SYM-AM-21-070

Excerpt from the Proceedings

of the

Eighteenth Annual Acquisition Research Symposium

A Study of MBSE Through the Development of Modeling and Data Exchange Processes

May 11–13, 2021

Published: May 10, 2021

Acquisition Research Program
Graduate School of Defense Management
Naval Postgraduate School

The research presented in this report was supported by the Acquisition Research Program of the Graduate School of Defense Management at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

Acquisition Research Program
Graduate School of Defense Management
Naval Postgraduate School

A Study of MBSE Through the Development of Modeling and Data Exchange Processes

William Emeny—graduated from Virginia Tech with a bachelor's degree in mechanical engineering and from the Naval Postgraduate School (NPS) with a master's degree in systems engineering. Emeny supports the Model Based Product Support NAVSEA program of record at the Naval Surface Warfare Center in Port Hueneme, CA. Emeny leads the effort to implement model-based systems engineering concepts and tools to support model driven design objects and to manage program requirements. Emeny is very active in the Department of Navy's MBSE community; he is an active member of the Naval Digital Engineering Body of Knowledge (NDEBOK) and the MBSE community of interest. [william.emeny1@navy.mil]

Ryan Robar—started his professional career with the Department of Homeland Security. From the Department of Homeland Security, he transitioned to Naval Sea Systems Command. Robar has spent the last 7 years holding different positions within supportability of the AEGIS Weapon System. Robar recently transitioned to focus his professional career to the cradle-to-grave supportability and integration of the Missile Defense Agency's Sea Based Weapon System Ballistic Missile Defense capabilities within the AWS. Robar is a recent graduate of NPS with a master's degree in Engineering Sciences and holds a bachelor's degree from California Lutheran University in Business. [ryan.robar@navy.mil]

Dustin Talley—is a Mechanical Engineer working at the Naval Surface Warfare Center, Port Hueneme Division. Talley started his career in 2015 working on research and development efforts related to renewable energy. In 2017, he transitioned his career and began working in test and evaluation. His primary role is planning and executing developmental testing of Directed Energy Weapons for naval platforms. Talley graduated with a Bachelor of Science in Mechanical Engineering from California State University, Long Beach. Talley also recently graduated with a Master of Science in Systems Engineering from NPS. [dustin.talley@navy.mil]

Michael Rubow—is an Engineer who works at the Naval Surface Warfare Center, Port Hueneme Division (NSWC PHD) as part of the In-Service Engineering Agent (ISEA) branch for the Harpoon and Over-The-Horizon Weapon Systems (HWS/OTH-WS). He earned his Master of Science degree in Systems Engineering from NPS in December 2020 and his Bachelor of Science degree in Mechanical Engineering from the University of California, Merced (UCM) in May 2014. Rubow enjoys participating in outdoor adventures such as hiking, skiing, and beach activities with his wife Caitlin. [Michael.rubow@navy.mil]

Lance Lowenberg—is a Robotics Engineer at the Naval Information Warfare Center (Pacific) who specializes in unmanned systems development. In addition to his work, he enjoys restoring classic cars and tinkering with other mechanical projects. He and his wife love camping, sailing, hiking, and traveling with their currently one but soon to be two furry four-legged pals. Lowenberg has a BS in Mechanical Engineering from UC Santa Barbara and an MS in Systems Engineering from NPS. [lance.lowenberg@navy.mil]

Lynn Nguyen—is a logistician at the NSWC PHD supporting readiness engineering for the Program Executive Office for Unmanned and Small Combatants (PEO USC) for the Littoral Combat Ship (LCS) program. Nguyen is involved in efforts to support system reliability, maintainability, and availability for LCS. She has a background in Integrated Product Support, DMSMS management, and task management. She completed her undergraduate education at UCSD and graduated from NPS with an MS in Engineering Systems. [lynn.p.nguyen@navy.mil]

Abstract

The DoN is undergoing a digital transformation that is set to address the needs of sustaining fleet assets for extended periods of time, while maintaining a superior lethality. Within the engineering domain, the DoN is starting to identify model-based systems engineering (MBSE) tools and concepts to streamline processes and enhance capability. The capstone looked to lay the foundation for a conceptual system model development process that utilizes SysML and OOSEM

to produce system model data and artifacts derived from a single scenario. During the digital transformation, communication of system model data to stakeholders was identified as a need, and a SysML tool was used to generate model-based documentation from a formatted Microsoft Word document. With incoming digital product support capabilities from the MBPS program, communication from a MBSE environment is critical and requires XML-formatted data. Using the information collected in the completion of the scenario, it was discovered that SysML elements will lose their SE-specific stereotypes when converted directly into XML format. To counter this, the capstone developed UML instances derived from the S3000L UML class-based data model to be converted into XML format. The findings and developments of this capstone support the ability for organizations to standardize the way system modeling data is developed, collected, and communicated to other systems external to the engineering domain.

Executive Summary

Currently, there is an initiative to transform legacy logistics information technology (IT) systems to use a model-centric approach to support products that aims to increase system uptime and reduce support costs. Model Based Product Support (MBPS) is a single piece of a larger digital readiness vision that includes new capabilities, such as predictive analytics, data-as-a-service, platform-as-a-service, process automation, and the integration of data across multiple platforms (SEA06L, 2019). This vision of a logistics digital transformation is shown in Figure 1. The new integrated product life-cycle management (PLM) platform supports the sharing of a standardized data model that enables the capability to perform logistics support analysis. The PLM platform inside the product support (PS) domain would have conduits with the engineering, maintenance, training, and other system life cycle communities to support better logistics models and better supported systems (NSRP, 2019).

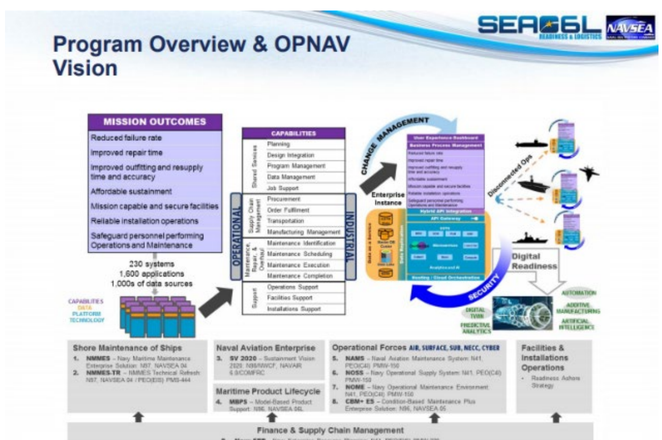

Figure 1. Logistics Digital Transformation Vision Overview (NSRP, 2019)

The current transformation occurring in the PS domain is also being pursued within the engineering domain with the exploration and implementation of model-based systems engineering (MBSE) concepts. Department of Defense (DoD) strategic documents have expressed the need that as systems become more complex, the DoD will require more robust engineering practices to develop weapon systems and maintain superiority over our enemies (Engineering, 2018). For many years, the DoD has relied on document-based, stove-piped engineering processes and is now looking to incorporate digital engineering practices to work more efficiently. The incorporation of digital engineering will require investment in new methods, processes, and tools in order to enable systems to become more lethal and affordable (Engineering, 2018). The Department of Navy (DoN) has embraced the goals set by the DoD Digital Engineering Strategy by developing its own set of high-level strategic documentation that discusses high-level implementation strategies and their alignment to the DoD documentation (DoN, 2020).

One of the alignment goals set in the DoD Digital Engineering Strategy and envisioned in the DoN Digital Systems Engineering Transformation Strategy is the formalization of the development, integration, and use of models. Using the system modeling language (SysML) and SysML tools, the capstone group built a conceptual system model development process based off the object-oriented systems engineering methodology (OOSEM). The OOSEM is a top-down, scenario-driven approach that leverages object-oriented concepts and other modeling techniques to support in the development of a more flexible and extensible system architecture that can accommodate the constant change in requirements or technologies (Friedenthal et al., 2012). The developed process encapsulates system modeling data within what is known in SysML as blocks, analogous to classes within the unified modeling language (UML).

