2010

Exploring the use of Model-based Systems Engineering (MBSE) to develop systems architectures in naval ship design

Tepper, Nadia A.

http://hdl.handle.net/10945/24368

Author(s)	Tepper, Nadia A.
Title	Exploring the use of Model-based Systems Engineering (MBSE) to develop systems architectures in naval ship design
Publisher	
Issue Date	2010
URL	http://hdl.handle.net/10945/24368

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
07-05-2010

2. REPORT TYPE
Master's Thesis

3. DATES COVERED *(From - To)*
DEC 2009 - MAY 2010

4. TITLE AND SUBTITLE
Exploring the use of Model-Based Systems Engineering (MBSE) to develop Systems Architectures in Naval Ship Design

5a. CONTRACT NUMBER
N62271-97-G-0026

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Nadia A. Tepper

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Massachusetts Institute of Technology

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943

10. SPONSOR/MONITOR'S ACRONYM(S)
NPS

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
1. DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The U.S. Navy designs and operates the most technologically advanced ships in the world. These ships incorporate the latest in weapons technology, phased array antennas, composite structures, signature reduction, survivability, modularity, power systems, computing systems, and automation. The modern day warship is an exceptionally complex system and the design process is long and intricate, spanning several years from feasibility studies to detailed design. Model-Based Systems Engineering (MBSE) has been a recent initiative in the systems engineering community to enhance the systems engineering process by streamlining requirements traceability and improving communication amongst the various stakeholders. MBSE methods have been used in industry to develop systems architecture in a robust and comprehensive manner. In the ship design process, there is a significant need to ensure that the architecture is not only well-defined, but also addresses the needs of the stakeholders. This thesis explores the use of MBSE to develop systems architecture with application to Navy ship design and acquisition.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE

17. LIMITATION OF ABSTRACT
UU

18. NUMBER OF PAGES
166

19a. NAME OF RESPONSIBLE PERSON
Sean Tibbitts, Educational Technician

19b. TELEPHONE NUMBER *(Include area code)*
(831) 656-2319 civins@nps.edu

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Exploring the use of Model-Based Systems Engineering (MBSE) to develop Systems Architectures in Naval Ship Design

by

Nadia A. Tepper

B.S., Systems Engineering, US Naval Academy, 2003

Submitted to the Department of Mechanical Engineering and the
System Design and Management Program on May 7, 2010
in Partial Fulfillment of the Requirements for the Degrees of

Naval Engineer
and
Master of Science in Engineering and Management

at the
Massachusetts Institute of Technology
June 2010

Signature of Author [redacted]
Department of Mechanical Engineering and
System Design and Management Program
May 7, 2010

Certified by [redacted]
Trent Gooding
Associate Professor of the Practice
Thesis Supervisor

Accepted by [redacted]
Pat Hale
Director, Systems Design and Management Fellows Program
Engineering Systems Division

Accepted by [redacted]
David E. Hardt
Chairman, Department Committee on Graduate Students
Department of Mechanical Engineering

Page Intentionally Left Blank

Exploring the use of Model-Based Systems Engineering (MBSE) to develop Systems Architectures in Naval Ship Design

by

Nadia A. Tepper

Submitted to the Department of Mechanical Engineering and the
System Design and Management Program on May 7, 2010
in Partial Fulfillment of the Requirements for the Degrees of

Naval Engineer

and

Master of Science in Engineering and Management

Abstract

The U.S. Navy designs and operates the most technologically advanced ships in the world. These ships incorporate the latest in weapons technology, phased array antennas, composite structures, signature reduction, survivability, modularity, power systems, computing systems, and automation. The modern day warship is an exceptionally complex system and the design process is long and intricate, spanning several years from feasibility studies to detailed design. The plethora of new technologies being introduced in any single ship design increases the complexity of the ship design process making it ever more challenging to meet the needs of the stakeholder in terms of capability, cost, and risk. Systems architecture provides a way to understand, design, and manage this complexity by representing the system as an abstraction of elements and the relationships between those elements.

Model-Based Systems Engineering (MBSE) has been a recent initiative in the systems engineering community to enhance the systems engineering process by streamlining requirements traceability and improving communication amongst the various stakeholders. MBSE methods have been used in industry to develop systems architecture in a robust and comprehensive manner. In the ship design process, there is a significant need to ensure that the architecture is not only well-defined, but also addresses the needs of the stakeholders. This thesis explores the use of MBSE to develop systems architecture with application to Navy ship design and acquisition.

Thesis Supervisor: Trent Gooding
Title: Associate Professor of the Practice

Thesis Supervisor: Pat Hale
Title: Director, Systems Design and Management Fellows Program

Acknowledgments

I would like to thank my advisors, Trent Gooding and Pat Hale, for their support of my research and their time and effort spent providing me with valuable feedback on my thesis.

This achievement of course would not be possible without the loving and constant support of my family. My husband Brad has always been there to push me when I needed motivation and comfort me when I was stressed to the max and just needed a break, and I thank him for that. To my dearest son Maddox whose sheer existence inspires me to be a better person, mommy loves you. Thanks to my wonderful parents, Susan and Salam, for instilling in me a drive to succeed in life and be the best that I can be, without which I would not be the person I am today.

Table of Contents

List of Figures

List of Tables

1.0 Introduction

The U.S. Navy designs and operates the most technologically advanced ships in the world. These ships incorporate the latest in weapons technology, phased array antennas, composite structures, signature reduction, survivability, modularity, power systems, computing systems, and automation. The modern day warship is an exceptionally complex system and designing the ship is not an easy task. The process is long and complex, spanning several years from feasibility studies to detailed design. The plethora of new technologies being introduced in any single ship design increases the complexity of the ship design process. This complexity presents a challenge when trying to meet the needs of the stakeholder in terms of capability, cost, and risk. Systems architecture provides an effective way to understand and manage complexity and helps to overcome the challenges that complexity introduces. Systems architecture is defined as an abstract description of the entities of a system and the relationships between those entities. (Crawley, Weck, et al. 2004) In the ship design process, there is a significant need to ensure that the architecture is well-defined and addresses the needs of the stakeholders. Well-defined systems architecture early in the design process can aid decision making by quantifying design options, conducting accurate change assessments, and improving communication amongst all stakeholders, thus reducing risk and minimizing costs downstream.

Currently, the Navy does a poor job of articulating systems architectures and there are several reasons for this.

- Confusion about what is meant by systems architecture…means different things to different people
- No explicitly defined step for developing systems architecture in the overall ship design process
- Process is lost when procurement methodology shifts: Navy vs. Industry – responsibility changes hands, therefore the process of developing systems architecture is interpreted differently
- Benefits of developing systems architecture in the ship design process have not been realized by the community

Systems Architecture means different things to different people

There is much confusion as to what exactly is meant by the term systems architecture. Unfortunately there is not a single universally agreed upon definition, and that is part of the problem. There are various definitions of systems architecture given in industry and academia that include:

- The arrangement of elements and subsystems and their functional allocation to meet system requirements. (INCOSE, 2008)
- The arrangement of the functional elements into physical blocks. (Ulrich & Eppinger, 2004)

System architecture is important to understanding, designing, and managing complex systems. Every system has an architecture, whether it is planned or unplanned, and that architecture influences system behavior. There are also many ways to represent architecture through the use of models and different modeling languages. A model is an approximation, representation, or idealization of selected aspects of the structure, behavior, operation, or other characteristics of a real-world process, concept or system. (Maier and Rechtin 2002) Models have many purposes, but the primary role in systems architecting is communication. There are various modeling languages and architectural frameworks available to the systems architect including Object Process Methodology (OPM) and Object Process Language (OPL) developed by Dov Dori, the Systems Modeling Language (SysML), the Unified Modeling Language (UML), Vitech CORE, the Zachman Framework, and the Department of Defense Architecture Framework (DoDAF). The confusion surrounding systems architecture is justified and the Navy needs to standardize the systems architecting process in future ship designs in order to alleviate the muddled perceptions.

Lack of Systems Architecture in the overall Ship Design Process

A typical naval ship design is produced through an iterative, multi-disciplinary process that spans many years and involves thousands of people. It progresses from early concept studies, using lower fidelity tools to efficiently explore the broad trade space, to detailed design using higher fidelity tools to specify how the ship will be produced, tested, maintained, and operated. Throughout the design process, decisions are made and modified as information becomes available. In the past, the US Navy has focused on point based design in which the design process is characterized as a “design spiral” (Evans 1959). The design spiral, shown in

Figure 1, is classified as a point based method because it emphasizes that the designer confronts issues of resistance, weight, stability, etc. in a sequential and iterative manner until a single balanced design meets all requirements. This single design can then be developed further or used as a starting point for various tradeoff studies. Designers who utilize this approach are able to attain a feasible design; however it may not be the global optimum.

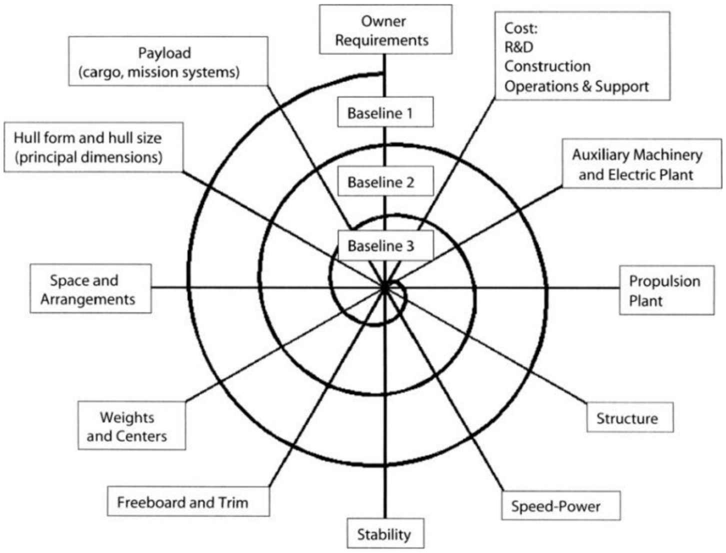


Figure 1: Classic Design Spiral (Lamb 2003)

Recently the U.S. Navy has attempted to move away from the traditional spiral in preliminary design and has advocated the use of Set Based Design (SBD) methods. Set Based Design defers detailed specifications until tradeoffs are more fully understood, therefore allowing more of the design effort to proceed concurrently. In other words, at each decision point regions of the design space where the solution is not likely to reside are eliminated. Once the design space is constricted, more detailed analysis is performed to generate additional knowledge to enable further restriction of the design space at the next decision point. The goal of Set Based Design is to obtain the global optimum design, not just a single balanced design. The Set Based Design process is depicted in Figure 2.

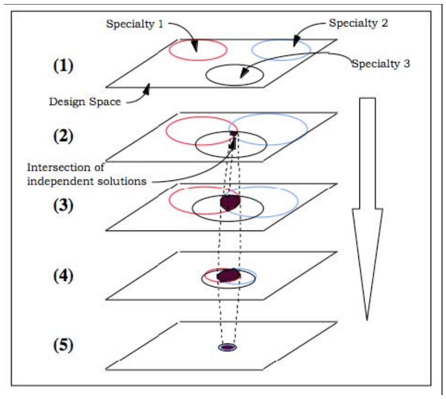


Figure 2: Set Based Design Process (Bernstein 1998)

Many experts in the field of naval ship design are concentrating more on the systems engineering process than the traditional design spiral. The two processes are essentially the same. The increasing complexity of naval ships today has made the "Ship Design Process" a "Systems Engineering Process". Ship designers are taking on the role of Systems Engineers and merging the two processes together. The traditional systems engineering process taught at Defense Acquisition University (DAU) is depicted in Figure 3 below.

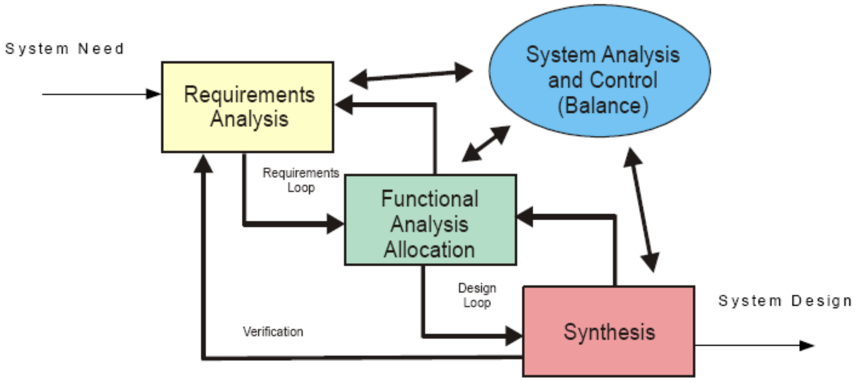


Figure 3: (DAU, Defense Acquisition Guidebook 2010)

What has been lost in this transition to a systems engineering approach to ship design is the importance of developing the systems functional, physical, and operational architecture. Ship designers spend little time on modeling and clearly defining the systems architecture which is a pivotal step in the systems engineering process. In the context of Set Based Design, it is absolutely necessary to determine early on what is going to be stable and what is going to be volatile and developing a good systems architecture is a way to enforce structure in the unchanging elements and provide flexibility in the areas that are subject to change. In industry, systems engineers are transitioning to Model-based Systems Engineering (MBSE). This transition complements the Navy's need for a process incorporating development of well-defined architecture in the context of Set Based Design. "Model-based systems engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases." (INCOSE 2007) A formalized process for developing systems architectures in early stage ship design is long overdue.

Pendulum swings: Navy vs. Industry

The naval ship designer is faced with many challenges and must draw on experience to manage the undertaking of designing such a complex system. The problem is that there is a small database of experience when it comes to naval ship design, due in large part to the small quantities of ships being produced and the long time horizons for design and construction. "One of the most remarkable characteristics of the human race is its ability not only to learn, but to pass on to future generations sophisticated abstractions of lessons learned from experience. Each generation knows more, learns more, plans more, tries more, and succeeds more than the previous one because it needn't repeat the time-consuming process of reliving prior experiences." (Maier and Rechtin 2002) The Navy has been unable to pass on these lessons learned because it has lost much of its in-house ship design experience over recent years due to a shift in the procurement methodology. Traditionally the Navy performed the ship's feasibility studies, preliminary design, and contract design, and then turned over the design to industry along with performance specifications to conduct detailed design and construction. In recent years the pendulum has swung over to industry in which they have assumed responsibility for much of the design effort starting with preliminary design. This change is highlighted in Figure 4 below.

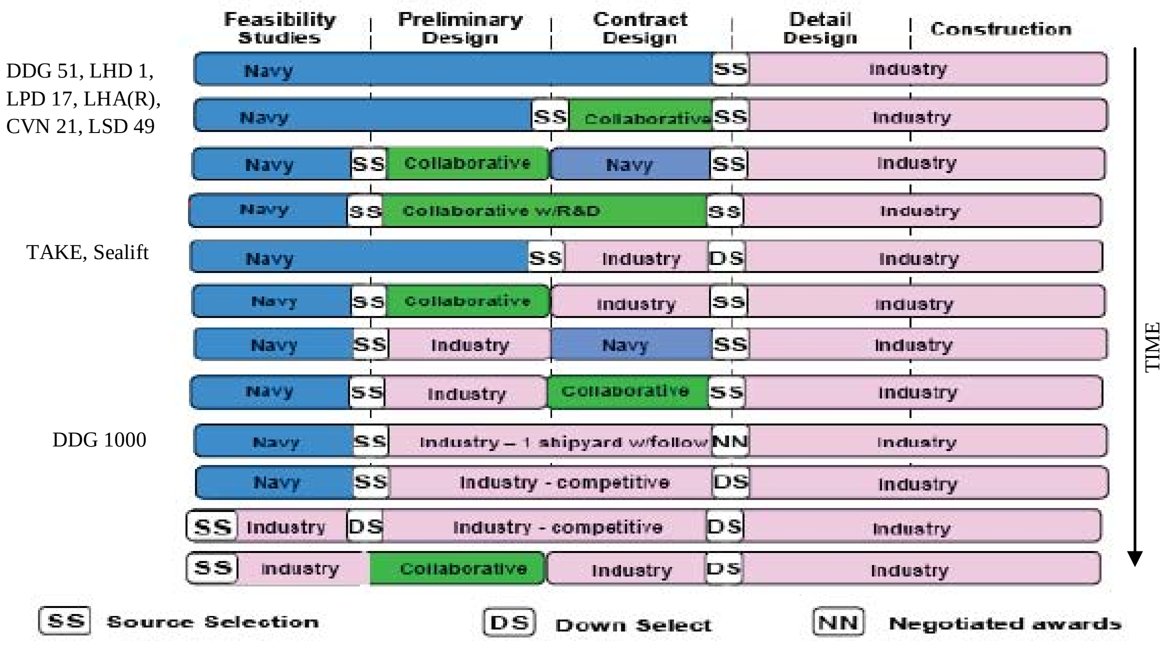
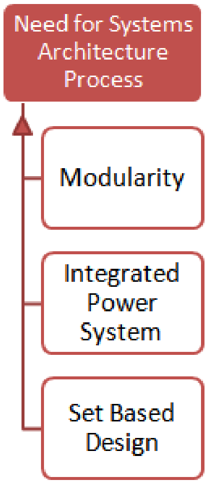


Figure 4: Navy vs. Industry (Walsh 2009)

For the DDG 1000, the new guided missile destroyer, industry was provided an overarching set of operational requirements and cost parameters instead of detailed design specifications. Operational requirements are qualitative and quantitative parameters that specify the desired capabilities of a system and serve as a basis for determining the operational effectiveness and suitability of a system prior to deployment. This acquisition strategy was put in place to encourage innovation and offer industry the maximum latitude to develop, build, deliver, and support a state-of-the-art warship. This paradigm shift forced the Navy to downsize their in-house engineering staff and send much of the design effort over to outside naval architecture firms. There have been many problems in the DDG 1000 program leading the Navy to reconsider this paradigm shift. The pendulum is starting to swing back to the Navy's side and they are trying to re-build the in-house loss of expertise in order to take back control of the design efforts from industry. As a result of these paradigm shifts and transitions between the Navy and industry, the responsibility for developing systems architectures has changed hands. The Navy's original process of developing the systems architecture was lost when the procurement methodology shifted to industry.

<u>Unrealized Benefits of developing Systems Architecture</u>

The benefits of developing well-defined systems architecture in the ship design process have not been realized by the community. The community of naval ship designers consists of highly skilled and seasoned individuals who draw heavily upon their experience in this field. They tend to leverage their tacit knowledge and let their experience and instinct guide them in their decision making. Although there is nothing inherently wrong with this methodology, it is nearly impossible to design in a solution neutral context and makes it difficult to quantify change assessments and provide traceable justification for early stage decisions. These early stage decisions lock in significant downstream effort and must be traceable back to requirements. MBSE and developing the systems architecture early in design could aid the ship designer in the decision making process. Often architectural decisions are made on a technical basis without thorough review of how it will affect the system as a whole. MBSE is a way to keep track of those decisions and requirements. In an interview with Paul Friedman, Principal Engineer at Bath Iron Works, he stated that “requirements management is a huge gap currently, and a poor function across acquisition. DDG 1000 had made some strides for better requirements management and robust traceability, but that was lost somewhere along the way. The near term compelling need for model-based architectures is to have requirements traceability.” (Friedman 2009) Despite the lack of any real process, the Navy has made attempts to force the development of systems architecture. For example, the CG(X) program was “encouraged” by the Navy to use the Department of Defense Architecture Framework (DODAF), and it was experimented with and quickly abandoned. The CG(X) designers found that the DODAF framework was challenging to use in a way that made sense and was productive. They found little advantage in using it and commented that it was clearly used primarily for software.



Systems architecture development is a critical step in the systems engineering process and enables some of the top priorities of the Navy including Modular Open System Architecture (MOSA), Integrated Power System (IPS), and Set-Based Design.

Modular Open Systems Architecture – Today’s Navy must be able to engage a range of new and existing threats and must be able to respond quickly to changing threats. In order to respond quickly to changing threats, the Navy must have the ability to easily exchange equipment to support the current mission. The Navy has addressed these concerns with a desire to move toward modular systems and open architecture. “Modularity as a governing tenet of ship design will enable more efficient reconfiguration, modernization, and maintenance, which amounts to greater operational flexibility and availability. Modular design allows the Navy to swap out and add state-of-the-art capabilities to a ship’s growth margin more rapidly than the current approach, where new tools must undergo a lengthy integration process and vie for scarce hull space.” (Edwards and Ulrich 2003) The Navy in conjunction with Northrop Grumman performed a modularity study focusing on the new cruiser design, CG(X). The study found that systems architecture development is critical for modularity and a necessary step in the design process to meet modularity demands. Open systems architecture is a term that has had quite an impact on the Navy acquisition strategy. The open architecture approach allows for horizontal integration of players and fosters an environment of competition. The ship design domain has historically been dominated by vertically integrated players, which means one company is responsible for the entire design. This approach prevents competition on subsets of the design and drives up the overall cost. As shown in Figure 5, the computer industry faced a similar situation in 1980. A single company was responsible for the entire package including chips, computer, operating system, application software, and sales which translates to an extremely expensive product for the end user. The stovepipes were ultimately dissolved in order to bring down the computer market price in response to consumer demand for lower prices.

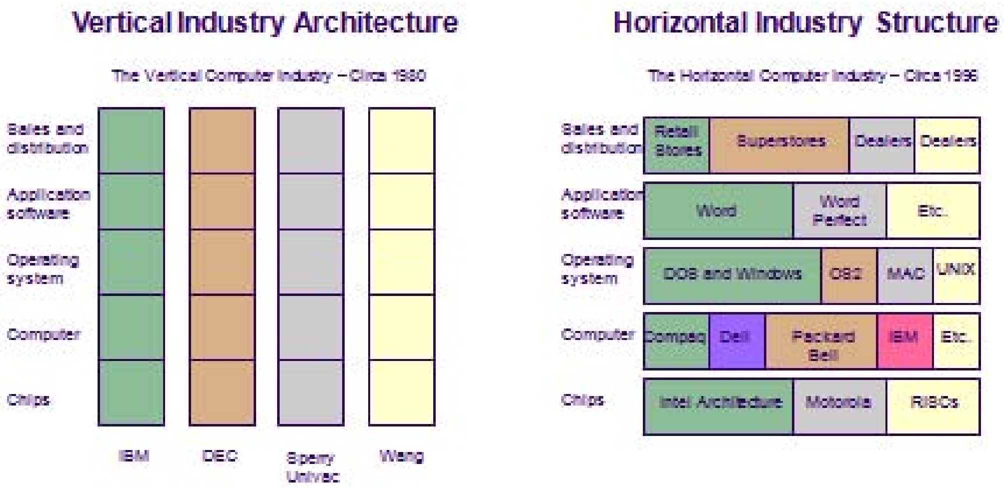


Figure 5: (Stoffel 2008)

The push for open systems architecture will serve to lower costs by creating opportunities for competition. In order to break the current stovepipes of vertically integrated companies in the DOD acquisition world, the Navy must clearly define and communicate the interfaces so that various companies can come in and bid on a portion of the design.

Integrated Power System (IPS) - The U.S. Navy has invested a considerable amount of resources over the past twenty years to develop electric power technology for the all electric ship concept, now called an Integrated Power System (IPS). This motivation for IPS ships is a result of the increasing electric load requirements anticipated in future warship designs. IPS technology is paving the way for all electric ship designs such as the *Makin Island* (LHD 8), *Lewis and Clark (T-AKE 1),* DDG-1000, and CG(X). All electric ship designs will provide the surface fleet with superior mission performance and the ability to incorporate new technology weapons such as free electron lasers, electromagnetic rail guns and electromagnetic launchers. Additionally, the IPS ship concept provides opportunities for unconventional designs that could optimize cost and performance. The systems architecture must be developed in a solution neutral manner in order to maximize the potential of all electric ships, meaning the Navy needs to re-think the way ship architectures are currently developed.

Set Based Design (SBD) - Set Based Design is an approach that constricts the design space by eliminating the regions where a feasible design does not exist, thus deferring critical decisions until further trade studies can be performed. Well-defined system architecture is absolutely critical for set based design methods to work successfully. The architecture provides the framework necessary to "keep score" of decisions and keep track of what elements of the design are stable and which are flexible. Clear communication of systems architecture is a way to enforce structure in the unchanging elements and provide flexibility in the areas that are subject to change.

MBSE has been an initiative of the International Council of Systems Engineers (INCOSE) and promises to be a more rigorous and effective means of developing complex systems. At the heart of MBSE is requirements traceability and enhanced communication. It also has the potential to improve decision making by providing accurate change assessments and by quantifying design options in terms of cost and risk.

Requirements Traceability – Model-based architecture provides requirements traceability for each and every element of the design. Too often in the ship acquisition community there are developed systems that do not effectively meet the needs of the stakeholders. Requirements get lost or manipulated over time and it is extremely difficult to maintain traceability between design documents and the requirements management tool.

Communication - Defining the overall architecture, the form and function interactions and interfaces, is a way to understand and communicate complex systems. One of the primary benefits of systems architecture development and model-based systems engineering is the ability to communicate clearly using a language that reaches out to all stakeholders. Stakeholders have different experiences and backgrounds, some are subject matter experts and some are not, and using a common system design language will bridge communications gaps between the experts and the systems engineers (or the Navy and the shipbuilder). Often knowing what to build, which includes requirements elicitation, technical specification, and prioritization, is the most difficult systems engineering phase in the life cycle. MBSE serves to mitigate ambiguity and promote consistency of thought and expression across the entire program team.

Decision-Making – "Two different kinds of decisions, both critical to success, are made in architecting – value judgments and technical choices." (Rechtin 1991) MBSE provides the architectural basis for those value judgments and technical decisions, driven by real functional requirements. The model keeps track of all decisions and rationale in a central repository, thus serving as the project memory. Additionally, the traceability inherent in a systems model allows for more accurate change assessments and alternatives analysis. The designer is able to see how a small change in one aspect of the design can drastically affect the whole. Risk and cost can also be incorporated into the model to enhance the decision-making process. Executable models can be used in an analysis of alternatives (AoA) by conducting system design trade-offs and use cases can be incorporated into the model to verify that the system capability satisfies mission requirements.

The primary responsibilities of the ship design team are to quantify design options, conduct change assessments and to have a sound basis for decision making presentable to the customer. Clearly defining the systems architecture of the ship early can improve requirements traceability, enhance communication, and augment the decision making process. This thesis explores a model-based approach to systems architecture with application to Navy ship design and acquisition. Specifically, this thesis seeks to answer the following questions:

1. *Can MBSE be used to develop the systems architecture of a naval warship?*
2. *Does MBSE provide any benefit to the designer? In what way?*
3. *Is the decision making process enhanced through the use of modeling?*
4. *Where does systems architecture development fit into the overall ship design process?*
5. *What is the right tool to be used in developing the architecture?*

2.0 Background

In order to set the context for the model-based methodology described in this thesis, a review of related concepts and terminology is presented.

2.1 DOD Acquisition Lifecycle

The U.S. Department of Defense (DOD) has put in place rigid acquisition guidelines for system developers called the Defense Acquisition System. The Defense Acquisition System exists so that there is proper management and oversight in the development and acquisition of large-scale, complex systems. The fundamental acquisition procedures and policies are outlined in DOD Directive 5000.01 *The Defense Acquisition System* and DOD Instruction 5000.02 *Operation of the Defense Acquisition System.* The acquisition process is structured into discrete phases separated by major programmatic reviews or decision points called “Milestones”. The DOD Acquisition Lifecycle framework is depicted in Figure 6.

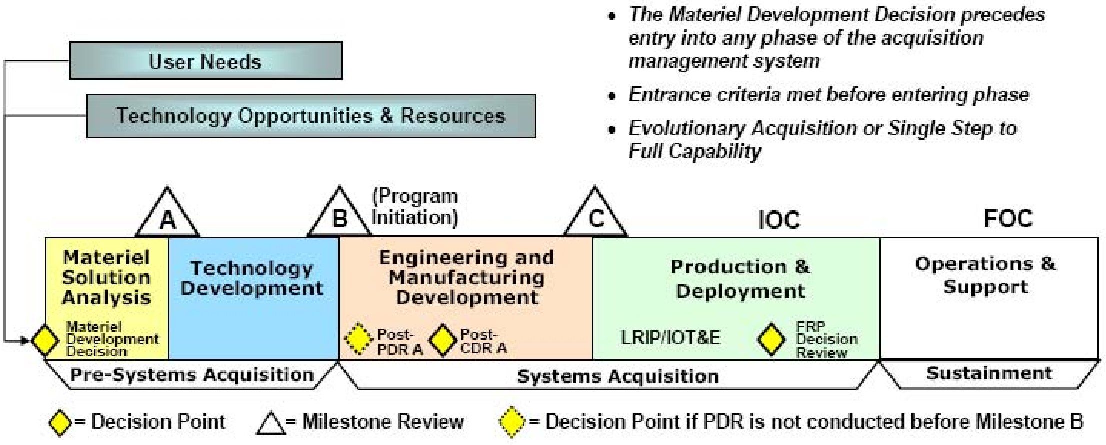
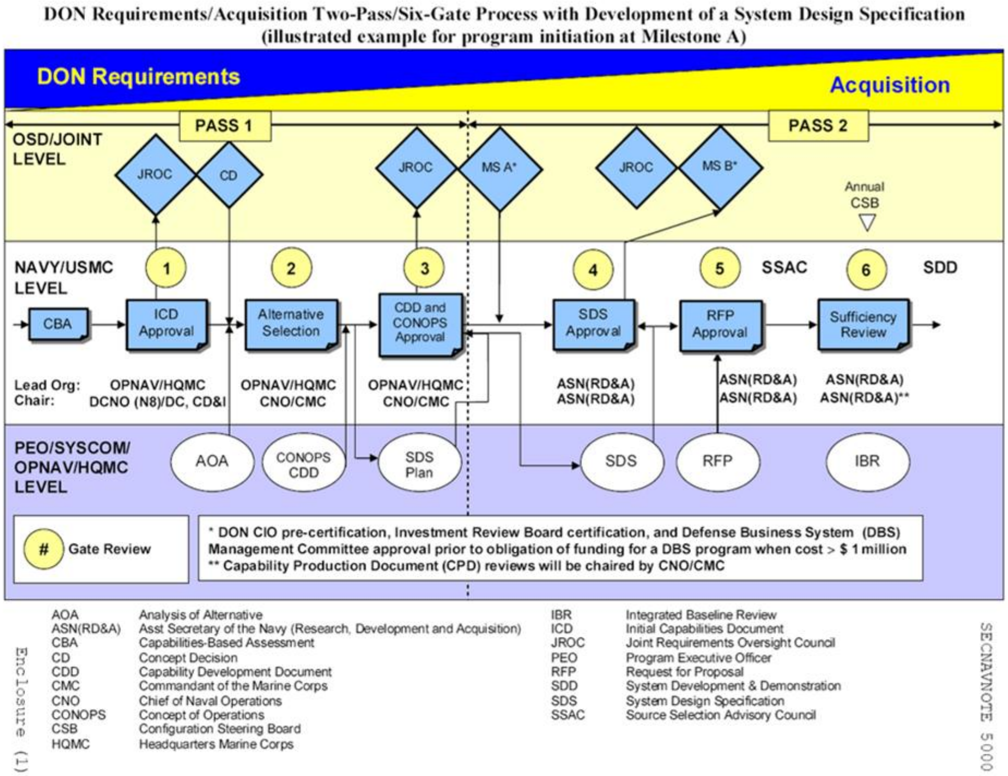


Figure 6: DOD Acquisition Lifecycle ((DOD 2008))

As shown, Milestone A represents the beginning of the technology development phase, Milestone B represents program initiation, and Milestone C corresponds to a production commitment. IOC stands for Initial Operational Capability and FOC stands for Full Operational Capability. In an attempt to improve governance and insight into the development, establishment, and execution of acquisition programs within the Department of the Navy (DON), SECNAVNOTE 5000 implemented the “2-pass, 6-gate” process. “The goal of the review process is to ensure alignment between Service-generated capability requirements and acquisition, as well as improving senior leadership decision-making through better understanding of risks and costs throughout a program's entire development cycle." (SECNAV 2008) This process for a program initiation at Milestone A is depicted in Figure 7.



* DON CIO pre-certification, Investment Review Board certification, and Defense Business System (DBS) Management Committee approval prior to obligation of funding for a DBS program when cost > $ 1 million

** Capability Production Document (CPD) reviews will be chaired by CNO/CMC

AOA	Analysis of Alternative	IBR	Integrated Baseline Review
ASN(RD&A)	Asst Secretary of the Navy (Research, Development and Acquisition)	ICD	Initial Capabilities Document
CBA	Capabilities-Based Assessment	JROC	Joint Requirements Oversight Council
CD	Concept Decision	PEO	Program Executive Officer
CDD	Capability Development Document	RFP	Request for Proposal
CMC	Commandant of the Marine Corps	SDD	System Development & Demonstration
CNO	Chief of Naval Operations	SDS	System Design Specification
CONOPS	Concept of Operations	SSAC	Source Selection Advisory Council
CSB	Configuration Steering Board		
HQMC	Headquarters Marine Corps		

Figure 7: (SECNAV 2008)

Pass 1 encompasses three gate reviews, gates 1, 2, and 3, led by the Chief of Naval Operations (CNO) or the Commandant of the Marine Corps (CMC). Pass 1 starts prior to Concept Decision (CD), progresses through the Concept Refinement phase, and ends after the Gate 3 review. It includes Department of the Navy (DON), the Office of the Secretary of Defense (OSD), and Joint processes for approval of the following documentation: Initial Capabilities Document (ICD), Analysis of Alternative (AoA), Capabilities Development Document (CDD), Concept of Operations (CONOPS), and the System Design Specification (SDS) Development Plan.

Pass 2 is led by the Component Acquisition Executive and encompasses gates 4, 5, and 6. Pass 2 starts after Gate 3 and ends after Milestone B which corresponds to the initial portion of the System Development and Demonstration (SDD) Phase. Gate 4 review approves the SDS and then authorizes a program to continue to Gate 5 or Milestone B. Gate 5 recommends to the

Milestone Decision Authority (MDA) approval of the release of the SDD Request for Proposal (RFP) to industry as authorized by the Acquisition Strategy. Gate 6 review serves to assess the overall program health including readiness for production, the sufficiency of the SDS, the Earned Value Management System (EVMS) Program Management Baseline (PMB), and the Integrated Baseline Review (IBR). Follow-up reviews will be conducted to endorse or approve the Capabilities Production Document (CPD).

The purpose for describing the DOD acquisition process is to highlight the fact that the current acquisition strategy is strictly document-driven, based on traditional programmatic review techniques. Figure 7 illustrates the emphasis of documentation in the "2-pass, 6-gate" process as the deliverable for decision milestones and gate review. Key acquisition documents include the Initial Capabilities Document (ICD), the Analysis of Alternatives (AoA), the Concept of Operations (CONOPS), the Capabilities Development Document (CDD), the Capabilities Production Document (CPD), the System Design Specification (SDS), the Test and Evaluation Master Plan (TEMP), the Acquisition Program Baseline (APB), and the contract. "The purpose of life-cycle reviews in the traditional development environment was to synchronize a program's cost, schedule, and technical baselines in order to review the program in its entirety. Such reviews necessarily relied upon paper documents because of the inability of early information systems to provide electronic reviews of such programs. Hence a practice of paper-oriented life-cycle reviews was built around available technology, and this practice continues to this day." (Balmelli, et al. 2006)

2.2 Capabilities Driven Architecture

The DOD has implemented a capabilities-driven development system called the Joint Capabilities Integration and Development System (JCIDS).

> *A central objective of the Quadrennial Defense Review was to shift the basis of defense planning from a "threat-based" model that has dominated thinking in the past, to a "capabilities-based" model for the future. This capabilities-based model focuses more on how adversaries fight, rather than specifically whom the adversary might be or where a war might occur. It recognizes that it is not enough to plan for large conventional wars in distant theaters. Instead, the United States must identify the capabilities required in order to defeat adversaries who will rely on surprise, deception, and asymmetric warfare to achieve their objectives.*
>
> *-Donald Rumsfeld*

This new process represents a methodology shift in which requirements are derived in a top-down fashion directly from operational capability as opposed to the traditional bottom-up approach. In Figure 8, the left side represents the way in which requirements used to be developed where all four services generated their respective requirements in-house and fed those requirements up to the next level. This led to difficult integration and sub-optimum System-of-System requirements generation and ultimately drove the Secretary of Defense Donald Rumsfeld to implement the JCIDS process, depicted on the right side of Figure 8.

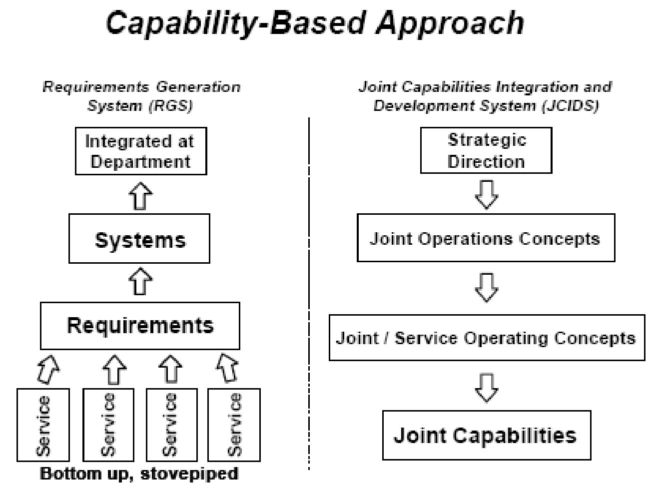


Figure 8: (Walker 2005)

The objective of the JCIDS process is to ensure the capabilities required by the joint warfighter are identified with their associated operational performance criteria in order to successfully execute the missions assigned. (CJCS, CJCSI 3170.01G Joint Capabilities Integration and Development System Instruction 2009) The JCIDS process is closely linked to the Defense Acquisition System and the relationships are shown in Figure 9.

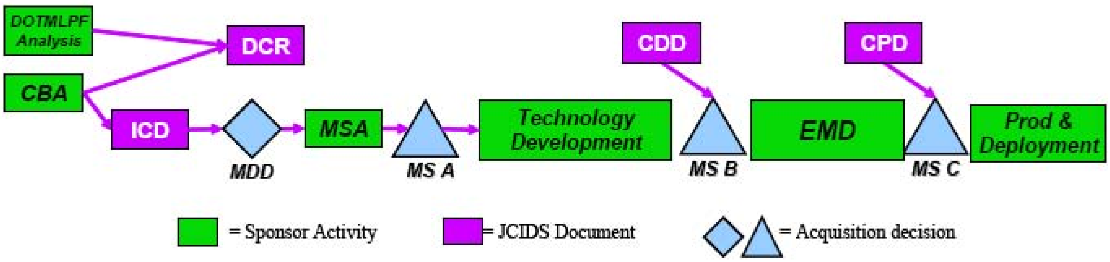


Figure 9: (CJCS 2009)

Figure 9 highlights the fact that JCIDS uses strictly a document-based approach to requirements generation. MBSE has tremendous potential to improve the JCIDS process by integrating all the required documentation into a single database. It would also provide a viable way to design with capabilities-based requirements in a solution neutral context. Solution neutral means starting from capabilities and deriving the system requirements and architecture in a top-down fashion, not jumping to the answer by pulling the last ship's requirements off the shelf (as the ship design community often does). The architecting process must be robust enough to accommodate emerging capability needs and must focus first on the problem space before jumping into the solution space. "While recognition of focus on capabilities-driven systems architecting has given rise to the fairly recent development of system architecting methods, frameworks, and processes, what is lacking at this time is a defined method for architecting—the development of the architecture itself." (Whitcomb, et al. 2008)

2.3 Systems Engineering (SE)

Systems today are expected to perform at levels undreamed of a generation ago. Increasing system complexity is driven by competitive pressures demanding increased capability at reduced costs and within shorter delivery cycles. The interconnectivity among systems and the requirement for increased functionality requires integrated, system of systems (SoS) optimization. The integrated nature of these complex systems presents quite a challenge to system designers.

