Aligning Early-Stage Ship Design and Model-Based Systems Engineering Methodologies

Lisa Marion*, Daniel R. Herber*

Abstract

Modern naval warships are becoming increasingly complex and highly technical systems that must be designed to meet growing global threats and the needs of warfighters. Current ship design methodologies, per contra, often reflect traditional practices that may struggle to effectively keep track of requirements changes and manage data commensurate with other digital tools utilized in design. This prompts the question: Is it time to modernize early ship design? Recognizing that a solution to managing design complexity is to migrate to a more integrated digital modeling environment, the Department of the Navy (DoN) has introduced the DoN Digital Systems Engineering Transformation Strategy [40]. The strategy recommends an authoritative knowledge repository to reduce risk, improve efficiency, and ensure that all requirements are tracked throughout the lifecycle to produce the warfighting system that users need. This paper explores the challenges posed by the increasing complexity of ships, how Systems Engineering (SE) and Model-Based Systems Engineering (MBSE) can be utilized in conjunction with current ship design methodologies to achieve efficiencies in modern military ship design, and how achieving this level of connectivity necessitates a shift in existing early-stage design methodologies and mindsets.

Keywords: MBSE, Ship Design, DoD, digital engineering, DOORS NextGen

1 Introduction

Present-day ships have become incredibly complicated, typically with a parts count of many millions, with diverse interoperability considerations for command, control, communications, computer, intelligence, surveillance, and reconnaissance (C4ISR), aviation, hull, mechanical, networks, electrical, and other systems [28]. Naval shipbuilding today requires a high-tech integration of systems that are a mix of various technologies, including electronics, mechanical systems, and software, which, particularly for weapon systems, are state-of-the-art and may still be undergoing development when a program begins [3].

Technology can be costly, and the current ship design process can be plagued by requirements changes and program restructuring [32]. A notable example is the USS *Zumwalt* (DDG-1000), often lauded as one of the most advanced warships ever but also criticized for cost overruns and challenges in integrating technology. Initially, the Navy planned to procure 32 ships, but only three were built at a cost of more than 8 billion each per Program Acquisition Unit Cost (PAUC), which is the total cost of developing, procuring, and constructing a program divided by the number of units procured. The PAUC is more than double the Acquisition Program Baseline (APB) estimate in 2005, as reported in the 2019 Selected Acquisition Report (SAR) [36].

Figure 1 shows a projection of the average number of ships per year that may be acquired as a function of time for three different budget-level assumptions (assuming budgets only increase for inflation) [3]. As costs continue to climb, fewer ships can be purchased, which potentially correlates to modern individual ship complexity. Ship programs can also suffer from repeated delays due to immature technologies [32]. On average, the ship cycle time from Material Development Decision (MDD), which the Defense Acquisition University (DAU) states is "the mandatory entry point into the major capability acquisition process, informed by a validated requirements document" [33] to Initial Operating Capability (IOC),

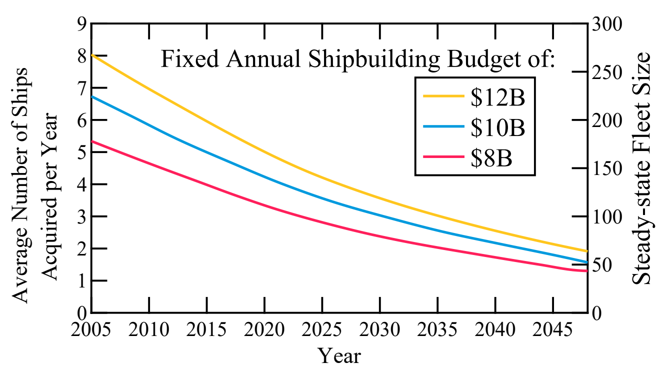


Figure 1: Average number of ships acquired per year and corresponding steady-state fleet size under varying levels of fixed shipbuilding budgets (adapted from RAND MG484-1.1) [3].

*Department of Systems Engineering, Colorado State University, Fort Collins, CO 80523, contact email addresses: lisa.marion@colostate.edu and daniel.herber@colostate.edu

or minimal operational capability, is over 15 years, which enables time for obsolescence and technology growth gaps [34].

The design of naval warships has historically been complex and is evident through the management challenges the US Navy has experienced in warship design [14]. The Department of the Navy has recognized the need for solutions to manage this complexity. The DoN Digital Engineering Transformation Strategy states an intention to "transform systems engineering capabilities by using common, composable, interactive, model-based systems to store and exchange data, models, and information within and across programs" [40]. Digital linkages, communication, and visibility become imperative as industry looks to MBSE to help capture ship design data in an "authoritative source of truth" [40] that a digital environment can provide.

To best align the state-of-the-art modern ship design with the digital age, an integrative design solution is needed to help increase cross-functional communication, identify interfaces, and ensure all requirements are met. Systems engineering using models can help reduce the risk of design complexity [9, 19].

In industries where systems engineering is already used, the evolution from a digitized paper-based design to a digitally connected environment and methodologies is challenging. But when systems engineering is not prevalent or well defined/adopted within a program, as can be the case in early naval ship design, making a decisive move to MBSE is extremely challenging. This paper analyzes the current ship design process, articulates the problem space, and guides the reader to understand converged solutions of traditional/modern ship design with SE/MBSE methodologies.

2 Common Navy Ship Design Methodologies

There are perhaps many ship design methodologies utilized, historically and today, but two prevalent methods employed in early ship design discussed here are the point-based spiral and set-based design.

2.1 Ship Design Spiral

Point-based design is a traditional iterative design, in which a baseline design is created, then configuration managed [24]. The ship design spiral is an iterative, point-based design approach that selects a solution early and iterates it to make it work [15]. The design spiral traditionally comprises 12 distinct phases of design activities, from mission requirements to cost estimates. The maturation of the design proceeds from the outer ring to the inner through four main phases of development: Concept Design, Preliminary Design, Contract Design, and Detailed Design. In the Concept Phase, Ship

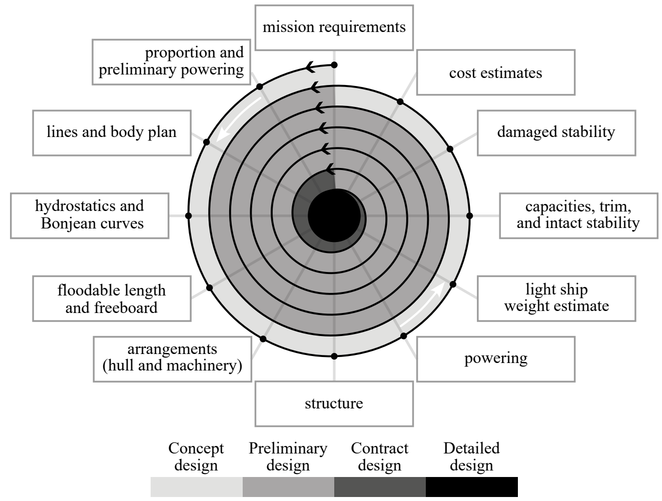


Figure 2: Ship spiral – design convergence tested and repeated [35].

Concepts (see Section 2.2) are developed and iterated as the design matures.

While there are many derivations of the design spiral first penned by Evans [18], a recent representation of the ship design spiral is shown in Figure 2. In this example, at the end of each cycle around the spiral, design convergence is tested. If the design does not converge, then another cycle at the same fidelity is reiterated. If converged, then the next stage of design is entered, where the steps are repeated at higher levels of fidelity [28].