The conceptual system modeling process was developed, and an example scenario was completed in which an organization has a need to develop and implement a model-based system engineering environment; henceforth named the Digital Engineering Environment (DEE), locally within the organization. The scenario walks through the development of the conceptual system model and pieces of the logical system model prior to a request for proposal (RFP) where vendors would bid on to develop a physical product based off the information presented to the vendor in the conceptual system model. The conceptual data model, shown in Figure 2, displays the type of models and artifacts that make up the system model and how they contribute to the development of the system of interest. The information and artifacts captured in the data model are developed within the system modeling process described in this capstone report.

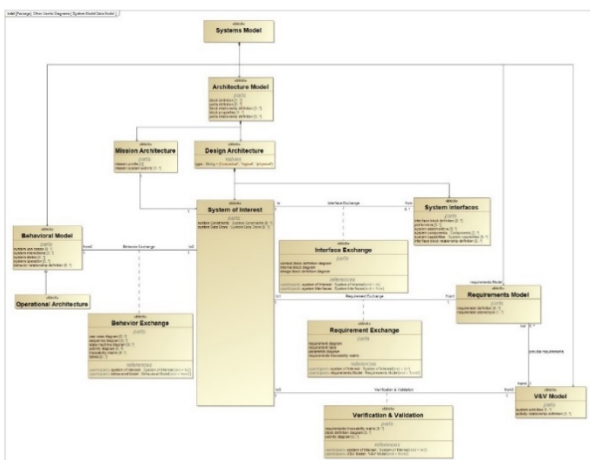

Figure 2. Proposed Conceptual Data Model for Developing a Systems Model

System model data collected over the design and development phases of a system must be capable of being consumed and of use to the PS domain to enable the reuse of system data for supportability analyses. The MBPS program overview presentation displayed the program's use of the S-Series specifications developed by the AeroSpace and Defense Industries Association of Europe and Aerospace Industries Association (ASD/AIA). These specifications layout an extensible markup language (XML) schema with data classes useful for different types of PS efforts, including provisioning, maintenance task analysis (MTA), level of repair analysis (LORA), software support analysis, and other logistics support analyses. There is not a current mapping between the data elements within SysML to the UML data elements within the S-Series specification; however, the developers of the specifications have developed a data model, which can be consumed and useful to a model developed in a SysML toolset. As shown in Figure 3, element instances contain the useful PS data which, if contained within an isolated model, could be manually translated into XML and exported to the S-Series database for analysis use.

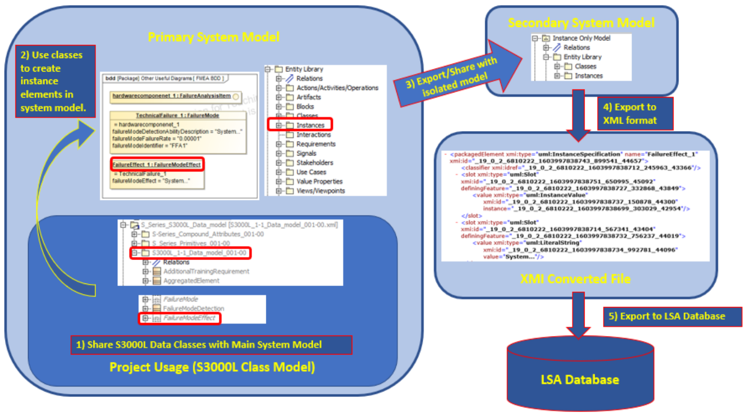

Figure 3. SysML Instances Translated into XML File Format for MBPS Consumption

Stakeholders were interested in verification of the system design which was supported with the presentation of system model data. Many stakeholders do not have experience in using system modeling tools but are familiar with many of the presentation formats within the model. Many system modeling tools have the capability of developing model-based documentation. Some of the presentation views within the developed model for the capstone's scenario were utilized to develop a model-based concept of operations (CONOPS). The CONOPS document template was downloaded from public online sources and configured using the velocity template language (VTL) to place model presentation artifacts into the CONOPS, automatically, upon a click of a button (Department of Veteran Affairs [VA], n.d.).

The model building process does explain the development of a conceptual data model but describes very little work on the development of a logical system model and does not approach the physical model development phase. More research is needed to understand the interfaces with other digital engineering tools and how related data can be used to further define certain aspects of the system model. The process completed a scenario in which useful products were developed for demonstration. To ensure its validity, verification and validation of the proposed process should occur using pilot projects to identify and fix any demonstrated gaps within the process. Future work should include the implementation of another scenario in which a fielded system wishes to undergo a system change. This scenario would require the system model to be updated and used to perform alternative analysis in both the engineering and PS domains.

The resulting scenario provided a collection of data points that represents different SOI viewpoints and that could be used within alternate domains to perform analyses. The conceptual system model in this instance would solely be used to demonstrate a problem and need to a design team or vendor. The instance of a problem would be derived from the technical capability audit (TCA) within the developed process whose following steps would be used to collect data and build presentation views. With the emergence of system of systems (SoS) modeling, it is theorized that existing and anticipated emerging gaps could also be a source of problems in which a TCA could be utilized to determine the necessary solution type (Mohammadi et al., 2014). Future work could explore the use of a TCA to identify future capability gaps as a second scenario to validate SysML models presented in this capstone. Using SysML and tailoring a process derived from the object-oriented system engineering methodology (OOSEM), enabled the encapsulation of system model data into a single SOI model element to communicate a design's architecture, behavior, requirements, and verification and validation activities. Review of the data developed during the simulation and the S3000L data model shows that there is a need for engineering data (Aerospace and Defense Industries Association of Europe and Aerospace Industries Association [ASD/AIA], 2014). The capstone presented a way to translate information from SysML into XML, but more work is needed to develop a data mapping to the S3000L XML data model that could lead to an automated conversion process.

Introduction and Background

Introduction

This project demonstrates a process that gives United States Navy (USN) organizations the capability to develop a conceptual system model, whose data can be used to initiate digital twin and digital thread capabilities. The process outlined in the appending pages is meant to be the foundation for creating the conceptual data model that would be created and matured over the life cycle of the system. This process utilizes the early steps of the object-oriented systems engineering methodology (OOSEM) approach, a model-based system design approach, as a guide in its design with the expectation that it will be used to assist Department of Navy (DoN) organizations in better defining and presenting conceptual system needs and requirements to design agents (Friedenthal et al., 2012). Process gaps within OOSEM were identified and tailored to better suit the needs of our stakeholders. For example, the project implements a data-driven approach to problem definition, something that is not included in OOSEM. To fulfill this capability, the technical capability audit (TCA) was added to the process. The TCA uses both quantitative and qualitative data from questionnaire or survey data to determine the type of problem the organization is facing (Mohammadi et al., 2014). Appended sections further expand upon this with the descriptions and applications of the technical capability audit (TCA) to perform problem analysis and parametric modeling for engineering analysis. At the conclusion of specified steps in the presented process, the modeler will have gathered enough data to enable the development of presentation artifacts. The systems modeling language (SysML) was utilized as the data model, while Cameo Enterprise Architecture (CEA) was used to produce SysML presentation artifacts. The produced artifacts were used as the process verification method and was performed using a generalized scenario, performing the outlined steps to create data points and artifacts that can be used to present to the system's stakeholders or to provide information to external systems in order to enable their own capabilities. The report will discuss the steps and artifacts developed through each step of the developed process. A discussion will follow that demonstrates potential uses for the data to support the development of acquisition documentation and the analysis of data communication with systems external to the systems engineering boundary.