The term "Systems Engineering" means different things to different people. One could say that systems engineering has suffered from an identity crisis over the years. The "classical view" of systems engineering leans toward being a way of thinking or approach to design, whereas recent definitions, or the "expanded view", term it as an engineering discipline. The distinction is significant, but heavily debated and to no avail. There have been numerous definitions of systems engineering presented over the years and they are shown in Table 1. The table shows that the definitions have evolved over the last 25 years to include the role of management in systems engineering and the increasing importance of life cycle considerations.

Source	**Definition of Systems Engineering**
Mil-Std 499A (1974)	The application of scientific and engineering efforts to: (1) transform an operational need into a description of system performance parameters and a system configuration through the use of an iterative process of definition, synthesis, analysis, design, test, and evaluation; (2) integrate related technical parameters and insure compatibility of all related, functional and program interfaces in a manner that optimizes the total system definition and design; (3) integrate reliability, maintainability, safety, survivability, human, and other such factors into the total technical engineering effort to meet cost, schedule, and technical performance objectives.
Chase (1974)	The process of selecting and synthesizing the application of the appropriate scientific and technical knowledge to translate system requirements into system design and subsequently to produce the composite of equipment, skills, and techniques that can be effectively employed as a coherent whole to achieve some stated goal or purpose.
Sailor (1990)	Both a technical and management process; the technical process is the analytical effort necessary to transform an operational need into a system design of the proper size and configuration and to document requirements in specifications; the management process involves assessing the risk and cost, integrating the engineering specialties and design groups, maintaining configuration control, and continuously auditing the effort to ensure that cost, schedule, and technical performance objectives are satisfied to meet the original operational need.
Wymore (1993)	The intellectual, academic, and professional discipline the primary concern of which is the responsibility to ensure that all requirements for a bioware/hardware/software system are satisfied throughout the life cycle of the system.
Ramo (1993)	A branch of engineering that concentrates on the design and application of the whole as distinct from the parts…looking at the problem in its entirety, taking into account all the facets and variables and relating the social to the technical aspects.
INCOSE - International Council on Systems Engineering (1999)	An interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem.

Table 1: Systems Engineering Definitions

The definition of Systems Engineering used throughout this paper is that which is given by the International Council on Systems Engineering (INCOSE), "Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem."

The discipline of Systems Engineering has emerged in response to ever increasing system complexity. It drives the balanced development of systems in terms of cost, schedule, performance, and risk and verifies that the technical solutions satisfy customer requirements. Systems Engineering has been proven as an effective way to manage complex and often technologically challenging problems.

2.4 Systems Engineering Process

The increasing complexity of naval ships today has made the "Ship Design Process" essentially a "Systems Engineering Process". Ship designers are filling the role as Systems Engineers and merging the two processes together. System engineering is an interdisciplinary approach as explained earlier and includes both management processes and technical processes. A process is defined as a logical sequence of tasks performed to achieve a particular objective.

There has been an attempt to codify the practice of systems engineering through standards which have evolved over the last several years. The taxonomy of standards includes systems engineering process standards, architecture frameworks, methods, modeling standards, and data exchange standards. Figure 10 shows the evolution of the process standards since 1969 starting with Mil-Std-499, a military standard of the U.S. Department of Defense. The early standards, such as the Mil-Std-499, focused mostly on the verification and development life cycle functions whereas the later standards encompass the entire system life cycle.

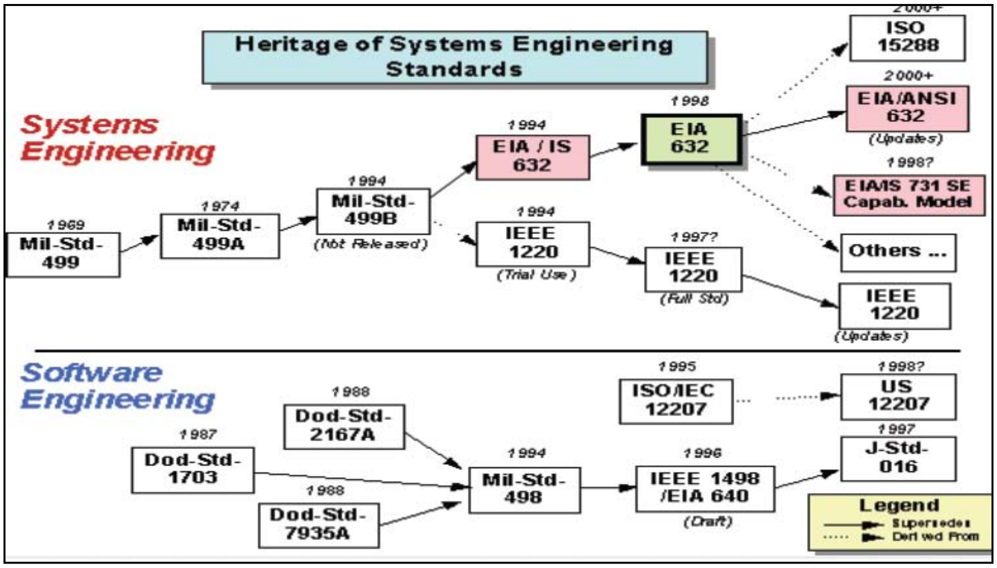


Figure 10: (INCOSE 2004)

"The Systems Engineering Process (SEP) is a comprehensive, iterative and recursive problem solving process, applied sequentially top-down by integrated teams. It transforms needs and requirements into a set of system product and process descriptions, generates information for decision makers, and provides input for the next level of development." (DAU 2001)

Figure 11 shows the systems engineering process currently taught at the Defense Acquisition University (DAU). Not all processes are the same, but the one represented here is fairly typical across the systems engineering community. The process includes Inputs/Outputs, Requirements Analysis, Functional Analysis and Allocation, Requirements Loop, Synthesis, Design Loop, Verification, and System Analysis and Control.

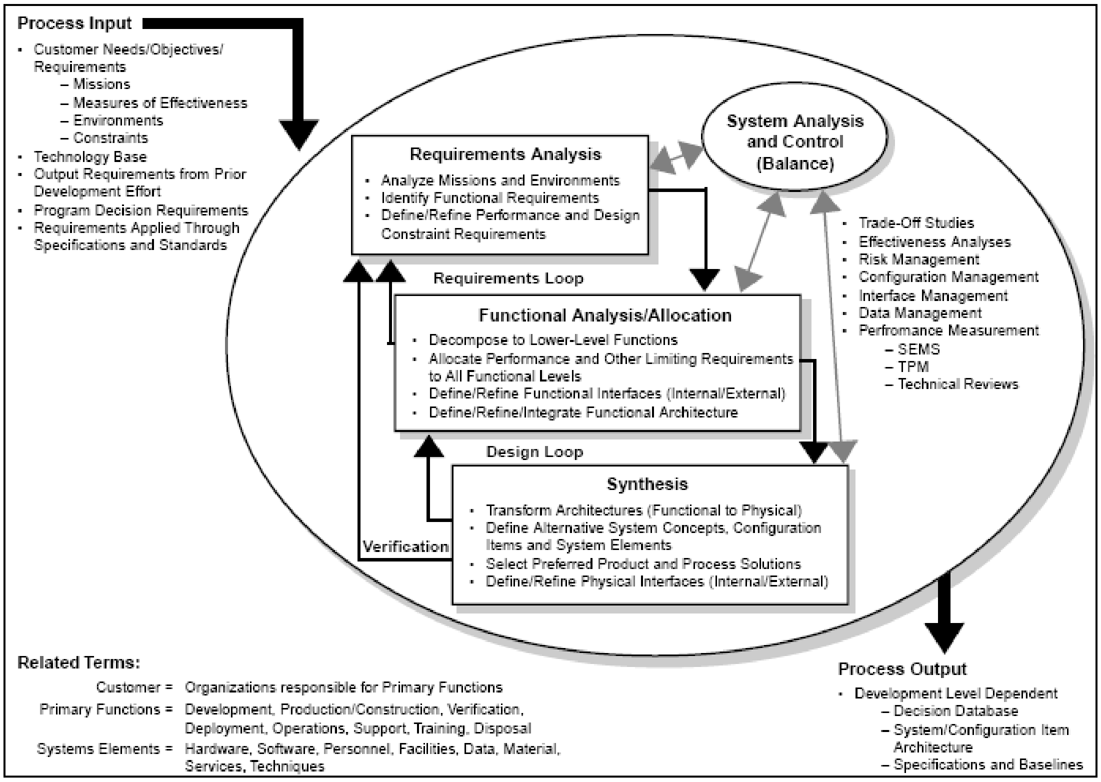


Figure 11: (DAU 2001)

The systems engineering process begins by identifying the stakeholders and gathering their needs, goals, and objectives. This is represented by "Process Inputs" in Figure 11 and is essentially a list of customer requirements including missions, measures of effectiveness, environments, and constraints. The first step of the systems engineering process is Requirements Analysis. The given customer requirements are translated into functional and performance requirements ensuring that they are unambiguous, measurable, verifiable, comprehensive, and concise.

The next step is Functional Analysis/Allocation in which the top level system functions are analyzed and decomposed into lower-level functions. The associated performance requirements are then parsed and allocated to the lower-level functions creating the systems functional architecture. "The nature of complex systems today requires a high degree of communication exchanges between distributed functions to achieve a given systems mission. This is extremely difficult to describe without the aid of a functional architecture that describes the organization of functions in the context of a desired operational mission or capability. A functional architecture expresses the detailed functional, interface, and temporal aspects of the system that are essential to gain sufficient insight and to communicate unambiguously the behavior of the system in its intended operational environment.” (DAU 2010)

The Synthesis phase represents the physical decomposition of the system and it evolves together with the requirements and functional architecture. The lower tier functional and performance requirements are allocated to the lower level components, thus creating the physical architecture. The development of the physical architecture is an iterative and recursive process that will define the systems form and the arrangement of the system components and associated interfaces. Synthesis is complete when the physical architecture has been decomposed down to the lowest system element. Verification is a critical part of the systems engineering process to ensure that the system design satisfies requirements. “The system engineering process is the engine that drives the balanced development of system products and processes applied to each level of development, one level at a time.” (DAU 2001)

2.5 Systems Architecture and Architecting

Systems today are increasing in complexity due to demands for more functionality, higher performance, lower costs, and improved human interfaces. Systems architecture development is a critical early step in the design process because it determines the system’s concept and behavior. “System architecture is an abstract description of the entities of a system and the relationships between those entities.” (Crawley, Weck, et al. 2004) Systems architecture is important because it provides a way to effectively understand, design, and manage complex systems. It plays a central role in giving a system its behavior and “ilities” (flexibility, adaptability, reliability, etc) as well as recognizing the systems emergent behavior and complexity as shown in Figure 12.

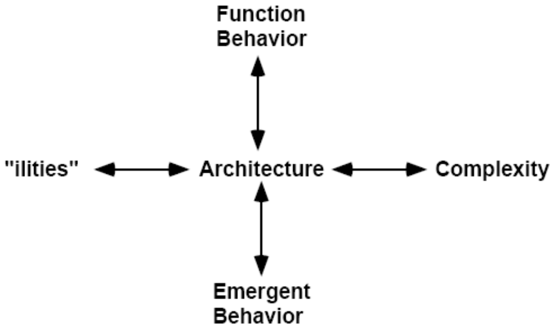


Figure 12: (Crawley, Weck, et al. 2004)

Systems architecture means different things to different people and there is no single universally agreed upon definition. Definitions from industry and academia include:

- The arrangement of elements and subsystems and their functional allocation to meet system requirements. (INCOSE, 2008)
- The arrangement of the functional elements into physical blocks. (Ulrich & Eppinger, 2004)
- The arrangement of function and feature that maximizes some objective. (Ring, 2001)
- The embodiment of concept, and the allocation of physical/informational function to elements of form and definition of structural interfaces among the elements. (Crawley, 2003)
- The structure (in terms of components, connections, and constraints) of a product, process, or element. (Rechtin & Maier, 2002)
- The structure of components, their relationships, and the principles and guidelines governing their design and evolution over time. (DoDAF, 2007)
- The fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution. (IEEE AWG)

The definition given by Edward Crawley, MIT professor, is the most inclusive and represents that which is desired in the ship design community. This definition is augmented by Figure 13, in which “Function” is related by “Concept” to “Form”. Function is defined as “the activities, operations and transformations that cause, create or contribute to performance”, where Form is “the physical/informational embodiment which exists or has the potential to exist”. (Crawley 2007) In other words, Function is what the system does and Form is what the system is. Concept is defined by Crawley as, “a product or system vision, idea, notion or mental image which maps Function to Form.” (Crawley 2007)

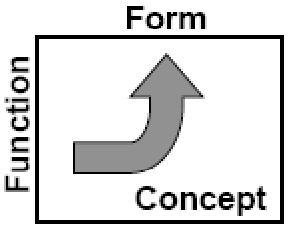
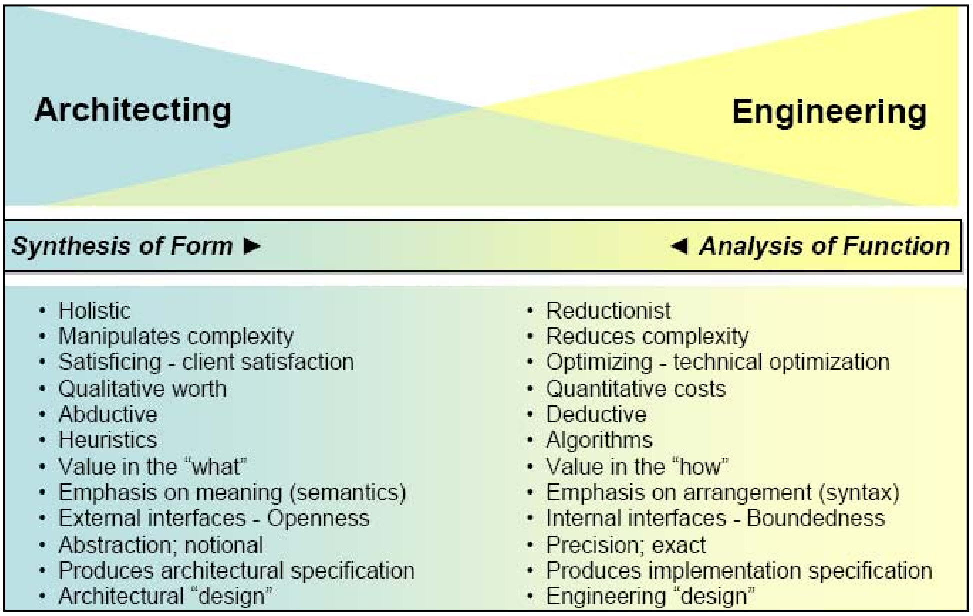


Figure 13: (Crawley 2007)

“Systems architecting combines the theory and engineering of systems with the theory and practice of architecting.” (Rechtin 1991) The distinction between systems engineering and systems architecting is often misunderstood and the line is not always clearly drawn. “Generally speaking, engineering deals almost entirely with measurables using analytic tools derived from mathematics and the hard sciences; that is, engineering is a deductive process. Architecting deals largely with unmeasurables using non-quantitative tools and guidelines based on practical lessons learned; that is, architecting is an inductive process.” (Maier and Rechtin 2002) A summary of the differences between architecting and engineering is shown in Figure 14.



Architecting	Engineering
Synthesis of Form ►	◄ *Analysis of Function*
Holistic	Reductionist
Manipulates complexity	Reduces complexity
Satisficing - client satisfaction	Optimizing - technical optimization
Qualitative worth	Quantitative costs
Abductive	Deductive
Heuristics	Algorithms
Value in the “what”	Value in the “how”
Emphasis on meaning (semantics)	Emphasis on arrangement (syntax)
External interfaces - Openness	Internal interfaces - Boundedness
Abstraction; notional	Precision; exact
Produces architectural specification	Produces implementation specification
Architectural “design”	Engineering “design”

Figure 14: (Mercer 2008)

Mercer gives the following definitions to highlight the differences between Architecting and Engineering.

Engineering	The application of scientific and mathematical principles to *practical ends* such as the design, manufacture, and operation of efficient and economical structures, machines, processes, and systems.
Architecting	The application of scientific and mathematical principles to the *representation of the form of a system in support of practical ends* such as the planning, analysis, and engineering of efficient and economical systems.

***Definitions from (Mercer 2008)

Despite the attempts to separate architecting and engineering, the overlap is unavoidable. It is safe to say that architects are not "general engineers" but are specialists in reducing complexity, uncertainty, and ambiguity to workable concepts whereas systems engineers are masters of making feasible concepts work. (Rechtin 1991) The real question is how does system architecture fit into the overall systems engineering process. Figure 15 shows the emphasis and duration of the Architecture Design process in a typical DOD acquisition life cycle.

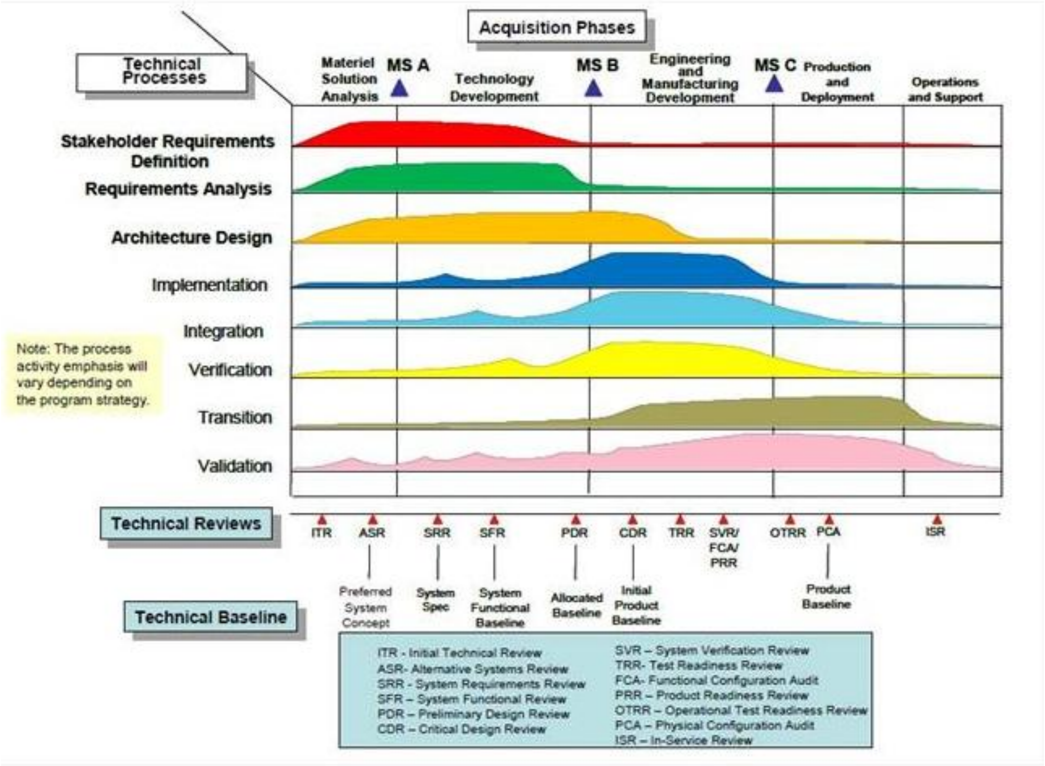


Figure 15: (DAU 2010)

Developing the systems architecture is a trade and synthesis process. “It translates the outputs of the Stakeholder Requirements Definition and Requirements Analysis processes into alternative design solutions and selects a final design solution.” (DAU 2010) The architecting takes place in the functional allocation block in the systems engineering process as defined by DAU in Figure 16 and also in the Vee Model presented by Mercer in Figure 17.

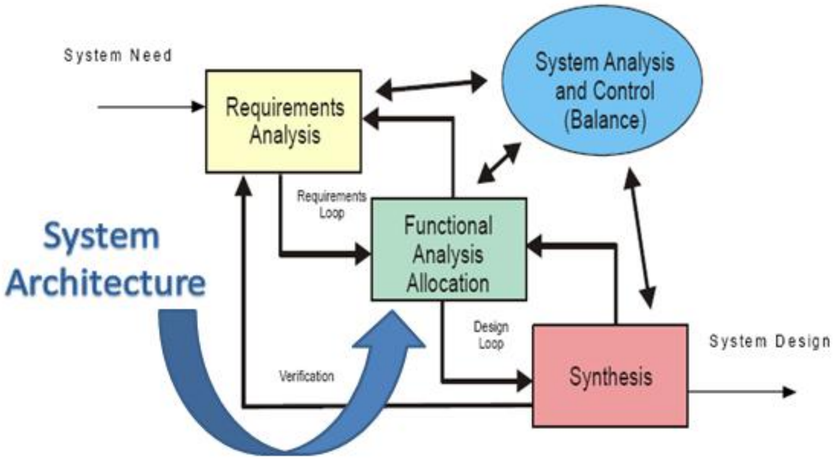


Figure 16: Role of Systems Architecting within Systems Engineering (DAU 2010) modified

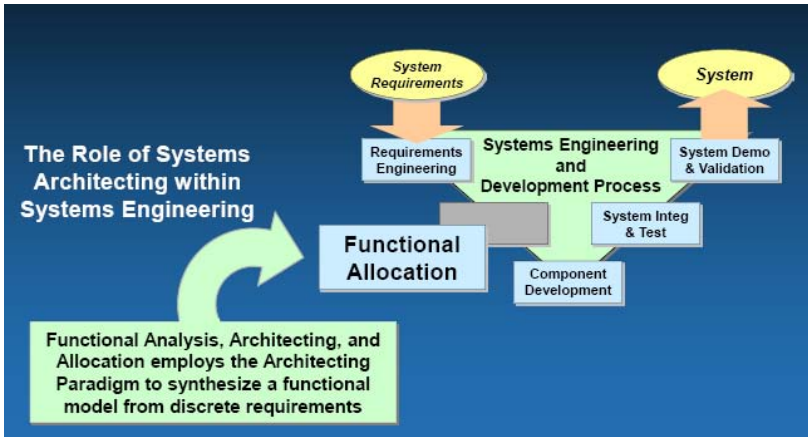


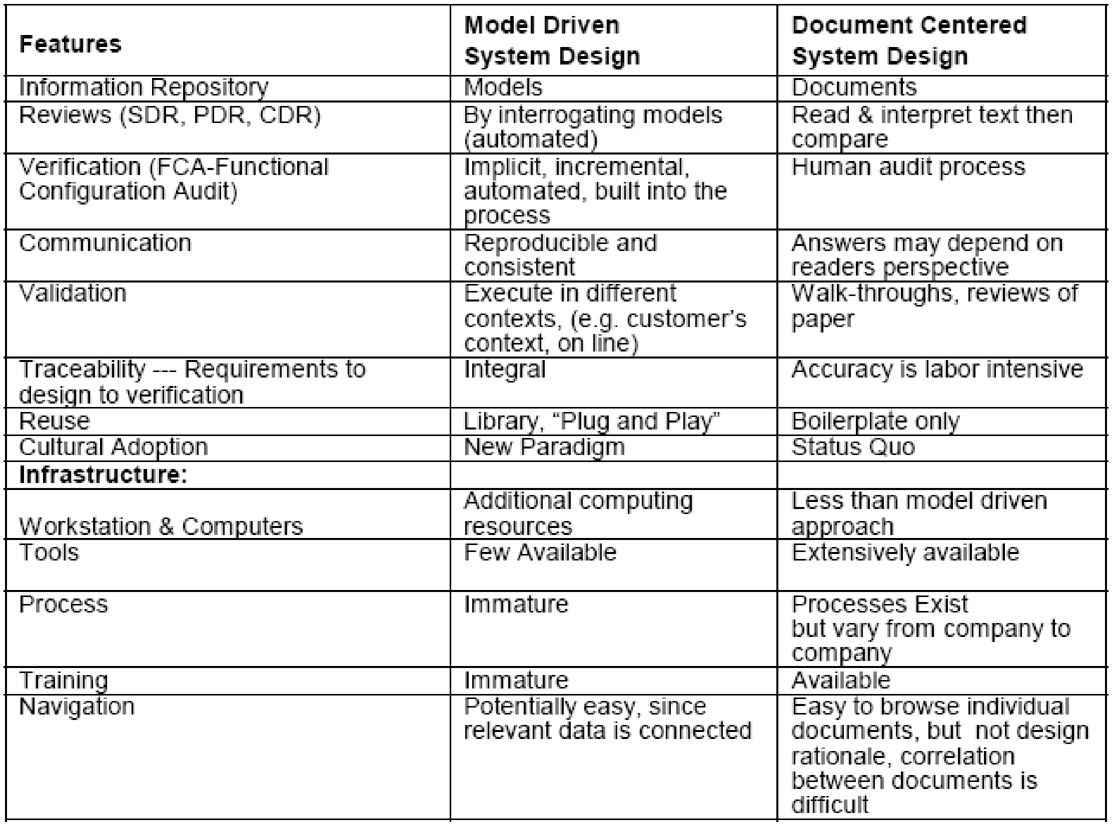
Figure 17: (Mercer 2008)

Systems architecture has become a critical step in the process for designing and developing complex systems. It is time to recognize the contribution and define a process for creating systems architecture within the ship design community.

2.6 Model-Based Systems Engineering (MBSE)

Model-Based Systems Engineering (MBSE) is not a new concept. In fact, the idea of using models to assist the systems engineering process has been around for quite some time and is used extensively by the software community, especially since the advent of the Unified Modeling Language (UML) in the 1990's. A model is "a collection of all the artifacts that describe the system." (Balmelli, et al. 2006) A key feature of a model is that is an abstraction and can be represented in many forms. The mathematical system theory behind MBSE was explicated by A. Wayne Wymore in 1993 and serves as the basis for the development of models and designs of large-scale, complex systems consisting of personnel, machines, and software. INCOSE defines MBSE as "the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases." (INCOSE 2007)

Traditionally, ship design has employed a document-based system engineering approach characterized by the generation of textual specifications, design documents, sketches, and diagrams that attempt to capture the system requirements and system specifications. A ship is a large system and therefore the requirements and system specifications typically represent two different documents. These documents are used to communicate design information to all stakeholders. The systems engineer is then responsible for controlling the documentation and ensuring the documents and drawings are valid, complete, and consistent, and that the developed system complies with the documentation. Document-based systems engineering relies on a concept of operation (CONOPS) document to define how the system is used to support the required missions. A functional analysis is then performed to allocate the top-level functions to the systems components. Block diagrams are used to capture the overall system design and are stored as separate files included in the system design documentation. Typically the requirements are managed through the use of requirements management tool such as Telelogic DOORS. Traceability between requirements and the ship design must be done manually using a tool like DOORS as there is no formal link between the requirements database and the architecture/design documents. The document-based approach can be rigorous and time consuming as information is often spread across several documents. It is also difficult to understand a particular aspect of the system and to perform the necessary traceability and change impact assessments necessary for a complex ship design. Engineers are forced to communicate by passing design documents back and forth which is not only inefficient, but highly error prone. A comparison table of model-based vs. document-based design is shown in Table 2.



Features	Model Driven System Design	Document Centered System Design
Information Repository	Models	Documents
Reviews (SDR, PDR, CDR)	By interrogating models (automated)	Read & interpret text then compare
Verification (FCA-Functional Configuration Audit)	Implicit, incremental, automated, built into the process	Human audit process
Communication	Reproducible and consistent	Answers may depend on readers perspective
Validation	Execute in different contexts, (e.g. customer's context, on line)	Walk-throughs, reviews of paper
Traceability --- Requirements to design to verification	Integral	Accuracy is labor intensive
Reuse	Library, "Plug and Play"	Boilerplate only
Cultural Adoption	New Paradigm	Status Quo
Infrastructure:		
Workstation & Computers	Additional computing resources	Less than model driven approach
Tools	Few Available	Extensively available
Process	Immature	Processes Exist but vary from company to company
Training	Immature	Available
Navigation	Potentially easy, since relevant data is connected	Easy to browse individual documents, but not design rationale, correlation between documents is difficult

Table 2: (Baker, et al. 2009)

The traditional document centered approach has several drawbacks in addition to those described in Table 2. Defining system functionality is an important step in the architecting process, and documents can often be unsuitable for capturing the various levels of functionality. It is also challenging to keep documentation synchronized with the current state of the design, especially in cases of extreme complexity and frequent design changes. When documents are shared electronically, it is easy to see how efforts can be duplicated, leading to inefficiencies in the design process. Developing a complex system involves many people across multiple engineering domains, therefore tracing the source of an error, should it occur, along a paper trail is extremely difficult.

MBSE provides the system designer a rigorous means for capturing and integrating system requirements, design, analysis, and verification information. With the increasing capability of computer processing, storage, and network technology, MBSE is becoming more prevalent in the field of systems engineering. In fact, the INCOSE 2020 vision anticipates that all systems engineering efforts will eventually transition from a document-based approach to a model-based approach as depicted in Figure 18.

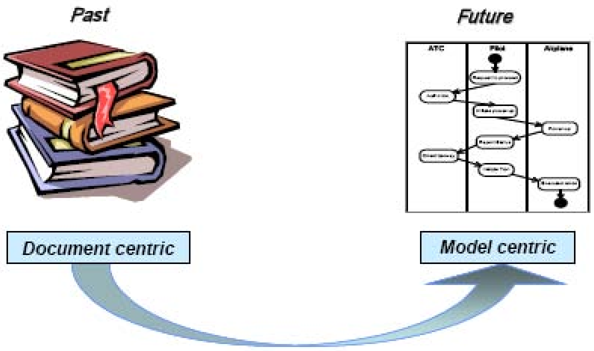


Figure 18 : (INCOSE 2007)

Friedenthal lists the benefits of MBSE over a document-based approach recognized by the overarching community of systems engineers. (Friedenthal 2008) This list includes:

- Enhanced communication

One of the key advantages of using MBSE is the ability to clearly communicate the system design using a language that reaches out to all stakeholders. The use of models and a common systems language provides a way to mitigate ambiguity and promote consistency of thought across the entire program team. MBSE enhances communication by providing a complete representation of the system in a single data repository, a “one stop shop”. MBSE helps to manage complexity by viewing the system at various levels of abstraction and provides the ability to integrate views of the system from multiple perspectives.

- Reduced development risk

MBSE supports continuous and ongoing requirements validation and design verification, thus helping to mitigate associated development risk. It has also been shown to provide more accurate cost estimates to develop the system.

- Improved quality

MBSE facilitates rigorous traceability between requirements, design, analysis, and testing which corresponds to improvement in quality over the traditional document centric method. The inherent traceability leads to more complete, unambiguous, and verifiable requirements. All aspects of the design are contained in a single relational database providing enhanced design integrity by eliminating redundancy and inconsistency.

- Increased productivity

One of the obvious benefits to using a MBSE approach is the quickness and ease in which a design change can be implemented. It allows for immediate feedback on change assessments or impact analyses. Another advantage that improves overall productivity is the potential for reuse of existing models to support design evolution. As mentioned earlier, the Defense Acquisition System is heavily reliant on document-based programmatic reviews in order to assess and approve the system design to move forward. MBSE provides automated document generation so the current state of the design can instantly be captured and the emphasis can be placed on developing the system instead of formatting the documentation.

When using a model-based approach, the modeling language is used to define the requirements architecture, system design and system architecture. In Figure 19 below, it is easy to see where the modeling language fits into the overall systems engineering process.

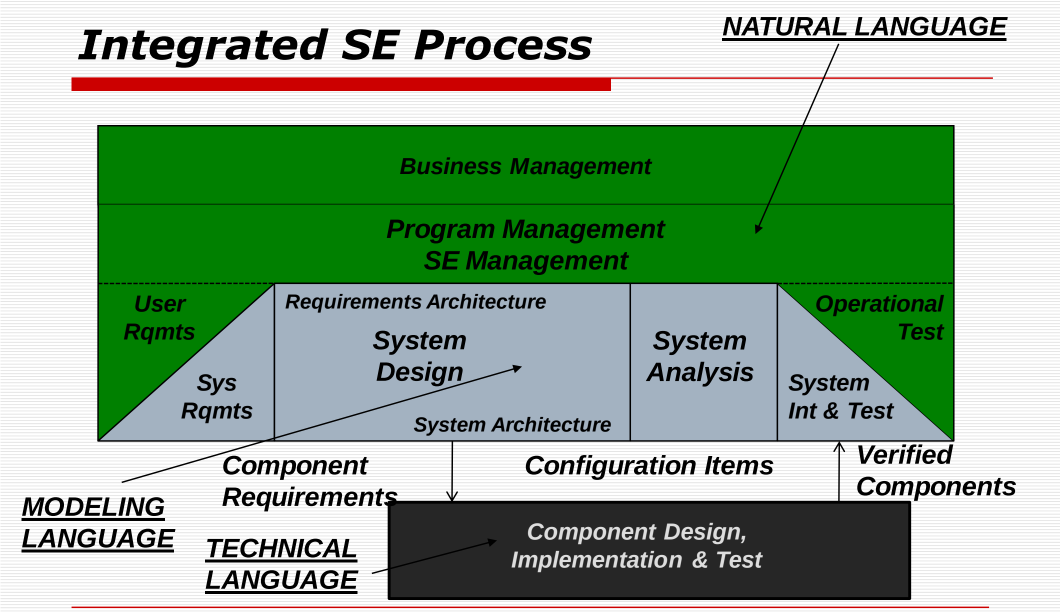


Figure 19: (Quayle 2009)

It is important to note that MBSE is not process dependent. It simply incorporates system modeling into the overall systems engineering effort and produces a system model as one of the primary artifacts. MBSE does not replace current process standards but serves to enhance the systems engineering process through the use of a centralized model repository.

2.7 MBSE Methodologies and Frameworks

A method is defined as "a set of related activities, techniques, and conventions that implement one or more processes and is generally supported by a set of tools." (Friedenthal 2008) As stated earlier, systems engineering standards have evolved over the years and now include various modeling standards and architecture frameworks. The following sections provide a summary review of these modeling standards, methods, and frameworks.

2.7.1 UML/SysML

UML is a software visual modeling language standard managed by the Object Management Group (OMG); an open membership, not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications. OMG in collaboration with The International Council on Systems Engineering (INCOSE) created an extension of UML, called the System Modeling Language (SysML) that incorporates additional modeling diagrams to model complex systems that include hardware, software, data, personnel and procedures. A Venn diagram depicting the relationship between UML and SysML is shown in Figure 20. The development of SysML has worked to improve the acceptance of system modeling across all systems engineering, not only software systems.

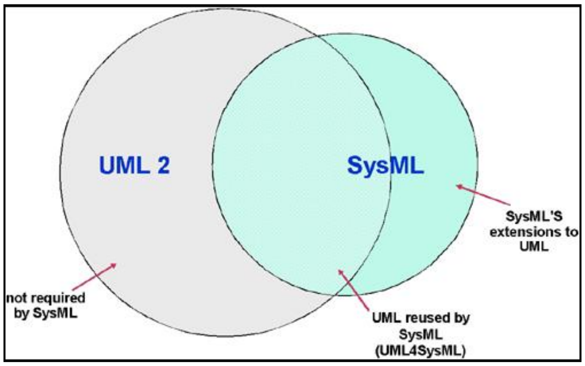


Figure 20 (OMG)

SysML is simply a graphical modeling language standard, and is therefore tool and methodology independent. It provides a means to capture the system modeling information without imposing a specific method. SysML is intended to help specify and architect systems in an unambiguous way that can be clearly communicated to all stakeholders. SysML includes nine diagrams as shown in the SysML diagram taxonomy in Figure 21.

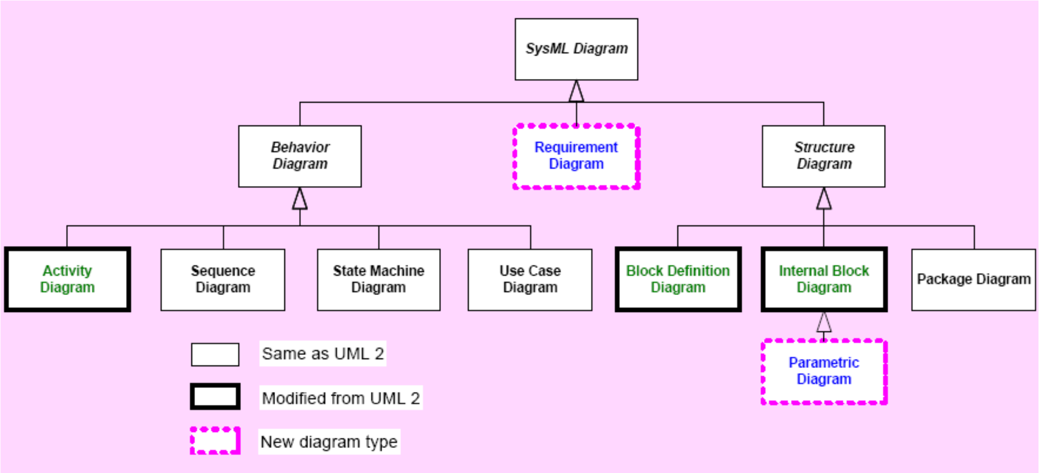


Figure 21: (OMG)

2.7.2 Vitech CORE

Vitech Corporation is the provider of the CORE product suite that combines modeling language and software tool, and the Vitech MBSE methodology. Where SysML is simply a modeling language, Vitech CORE combines modeling language, software tool, and methodology in one. The recent release of CORE 6.0 now provides SysML support by incorporating three of the most utilized SysML diagrams including the activity diagram, sequence diagram, and requirements diagram. CORE is built around a central integrated design repository that is linked to four primary concurrent system engineering activities as shown in Figure 22 , which are then associated to “domains” as shown in Figure 23.

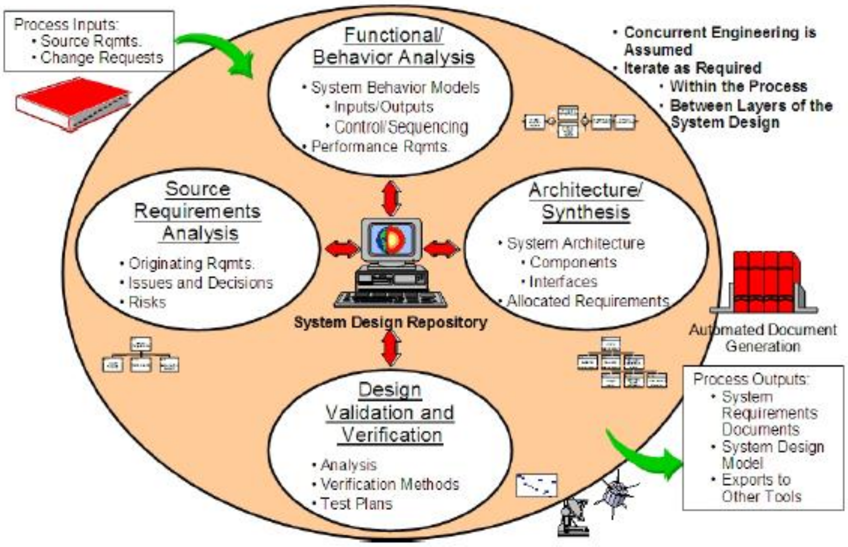


Figure 22: (Vitech Corporation 2009)

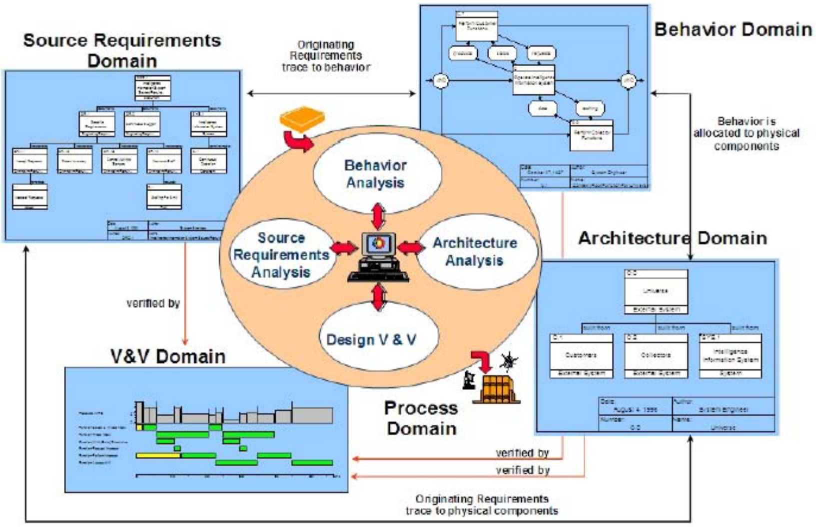


Figure 23: (Vitech Corporation 2009)

2.7.3 OPM

Object Process Methodology (OPM) is a holistic systems paradigm that combines system structure and behavior in a single integrated graphic and natural language model. OPM represents the system in the form of objects, processes, and states. Processes can affect, generate, or consume objects. A state characterizes an object's condition which can be changed by processes. This methodology is currently taught at MIT, Technion, Israel Institute of Technology, and the University of Rochester. OPM is not as widely known as others such as SysML, but offers the advantage of a single graphic with various levels of abstraction. OPM is enhanced though an automatic translation of the model into an Object-Process Language (OPL) script. OPL is essentially the model description in natural English. System complexity is managed through graphical scaling including process zooming, unfolding and folding objects, and expressing or suppressing states. OPM has evolved in recent years from an analysis method into a systems engineering method, encompassing the entire lifecycle of the system. OPM has been used in a number of large-scale projects in the U.S., Germany, and Israel and is currently being experimented with at Ford and NASA.

