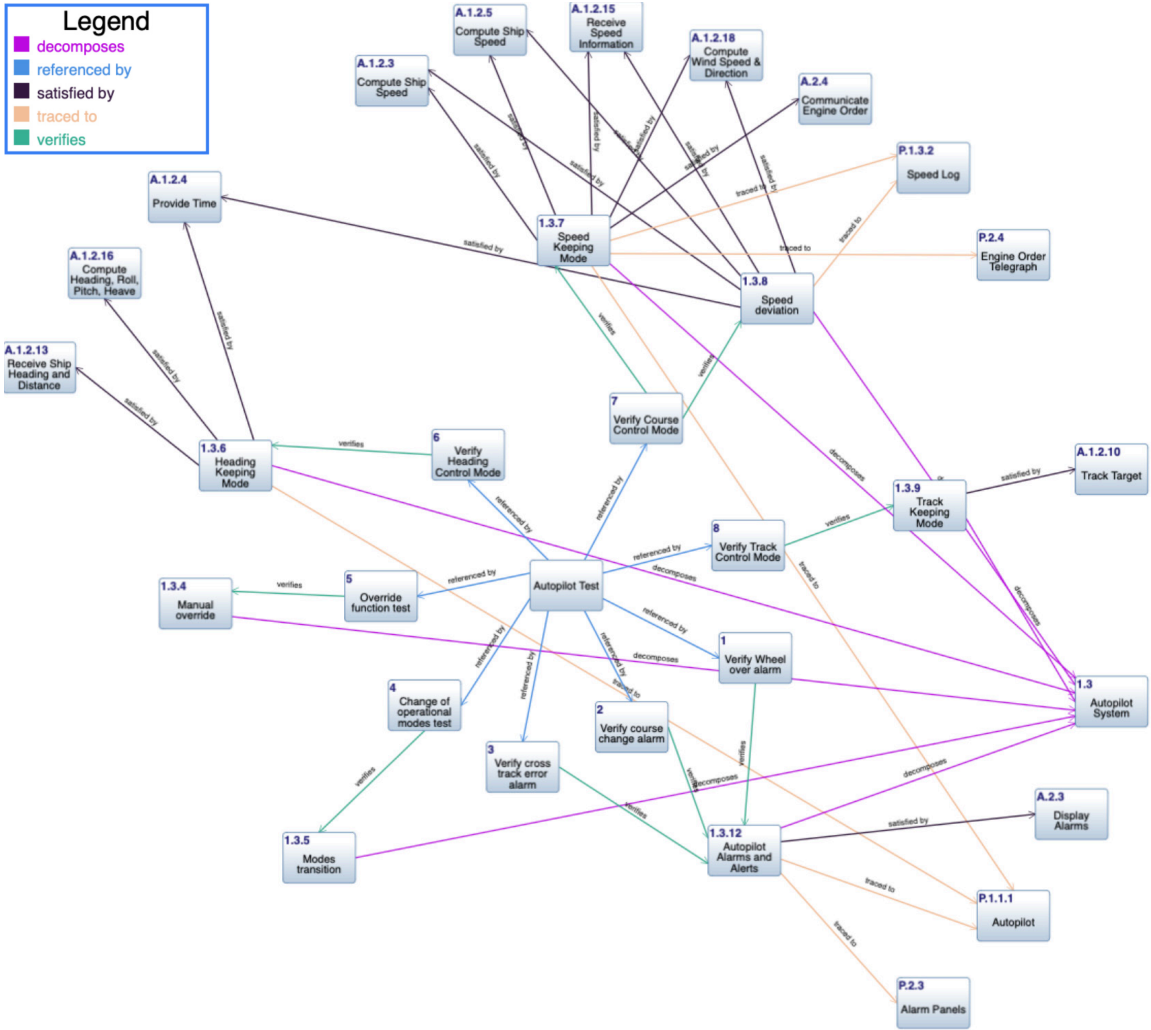


Figure 21. Traceability links between requirements and other artifacts

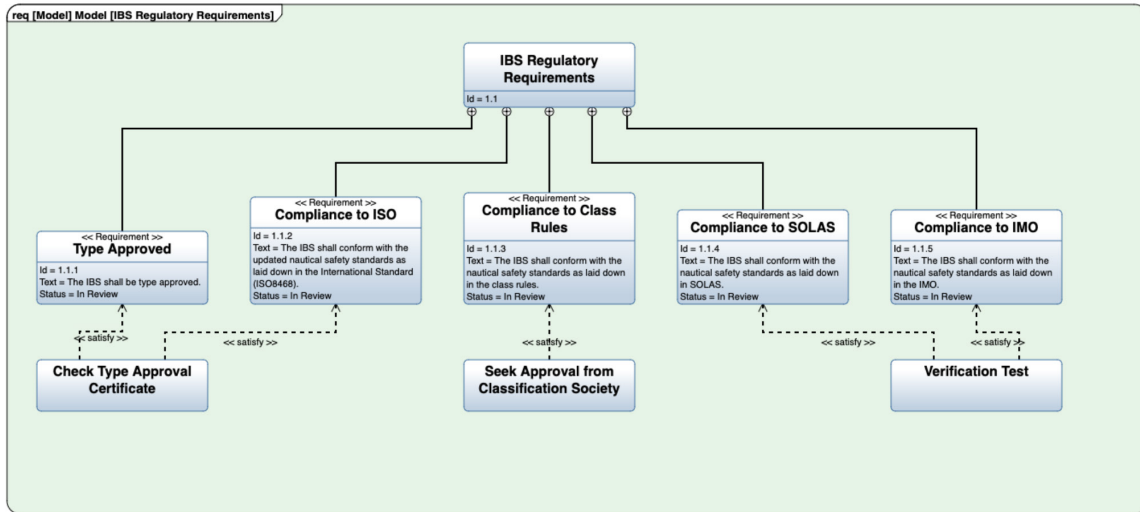


Figure 22. Requirement diagram link to Actions

By leveraging MBSE principles, the traceability of requirements was improved by providing a centralized repository, establishing hierarchical structures, enabling traceability links, and supporting impact analysis. This enhances the ability to track, manage, and validate requirements throughout the system development process, promoting consistency, transparency, and effective decision-making.

Using the autopilot sub-system as an example, all requirements are linked to their corresponding verification test by Innoslate’s Test Center (Figure 18). The Test Center integrates seamlessly with the requirements and designs aspects of the system model to link test cases to specific requirements and design elements, establishing traceability and ensuring that all system components are verifiable.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND FUTURE WORK

A. SUMMARY

Conventionally, the users' needs are expressed in text-based requirement statements that often suffer from shortcomings that can impact the effectiveness of requirements management. This thesis explored the benefits of utilizing MBSE for requirements management with the goal of addressing these shortcomings by achieving improvements in communications, enhanced reusability, and better traceability. Innoslate was the tool selected to implement the MBSE approach to requirements management by modeling a set of requirements that were extracted from the requirements of IBPMS of a naval vessel.

The requirements model was developed with various diagrams generated to represent the different views that facilitate the stakeholders' understanding of the system. The virtues of each diagram were assessed to determine its advantage over the traditional document-centric approach. The benefits of MBSE were evaluated to ascertain if the MOEs were fulfilled.

B. CONCLUSIONS

The MOE established in Chapter III, along with the benefits discussed, serves as a critical framework for evaluating the impact of MBSE on requirements management. In Chapter IV, the discussions presented showed compelling evidence of MBSE capabilities in meeting the MOE and the overall research goal to improve requirements management through the use of MBSE.

Overall, requirements management using MBSE offers numerous advantages. MBSE provides a systematic and model-centric approach to capturing, analyzing, and managing requirements during the design phase of the system lifecycle. By leveraging visual models and notations, MBSE facilitates a clear and concise representation of system requirements, promoting better stakeholder communication and understanding.

One key benefit of MBSE in requirements management is the ability to create dynamic and interconnected models that capture the complex relationships between requirements, system elements, and constraints. These models enable comprehensive traceability, allowing for effective impact analysis and change management. Furthermore, MBSE supports the requirements validation and verification processes, ensuring all requirements are traced to their corresponding test.

Another advantage of MBSE in requirements management is its ability to enhance collaboration and foster effective communications among geographically dispersed teams. With MBSE, diverse stakeholders can contribute to a single, integrated model, which anyone can access and modify with the proper permissions. The use of visual models facilitates shared understanding, reducing ambiguity for any existing and new member of the team.

MBSE's modeling approach can significantly improve reusability through the modeling of requirements by creating modular, flexible, and standardized requirement components that can be easily adapted and reused across different projects. By reducing the time and effort required for requirements development, MBSE can help to increase efficiency and reduce costs while also improving the overall quality and performance of the system.

However, it is important to note that successful implementation of MBSE for requirements management requires proper planning, training, and infrastructure. The management must embrace the paradigm shift brought about by MBSE as the current reliance on document-centric methods impedes the adoption of MBSE in most organizations (Madni and Sievers 2018, 186). Also, other hurdles, such as workplace cultures, learning curves, and skillsets availability (Carroll and Malins 2016), inhibit the adoption of MBSE methods (Huldt and Stenius 2019). Organizations must invest in suitable modeling tools, establish robust processes, and ensure effective knowledge transfer to maximize the benefits of MBSE.

In conclusion, requirements management using MBSE brings substantial improvements to the system development process. By harnessing visual models,

traceability, communication, and reusability are improved, leading to enhancements in the efficiency, effectiveness, and quality of requirements management, ultimately contributing to the successful delivery of complex systems.

C. FUTURE WORK

Looking ahead, the author recognizes the need for continued research and development to further refine and expand upon MBSE's modeling approach to requirements management. The current work can be further refined by conducting a stakeholder perception survey to analyze the various enhancements to communication among stakeholders.