Problem Overview

The Department of Defense (DoD) produced the DoD Digital Engineering Strategy to help spark and align a digital transformation in the engineering community. More recently, the DoN and Marine Corps delivered Digital Systems Engineering Transformation documentation that describes the goals for model-based systems engineering (MBSE) and lays a framework for MBSE implementation (DoN, 2020). Currently, MBSE is still immature relative to model-based product support and the enterprise technical reference framework (ETRF) and a fully matured enterprise capability may be some time off. In this scenario, it is assumed that the need for better, faster, and centralized tools and process in the system engineering community has been identified, and MBSE is the identified solution. With MBSE being as immature within the enterprise as it is, the DoN is still researching for more information on the MBSE subject and trying to identify how it will best be implemented alongside the product support digital transformation. There is not yet a formal standard set of processes, models, data and tools at the DoN enterprise level that align to all of the objectives in the Digital Engineering Strategy and local commands are beginning to develop their own local instances of MBSE environments. The lack of standardization of the processes, data formats and exchanges may lead to systems again becoming isolated and less efficient as their potential.

Background

As systems experience a never-ending increase in complexity, rapidly changing operational and threat environments, increased budget constraints, and more demanding schedules, the DoD needs more robust engineering practices. Current engineering processes are often document intensive and stove-piped. To meet their needs, the DoD is transforming its engineering practices to a digital engineering methodology utilizing model-based approaches, including MBSE (Engineering, 2018). MBSE is a subset of digital engineering and can be defined as the use of models to support the activities within systems engineering (SE) process, including requirements, architecture, design, verification, and validation (Giachetti, 2020). The implementation of MBSE has been theorized to enable new capabilities within the SE process (DoN, 2020). One of the primary objectives of implementing MBSE is to develop an integrated set of digitally integrated views that enables the capability of automating the engineering assessment of proposed designs. This automated capability would be able to identify risks and gaps through the simulation of operational scenarios. The digital environment would provide feedback data to enable the application of data-driven decision-making.

To maximize the effectiveness of MBSE, an organization must find a cohesive set of modeling tools and methods. The process supporting these activities is laid out in the implementation of OOSEM, applying SysML as the model syntax. The OOSEM is a top-down, scenario-driven approach that leverages object-oriented concepts and other modeling techniques to support in the development of a more flexible and extensible system architecture that can accommodate the constant change in requirements or technologies (Friedenthal et al., 2012). The activities within the OOSEM process reflect those of the fundamental SE process, including needs analyses, requirements analyses, architecture design, trade studies and analyses, and verification (Friedenthal et al., 2012). The primary output to the OOSEM process is a model of the system of interest (SOI). The collected data on the SOI is captured and encapsulated using a SysML block, an extension of the Unified Modeling Language (UML) class that includes allocated system elements describing different system views. This project explored a system's architecture, behavior, requirements, and verification and validation (V&V) views. Each view contains a set of SysML diagrams, matrices, or tables to create a model of each system model view. These diagrams are presentation mechanisms to display different data sets of the system model to different stakeholders.

Digital transformation inside the DoN is not only an interest within the engineering domain, but within the entire enterprise. The DoN has a vision for digital transformation, and it has begun in the logistics IT domain with the implementation of the ETRF. The ETRF vision will provide a framework that will generate scalable, interoperable, flexible, and fluid technology solutions that will provide access to information and data at anytime, anywhere. One of the major capabilities of the ETRF is the implementation of an integrated platform as a service (PaaS) environment that will unify all logistics applications internal to the ETRF system and will deploy a set of application programming interfaces (API) to integrate with future and legacy systems. The vision of the ETRF will contain many logistics applications that will be managed by the PaaS. Applications within the ETRF will fall into one of the following four key mission areas: integrated readiness, supply chain management (SCM), maintenance, repair, and overhaul (MRO), or product life-cycle management (PLM; Accenture, 2019).

There are currently two major programs sponsored by the Office of the Chief of Naval Operations (OPNAV), Model Based Product Support (MBPS) and Navy MRO (NMRO), that are developing the applications to meet the objectives of these mission areas. These applications will be developed to deploy new methodologies, including model-based approaches, and replace legacy systems with new systems that utilize digital tools and processes to replace the old capability set. One of these programs is MBPS, which spans across all four of these mission areas and is of special importance to this project. MBPS is an initiative within the Naval Sea Systems Command (NAVSEA) with cooperation from the Program Executive Office (PEO) that will create and implement a digitally integrated environment focused on the support of Naval systems. The MBPS environment will support the production of many artifacts in the support of sustaining engineering, including reliability centered maintenance (RCM) artifacts, level of repair analysis (LORA), readiness at cost analysis, reliability block diagrams, fault tree analysis (FTA), and other product support documentation and analyses. An authoritative source of product support data, that will enable the supportability analyses listed above. The authoritative data structure will be established and MBPS and developed using industry standards to support the communication and exchange of data between systems internal and external to the MBPS environment (SEA06L, 2019). The integration of MBSE and MBPS is of great interest. It has been theorized that this integration could lead to systems that maximize availability, effectiveness, capability, and affordability (Kwon et al., 2018).

In order to perform cross-platform verification and analysis, data must be accessible by both environments through an authoritative data source. Currently, there are two identified potential authoritative data sources within the ETRF that are being sponsored for development. Within MBPS, there is the Navy Product Data Management (NPDM) that is being established as the authoritative data source for all system technical data once a system reaches the operation and sustainment phase of the system's life cycle. The ETRF will also be deploying the agile warfighter analytics readiness environment (AWARE) within NMRO. The AWARE is a data-as-a-service (DaaS) platform to manage and communicate maintenance data from data collected by ship-based NMRO applications to the AWARE. Any data needed by the applications will be stored and transferred through at least one of these data sources. For MBSE, this has been identified as a major integration point between SE and product support (PS) capabilities which, in the future, will communicate and supplement the capabilities of one another (Accenture, 2019).

Problem Statement

The USN has produced documentation describing the characteristics of a model-based engineering environment but has not yet realized a solution for a model-based engineering environment and how that environment would be implemented and integrated into the system of systems (SoS) enterprise digital transformation vision (DoN, 2020). A need has been identified by the SE community at the Naval Sea Systems Command, Port Hueneme Division (NSWC PHD) to implement a local model-based system engineering (MBSE) environment and to understand how the MBSE data set, capabilities and tools would integrate into the ETRF.

With the MBPS capability set being more mature than the MBSE capability set, this capstone looked to identify potential avenues of implementation that aligned to the high-level objectives within the DoD and DoN strategic documents. With the development of a standard modeling process, the standardization of data sets, presentation artifacts, tool sets, etc. will follow, enabling many of the MBSE capabilities. A standardized set of data of system model data will enable external boundary communication and the development of model-based documentation.

Project Objectives

This capstone team had two high-level objectives: develop a formal process using SE methodologies that would be capable of developing a conceptual system model and compile a final report that will explain the problem space, describe the solution space and how it solves identified issues, describe and explain the processes used, present the developed artifacts, and provide recommendations for future work or action.

The objective of the model is to provide a standard process for organizations to develop a conceptual system model that contains early system architecture, behavior, requirements, and verification and validation models. The conceptual model would be the starting point for a program's digital twin and thread that would mature along with the design to include data from the logical and physical levels of the design. The process and development of system model data enables the capability of producing model-based documentation that supports the development of programmatic documentation from templates. The report will demonstrate and explain the process of how the capstone team developed and produced a model-based concept of operations (CONOPS) from a Microsoft Word template found in the public internet domain.

To ensure the process satisfies the stakeholder objective and requirements, the capstone team applied the process to a development scenario to support the verification of the process. The model will be supplemented by a textual report that will further include explanations of the processes and recommendations for future work.

Project Scope

The scope of the capstone is set based on the scenario outlined in the Problem Overview section. Verifying the developed process with these scenarios will produce a set of artifacts that will be used to demonstrate to organizations how MBSE can be used. The documented process and developed artifacts are a part of the framework of this report, and the discussions that follow will be based off the development of the system model and the verification methods using the use case scenarios.

Section Summary

This section introduced the capstone, overall problem, background information, problem statement, scope, and objectives. This information is used in the understanding of the information and processes that will be discussed in the appended sections.