2.7.4 Department of Defense Architecture Framework (DoDAF)

The Department of Defense Architecture Framework (DoDAF) provides a foundational framework for developing and characterizing architecture descriptions. “The Department of Defense Architecture Framework (DoDAF), Version 2.0 is the overarching, comprehensive framework and conceptual model enabling the development of architectures to facilitate the ability of Department of Defense (DoD) managers at all levels to make key decisions more effectively through organized information sharing across the Department, Joint Capability Areas (JCAs), Mission, Component, and Program boundaries.” (DoD 2009) DoDAF incorporates three views: Operational View (OV), Systems and Services View (SV), and Technical Standards View (TV). These views are depicted in Figure 24 and provide the basis for deriving measure of interoperability or performance, and for measuring the impact of the values of these metrics on operational mission and task effectiveness.

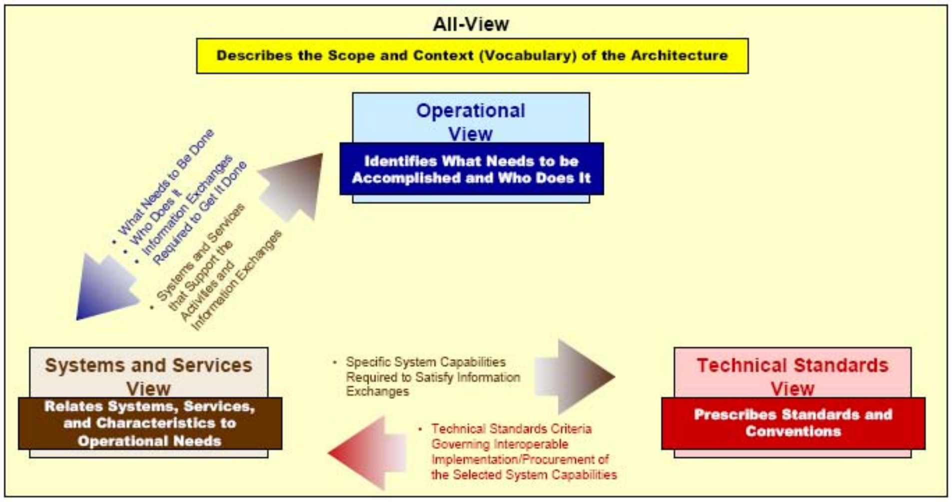


Figure 24: (DoD 2009)

3.0 Case Studies

One of the best ways to validate a process or foresee the advantages is by researching industry best practices and success stories. The Navy has often looked to industry to verify that they are upholding and/or surpassing standard practices. Most revolutionary ideas within the naval ship design arena have been taken from industry in one form or another including lean engineering, set-based design, and open systems architecture. Model-Based Systems Engineering is not an exception as it has been used extensively in the software development community (termed Model Driven Architecture or MDA) and has started to trickle into the systems engineering community since the advent of SysML. The case studies below will serve as an "industry review" of MBSE, highlighting claimed benefits and lessons learned.

3.1 Software Engineering Cases

Model-Driven Architecture (MDA) is a model based approach for software architecture and design developed by OMG[1] which includes a set of key principles intended to improve software interoperability, reusability, portability, maintainability, and reliability. The approach has had resounding success in the software community and many experts suggest that such an approach applied to systems engineering could produce similar results. Cloutier hypothesizes that the application of MDA may provide 10-20% efficiency increase in the Systems Engineering effort over using what have become the SE tools of choice (PowerPoint, Word, Excel). To put this into perspective, for a $20M engineering project with SE representing 10% of the overall effort, the improved efficiency might range from $200k - $400k. The following case studies demonstrate the value of MDA to the software community.

Carter Ground Fueling Ltd.[2]

Carter Ground Fueling Ltd. is the Airline Industry's leading supplier of fuel delivery software and hardware. They recently brought to market their leading edge AvR2057 in-cab refueling system in record time due in large part to the adoption of a full Model-Driven Development environment. The benefits of using a model-driven approach experienced by Carter Ground Fueling include:

---

[1] OMG (Object Management Group) has been an international, open membership, not-for-profit computer industry consortium since 1989. OMG's current standards include: UML (Unified Modeling Language), SysML (Systems Modeling Language), MOF (Meta Object Facility), and MDA (Model Driven Architecture).

[2] http://www.omg.org/mda/mda_files/CarterGround2004.pdf

- Flexibility-functional changes can be made quickly and cost effectively.
- Communication -“It would have been impossible to achieve and maintain such a strong design and architecture without the ability to express and share these visually through models” said Darren Hale, project manager.
- Risk Reduction - Reduction in program risk by early and often verification

DaimlerChrysler TSS[3]

Daimler Chrysler TSS is a wholly owned subsidiary of Daimler Chrysler AG, founded in 1998. They used MDA to develop their Electronic Production Planning (ePeP) system with the goal of achieving a 10% increase in productivity. The project presented many challenges including:

- Very complex business and process logic
- Integration into existing, complex system landscape
- Required 10% improvement in productivity to achieve cost saving targets
- Multi-site development in Germany and Malaysia

Despite the many challenges, they were able to exceed their original goal and increased development productivity by 15%. They cited a key benefit of the MDA approach was ensuring architectural consistency of the complex ePeP system. Other benefits of applying MDA realized by Daimler Chrysler TSS include:

- Improves project communication and coordination, reducing friction losses normally caused by multi-site development
- Increases project transparency allowing for early identification of problems and issues
- Streamlines the development process with fewer misinterpretations of requirements
- Reduces architectural complexity
- Automatically ensures architectural consistency across all application tiers and functional layers

---

[3] http://www.omg.org/mda/mda_files/SuccesStory_DC_TSS_MDO_English.pdf

3.2 Systems Engineering Cases

Systems Engineers have historically been prolific producers of documentation, whether it be the system specification, sub-system specification or a trade-off study report. However, the role of the systems engineer is gradually changing as companies begin to adopt MBSE methodologies. The adoption and diffusion rate of MBSE in industry has been slow, but as standards and processes improve the tides will surely change. Systems engineers will abandon the long-standing document centric approach in place of MBSE and will be called "paper pushers" no more. The case studies below reflect the current state of MBSE and describe the benefits and/or deficiencies with the existing tools.

Toyota Motor Corporation[4]

Toyota Motor Corporation is one of the world's largest automobile companies and is known for its innovative use of technology. Toyota was one of the earliest companies to embrace model-based design in the hopes to improve time-to-market, quality, and reliability, while reducing cost. At the end of 2007, Toyota entered a partnership with Maplesoft, the leading provider of high-performance software tools for engineering, science, and mathematics, in order to move them to a new model-based development process. "Model-Based Development will set new industry standards for the use of software tools and models in automotive systems development," said Dr. Akira Ohata, Project General Manager of Toyota Motor Corporation. Toyota has not to this date published any reports on the success or failure of adopting a model-based approach.

NASA[5]

NASA (National Aeronautics and Space Administration) currently uses MBSE to streamline requirements development on various projects. NASA presented the MBSE requirements development process for the Altair project during the Seventh Annual NASA Project Management Challenge in February 2010. Altair is the lunar lander spacecraft component of NASA's Constellation fleet. NASA envisions Altair lunar lander to transfer up to four astronauts from the Orion crew capsule to the lunar surface, then to serve as a life support

---

[4] http://www.maplesoft.com/company/publications/articles/view.aspx?SID=5476

[5] http://pmchallenge.gsfc.nasa.gov/docs/2010/Presentations/Robert.Bayt.pdf

base for surface exploration missions lasting up to one week, and then finally returning the astronauts back to the Orion spacecraft. MBSE has allowed NASA to communicate to suppliers what they want from Altair through a central database (or model) including operational concepts, functional architecture and design constraints. The use of a central database allows for system attributes to be tracked and linked directly to requirements and provides the capability to generate products as reports from a common set of data. NASA's use of MBSE improves quality and timeliness of the requirements and reduces the resources required to develop and maintain them.

4.0 Developing the Architecture

The potential benefits of MBSE have been realized by the overarching community of systems engineers, but that is not to say models should be used in every situation. "*Just because you can, doesn't mean you should.*" There must be a clear purpose for modeling a system and it must be defined in terms of the expected results before the modeling effort begins. This will help to define the scope of the model in terms of breadth, depth, and fidelity. A design team could start by modeling a ship concept, and without proper scope could end up modeling the entire Navy. The purpose and scope provide the basis for establishing realistic expectations of the modeling effort. There are several standard purposes for modeling systems and they include:

1. To characterize an existing system
2. To analyze or evaluate a system
3. To specify and design a new system
4. To train users on operation or maintenance of a system

In early stage ship design, the purpose of modeling would be to design a new system, specifically to represent the ship concept architecturally.

MBSE methods will be used in the subsequent paragraphs to investigate how the propulsion system of a naval ship could be modeled and architected in CORE. This architecting process will explore the advantages and disadvantages of using MBSE in ship design and acquisition.

4.1 Vitech CORE Overview

Vitech CORE was introduced earlier and is the tool used for developing the propulsion system architecture in this thesis. A comparison of MBSE tools and methodologies extends beyond the scope of this thesis, therefore CORE was chosen simply because it was the easiest MBSE tool to access. At the heart of the CORE systems engineering environment is a central design repository that maintains every aspect of the system design. The centralized repository or database allows for various representations of the data in order to facilitate communication amongst the various stakeholders. The design repository stores and maintains all the system attributes in an integrated and consistent manner and allows for documents to be produced on an as-needed basis. Additionally, when a design change is made within CORE, all the subsequent views and documentation generated will reflect the change. In this way, no one is concerned about whether they have the latest documentation as it can be easily generated from the repository. The repository contains the following artifacts:

- Requirements
- Functional descriptions and graphical models
- Behavioral executable models
- Performance characteristics and constraints
- Operational architectures
- Physical architectures
- Interfaces, data flows and rates
- Responsible organizations
- Technical guidance

CORE has also extended the systems engineering environment to integrate with DODAF semantics. The operational architecture domain was developed in addition to the system architecture domain. Figure 25 shows the relationships within and between the operational architecture and the system architecture. However, only the system architecture domain is used in this thesis to develop the architecture of a ship's propulsion system.

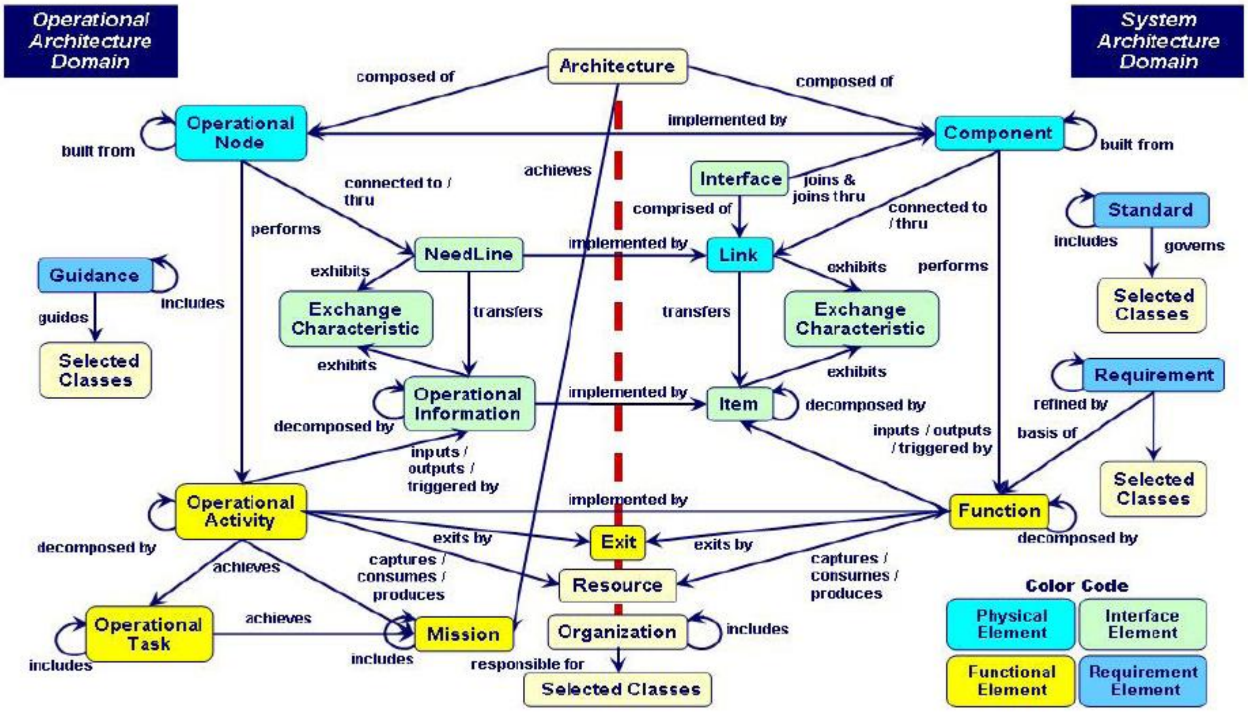


Figure 25: (Vitech Corporation 2009)

4.2 Requirements Development

One of the challenges facing the ship design team is ensuring the ship meets operational objectives and goals. In order to ensure real world applicability, mission-based operational needs should drive the system definition, architecture and design. There are many organizations involved in the ship design process and it is easy to lose sight of the operational end-state. If operational capability is used upfront to drive requirements, then the design team is able to maintain linkage between operational and technical requirements throughout development. "Systems engineering must have a mission focus to ensure that each organization contributes to a design that meets operational needs and objectives." (Adams and Kott 2008)

Proper development of the architecture requires a comprehensive modeling technique based on well-specified, capability-based requirements. One method, described in Adams and Kott (2008), proposes to use the Required Operational Capabilities and the Projected Operating Environment (ROC/POE) upfront to define requirements. An alternate method uses the Universal Naval Task List (UNTL) as a source for deriving customer requirements and then allocates those requirements to mission system packages. (Doerry 2006) Whichever way requirements are developed, they are almost always captured in some sort of requirements document or capabilities document. In order to maintain real-world applicability, the propulsion system developed below will start with the actual Performance Specification Document for the Auxiliary Dry Cargo Ship, T-ADC(X). (NAVSEA 1998) The document was added in CORE as the system reference document. The first step in the architecting process is to define the need and system concept within the database. In designing the propulsion system, the designer must understand that it is a system-of-systems (SoS). The propulsion system is in fact a subsystem of a larger system, the ship itself. The ship itself is also a subsystem of a larger system, the Navy fleet. The Navy fleet is a subsystem of a larger system, the joint environment including the Army, Navy, Air Force, Marines, and Allied nation components. In this design, the focus is solely on the propulsion system of the Auxiliary Dry Cargo Ship, T-ADC(X). The system was created in CORE by defining a component called Sys.1 Propulsion System.

To capture the source or originating requirements, the propulsion system requirements were extracted from the source document and added to the CORE database. It is also possible to augment requirements with external files. Two external files in the form of tables were added to the database to augment the tagged requirements. Since all of these top-level requirements came from the source document, they are also linked to the document within CORE. Verification requirements explicated in the source document were also added to the database. Figure 26 shows how the Document links to Requirements, how the Document links to the System, and how a Requirement can be augmented by an External File.

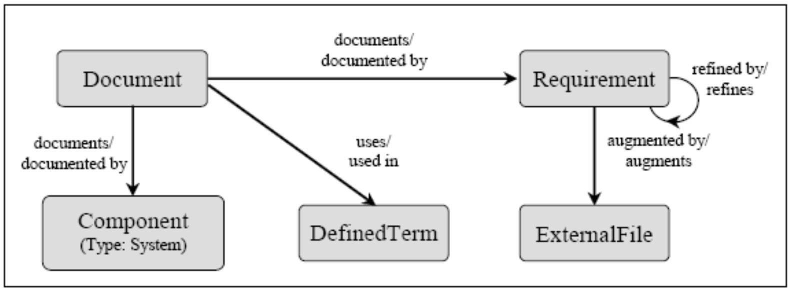


Figure 26: Source Requirements (Vitech Corporation 2009)

In order to define the system and its boundary, the top-level components and top-level root functions must be identified. This is called the system context. The context is made up of the system, the external components, and their respective interfaces as shown in Figure 27.

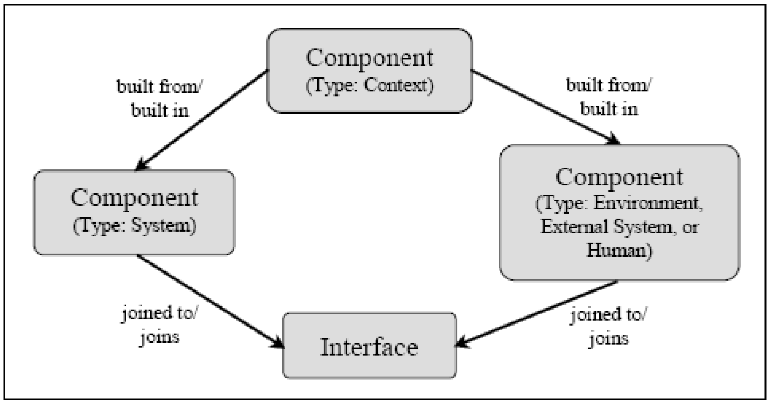


Figure 27: (Vitech Corporation 2009)

The external components created include the atmosphere, fuel, operator, ship hull, and water. The propulsion system context is shown in Figure 28. This context was placed in a new folder in the database in order to separate it from the evolving component hierarchy.

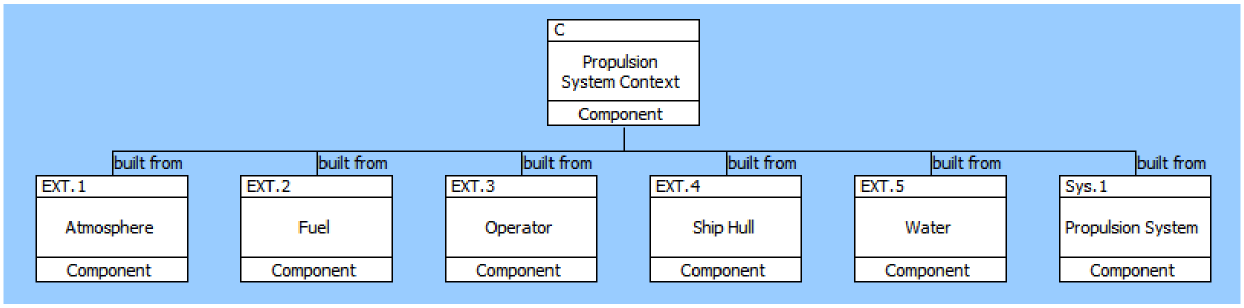


Figure 28: System Context

4.3 Requirements Analysis

"Requirements Analysis encompasses the definition and refinement of system, subsystem, and lower-level functional and performance requirements and interfaces to facilitate the Architecture Design process." (DAU 2010) It is extremely important that the system has measurable and verifiable requirements. The originating requirements need to be parsed into single, testable requirements statements. In the database, the originating requirements were refined and parsed into leaf-level requirements. These single requirement statements are noted to be "derived" requirements with linkages back to their origins. This is a long and often iterative process, so it is important to maintain all linkages back to the originating requirements, which CORE does automatically. Additionally, certain requirements can generate issues or problems. They could be poorly stated requirements or could be conflicting with other requirements. CORE allows the designer to capture both the issue and the decision that resolved the issue. In this way the user can keep track of decisions, alternatives, and rationale. In the database, four issues were "generated by" requirements. Three were closed in the database and documented with rationale and one was left as an open issue to see the impact on the overall model. The open issue is related to what speeds of advance the "Stopping" requirement must be met, as shown in Figure 29.

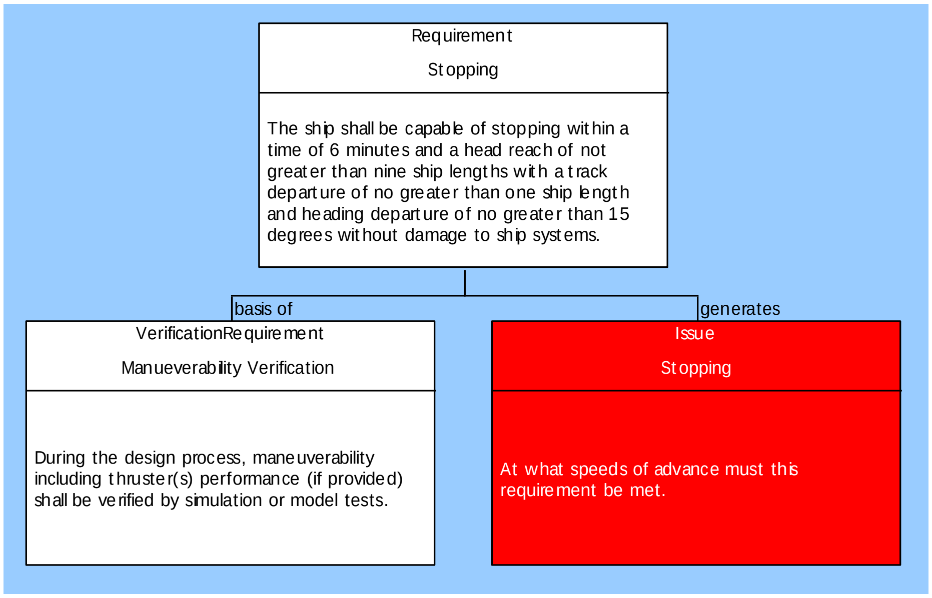


Figure 29: Requirements Issues

Requirements can also cause a certain amount of risk that must be captured within the database. Requirements risk is real and is not always documented in a systems design as it should be. In the database one requirement risk related to the sustained speed requirement was created and assigned to Naval Sea Systems Command (NAVSEA) as shown in Figure 30.

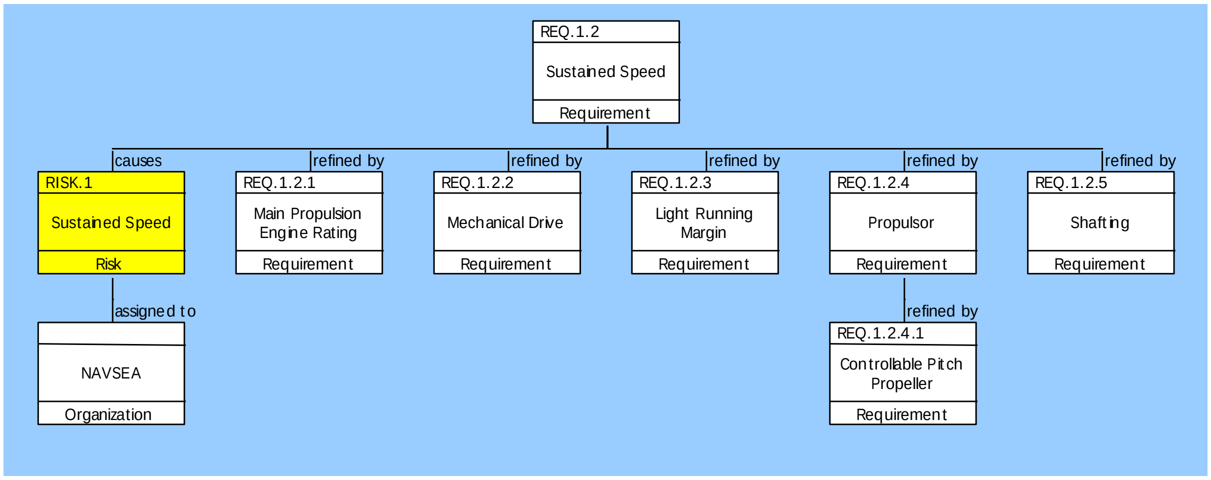


Figure 30: REQ.1.2 Sustained Speed

Requirements can be classified in the databases by the type of requirement: performance, functional, constraint, or verification. Non-functional requirements (such as availability and reliability) are captured as constraints. The system level constraint requirements were linked to the system component, Sys.1 Propulsion System, in the CORE database.

4.4 Functional & Physical Architecture

"A functional architecture expresses the detailed functional, interface, and temporal aspects of the system that are essential to gain sufficient insight and to communicate unambiguously the behavior of the system in its intended operational environment. The development of a functional architecture and definition of system functions should not be performed in isolation; it should be developed incrementally with stakeholder requirements and the physical architecture to ensure that the appropriate functions and interfaces are identified." (DAU 2010) There are many essential benefits to developing a functional architecture, specifically it provides[6]:

- A definition of the system functional baseline,
- A measure of the system's ability to fulfill its functional objectives as defined by the system functional requirements,
- A measure of the system's ability to fulfill its performance objectives as defined by the system performance requirements.
- The system's ability to operate within resource constraints,

---

[6] Functional architecture benefits stated here come from (DAU, Defense Acquisition Guidebook 2010)

- Costs, economic and otherwise, of implementing and operating the system over its entire life cycle, and
- Side effects, both positive and adverse, associated with architectural options

The functional architecture and physical architecture of the propulsion system were developed concurrently. The physical architecture is created as system functions are allocated to their respective components. The first step was to allocate a root function to the total system. This root function, Perform Propulsion System Functions, encompasses all functions of the system and was allocated to Sys.1 Propulsion System. Another root level function was created, Perform Operator Functions, and was appropriately allocated to the Operator. An Enhanced Functional Flow Block Diagram (EFFBD) was used to insert a parallel structure in the functional context because propulsion system functions and operator functions are performed in parallel. An N2 Diagram was then used to show that the operator provides Desired Speed (item) and Machinery Request (item) as an input to the propulsion system. Desired Speed and Machinery Request are entered into the CORE database as input "Items". The EFFBD is shown in Figure 31.

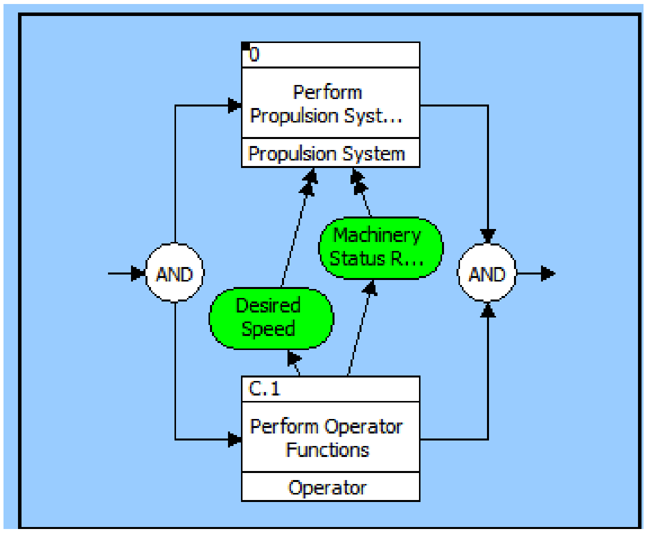


Figure 31: EFFBD

Logically, the functional decomposition follows by decomposing the top level function, Perform Propulsion System Functions, into lower level functions. These lower level functions should be traced back to requirements ("based on"). Functions are decomposed until they can be uniquely allocated to the next level of Component. This functional hierarchy and allocation provides the organization of the performance requirements in a specification for a component. Every function that is allocated to a component should have associated performance requirements linked to it. The relationships between functions, components, and requirements defined in CORE are shown in Figure 32.

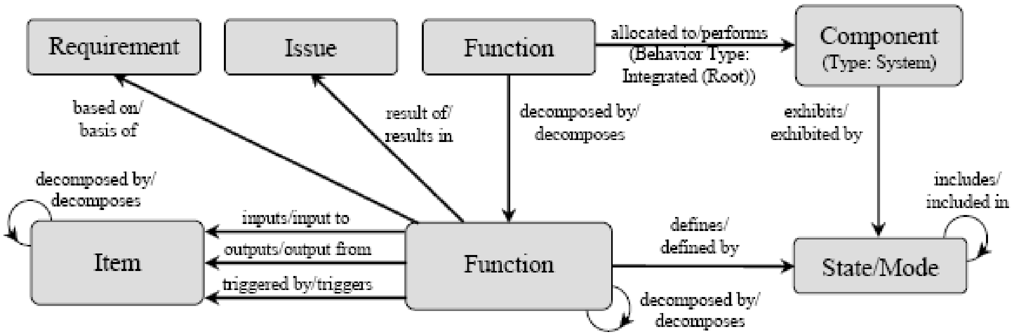


Figure 32: (Vitech Corporation 2009)

The EFFBD was used again to add constructs and functions under Perform Propulsion System Functions. The constructs are used to group lower-level functions into categories which include propulsion, control, support, and survivability as shown in Figure 33 below.

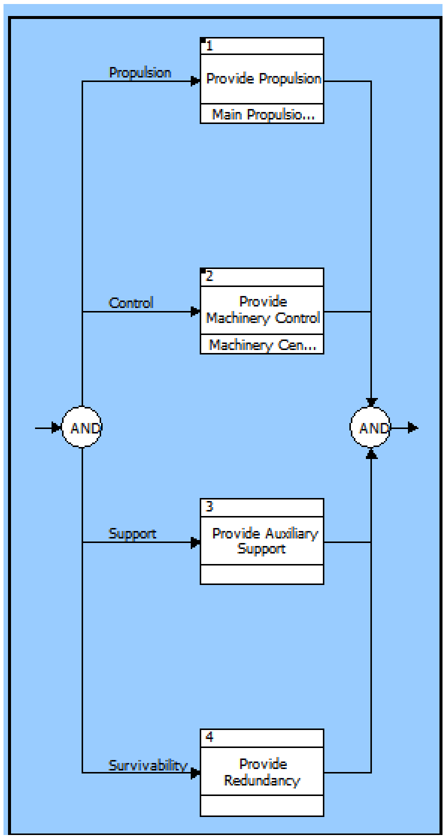


Figure 33: EFFBD Functional Decomposition

Now that the functional constructs are defined, the following functions can be decomposed: Provide Propulsion, Provide Machinery Control, Provide Auxiliary Support, and Provide Redundancy.

Provide Propulsion

An "OR" structure was selected in the EFFBD to show two possible architectures of the Propulsion System: Mechanical or IPS. The propulsion top-level functions were added in sequential order via the EFFBD for both the mechanical drive and electric drive architectures.

- Mechanical Functions: Generate Mechanical Energy, Transfer Mechanical Energy, Generate Thrust, Transfer Thrust
- IPS Functions: Generate Mechanical Energy, Generate Electrical Power, Distribute Electrical Power, Convert to Mechanical Energy, Transfer Mechanical Energy, Generate Thrust, Transfer Thrust

The above top-level functions were then allocated to the components that must perform them (prime mover, transmission, propulsor, motor, generator, power distribution module). This allocation and decomposition is shown in the EFFBD in Figure 34.

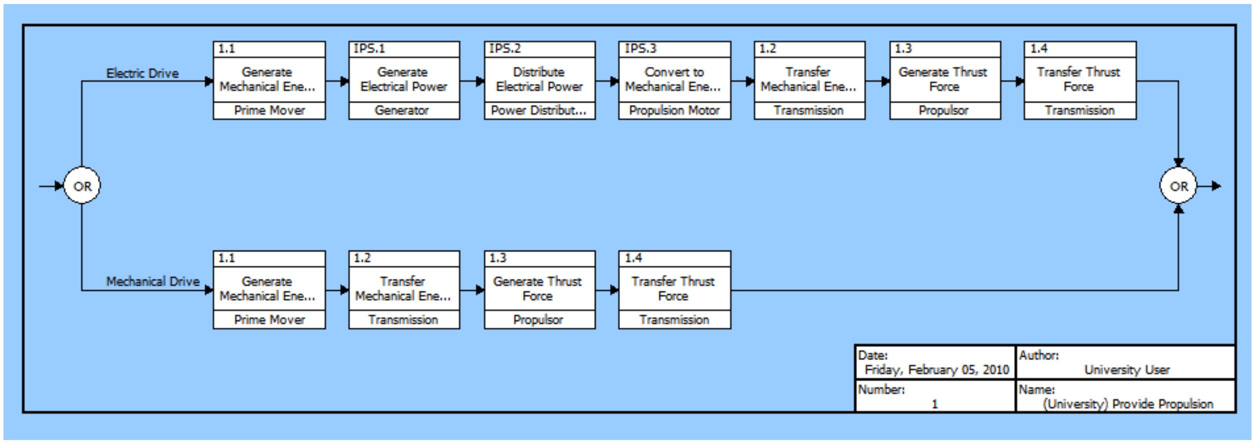


Figure 34: EFFBD Provide Propulsion

The next step is to define the flows between the functions, i.e. the inputs, outputs and triggers, as shown in Figure 35.

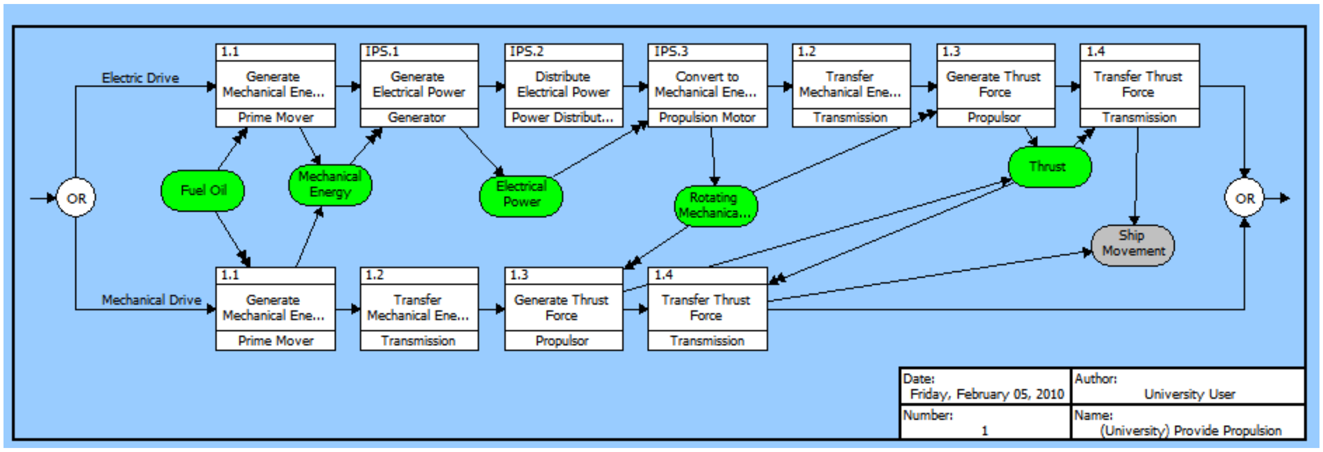


Figure 35: Provide Propulsion functional flow

These flows can also be represented in an N2 Diagram (Figure 36) which displays the flows in a neater fashion.

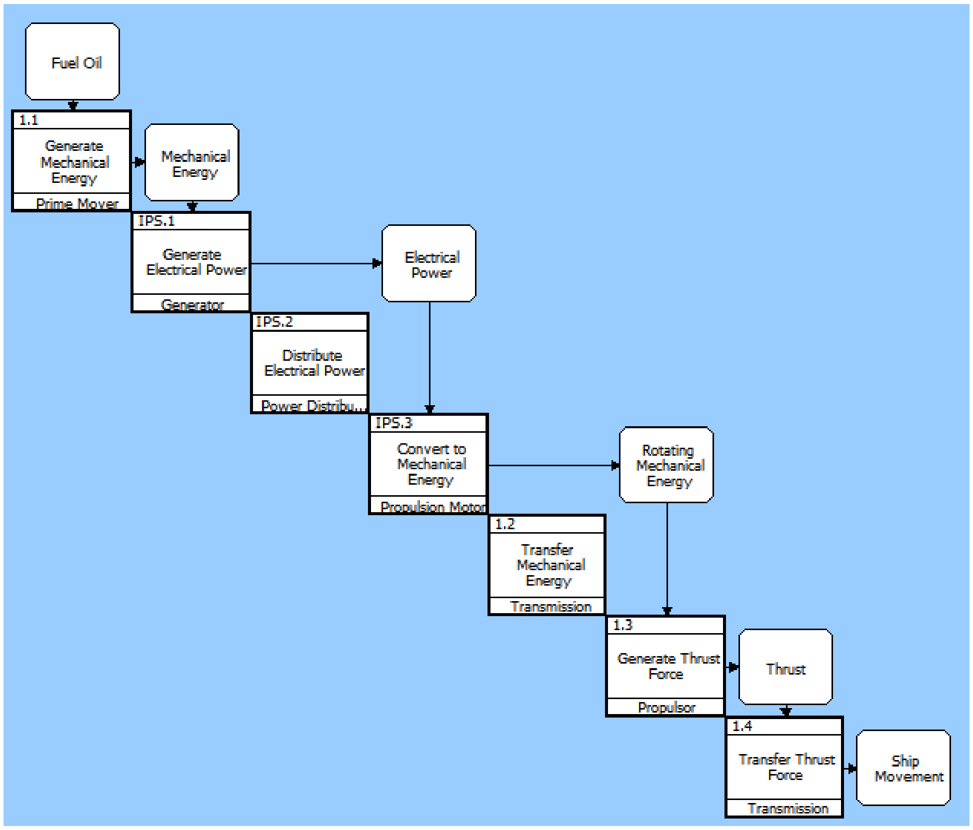


Figure 36: N2 Diagram Provide Propulsion

The functions must be based on performance requirements to complete traceability. Since functions may be aggregated to enhance understanding, not every function will have performance requirements, but every function that is allocated to a component should have associated performance requirements. The previous performance requirements must be decomposed in order to allocate them to the respective lower-level functions. This decomposition was performed in order ensure every function allocated to a component is traceable back to a requirement.

Provide Machinery Control

The next top-level function of the propulsion system is "Provide Machinery Control". To decompose control functions, the control requirements of the ship in regards to the propulsion system must be revisited. Provide Machinery Control is decomposed into lower-level functions: Provide Local Control, Provide Remote Speed Control, Collect Propulsion System Data, Display Propulsion System Data, Provide Connectivity, and Perform Data Logging. The decomposition is represented by Figure 37.

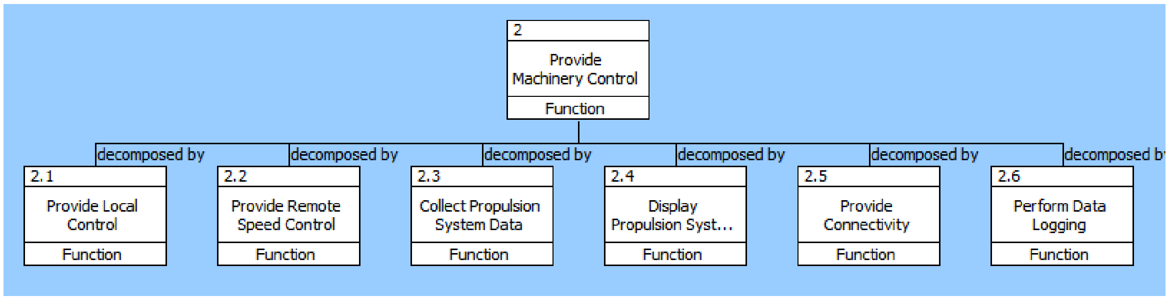


Figure 37: Provide Machinery Control Functional Decomposition

An EFFBD was again used to create the functional flows. Speed control cannot be simultaneously performed at the bridge and EOS, therefore an "OR" construct was created. "Iteration" was added for collecting and displaying propulsion system data, based on the machinery data refresh rate. Iteration was also added to Perform Data Logging, which shall be performed every 4 hours and upon operator request based on requirements. These functions were then allocated to their respective system components. The EFFBD is shown in Figure 38.