Another metric that can be expanded upon is reusability. A study can be conducted to examine improvements in reusability through the modeling of requirements by providing a standardized approach to requirements testing and validation. With models to represent system requirements, designers and engineers can create simulation environments that can be used to test the system's compliance with the requirements under different conditions. The study could examine whether these simulation environments can be easily adapted and reused for future projects, reducing the need for new simulation environments to be created.

Future investigations should also explore additional performance metrics and evaluate their applicability in other major systems on naval platforms. A potential system is the naval combat management system, given its complexity and system of systems characteristics.

Additionally, future work can be conducted to explore integrating requirements management with other engineering disciplines, such as architecture, design, and verification. These integrations possess great potential in promoting consistency, coherence, and alignment across system development activities, reducing the risk of errors and rework.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. SELECTED LIST OF REQUIREMENTS

Table 3. List of requirements from selected systems

Number	Name	Description	Class
1	Integrated Bridge System		Statement
1.1	IBS Regulatory Requirements		Statement
1.1.1	Type Approved	The IBS shall be type approved.	Requirement
1.1.2	Compliance to ISO	The IBS shall conform with the updated nautical safety standards as laid down in the International Standard (ISO8468).	Requirement
1.1.3	Compliance to Class Rules	The IBS shall conform with the nautical safety standards as laid down in the class rules.	Requirement
1.1.4	Compliance to SOLAS	The IBS shall conform with the nautical safety standards as laid down in SOLAS.	Requirement
1.1.5	Compliance to IMO	The IBS shall conform with the nautical safety standards as laid down in the IMO.	Requirement
1.2	Warship Electronic Chart Display Identification System (WECDIS)		Statement
1.2.1	Display Charts	The WECDIS shall serve as a navigational information system to display simultaneously the charts and the Vessel's own position.	Requirement
1.2.2	Conform to International Standards	The WECDIS shall conform to latest IMO, IHO and IEC Standard.	Requirement

Number	Name	Description	Class
1.2.3	Chart Standards	The system shall be able to accept and display sea charts of the following standards: a. Raster charts in the UK raster charts standard ARCS b. Vector charts in DX-90 standards.	Requirement
1.2.4	Type Approved	The WECDIS shall be type-approved with a valid certificate covering the operation of the WECDIS and comes with dual-WECDIS system to support paperless navigation.	Requirement
1.2.5	Presentation Standards	The WECDIS shall conform to presentation standards for WECDIS set by International Hydrographic Office (IHO).	Requirement
1.2.6	Content And Display Aspects	The electronic chart information presented shall conform to the Specifications for Chart Content and Display Aspects of WECDIS set by IHO.	Requirement
1.2.7	Overlay Targets	The WECDIS shall be able to take in selected targets in real time from NRS and CMS and display them over the chart.	Requirement
1.2.8	Manual Fixing	The WECDIS shall support manual fixing and datum plotting.	Requirement
1.2.9	Route Planning	The WECDIS shall have the ability to perform all route planning functions as laid out in the IMO Resolution.	Requirement
1.2.10	Projection and Dead Reckoning	The WECDIS shall provide automatic dead reckoning features to display future positions with time markings. Extrapolation shall follow the assigned route or the last known course if no route has been assigned based on either Vessel's logged speed or user defined speed.	Requirement
1.2.11	History Path	The WECDIS shall show the history path of own Vessel and other tracked targets using dotted lines with the option to select a color to be assigned to the history paths.	Requirement
1.2.12	Chart Correction	The WECDIS shall allow the operator to make chart correction.	Requirement

Number	Name	Description	Class
1.2.13	Measurement Vector	The WECDIS shall provide a measurement vector for the operator to measure range and bearing between two points.	Requirement
1.2.14	Planting of Temporary Reference Point, Datum and Hazard Marks	The WECDIS shall allow the operator to add and delete on-line reference points, datum, and hazard marks.	Requirement
1.2.15	De-cluttering Features	The WECDIS shall be able to de-clutter certain information not necessary for display.	Requirement
1.2.16	Display Alarms	The WECDIS shall display warnings to the nearest navigation mark, hazard, or tracked targets.	Requirement
1.2.17	User Profiles and Layers	WECDIS shall provide the creation of customized user profiles with password authorization logins.	Requirement
1.2.18	Software Update	The WECDIS shall support chart and software update/upgrade via USB or DVD.	Requirement
1.2.19	Integration with Maritime Automatic Identification System (MAIS)	The WECDIS shall be integrated with the MAIS and display at least 1000 tracks from MAIS.	Requirement
1.2.20	Integration with Autopilot	The WECDIS shall integrate with the autopilot to facilitate the planting of route in autopiloting.	Requirement
1.3	Autopilot System		Statement
1.3.1	Navigate Vessel Automatically	The autopilot system shall be capable of navigating the Vessel automatically on a pre-planned mode of operation under various conditions.	Requirement
1.3.2	Control Logic	The control system logic of the autopilot shall be robust and shall be able to compensate for the following conditions: a. Up to Sea State 5 b. Water current of up to 3 knots	Requirement

Number	Name	Description	Class
1.3.3	Conform to International Standards	The autopilot system shall conform to IMO Safety requirements and IEC Standard IEC 62065.	Requirement
1.3.4	Manual Override	There shall be a manual override button to allow the operator to take over propulsion and steering control.	Requirement
1.3.5	Modes Transition	Transition between different control modes shall not cause any sudden or unexpected movement of the Vessel.	Requirement
1.3.6	Heading Keeping Mode	The autopilot system shall be capable of maintaining the Vessel at a constant heading at up to Sea State 5 with Heading deviation < 3° for 90% of the time.	Requirement
1.3.7	Speed Keeping Mode	The autopilot system shall be capable of maintaining the Vessel at a desired speed over water or a desired speed over ground or to a desired engine RPM entered by the operator.	Requirement
1.3.8	Speed Deviation	The Speed Keeping Mode shall maintain speed deviation < 1 kts for 90% of the time up to Sea State 5 conditions	Requirement
1.3.9	Track Keeping Mode	The Track Keeping Mode shall automatically control the Vessel to receive and follow a track defined by waypoints planned on the WECDIS.	Requirement
1.3.10	Interface with GNSS	The autopilot system shall interface with the GNSS to receive positional information.	Requirement
1.3.11	Interface with Other Navigation System	The autopilot system shall interface with other navigation systems such as Inertial Navigation System (INS), speed log and Anemometer for dead-reckoning computations.	Requirement
1.3.12	Autopilot Alarms and Alerts	Visual and audio alarms/alerts shall be provided all malfunctions or exceed of threshold.	Requirement

Number	Name	Description	Class
1.4	Navigation Radar System (NRS)		Statement
1.4.1	X-Band Frequency	The NRS shall be consist of X -band solid state navigation radars.	Requirement
1.4.2	Radar Coverage	The NRS shall provide all round coverage.	Requirement
1.4.3	ARPA Function	The navigation radars shall have ARPA function.	Requirement
1.4.4	Guard Zone Function	The navigation radars shall have guard zone function.	Requirement
1.4.5	Track Targets	The NRS shall have ability to track 100 surface targets per scanner.	Requirement
1.4.6	System Redundancy	The NRS shall have no single point of failure in terms of hardware and software.	Requirement
1.4.7	Operating Conditions	The NRS shall be able to detect surface vessels, small boats, floating objects in congested waters in all weather conditions.	Requirement
1.4.8	Blanking Signal	The NRS shall provide blanking signals to combat systems during transmission.	Requirement
1.4.9	Electromagnetic Interference	The NRS shall not cause interference with other navigation radar.	Requirement
1.4.10	Compliance to IEC 62388	The NRS shall conform to IEC 62388 or equivalent for safe and effective performance at sea.	Requirement
1.4.11	Compliance to IMO Standards	The NRS shall comply with International Maritime Organization (IMO) standards for collision avoidance:	Requirement
1.4.12	Man-Aloft Switch	The NRS shall have a man-aloft switch to cut off power supply during emergency and maintenance.	Requirement
1.4.13	Power Up Time	The power up time for the NRS from cold shall be less than 4 minutes and instantaneous from standby.	Requirement

Number	Name	Description	Class
1.4.14	Common Reference Point	The range and bearing of the reported track from the NRS shall be referenced to a common reference point to be finalized in the Design Review.	Requirement
1.4.15	Clutter Rejection	The NRS shall have very good clutter rejection.	Requirement
1.4.16	Mitigate ducting artefacts	The NRS shall have processing capabilities to reduce artefacts caused by ducting.	Requirement
1.4.17	Mitigate False Echoes	The NRS shall be able to mitigate false echoes from large passing vessels.	Requirement
1.4.18	NRS Display	The numbering of the tracks, the track symbols, the colors and other items displayed by the NRS shall adhere to the standard defined by the user.	Requirement
1.4.19	Display Range	The radar range scale for the NRS display mimic shall be 0.25 nm, 0.50 nm, 0.75 nm, 1.5 nm, 3.0 nm doubling to 48 nm.	Requirement
1.4.20	Variable Range Maker (VRM)	The VRM shall provide the means to measure the range of a target or feature and a means to set a range reference from own Vessel for navigation.	Requirement
1.4.21	Electronic Bearing Line (EBL)	The EBL shall provide the means to measure the bearing of a target or feature and a means to set a bearing reference from own Vessel for navigation and for basic collision avoidance.	Requirement
1.4.22	Electronic Range and Bearing Line (ERBL)	The ERBL shall measure the range and bearing of one position on the display, relative to any other position, within the operation display area.	Requirement
1.4.23	Parallel Index (PI)	A total of 4 independent PI lines shall be provided to set a beam range and bearing reference to own Vessel.	Requirement
1.4.24	Alarms and Indicators	The NRS shall provide alarms and indications on the screen to signify total or partial failure with the respective message for the activated alarms.	Requirement