Having identified the need to utilize MBSE concepts to enhance the DoN's engineering capabilities, the capstone team documented a standard process. The process is used to support an organization's capability to develop conceptual system models. The process was developed using the object-oriented SE methodology as a guide as to what data is required for the development of the system model and the presentation artifacts were produced using Cameo Enterprise Architecture. To provide examples of artifacts to the stakeholders and this report, a fictitious scenario was applied. The appending sections will provide more detailed explanations for each phase of the process and the artifacts that are consumed and produced by each phase.

Problem Analysis

Important Definitions and Terms

Common definitions and terms are used throughout this report. These definitions were researched and established during the literature review. These terms are defined in this section to give the reader a general understanding of the topics to be discussed.

Model-Based Systems Engineering

The use of models to convey SE concepts and data either in place of or in conjunction with traditional textual methods has gained wide acceptance in recent years. This was introduced by INCOSE in 2007 as follows:

> 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. MBSE is part of a long-term trend toward model-centric approaches adopted by other engineering disciplines, including mechanical, electrical and software. (INCOSE Technical Operations, 2007)

Model-Based Product Support

There is no official definition of model-based product support (MBPS) in literature, but the collective sources support a general definition. Model-based product support is a broad term that essentially translates to model-based electronic tools and information systems that enable the support of logistics functions such as training, maintenance, operations, and sustainment. Model-Based Product Support is the cooperative initiative between NAVSEA and PEO that will provide multiple digital logistics capabilities to the DoN.

Architecture Framework

The Architecture Framework defines how an architecture will be created and subsequently utilized through a set of rules and practices. It is defined by the MITRE Corporation as follows:

> An architecture framework is an encapsulation of a minimum set of practices and requirements for artifacts that describe a system's architecture. Models are representations of how objects in a system fit structurally in and behave as part of the system. Views are a partial expression of the system from a perspective. A viewpoint is a set of representations (views and models) of an architecture that covers a stakeholder's issues. (MITRE, 2015)

Enterprise Technical Reference Framework

The push to consolidate existing systems into a new common logistics platform that leverages new technologies and innovations is necessary in order to adapt to the Navy's changing needs. The following two quotes describe this:

> The vision of Enterprise Technical Reference Framework (ETRF) is to enable and accelerate the overall objective of Navy Logistics IT. ETRF provides a digital logistics IT architecture that will generate scalable, interoperable, flexible and fluid technology solutions; maximizing access to information/data via applications anywhere, on any device at any time. (Accenture, 2019)

> The Enterprise Technical Reference Framework will leverage the Digital Transform Plan Services, Data, Technology, Security and Change Management strategies to provide a framework and roadmap to transform 1600+ current Applications and 5000+ data sources to a common unified logistics IT platform. (Accenture, 2019)

Digital Information Technology (IT) Transformation

The DoD digital IT transformation exists within the Joint Information Environment (JIE) framework that is comprised of a comprehensive Department-wide IT modernization that exists within the DoD Information Network (DoDIN). The JIE purpose is to "improve mission effectiveness, increase cybersecurity, improve interoperability, deliver capabilities faster, and realize IT efficiencies" (DoD, 2019). The DoD JIE framework is comprised of 10 Capability Objectives, as shown in Table 2.

Table 2. Alignment of DoD CIO Objectives to JIE Capability Objectives and Initiatives (DoD, 2019)

JIE Capability Objective	JIE Initiatives	DoD CIO Objectives
Modernize Network Infrastructure	Optical Transport Upgrades, MPLS Routers Buildout, ATM Switch and low speed TDM Circuit Elimination, Satellite Communications Gateway Consolidation and Modernization, IPv6 Implementation	• Modernize Warfighter C4 Infrastructure and Systems • Modernize DISN Transport Infrastructure • Modernize and Optimize DoD Component • Networks and Services
Enable Enterprise Network Operations	Establish global and regional operations centers, Establish the JIE Management Network, Converge IT Service Management (ITSM) solutions	• Modernize and Optimize DoD Component Networks and Services • Shift from Component-Centric to Enterprise-Wide Operations and Defense Model
Implement Regional Security	JRSS, JMS	• Modernize DISN Transport Infrastructure
Provide Mission Partner Environment (MPE)	Virtual Data Center, Application and Services, MPE Transport, Mission Partner Gateways	• Strengthen Collaboration, International Partnerships, and Allied Interoperability
Optimize Data Center Infrastructure	Data Center Optimization Initiative (DCOI) and Application Rationalization Initiative	• Optimize DoD Data Centers
Implement Consistent Cybersecurity Protections	Enterprise Perimeter Protection Capabilities, Operate Securely in the Cloud, Endpoint Security, Data Center Security, Cyber Situational Awareness Analytic Capabilities (CSAAC)/ Big Data Platform (BDP), Identity, Credential, and Access Management (ICAM)	• Transform the DoD Cybersecurity Architecture to Increase Agility and Strengthen Resilience • Deliver a DoD Enterprise Cloud Environment to Leverage Commercial Innovation • Deploy an End-to-End ICAM Infrastructure
Enhance Enterprise Mobility	Purebred for Mobile, Defense Enterprise Mobility-Classified Consolidation, DoD Mobile Application Store, Pentagon Mobility	• Improve Information Sharing to Mobile Users
Standardize IT Commodity Management	Enterprise Software Agreements, Enterprise License Agreements, Enterprise Hardware Agreements, IT Asset Management, Windows 10 SHB Fourth Estate Network Optimization	• Improve IT Category Management • Transform the DoD Cybersecurity • Architecture to Increase Agility and Strengthen Resilience
Establish End-User Enterprise Services	Enterprise Collaboration and Productivity Services	• Optimize DoD Office Productivity and Collaboration Capabilities (ECAPS Capability Set 1) • Optimize DoD Voice & Video Capabilities (ECAPS Capability Sets 2 & 3)
Provide Hybrid Cloud Computing Environments	Cloud Services	• Deliver a DoD Enterprise Cloud Environment to Leverage Commercial Innovation • Optimize DoD Office Productivity and Collaboration Capabilities (ECAPS Capability Set 1) • Optimize DoD Voice & Video Capabilities (ECAPS Capability Sets 2 & 3)

Systems Modeling Language (SysML)

The Systems Modeling Language (SysML) is a general purpose MBSE language that uses "graphical modeling for specifying, analyzing, designing, and verifying complex systems that [include] hardware, software, information, personnel, procedures, and facility elements" (Object Management Group, n.d). SysML originated from the Unified Modeling Language (UML) 2 framework. Further, SysML "provides graphical representations with a semantic foundation for modeling system requirements, behavior, structure, and parametrics, which is used to integrate with other engineering analysis models" (Object Management Group, n.d).

Section Summary

This section discussed problem analysis to include stakeholders, definitions, and a literature review. Definitions were introduced to familiarize the reader with MBSE and MBPS and the environment they operate within. Policies such as the ETRF and digital IT transformation explain how DoD policies affect both modeling areas. A list of stakeholders was presented that explained their functional area, their relationship to this project, and how they are impacted by this project. The literature review familiarized the capstone group with modeling efforts within SE and product support. The literature review presented an overview of definitions and applications of MBSE and MBPS. Furthermore, the literature reviewed focused on DoD specific applications of modeling, including

- The USN's legacy process being used at the time of this capstone.
- The capability gaps of the legacy processes.
- Future DoD-specific modeling trends in MBSE and MBPS.

Model Development

System Model Development Process

The system model development process, shown in Figure 4, was developed to establish a standard procedure in developing conceptual system models early in a system or project's life cycle. The model development process was created using the OOSEM as a guide for the phases within the process. The process begins in the problem definition and analysis phase where the problem was defined with stakeholder concurrence and analysis to determine a recommended solution. A decision is then made based on the maturity of the solution to either integrate the existing solution set, if it is mature enough, or to develop a solution if one does not exist or is too immature. For this report is assumed that the decision has already been made that an immature solution will be pursued in the local implementation of the Digital Engineering Environment (DEE).