Ship design (based on the spiral) starts with requirements for the mission, possibly including a Concept of Operation (CONOPS). A CONOPS is typically a document containing characteristics of a proposed system from the user's viewpoint and how that system fits into a larger system [33]. Once the system is defined, the spiral process proceeds with defining proportions and powering, hull form, and general arrangements of the ship, moving from a part of the spiral with less information into very specific instructions. The use of the design spiral typically continues through Contract Design and Detailed Design [28].

As the spiral often involves design activities within functional areas, product-driven bottom-up design transpires due primarily to 1) legacy ship builds and 2) domain subject matter expertise (e.g., warfare systems). In this approach, design begins with specifying the requirements and capabilities of individual subsystems, and the iterative spiral interactions among constituent subsystems and between those subsystems and the environment [28].

2.2 Set-Based Design (SBD)

Set-based design is an approach used in engineering and product development that explores a wide range of design alternatives. Through a process of elimination, infeasible or highly dominated regions of the design space are discarded, and the design space becomes more restricted [15]. Thanks in large part to a policy memo from the Commander of Naval Sea Systems outlining high-level SBD goals in early ship design phases [34], SBD is increasingly being used to improve quality and responsiveness in US naval ship design projects [15].

In the SBD process, engineers of different systems (i.e., electrical systems, combat systems, etc.) communicate ranges of solutions with associated derived requirements on other systems and levels of performance [28]. Focus is placed on identifying key knowledge gaps, conducting experiments and analyses to resolve the knowledge gaps, and deferring associated design decisions until the knowledge gap has been closed [15]. Initially, the range of solutions may need to grow to enable a sufficiently large region of feasibility at the intersection of independent solutions. After the design space has been reduced, engineers produce additional levels of detail of the subsystems to refine the solution.

In early-stage exploration, requirements should be analyzed to determine which are "tradable" and which are not. Requirements that are not tradable are those in which the system solution would not have significant value if the system did not meet the requirement. Multiple concepts are developed and compared, and the concentration is on the tradable requirements; it assesses the combinations of tradable requirement values for feasibility, effectiveness/utility, and affordability [16].

A Ship Concept is a representation of the Key Performance Parameters (KPP) and Key System Attributes (KSA) of the ship [28]. A Ship Concept is often documented as a Ship Placemat, which is ubiquitous in the ship industry as a representation of the key parameters of a ship design. At the start of Preliminary Design, following a Capabilities Development Document (CDD) approval, the requirements and CONOPS for the ship are mostly fixed [33]. A Ship Concept helps ship designers keep track of notable SBD outcomes and KPPs of the design as it matures.

In "A Case for Continuous Concept Development in Ship Design", CDR Page outlines how SBD is being used successfully in the DDG(X) Program, stating " the design team proved it could scale to a system of systems level, making it appropriate to apply (SBD) to early-stage efforts" [34]. Similarly, in another program, the Small Surface Combatant Task Force also used SBD to help identify and mature requirements [34]. SBD affords the opportunity for the design to take on a more modern, agile design approach. Agile here is defined as an adaptive, iterative, incremental development. SBD projects arrive at a design solution by systematically eliminating regions of the design space rather than by selecting a solution early and iterating it through a design spiral to make it work [15].

In summary, SBD is about *eliminating solutions that are likely not optimum rather than choosing one and modifying it to solve the given problem*. SBD techniques are ideally suited for communicating individual design solution opportunities and requirements to systematically narrow down the design space while improving design fidelity [28].

2.3 Are Complexity Needs Met by the Design Methods?

In the Ship Spiral Design, early design decisions are considerably more significant than later-stage decisions. Often the ship design is "done" when time has run out, not necessarily when the design is converged or optimal [28]. For this reason, the design spiral may be more appropriate to refine an existing solution, rather than as a method to achieve the initial, almost optimal converged starting concept [28]. Also lacking in the design spiral is a proper approach to define the ship requirements and fully explore the problem domain before formulating trade-off analysis of possible solutions, because the process starts when requirements have already been defined in agreement with the client, which are used as inputs to develop a balanced solution [10].

The SBD method is conventionally described as a process of generation and elimination, but there are pitfalls that arise in applying this method. The way specific details are managed can directly impact the success or failure of the design outcome. For example, delaying decisions confers no intrinsic benefit of its own; value is created only when such a delay is designed to generate lead time to gain specific types of additional information needed to make a better decision. Otherwise, delay is merely procrastination, which reduces focus and dissipates momentum [15].

The design of a warship should seek to reduce complexity through the acquisition and construction of the warship, and the design process itself [14]. Perhaps, due to design complexity, failure to look at ship design holistically as stated in Section 1, and the balance of refining the optimal solution, non-optimal outcomes have resulted. While by many accounts a success, the USS *Zumwalt* (DDG-1000) is missing a variety of necessary defense features for any modern ship, such as anti-torpedoes, long-range air defense missiles, and anti-ship missiles [32]. As the time and costs involved in acquiring technologically advanced ships grow, it is possible that the complex user needs are not being adequately addressed.

3 Confluence of a Modern Solution

Systems engineering (SE) has been practiced for many years, long before the term was coined to describe its methodology. As the name implies, it is engineering related to the whole system.

The International Council on Systems Engineering (INCOSE) defines systems engineering as the "transdisciplinary and integrative approach to enable successful realization, use and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods" [23]. At its core, it is a holistic analysis of the system to ensure that what the user wants is what the user gets.

Systems engineering starts with a definition of the desired capabilities and evolves the solution through various subsystems and components required to meet capabilities over the life of the system. With an understanding of SE principles, one can appropriately tailor the SE process to the development of any system or find a solution to any problem [43].

3.1 Types of Requirements

Given how important the idea of a requirement is to what the Office of the Chief of Naval Operations (OPNAV) does, it is surprising how often the term "requirements" is used in the absence of a specific context. In the Navy, a requirement is nominally defined as the capabilities or conditions possessed by a system or a component needed to solve a problem or achieve an objective [5].

User needs are requirements that are often defined early in the life cycle of a system as *capabilities*. Capabilities are high-level concepts that describe the key performance or tasks that a system must capture in its design [33]. Capabilities have a connection to system requirements, often in the form of derivation relationships. Future capability requirements usually must evolve from existing capabilities or take present or projected capabilities into account. For example, a completely unconstrained requirement that suggests a platform or weapon that is physically incapable of operating with existing or programmed platforms is likely to be changed to ensure interoperability [5].

A key difference between capabilities and requirements is that requirements can originate from many sources, be sponsored by different stakeholders, and are usually captured at different levels of abstraction. System capabilities should always represent a consolidated view of the system or enterprise. Capabilities in the DoD are typically captured in the form of an Initial Capabilities Document (ICD), a subsequent CDD, or both [33].

The terms "Big R" and "little r" refer to whether a requirement has been validated by a recognized decision authority, such as the Resources and Requirements Review Board (R3B) or the Joint Requirements Oversight Council (JROC). Generally, "Big R" requirements are JROC or R3B validated requirements. Validation is essentially an entry fee to qualify to compete for resources. So "little r" requirements are any non-validated expression of need [5]. "Big R" requirements can be equated to stakeholder needs, and "little r" often describes derived requirements.

3.2 Importance of Requirements and Functional Analysis

To quote a senior naval architect, "the spiral is understood, but how do we design so that we don't miss requirements?" To ensure a converged, integrated design in which requirements aren't missed requires an SE methodology [23]. SE can help ship design to "not lose requirements" because it is a holistic and interconnected evaluation of the system. It helps inform the ship design and keep it accountable to the user's needs [9].