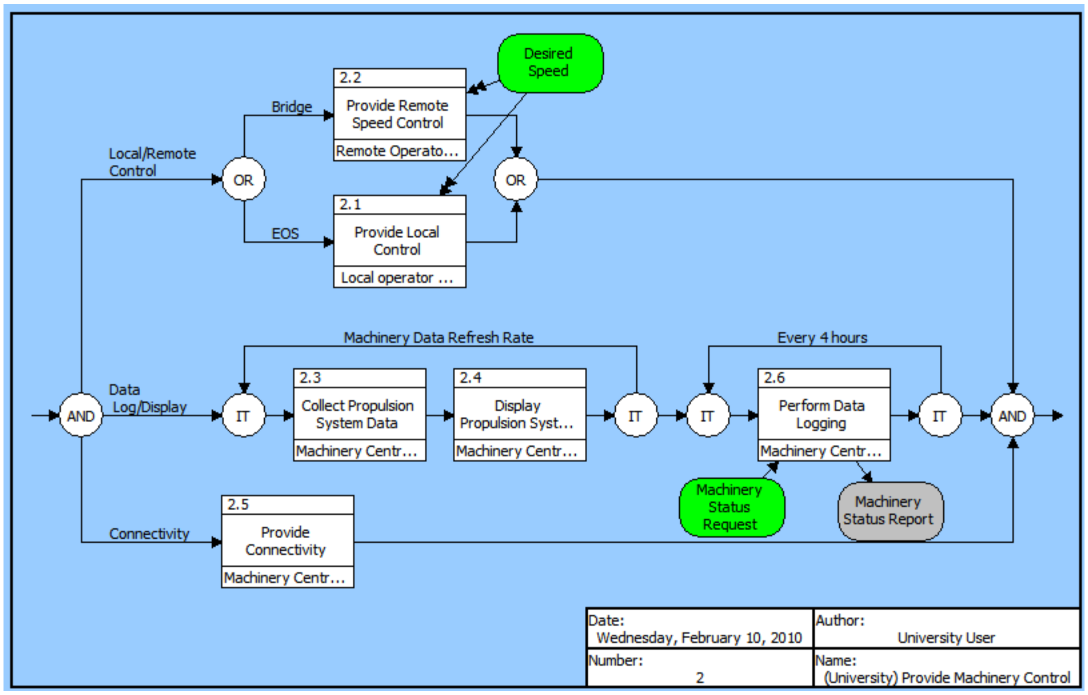


Figure 38: EFFBD Provide Machinery Control

Again, each of the decomposed control functions must be based on a requirement (if they are allocated to a component) in order to complete traceability. In CORE, the relationship "based on" is used to attribute performance requirements to all leaf-level control functions.

Provide Auxiliary Support

The next top-level function is to "Provide Auxiliary Support", which can be decomposed into the following sub-functions: Provide Start Air, Provide Fuel, Provide Lubrication, Provide Cooling Water, and Provide Combustion Air.

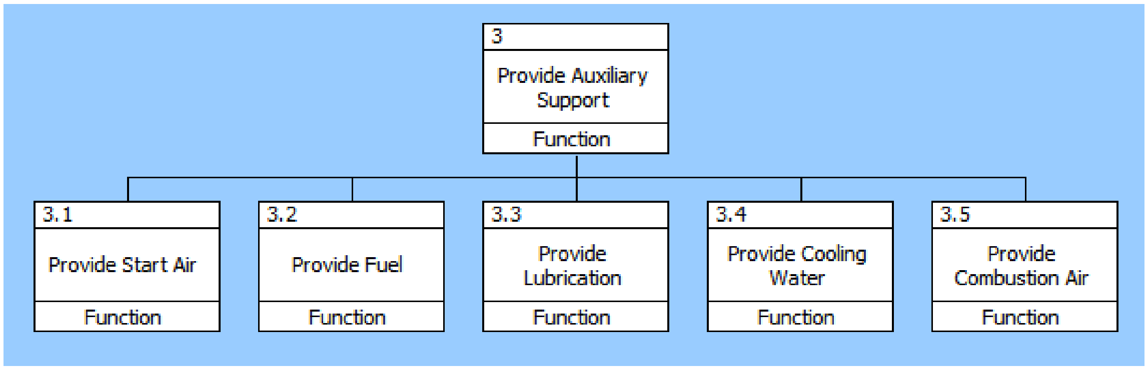


Figure 39: Provide Auxiliary Support Functional Decomposition

All the sub-functions were allocated to their respective components within the CORE database and linked back to requirements to complete traceability. Again an EFFBD was used to create the auxiliary support function parallel constructs as shown in Figure 40.

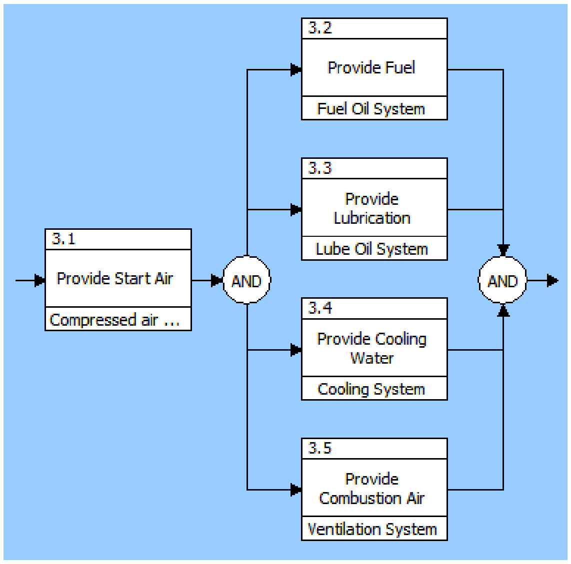


Figure 40: EFFBD Provide Auxiliary Support

Provide Redundancy

The function, Provide Redundancy, was decomposed into sub-level functions in the same way as Provide Propulsion, Provide Machinery Control, and Provide Auxiliary Support. The lower level functions were allocated to components to develop the physical architecture and were then linked back to performance requirements.

4.5 Capture Functional and Performance Issues and Risks

While developing the system's functional hierarchy and deriving the associated performance requirements, additional issues and risks may be identified. The issue could lead to a design decision that results in an additional requirement or could result in an additional function or functions. If this is a major design decision, it should be augmented with an issue to capture the details of the decision. As an example, the Fuel Efficiency Requirement generated an Issue that led to a prime mover trade study. The results and rationale of the trade study were documented in the Issue, and ultimately resulted in an additional refining requirement. The refining requirement is captured as a design decision in the CORE database by changing the origin to "design decision". This Trade Study Issue was assigned to NAVSEA and documented by the

Alternate Propulsion Study Report (March 2007). The specifics of conducting a trade-study and capturing it in the database will be discussed later. The relationships described above related to the Fuel Efficiency Trade Study Issue are shown in Figure 41 and Figure 42.

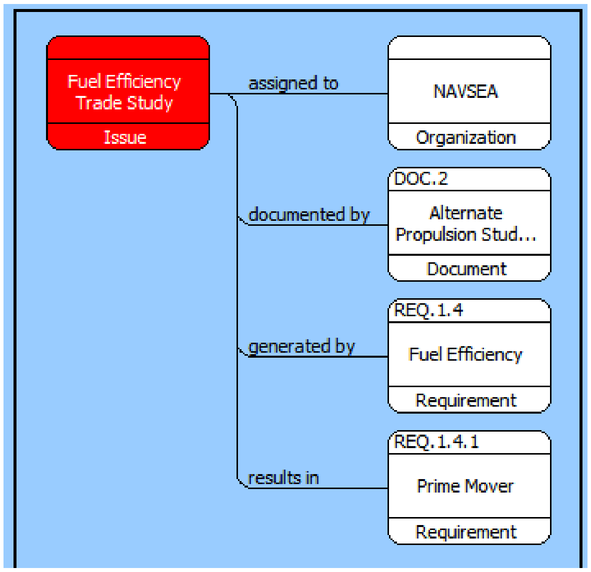


Figure 41: Fuel Efficiency Trade Study

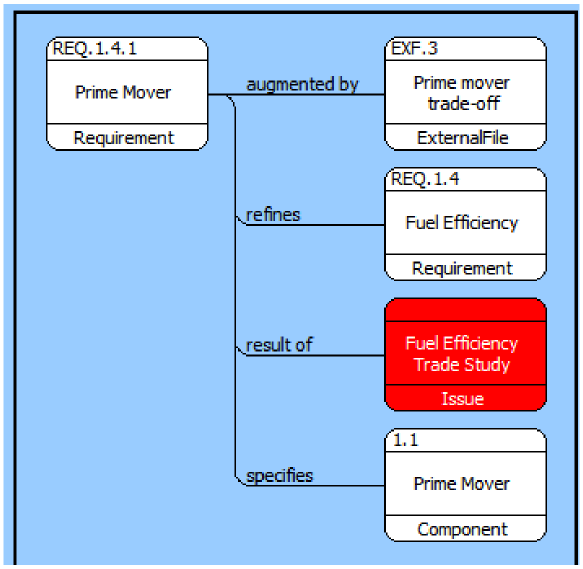
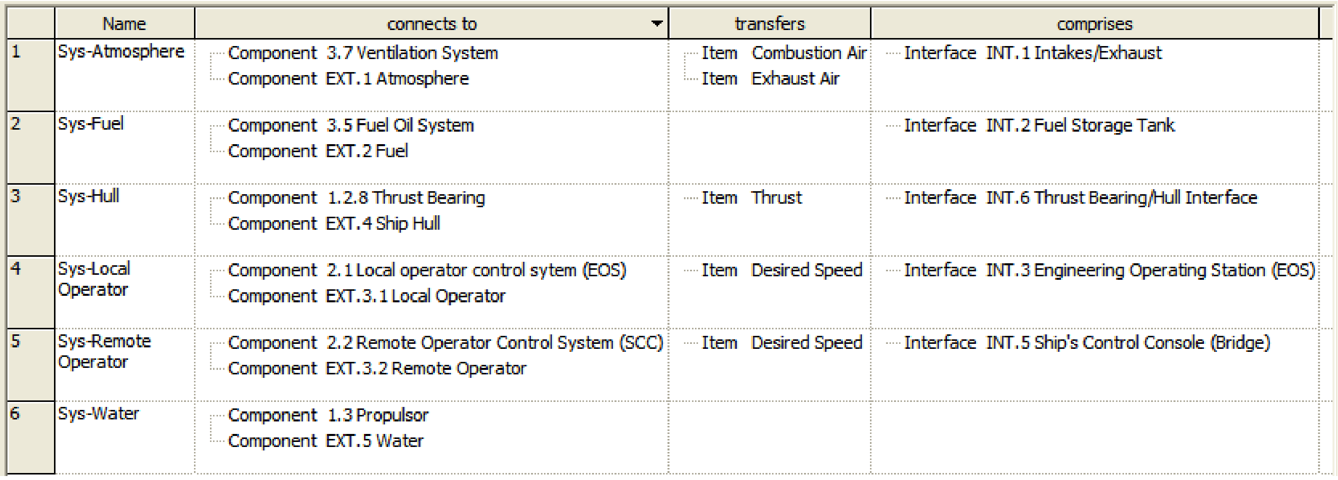


Figure 42: Additional Requirement "Design Decision"

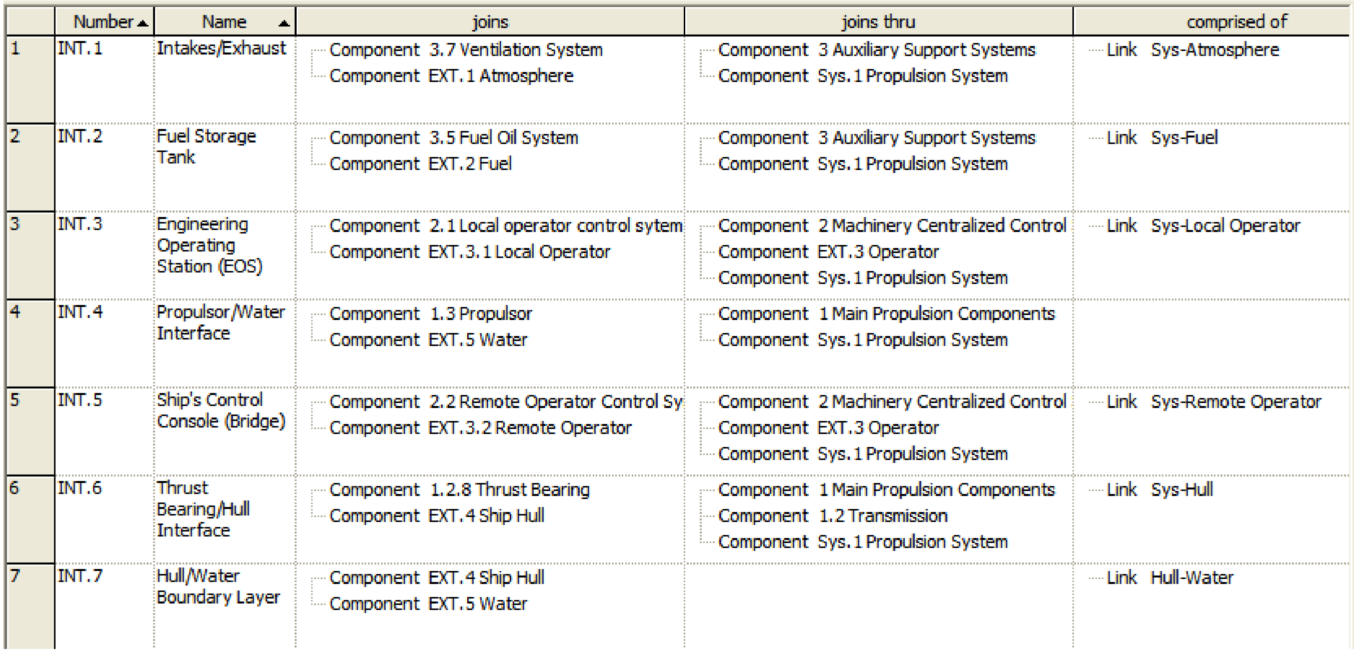
4.6 Refine External Interfaces & Links

An external interface element identifies the fact that the system communicates in some manner with an external component. Details of the interface are captured in Link element definitions. Link elements represent the actual physical connections in CORE. The external interfaces were defined earlier, but as the system component hierarchy evolves, the terminus point for those interfaces are changed (i.e. subordinate components of the Sys.1 Propulsion System are now the terminus points of the interfaces). The external components are “joined to” the lower level components and they are “joined thru” the system. The links and interfaces established within CORE are displayed in Table 3 and Table 4 respectively.



	Name	connects to	transfers	comprises
1	Sys-Atmosphere	Component 3.7 Ventilation System Component EXT.1 Atmosphere	Item Combustion Air Item Exhaust Air	Interface INT.1 Intakes/Exhaust
2	Sys-Fuel	Component 3.5 Fuel Oil System Component EXT.2 Fuel		Interface INT.2 Fuel Storage Tank
3	Sys-Hull	Component 1.2.8 Thrust Bearing Component EXT.4 Ship Hull	Item Thrust	Interface INT.6 Thrust Bearing/Hull Interface
4	Sys-Local Operator	Component 2.1 Local operator control sytem (EOS) Component EXT.3.1 Local Operator	Item Desired Speed	Interface INT.3 Engineering Operating Station (EOS)
5	Sys-Remote Operator	Component 2.2 Remote Operator Control System (SCC) Component EXT.3.2 Remote Operator	Item Desired Speed	Interface INT.5 Ship's Control Console (Bridge)
6	Sys-Water	Component 1.3 Propulsor Component EXT.5 Water		

Table 3: Component Links



	Number ▲	Name ▲	joins	joins thru	comprised of
1	INT.1	Intakes/Exhaust	Component 3.7 Ventilation System Component EXT.1 Atmosphere	Component 3 Auxiliary Support Systems Component Sys.1 Propulsion System	Link Sys-Atmosphere
2	INT.2	Fuel Storage Tank	Component 3.5 Fuel Oil System Component EXT.2 Fuel	Component 3 Auxiliary Support Systems Component Sys.1 Propulsion System	Link Sys-Fuel
3	INT.3	Engineering Operating Station (EOS)	Component 2.1 Local operator control sytem Component EXT.3.1 Local Operator	Component 2 Machinery Centralized Control Component EXT.3 Operator Component Sys.1 Propulsion System	Link Sys-Local Operator
4	INT.4	Propulsor/Water Interface	Component 1.3 Propulsor Component EXT.5 Water	Component 1 Main Propulsion Components Component Sys.1 Propulsion System	
5	INT.5	Ship's Control Console (Bridge)	Component 2.2 Remote Operator Control Sy Component EXT.3.2 Remote Operator	Component 2 Machinery Centralized Control Component EXT.3 Operator Component Sys.1 Propulsion System	Link Sys-Remote Operator
6	INT.6	Thrust Bearing/Hull Interface	Component 1.2.8 Thrust Bearing Component EXT.4 Ship Hull	Component 1 Main Propulsion Components Component 1.2 Transmission Component Sys.1 Propulsion System	Link Sys-Hull
7	INT.7	Hull/Water Boundary Layer	Component EXT.4 Ship Hull Component EXT.5 Water		Link Hull-Water

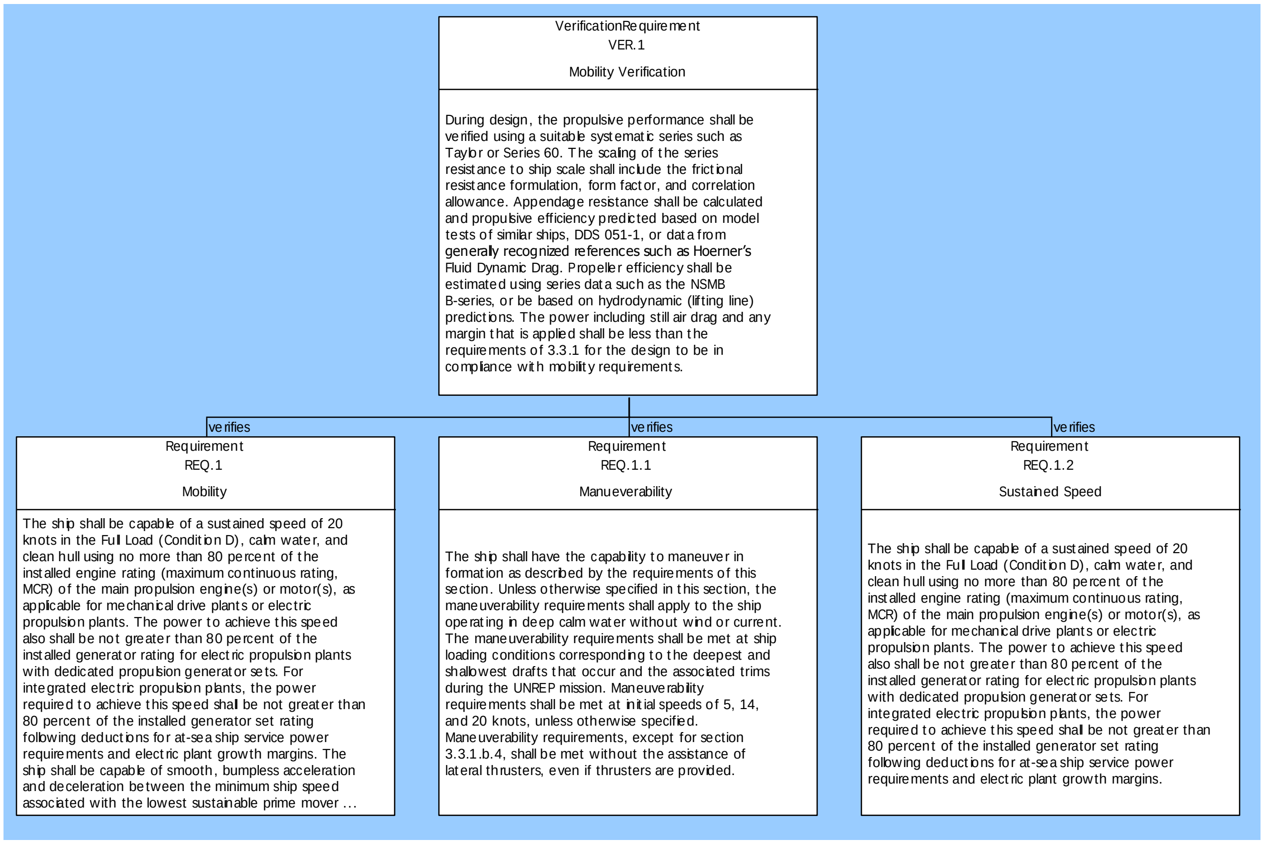
Table 4: Component Interfaces

4.7 Refine Internal Interfaces & Links

Within the system hierarchy, the allocation of Functions to their respective Components establishes the internal interfaces of the system based on the Items that flow between the allocated functions. The internal interfaces are formalized in the database using the Interface and Link element classes as done with the External Interfaces and Links.

4.8 Verification/Validation

Verification requirements are captured in the database and specify how each requirement is to be verified. It is possible for a single "Verification Requirement" to verify multiple requirements as shown below.



During design, the propulsive performance shall be verified using a suitable systematic series such as Taylor or Series 60. The scaling of the series resistance to ship scale shall include the frictional resistance formulation, form factor, and correlation allowance. Appendage resistance shall be calculated and propulsive efficiency predicted based on model tests of similar ships, DDS 051-1, or data from generally recognized references such as Hoerner's Fluid Dynamic Drag. Propeller efficiency shall be estimated using series data such as the NSMB B-series, or be based on hydrodynamic (lifting line) predictions. The power including still air drag and any margin that is applied shall be less than the requirements of 3.3.1 for the design to be in compliance with mobility requirements.

The ship shall be capable of a sustained speed of 20 knots in the Full Load (Condition D), calm water, and clean hull using no more than 80 percent of the installed engine rating (maximum continuous rating, MCR) of the main propulsion engine(s) or motor(s), as applicable for mechanical drive plants or electric propulsion plants. The power to achieve this speed also shall be not greater than 80 percent of the installed generator rating for electric propulsion plants with dedicated propulsion generator sets. For integrated electric propulsion plants, the power required to achieve this speed shall be not greater than 80 percent of the installed generator set rating following deductions for at-sea ship service power requirements and electric plant growth margins. The ship shall be capable of smooth, bumpless acceleration and deceleration between the minimum ship speed associated with the lowest sustainable prime mover ...

The ship shall have the capability to maneuver in formation as described by the requirements of this section. Unless otherwise specified in this section, the maneuverability requirements shall apply to the ship operating in deep calm water without wind or current. The maneuverability requirements shall be met at ship loading conditions corresponding to the deepest and shallowest drafts that occur and the associated trims during the UNREP mission. Maneuverability requirements shall be met at initial speeds of 5, 14, and 20 knots, unless otherwise specified. Maneuverability requirements, except for section 3.3.1.b.4, shall be met without the assistance of lateral thrusters, even if thrusters are provided.

The ship shall be capable of a sustained speed of 20 knots in the Full Load (Condition D), calm water, and clean hull using no more than 80 percent of the installed engine rating (maximum continuous rating, MCR) of the main propulsion engine(s) or motor(s), as applicable for mechanical drive plants or electric propulsion plants. The power to achieve this speed also shall be not greater than 80 percent of the installed generator rating for electric propulsion plants with dedicated propulsion generator sets. For integrated electric propulsion plants, the power required to achieve this speed shall be not greater than 80 percent of the installed generator set rating following deductions for at-sea ship service power requirements and electric plant growth margins.

Figure 43: Verification Requirement

Verification activities can be captured in the model as Verification Events that include test procedures and/or test configurations. The actual tests are performed external to the CORE database, but are referenced and tracked within the model. After a Verification Event takes place, the respective Verification Requirement should be updated in the model to reflect the status. The relationships involved with verification are shown in the figure below.

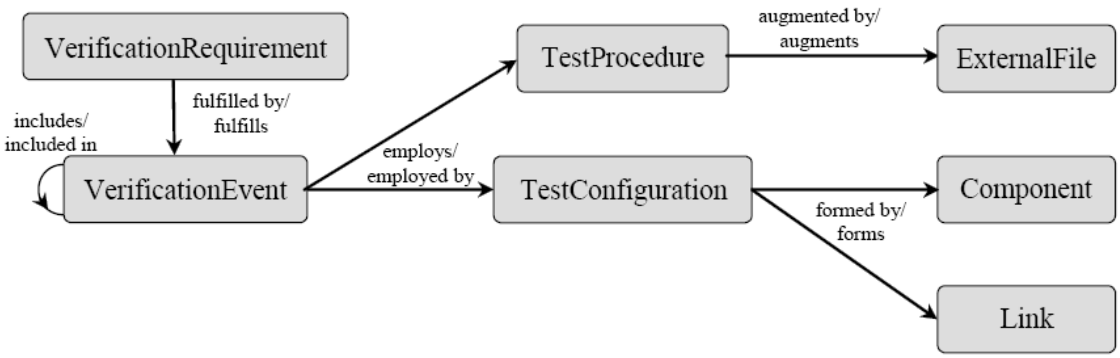


Figure 44: (Vitech Corporation 2009)

4.9 Requirements Traceability

Requirements traceability is one of the key motivators for MBSE. The model provides traceability for each and every element of the design by keeping track of the linkages back to source requirements. Too often in the ship acquisition community there are developed systems that do not effectively meet the needs of the stakeholders. Requirements get lost or manipulated over time and it is extremely difficult to maintain traceability between design documents and the requirements management tool. CORE generates a traceability diagram that shows the relationships graphically and can also output a Requirements Traceability Matrix (RTM) in an easily readable table format. The traceability diagram is too large to display in its entirety, but an excerpt is shown in Figure below. The RTM is displayed in Appendix I as part of the entire System Description Document (SDD).

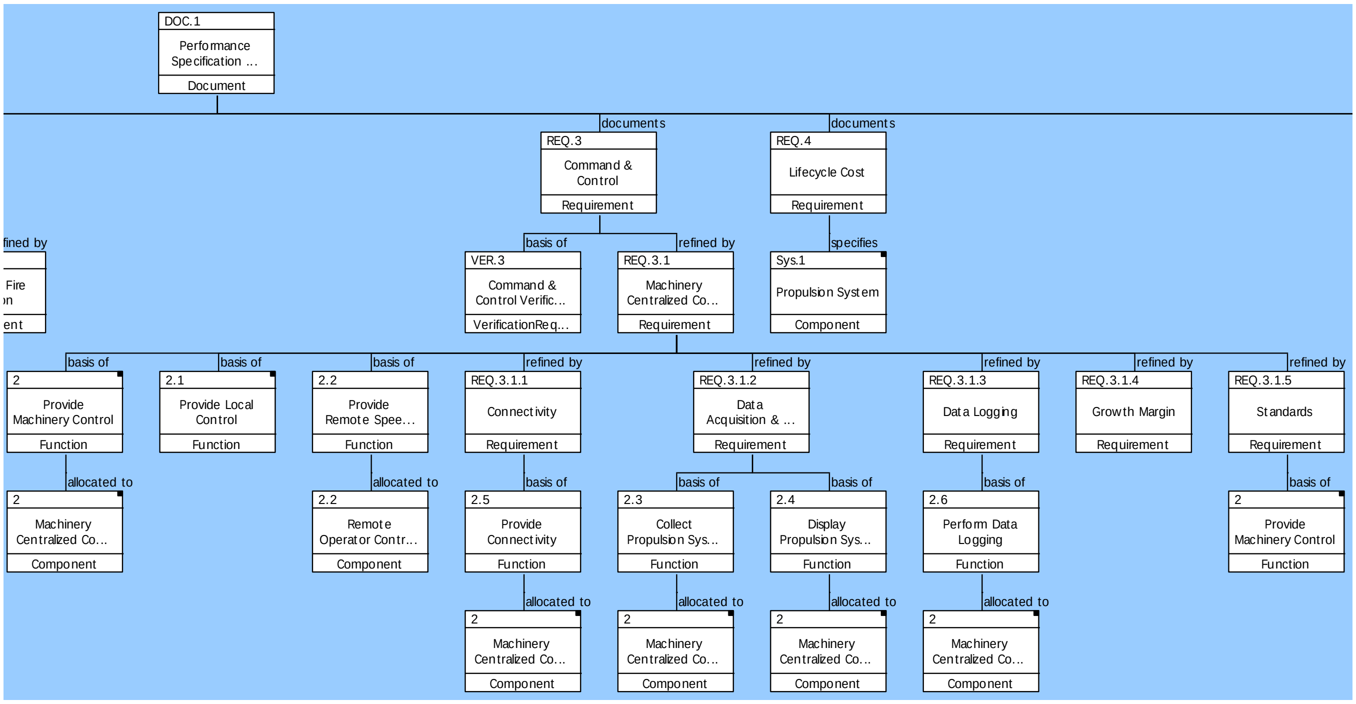


Figure 45: Traceability Diagram

5.0 Decision Making

Architectural models of the system provide a basis for decision making and support architectural decisions driven by real functional requirements. The traceability inherent in a systems model allows for more accurate change assessments and alternatives analysis. The designer is able to see how a small change in one aspect of the design can drastically affect the whole. Risk and cost are able to be incorporated into the model to enhance the decision-making process. Executable models can also be used in an analysis of alternatives (AoA) by conducting system design trade-offs and use cases can be incorporated into the model to verify that the system capability satisfies mission requirements. The following paragraphs will demonstrate a few of these enhanced decision-making attributes of MBSE.

5.1 Trade Study

A trade study is used by systems engineers to compare alternative solutions for a given problem based on some criteria. A measure of effectiveness (MOE) is used to define a property that needs to be evaluated in a trade study. Design decisions as a result of a trade study are entered in the database as requirements and are augmented with an issue to capture the details of the design decision. In this example, a trade study was conducted for two related aspects of the design.

1. IPS or Mechanical Drive
2. Propulsor Selection

IPS or Mechanical: Because the selection of IPS or Mechanical will affect the propulsor selection criteria, that trade study is performed first. An analysis of the originating requirements concludes that there is no indication of preference for one design over the other from the customer. A trade study was conducted with the following measures of effectiveness (that come from the requirements): Fuel Efficiency and Cost. Based on the criteria, a diesel mechanical drive was selected in the trade study because it was cheaper compared to IPS and had comparable fuel savings. IPS offers many advantages such as flexibility in arrangements and optimum loading, but this was not important to the customer and came with a higher price tag. The results and rationale of the trade study were documented in the Issue and lead to a refining requirement. Note that the trade study itself is not conducted in CORE, but the details and findings of the external trade study are captured within the database, along with the organization that conducted it. This allows for traceability of design decisions back to the originating rationale. Within the database, Component "1.2 Transmission" generates the issue "IPS-Mechanical Drive Trade Study" which results in a requirement "REQ.1.2.2 Mechanical Drive", as shown in Figure 46.

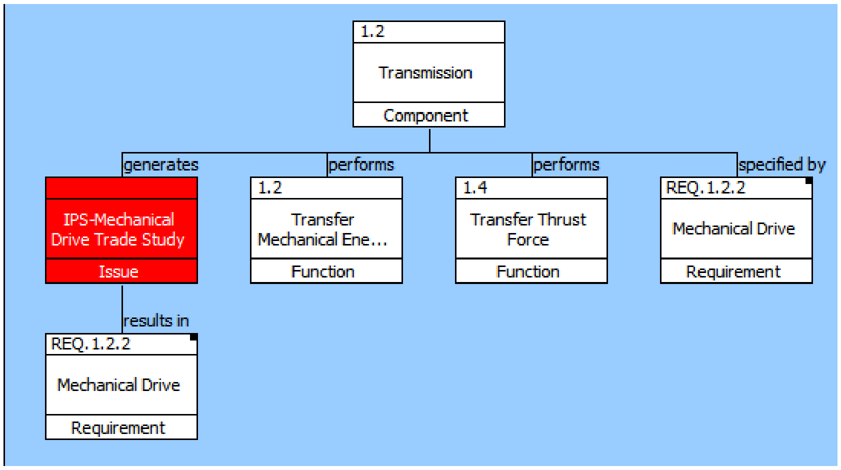


Figure 46: IPS Trade Study

Propulsor: Based on the selection of Mechanical Drive, the controllable pitch propeller was selected as the best choice of propulsor. The choices were fixed pitch propeller, controllable pitch propeller, ducted or shrouded propeller, or waterjet. This is all documented in the database as an issue and refining requirement. The fixed pitch (FP) propeller requires special measures for stopping and reversing: it must be possible to change the direction of rotation of the propeller in either the gearbox or the driving machinery. Controllable pitch propellers offer advantages in maneuverability (i.e. reversing and low speed capability). Disadvantages of controllable pitch propellers (CPP) are a larger hub, a hollow shaft, a hydraulic control system, and a lower efficiency. Overall the CPP is more complicated and expensive, and is more prone to cavitation than a FP propeller. Ducted propellers offer protection to the propeller blades and contribute to the thrust generated by the propeller, particularly at low loads. This allows the propeller to have a smaller diameter than it would as an open propeller. However, the additional friction between the flow and the duct causes slightly lower overall efficiency compared to an open propeller. A waterjet is an option for high-speeds and it is light and efficient. It has no underwater appendages, high efficiency, low weight, low underwater noise, no reversing gear, and no long transmission line. A trade study comparing evaluating propulsor performance requirements and cost has led to the selection of the controllable pitch propeller as the best choice. A performance analysis of the fixed pitch propeller design found that the ship did not meet REQ.1.1.2 Stopping or REQ.1.1.3 Thrust. The controllable pitch ship design was able to meet both requirements. The relationships related to this trade study are shown in Figure 47.

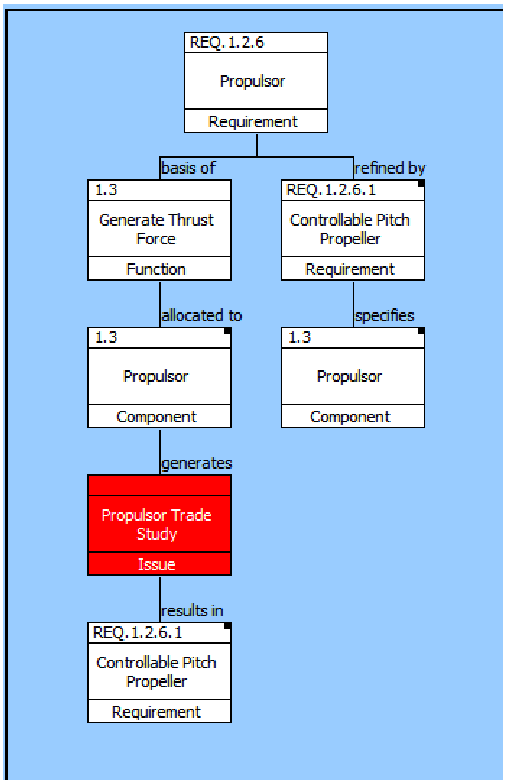


Figure 47: Propulsor Trade Study

5.2 Change Assessment

One of the advantages of storing the system design in an integrated design repository is the ease in which impact analysis and change assessments can be performed. For example, suppose the customer wishes to know the impact of exchanging or replacing the prime mover. The “Behavior Impact of Physical Change” was selected in which a diagram is displayed that shows which functions and which inputs and outputs may be affected. It shows which functions the replacement must perform and shows the data interfaces between the replacement component and the other elements of the system/context. In Figure 48, it is easy to see that changing the prime mover affects the system’s ability to generate mechanical energy, thus affecting the engine break power, exhaust air, combustion air, fuel oil, and start air. This is a simplified example, but one can imagine in an intricately modeled system how this capability can be extremely advantageous.

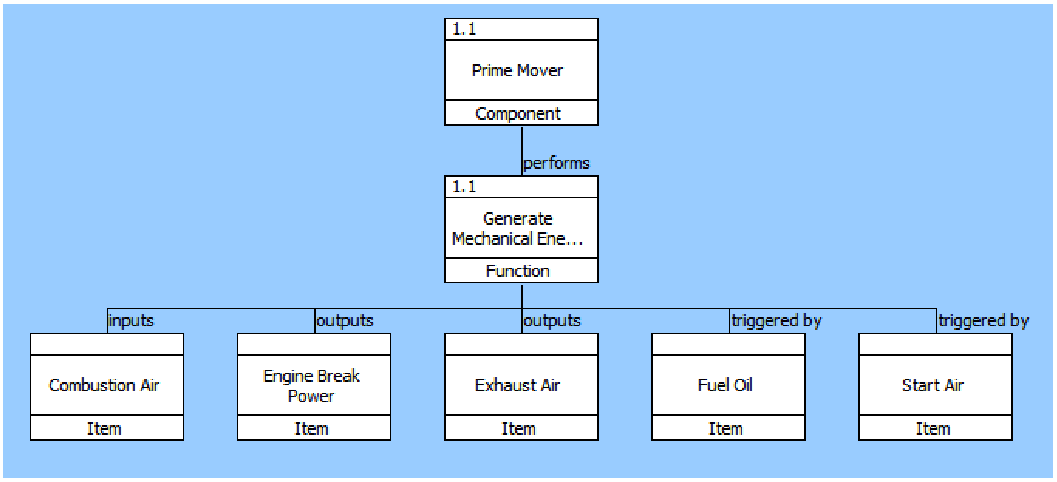


Figure 48: Behavior Impact of Prime Mover Change

Suppose the customer wishes to change the Machinery Centralized Control Station (MCCS). Figure 49 was quickly generated to show the customer exactly what functions will be affected by changing the MCCS, and which inputs and outputs are affected. This is a valuable tool that not only generates the information instantly, but also produces it in a readable form that all stakeholders can easily understand.

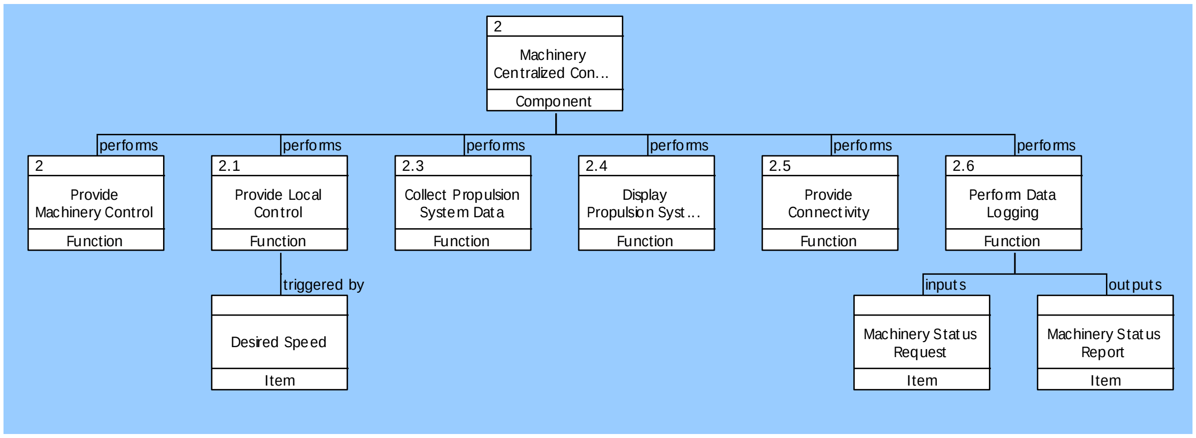
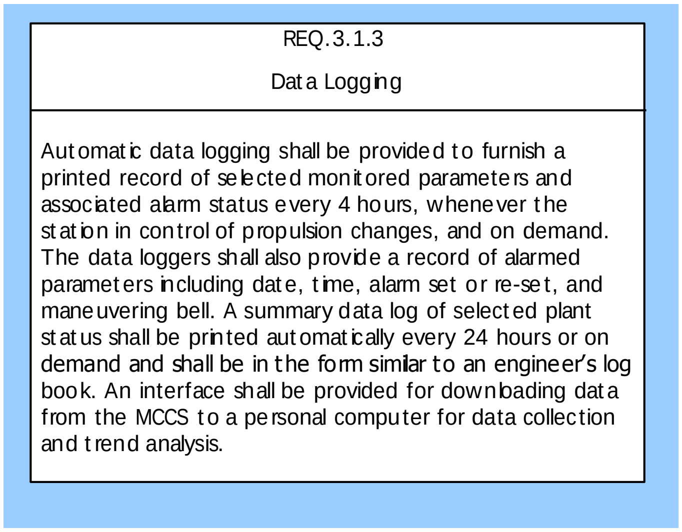


Figure 49: Behavior Impact of Machinery Centralized Control Station Change

A similar assessment was done in response to a changing requirement in which all the functions and components associated with the changed requirement were quickly identified. The initial requirement for data logging is shown in Figure 50 below.