Number	Name	Description	Class
1.4.25	Picture Freeze	The NRS shall provide a means to alert the operator of “picture freeze.”	Requirement
1.4.26	CMS Tracks	The NRS shall be able to receive tracks from the CMS and display them on the NRS mimic display via the Combat Management System. Number of tracks to be finalized in Design Review.	Requirement
1.4.27	Common Track Numbering Scheme	The NRS shall enable the CMS and the NRS system to display the same track number for the same target. The scheme shall be finalized at Integration Review.	Requirement
1.4.28	NR Videos	The NRS shall be able to send the NR videos to CMS.	Requirement
1.4.29	Interface with WECDIS	The NRS shall interface with WECDIS to allow overlaying for chart display.	Requirement
1.4.30	Interface with MAIS	The NRS shall be able to interface to a commercial AIS transponder, mandated by the 1974 International Convention for the Safety Of Life At Sea (SOLAS).	Requirement
1.4.31	Interface with VDR	The NRS shall interface with the VDR to provide output of the display information.	Requirement
1.4.32	Interface with Other Navigation System	The NRS shall interface with other navigation systems such as Inertial Navigation System (INS), GNSS and speed log.	Requirement

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. COMPLETE ACTION DIAGRAMS

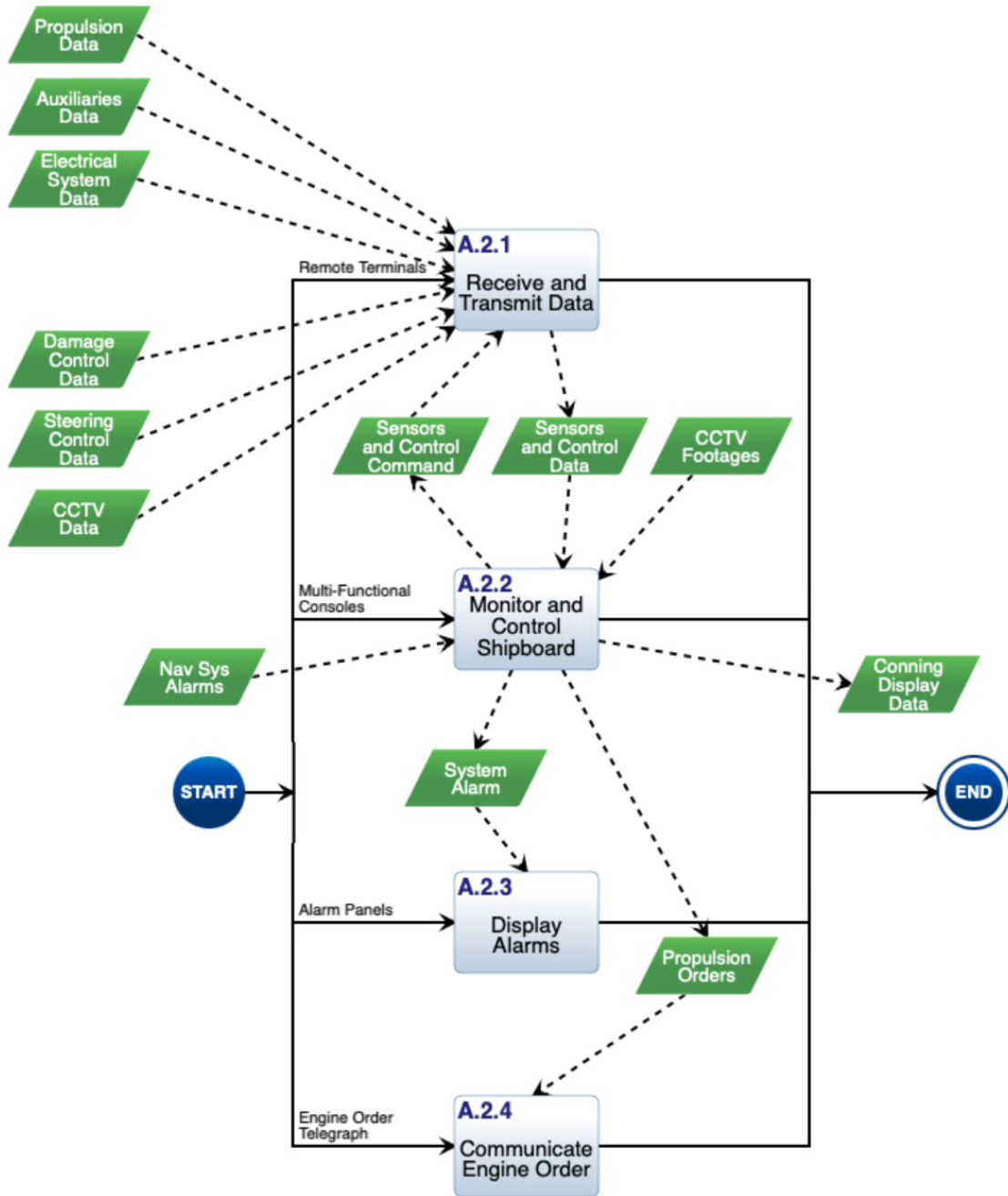


Figure 23. Action diagram of IPMS (decomposed from IBPMS)

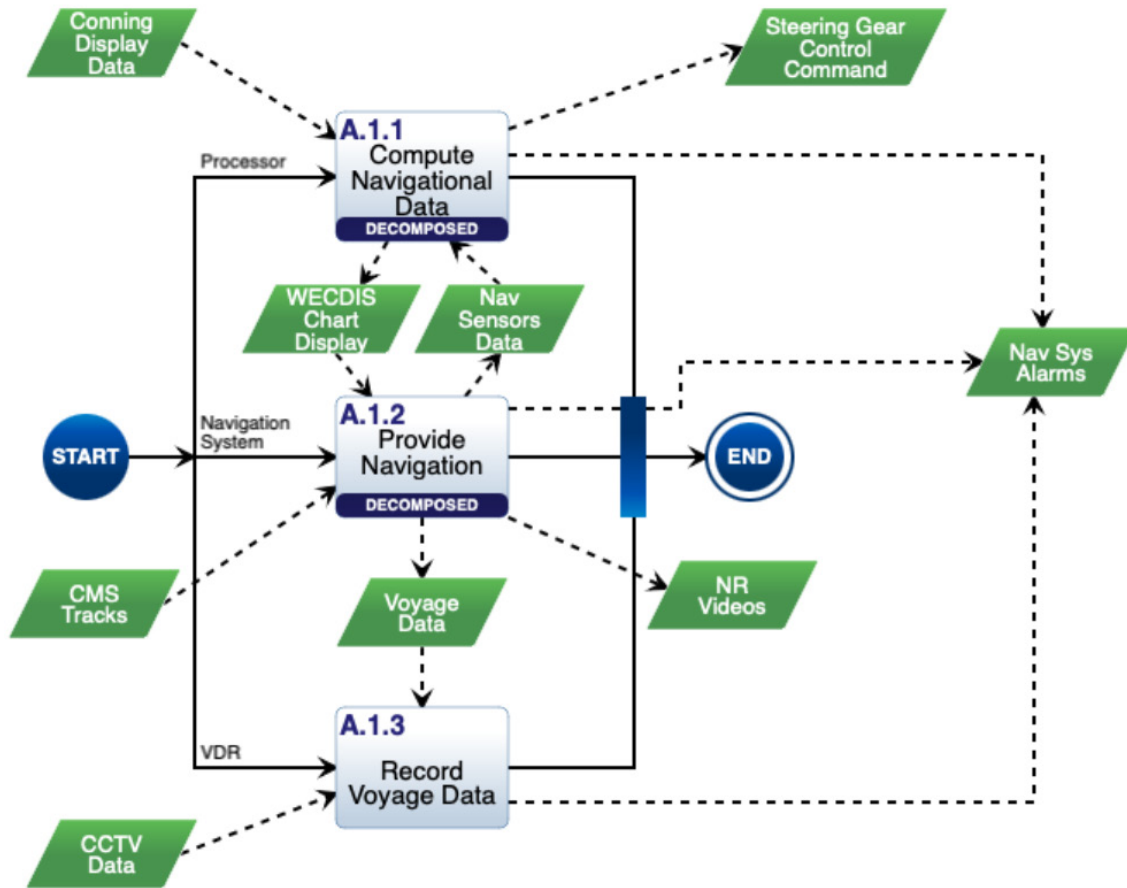


Figure 24. Action diagram of IBS (decomposed from IBPMS)