As described in Figure 2, the ship design spiral's initial phase is "mission requirements". To mitigate the unrequired needs identified in Section 1, the proper analysis of requirements forms the imperative first step in any design methodology. In systems engineering, 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 [33].

Ship requirements analysis often starts with parent specifications or requirements from other like-ship designs of similar class [28] to build into a system-level requirements baseline. Typically led by the Ship Design Manager, a Final Specification reading session lasts about six weeks and is reviewed based on SWBS (Ship Work Breakdown Structure) groups. SWBS is the basic context within which the entire ship design effort is planned, managed, and documented [28]. These reading sessions are usually held at critical milestones in the development cycle and are often attended by all Subject Matter Experts (SMEs) to ensure visibility. Multiple reading sessions are required to mature the requirements [28].

It is extremely important that the ship as a system has necessary, measurable, and verifiable requirements, which must be parsed into single, testable, standalone requirements statements [33]. In this first step of analysis, requirements are often grouped by type, which are typically either functional, nonfunctional, or programmatic [33]. A functional requirement is a task, also called an action or activity, that must be accomplished to provide an operational capability or to satisfy an operational requirement. Non-functional requirements describe quality attributes of the system, such as reliability, security, and availability. Programmatic requirements are imposed on the management of the system as opposed to the system itself (i.e., schedule, required meetings, etc.).

Requirements analysis must clearly define the functional requirements and design constraints [20], prioritizing the functions of a naval ship rather than the individual components that constitute it [10]. In functional analysis, functions are analyzed by decomposing higher-level functions into lower-level functions, resulting in a description in terms of what the product should logically do and what performance is needed [20]. Functional requirements define quality (how good), quantity (how many), coverage (how far), timelines (when and how long), and availability (how often). Constraints in the design stage define those factors that limit design flexibility, customer

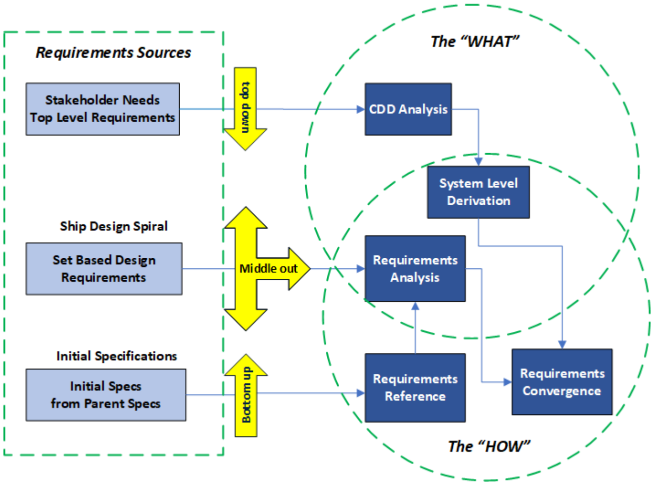
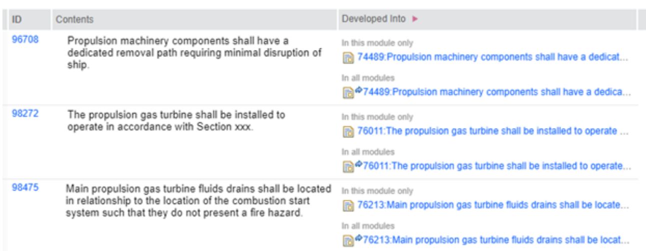


Figure 3: Top-down and bottom-up requirements convergence.



ID	Contents	Developed Into
96708	Propulsion machinery components shall have a dedicated removal path requiring minimal disruption of ship.	In this module only 74489:Propulsion machinery components shall have a dedicat... In all modules 74489:Propulsion machinery components shall have a dedica...
98272	The propulsion gas turbine shall be installed to operate in accordance with Section xxx.	In this module only 76011:The propulsion gas turbine shall be installed to operate ... In all modules 76011:The propulsion gas turbine shall be installed to operate...
98475	Main propulsion gas turbine fluids drains shall be located in relationship to the location of the combustion start system such that they do not present a fire hazard.	In this module only 76213:Main propulsion gas turbine fluids drains shall be locate... In all modules 76213:Main propulsion gas turbine fluids drains shall be locat...

Figure 4: DOORS NextGen Requirements Traceability Matrix.

or regulatory standards, such as environmental conditions, internal or external threats, and contract [20].

3.3 No Requirement Left Behind

A solution to the problems of missed requirements and silos of design is a convergence of top-down, traditional SE and bottom-up, integrated ship design. In a converged approach, engineering often begins with various operational scenarios and uses an SBD methodology to identify gaps in understanding. Analyses based on these gaps can provide information to help update and/or add additional design requirements [39] and merge with existing system requirements (or initial specifications), and can be derived to identify the requirements of the subsystems and related components. In Figure 3, the "what" or problems that have been identified by the user (stakeholder needs) overlap and converge with the set-based design requirements and the "how" or solutions to the problems via systems engineering and requirements analysis. The "what" requirements are derived down to answer the "how", and to ensure that SBD solutions are also answering stakeholder needs.

Systems engineers typically employ a different methodology to mature requirements than explained in Section 3.2. Requirements development is usually handled within working groups, which are attended by specific functional area SMEs. A requirements management tool, such as DOORS, is a system engineer's legacy choice for maturing and managing requirements. DOORS NextGen, which is web-based (depicted in Figure 4), can be part of the digital thread of connectivity to other digital authoring tools [19]. A web-based tool offers real-time requirements collaboration between systems engineers and designers, greatly improving productivity and efficiency and reducing errors. Within any requirements management tool, a unique ID is created for each standalone item (which can be programmatic/functional/non-functional requirements or "information only", described in 3.2). This is an identification (numbers, letters, and both) uniquely associated with the artifact to enable traceability to all derivations and products throughout the lifecycle.

3.4 Systems Engineering Can Help Manage Early Design

Steps within the spiral involve independent teams, so even with cross-functional and integrated product teams, miscommunication and gaps in visibility can result in silos of design, inefficiency, and missed requirements. A combined set-based and SE methodology has a great potential to manage the constraints of the design due to the ability to build knowledge in parallel, especially for advanced and non-mature technologies [8].

Spiral design looks at how smaller systems fit into the larger system needs, while SE looks at the user's needs to develop the system. SE cannot mandate a top-down approach — rather, SE can help complement a bottom-up approach with SE methodology to ensure bidirectional traceability of all user needs.

A solution to the *convergence* with SE looks more like a "meet-in-the-middle" approach to requirements analysis and synthesis, as seen in Figure 3. This compromise ensures that design approaches are not lost, and are strengthened as they merge together, which are hosted and connected within a digital environment.

Although SBD affords a more agile approach than the design spiral (see Section 2.2) in early ship design, the development of the system and its operational, functional, and physical elements (defined in Section 4) and their interrelationships and constraints must also be clearly defined. Functions must be based on performance requirements — what the system has to do — to complete traceability. The top-level performance requirements must be derived so that they can be allocated to their respective lower-level functions. This ensures that every function allocated to a component is traceable back to a top-level requirement.

This process is not complete until all performance requirements have been allocated to elements of the system [23].