REQ.3.1.3

Data Logging

Automatic data logging shall be provided to furnish a printed record of selected monitored parameters and associated alarm status every 4 hours, whenever the station in control of propulsion changes, and on demand. The data loggers shall also provide a record of alarmed parameters including date, time, alarm set or re-set, and maneuvering bell. A summary data log of selected plant status shall be printed automatically every 24 hours or on demand and shall be in the form similar to an engineer's log book. An interface shall be provided for downloading data from the MCCS to a personal computer for data collection and trend analysis.

Figure 50: REQ.3.1.3 Data Logging

Suppose the Data Logging requirement was to change. The requirement to have an automatic printed record of monitored parameters and alarm status every 4 hours was removed so that the only requirement was to provide the record whenever the station in control of propulsion changes and on demand. The impact of the requirement change was analyzed in a matter of seconds with the use of the CORE database. The impact diagram, Figure 51, shows that changing the Data Logging requirement affects the Perform Data Logging function, the Provide Machinery Control function, the MCCS, and the functional domain sets.

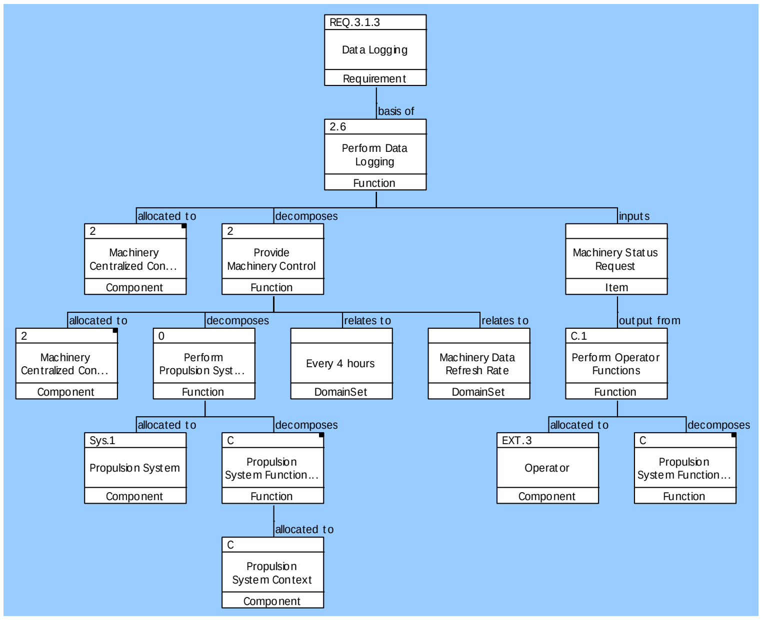


Figure 51: Impact of changing REQ.3.1.3

In addition to providing quick change assessments, MBSE provides a way to store the numerous changes, decisions, and rationale in the database serving as a kind of “corporate memory”. As stated before, the ship design process could span several years and it is extremely difficult to keep track of every single design change over that length of time using a document-centric approach. Additionally, people on the design team at the beginning of a ship design process may or may not be the same people at the end. Personnel change jobs often and take their respective knowledge with them. CORE is able to capture decisions and design attributes as they evolve over time through “versioning”. Versioning allows users to manage and report changes in the central design repository, and view all changes with the attribute history report. This provides all members of the design team with a comprehensive look at the database evolution and visibility of who made the change and when it was made. Previous versions of the design are maintained in the repository and can be restored at any time. An example of versioning is not incorporated in this thesis as the versioning capability was not provided with the CORE University Edition software.

6.0 Results

The goal of this thesis was to explore the possibilities and potential benefits of using a model-based approach to developing systems architecture in naval ship design. The specific questions that this thesis sought to answer, stated in the introductory paragraph, include:

1. *Can MBSE be used to develop the systems architecture of a naval warship?*
2. *Does MBSE provide any benefit to the designer? In what way?*
3. *Is the decision making process enhanced through the use of modeling?*
4. *Where does systems architecture development fit into the overall ship design process?*
5. *What is the right tool to be used in developing the architecture?*

Although the answers to these questions have been given throughout the body of this thesis indirectly, a brief summary of the answers is provided.

Can MBSE be used to develop the systems architecture of a naval warship?

> In this thesis the systems architecture of a ship's subsystem, the propulsion system, was developed using MBSE. This architecting process can clearly be extended to develop the systems architecture of a naval warship. The increased complexity of designing the entire ship could only enhance the relevance and benefits of using MBSE. A significant barrier in the use of MBSE in ship design is the deeply embedded document-centric nature of the ship acquisition process. The transition to a model-based development strategy should be incremental in nature and proceed in a steady, goal-oriented way. The approach should start with the introduction of various MBSE pilot projects in the ship design process in order to quantify benefits in terms of efficiency, cost, schedule, and risk. The transition would include a considerable amount of training in order to effectively use the system model and to maximize the perceived benefits of a model-based environment.

Does MBSE provide any benefit to the designer?

> The potential benefits of using MBSE were described in detail in the previous chapters and include: enhanced communication, requirements traceability, and improved decision making. The process of developing the architecture of the propulsion system

demonstrated the power of MBSE with regards to requirements traceability and decision-making. Although communication was not specifically addressed when architecting the propulsion system, MBSE would clearly enhance communication between stakeholders by providing a single repository of information. The designer would be able to easily keep the customer informed and engaged at all stages of the design process. This will ensure early on that the design is consistent with the customer's requirements and will eliminate the painful situation of presenting a final design that does not meet the needs of the customer.

Is the decision making process enhanced through the use of modeling?

As stated before, architectural models of the system provide a basis for decision making and support architectural decisions driven by real functional requirement. The traceability inherent in a systems model allows for more accurate change assessments and alternatives analysis. The designer is able to see how a small change in one aspect of the design can drastically affect the whole. Risk and cost can also be incorporated into the model to enhance the decision-making process. Executable models can be used in an analysis of alternatives (AoA) by conducting system design trade-offs and use cases can be incorporated into the model to verify that the system capability satisfies mission requirements. A few of these enhanced decision-making attributes of MBSE were presented including a trade-off analysis, a change assessment, and the ability to easily track changes and design decisions.

Where does systems architecture development fit into the overall ship design process?

Systems architecture is important because it provides a way to understand, design, and manage complexity. In the ship design process, there is a significant need to ensure that the architecture is not only well-defined, but also addresses the needs of the stakeholders. For this reason, systems architecture development must begin at the very early stages of the ship design process. The MBSE process used in developing the propulsion system started at the beginning with developing requirements. As explained in the case studies section, there are those in industry at the forefront of MBSE adoption that use system models to communicate design and performance requirements. The key is to define a standard process for developing systems architectures to be used consistently across DOD.

What is the right tool to be used in developing the architecture?

A comparison of MBSE tools and methodologies extends beyond the scope of this thesis. Vitech CORE was an easily accessible software tool through the Vitech CORE University program, and therefore was chosen simply for that reason. NoMagic's MagicDraw UML software with SysML plug-in was experimented with, but Vitech CORE was found to be more user-friendly from an untrained perspective. Further research and evaluation is required in order to determine the right MBSE tool to use in ship design applications.

In addition to answering the posed questions in the introduction, the research described in this thesis has provided the author some useful insight into MBSE and its applicability to naval ship design and acquisition.

A MBSE approach could be instrumental in streamlining the Navy Acquisition process by providing improved visibility and communication of the system design specification with a centralized database. As described earlier, the DOD acquisition process is based on traditional, document-driven programmatic reviews. SECNAVNOTE 5000 (Feb 2008) implemented the "2-pass, 6-gate" process which requires the development and approval of a System Design Specification (SDS) prior to Milestone B. The SDS currently takes the form of a single document that aims to identify derived requirements, technology development risks, design standards, and expected system attributes. The intent of the SDS is to provide decision makers improved visibility and insight into the capabilities, costs, and risks of the system earlier in the acquisition process in order to facilitate better early stage decisions. The current process puts emphasis on developing the documentation for approval, instead of developing the system for approval. The value of using a MBSE approach is that emphasis is placed on developing the system first, and generating documentation is secondary. Using a central database to capture the derived requirements, design standards, and expected systems attributes allows for instant generation of desired views or documents while also ensuring consistency. Instead of creating various documents throughout the acquisition process, one model could serve as the project specification and desired reports could be instantly generated with a click of a button. Changes can be made quickly and there is never confusion as to which “document” is the most up to date.

MBSE complements the set-based design methodology because it can clearly convey which elements of the design are stable and which are flexible. Set-based design defers detailed specification until tradeoffs are more fully understood, therefore allowing more of the design effort to proceed concurrently. In CORE, an element of the design that requires further analysis can be tagged with an issue that captures the details of the tradeoff study including the current status, the rationale, and the due date. The issue can then be assigned to a particular organization or the persons responsible for the trade study. This is the way in which the database is managed in order to “keep score” of the variable attributes. “Traditional, document-driven systems development methods are designed to create a ‘point solution’ – that is, a solution for a specific and static set of requirements. These methods result in systems that are sluggish in their response to dynamic conditions and changing requirements, expensive to maintain over extended periods of time, and prone to system failure.” (Balmelli, et al. 2006)

Communicating ship requirements to the shipbuilder is currently done in a document-centric way through a Statement of Work (SOW) and Ship Specification. The SOW document details the work the contractor will perform and specifies when necessary how the work is to be performed. The Ship Specification document sets forth the technical performance requirements that the ship must achieve (what the ship will do). This method is by all accounts inefficient and known to be extremely error prone. As explained throughout this thesis, MBSE offers enhanced communication through the use of a single design repository as opposed to various documents and diagrams used in a document-based approach. MBSE has the potential to more effectively communicate the Navy’s requirements in order to establish a contractual baseline between the Navy and the shipbuilder. Issues, derived requirements, questions, rationale, risk, etc. can all be captured in the model to facilitate communication. Clear concise communication of requirements and expectations earlier in the design process would reduce risk and downstream cost and/or re-work as experienced in recent ship programs.

Traditional system development methods are based on a static and predictable set of system requirements. In reality, requirements are volatile and have potential to be changed over time as the system development process evolves. As a whole, the ship design community has not appropriately managed the risk of requirements changing which has led to numerous programs running over time and over budget. The approaches for dealing with the requirements volatility have been inconsistent and sometimes include using margins based on past ship performance problems. The systems engineering process should be robust enough to quickly and easily adapt to changing requirements, and this is where the legacy document-driven approach falls short. “Experience has shown that traditional requirements-driven methodologies result in systems that are limited in their capability to self-modify in response to evolving mission or business needs, brittle and difficult to manage in adapting to new requirements, and expensive to maintain over an entire product life cycle.” (Balmelli, et al. 2006) MBSE development is much better suited to handle the unpredictable requirements changes because all the design information is contained in one place. MBSE makes impact and change assessments almost trivial because all the system attributes and element relationships in the model are instantly updated when a change is made.

MBSE has potential to improve the capabilities-based architecture development process. There are several published documents and papers that describe this potential in detail using the operational architecture domain in CORE. ((Whitcomb, et al. 2008) (Dickerson and Soules 2002)

7.0 Conclusions

The purpose for modeling a system must be clearly defined upfront in terms of the expected results of the modeling effort before the process begins. The purpose for modeling should be used to determine the scope of the modeling effort in terms of model breadth, depth, and fidelity. Based on the results of the research described herein, MBSE has tremendous potential in various aspects of ship design and acquisition and further research and pilot projects are recommended to quantify the projected benefits in terms of schedule, cost, and risk.

MBSE is a significant paradigm shift. Adopting MBSE in ship design would require a shift in traditional acquisition strategy which still remains purely document-driven. Any shift toward MBSE would surely meet with resistance at first, but there are many lessons learned from industry on how to implement MBSE into an organization.

Future Work

- Further explore the use of MBSE to streamline requirements and communication between the Navy and the Shipbuilder.
- Quantify benefits of using MBSE in ship design in terms of schedule, cost, and risk
- Explore the possibilities of deriving a DSM (Design Structure Matrix) from the system model. Specifically, devise an algorithm for automatically (or semi-automatically) constructing a Model-based DSM (MDSM) directly from the CORE model. This has been demonstrated manually from OPM to DSM and could add significant value to the project management aspect of the design. (Sharon, Dori and de Weck 2009)
- Integrate ship design analysis tools such as ASSET, POSSE, and MaxSurf with the architecting process and system design model (CORE or SysML) in order to allow for a physics-based quantitative analysis.

Bibliography

Adams, Kenneth, and Ken Kott. "Using Required Operational Capabilities to Connect Operational and Technical Requirements." *Naval Engineers Journal* 120, no. 4 (2008): 37-44.

Baker, Loyd, Paul Clemente, Bob Cohen, Larry Permenter, Byron Purves, and Pete Salmon. "Foundational Concepts for Model Driven System Design." *Vitech Corporation.* 2009. http://www.vitechcorp.com/whitepapers/files/200701031636590.baker_etal96.pdf (accessed October 6, 2009).

Balmelli, L., D. Brown, M. Cantor, and M. Mott. "Model-driven Systems Development." *IBM Systems Journal* 3, no. 45 (2006).

Bernstein, Joshua I. *Design Methods in the Aerospace Industry: Looking for Evidence of Set-Based Practices.* Master of Science Thesis, Massachusetts Institute of Technology, 1998.

CJCS, Chairman of the Joint Chiefs of Staff. "CJCSI 3170.01G Joint Capabilities Integration and Development System (JCIDS) Manual." July 31, 2009.

—. "CJCSI 3170.01G Joint Capabilities Integration and Development System Instruction." March 1, 2009.

Crawley, Edward. "ESD.34 Systems Architecture Lecture Notes." Cambridge: MIT, January 2007.

Crawley, Edward, et al. "The Influence of Architecture in Engineering Systems." *ESD Monograph.* Cambridge, MA: MIT, March 29-31, 2004.

DAU. "Defense Acquisition Guidebook." *Defense Acquisition University.* February 19, 2010. https://dag.dau.mil/ (accessed March 5, 2010).

—. "Systems Engineering Fundamentals." Defense Acquisition University, January 2001.

Dickerson, Dr. Charles, and CAPT (USN Ret) Soules. *Using Architecture Analysis for Mission Capability Acquisition.* Washington DC: Asst Sec of the Navy for Research, Development and Acquisition, 2002.

DoD, Department of Defense. "Deparment of Defense Architecture Framework Version 2.0 Volume I." May 28, 2009.

DOD, Department of Defense. "DODI 5000.02 Operation of the Defense Acquisition System." December 8, 2008.

Doerry, Norbert. "Systems Engineering and Zonal Ship Design." *ASNE Engineering the Total Ship Symposium (ETS).* Arlington, VA: ASNE, 2006.

Doerry, Norbert, and Howard Firemean. "Designing All Electric Ships." *International Marine Design Conference.* Ann Arbor, 2006.

Edwards, Rear Admiral Mark J., and Rear Admiral H. G. III Ulrich. "The Next Revolution at Sea." *U.S. Navy Proceedings*, October 2003.

Estefan, Jeff A. *Survey of Model-Based Systems Engineering (MBSE) Methodologies.* MBSE methodology survey, Seattle, WA: INCOSE, 2008.

Evans, J. Harvey. "Basic Design Concepts." *Naval Engineers Journal Volume 21*, 1959.

Friedenthal, Sanford. *A Practical Guide to SysML: The Systems Modeling Language.* Burlington: Elsevier/Morgan Kaufmann Publishers, 2008.

Friedman, Paul, interview by Nadia Tepper. *Principal Engineer and Advanced Concepts Engineer, Bath Iron Works* (October 26, 2009).

INCOSE. *International Council on Systems Engineering: Standards Resource Center.* Novemeber 23, 2004. http://www.incose.org/practice/standards/heritage.aspx (accessed March 10, 2010).

INCOSE, International Coucil on Systems Engineering. "Systems Engineering Vision 2020, Version 2.03, TP-2004-004-02." 2007.

Lamb, Thomas. *Ship Design and Construction, Volume I.* The Society of Naval Architects and Marine Engineers, 2003.

Maier, Mark, and Eberhardt Rechtin. *The Art of Systems Architecting.* Boca Raton: CRC Press LLC, 2002.

Mercer, Brad. "Thoughts on Architecting...and How to Improve the Practice." *Naval Post Graduate School.* The MITRE Corporation. January 31, 2008. http://www.nps.edu/Academics/Institutes/Meyer/docs/Jan%2031%202008Thoughts_on_Architecting.pdf (accessed March 5, 2010).

NAVSEA . "Performance Specification System Specification for the Auxiliary Dry Cargo Ship T-ADC(X)." *FAS Military Analysis Network.* March 18, 1998. http://www.fas.org/man//dod-101/sys/ship/adc-x.htm (accessed January 25, 2010).

OMG, Object Management Group. *OMG Systems Modeling Language: The Official OMG SysML site.* www.omgsysml.org (accessed March 8, 2010).

Quayle, Tony. "ESD.33 System Engineering Lecture Notes 2009." Massachusetts Institute of Technology, July 16, 2009.

Rechtin, Eberhardt. *Systems Architecting: Creating & Building Complex Systems.* Englewood Cliffs, NJ: Prentice-Hall, Inc., 1991.

Rumsfeld, Donald H., Secretary of Defense. "Quadrennial Defense Review Report." Washington D.C., 2001.

SECNAV, Secretary of the Navy. "SECNAVNOTE 5000." *SECNAVNOTE 5000 Department of the Navy (DON) Requirements and Acquisition Process Improvements.* Feb 26, 2008.

Sharon, Amira, Dov Dori, and Olivier de Weck. "Model-Based Design Structure Matrix: Deriving a DSM from an Object-Process Model." *Second International Symposium on Engineering Systems.* Cambridge, MA: MIT ESD, 2009.

Stoffel, Dr. James C. "Systems Engineering - An Imperative for Economic Growth." *INCOSE Keynote Address.* July 12, 2008.

Vitech Corporation. "CORE 6 Architecture Definition Guide." November 2009. www.vitechcorp.com (accessed January 25, 2010).

—. "CORE 6 System Definition Guide." *www.vitechcorp.com.* 2009.

Walker, Stephen K. *Capabilities-Based Planning How it is Intended to Work and Challenges.* USAWC Strategy Research Project, Carlisle, PA: U.S. Army War College, 2005.

Walsh, Sean. "Early Stage Ship Design Seminar." *SNAME Annual Meeting.* Providence, 2009.

Whitcomb, C., R. Vaidyanathan, J. Pluta, J. Kiihne, and W. Solitario. "Architecture development for an unmanned vehicle sentry systems of systems." *23rd Bristol International Unmanned Air Vehicle Systems (UAVS) Conference.* Bristol, UK, 2008.

Woud, Hans, and Douwe Stapersma. *Design of Propulsion and Electric Power Generation Systems.* London: IMarEST, 2002.

Appendix I - System Description Document (SDD)

SYSTEM DESCRIPTION DOCUMENT

FOR

Propulsion System

Prepared on:

Friday, March 26, 2010

Prepared By:

Nadia A. Tepper
77 Massachusetts Ave.
Cambridge, MA 02139

1 Component Overview

Propulsion System

Description:

The propulsion system is one of the most important systems onboard a marine vessel. The function of the propulsion system is to generate thrust, which enables the ship to move at the desired speed. The propulsion system consists of three main components: prime mover, transmission, and propulsor.

System Mission:

The overall mission of the propulsion system is to propel the ship through the water at a desired speed.

Allocated Functions:

0 Perform Propulsion System Functions

Source Document(s):

Performance Specification Document

Document Date: Wednesday, March 18, 1998

Description: This specification is a description of the system requirements for T-ADC(X). Included are the mission, capabilities, major systems requirements, interfaces, environmental constraints, interchange requirements, logistics concept, personnel, and verification requirements.

This specification establishes overall system requirements to guide the subsequent engineering development and more detailed specifications.

External Interfacing System(s):

EXT.1 Atmosphere
EXT.2 Fuel
EXT.3 Operator
EXT.4 Ship Hull
EXT.5 Water

Assigned Design Constraints:

REQ.4 Lifecycle Cost
REQ.6.1 Availability
REQ.6.2 Reliability w/ repair
REQ.6.3 Reliability w/out repair

REQ.7 Service Life

Triggers from External Source(s):

Desired Speed

Source of Trigger(s):

C.1 Perform Operator Functions

Machinery Status Request

Source of Trigger(s):

C.1 Perform Operator Functions

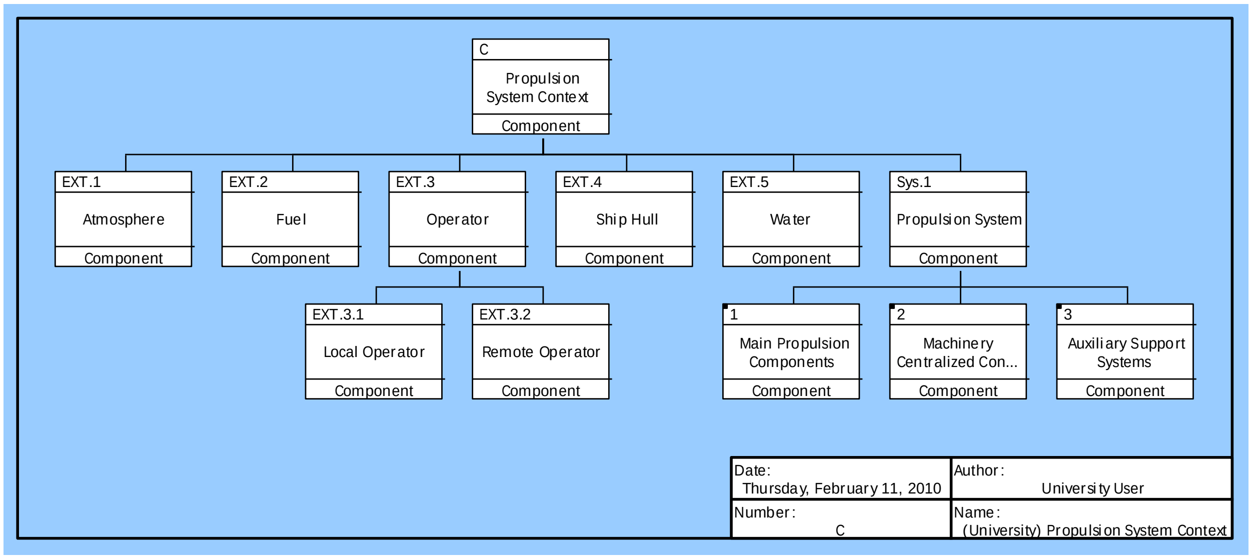


Figure 1 Propulsion System Physical Context

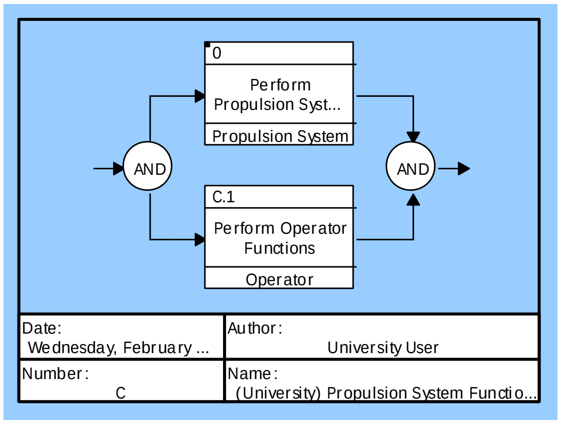


Figure 2 Propulsion System Functional Context

1 Component Overview

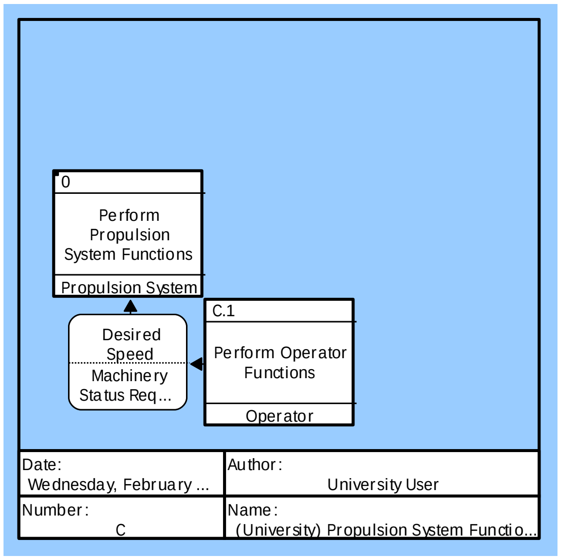


Figure 3 Propulsion System Functional Interface Context

2 Originating Requirements

REQ.1 Mobility

Requirement Statement:

The ship shall be capable of a sustained speed of 20 knots in the Full Load (Condition D), calm water, and clean hull using no more than 80 percent of the installed engine rating (maximum continuous rating, MCR) of the main propulsion engine(s) or motor(s), as applicable for mechanical drive plants or electric propulsion plants. The power to achieve this speed also shall be not greater than 80 percent of the installed generator rating for electric propulsion plants with dedicated propulsion generator sets. For integrated electric propulsion plants, the power required to achieve this speed shall be not greater than 80 percent of the installed generator set rating following deductions for at-sea ship service power requirements and electric plant growth margins. The ship shall be capable of smooth, bumpless acceleration and deceleration between the minimum ship speed associated with the lowest sustainable prime mover rpm and corresponding propeller pitch (where controllable pitch propeller(s) are provided) setting and maximum ship speed.

Source Document(s):

Performance Specification Document

Refined By Subordinate Requirements:

REQ.1.1 Manueverability
REQ.1.2 Sustained Speed
REQ.1.3 Endurance
REQ.1.4 Fuel Efficiency
REQ.1.5 Mobility Support Systems

REQ.1.1 Manueverability

Requirement Statement:

The ship shall have the capability to maneuver in formation as described by the requirements of this section. Unless otherwise specified in this section, the maneuverability requirements shall apply to the ship operating in deep calm water without wind or current. The maneuverability requirements shall be met at ship loading conditions corresponding to the deepest and shallowest drafts that occur and the associated trims during the UNREP mission. Maneuverability requirements shall be met at initial speeds of 5, 14, and 20 knots, unless otherwise specified. Maneuverability requirements, except for section 3.3.1.b.4, shall be met without the assistance of lateral thrusters, even if thrusters are provided.

Parent Requirement's Source Document(s):

Performance Specification Document

Refines Higher-Level Requirement:

REQ.1 Mobility

Refined By Subordinate Requirements:

REQ.1.1.1 UNREP
REQ.1.1.2 Stopping
REQ.1.1.3 Thrust

REQ.1.1.1 UNREP

Requirement Statement:

The ship shall be capable of simultaneous UNREP of two customer ships alongside at speeds of 12-16 knots.

Parent Requirement's Source Document(s):

Performance Specification Document

Refines Higher-Level Requirement:

REQ.1.1 Manueverability

REQ.1.1.2 Stopping

Requirement Statement:

The ship shall be capable of stopping within a time of 6 minutes and a head reach of not greater than nine ship lengths with a track departure of no greater than one ship length and heading departure of no greater than 15 degrees without damage to ship systems.

Parent Requirement's Source Document(s):

Performance Specification Document

Refines Higher-Level Requirement:

REQ.1.1 Manueverability

Generates Issues:

Stopping

2 Originating Requirements

REQ.1.1.3 Thrust

Requirement Statement:

The thrust bearing shall be capable of withstanding transfer of maximum thrust in one direction to maximum thrust in the opposite direction. The ship shall be capable of transfering maximum thrust in one direction to maximum thrust in the opposite direction in 12 seconds or less.

Parent Requirement's Source Document(s):

Performance Specification Document

Refines Higher-Level Requirement:

REQ.1.1 Manueverability

Basis Of:

Function: 1.4 Transfer Thrust Force

REQ.1.2 Sustained Speed

Requirement Statement:

The ship shall be capable of a sustained speed of 20 knots in the Full Load (Condition D), calm water, and clean hull using no more than 80 percent of the installed engine rating (maximum continuous rating, MCR) of the main propulsion engine(s) or motor(s), as applicable for mechanical drive plants or electric propulsion plants. The power to achieve this speed also shall be not greater than 80 percent of the installed generator rating for electric propulsion plants with dedicated propulsion generator sets. For integrated electric propulsion plants, the power required to achieve this speed shall be not greater than 80 percent of the installed generator set rating following deductions for at-sea ship service power requirements and electric plant growth margins.

Parent Requirement's Source Document(s):

Performance Specification Document

Refines Higher-Level Requirement:

REQ.1 Mobility

Refined By Subordinate Requirements:

REQ.1.2.1 Main Propulsion Engine Rating
REQ.1.2.2 Mechanical Drive
REQ.1.2.3 Light Running Margin
REQ.1.2.4 Propulsor

REQ.1.2.5 Shafting

Basis Of:

Function: 1 Provide Propulsion

Causes Risks:

RISK.1 Sustained Speed

REQ.1.2.1 Main Propulsion Engine Rating

Requirement Statement:

The ship shall be capable of a sustained speed of 20 knots in the Full Load (Condition D), calm water, and clean hull using no more than 80 percent of the installed engine rating (maximum continuous rating, MCR) of the main propulsion engine(s)

Refines Higher-Level Requirement:

REQ.1.2 Sustained Speed

Basis Of:

Function: 1.1 Generate Mechanical Energy

REQ.1.2.3 Light Running Margin

Requirement Statement:

Light Running Margin (LRM) shall be between 5% and 6%. This LRM will offer sufficient engine speed margin to maintain constant engine power when the ship deteriorates from trial condition to service condition.

Refines Higher-Level Requirement:

REQ.1.2 Sustained Speed

Basis Of:

Function: 1.1 Generate Mechanical Energy

REQ.1.2.4 Propulsor

Requirement Statement:

The propulsor(s) shall survive the marine environment at the speed-time profile specified with no visible erosion between scheduled drydockings. The propulsor design shall maximize propulsive efficiency and minimize cavitation at all steady ahead operating conditions consistent with other requirements.

Refines Higher-Level Requirement:

REQ.1.2 Sustained Speed

Refined By Subordinate Requirements:

REQ.1.2.4.1 Controllable Pitch Propeller

Basis Of:

Function: 1.3 Generate Thrust Force

REQ.1.2.5 Shafting

Requirement Statement:

The shafting shall survive the marine environment at the speed-time profile specified with no visible erosion between scheduled drydockings. Means shall be provided for locking of shaft(s).

Refines Higher-Level Requirement:

REQ.1.2 Sustained Speed

Basis Of:

Function: 1.2 Transfer Mechanical Energy

REQ.1.3 Endurance

Requirement Statement:

The ship’s machinery shall be capable of continuous operation using distillate fuel in accordance with ASTM D975, Grade 2-D; ISO 8217, F-DMA DFM (North Atlantic Treaty Organization (NATO) Code F-76); and capable of operation for 10,000 nautical miles at 20 knots on JP-5 (NATO Code F-44).

Parent Requirement's Source Document(s):

Performance Specification Document

Refines Higher-Level Requirement:

REQ.1 Mobility

Refined By Subordinate Requirements:

REQ.1.3.1 Fuel Tankage

REQ.1.3.1 Fuel Tankage

Requirement Statement:

The ship's machinery shall be capable of operation for 10,000 nautical miles at 20 knots on JP-5 (NATO Code F-44) and fuel tankage shall be sized accordingly to satisfy endurance requirement (without re-fuel and without falling below 50% of full capacity).

Refines Higher-Level Requirement:

REQ.1.3 Endurance

Basis Of:

Function: 3.2 Provide Fuel

REQ.1.4 Fuel Efficiency

Requirement Statement:

The specific fuel consumption (SFC) shall not exceed 225 g/kWh at 80% MCR (maximum continuous rating). Fuel rates shall be determined using diesel fuel marine (DFM) and shall be calculated based on fuel with a lower calorific value of 42,000 kJ/kg, and ambient air and sea water temperatures of 38 degrees C and 32 degrees C, respectively

Refines Higher-Level Requirement:

REQ.1 Mobility

Refined By Subordinate Requirements:

REQ.1.4.1 Prime Mover
REQ.1.4.2 Fuel

Basis Of:

Function: 1.1 Generate Mechanical Energy

Generates Issues:

Fuel Efficiency Trade Study

REQ.1.4.2 Fuel

Requirement Statement:

Fuel shall be diesel fuel marine (DFM). Fuel levels shall not fall below 50% of total capacity.

Parent Requirement's Source Document(s):

Performance Specification Document

Refines Higher-Level Requirement:

REQ.1.4 Fuel Efficiency

Basis Of:

Function: 3.2 Provide Fuel

REQ.1.5 Mobility Support Systems

Requirement Statement:

Mobility Support Systems shall comply with all mil-spec requirements and standards

Refines Higher-Level Requirement:

REQ.1 Mobility

Refined By Subordinate Requirements:

REQ.1.5.1 Start Air Pressure

Basis Of:

Function: 3 Provide Auxiliary Support
Function: 3.3 Provide Lubrication
Function: 3.4 Provide Cooling Water
Function: 3.5 Provide Combustion Air

REQ.1.5.1 Start Air Pressure

Requirement Statement:

Start air pressure shall be 450 psi

Refines Higher-Level Requirement:

REQ.1.5 Mobility Support Systems

Basis Of:

Function: 3.1 Provide Start Air

REQ.2 Survivability

Source Document(s):

Performance Specification Document

2 Originating Requirements

Refined By Subordinate Requirements:

REQ.2.1 Firefighting

REQ.2.2 Redundancy and Separation

REQ.2.3 Structural Fire Insulation

REQ.2.1 Firefighting

Requirement Statement:

Water mist fire protection system in accordance with NFPA 750 or total flooding systems that do not use gases lethal to humans at fire fighting concentrations shall be used for coverage of category A machinery spaces and spaces containing flammable and combustible liquids and pumping systems.

Parent Requirement's Source Document(s):

Performance Specification Document

Refines Higher-Level Requirement:

REQ.2 Survivability

REQ.2.2 Redundancy and Separation

Requirement Statement:

The number of propulsion engines and generators shall meet redundancy standards of US Navy vessels for survivability. Lube oil service and jacket water systems for propulsion and generator engines shall be designed such that any single failure of a system component or any single break in distributive piping shall not affect more than a single propulsion or generator engine.

Where functionally redundant distributive systems are required herein, the redundant distributive systems shall be separated athwartships by not less than one half the ship’s beam and vertically by not less than two decks. In way of machinery spaces, redundant distributive systems are not required to be run through tanks to maintain separation.

Parent Requirement's Source Document(s):

Performance Specification Document

Refines Higher-Level Requirement:

REQ.2 Survivability

Basis Of:

Function: 4 Provide Redundancy

REQ.2.3 Structural Fire Insulation

Requirement Statement:

In addition to regulatory body requirements, A-60 structural fire insulation shall be provided in accordance with Table IV.

Parent Requirement's Source Document(s):

Performance Specification Document

Refines Higher-Level Requirement:

REQ.2 Survivability

REQ.3 Command & Control

Requirement Statement:

The primary ship control location shall be the Navigating Bridge, with secondary propulsion and thruster (if applicable) control from bridge wings, port and starboard. The command, control, and communications systems and equipment shall be in accordance with the requirements of the Regulatory Body requirements, Classifications Rules, SOLAS, and the ABS Guide for One Man Bridge Operated (OMBO) Ships.

Source Document(s):

Performance Specification Document

Refined By Subordinate Requirements:

REQ.3.1 Machinery Centralized Control System

REQ.3.1 Machinery Centralized Control System

Requirement Statement:

Propulsion Control from the ship control console (SCC) at the Navigating Bridge and main control console (MCC) at the EOS shall include independent and combined speed control of each shaft and propeller pitch where controllable pitch propeller(s) are provided.

The MCCS shall be designed for main control from the MCC and secondary control from the SCC. Transfer of control shall be accomplished by a request to the controlling console and an acknowledgment from the controlling console. During plant operation, the MCCS shall also continuously monitor and control: auxiliary plant temperatures, pressures, flows, and levels; electric plant characteristics; and damage control systems. Abnormal conditions shall actuate alarms to warn of the condition and provide for automatic shutdown in the case of malfunctions which could lead to equipment damage or personnel hazard.

Parent Requirement's Source Document(s):

Performance Specification Document

Refines Higher-Level Requirement:

REQ.3 Command & Control

Refined By Subordinate Requirements:

REQ.3.1.1 Connectivity
REQ.3.1.2 Data Acquisition & Display
REQ.3.1.3 Data Logging
REQ.3.1.4 Growth Margin
REQ.3.1.5 Standards

Basis Of:

Function: 2 Provide Machinery Control
Function: 2.1 Provide Local Control
Function: 2.2 Provide Remote Speed Control

REQ.3.1.1 Connectivity

Requirement Statement:

The MCCS shall be capable to attach and communicate to the local area network (LAN) to download data via open database connectivity to an SQL compliant client/server database installed on the LAN. Data download shall be configurable for both timing and parameter download definition, including bell logging, alarm logging, alarm set or reset. Date and time stamping of all parametric and logging shall be incorporated.

Parent Requirement's Source Document(s):

Performance Specification Document

Refines Higher-Level Requirement:

REQ.3.1 Machinery Centralized Control System

Basis Of:

Function: 2.5 Provide Connectivity

REQ.3.1.2 Data Acquisition & Display

Requirement Statement:

Central data acquisition and display shall be incorporated as an integral part of the MCCS. Multiple color flat panel or CRT monitors shall be provided in the MCC and one color flat panel or CRT shall be provided in the SCC and Chief Engineer’s office for selective display of data items, alarms, and mimics. Color flat panels and CRTs shall be a minimum of 483 mm diagonal and shall be capable of being configured independently of each other to permit display of data, alarms, and mimic on different monitors simultaneously. Mimics shall dynamically display the status of machinery, valves, tank levels and controls on a schematic representation of the system.

Parent Requirement's Source Document(s):

Performance Specification Document

Refines Higher-Level Requirement:

REQ.3.1 Machinery Centralized Control System

Basis Of:

Function: 2.3 Collect Propulsion System Data

Function: 2.4 Display Propulsion System Data

REQ.3.1.3 Data Logging

Requirement Statement:

Automatic data logging shall be provided to furnish a printed record of selected monitored parameters and associated alarm status every 4 hours, whenever the station in control of propulsion changes, and on demand. The data loggers shall also provide a record of alarmed parameters including date, time, alarm set or re-set, and maneuvering bell. A summary data log of selected plant status shall be printed automatically every 24 hours or on demand and shall be in the form similar to an engineer’s log book. An interface shall be provided for downloading data from the MCCS to a personal computer for data collection and trend analysis.

Parent Requirement's Source Document(s):

Performance Specification Document

Refines Higher-Level Requirement:

REQ.3.1 Machinery Centralized Control System

Basis Of:

Function: 2.6 Perform Data Logging

REQ.3.1.4 Growth Margin

Requirement Statement:

MCCS equipment, including computer hardware and software, shall include provisions for at least 20 percent growth for future alarms and controls.