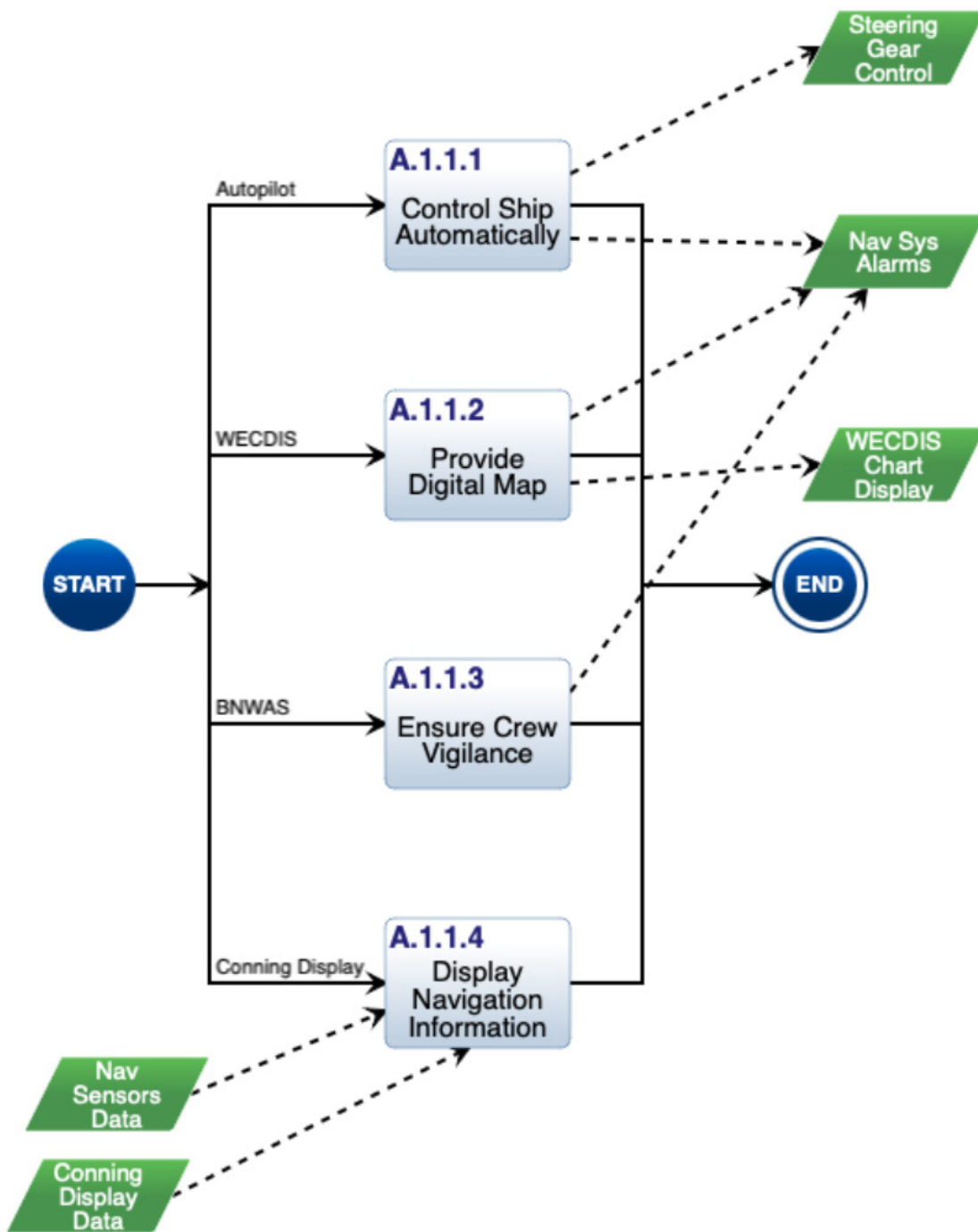


Figure 25. Action diagram of “Compute Navigational Data” (decomposed from IBS)

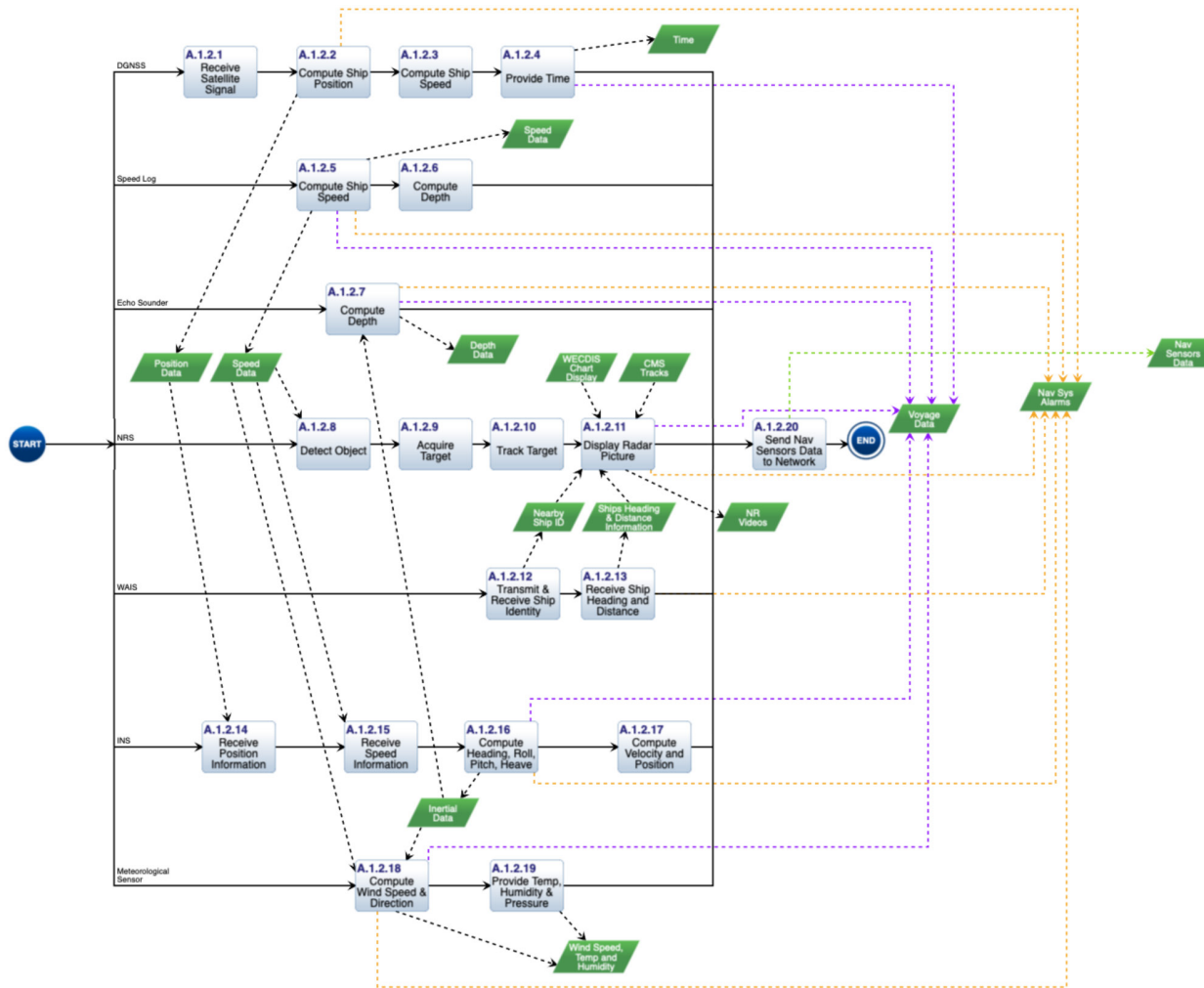


Figure 26. Action diagram of “Provide Navigation” (decomposed from IBS)