4 Shift to MBSE

4.1 Definition of MBSE

Traditional Systems Engineering is often referred to as a document-based approach. This approach is characterized by the generation of textual specifications and design documents and is rigorous, but has some limitations. Because information is spread across many documents, it can be difficult to assess the relationships between requirements, design, engineering analysis, and test [19]. Progress of the systems engineering effort is based on the documentation status, which may not adequately reflect the quality of the requirements and design. Because of these limitations, traditional SE method practitioners now increasingly look to Model-Based Systems Engineering (MBSE) methodologies. MBSE is the formalized application of modeling to support the requirements, design, analysis, verification, and validation activities of the system, beginning in the conceptual design phase and continuing throughout the development and later phases of the life cycle [23].

The central point of MBSE is that it is still SE — just enhanced with models and diagrams.

4.2 MBSE in Practice

MBSE can help address the problem of complexity of ship design by offering a single source of truth: a scalable and traceable digital model of the system and its requirements [19]. It also improves efficiency and ensures that all requirements are validated in the operational environment of the system [9]. MBSE helps to graphically relate requirements to the functional abstraction and design documentation of the ship. MBSE modeling languages also demand precision. This yields clear and precise requirements, which are more readily verified and tested [23]. MBSE also enables stakeholders to interact within the collaborative modeling environment as SE artifacts are developed, vetted, and approved.

Industry and Department of Defense programs are moving to MBSE to help direct system design. The Office of Naval Research (ONR) has sponsored research for a new Power Electronic Power Distribution System (PEPDS) for Navy vessels. The PEPDS architecture team has successfully baselined a functional architecture using an MBSE approach that it believes will help propose alternative designs and select a preferred design [2]. Also, in 2009, the Submarine Warfare Federated Tactical Systems (SWFTS) program conducted a Model Driven Architecture (MDA) study to determine if MDA should be the next step in the program's continuous SE process improvement. The MDA study predicted a positive Return on Investment (ROI) when converting to MBSE. Based on that successful validation, the SWFTS program transitioned to MBSE for all ongoing work [25].

NASA has been investigating the use of MBSE since 2009. Currently, there is an MBSE leadership team at every center, NASA has published its own SE Modeling Handbook, and it is applying MBSE to a number of projects, including Advance Air Mobility projects (includes unmanned aerial systems), Power Propulsion Element (a Gateway/Artemis system), and Lunar Surface Architecture projects [26].

4.3 MBSE Features and Advantages

MBSE can support SBD by keeping track of all knowledge gaps, analyses, and resultant requirements that arise. This enables naval architects and engineers to work with a common set of data throughout the life cycle. In addition, the MBSE environment offers an Authoritative Source of Truth (ASoT) — or a central element around which all activities and data revolve — for all SBD data [37]. The MBSE environment provides complete traceability between any SBD artifact, requirements baselines, and eventual ship specifications. Without this type of modeling, there is no means to update in near-real time the response variables based upon design decisions (inputs) and/or requirement changes. This update with SBD and an integrated MBSE model provides an improved decision analysis to evaluate and select alternatives in early design [39].

A modeling environment capable of managing the capability concepts and configurations is an important enabler of using SBD in concept exploration [16]. A modeling environment called Framework for Assessing Cost and Technology (FACT) leverages SysML (to be discussed in Section 4.5.2). FACT is a framework that provides a structure to collaboratively conduct analysis of complex systems [16]. Used within an MBSE environment, FACT, coupled with other structures (like a SWBS or Product Breakdown Structure), could enable the management of SBD in the concept phase.

MBSE establishes a visually traceable and scalable systems model to reduce risk and gain efficiency to help ensure that all requirements are validated. One dynamic view of the model is a traceability matrix. This kind of matrix enables the project team to ensure that requirements are allocated from the system to the subsystem, to ensure that each requirement is validated, and to evaluate gaps in allocation and validation [23]. An MBSE environment can be utilized instead of an SE tool like DOORS (discussed in Section 3.3) for requirements management, or it can be used in conjunction with a tool like DOORS via an MBSE software plug-in as part of a digital environment.

Other advantages of MBSE include:

- *Auto-generation of documents*: reports, specifications, etc. are exported from the repository, not self-authored, using templates created and stored in the MBSE model.
- *Reduces risk*: proposed changes to the baseline can be quickly and comprehensively assessed for impact on the integrated system [9].
- *Facilitates communication*: Using a single repository for

the technical baseline provides an ASoT, reducing miscommunication amongst the team.

- *Improves quality*: mistakes made by trying to maintain configuration control of technical elements contained in disparate documents, tables, etc., can be avoided by updating the model element once and auto-propagating everywhere it is used [9]. This enables the engineering team to "define it once, use it many times . . . ".

Finally, perhaps one of the biggest advantages of a program using MBSE is the ease of reporting metrics, as the data originates from one source. MBSE can define metrics to assess design progress by establishing completion criteria for the design, which is beneficial to leadership and program management [19].

4.4 MBSE — SE Methods Modernized with Models

Models serve as the principal means of capturing and communicating knowledge about a system over its life, replacing the typically large number of discrete static artifacts (documents) that increase in inconsistency as each evolves over time [29].

In Systems Engineering, the "V" is an illustration of SE activities throughout a system's lifecycle — time and maturity proceed from left to right, while the core depicts evolving the baseline from user requirements to final system [23].

Figure 5 shows the traditional SE "V", or the roadmap of ship design from concept to final verification and validation of the system, coupled with model-based methods, to illustrate the convergence of early ship design and MBSE. The various model views are created to support the iterative ship design process. MBSE begins with analyzing the requirements of the user, on the left side of the "V", while the engineering teams are analyzing the inherent structure and required functions. Simultaneously, a system of systems breakdown and functional analysis may take place. The "Model" is the core of the "V", depicted by the requirements/structure/behavior cube, which is the hub of all design and behavior data. The green "agile" spiral on the left side of the "V" indicates a more adaptive, iterative development, as discussed in Section 2.2.

4.5 How to Get Started with MBSE

4.5.1 Have a Plan . . . Literally

For any new undertaking to be successful, especially one of such magnitude as converging methodologies, a plan of action is advisable. A Digital Engineering Strategy (DES) provides a holistic overview of the goals of digital engineering within a project. The DES, which could be based on the DoN Digital Systems Engineering Transformation Strategy [40], should outline the goals of the use of models and the digital environment, such as:

- Authoritative source(s) of information
- Accurate and complete requirements
- Digital infrastructure and interfaces
- Reduced risk earlier in the lifecycle than with traditional tools

The Digital Engineering Strategy is an important roadmap to help the Lead Systems Engineer and the Program Manager manage the MBSE aspects of the digital engineering expectations for the program. It must identify needs and articulate vision, objectives, and metrics [44].

The DES is a living document that will evolve along with the lifecycle and will be subordinate to the Systems Engineering Plan (SEP). DAU defines the SEP as "help for Program Managers to develop, communicate, and manage the overall SE approach that guides all technical activities of the program, and documents key technical risks, processes, resources, metrics, SE products, and completed and scheduled SE activities" [33].

As the digital model-based strategy is facilitated by the systems engineering team, the DES and SEP will together communicate the details of systems engineering planning to the entire team. Once a commitment to MBSE is made and a strategy is in place, the program will look to the systems engineering team for the next steps. MBSE implementation requires three things: a modeling language, a modeling methodology, and a digital tool, which are discussed in the following sections.

Please note that this paper does not compare or recommend one language, methodology, or tool over another, but offers some possibilities for further exploration.