Parent Requirement's Source Document(s):

Performance Specification Document

Refines Higher-Level Requirement:

REQ.3.1 Machinery Centralized Control System

REQ.3.1.5 Standards

Requirement Statement:

MCCS software shall be in an industry standard, high level, non-proprietary language. The system configuration shall permit the system user to change set point levels, add and delete equipment items to be monitored or controlled and to change the contents and format of the bell and data logger printed outputs. Means to prevent unauthorized tampering with MCCS software data and bell logs, and set points shall be provided.

Parent Requirement's Source Document(s):

Performance Specification Document

Refines Higher-Level Requirement:

REQ.3.1 Machinery Centralized Control System

Basis Of:

Function: 2 Provide Machinery Control

REQ.4 Lifecycle Cost

Requirement Statement:

Life cycle cost as defined herein is total cost, which can be divided into two parts, charter plus operating and support costs. Operating and support costs shall include all costs directly attributable to the ship operation and support including such costs as waste oil disposal and trash disposal.

Source Document(s):

Performance Specification Document

Specifies:

Component: Sys.1 Propulsion System

REQ.5 Human Design Integration

Requirement Statement:

An EOS shall be provided for control and monitoring of the propulsion and auxiliary machinery plants. The EOS shall be enclosed, environmentally controlled, and acoustically protected for the safety and comfort of engineering personnel. The EOS shall be located to provide good visibility of and convenient access to the main and auxiliary machinery.

Source Document(s):

Performance Specification Document

Basis Of:

Function: 2.1 Provide Local Control

REQ.6 Reliability, Maintainability, Availability (RMA)

Requirement Statement:

The reliability and maintainability characteristics of the ship's systems shall be high enough to ensure high probabilities of completing all phases of the operating profiles.

Each propulsion engine shall be capable of continuous operation at rated power in all ahead propulsion modes. Quantitative reliability and availability requirements of critical systems are identified in Table IX.

The T-ADC(X) shall be capable of operating throughout the full realm of peacetime and wartime scenarios with minimum time out of service for emergent repairs. The objective for maximum time out of service (i.e., time not available to carry out an existing mission) should be less than 2.5 days per year.

Source Document(s):

Performance Specification Document

Refined By Subordinate Requirements:

REQ.6.1 Availability
REQ.6.2 Reliability w/ repair
REQ.6.3 Reliability w/out repair

REQ.7 Service Life

Requirement Statement:

The ship shall be designed and constructed to provide a 40-year service life with minimum maintenance and repair.

Source Document(s):

Performance Specification Document

Specifies:

Component: Sys.1 Propulsion System

REQ.8 Vibration

Requirement Statement:

The ship and ship components shall be free from excessive vibration. Vibration is excessive when it results in damage or potential of damage to ship structure, machinery, equipment, or systems, or when it interferes or threatens to interfere with the required operation of the ship, its cargo systems, or any ship component. Hull girder, deckhouse, kingpost and crane foundation vibration shall be 10 percent below the upper curve of the peak acceleration or peak velocity values represented by ISO Standard 6954.

Source Document(s):

Performance Specification Document

Refined By Subordinate Requirements:

REQ.8.1 Equipment Foundation
REQ.8.2 Propulsion Shafting Vibration

REQ.8.2 Propulsion Shafting Vibration

Requirement Statement:

Longitudinal and lateral propulsion shafting vibration shall meet the acceptability constraints of Section 4 and 5 of SNAME T & R Code C-5 with the following modification to section 4:

The highest exciting frequency in Section 4.3.2(d) shall be:

(Design RPM/60) (Number of Propeller Blades) (1.41) = a frequency which has to be rounded up to the next higher integral frequency.

Torsional propulsion shafting vibrations shall meet the acceptability constraints of Section 3 of SNAME T & R Code C-5 with the following modification to paragraph 3.2.1:

> For propulsion diesel engine installations, excessive vibratory torque at any operating speed shall be defined as vibratory torque greater than 75 percent of the driving torque at the same speed, or 25 percent of the full load torque, whichever is smaller.

Refines Higher-Level Requirement:

REQ.8 Vibration

Basis Of:

Function: 1.2 Transfer Mechanical Energy

3 Design Constraints

REQ.4 Lifecycle Cost

Design Constraint Statement:

Life cycle cost as defined herein is total cost, which can be divided into two parts, charter plus operating and support costs. Operating and support costs shall include all costs directly attributable to the ship operation and support including such costs as waste oil disposal and trash disposal.

Source Document(s):

Performance Specification Document

Constrains:

Component: Sys.1 Propulsion System

REQ.6.1 Availability

Design Constraint Statement:

The propulsion system should have an availability of 0.8 (threshold), with a goal of 0.98

Refines Higher-Level Requirement:

REQ.6 Reliability, Maintainability, Availability (RMA)

Constrains:

Component: Sys.1 Propulsion System

REQ.6.2 Reliability w/ repair

Design Constraint Statement:

Mean time before failure = 20,000 hours

"Reliability with repair" allows repair or replacement of redundant equipment in the system provided that minimum acceptable system performance can be maintained until the repairs are completed.

Refines Higher-Level Requirement:

REQ.6 Reliability, Maintainability, Availability (RMA)

Constrains:

Component: Sys.1 Propulsion System

REQ.6.3 Reliability w/out repair

Design Constraint Statement:

Reliability w/out repair = 2500 hours

"Reliability without repair" is a run to failure condition. Replacement or repair of failed components, even in repairable redundant sections of the system, is forbidden.

Refines Higher-Level Requirement:

REQ.6 Reliability, Maintainability, Availability (RMA)

Constrains:

Component: Sys.1 Propulsion System

REQ.7 Service Life

Design Constraint Statement:

The ship shall be designed and constructed to provide a 40-year service life with minimum maintenance and repair.

Source Document(s):

Performance Specification Document

Constrains:

Component: Sys.1 Propulsion System

4 Performance Requirements

REQ.1.2.2 Mechanical Drive

Performance Requirement Statement:

The propulsion drive shall consist of a mechanical drive train.

Refines Higher-Level Requirement:

REQ.1.2 Sustained Speed

Specifies:

Component: 1.2 Transmission

Result Of:

Issue: IPS-Mechanical Drive Trade Study

REQ.1.2.4.1 Controllable Pitch Propeller

Performance Requirement Statement:

The ship shall have a controllable pitch propeller.

Refines Higher-Level Requirement:

REQ.1.2.4 Propulsor

Specifies:

Component: 1.3 Propulsor

Result Of:

Issue: Propulsor Trade Study

REQ.1.4.1 Prime Mover

Performance Requirement Statement:

The prime mover(s) shall be an all Diesel configuration.

Refines Higher-Level Requirement:

REQ.1.4 Fuel Efficiency

Specifies:

Component: 1.1 Prime Mover

Result Of:

Issue: Fuel Efficiency Trade Study

5 Issues & Decisions

Part I - Open Issues

Stopping

Issue Description:

At what speeds of advance must this requirement be met.

Originator: University User

Originating Date: Monday, January 25, 2010 at 11:36:26 AM

Severity: Important

Status: Open

Assumptions: Specification section 3.3.1.b states that "Maneuverability requirements shall be met at initial speeds of 5, 14, and 20 knots, unless otherwise specified." This will be assumed as the speeds of advance to meet the stated stopping requirement until the customer clarifies.

Generated By:

Requirement: REQ.1.1.2 Stopping

Part II - Closed Issues

Availability

Issue Description:

What is the meaning of "(Hrs)" in the Availability column. Both inherent availability (A_i) and operational availability (A_o) are measured in percentages.

Originator: University User

Originating Date: Monday, January 25, 2010 at 11:50:15 AM

Severity: Critical

Status: Closed

Decision: Hours has been removed from this column and it has been clarified to be A_i.

Rationale: Customer review provided clarification.

5 Issues & Decisions

Fuel Efficiency Trade Study

Issue Description:

The fuel efficiency requirement leads to a trade study to determine the optimum propulsion plant configuration to meet this requirement.

Originator: University User

Originating Date: Tuesday, February 09, 2010 at 04:01:22 PM

Severity: Critical

Assigned To:

NAVSEA

Status: Closed

Assumptions: It is assumed that the prime mover selection will be the critical factor in meeting this requirement.

Alternatives: 1. Diesel
2. Gas Turbine
3. Combination Diesel & Gas Turbine

Decision: An all diesel configuration (Alternative 1) was selected based on an external trade-study and the US Navy Alternate Propulsion Study published in March 2007

Rationale: Diesels are more fuel efficient than gas turbines, so an all gas-turbine configuration (Alternative 2) was quickly abandoned. A more in depth analysis revealed that the ship has space to accomodate an all diesel configuration which also satisfies the sustained speed requirement of 20 knots. In this case, the gas tubine configuration would lead to an over designed ship (exceeding speed requirements) and fell short of the diesel configuration in terms of fuel efficiency.

Source Document(s):

Alternate Propulsion Study Report

Generated By:

Requirement: REQ.1.4 Fuel Efficiency

Results In Requirement:

Requirement: REQ.1.4.1 Prime Mover

5 Issues & Decisions

IPS-Mechanical Drive Trade Study

Issue Description:

An IPS-Mechanical Drive trade study was conducted in order to come to a design decision on the Transimission system.

Originator: University User

Originating Date: Tuesday, February 16, 2010 at 11:30:58 AM

Severity: Critical

Status: Closed

Assumptions: It is assumed that an electical drive choice would be integrated into the ship service electrical load, thus being termed an integrated power system.

Alternatives: Mechanical Drive, Integrated Power System

Decision: Mechanical Drive

Rationale: The alternative were evaluated based on the criteria of Fuel Efficiency and Cost. Based on the criteria, a diesel mechanical drive was selected in the trade study because it was a lot cheaper and comparable in fuel savings. IPS offers many advantages such as flexibility in arrangements and optimum loading, but this was not important to the customer and came with a higher price tag.

Generated By:

Component: 1.2 Transmission

Results In Requirement:

Requirement: REQ.1.2.2 Mechanical Drive

Propulsor Trade Study

Issue Description:

The propulsor trade study is a direct result of the selected drive.

Originator: University User

Originating Date: Tuesday, February 16, 2010 at 11:31:47 AM

Severity: Critical

Status: Closed

Assumptions: The customer has not stated any preference in propulsor design, thus the decision is left to the designer based on the originating ship performance requirements.

Alternatives: Fixed Pitch Propeller, Controllable Pitch Propeller, Ducted or Shrouded Propeller, Waterjet

Decision: Controllable Pitch Propeller

Rationale: A trade study comparing evaluating propulsor performance requirements and cost has led to the selection of the controllable pitch propeller as the best choice. In terms of cost only, the fixed pitch propeller seemed the best choice, but a performance analysis was done to verify requirements. A performance analysis of the fixed pitch propeller design found that the ship did not meet REQ.1.1.2 Stopping or REQ.1.1.3 Thrust. The controllable pitch ship design was able to meet both originating requirements, but it comes with a higher price tag.

Generated By:

Component: 1.3 Propulsor

Results In Requirement:

Requirement: REQ.1.2.4.1 Controllable Pitch Propeller

Redundancy

Issue Description:

The requirement, "…shall be separated athwartships…and vertically…" should be presented as guidance and an objective vulnerability assessment should be made instead. It is recommended that NSWC-CD be permitted to work with contractors to conduct such assessments, using their System Vulnerability Model (for example).

Originator: University User

Originating Date: Monday, January 25, 2010 at 11:54:59 AM

Severity: Critical

Status: Closed

Decision: The Government has already done vulnerability assessments and determined that this requirement is necessary.

Reliability

Issue Description:

Why are there two reliability factors, one with and one without repairs. Reliability is defined in terms of Mean Time Between Failures (MTBF). If a failure occurs that affects performance, repair is required.

Originator: University User

Originating Date: Monday, January 25, 2010 at 11:51:57 AM

Severity: Critical

Status: Closed

Decision: Definitions have been added to the specification for clarification.

Part III - Rejected Issues

None

6 Risks

RISK.1 Sustained Speed

Risk Description:

The requirement for sustained speed is one that could change (as stated by the customer).

Risk Type: Other

Impact: Medium

Status: Ship speed study in progress

Mitigation Plan:

A ship speed study will be conducted to determine the correct sustained speed required for this type of marine vessel.

Assigned To:

NAVSEA

Caused By:

Requirement: REQ.1.2 Sustained Speed

RISK.2 Podded Propulsor

Risk Description:

There is much risk associated with incorporating a podded propulsor in the design of a naval warship. It is unproven in applications related to the naval warship.

Risk Type: Technical

Impact: High

7 Functional Behavior Model

Part I - Hierarchical Function List

- 0 Perform Propulsion System Functions
  - 1 Provide Propulsion
    - 1.1 Generate Mechanical Energy
    - 1.2 Transfer Mechanical Energy
    - 1.3 Generate Thrust Force
    - 1.4 Transfer Thrust Force
  - 2 Provide Machinery Control
    - 2.1 Provide Local Control
    - 2.2 Provide Remote Speed Control
    - 2.3 Collect Propulsion System Data
    - 2.4 Display Propulsion System Data
    - 2.5 Provide Connectivity
    - 2.6 Perform Data Logging
  - 3 Provide Auxiliary Support
    - 3.1 Provide Start Air
    - 3.2 Provide Fuel
    - 3.3 Provide Lubrication
    - 3.4 Provide Cooling Water
    - 3.5 Provide Combustion Air
  - 4 Provide Redundancy

Part II - Behavior Model

0 Perform Propulsion System Functions

Description:

This top-level root function represents the total functionality of the entire propulsion system.

Allocated To:

Sys.1 Propulsion System

Table 1 0 Perform Propulsion System Functions Interfacing Items

Interfacing Items	Source / Destination
Desired Speed	Triggers Function(s): 0 Perform Propulsion System Functions 2.1 Provide Local Control 2.2 Provide Remote Speed Control Output From: C.1 Perform Operator Functions
Machinery Status Request	Input To: 2.6 Perform Data Logging Triggers Function(s): 0 Perform Propulsion System Functions Output From: C.1 Perform Operator Functions

7 Functional Behavior Model

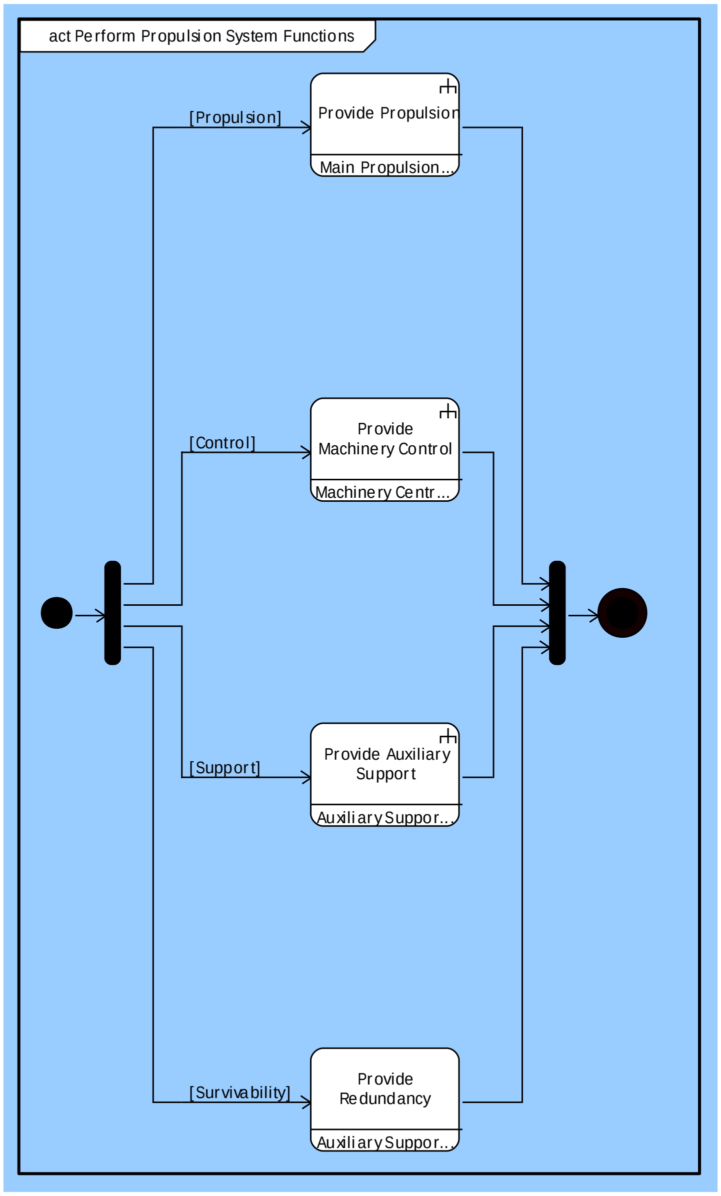


Figure 4 Perform Propulsion System Functions Activity Diagram

7 Functional Behavior Model

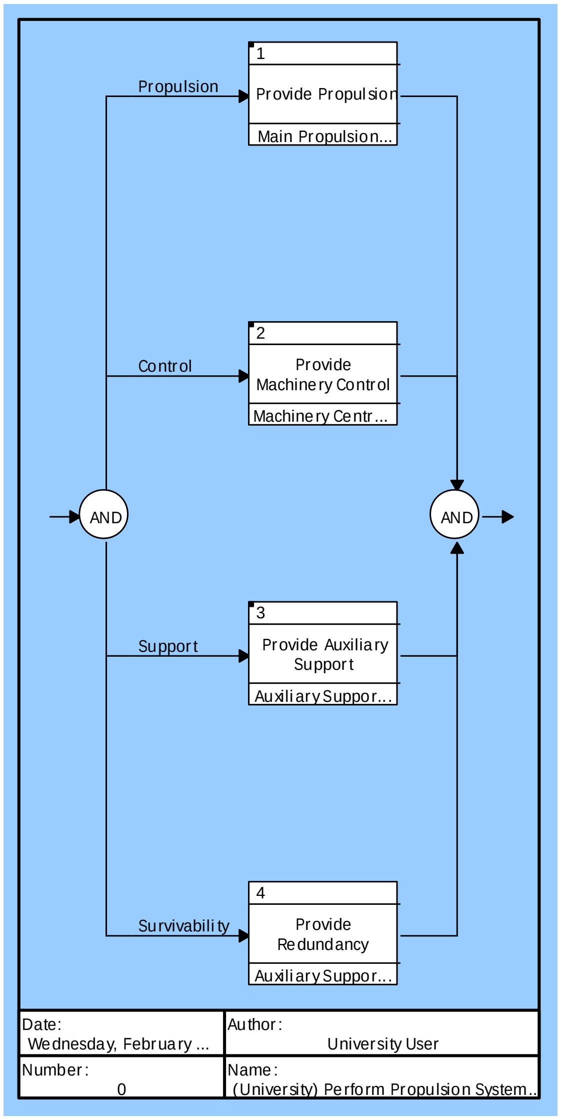


Figure 5 Perform Propulsion System Functions Enhanced FFBD

7 Functional Behavior Model

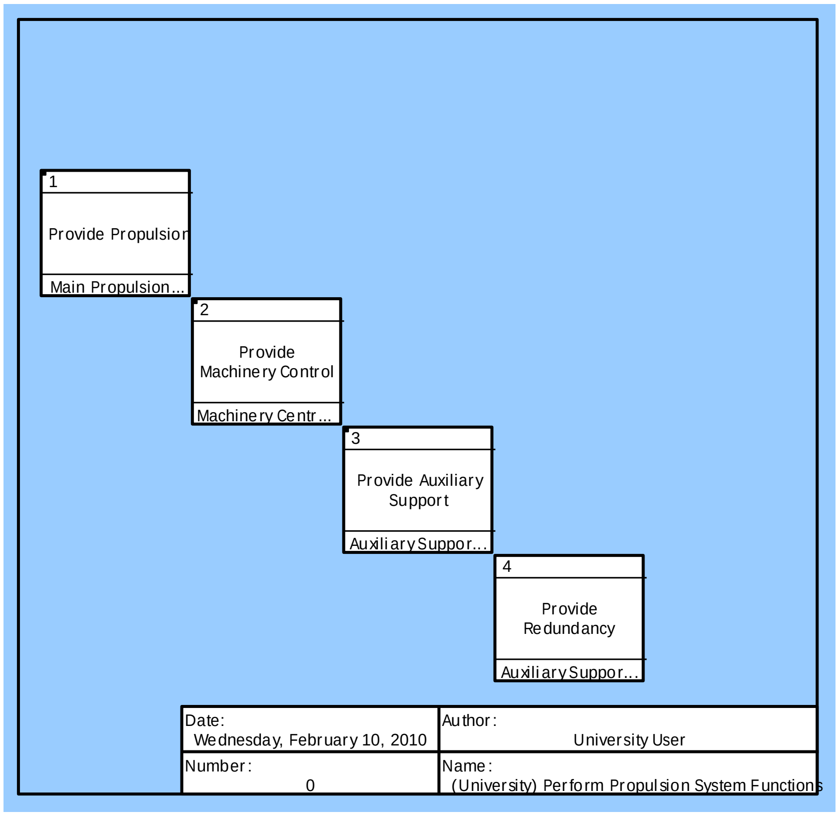


Figure 6 Perform Propulsion System Functions N2 Diagram

7 Functional Behavior Model

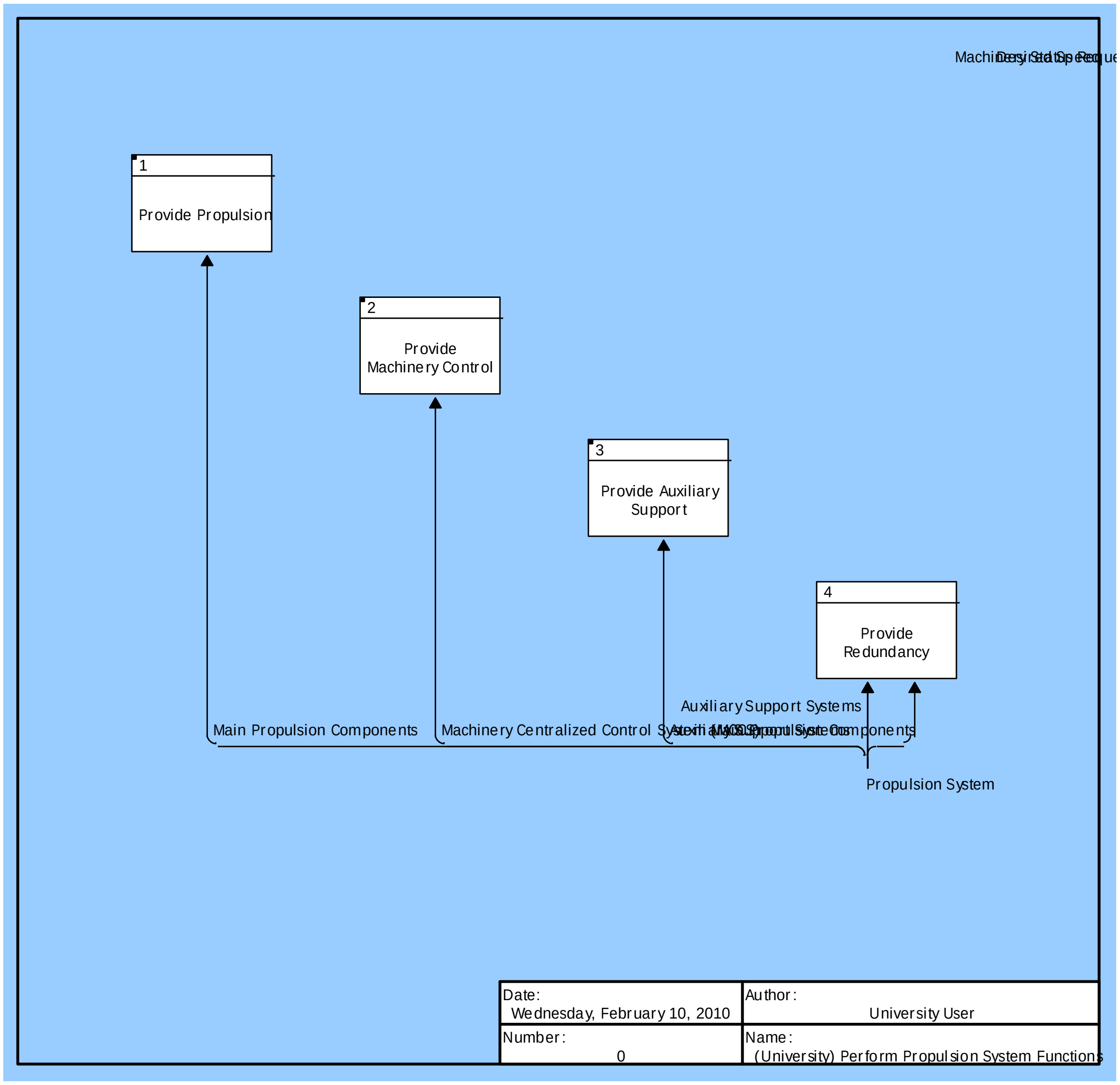


Figure 7 Perform Propulsion System Functions IDEF0 Diagram

7 Functional Behavior Model

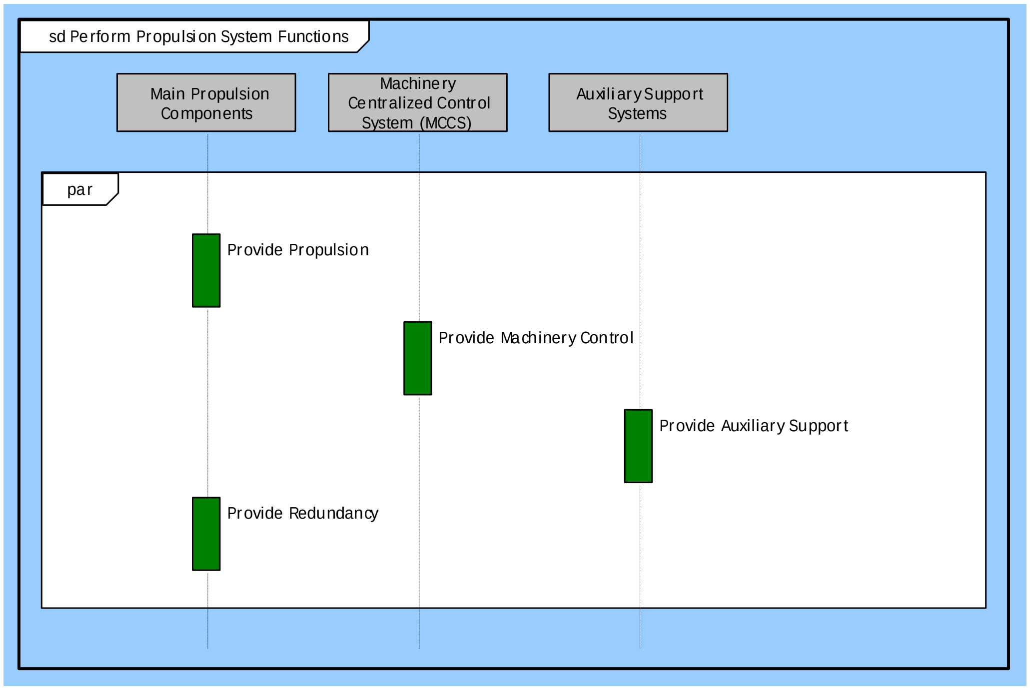


Figure 8 Perform Propulsion System Functions Sequence Diagram

1 Provide Propulsion

Allocated To:

1 Main Propulsion Components

Based On:

REQ.1.2 Sustained Speed

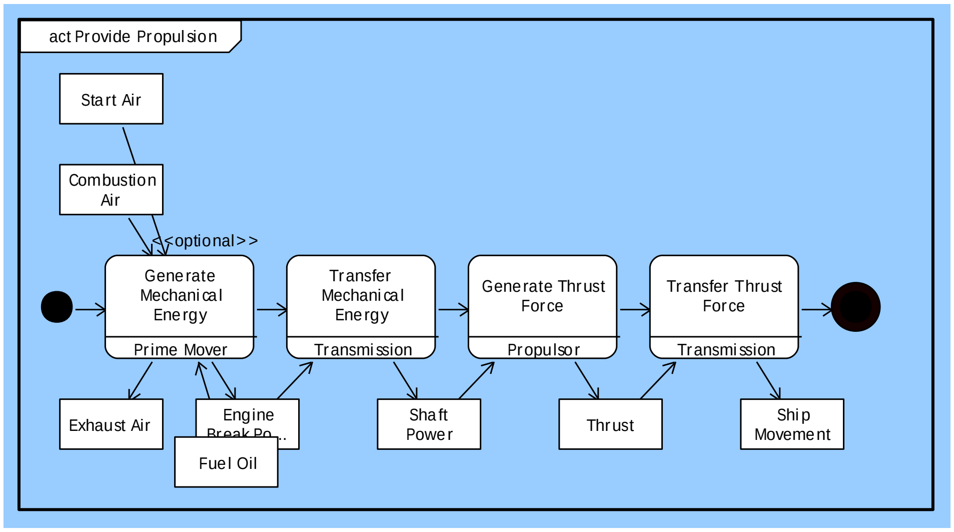


Figure 9 Provide Propulsion Activity Diagram

7 Functional Behavior Model

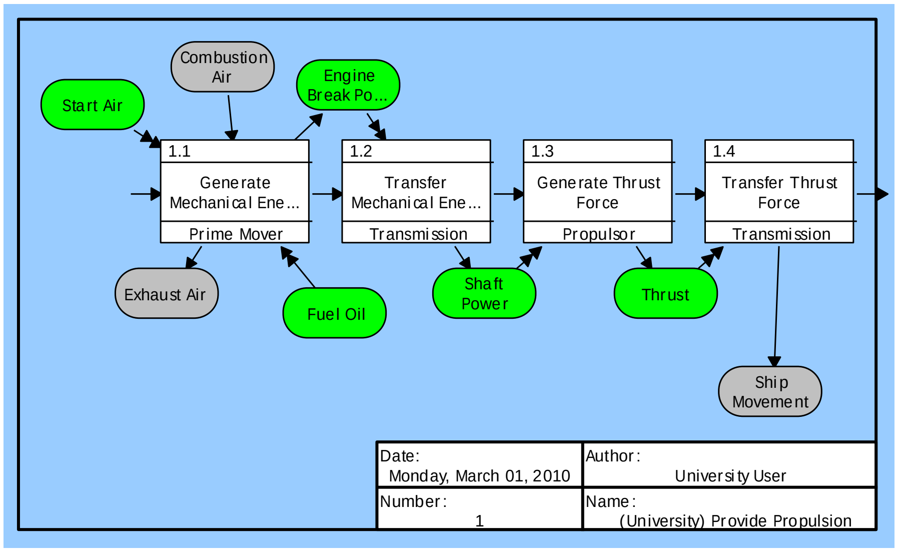


Figure 10 Provide Propulsion Enhanced FFBD

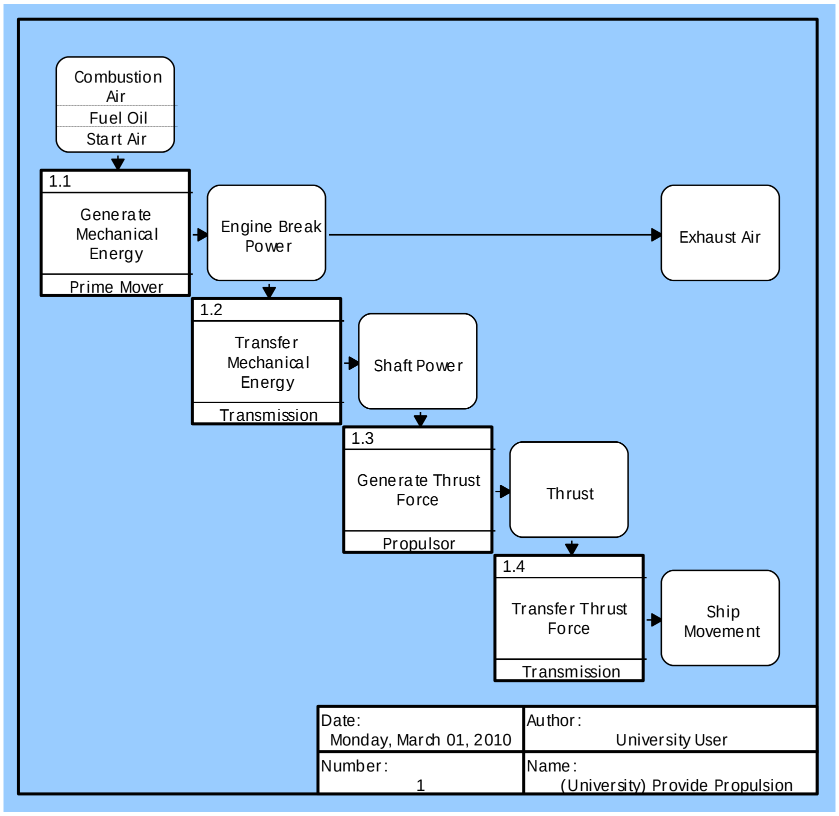


Figure 11 Provide Propulsion N2 Diagram

7 Functional Behavior Model

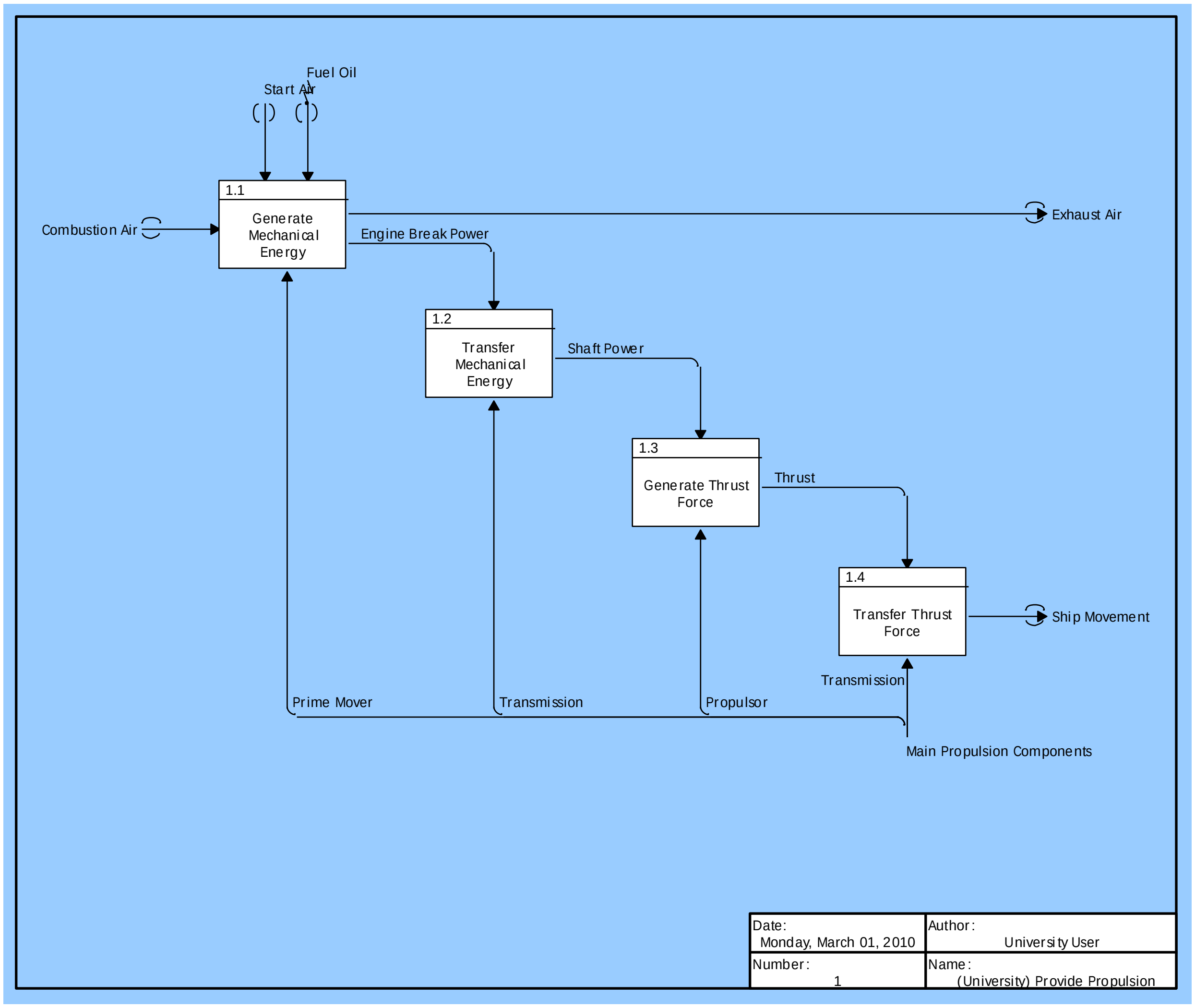


Figure 12 Provide Propulsion IDEF0 Diagram

7 Functional Behavior Model

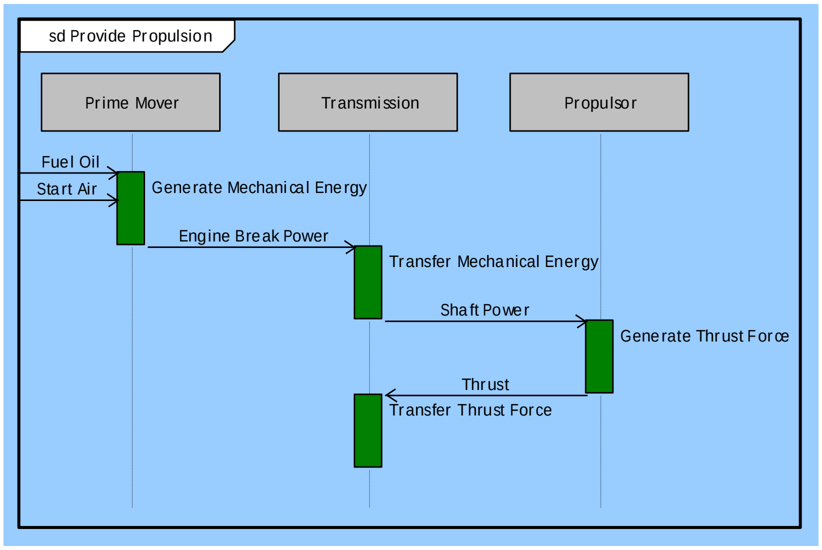


Figure 13 Provide Propulsion Sequence Diagram

1.1 Generate Mechanical Energy

Description:

Generate mechanical energy by converting chemical energy (fuel) into mechanical energy

Allocated To:

1.1 Prime Mover

Based On:

REQ.1.2.1 Main Propulsion Engine Rating
REQ.1.2.3 Light Running Margin
REQ.1.4 Fuel Efficiency

Table 2 1.1 Generate Mechanical Energy Interfacing Items

Interfacing Items	Source / Destination
Combustion Air	Input To: 1.1 Generate Mechanical Energy
Engine Break Power	Triggers Function(s): 1.2 Transfer Mechanical Energy Output From: 1.1 Generate Mechanical Energy

Table 2 1.1 Generate Mechanical Energy Interfacing Items

Interfacing Items	Source / Destination
Exhaust Air	Output From: 1.1 Generate Mechanical Energy
Fuel Oil	Triggers Function(s): 1.1 Generate Mechanical Energy Output From: 3.2 Provide Fuel
Start Air	Triggers Function(s): 1.1 Generate Mechanical Energy Output From: 3.1 Provide Start Air

1.2 Transfer Mechanical Energy

Description:

This is a function of the Transmission System. To transfer mechanical energy generated by the prime mover to the propulsor (or if electric drive...to transfer mechanical energy from propulsion motor to the propulsor)

Allocated To:

1.2 Transmission

Based On:

REQ.1.2.5 Shafting
REQ.8.2 Propulsion Shafting Vibration

Table 3 1.2 Transfer Mechanical Energy Interfacing Items

Interfacing Items	Source / Destination
Engine Break Power	Triggers Function(s): 1.2 Transfer Mechanical Energy Output From: 1.1 Generate Mechanical Energy

Table 3 1.2 Transfer Mechanical Energy Interfacing Items

Interfacing Items	Source / Destination
Shaft Power	Triggers Function(s): 1.3 Generate Thrust Force Output From: 1.2 Transfer Mechanical Energy

1.3 Generate Thrust Force

Description:

Convert rotating mechanical power to translating mechanical power. The thrust force must overcome the resistance *R* of the hull and performance is measured by propulsive efficiency.