APPENDIX C. COMPLETE REQUIREMENT DIAGRAMS

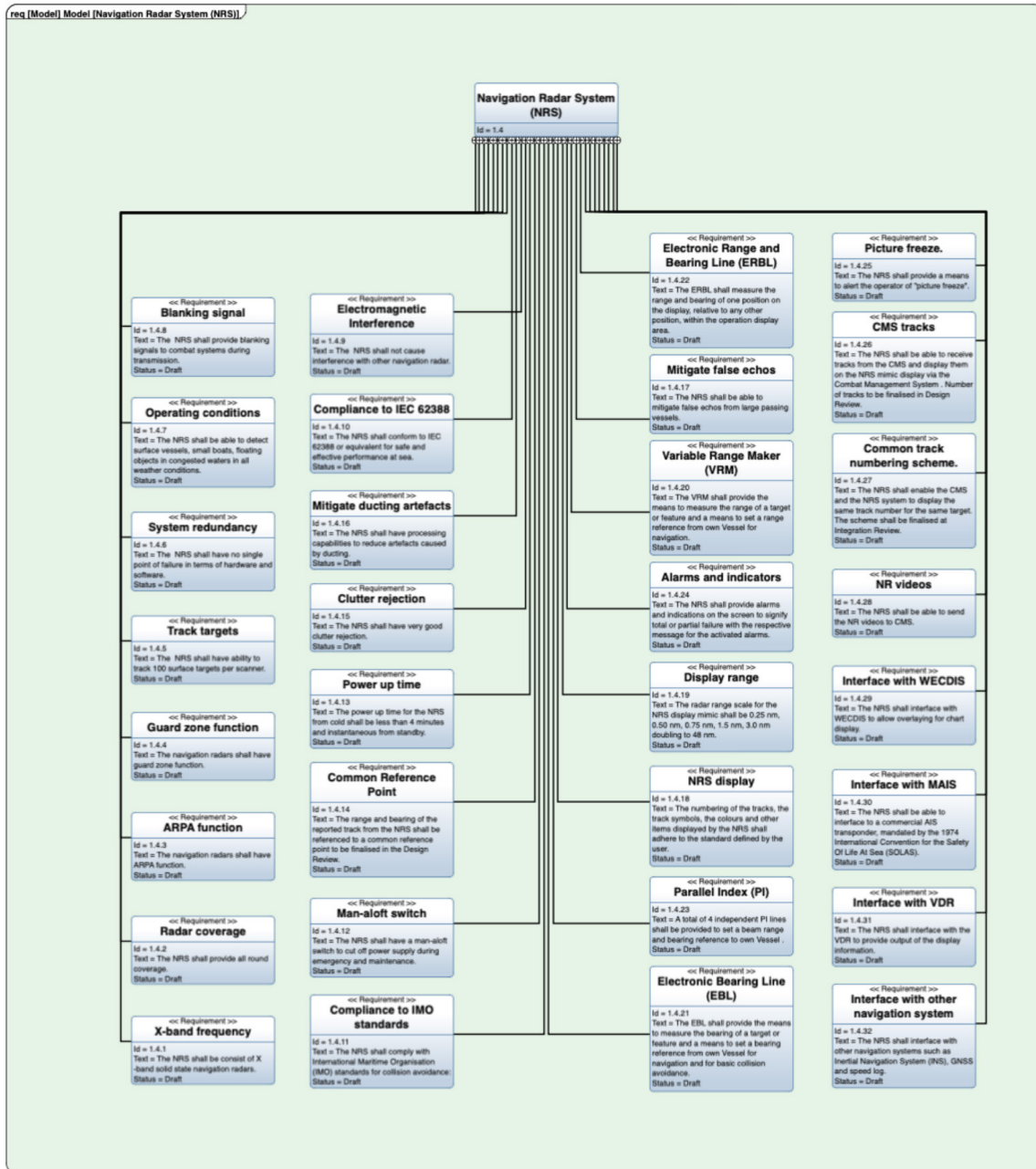


Figure 27. Requirement diagram for NRS

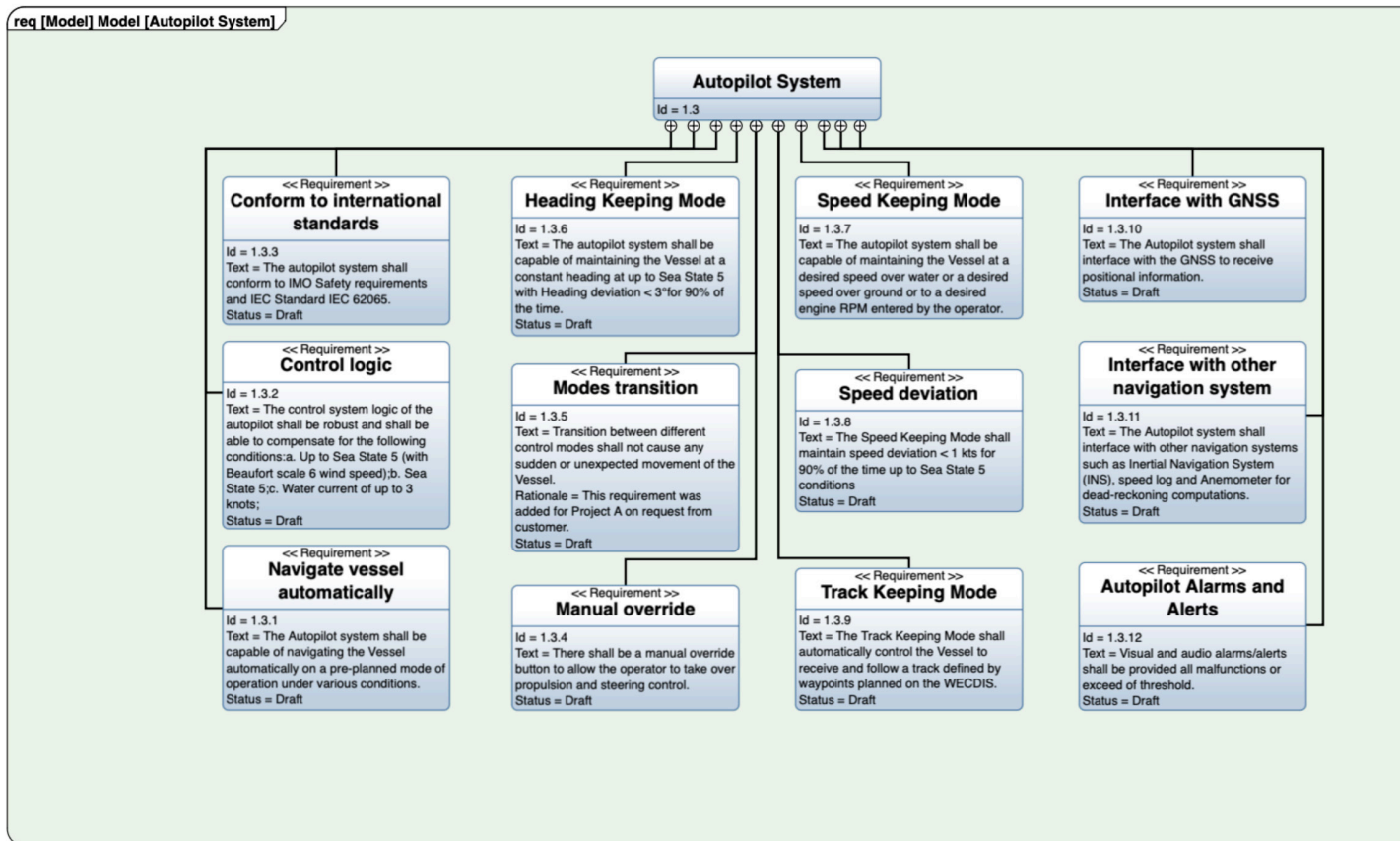


Figure 28. Requirement diagram for autopilot system

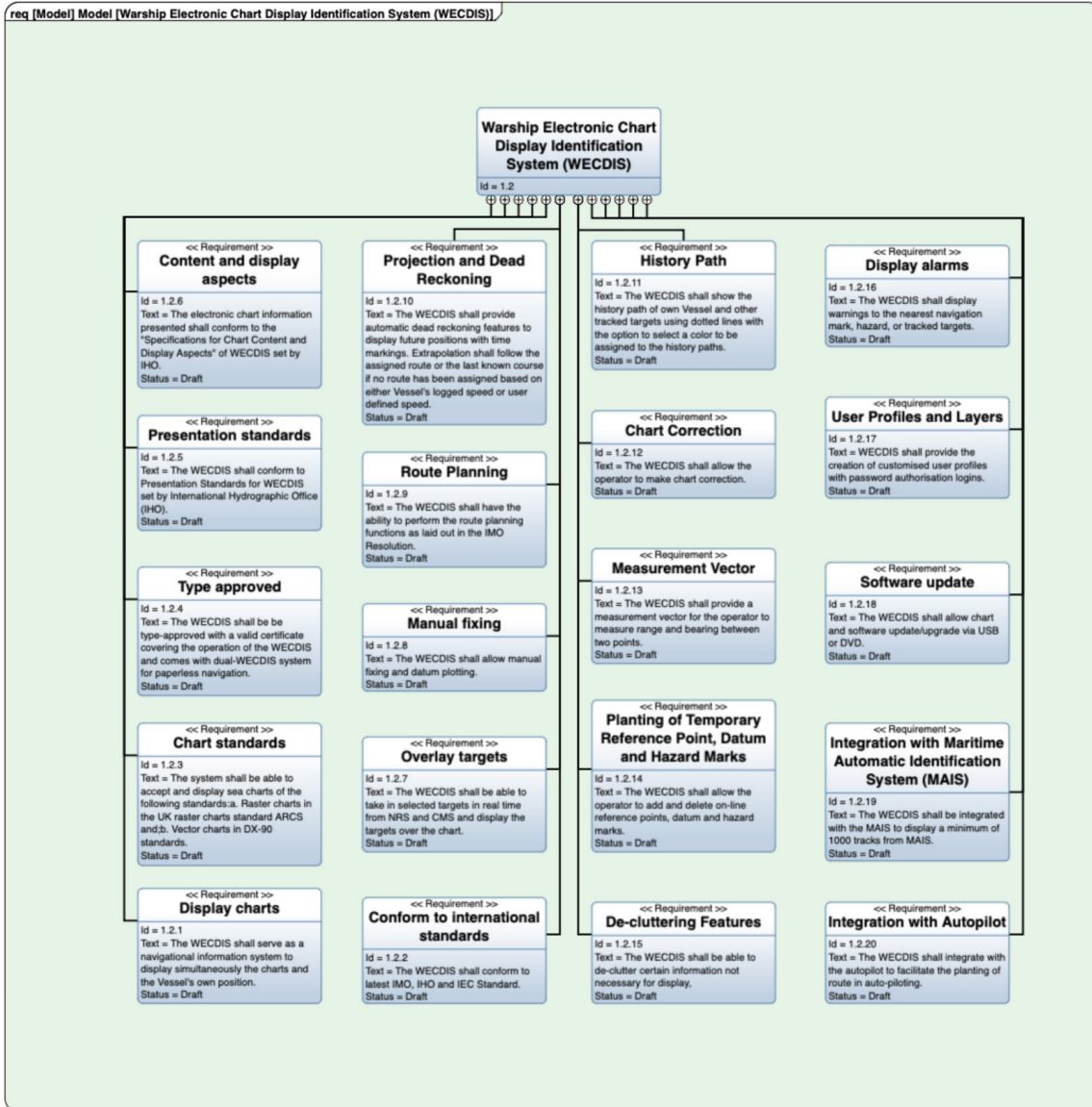


Figure 29. Requirement diagram for WECDIS

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. SPIDER DIAGRAM

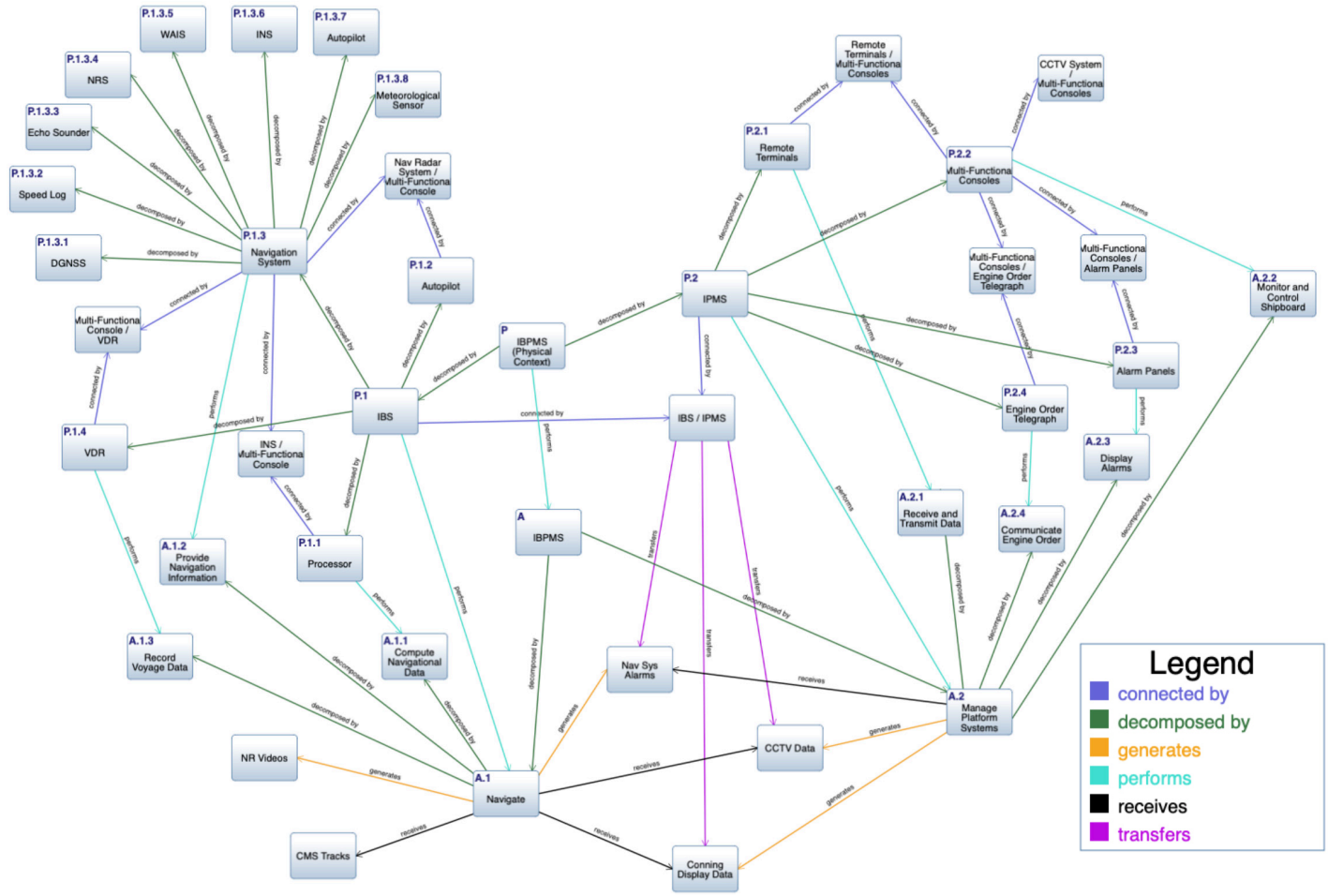


Figure 30. Traceability paths between IBPMS Assets, Actions, and Input/Output signals