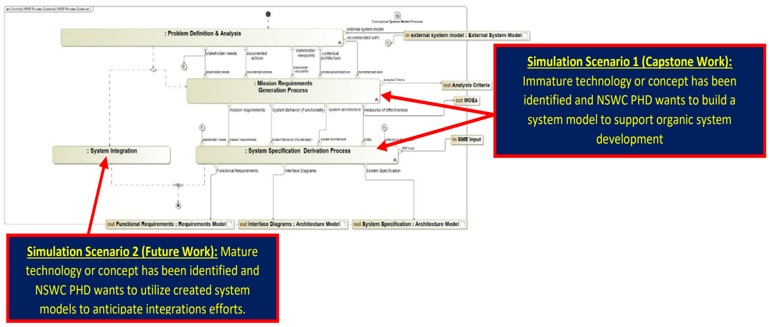

Figure 4. Overall System Model Development Process

During the model development process, the capstone team sought to meet the project objectives by utilizing the scrum framework with an iterative development approach. Team members were individually assigned diagrams through the sprint planning process. Completed diagrams were peer reviewed for content, flow, and formatting, and then were added into the master model. Periodic stakeholder reviews, including progress reviews, were conducted to gather feedback; feedback influenced model design and development to meet the stakeholder needs. Simulations of the model using a designed scenario were also performed iteratively throughout the development process to produce model data and artifacts.

Simulation Scenarios

The basis for the selection of the simulation scenarios and the corresponding activity diagrams were determined by the project objectives. Stakeholder analysis and the sponsor command objectives played a key role in the selection of the example scenarios to represent model function. The sponsor's prime objectives for in-service engineering played a key role in the selection of the following scenarios:

Addressing new business capabilities (Simulation Scenario): A new incoming business capability has been identified; or the command performs an internal audit which identifies a desired new capability. The capability set is immature, and there is not an existing system infrastructure that supports the capabilities. A system model is to be built from scratch to present conceptual information and high-level requirements of the desired solution. Post model development, the system model would be distributed to a development team for to be updated and refined as the system supporting the capability matures.

Addressing new capabilities to an existing system (Future Simulation Scenario): This scenario would focus on the addition of a capability set to an already existing system. A system model or system of systems (SoS) model exists and would be utilized to perform alternative analysis on the change prospects. Updates to the system model would happen iteratively as the change design matures and is implemented.

Activity diagrams were derived from these use cases. The pertinent activity diagrams were identified by determining the key aspects that affect the example scenarios. The activity diagrams that were modeled were:

- Problem Definition and Analysis
- Mission Requirement Generation Process
- System Requirements Generation Process
- System Integration

Object Oriented Systems Engineering Method

This capstone has utilized elements of the Object Oriented Systems Engineering Method (OOSEM) found within the practical guide to SysML. “[The] OOSEM is a top-down, scenario-driven process that uses SysML to support the analysis, specification, design, and verification of systems. The process leverages object-oriented concepts and other modeling techniques to help architect more flexible and extensible systems that can accommodate evolving technology and changing requirements” (Friedenthal et al., 2012). The OOSEM was created in 1998 and has been further refined by an International Council on Systems Engineering (INCOSE) OOSEM working group (Friedenthal et al., 2012). It is an INCOSE accepted SE management process. Most of the capstone artifacts have been captured using MBSE and SysML artifacts. These artifacts include stakeholder requirements, system requirements, problem space architecture, solution spaces architecture, use cases, and parametric diagrams. Due to the large nature of model-based artifacts, this capstone chose to employ elements of OOSEM due to its applicability in both SysML development and SysML enabled management. Figure 5 shows the OOSEM steps that helped this capstone team design a tailored process for developing a conceptual system model.

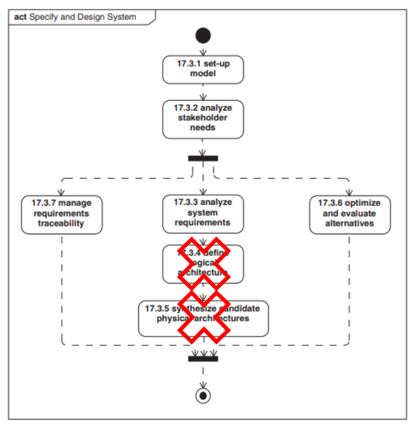

FIGURE 17.2

OOSEM *Specify* and *Design System* process. The action numbers refer to subsection where the action is described.

Figure 5. OOSEM Specify and Design Process (Friedenthal et al., 2012)

The SE steps shown in the figures are set-up model, analyze stakeholder needs, manage requirements traceability, analyze system requirements, optimize and evaluate alternatives, define logical architecture, and synthesize candidate physical architectures. This process was tailored to not include the optimize and evaluate alternatives, define logical architecture or synthesize physical architecture. These steps were removed as this capstone will not produce a full logical or physical system and would be up to the development team to refine the model to include the architecture definition. Instead, the focus will remain on developing a conceptual SysML model that describes the objectives laid out in the simulation scenario: The need of a MBSE environment that provides digital SE capabilities and can exchange meaningful data with other platforms within the digital transformation domain.

The model development utilized an iterative design process where incremental builds of the model were developed. These iterative builds incorporated a feedback loop to receive stakeholder input on the developed models. Stakeholder feedback has subsequently been incorporated into each iterative design of the model.

Problem Definition and Analysis

The problem definition and analysis phase, as shown in Figure 6, is meant to support the identification of the problem and need in a data-driven way, and to devise a solution that will help satisfy the needs. The process begins with a signal that triggers the first step in the process. The trigger can be scheduled or unscheduled, as in this process could be performed with a scheduled integrated product team (IPT) annually, every 6 months, etc., or it could be spontaneous, driven by innovation within the enterprise or based on direction provided by enterprise leadership.

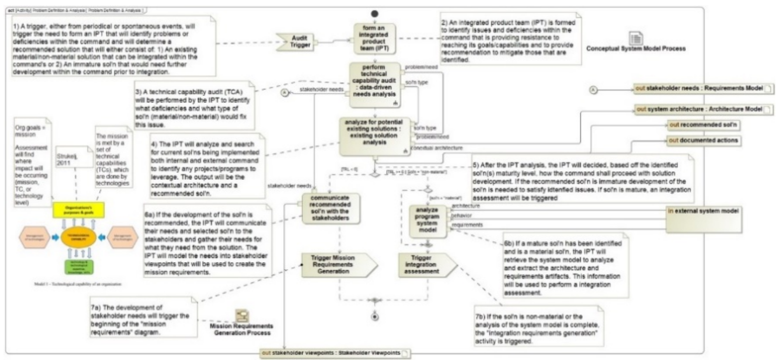

Figure 6. Problem Definition and Analysis Activity Diagram

Once the IPT is formed, their first responsibility would be to perform the technical capability audit (TCA). The TCA is the process of analyzing technical capabilities within an organization in a data-driven way to identify potential problems and solutions to those problems (Mohammadi et al., 2014, pp. 5–8). Technical capability in this context is defined as an organization's ability to utilize technologies in a way that is most useful to the organization's goals and mission. Technologies in this case refer to the machines and processes that the people of an organization utilize to perform their daily activities. Technology capabilities are influenced by technological innovation and changes in organizational goals or missions (Strukelj & Dolinsek, 2011).

The IPT develops a set of quantitative and qualitative indicators in which they can disburse to the workers of an organization to receive feedback. The indicators that form the TCA have four different aspects: hard, human, knowledge, and organizing and managing of technical capabilities. Hard aspects are the physical equipment, are the tools currently available to the workforce meet their needs. Human aspects relate to the skill set of the workforce and answers the question, "Does the workforce have the right skill set to perform this technical capability?" Knowledge aspects pertain to the understanding of the technological capability and is enough information known about it to make it a worthy investment. Lastly, organizing and managing of the technical capability is an aspect that focuses on how well an organization is structured, or funded, to develop new technical capabilities and the quality of the technical capability management process (Mohammadi et al., 2014, pp. 8–12). Feedback to the IPT from the workforce on the indicators can support the identification of problem areas where a solution is needed in order to satisfy the technical capability (Mohammadi et al., 2014, pp. 13–14).