4.5.2 Modeling Language

A modeling language provides a holistic understanding of a system or structure that people can consistently understand. The modeling language facilitates the description of the system-of-interest primarily leveraging graphical constructs [27]. Some examples of modeling languages are: 1) *Object-Process Language (OPL)*: a subset of English that employs both the visual (graphical) modality and the verbal (textual) modality. It is the first modeling language to combine high-level algebraic and set notations of modeling languages with a rich constraint language and the ability to specify search procedures and strategies [41]. 2) *SysML (Systems Modeling Language)*: a general-purpose system architecture modeling language for SE applications defined through an open standard and the de facto language for MBSE [17, 19]. 3) *UML (Unified Modeling Language)*: a language that primarily helps software developers visualize and construct new software systems [30].

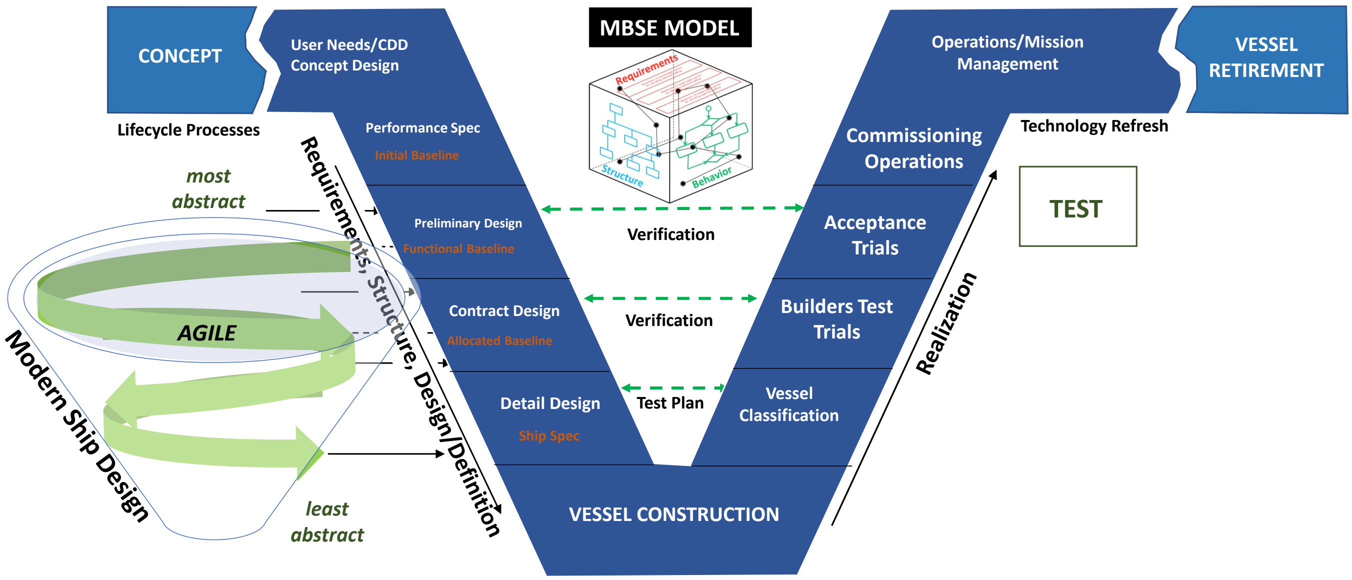


Figure 5: MBSE model-centric “V” coupled with agile ship design.

4.5.3 Modeling Methodology

An MBSE methodology implements a process using a modeling language and describes how the process steps should be performed (what to model and when) [7]. Some examples of MBSE methodologies are: 1) *OOSEM (Object-Oriented Systems Engineering Method)*: an integrated framework that combines object-oriented techniques, a model-based design approach, and traditional top-down SE practices [19, 27]. 2) *MagicGrid*: aligned with SysML and IEEE 15288, tool-independent roadmap based on a framework of domains and MBSE pillars [1]. 3) *Harmony-SE*: an agile approach to systems and software development that is centered on requirements and architecture-focused [42].

4.5.4 Modeling Software Tool

A modeling tool is a software package that is used to create, review, and manage system models in accordance with a modeling language. The MBSE methodology can be implemented by one or more MBSE tools. But choosing the right modeling tool for a program really starts with a good understanding of the needs and processes. Some examples of MBSE modeling software tools are: 1) *Cameo Systems Modeler* or similar products from CATIA No Magic [1], 2) *Enterprise Architect* from Sparx Systems [38], and 3) *Rhapsody Architect* from IBM [21].

4.6 Architecting the System with MBSE

Using digital modeling tools and methods over traditional system engineering techniques can help best create the system architecture. Many groups, including the International Council on Systems Engineering (INCOSE), advocate for an architecture-centric model [9]. Architecture is defined as “the art of designing and building something.” In SE terms, a representation of the system can be referred to as the *system architecture*.

Although there is not a universally accepted definition of system architecture, [9] defines system architecture as the formal representation of a system or other complex entity to clarify:

- Its structure, interfaces, and internal and external relationships
- The behaviors exhibited by the entity and its elements, both internally and externally
- The global rules to which the entity and its elements must conform in order to meet their allocated requirements

An example of a notional architecture is shown in Figure 6.

Building the architectures within the MBSE model enables configuration control of requirements, visibility to interfaces, and the ability to track cost and design decisions. By ensuring that the desired beginning is manifested in the end state, MBSE, like SE, boils down to “connecting the dots”. To implement a design roadmap in MBSE, a representation of the system must be created; the genesis of this can be referred to as the system architecture.

The time, cost, and complexity problem of new ship construction requires a sophisticated system architecture to track the intricate design and ensure user needs are met. This focuses on suitability, viability, and desirability, while design focuses on compatibility with technologies and other design elements and the feasibility of construction and integration [22].