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E. HIERARCHY DIAGRAMS

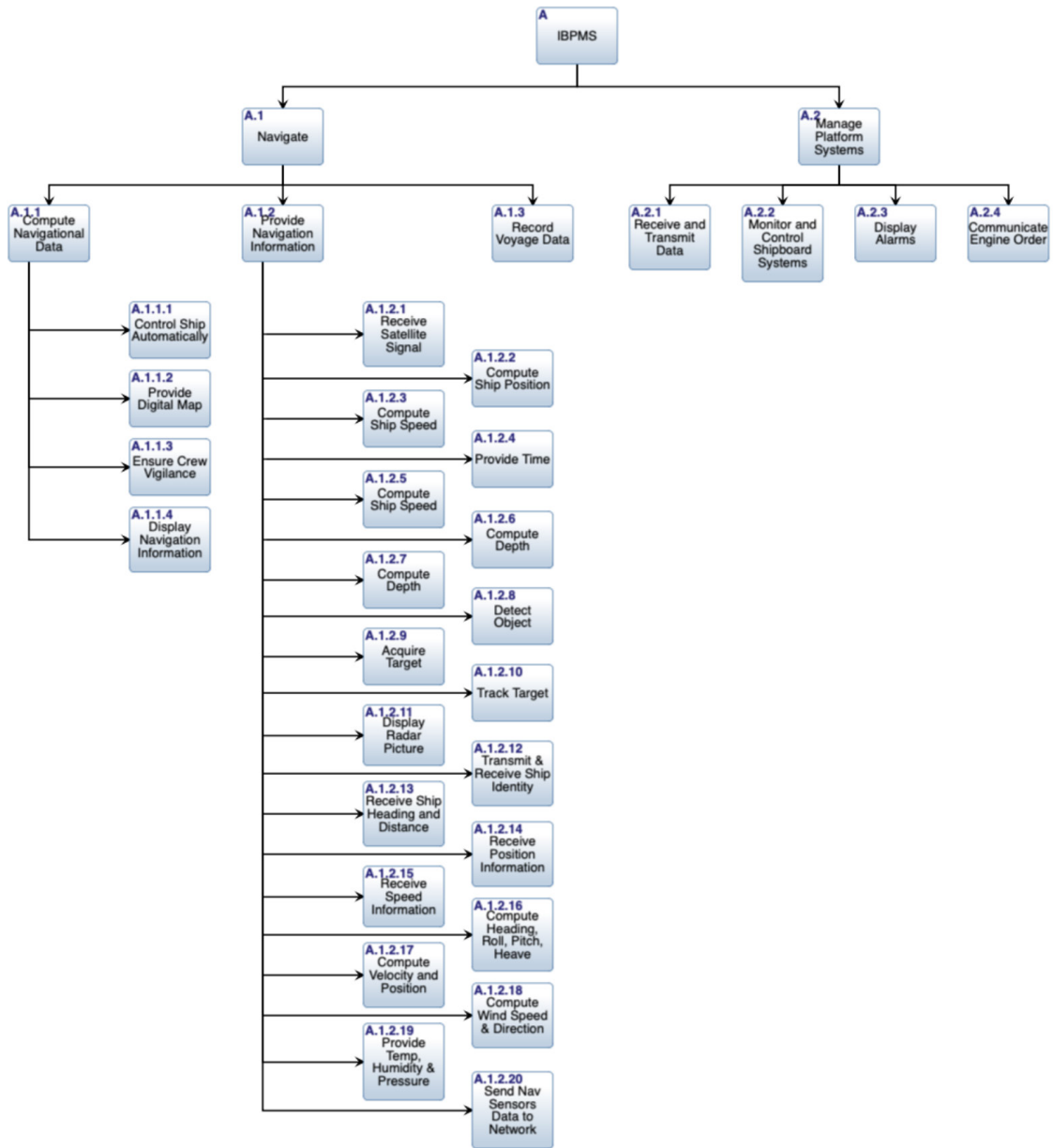


Figure 31. Functional hierarchy diagram of IBPMS generated from action diagram

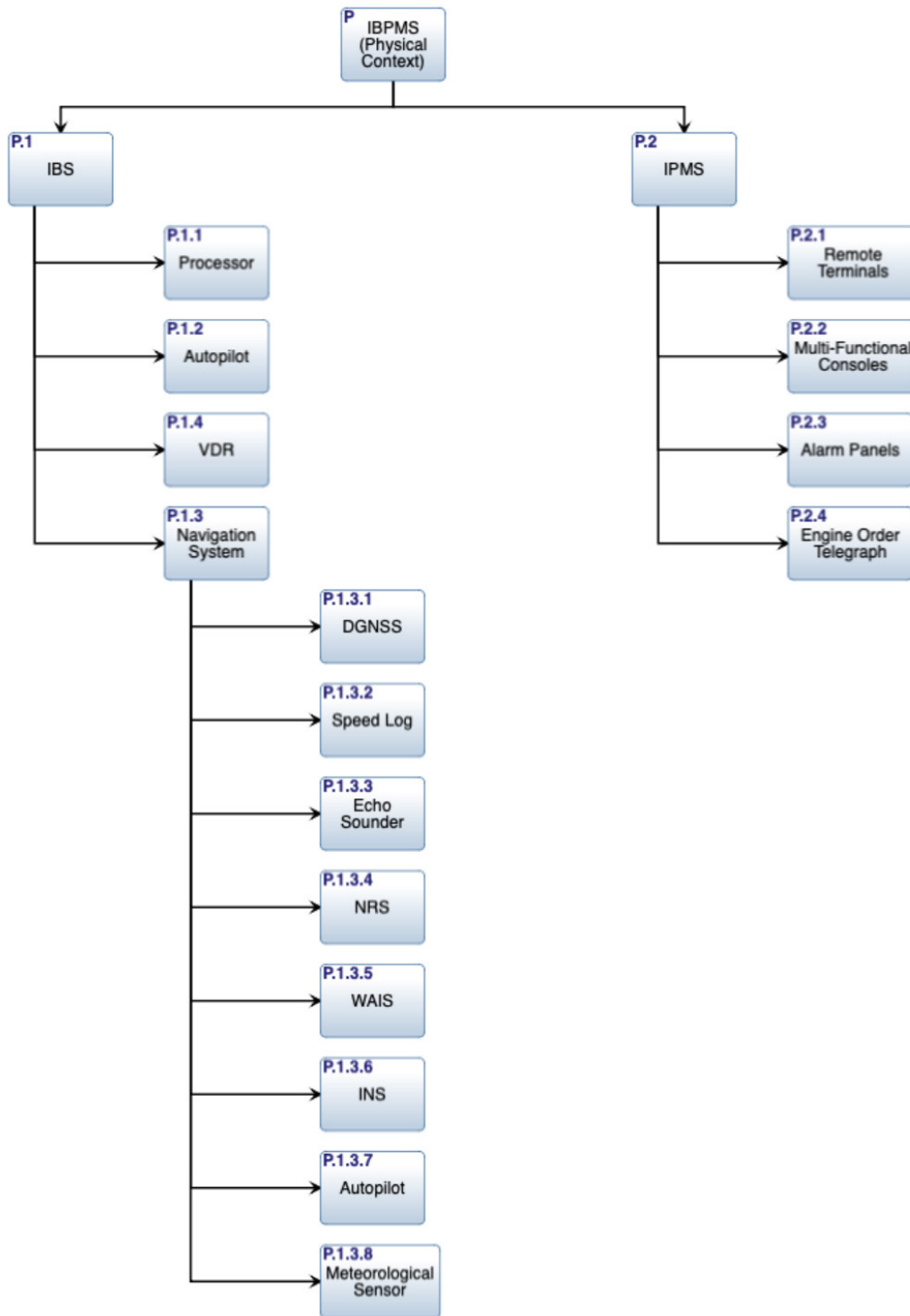


Figure 32. Physical hierarchy diagram of IBPMS generated from action diagram

	A.1.1 Navigate	A.1.1.1 Compute Navig...	A.1.1.1.1 Control Ship	A.1.1.3 Provide Digit...	A.1.1.4 Display Navig...	A.1.2 Ensure Crew	A.1.2.1 Provide Navig...	A.1.2.2 Receive Navig...	A.1.2.3 Compute Sate...	A.1.2.4 Compute Ship	A.1.2.5 Compute Ship	A.1.2.6 Compute Time	A.1.2.7 Compute Ship	A.1.2.8 Compute Dep...	A.1.2.9 Detect Object	A.1.2.10 Acquire Targe	A.1.2.11 Track Target	A.1.2.12 Display Rad...	A.1.2.13 Transmit & R...	A.1.2.14 Receive Ship	A.1.2.15 Receive Pos...	A.1.2.16 Receive Spee...	A.1.2.17 Compute He...	A.1.2.18 Compute Ve...	A.1.2.19 Provide Wi...	A.1.3 Record Term...	A.2 Manage Voyage	A.2.1 Send Nav Se...	A.2.1 Receive and Tr...	A.2.2 Monitor and Tr...	A.2.3 Display Alarms	A.2.4 Communicate L...		
1.3 Autopilot System																																		
1.3.1 Navigate vessel auto...																																		
1.3.2 Control logic		X																				X											X	
1.3.3 Conform to internati...																							X										X	
1.3.4 Manual override																																		
1.3.5 Modes transition																																		
1.3.6 Heading Keeping Mode																		X																
1.3.7 Speed Keeping Mode																																		X
1.3.8 Speed deviation																																		
1.3.9 Track Keeping Mode																																		
1.3.10 Interface with GNSS																																		
1.3.11 Interface with other ...	X	X																																
1.3.12 Autopilot Alarms an...																																		
1.4 Navigation Radar Syst...																																		
1.4.1 X-band frequency																																		
1.4.2 Radar coverage																																		
1.4.3 ARPA function																																		
1.4.4 Guard zone function																																		
1.4.5 Track targets																																		
1.4.6 System redundancy																																		
1.4.7 Operating conditions																																		
1.4.8 Blanking signal																																		
1.4.9 Electromagnetic Inte...																																		
1.4.10 Compliance to IEC ...																																		

Figure 34. Traceability matrix of requirements vs. Actions (Part 2 of 3)

	A.1 Navigate	A.1.1 Compute Navig	A.1.1.1 Control Ship	A.1.1.2 Provide Digita	A.1.1.3 Ensure Crew	A.1.1.4 Display Navig	A.1.2 Provide Naviga	A.1.2.1 Receive Sate	A.1.2.2 Compute Shp	A.1.2.3 Compute Shp	A.1.2.4 Provide Time	A.1.2.5 Compute Shp	A.1.2.6 Compute Dep	A.1.2.7 Compute Dep	A.1.2.8 Detect Object	A.1.2.9 Acquire Targe	A.1.2.10 Track Target	A.1.2.11 Display Rad	A.1.2.12 Transmit & R	A.1.2.13 Receive Shp	A.1.2.14 Receive Pos	A.1.2.15 Receive Spe	A.1.2.16 Compute He	A.1.2.17 Compute He	A.1.2.18 Compute Ve	A.1.2.19 Provide Tem	A.1.3 Record Nav Se	A.2 Manage Voyage	A.2.1 Receive and Tr	A.2.2 Monitor and Tr	A.2.3 Display Alarms	A.2.4 Communicate L		
1.4.11 Compliance to IMO ...																																		
1.4.12 Man-aloft switch																																		
1.4.13 Power up time								X																										
1.4.14 Common Reference...															X																			
1.4.15 Clutter rejection																																		
1.4.16 Mitigate ducting art...																																		
1.4.17 Mitigate false echos																																		
1.4.18 NRS display															X																			
1.4.19 Display range																X																		
1.4.20 Variable Range Mak...																																		
1.4.21 Electronic Bearing ...																																		
1.4.22 Electronic Range a...																																		
1.4.23 Parallel Index (PI)																																		
1.4.24 Alarms and indicators																																	X	
1.4.25 Picture freeze.																X																		
1.4.26 CMS tracks															X																	X		
1.4.27 Common track num...															X																			
1.4.28 NR videos																																		
1.4.29 Interface with WEC...																																		
1.4.30 Interface with MAIS																																		