Allocated To:

1.3 Propulsor

Based On:

REQ.1.2.4 Propulsor

Table 4 1.3 Generate Thrust Force Interfacing Items

Interfacing Items	Source / Destination
Shaft Power	Triggers Function(s): 1.3 Generate Thrust Force Output From: 1.2 Transfer Mechanical Energy
Thrust	Triggers Function(s): 1.4 Transfer Thrust Force Output From: 1.3 Generate Thrust Force

1.4 Transfer Thrust Force

Description:

Transfer thrust force generated by the propulsor to the ships hull

Allocated To:

1.2 Transmission

Based On:

REQ.1.1.3 Thrust

Table 5 1.4 Transfer Thrust Force Interfacing Items

Interfacing Items	Source / Destination
Ship Movement	Output From: 1.4 Transfer Thrust Force
Thrust	Triggers Function(s): 1.4 Transfer Thrust Force Output From: 1.3 Generate Thrust Force

2 Provide Machinery Control

Allocated To:

2 Machinery Centralized Control System (MCCS)

Based On:

REQ.3.1 Machinery Centralized Control System

REQ.3.1.5 Standards

7 Functional Behavior Model

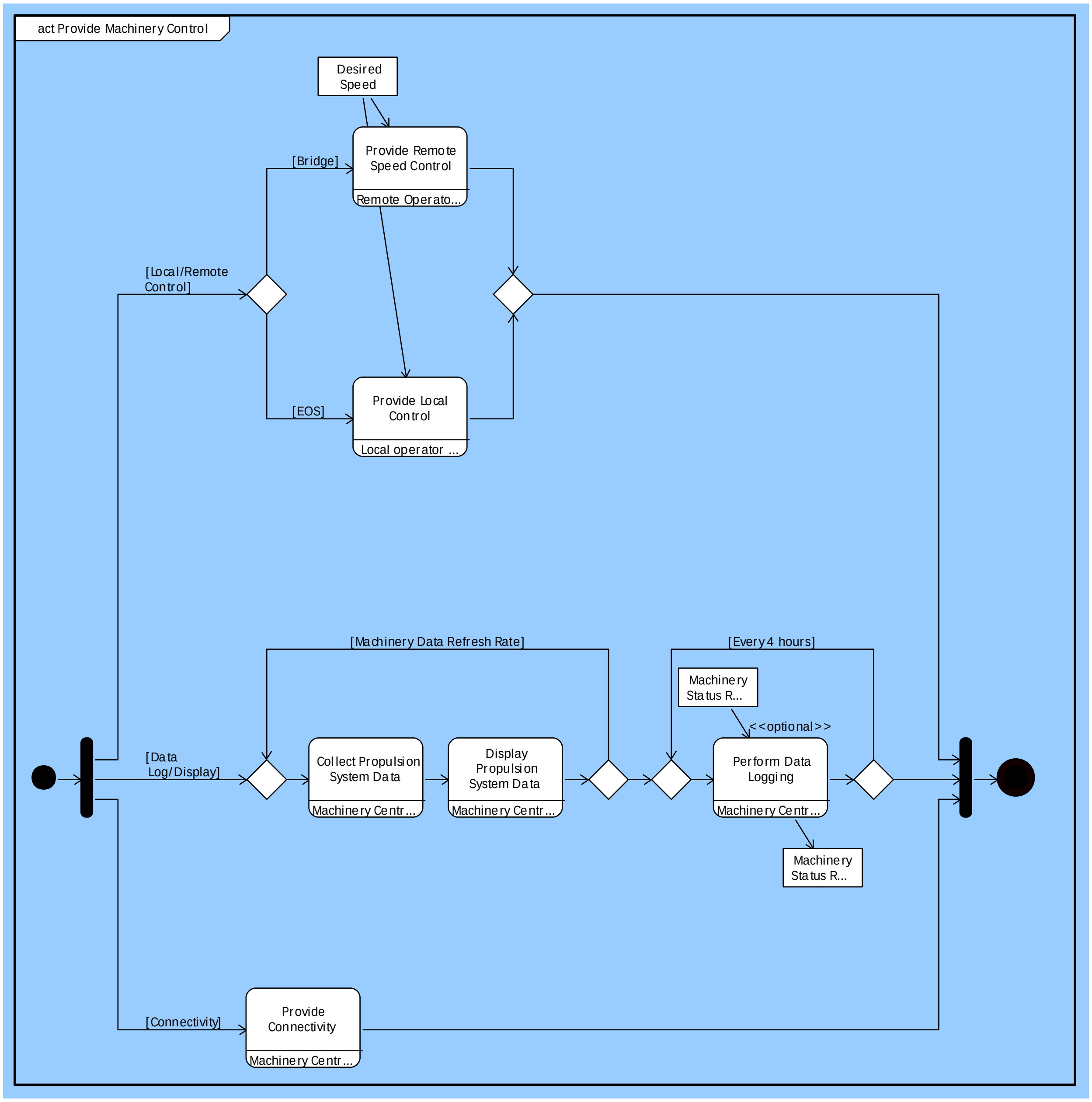


Figure 14 Provide Machinery Control Activity Diagram

7 Functional Behavior Model

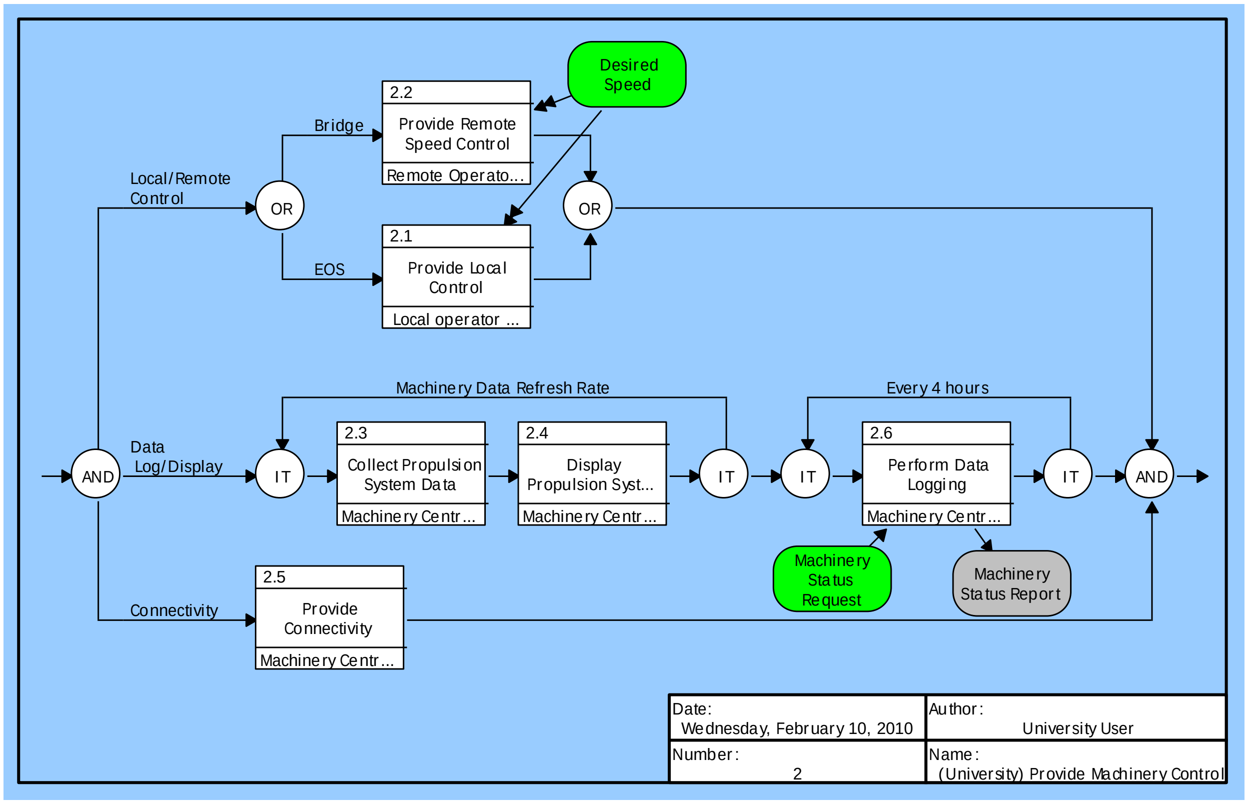


Figure 15 Provide Machinery Control Enhanced FFBD

7 Functional Behavior Model

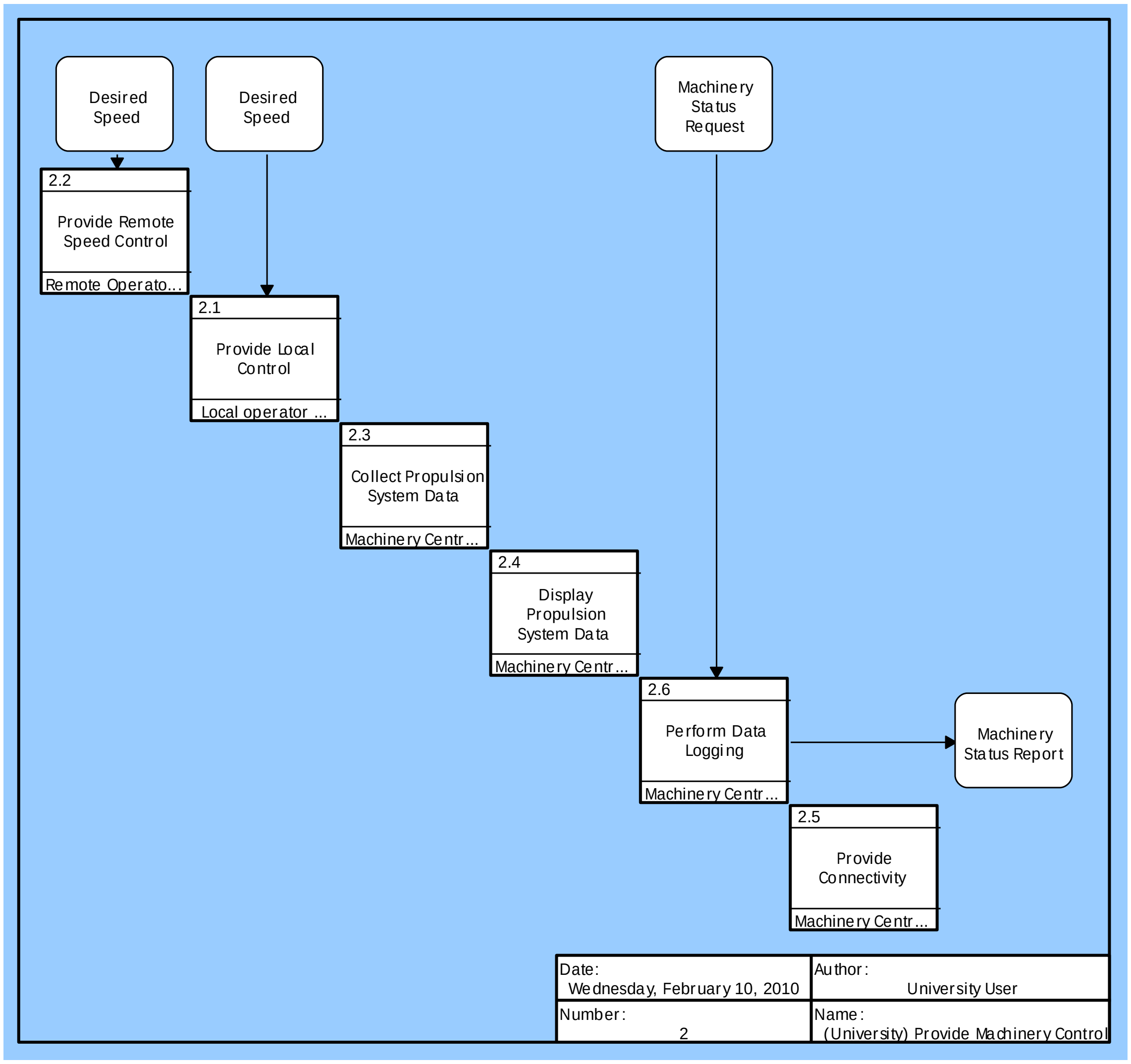


Figure 16 Provide Machinery Control N2 Diagram

7 Functional Behavior Model

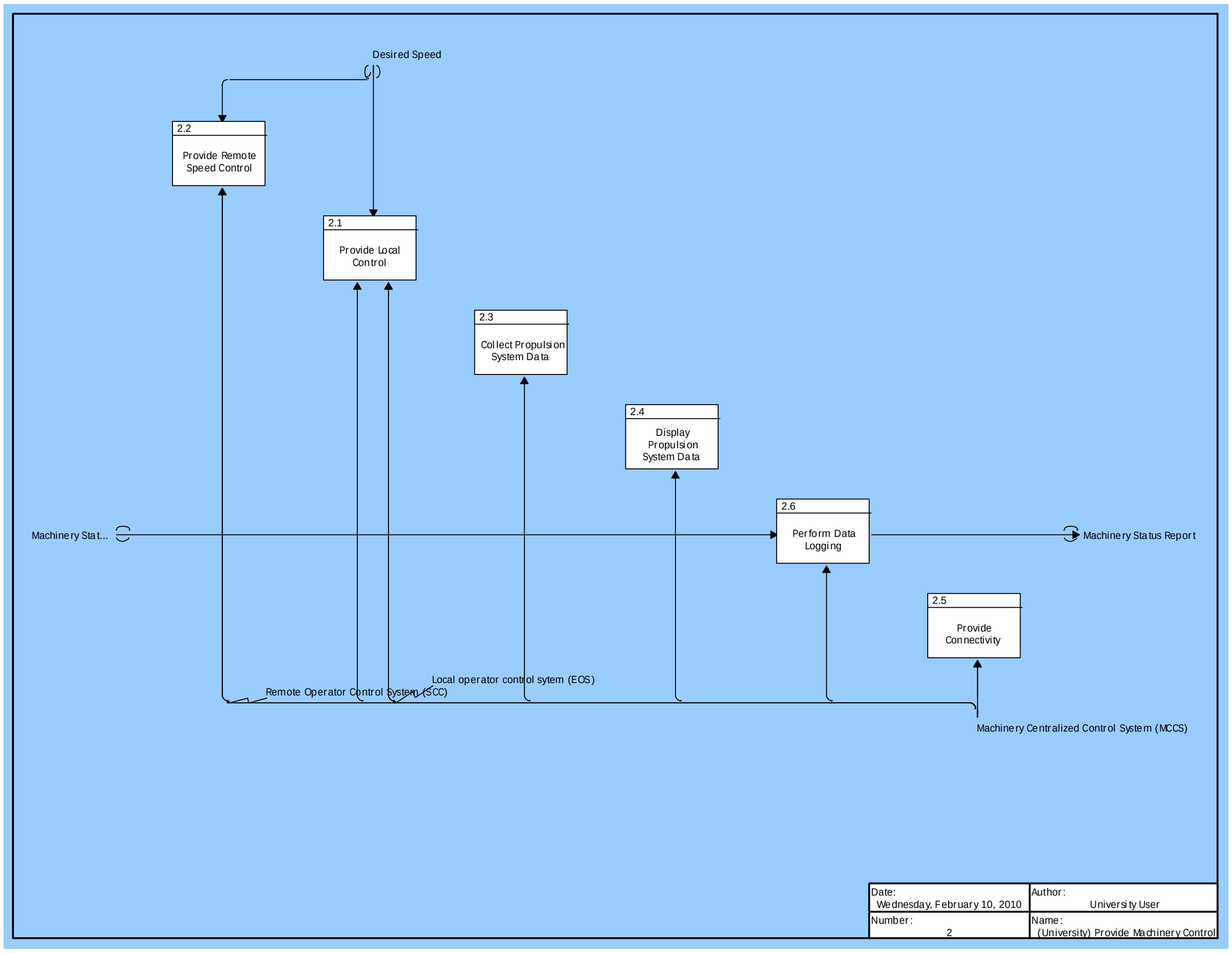


Figure 17 Provide Machinery Control IDEF0 Diagram

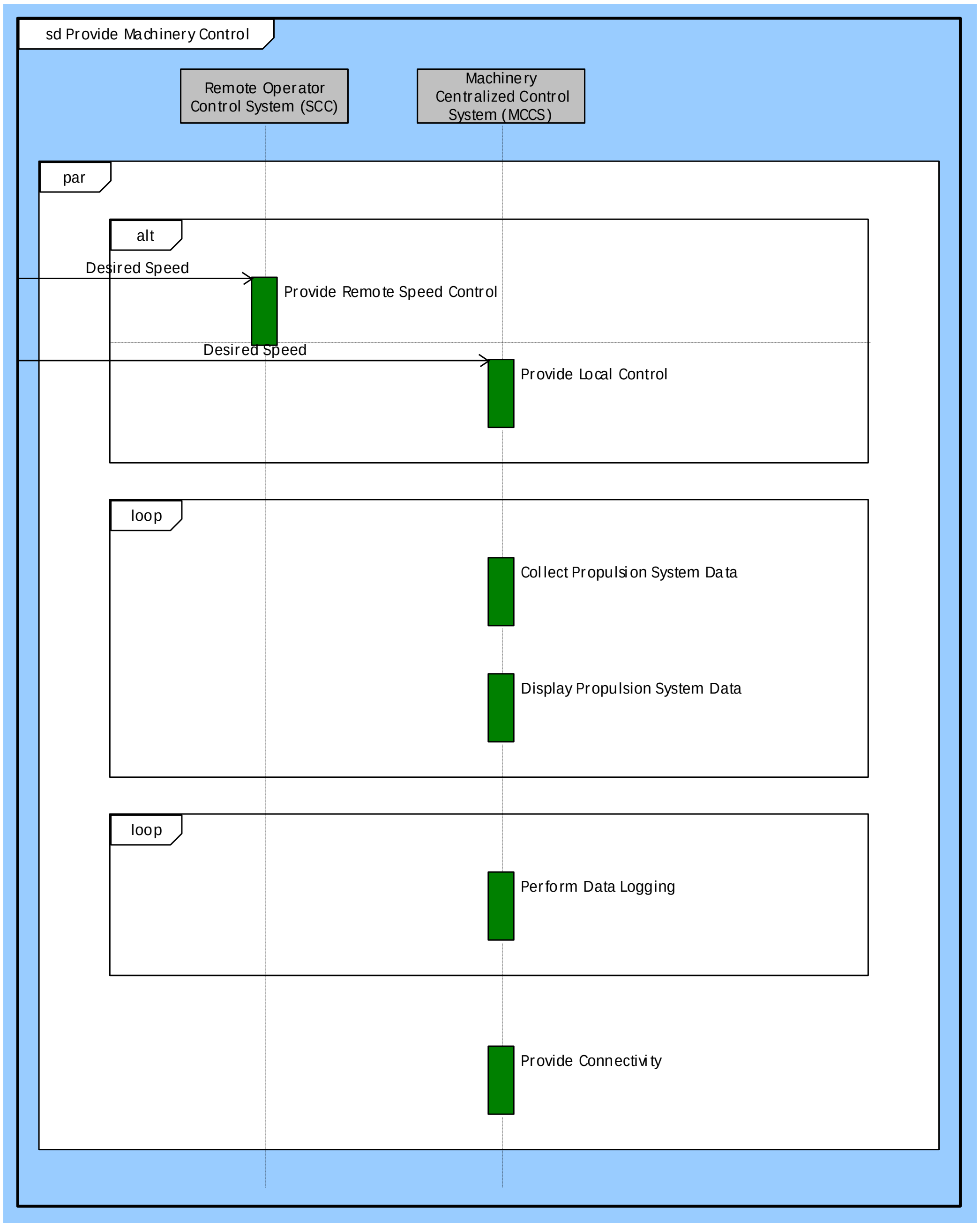


Figure 18 Provide Machinery Control Sequence Diagram

2.1 Provide Local Control

Description:

Provide Local Machinery Control at EOS to include: Provide independent shaft control and propeller control. Provide local start/stop control of prime mover. Provide local clutch engagement. Provide local system override.

Allocated To:

2 Machinery Centralized Control System (MCCS)

2.1 Local operator control sytem (EOS)

Based On:

REQ.3.1 Machinery Centralized Control System

REQ.5 Human Design Integration

Table 6 2.1 Provide Local Control Interfacing Items

Interfacing Items	Source / Destination
Desired Speed	Triggers Function(s): 0 Perform Propulsion System Functions 2.1 Provide Local Control 2.2 Provide Remote Speed Control Output From: C.1 Perform Operator Functions

2.2 Provide Remote Speed Control

Description:

Provide Remote speed control to the Navigation Bridge via the Ship's Control Console to include independent control of either shaft and propeller.

Allocated To:

2.2 Remote Operator Control System (SCC)

Based On:

REQ.3.1 Machinery Centralized Control System

Table 7 2.2 Provide Remote Speed Control Interfacing Items

Interfacing Items	Source / Destination
Desired Speed	Triggers Function(s): 0 Perform Propulsion System Functions 2.1 Provide Local Control

Table 7 2.2 Provide Remote Speed Control Interfacing Items

Interfacing Items	Source / Destination
	2.2 Provide Remote Speed Control Output From: C.1 Perform Operator Functions

2.3 Collect Propulsion System Data

Allocated To:

2 Machinery Centralized Control System (MCCS)

Based On:

REQ.3.1.2 Data Acquisition & Display

2.4 Display Propulsion System Data

Allocated To:

2 Machinery Centralized Control System (MCCS)

Based On:

REQ.3.1.2 Data Acquisition & Display

2.5 Provide Connectivity

Allocated To:

2 Machinery Centralized Control System (MCCS)

Based On:

REQ.3.1.1 Connectivity

2.6 Perform Data Logging

Allocated To:

2 Machinery Centralized Control System (MCCS)

Based On:

REQ.3.1.3 Data Logging

Table 8 2.6 Perform Data Logging Interfacing Items

Interfacing Items	Source / Destination
Machinery Status Report	Output From: 2.6 Perform Data Logging
Machinery Status Request	Input To: 2.6 Perform Data Logging Triggers Function(s): 0 Perform Propulsion System Functions Output From: C.1 Perform Operator Functions

3 Provide Auxiliary Support

Allocated To:

3 Auxiliary Support Systems

Based On:

REQ.1.5 Mobility Support Systems

7 Functional Behavior Model

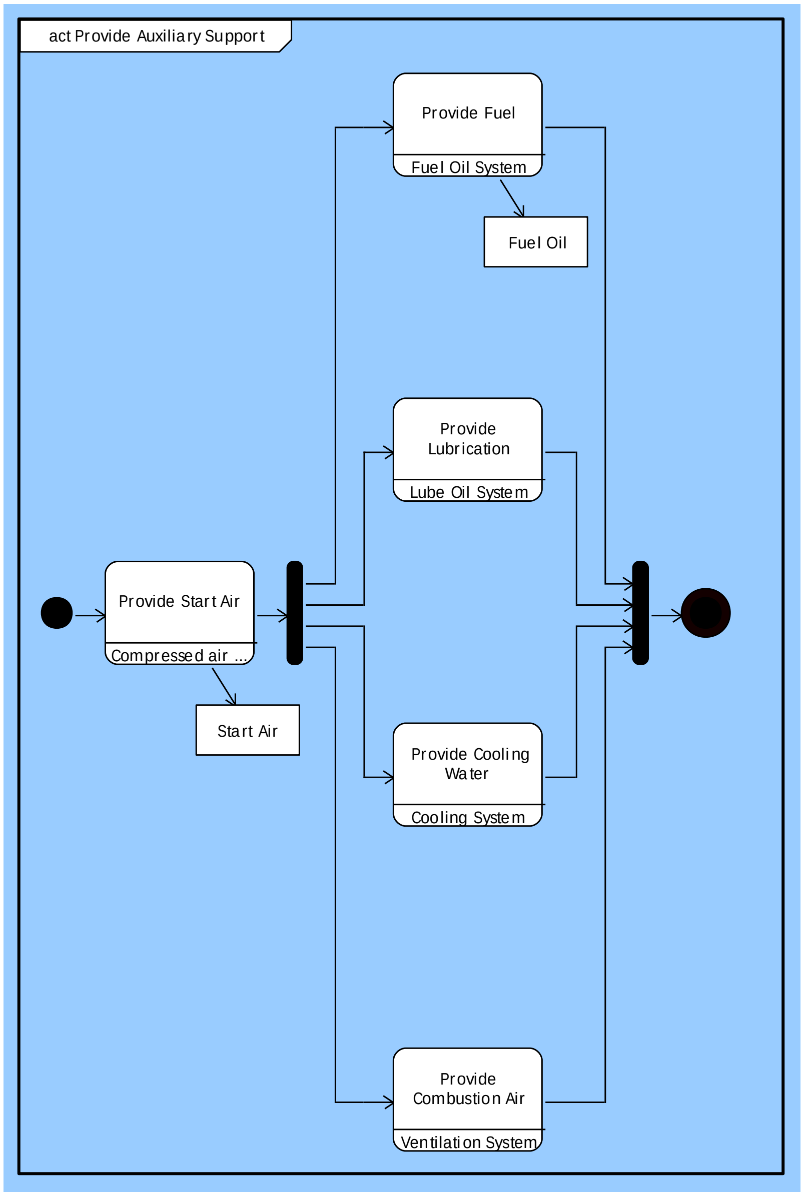


Figure 19 Provide Auxiliary Support Activity Diagram

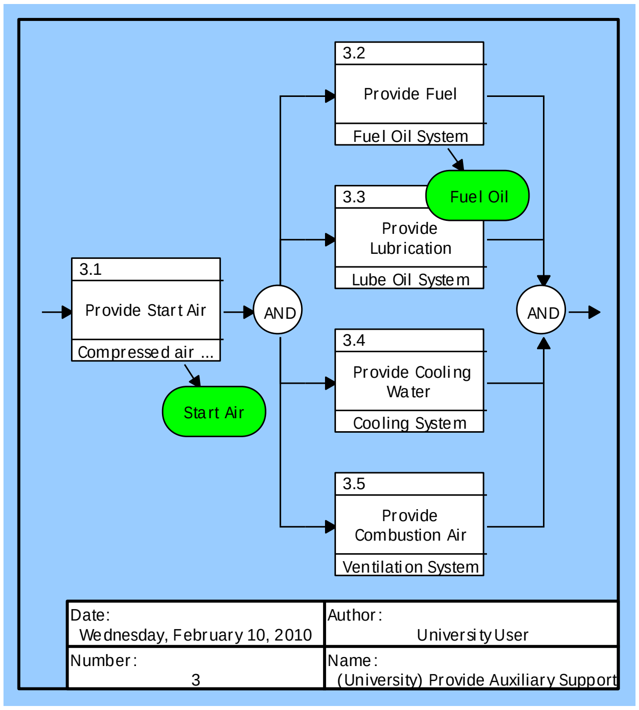


Figure 20 Provide Auxiliary Support Enhanced FFBD

7 Functional Behavior Model

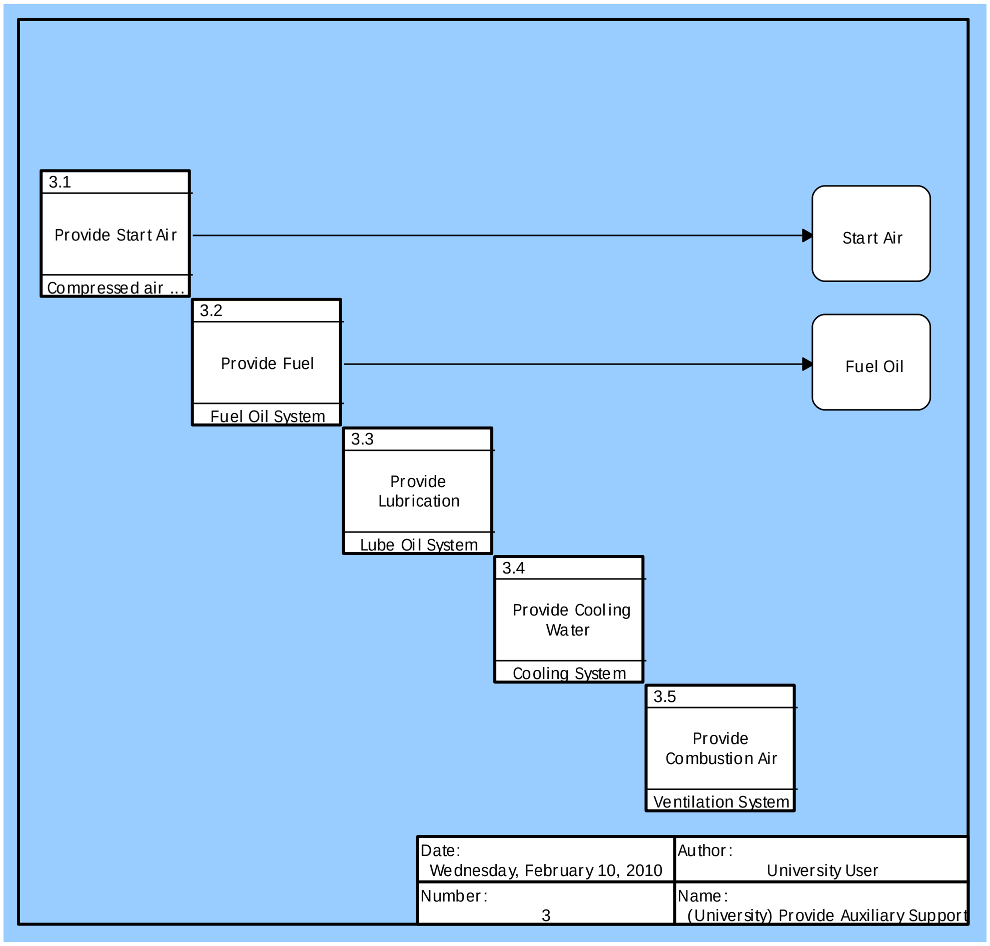


Figure 21 Provide Auxiliary Support N2 Diagram

7 Functional Behavior Model

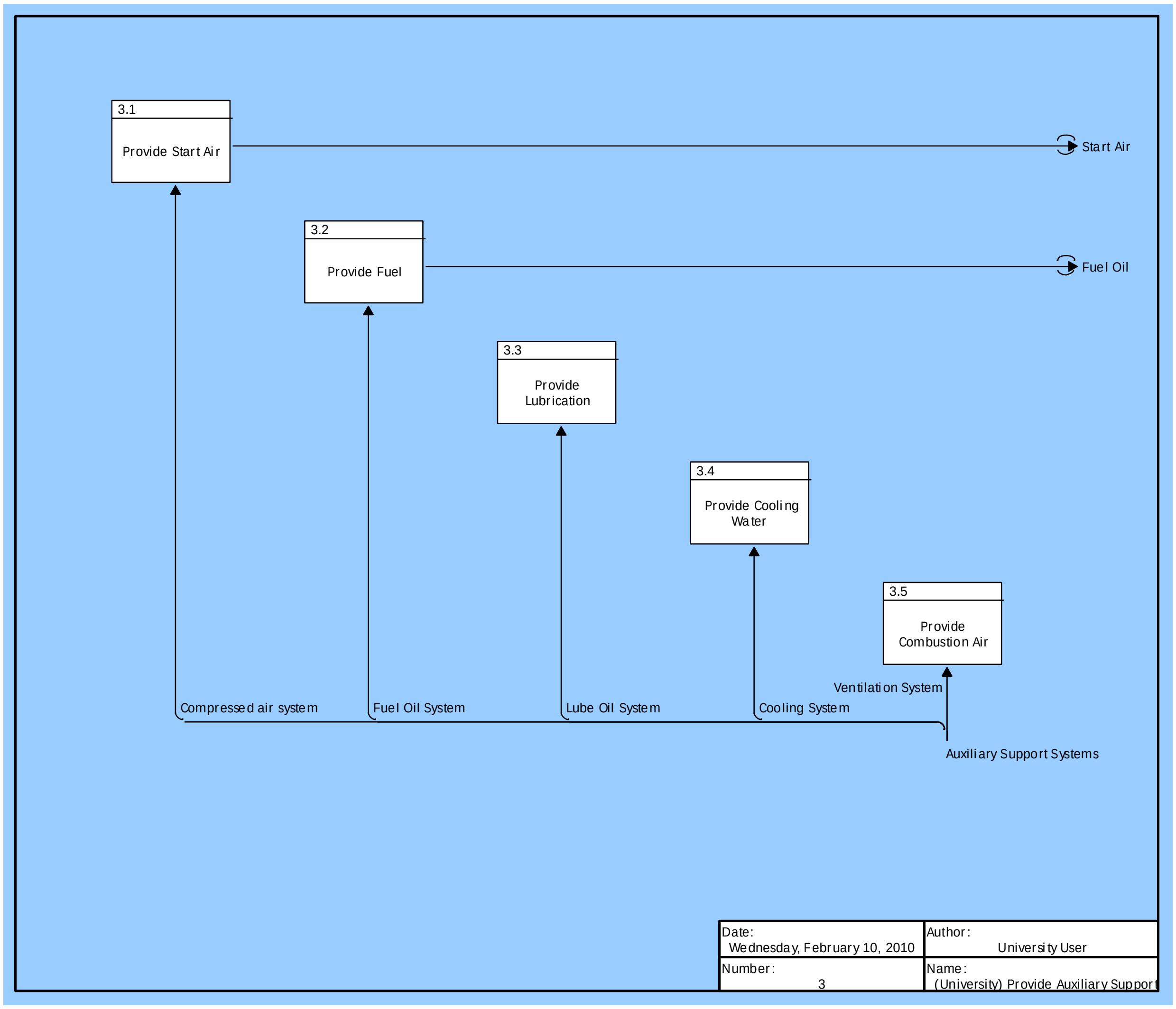


Figure 22 Provide Auxiliary Support IDEF0 Diagram

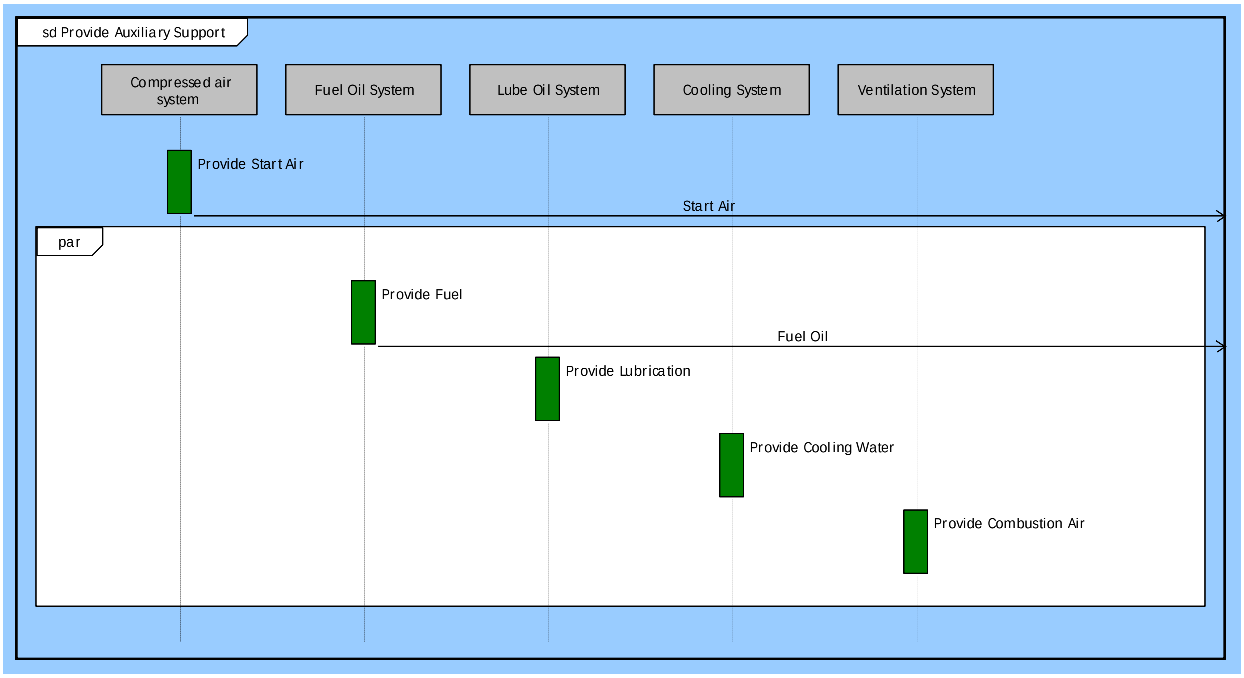


Figure 23 Provide Auxiliary Support Sequence Diagram

3.1 Provide Start Air

Allocated To:

3.2 Compressed air system

Based On:

REQ.1.5.1 Start Air Pressure

Table 9 3.1 Provide Start Air Interfacing Items

Interfacing Items	Source / Destination
Start Air	Triggers Function(s): 1.1 Generate Mechanical Energy Output From: 3.1 Provide Start Air

3.2 Provide Fuel

Allocated To:

3.6 Fuel Oil System

Based On:

REQ.1.3.1 Fuel Tankage

REQ.1.4.2 Fuel

Table 10 3.2 Provide Fuel Interfacing Items

Interfacing Items	Source / Destination
Fuel Oil	Triggers Function(s): 1.1 Generate Mechanical Energy Output From: 3.2 Provide Fuel

3.3 Provide Lubrication

Allocated To:

3.7 Lube Oil System

Based On:

REQ.1.5 Mobility Support Systems

3.4 Provide Cooling Water

Allocated To:

3.3 Cooling System

Based On:

REQ.1.5 Mobility Support Systems

3.5 Provide Combustion Air

Allocated To:

3.5 Ventilation System

Based On:

REQ.1.5 Mobility Support Systems

7 Functional Behavior Model

4 Provide Redundancy

Allocated To:

- 1 Main Propulsion Components
- 3 Auxiliary Support Systems

Based On:

- REQ.2.2 Redundancy and Separation

8 Components

Part I - Hierarchical Component List

- Sys.1 Propulsion System
  - 1 Main Propulsion Components
    - 1.1 Prime Mover
    - 1.2 Transmission
      - 1.2.1 Line Shaft Bearing
      - 1.2.2 Main Reduction Gear
      - 1.2.3 Shafts
      - 1.2.4 Clutch
      - 1.2.5 Thrust Bearing
    - 1.3 Propulsor
      - 1.3.1 Controlable Pitch Propeller
  - 2 Machinery Centralized Control System (MCCS)
    - 2.1 Local operator control sytem (EOS)
    - 2.2 Remote Operator Control System (SCC)
  - 3 Auxiliary Support Systems
    - 3.1 Hydraulic Oil System
    - 3.2 Compressed air system
    - 3.3 Cooling System
    - 3.4 Exhaust Gas System
    - 3.5 Ventilation System
    - 3.6 Fuel Oil System
      - 3.6.1 Fuel Oil Cleaning System
      - 3.6.2 Fuel Oil Service Tanks
      - 3.6.3 Fuel Oil Storage Tanks
      - 3.6.4 Fuel Oil Transfer Pumps
    - 3.7 Lube Oil System
      - 3.7.1 Lube Oil Cleaning System

        - 3.7.2 Lube Oil Pumps
        - 3.7.3 Lube Oil Tanks
        - 3.7.4 Oily Waste Pumps

Part II - Component Definitions

Sys.1 Propulsion System

Description:

The propulsion system is one of the most important systems onboard a marine vessel. The function of the propulsion system is to generate thrust, which enables the ship to move at the desired speed. The propulsion system consists of three main components: prime mover, transmission, and propulsor.