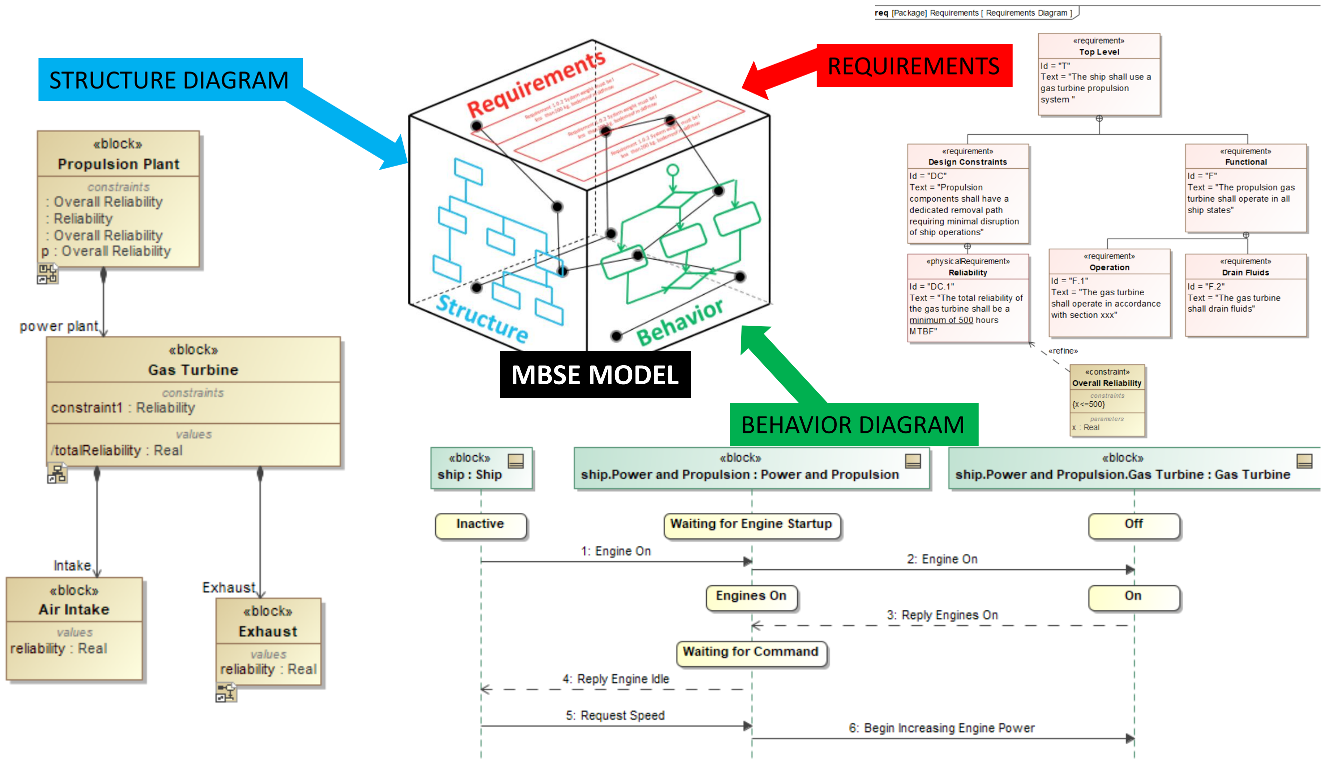


Figure 6: Notional ship architecture interrelationship through a central single source of truth MBSE model [4].

MBSE typically begins with an analysis of the problem without prior knowledge of its internal structure or behavior. In ship design, there is a need to prioritize the functions of a naval ship rather than the individual components that constitute it [10], which is where MBSE practice and a top-down approach can help. A well-modeled architecture enables the reuse of components to ease development and management for current and future programs or projects [11], can provide the necessary linkages to SBD decisions to track static and dynamic elements of design (the "middle out" process described in Figure 3), and can improve the overall decision-making process.

The holistic architecture can be defined as a conceptual model that describes the operational, logical, and physical architectures of the vessel, together with the interrelationships between the three architectures [11].

4.6.1 Operational Architecture

The MBSE model begins with the operational scenario and decomposes it into its requisite parts and levels of abstraction to define the operational needs of the system [9]. Operational architecture is a description of the requirements, tasks, operational elements, and information flows required to perform or support a warfighting function. In the DoD, an operational scenario is often described using a CONOPS [33] (see section 2.3). The operational architecture contains information on the temporal behavioral characteristics of the vessel, in a given mission scenario [11].

4.6.2 Logical/Functional Architecture

A logical, or 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 [33]. It embodies the design-agnostic physical representation of the system to allow maximum flexibility in the design trade space [22]. The logical architecture may be aligned with the ship SWBS breakdown.

The development of a functional architecture and the definition of system functions should not be performed in isolation; they should be developed incrementally with stakeholder requirements and the architecture to ensure that the appropriate functions and interfaces are identified [33]. Defining the functional or logical architecture of the ship design mitigates the impact of requirements and technology changes of system design [23] and identifies the functional baseline of the system.

Figure 6 depicts the MBSE model as the center of requirements, structure, and behavior of a system. The structure and behavior of a system define the logical/functional architecture of the design. In this notional example, the propulsion plant is broken down into a few elements. The requirements (functional and non-functional) of the subsystem are identified at the top right, and the bottom right is a sequence diagram describing one of the behaviors of the subsystem. All of this data is connected in the model so the designer can see how the requirements relate to the structure and identify interfaces in the behavior of the requirements.

4.6.3 Physical Architecture

Once a logical architecture is defined (see Section 4.6.2), concrete physical elements must be identified that can support functional, behavioral, and temporal features as well as the expected properties of the system deduced from non-functional system requirements. The term *physical architecture* is a contraction of the expression Physical View of the System Architecture [6]. The physical architecture is defined by the hardware, software, and interfaces that implement the behaviors in the logical model and describe relationships among actual physical system elements, including hardware and software [23]. The System of Systems (SoS) will be defined by a high-level physical architecture, which will be utilized to define the relevant SoS relationships, interfaces, and constraints [6]. Programs, such as the Expeditionary Mobile Base ship, have employed SWBS for the ship system of systems and broken out components of the ship architecture [28].

The physical architecture is also quite important to build test cases for verification of performance requirements, and can be used to manage the ship's lowest-level configuration items (i.e., hardware and software entities that satisfy an end-use function and that can be uniquely identified at a given reference point).

4.7 Overcoming MBSE Challenges

The successful integration and implementation of modern ship design with SE and MBSE present significant technical challenges in the ship acquisition process, as previously outlined. To achieve the best results, change must be planned, communicated, and implemented at all levels of design and management. As discussed in Section 4.5.1, including a Digital Engineering Strategy to inform management of the intentions of the digital engineering team helps manage expectations of what is achievable.

With any new shift in methodology or process, training is recommended. A program must establish and evaluate the competencies of the MBSE workforce that require new skills, proficiency, knowledge, and training, and must provide retooling of the workforce to maintain subject matter expertise [40]. To integrate MBSE into the existing ship design process, a certain level of MBSE capability is preferred, depending on the number of subsystems or the sophistication of the system, and who is using or referencing the model. Some training recommendations for hands-on practitioners include SysML (an MBSE language, see 4.5.2) by OMG, which offers industry standard certifications at different levels of competency [31]. The Government also provides resources for MBSE guidance/training, such as those offered by DAU and the Digital Engineering Book of Knowledge [13].

Beyond the planning and technical challenges, a further formidable barrier to success, as noted in Section 1, is cultural change. Implementing MBSE requires not only new tools and updated processes but also a fundamental shift in the organization's culture. For veterans accustomed to traditional and legacy processes, adapting to these changes may be the most imposing obstacle.

Another, arguably obvious, challenge to MBSE adoption is communication. Effective communication between Program Management, the Lead Systems Engineer and Ship Design Manager, the MBSE Team and the Integrated Product Teams (IPTs) and Cross Product Teams (CPTs) is imperative to ensure that the teams are on the same path in implementing MBSE (this is where a Digital Engineering Strategy can be most effective; see Section 4.5.1). If the leadership teams agree on methodology, proper tasking will flow to the teammates. This ensures that the subject matter experts' output changes in the design to the SE team, so that necessary updates can be made to the model once, propagated throughout the model data, and appropriately configuration controlled.

Embracing new ideas and methodologies also introduces issues of trust. It is important to ensure that managers/team leads trust the new way of doing things when there may be limited observable metrics and success stories to "prove" MBSE adoption will be a success. Trust is inherently coupled to communication. Clearly defining how and why MBSE enhances, rather than disrupts, existing processes - documented in a framework like the DES - helps the broader team align with MBSE adoption, minimizes uncertainties, and fosters trust.

4.7.1 Unify the Vernacular

Merging unique design methodologies has another unanticipated problem — an uncommon lexicon. Seemingly benign, the lack of a ubiquitous language within the program can result in misunderstanding, miscommunication, and, most egregiously, rework.

Systems engineering has its own definitions, and MBSE defines the granularity of these based on a chosen modeling language (discussed in Section 4.5.2). Ship design also has its own language — for example, design is often described in cycles [28], while SE milestones might be described in phases [33], or as mentioned in Section 3.1, requirements can be referred to as "Big R"/"little r" in OPNAV, while Systems Engineering may refer to these requirements as stakeholder needs/derived requirements.