1.4.31 Interface with VDR																																		
1.4.32 Interface with other ...	X	X					X	X																								X	X	

Figure 35. Traceability matrix of requirements vs. Actions (Part 3 of 3)

	P-1 IBS	P-1.1 Processor	P-1.1.1 Autopilot	P-1.1.2 WECDIS	P-1.1.3 BWAS	P-1.1.4 Conning Display	P-1.3 Navigation System	P-1.3.1 DGNSS	P-1.3.2 Speed Log	P-1.3.3 Echo Sounder	P-1.3.4 NRS	P-1.3.5 WAS	P-1.3.6 INS	P-1.3.7 Meteorological	P-2 JPMS	P-2.1 Remote Terminal	P-2.2 Multi-Functional	P-2.3 Alarm Panels	P-2.4 Engine Order Te
1 Integrated Bridge System																			
1.1 IBS Regulatory Requir...	X																		
1.1.1 Type Approved	X																		
1.1.2 Compliance to ISO	X																		
1.1.3 Compliance to Class ...	X																		
1.1.4 Compliance to SOLAS	X																		
1.1.5 Compliance to IMO	X																		
1.2 Warship Electronic Ch...		X		X															
1.2.1 Display charts		X		X	X														
1.2.2 Conform to internati...				X															
1.2.3 Chart standards		X																	
1.2.4 Type approved				X															
1.2.5 Presentation standards				X															
1.2.6 Content and display ...		X		X	X														
1.2.7 Overlay targets		X		X															
1.2.8 Manual fixing				X															
1.2.9 Route Planning				X															
1.2.10 Projection and Dea...				X					X										
1.2.11 History Path				X															
1.2.12 Chart Correction		X		X															
1.2.13 Measurement Vector				X															
1.2.14 Planting of Tempora...				X															
1.2.15 De-cluttering Features		X		X															
1.2.16 Display alarms				X	X													X	
1.2.17 User Profiles and L...				X															
1.2.18 Software update				X															
1.2.19 Integration with Mar...				X															
1.2.20 Integration with Aut...			X	X															
1.3 Autopilot System			X																
1.3.1 Navigate vessel auto...			X						X										
1.3.2 Control logic			X																
1.3.3 Conform to internati...			X																
1.3.4 Manual override																			
1.3.5 Modes transition																			
1.3.6 Heading Keeping Mode			X																
1.3.7 Speed Keeping Mode			X						X										X
1.3.8 Speed deviation									X										
1.3.9 Track Keeping Mode																			
1.3.10 Interface with GNSS			X																
1.3.11 Interface with other ...			X			X	X	X	X	X									
1.3.12 Autopilot Alarms an...			X														X		

Figure 36. Traceability matrix of requirements vs. Assets (Part 1 of 2)

	P.1 IBS	P.1.1 Processor	P.1.1.1 Autopilot	P.1.1.2 WEC/DIS	P.1.1.3 BNWAS	P.1.1.4 Conning Display	P.1.3 Navigation System	P.1.3.1 DGNSS	P.1.3.2 Speed Log	P.1.3.3 Echo Sounder	P.1.3.4 NRS	P.1.3.5 WAIS	P.1.3.6 INS	P.1.3.7 Meteorological	P.1.4 VDR	P.2 IPMS	P.2.1 Remote Terminal	P.2.2 Multi-Functional	P.2.3 Alarm Panels	P.2.4 Engine Order Te
1.4 Navigation Radar Syst...									X											
1.4.1 X-band frequency									X											
1.4.2 Radar coverage									X											
1.4.3 ARPA function									X											
1.4.4 Guard zone function																				
1.4.5 Track targets																				
1.4.6 System redundancy																				
1.4.7 Operating conditions																				
1.4.8 Blanking signal																				
1.4.9 Electromagnetic Inte...									X											
1.4.10 Compliance to IEC ...																				
1.4.11 Compliance to IMO ...																				
1.4.12 Man-aloft switch																				
1.4.13 Power up time																				
1.4.14 Common Reference...																				
1.4.15 Clutter rejection																				
1.4.16 Mitigate ducting art...																				
1.4.17 Mitigate false echos									X											
1.4.18 NRS display					X				X											
1.4.19 Display range					X				X											
1.4.20 Variable Range Mak...																				
1.4.21 Electronic Bearing ...																				
1.4.22 Electronic Range a...	X				X															
1.4.23 Parallel Index (PI)																				
1.4.24 Alarms and indicators																			X	
1.4.25 Picture freeze.																				
1.4.26 CMS tracks					X				X											
1.4.27 Common track num...									X											
1.4.28 NR videos									X											
1.4.29 Interface with WEC...	X	X																		
1.4.30 Interface with MAIS																				
1.4.31 Interface with VDR	X												X							
1.4.32 Interface with other ...					X	X	X	X	X	X										

Figure 37. Traceability matrix of requirements vs. Assets (Part 2 of 2)