For this example scenario, it is assumed that the TCA has already occurred and the problem has been identified to be a lack of hard aspects that is causing the greatest deficiency in achieving a MBSE technical capability at the organization. Upon completion of steps 3 and 4, as shown in Figure 6, the IPT should have completed the development of the stakeholder analysis, viewpoints and contextual architecture presentation views. An example of the stakeholder analysis is presented in Figure 7, while examples of viewpoints and contextual architecture are shown in the following section in Figures 12 and 13, respectively. For the purposes of this capstone, a formal stakeholder analysis was not performed and the Unified

Architecture Framework (UAF) provided guidance on stakeholder composition, concerns, purpose and presentation methods (OMG, 2020).

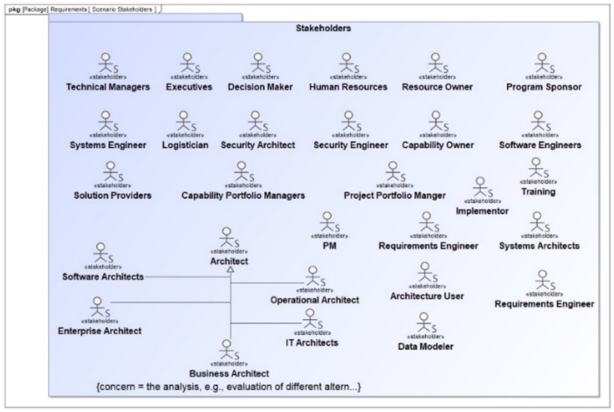

Figure 7. Analysis of the Stakeholders

Based on the maturity of the solution, the acquiring organization decided whether the system model developing process needs to begin at the beginning or a system model or artifacts exist, and the solution will be integrated into the model. For the capstone's scenario it is assumed that a new model describing the solution and the process will signal for the mission requirements generation process to begin. If the solution needs to be integrated into an already existing system model or SoS model, the model is consumed by the solution's model project and any updates and refinements are made as the solution matures.

Mission Requirements Generation

The mission requirements activity diagram in Figure 8 provides a set of top-level requirements and traceability to the other aspects of the system model to support the management of requirements throughout the completion of the process. This diagram will be explained by showing expanded views of each section to enhance readability and to summarize key actions occurring in each section.

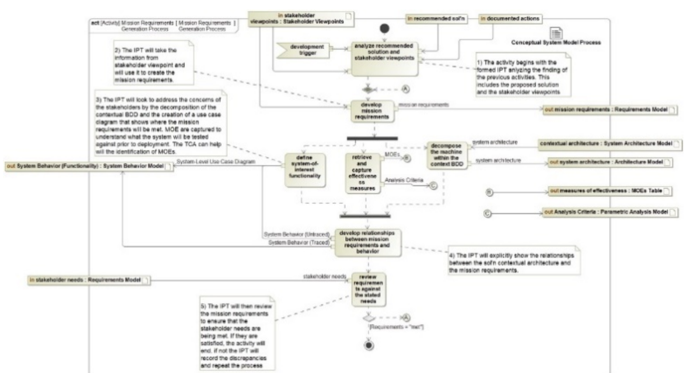

Figure 8. Mission Requirements Activity Diagram

The problem definition and analysis phase has several output object flows that are used as inputs to the mission requirement phase. The stakeholder viewpoints, recommended solution, and contextual architecture feed directly into various mission requirement blocks to further develop the system. The mission requirements diagram is the precursor to the system specification derivation process. As the precursor diagram, all outputs and generated artifacts are utilized in the system specification derivation process activity diagram.

The mission requirements generation process begins with a formed IPT analyzing the finding of the previous activities. The mission requirements phase initializes with the stakeholder viewpoints as well as the recommended solution from the problem definition and analysis phase. From the initialization, the IPT will enter a singular direction merge node, which allows for a repeat of the process should all requirements not be met. This merge node has no effect on the control flow of the process the IPT goes through from the initialization.

The control flow continues into the development of mission requirements. Mission requirements are built from the understanding of the problem and stakeholder needs that were established in the previous phase. From the development of mission requirements, the control flow then goes into a SysML fork where the IPT would perform three data collection tasks simultaneously. To exit the fork node the IPT must generate a block definition diagram (BDD) for system context, retrieve and capture measures of effectiveness, and decompose the machine within the context of the BDD.

The IPT will look to address the concerns of the stakeholders by the decomposition of the contextual BDD and the creation of the use case diagram that shows where the mission requirements will be met. The measures of effectiveness are captured to understand what the system of interest (SOI) will be tested against prior to deployment and implementation. The developed indicator from the TCA performed in the problem definition and analysis phase can be utilized to further strengthen the measures of effectiveness. From this block, the output of the BDD system context diagram is generated. This artifact is used to initialize the system specification diagram.

The last block within the fork requires the IPT to decompose the SOI within the context of the BDD. The object flow needed to complete this task is derived from the contextual architecture of the problem definition and analysis activity diagram.

With the satisfaction of the three proceeding taskings, the IPT control flow moves to join the control flows. The IPT will now be capable of defining the relationships between the solution contextual architecture and the mission requirements. As this development matures, the object flow output of a high-level system architecture transfers to the system specification derivation activity diagram.

The final logical control of the mission requirements activity diagram is to ensure that the stakeholders needs are being achieved. If gaps in requirements are identified, then the control flow allows for a repeat of the process flow for the IPT. The exit criteria for the mission requirements activity diagram is for the IPT to review the stakeholder requirements against the generated mission requirements If the stakeholder requirements are sufficiently satisfied the control flow exits the mission requirements activity diagram.

Mission requirements definition and refinement is an integral phase of the overall capability achievement of MBSE and/or MBPS within the digital engineering environment. The established object and control flows that this capstone project illustrates during the mission requirements activity diagram through the generated artifacts demonstrate the importance for an IPT to decompose and address the overall stakeholder need(s). The traceability aspect that OOSEM provides to the overall intent of the mission requirements diagram allows for further exploration of validation and verification that the system and component requirements satisfy the stakeholder requirements.

This part of the process was verified by the development of the input and output artifacts to ensure the required system model data was being produced, the following sections will discuss a selected number of these artifacts and will provide a short description pertaining to the artifacts importance to the overall presentation of the system model data.

Simulation Results

The process above describes an overall method for the second iteration of system model development. The process includes further refinement of the architecture facet of the system model, and it introduces the behavior and requirement viewpoints. In order to validate this method, scenario one was used as a use case and the system specification process was executed. The assumptions prior to moving into the process are that all required input artifacts have been completed from the previous activity diagrams. These input artifacts are displayed below as shown from the system context of the DEE and MBPS, where DEE is the SOI and MBPS is the identified external system.

Mission Requirements Generation Inputs

The first activity within the process requires the integrated product team (IPT) to revisit the information provided from the problem definition and analysis process. Other than the recommended solution, the IPT will be using the information provided in the stakeholder viewpoints as a guide to developing the different presentation views within the model. The stakeholder viewpoints represent different stakeholder perspectives and helps capture subsets of the model that are of interest to the stakeholder (Friedenthal et al., 2012). Shown in Figure 9 is the actual resources viewpoint. This viewpoint is of interest to a few different stakeholders, including the solution provider, business architect, human resources, and the systems engineer. Viewpoints capture stakeholder concerns and their preferred methods of presentation (OMG, 2020).

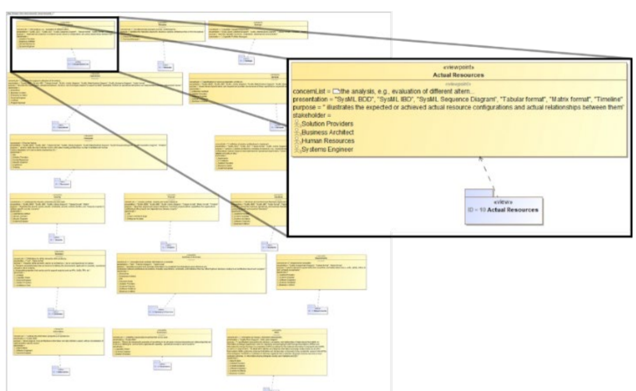

Figure 9. Stakeholder Viewpoint (Example)

Figure 10 displays another input from the problem definition and analysis phase that is used to help support the decomposition of the SOI. This artifact will define what is being decomposed, but the majority of the information needed to support the development would come from other programmatic artifacts, like a concept of operations (CONOPS), that would give the modeler a better idea of the necessary sub-systems or components needed to support the requirements for the SOI.