The dictionary is a useful artifact for reconciling domain-specific definitions. Architecture in the ship world could mean the physical entities of the design, while SE systems architecture could be operational, logical, and physical views of the system. An MBSE dictionary could also include an agreed meaning of terms like verification – which answers the questions "Does it meet the requirements, and did we build it correctly?" and validation – which answers the questions "Are we building the right thing, and do we have the right design?" [33]. These two terms are often used interchangeably in many domains.

5 Discussion and Future Work

The implementation of MBSE in new programs, especially those without a defined SE presence, poses unique challenges outside of the obvious technical issues. MBSE does not replace traditional SE, nor does it replace traditional ship design – rather, it strengthens and enhances both. Success depends on full involvement in and commitment to the scope of the MBSE effort, for example, by a DES as described in Section 4.5.1, and in ensuring appropriate resources for the MBSE team, its methodologies, and its goals. Creating a collaborative digital environment helps the systems engineers, and functional area engineers, look at system requirements, how they are allocated to their respective areas, understand areas of overlap, and model behaviors that beget new requirements and constraints on the system that need to be addressed.

Program leadership must recognize that the potential benefits of MBSE extend throughout the entire lifecycle of the program rather than being limited to early development phases alone [12]. MBSE Return on Investment (ROI) is not easily quantifiable, so ROI is often expressed as an overall reduction in risk, with better communication and traceability compared to traditional ship design. Currently, using MBSE metrics to justify ROI should be avoided to refrain from "over-promising and under-delivering" the system. To quantify valid metrics, traditional design and MBSE-driven ship design would — perhaps in a small subsystem — need to be performed separately and compared to each other. Using this "apples to apples" comparison could lead to a standard by which to gauge MBSE ROI.

Future work regarding SE/MBSE could center around establishing an SE/MBSE team for effective communication and trust within the program and could identify training/expertise levels needed within a program. The SE Guidebook provides a framework for program structure [33].

Another potential future discussion of MBSE in ship design could involve redefining the requirements analysis and convergence process as seen in Figure 3. Using parent specifications to define the initial specifications, as described in Section 3.2, and then tailoring those to meet the top-down analysis and SBD requirements, is a legacy process that may warrant introspection to identify inefficiencies.

While this paper focuses on the left side of the SE "V", another discussion of MBSE in ship design could elaborate on the lifecycle after construction, in the realization of the design and operation of the ship. MBSE can also be helpful in understanding the breadth of programmatic requirements like training, support, and sustainment. MBSE can work together with other digital tools and methodologies that comprise the full breadth of Digital Engineering, such as Product Lifecycle Management (PLM), which would more fully define the lifecycle digital thread.

6 Conclusions

The Navy ship design process has been implemented for generations, but to meet the complexities of current global threats, the DoN Digital Systems Engineering Transformation encourages a more connected, digital environment using models and an ASoT. This could be a forcing function for the military and industry to rethink the current approach to early ship design.

An analysis of current ship design methodologies uncovers shortcomings and proposes a convergence of the outlined methods and Model-Based Systems Engineering (MBSE). Although SBD affords a more agile approach than the design spiral in early ship design, the development of the system and its operational, functional, and physical elements and their interrelationships and constraints must also be clearly defined. MBSE and a well-modeled architecture can provide the necessary linkages to SBD decisions to track static and dynamic elements of design, so that functions of the ship are prioritized over individual components, defining "what" is necessary to meet the customer's requirements before solving "how" to achieve those objectives. A proposed convergence solution begins with a disciplined application of Systems Engineering (SE) best practices, which includes top-down requirements analysis, architecture development, and traceability. MBSE builds on traditional SE rigor and satisfies the digital connectivity need by offering an ASoT and providing the methodology to create an integrated, digital, graphical representation of the ship's requirements, structure, and behavior. MBSE offers a configuration-controlled, collaborative digital environment, which is especially necessary during the concept and preliminary stages of ship design but is also vital throughout the lifecycle to create the digital thread that ensures user needs are met. MBSE mitigates risk by reducing the chances of overlooking the implications of changes and decisions made throughout the lifecycle and provides a legacy of connected data for the next generation ship build.

The SE team, Program Management, and engineering teams must work collaboratively to set and manage expectations for the efficacy of MBSE. Achievable goals must be defined to overcome the cultural shift and barriers to success that will allow a model-driven ship design process that ensures that all necessary requirements are validated in the final product.

A Acronyms

- APB – Acquisition Program Baseline
- ACAT – Acquisition Category
- ASoT – Authoritative Source of Truth
- C4ISR – Command, Control, Communications, Computer, Intelligence, Surveillance and Reconnaissance
- CDD – Capabilities Description Document
- CDR – Commander
- CONOPS – Concept of Operations
- CPT – Cross Product Team
- DAU – Defense Acquisition University
- DES – Digital Engineering Strategy
- DoD – Department of Defense
- DoN – Department of the Navy
- DOORS – Dynamic Object Oriented Requirements System
- FACT – Framework for Assessing Cost and Technology
- ICD – Initial Capabilities Document
- IPT – Integrated Product Team
- INCOSE – International Council on Systems Engineering
- KPP – Key Performance Parameter
- KSA – Key System Attribute
- MBSE – Model-Based Systems Engineering
- MDAP – Major Defense Acquisition Program
- NASA – National Aeronautics and Space Administration
- ONR – Office of Naval Research
- OOSEM – Object-Oriented Systems Engineering Method
- OPNAV – Office of the Chief of Naval Operations
- PAUC – Program Acquisition Unit Cost
- PEPDS – Power Electronic Power Distribution System
- PLM – Product Lifecycle Management
- ROI – Return on Investment
- SAR – Selected Acquisition Report
- SBD – Set-Based Design
- SE – Systems Engineering
- SEP – Systems Engineering Plan
- SME – Subject Matter Expert
- SWBS – Ship Work Breakdown Structure
- SysML – Systems Modeling Language
- UML – Unified Modeling Language

Declaration

The authors declare no potential conflicts of interest with the publication of this article. The views expressed herein are solely of the authors, not of a particular company or the Government.

References

[1] Aleksandraviciene, A., Morkevicius, A., 2018. MagicGrid Book of Knowledge. Vitae Litera.

[2] Araujo, C., Gross, D., Steurer, M., Song, S., Schegan, C., 2023. Baselining a functional architecture for a power electronic power distribution system for navy vessels, in: 2023 IEEE Electric Ship Technologies Symposium (ESTS). doi: 10.1109/ESTS56571.2023.10220545.

[3] Arena, M.V., Blickstein, I., Younossi, O., Grammich, C.A., 2006. Why has the cost of Navy ships risen? A macroscopic examination of the trends in U.S. Naval ship costs over the past several decades. RAND Corporation.

[4] Bajaj, M., Cole, B., Zwemer, D., 2016. Architecture to geometry – integrating system models with mechanical design, in: AIAA SPACE 2016. doi: 10.2514/6.2016-5470.

[5] Blickstein, I., Yurchak, J., Martin, B., Sollinger, J., Tremblay, D., 2016. Naval Planning, Programming, Budgeting, and Execution. Technical Report. Rand Corporation.

[6] Board, S.E., 2024. The Guide to the Systems Engineering Body of Knowledge (SEBoK). Technical Report. The Trustees of the Stevens Institute of Technology.

[7] Boggero, L., Chojnacki, A., Bussemaker, J., Bartels, J., Quantius, D., Nagel, B., 2023. The MBSE competence at the German Aerospace Center, in: INCOSE International Symposium. doi: 10.1002/iis2.13061.