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX G. TREE DIAGRAM

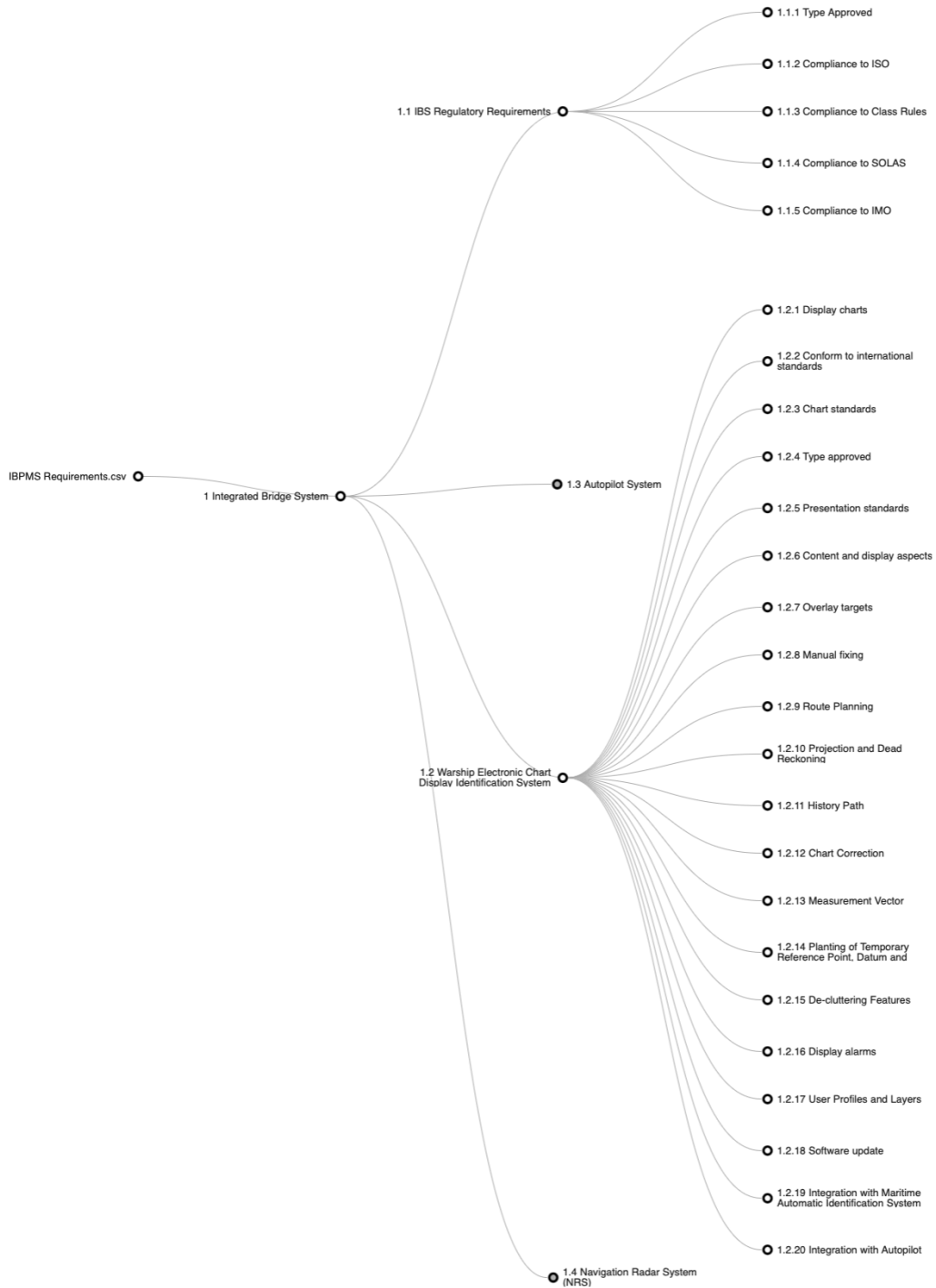


Figure 38. Tree diagram showing relationships between IBPMS requirements

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- Anschutz. n.d. "Synapsis NX." Accessed May 17, 2023. <https://www.anschuetz.com/what-we-do/synapsis-nx-integrated-navigation-system/>.
- Carroll, Edward R, and Robert J. Malins. 2016. *Systematic Literature Review: How Is Model-Based Systems Engineering Justified?*. SAND2016-2607. Albuquerque, NM: Sandia National Lab. <https://www.osti.gov/servlets/purl/1561164>.
- Clark, Eric D. 2006. "Analysis and Comparison of Various Requirements Management Tools for Use in the Shipbuilding Industry." Master's thesis, Naval Postgraduate School. <https://calhoun.nps.edu/handle/10945/2546>.
- Dam, Steven H. 2023. "Real MBSE Model-Based Systems Engineering (MBSE) Using LML and Innoslate®." Unpublished manuscript, April 22, 2023.
- Friedenthal, Sanford, Regina Griego, and Mark Sampson. 2009. "INCOSE Model Based Systems Engineering (MBSE) Initiative." In *INCOSE Symposium 11*. <https://www.researchgate.net/publication/267687693>.
- Friedenthal, Sanford., Alan Moore, and Rick Steiner. 2012. *A Practical Guide to SysML the Systems Modeling Language*. 2nd ed. Waltham, MA: Morgan Kaufmann.
- Hood, Colin., Simon Wiedemann, Stefan Fichtinger, and Urte Pautz. 2008. *Requirements Management: The Interface Between Requirements Development and All Other Systems Engineering Processes*. 1st ed. Berlin, Heidelberg: Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-540-68476-3>.
- Huldt, T., and I. Stenius. 2019. "State-of-Practice Survey of Model-Based Systems Engineering." *Systems Engineering* 22 (2): 134–45. <https://doi.org/10.1002/sys.21466>.
- Kravari, Kalliopi, Christina Antoniou, and Nick Bassiliades. 2021. "SENSE: A Flow-Down Semantics-Based Requirements Engineering Framework." *Algorithms* 14(10): 298. <https://doi.org/10.3390/a14100298>.
- L3Harris. n.d. "Integrated Platform Management System." Accessed May 17, 2023. <https://www.l3harris.com/all-capabilities/integrated-platform-management-system>.
- Lebeaupin, Benoît, and Antoine Rauzy. 2020. "Toward a Better Integration of Requirements and Model-Based Specifications." *Systems Engineering* 23 (6): 751–69. <https://doi.org/10.1002/sys.21560>.

- Lerche, Felix. 2019. “5 Reasons Why a Requirements Traceability Matrix Is Not Enough.” *itemis* (blog), August 13, 2019. <https://blogs.itemis.com/en/5-reasons-why-a-requirements-traceability-matrix-is-not-enough>.
- Madni, Azad M., and Michael Sievers. 2018. “Model-Based Systems Engineering: Motivation, Current Status, and Research Opportunities.” *Systems Engineering* 21 (3): 172–90. <https://doi.org/10.1002/sys.21438>.
- OSI Maritime Systems. n.d. “Integrated Navigation & Tactical System.” Accessed June 2, 2023. <https://osimaritime.com/solutions/systems/int/>.
- Ramesh, Balasubramaniam, Lan Cao, and Richard Baskerville. 2010. “Agile Requirements Engineering Practices and Challenges: An Empirical Study.” *Information Systems Journal* 20 (5): 449–80. <https://doi.org/10.1111/j.1365-2575.2007.00259.x>.
- Satyarthi, Sandhya, and Dharendra Pandey. 2017. “Framework for Requirement Management Using Requirement Traceability.” *International Journal of Advanced Research in Computer Science* 8 (5) (05): 904–908. <https://doi.org/10.26483/ijarcs.v8i5.3480>.
- Schon, Eva-Maria, Dominique Winter, Maria Jose Escalona, and Jorg Thomaschewski. 2017. “Key Challenges in Agile Requirements Engineering.” In *Agile Processes in Software Engineering and Extreme Programming* 283:37–51. https://doi.org/10.1007/978-3-319-57633-6_3.
- SEBoK Contributors. 2023. “Guide to the Systems Engineering Body of Knowledge (SEBoK), Version 2.8.” Accessed June 1, 2023. [https://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_\(SEBoK\)](https://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK)).
- SPEC Innovations. 2022. “Innoslate 101 | Innoslate Help Center.” Accessed April 22, 2023. <https://help.innoslate.com/innoslate-101/>.
- Subarna, Ankita K. Jawale, Akshay S. Vidap, Subodh D. Sadachar, Shana Fliginger, and Srimannarayana Myla. 2020. “Using a Model Based Systems Engineering Approach for Aerospace System Requirements Management.” In *AIAA/IEEE 39TH Digital Avionics Systems Conference Proceedings* 1–8. <https://doi.org/10.1109/DASC50938.2020.9256589>.
- Tepper, Nadia A. 2010. “Exploring the Use of Model-Based Systems Engineering (MBSE) to Develop Systems Architectures in Naval Ship Design.” Master’s thesis, Massachusetts Institute of Technology. <https://hdl.handle.net/1721.1/61910>.

- Vierlboeck, Maximilian, and Roshanak R. Nilchiani. 2021. "Requirement Engineering in the Age of System and Product Complexity - A Literature Review." In *2021 IEEE International Symposium on Systems Engineering (ISSE)* 1–8. <https://doi.org/10.1109/ISSE51541.2021.9582439>.
- Zhang, Xinyu, and Xinai Zhang. 2020. "Integrated Requirements Management of Civil Aircraft." In *IOP Conference Series: Materials Science and Engineering*. Vol. 751. <https://doi.org/10.1088/1757-899X/751/1/012034>.

THIS PAGE INTENTIONALLY LEFT BLANK



DUDLEY KNOX LIBRARY

NAVAL POSTGRADUATE SCHOOL

WWW.NPS.EDU

WHERE SCIENCE MEETS THE ART OF WARFARE