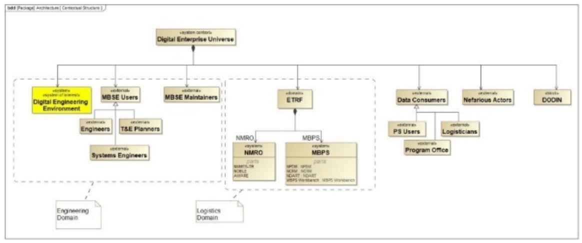

Figure 10. Contextual Architecture

Figure 11 represents an example of a set of stakeholder needs in diagram form. A diagram was chosen for this artifact, but a table is also an acceptable way for the same information to be displayed with the SysML syntax. The stakeholder requirements should always be the alignment mechanism during the development of systems and system models. SysML toolsets provide the platform for modelers to show stakeholders that their needs are being met and can provide traceable relationships to the modeled needs to ensure the designs are, in fact, meeting the modeled needs. An example of a requirement traceability matrix (RTM) is shown in Figure 12.

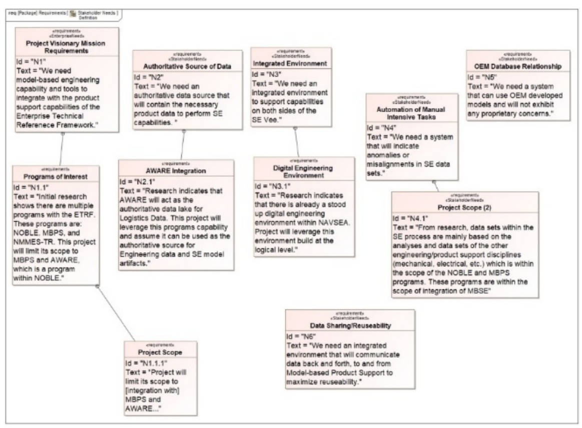

Figure 11. Stakeholder Needs Model

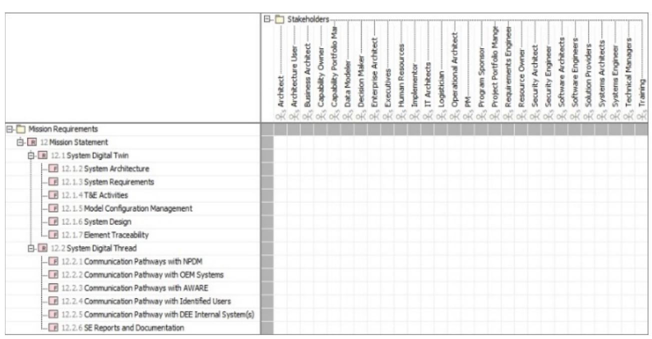

Figure 12. Example of a Requirement Traceability Matrix

Mission Requirements Generation Outputs

Measures of effectiveness (MOE) are captured in the model as shown in Figure 13. “[Measures of effectiveness] are mission-level performance requirements that reflect value to the customer and other stakeholders. They are derived from the stakeholder needs analysis that includes causal analysis and mission performance analysis” (Friedenthalet al., 2012). The MOEs help refine the black box behavior of the SOI by showing which properties and metrics are used to evaluate the system. For example, MOE 12 “required storage space” implies that the system must have a capability of storing data and that the size of the storage is important to the system final capability. The MOEs are also used in the mission requirements diagram to evaluate recommended system solutions.

#	Name	Applied Stereotype
1	bandwidth consumption	moe [Property]
2	development cost	moe [Property]
3	lifecycle sustainment cost	moe [Property]
4	past performance	moe [Property]
5	required storage space	moe [Property]
6	security factors	moe [Property]
7	software architecture	moe [Property]
8	software maturity	moe [Property]
9	staff profile	moe [Property]
10	staff turnover rate	moe [Property]
11	standard hardware compatibiliy	moe [Property]
12	system availability	moe [Property]
13	system redundency	moe [Property]
14	user capacity	moe [Property]
15	XML support	moe [Property]

Figure 13. Example MOE Table

Another function of the MOEs within the system model can be to create a criterion to which the program can base its decision-making. Shown in Figure 14 is a parametric diagram that provides an example of how parametric diagrams can be used in the design selection process. Contracting firms may submit bids to design the system laid out in Figure 14. The organization that sent out the request could use engineering analysis criteria in the parametric diagram, based off the modeled MOEs, to establish a plan for evaluating each submission. By placing a value on each MOE based on how well the contractor met that MOE, the evaluators will determine an overall score based on the selection criteria. As mission requirements have been developed within the model, the system modeler will look to begin the decomposition of the system architecture, based off the understanding of what is required of the system. The program or project is still in the very early stages in this scenario and there may be little information. Our simulation scenario from the overall process description is based off a set of known, but immature, concepts and capabilities. As shown in Figure 15, like-capabilities were grouped inside the capabilities boundary and assigned to the different capability areas, they could be called sub-systems, within our SOI. These capabilities would support the development of the top-level objectives, to create model and document artifacts that reflect the SOI.

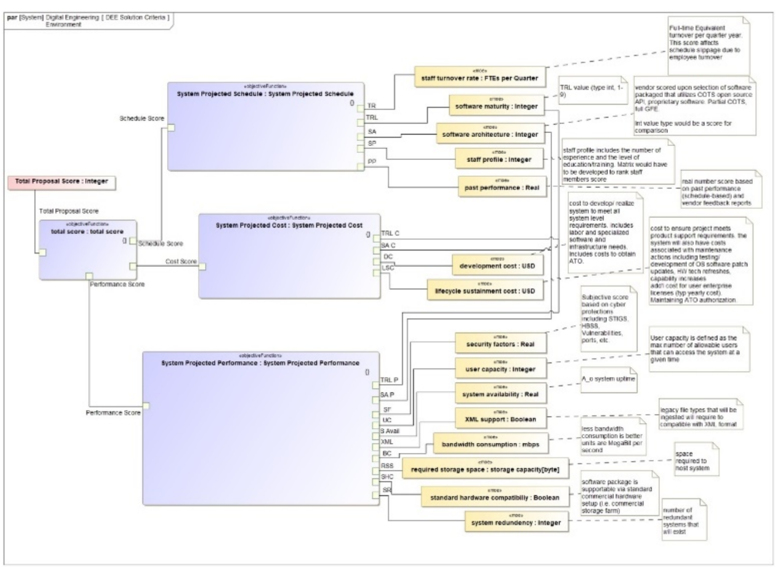

Figure 14. Engineering Analysis Criteria/Selection Criteria

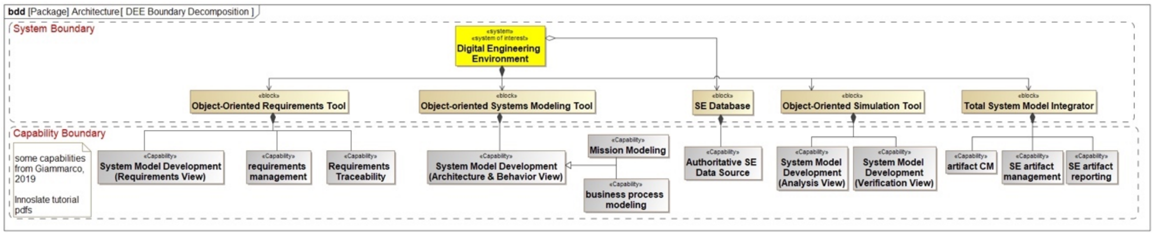

Figure 15. Boundary Decomposition

Figure 16 shows the final output and focus of the mission requirements process. The complete list of mission requirements is captured in the model and the proceeding processes use this diagram as an input at the start of the next process, system specification process.