[8] Borchani, M.F., Hammadi, M., Yahia, N.B., Choley, J.Y., 2019. Integrating model-based system engineering with set-based concurrent engineering principles for reliability and manufacturability analysis of mechatronic products. Concurrent Engineering 27, 80–94. doi: 10.1177/1063293X18816746.

[9] Borky, J.M., Bradley, T.H., 2019. Effective Model-Based Systems Engineering. Springer. doi: 10.1007/978-3-319-95669-5.

[10] Bottero, M., Gualeni, P., 2024. Systems engineering for naval ship design evolution. Journal of Marine Science and Engineering doi: 10.3390/jmse12020210.

[11] Brefort, D., et al., 2018. An architectural framework for distributed naval ship systems. Ocean Engineering 147, 375–385. doi: 10.1016/j.oceaneng.2017.10.028.

[12] Carroll, E.R., Malins, R.J., 2016. Systematic Literature Review: How is Model-Based Systems Engineering Justified? Technical Report SAND2016-2607. Sandia National Laboratories. doi: 10.2172/1561164.

[13] DEBoK, version 3.1.0. URL: https://de-bok.org/.

[14] Doerry, N., Amy, J., 2024. Reducing complexity in ship design, in: ASNE Advanced Machinery Technology Symposium.

[15] Doerry, N., Koenig, P., 2019. Implementing set-based design in DOD acquisitions, in: Annual Acquisition Research Symposium Naval Postgraduate School.

[16] Doerry, N., et al., 2014. Using set-based design in concept exploration, in: SNAME Chesapeake Section Technical Meeting.

[17] Dori, D., 2016. Model-Based Systems Engineering with OPM and SysML. Springer. doi: 10.1007/978-1-4939-3295-5.

[18] Evans, J.H., 1959. Basic design concepts. Journal of the American Society for Naval Engineers 71, 671–678. doi: 10.1111/j.1559-3584.1959.tb01836.x.

[19] Friedenthal, S., Moore, A., Steiner, R., 2012. A Practical Guide to SysML. 3rd ed., Morgan Kaufmann. doi: 10.1016/C2013-0-14457-1.

[20] Guajardo, R., 2020. Systems engineering modelling and simulation to support defence acquisition system. Hadmérnök 15, 17–42. doi: 10.32567/hm.2020.3.2.

[21] IBM, 2024. IBM engineering systems design Rhapsody. Webpage by IBM. URL: https://www.ibm.com/products/systems-design-rhapsody.

[22] IEEE, 2023. 15288-2023 - ISO/IEC/IEEE International Standard - Systems and software engineering – System life cycle processes. Standard ISO/IEC/IEEE 15288:2023(E). doi: 10.1109/IEEESTD.2023.10123367.

[23] INCOSE, 2023. Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities.

[24] Keane, R.G., Tibbitts, B.F., 2015. The fallacy of using a parent design: "The design is mature". Transactions (Society of Naval Architects and Marine Engineers [SNAME]) , 91–104.

[25] Mitchell, S., 2014. Converting the SWFTS program combat system product family from traditional document-centric to model-based systems engineering. Systems Engineering 14, 313–329. doi: 10.1002/sys.21271.

[26] Nadile, S., 2023. State of the Practice for MBSE at NASA GRC. Presentation 20230012952. NASA. URL: https://ntrs.nasa.gov/citations/20230012952.

[27] NASA, 2022. NASA Systems Modeling Handbook for Systems Engineering. Technical Report NASA-HDBK-1009. NASA Technical Standard.

[28] Naval Sea Systems Command, 2012. Ship Design Manager (SDM) and Systems Integration Manager (SIM) Manual. Technical Report S9800-AC-MAN-010.

[29] Noguchi, R.A., Minnichelli, R.J., Wheaton, M.J., 2019. Architecting success in model based systems engineering pilot projects, in: IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 755–760. doi: 10.1109/SMC.2019.8914417.

[30] OMG, 2017. Unified modeling language. Version 2.5.1. URL: https://www.omg.org/spec/UML.

[31] OMG, 2025. URL: https://www.omg.org/certification/sysml/.

[32] O'Rourke, R., 2023. Navy DDG-51 and DDG-1000 Destroyer Programs: Background and Issues for Congress. Technical Report RL32109 - Version: 270. Congressional Research Service.

[33] OUSD(R&E), 2022. Systems Engineering Guidebook.

[34] Page, J., 2022. A case for continuous concept development in ship design. Naval Engineers Journal 134, 67–73.

[35] Perez-Martinez, J., Perez Fernandez, R., 2023. Material and production optimization of the ship design process by introducing cads from early design stages. Journal of Marine Science and Engineering 11. doi: 10.3390/jmse11010233.

[36] Program Executive Office Ships (PMS 500), 2019. DDG 1000 Zumwalt Class Destroyer (DDG 1000). Technical Report. Defense Acquisition Management Information Retrieval (DAMIR).

[37] Schenker, A.R., Guertin, N.H., 2021. Using value engineering to propel cyber-physical systems acquisition. Naval Engineers Journal 133, 65–75.

[38] Sparx Systems, 2024. Welcome to the enterprise architect user guide. Webpage by Sparx Systems. URL: https://sparxsystems.com/enterprise_architect_user_guide.

[39] Specking, E., Parnell, G., Pohl, E., Buchanan, R., 2018. Early design space exploration with model-based system engineering and set-based design. Systems doi: 10.3390/systems6040045.

[40] United States Navy and Marine Corps, 2020. Digital Systems Engineering Transformation Strategy. Technical Report.

[41] Van Hentenryck, P., Michel, L., 2003. Optimization Software Class Libraries. Springer. doi: 10.1007/b101931.

[42] Yang, Z., Du, H., Liu, Y., Liu, R., Liu, Y., 2021. Use the harmony-SE approach to extend the advantages of MBSE, in: IEEE 16th Conference on Industrial Electronics and Applications (ICIEA), pp. 223–227. doi: 10.1109/ICIEA51954.2021.9516269.

[43] Yasseri, S., 2014. Application of systems engineering to subsea development. Underwater Technology 32, 93–109. doi: 10.3723/ut.32.093.

[44] Zimmerman, P., Gilbert, T., Salvatore, F., 2019. Digital engineering transformation across the Department of Defense. The Journal of Defense Modeling and Simulation 16, 325–338. doi: 10.1177/1548512917747050.

Authors

Lisa Marion is employed in the public sector as a Lead MBSE engineer and has previously supported a ship program as the lead digital engineer. She has supported various other DoD programs in her career as a test and systems engineer, from missile systems to fixed-wing aircraft to autonomous and destroyer-class ships. Ms. Marion is a Ph.D. Candidate in Systems Engineering at Colorado State University.

Daniel R. Herber is an Associate Professor in the Department of Systems Engineering at Colorado State University in Fort Collins, CO, USA. His research interests and projects have been in design optimization, model-based systems engineering, system architecture, digital engineering, dynamics and control, and combined physical and control system design (control co-design), frequently collaborating with academia, industry, and government laboratories. His work has involved several application domains, including energy, aerospace, defense, and software systems. He teaches courses in model-based systems engineering, system architecture, controls, and optimization. He is a member of INCOSE, ASME, and AIAA. He studied at the University of Illinois at Urbana-Champaign, earning his B.S. (2011) in General Engineering and his M.S. (2014) and Ph.D. (2017) in Systems and Entrepreneurial Engineering. He held a postdoctoral position (2018-2019) with the NSF ERC for Power Optimization for Electro-Thermal Systems (POETS).