Type: System

Built In Higher-Level Component(s):

C Propulsion System Context

Built From Lower-Level Component(s):

1 Main Propulsion Components
2 Machinery Centralized Control System (MCCS)
3 Auxiliary Support Systems

Joined Through Logical Interface:

INT.1 Intakes/Exhaust
INT.2 Fuel Storage Tank
INT.3 Engineering Operating Station (EOS)
INT.4 Propulsor/Water Interface
INT.5 Ship's Control Console (Bridge)
INT.6 Thrust Bearing/Hull Interface

Connected through Physical Link(s):

Sys-Atmosphere
Sys-Fuel
Sys-Hull
Sys-Local Operator
Sys-Remote Operator

Sys-Water

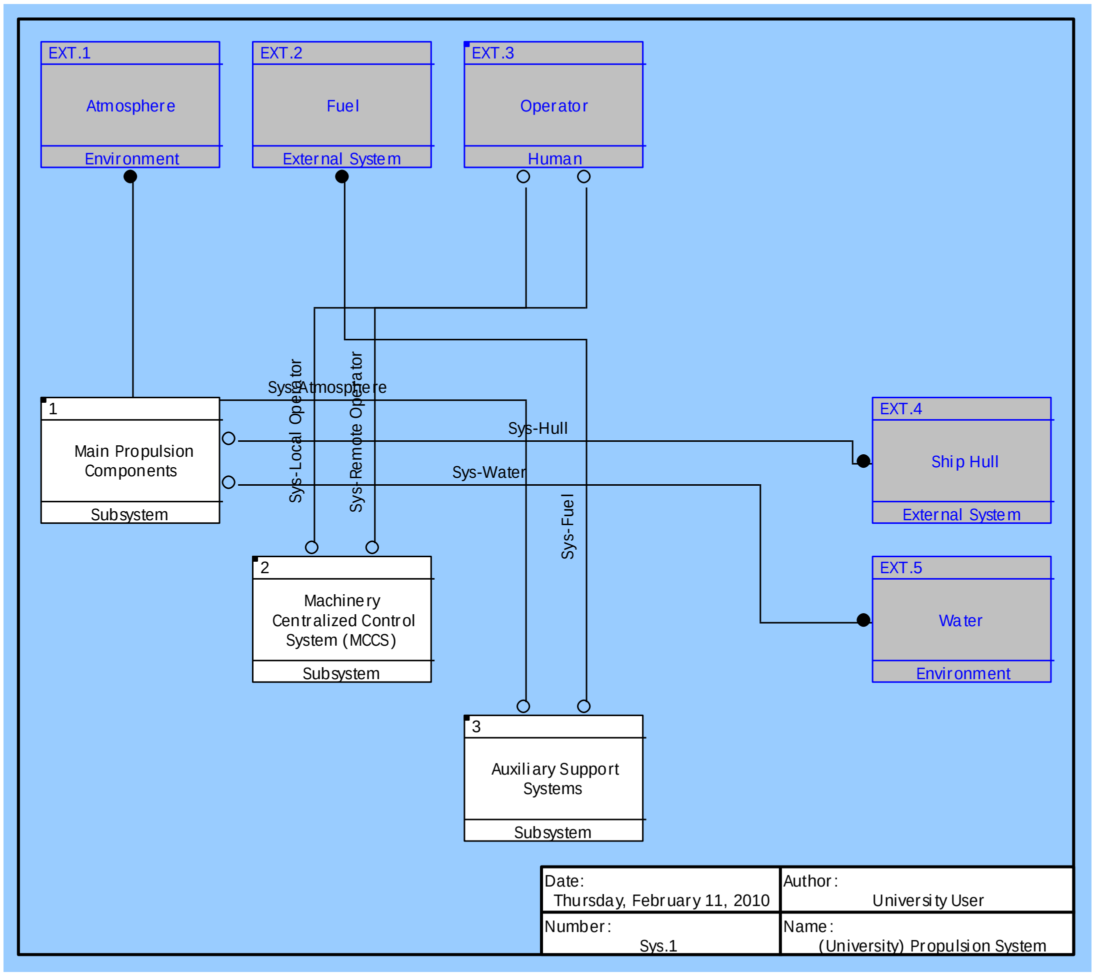


Figure 24 Propulsion System Subcomponent Links

Performs Function(s):

0 Perform Propulsion System Functions

Specified By:

REQ.4 Lifecycle Cost

REQ.6.1 Availability

REQ.6.2 Reliability w/ repair

REQ.6.3 Reliability w/out repair

REQ.7 Service Life

Source Documents:

DOC.1 Performance Specification Document

8 Components

1 Main Propulsion Components

Type: Subsystem

Built In Higher-Level Component(s):
  Sys.1 Propulsion System

Built From Lower-Level Component(s):
  1.1 Prime Mover
  1.2 Transmission
  1.3 Propulsor

Joined Through Logical Interface:
  INT.4 Propulsor/Water Interface
  INT.6 Thrust Bearing/Hull Interface

Connected through Physical Link(s):
  Sys-Hull
  Sys-Water

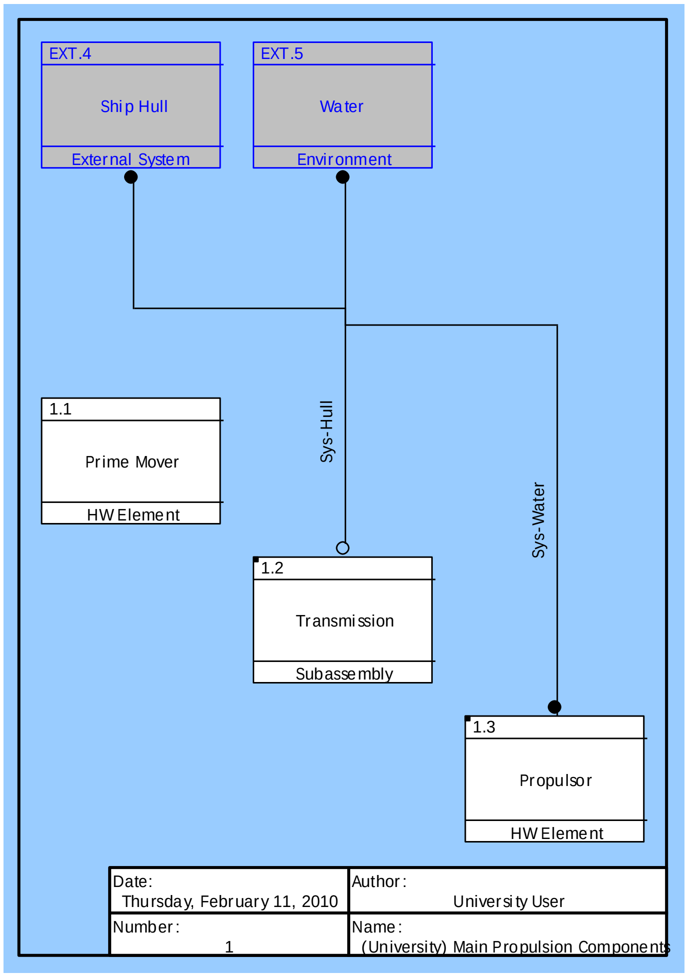


Figure 25 Main Propulsion Components Subcomponent Links

Performs Function(s):

1 Provide Propulsion

4 Provide Redundancy

1.1 Prime Mover

Description:

Diesel Engine, Gas Turbine, or Steam plant

Type: HW Element

Built In Higher-Level Component(s):

1 Main Propulsion Components

Performs Function(s):

1.1 Generate Mechanical Energy

Specified By:

REQ.1.4.1 Prime Mover

1.2 Transmission

Type: Subassembly

Built In Higher-Level Component(s):

1 Main Propulsion Components

Built From Lower-Level Component(s):

1.2.1 Line Shaft Bearing
1.2.2 Main Reduction Gear
1.2.3 Shafts
1.2.4 Clutch
1.2.5 Thrust Bearing

Joined Through Logical Interface:

INT.6 Thrust Bearing/Hull Interface

Connected through Physical Link(s):

Sys-Hull

8 Components

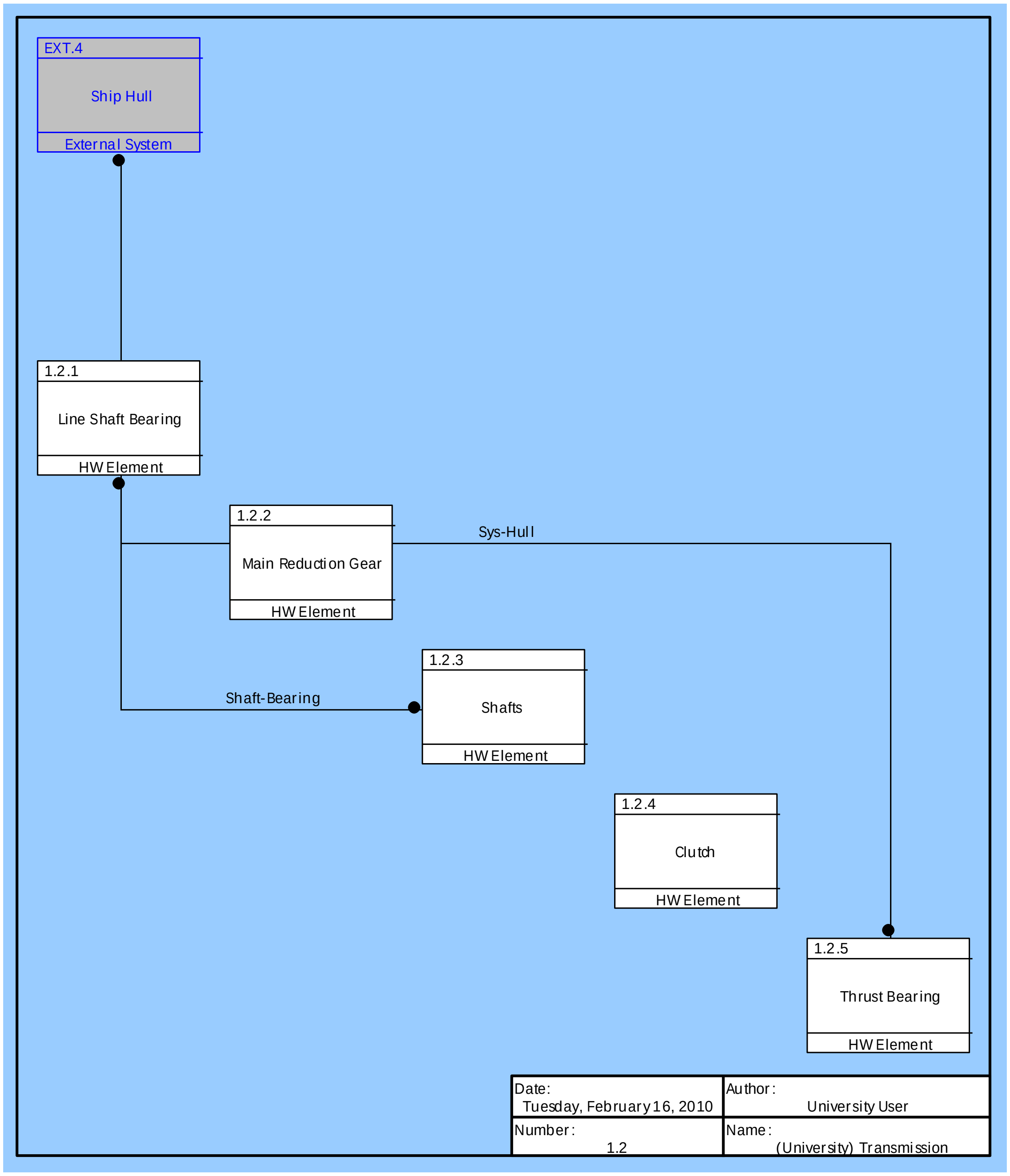


Figure 26 Transmission Subcomponent Links

Performs Function(s):

1.2 Transfer Mechanical Energy

1.4 Transfer Thrust Force

Specified By:

REQ.1.2.2 Mechanical Drive

Generates Issue(s):

IPS-Mechanical Drive Trade Study

1.2.1 Line Shaft Bearing

Type: HW Element

Built In Higher-Level Component(s):
1.2 Transmission

Connected to Physical Link(s):
Shaft-Bearing

1.2.2 Main Reduction Gear

Type: HW Element

Built In Higher-Level Component(s):
1.2 Transmission

1.2.3 Shafts

Type: HW Element

Built In Higher-Level Component(s):
1.2 Transmission

Connected to Physical Link(s):
Shaft-Bearing

1.2.4 Clutch

Type: HW Element

Built In Higher-Level Component(s):
1.2 Transmission

1.2.5 Thrust Bearing

Type: HW Element

Built In Higher-Level Component(s):
1.2 Transmission

Joined To Logical Interface:

INT.6 Thrust Bearing/Hull Interface

Connected to Physical Link(s):

Sys-Hull

1.3 Propulsor

There will be a trade study performed to determine the optimum propulsor design for the selected propulsion drive (mechanical vs. integrated electric)

Type: HW Element

Built In Higher-Level Component(s):

1 Main Propulsion Components

Built From Lower-Level Component(s):

1.3.1 Controlable Pitch Propeller

Joined To Logical Interface:

INT.4 Propulsor/Water Interface

Connected to Physical Link(s):

Sys-Water

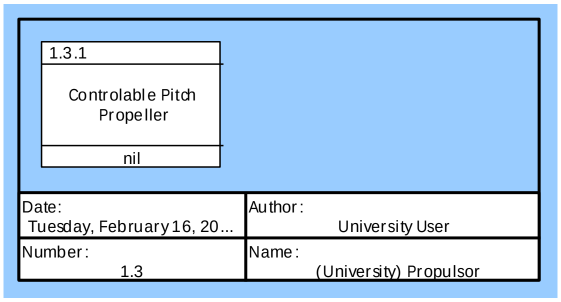


Figure 27 Propulsor Subcomponent Links

Performs Function(s):

1.3 Generate Thrust Force

Specified By:

REQ.1.2.4.1 Controllable Pitch Propeller

Generates Issue(s):

Propulsor Trade Study

1.3.1 Controlable Pitch Propeller

Description:

Propulsor design option

Built In Higher-Level Component(s):

1.3 Propulsor

2 Machinery Centralized Control System (MCCS)

Type: Subsystem

Built In Higher-Level Component(s):

Sys.1 Propulsion System

Built From Lower-Level Component(s):

2.1 Local operator control sytem (EOS)
2.2 Remote Operator Control System (SCC)

Joined Through Logical Interface:

INT.3 Engineering Operating Station (EOS)
INT.5 Ship's Control Console (Bridge)

Connected through Physical Link(s):

Sys-Local Operator
Sys-Remote Operator

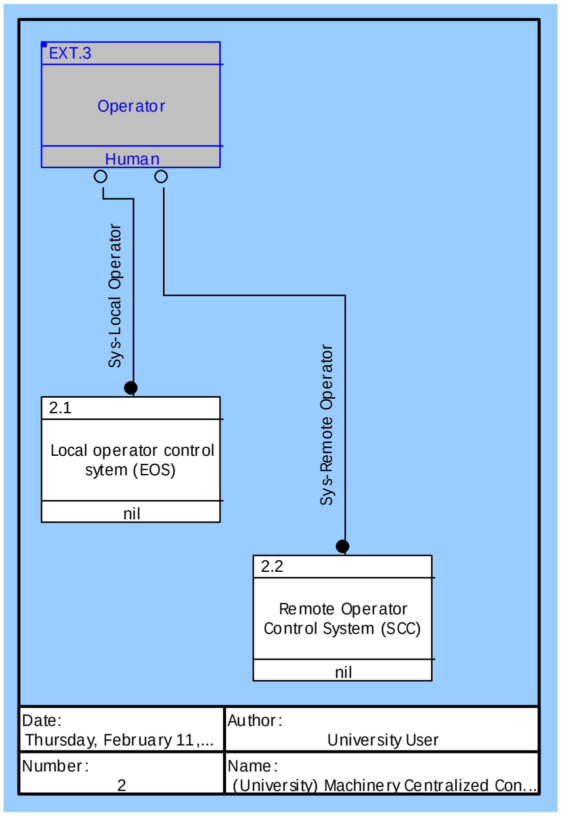


Figure 28 Machinery Centralized Control System (MCCS) Subcomponent Links

Performs Function(s):

- 2 Provide Machinery Control
- 2.1 Provide Local Control
- 2.3 Collect Propulsion System Data
- 2.4 Display Propulsion System Data
- 2.5 Provide Connectivity
- 2.6 Perform Data Logging

2.1 Local operator control sytem (EOS)

Built In Higher-Level Component(s):

2 Machinery Centralized Control System (MCCS)

Joined To Logical Interface:

INT.3 Engineering Operating Station (EOS)

Connected to Physical Link(s):

Sys-Local Operator

Performs Function(s):

2.1 Provide Local Control

2.2 Remote Operator Control System (SCC)

Built In Higher-Level Component(s):

2 Machinery Centralized Control System (MCCS)

Joined To Logical Interface:

INT.5 Ship's Control Console (Bridge)

Connected to Physical Link(s):

Sys-Remote Operator

Performs Function(s):

2.2 Provide Remote Speed Control

3 Auxiliary Support Systems

Type: Subsystem

Built In Higher-Level Component(s):

Sys.1 Propulsion System

Built From Lower-Level Component(s):

3.1 Hydraulic Oil System
3.2 Compressed air system
3.3 Cooling System
3.4 Exhaust Gas System
3.5 Ventilation System
3.6 Fuel Oil System
3.7 Lube Oil System

Joined Through Logical Interface:

INT.1 Intakes/Exhaust
INT.2 Fuel Storage Tank

Connected through Physical Link(s):

Sys-Atmosphere
Sys-Fuel

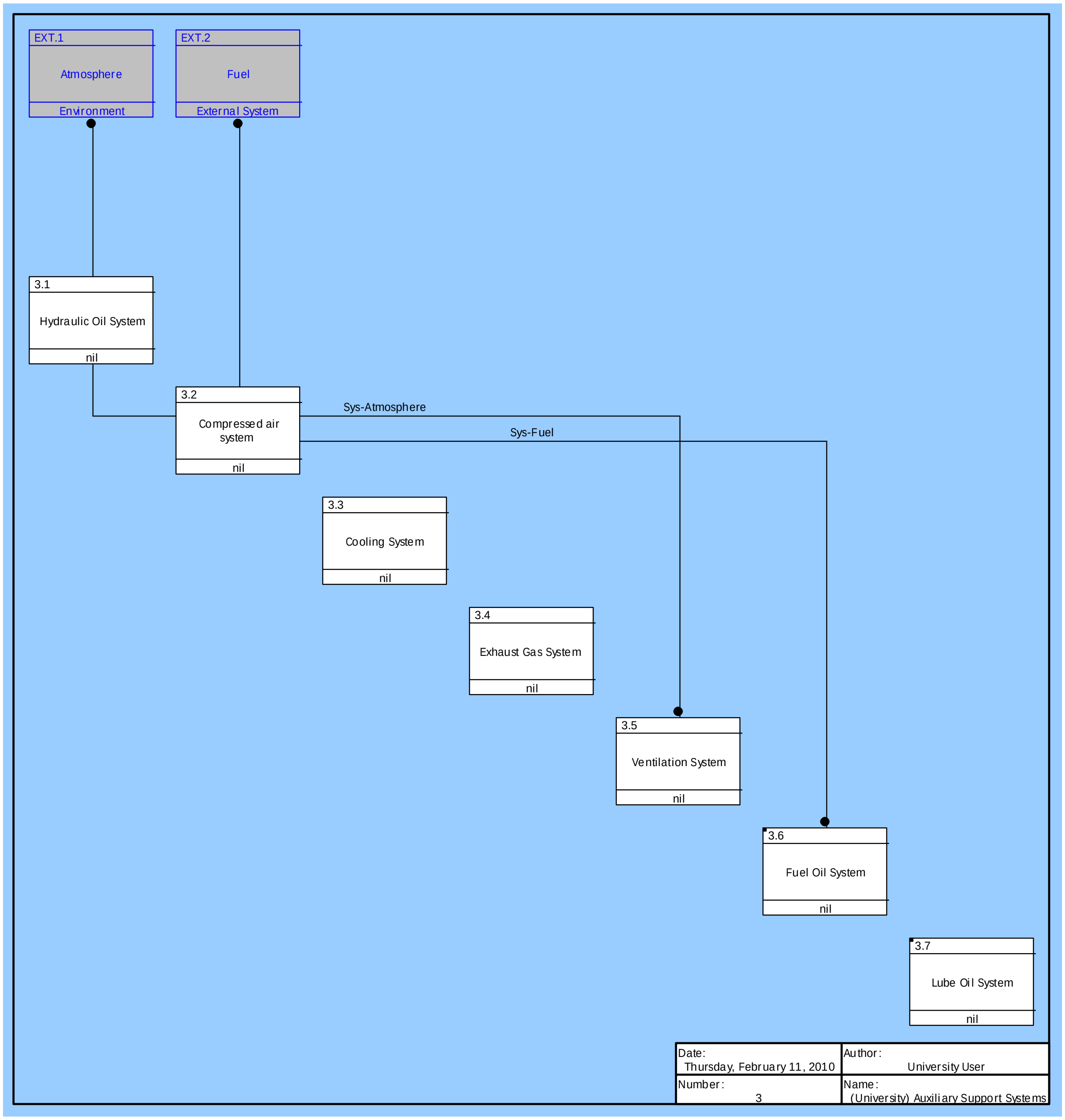


Figure 29 Auxiliary Support Systems Subcomponent Links

Performs Function(s):

3 Provide Auxiliary Support

4 Provide Redundancy

8 Components

3.1 Hydraulic Oil System

Built In Higher-Level Component(s):

3 Auxiliary Support Systems

3.2 Compressed air system

Description:

Starting air for the prime mover is usually compressed air from a high pressure air compressor.

Built In Higher-Level Component(s):

3 Auxiliary Support Systems

Performs Function(s):

3.1 Provide Start Air

3.3 Cooling System

Built In Higher-Level Component(s):

3 Auxiliary Support Systems

Performs Function(s):

3.4 Provide Cooling Water

3.4 Exhaust Gas System

Built In Higher-Level Component(s):

3 Auxiliary Support Systems

3.5 Ventilation System

Description:

Supplies the prime move with combustion air

Built In Higher-Level Component(s):

3 Auxiliary Support Systems

Joined To Logical Interface:

INT.1 Intakes/Exhaust

Connected to Physical Link(s):

Sys-Atmosphere

Performs Function(s):

3.5 Provide Combustion Air

3.6 Fuel Oil System

Description:

Includes Fuel Oil Transfer pumps & Fuel Oil Storage/Service Tanks

Built In Higher-Level Component(s):

3 Auxiliary Support Systems

Built From Lower-Level Component(s):

3.6.1 Fuel Oil Cleaning System
3.6.2 Fuel Oil Service Tanks
3.6.3 Fuel Oil Storage Tanks
3.6.4 Fuel Oil Transfer Pumps

Joined To Logical Interface:

INT.2 Fuel Storage Tank

Connected to Physical Link(s):

Sys-Fuel

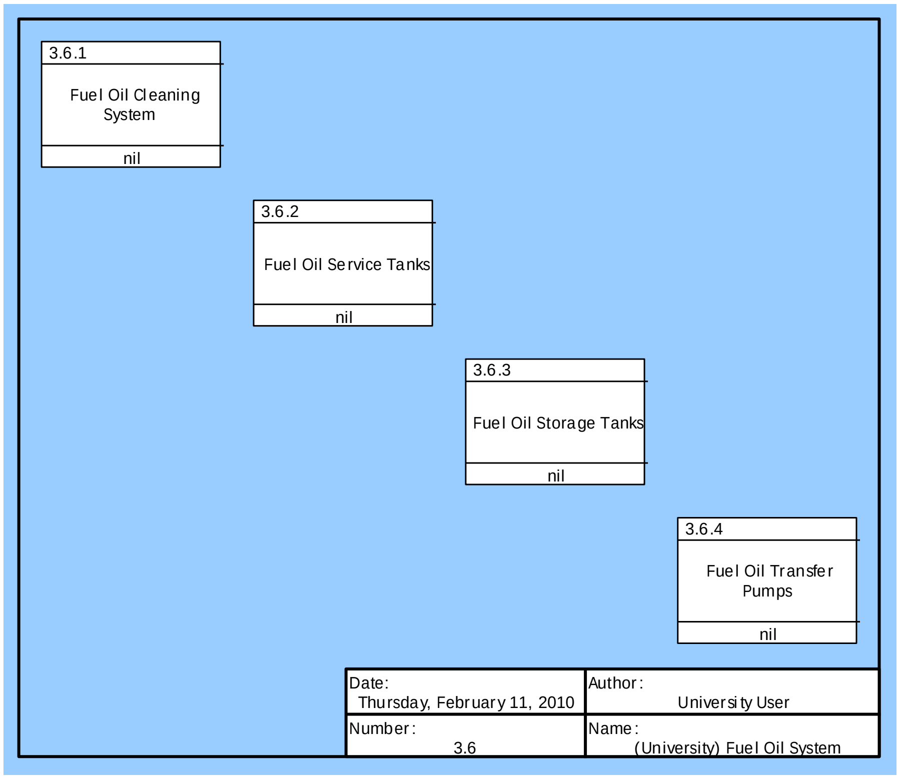


Figure 30 Fuel Oil System Subcomponent Links

Performs Function(s):

3.2 Provide Fuel

3.6.1 Fuel Oil Cleaning System

Built In Higher-Level Component(s):

3.6 Fuel Oil System

3.6.2 Fuel Oil Service Tanks

Built In Higher-Level Component(s):

3.6 Fuel Oil System

3.6.3 Fuel Oil Storage Tanks

Built In Higher-Level Component(s):

3.6 Fuel Oil System

3.6.4 Fuel Oil Transfer Pumps

Built In Higher-Level Component(s):

3.6 Fuel Oil System

3.7 Lube Oil System

Built In Higher-Level Component(s):

3 Auxiliary Support Systems

Built From Lower-Level Component(s):

3.7.1 Lube Oil Cleaning System
3.7.2 Lube Oil Pumps
3.7.3 Lube Oil Tanks
3.7.4 Oily Waste Pumps

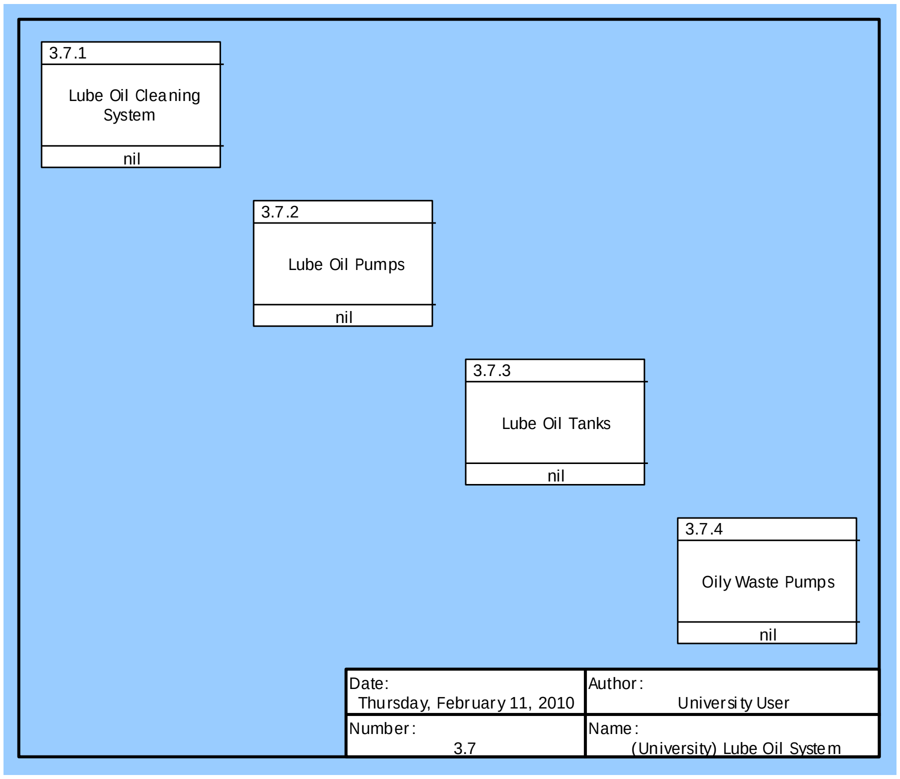


Figure 31 Lube Oil System Subcomponent Links

Performs Function(s):

3.3 Provide Lubrication

3.7.1 Lube Oil Cleaning System

Built In Higher-Level Component(s):

3.7 Lube Oil System

3.7.2 Lube Oil Pumps

Built In Higher-Level Component(s):

3.7 Lube Oil System

3.7.3 Lube Oil Tanks

Built In Higher-Level Component(s):

3.7 Lube Oil System

3.7.4 Oily Waste Pumps

Built In Higher-Level Component(s):

3.7 Lube Oil System

9 Interfaces

Part I - Derived Functional Interfaces

Table 11 Sys.1 Propulsion System External I/O

Functions	Interface Items	Interfacing Elements
0 Perform Propulsion System Functions	← Desired Speed	C.1 Perform Operator Functions EXT.3 Operator
	← Machinery Status Request	C.1 Perform Operator Functions EXT.3 Operator

Table 12 1.1 Prime Mover External I/O

Functions	Interface Items	Interfacing Elements
1.1 Generate Mechanical Energy	→ Engine Break Power	1.2 Transfer Mechanical Energy 1.2 Transmission
	← Fuel Oil	3.2 Provide Fuel 3.6 Fuel Oil System
	← Start Air	3.1 Provide Start Air 3.2 Compressed air system

Table 13 1.2 Transmission External I/O

Functions	Interface Items	Interfacing Elements
1.2 Transfer Mechanical Energy	→ Shaft Power	1.3 Generate Thrust Force 1.3 Propulsor
	← Engine Break Power	1.1 Generate Mechanical Energy 1.1 Prime Mover
1.4 Transfer Thrust Force	← Thrust	1.3 Generate Thrust Force 1.3 Propulsor

9 Interfaces

Table 14 1.3 Propulsor External I/O

Functions	Interface Items	Interfacing Elements
1.3 Generate Thrust Force	→ Thrust	1.4 Transfer Thrust Force 1.2 Transmission
	← Shaft Power	1.2 Transfer Mechanical Energy 1.2 Transmission

Table 15 2 Machinery Centralized Control System (MCCS) External I/O

Functions	Interface Items	Interfacing Elements
2.1 Provide Local Control	← Desired Speed	C.1 Perform Operator Functions EXT.3 Operator
2.6 Perform Data Logging	← Machinery Status Request	C.1 Perform Operator Functions EXT.3 Operator

Table 16 2.1 Local operator control sytem (EOS) External I/O

Functions	Interface Items	Interfacing Elements
2.1 Provide Local Control	← Desired Speed	C.1 Perform Operator Functions EXT.3 Operator

Table 17 2.2 Remote Operator Control System (SCC) External I/O

Functions	Interface Items	Interfacing Elements
2.2 Provide Remote Speed Control	← Desired Speed	C.1 Perform Operator Functions EXT.3 Operator

Table 18 3.2 Compressed air system External I/O

Functions	Interface Items	Interfacing Elements
3.1 Provide Start Air	→ Start Air	1.1 Generate Mechanical Energy 1.1 Prime Mover

9 Interfaces

Table 19 3.6 Fuel Oil System External I/O

Functions	Interface Items	Interfacing Elements
3.2 Provide Fuel	→ Fuel Oil	1.1 Generate Mechanical Energy 1.1 Prime Mover

Part II - Logical Interfaces

INT.1 Intakes/Exhaust

Physical Links:
  Sys-Atmosphere

Connecting Elements:
- 3.5 Ventilation System
  - 3 Auxiliary Support Systems
    - Sys.1 Propulsion System
- EXT.1 Atmosphere

INT.2 Fuel Storage Tank

Physical Links:
  Sys-Fuel

Connecting Elements:
- 3.6 Fuel Oil System
  - 3 Auxiliary Support Systems
    - Sys.1 Propulsion System
- EXT.2 Fuel

INT.3 Engineering Operating Station (EOS)

Description:

This is local control from the operator inside the Engineering Operating Station (EOS).

Physical Links:

Sys-Local Operator

Connecting Elements:

- 2.1 Local operator control sytem (EOS)
  - 2 Machinery Centralized Control System (MCCS)
    - Sys.1 Propulsion System
- EXT.3.1 Local Operator
  - EXT.3 Operator

INT.4 Propulsor/Water Interface

Connecting Elements:

- 1.3 Propulsor
  - 1 Main Propulsion Components
    - Sys.1 Propulsion System
- EXT.5 Water

INT.5 Ship's Control Console (Bridge)

Description:

This system assumes remote control via the ship's control console (SCC) by the operator from the navigation bridge.

Physical Links:

Sys-Remote Operator

Connecting Elements:

- 2.2 Remote Operator Control System (SCC)
  - 2 Machinery Centralized Control System (MCCS)
    - Sys.1 Propulsion System
- EXT.3.2 Remote Operator
  - EXT.3 Operator

INT.6 Thrust Bearing/Hull Interface

Physical Links:

Sys-Hull

Connecting Elements:

- 1.2.5 Thrust Bearing
  - 1.2 Transmission
    - 1 Main Propulsion Components
      - Sys.1 Propulsion System
- EXT.4 Ship Hull

Part III - Physical Interfaces

Shaft-Bearing

Transmitted Data:

- Load

Connecting Elements:

- 1.2.1 Line Shaft Bearing
- 1.2.3 Shafts

Sys-Atmosphere

Transmitted Data:

- Combustion Air
- Exhaust Air

Connecting Elements:

- 3.5 Ventilation System
  - 3 Auxiliary Support Systems
    - Sys.1 Propulsion System
- EXT.1 Atmosphere

Sys-Fuel

Connecting Elements:

- 3.6 Fuel Oil System
  - 3 Auxiliary Support Systems
    - Sys.1 Propulsion System
- EXT.2 Fuel

9 Interfaces

Sys-Hull

Transmitted Data:
  - Thrust

Connecting Elements:
  - 1.2.5 Thrust Bearing
    - 1.2 Transmission
      - 1 Main Propulsion Components
        - Sys.1 Propulsion System
  - EXT.4 Ship Hull

Sys-Local Operator

Transmitted Data:
  - Desired Speed

Connecting Elements:
  - 2.1 Local operator control sytem (EOS)
    - 2 Machinery Centralized Control System (MCCS)
      - Sys.1 Propulsion System
  - EXT.3.1 Local Operator
    - EXT.3 Operator

Sys-Remote Operator

Transmitted Data:
  - Desired Speed

Connecting Elements:
  - 2.2 Remote Operator Control System (SCC)
    - 2 Machinery Centralized Control System (MCCS)
      - Sys.1 Propulsion System
  - EXT.3.2 Remote Operator
    - EXT.3 Operator

Sys-Water

Connecting Elements:

9 Interfaces

1.3 Propulsor
  1 Main Propulsion Components
    Sys.1 Propulsion System
EXT.5 Water

Allocated Capabilities/Requirements	Traced From Higher-Level Elements
Sys.1 Propulsion System (Component)	
1 Provide Propulsion (Function)	REQ.1.2 Sustained Speed (Requirement)
2 Provide Machinery Control (Function)	REQ.3.1 Machinery Centralized Control System (Requirement) REQ.3.1.5 Standards (Requirement)
3 Provide Auxiliary Support (Function)	REQ.1.5 Mobility Support Systems (Requirement)
4 Provide Redundancy (Function)	REQ.2.2 Redundancy and Separation (Requirement)
REQ.4 Lifecycle Cost (Requirement)	DOC.1 Performance Specification Document (Document)
REQ.6.1 Availability (Requirement)	REQ.6 Reliability, Maintainability, Availability (RMA) (Requirement)
REQ.6.2 Reliability w/ repair (Requirement)	REQ.6 Reliability, Maintainability, Availability (RMA) (Requirement)
REQ.6.3 Reliability w/out repair (Requirement)	REQ.6 Reliability, Maintainability, Availability (RMA) (Requirement)
REQ.7 Service Life (Requirement)	DOC.1 Performance Specification Document (Document)
1 Main Propulsion Components (Component)	
1 Provide Propulsion (Function)	REQ.1.2 Sustained Speed (Requirement)
4 Provide Redundancy (Function)	REQ.2.2 Redundancy and Separation (Requirement)
1.1 Prime Mover (Component)	
1.1 Generate Mechanical Energy (Function)	REQ.1.4 Fuel Efficiency (Requirement) REQ.1.2.1 Main Propulsion Engine Rating (Requirement) REQ.1.2.3 Light Running Margin (Requirement)
REQ.1.4.1 Prime Mover (Requirement)	Fuel Efficiency Trade Study (Issue)
1.2 Transmission (Component)	
1.2 Transfer Mechanical Energy (Function)	REQ.1.2.5 Shafting (Requirement)

Allocated Capabilities/Requirements	Traced From Higher-Level Elements
	REQ.8.2 Propulsion Shafting Vibration (Requirement)
1.4 Transfer Thrust Force (Function)	REQ.1.1.3 Thrust (Requirement)
REQ.1.2.2 Mechanical Drive (Requirement)	IPS-Mechanical Drive Trade Study (Issue) REQ.1.2 Sustained Speed (Requirement)
1.2.1 Line Shaft Bearing (Component)	
1.2.2 Main Reduction Gear (Component)	
1.2.3 Shafts (Component)	
1.2.4 Clutch (Component)	
1.2.5 Thrust Bearing (Component)	
1.3 Propulsor (Component)	
1.3 Generate Thrust Force (Function)	REQ.1.2.4 Propulsor (Requirement)
REQ.1.2.4.1 Controllable Pitch Propeller (Requirement)	Propulsor Trade Study (Issue)
1.3.1 Controlable Pitch Propeller (Component)	
2 Machinery Centralized Control System (MCCS) (Component)	
2 Provide Machinery Control (Function)	REQ.3.1 Machinery Centralized Control System (Requirement) REQ.3.1.5 Standards (Requirement)
2.1 Provide Local Control (Function)	REQ.5 Human Design Integration (Requirement) REQ.3.1 Machinery Centralized Control System (Requirement)
2.3 Collect Propulsion System Data (Function)	REQ.3.1.2 Data Acquisition & Display (Requirement)
2.4 Display Propulsion System Data (Function)	REQ.3.1.2 Data Acquisition & Display (Requirement)
2.5 Provide Connectivity (Function)	REQ.3.1.1 Connectivity (Requirement)
2.6 Perform Data Logging (Function)	REQ.3.1.3 Data Logging (Requirement)
2.1 Local operator control sytem (EOS) (Component)	

Allocated Capabilities/Requirements	Traced From Higher-Level Elements
2.1 Provide Local Control (Function)	REQ.5 Human Design Integration (Requirement) REQ.3.1 Machinery Centralized Control System (Requirement)
2.2 Remote Operator Control System (SCC) (Component)	
2.2 Provide Remote Speed Control (Function)	REQ.3.1 Machinery Centralized Control System (Requirement)
3 Auxiliary Support Systems (Component)	
3 Provide Auxiliary Support (Function)	REQ.1.5 Mobility Support Systems (Requirement)
4 Provide Redundancy (Function)	REQ.2.2 Redundancy and Separation (Requirement)
3.1 Hydraulic Oil System (Component)	
3.2 Compressed air system (Component)	
3.1 Provide Start Air (Function)	REQ.1.5.1 Start Air Pressure (Requirement)
3.3 Cooling System (Component)	
3.4 Provide Cooling Water (Function)	REQ.1.5 Mobility Support Systems (Requirement)
3.4 Exhaust Gas System (Component)	
3.5 Ventilation System (Component)	
3.5 Provide Combustion Air (Function)	REQ.1.5 Mobility Support Systems (Requirement)
3.6 Fuel Oil System (Component)	
3.2 Provide Fuel (Function)	REQ.1.3.1 Fuel Tankage (Requirement) REQ.1.4.2 Fuel (Requirement)
3.6.1 Fuel Oil Cleaning System (Component)	
3.6.2 Fuel Oil Service Tanks (Component)	
3.6.3 Fuel Oil Storage Tanks (Component)	
3.6.4 Fuel Oil Transfer Pumps (Component)	
3.7 Lube Oil System (Component)	
3.3 Provide Lubrication (Function)	REQ.1.5 Mobility Support Systems (Requirement)
3.7.1 Lube Oil Cleaning System (Component)	

Allocated Capabilities/Requirements	Traced From Higher-Level Elements
3.7.2 Lube Oil Pumps (Component)	
3.7.3 Lube Oil Tanks (Component)	
3.7.4 Oily Waste Pumps (Component)	