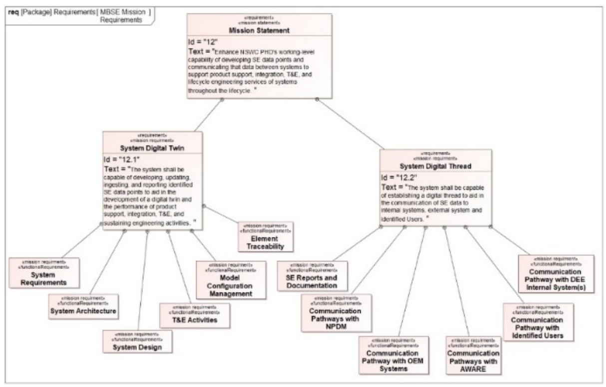

Figure 16. Mission Requirements: Requirements Model

With an understanding of the capabilities and requirements, the system modeler can begin brainstorming system use cases that will be later refined to describe behavior or be selected as a test case for system verification. Use case diagrams present the basic functionality of the system and its relation to performers or requirements. Figure 17 is the developed use case from the capstone's scenario simulation.

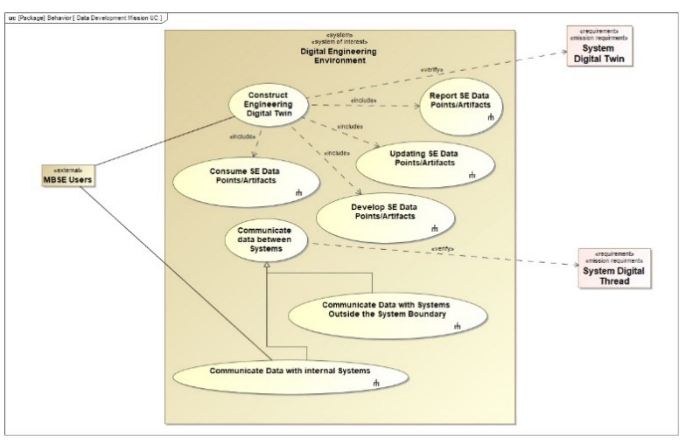

Figure 17. DEE Use Case Diagram

System Specification Derivation

The second activity that follows the mission requirements generation is the system specification derivation process. The purpose of the system specification diagram is to further mature the system model viewpoints, allowing for the further development of more mature requirements and a system specification. This activity is necessary to build an understanding of how the SOI will behave within the context of external and internal systems. Some constraints imposed on this activity flow down as inputs created during the mission requirements generation. These constraints include mission requirements and a block definition diagram (BDD) system context diagrams of the machine. The output of the activity is a system specification, an encapsulation of the SysML elements that are allocated to or share relationships with the SOI. With a clear definition of system behavior and function, a modeler and stakeholder can use the process to develop a list of functional requirements that describe what the SOI is required to do. However, this diagram does not specify how the SOI will perform its functions. This process occurs earlier in the life cycle in the conceptual system design phase. The machine specified in the diagram is the SOI for which the functional requirements are being generated. An overview of the activity is shown in Figure 18.

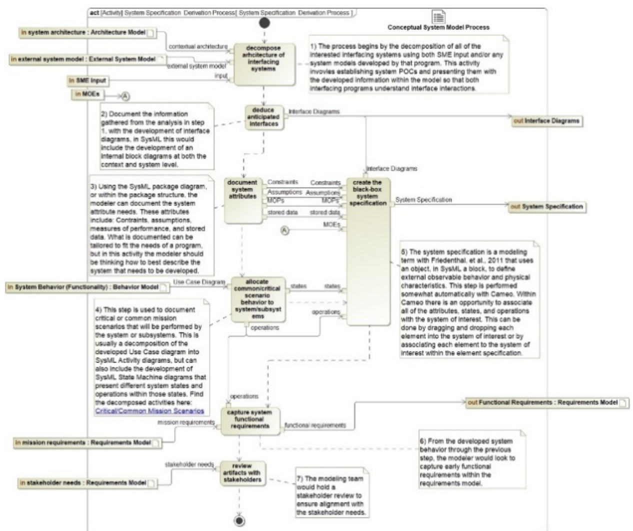

Figure 18. System Specification: Derivation Activity Diagram

The diagram inputs in Figure 18, shown on the border of the diagram are: mission requirements, system architecture, external system model, MOEs, subject matter expert (SME) input, system behavior (functionality), and stakeholder needs. Many of these artifacts were developed in the previous phases and will not be discussed further. Artifacts consumed in this process that were not developed in a previous phase will be discussed in the simulation results for the system specification derivation phase.

The first action of the process is to decompose the architecture of interfacing systems. The action focuses on defining touch points between the SOI and the external systems it will interact with. There are three inputs that facilitate this process: the architecture model, the external system model, and SME input. This first action involves searching for SMEs of external systems that can provide detailed interface diagrams and/or system models. The identified SMEs will also be presented with developed information for the SOI. In this manner, both groups will be able to identify potential points of integration and the types of data that will need to flow between the two. Once complete the next action is initiated, to deduce anticipated interfaces. In this action, the two groups will use the information found in the previous step and create an interface diagram within SysML. Internal block diagrams (IBD) of the external systems and SOI will assist in defining interfaces. Subsystems and subfunctions can help identify exact interface requirements between the systems. A model artifact is created, and the action outputs a developed interface diagram. Working through the rest of the diagram, the next actions support the development of the black box specification of the SOI. To accomplish this, system attribute needs are documented. This includes defining constraints, assumptions, measures of performance, measures of effectiveness and data requirements. By defining the attributes of the system, the black box specification can be refined to fit the constraints and needs of the system. In addition to system attributes, behavior models are created to show high level behavior based on system needs. This is accomplished by creating common mission scenarios for the SOI and designating critical/common behaviors or functions that system is expected to perform. These functions lead to the creation of behavior diagrams show interactions between subsystems previously identified in the IBD. Using all these inputs and constraints, the black box specification is developed. This can be captured as a BDD that lists model properties including constraints, parts or subsystems, properties or system functions, references, and value blocks tied model such as associated MOEs.

Lastly, the functional requirements are generated with the last two actions in the process in the functional requirements phase shown in Figure 18. The functional requirements are the main desired output artifacts of this process. and all other actions have led to its final production. This artifact is the focus of what the process is trying to create.

Using all the information from the previous steps, the functional requirements are drafted and tied to mission requirements. The mission requirement feed directly into this action to ensure that the functional requirements are derived and traced back to higher level mission requirements. A detailed list of functional requirements is generated and captured either in a requirements diagram or table. These requirements are then reviewed with stakeholder in order to receive concurrence on the final product. This review also ensures that the stakeholder needs are accurately addressed and traced to the functional requirements.

Simulation Results

The process above describes an overall method for developing the system specification and decomposing top-level requirements into functional system requirements. It is assumed that all required input artifacts have been completed from the previous activity diagrams prior to moving into the system specification process.

System Specification Process Inputs

Artifacts developed in the mission requirements generation phase and presented in the previous section are fed into the system specification process from the mission requirements generation. Mission requirements are used in the system specification process to refine and constrain system behavior and is ultimately traced directly to the functional requirements output. The system operational behavior is derived from the basic functionality expressed in the use case diagram and allocated to systems and sub-systems. As shown in Figure 17, functionality is traced to a mission requirement, enabling the support of system verification later. This analysis ensures that the system function requirements, which are generated from the behavior diagrams, are also traced back to a mission requirement.

The stakeholder needs in Figure 11 are compared against the developed functional requirements of the SOI. This is the last step in the diagram and is performed to ensure that the functional requirements align and address the previously created stakeholder needs. The mission requirements and stakeholder needs are reviewed with the stakeholder prior to finishing the process.

The BDD in Figure 19 shows the subsystems and properties of the overall external system MBPS. The MBPS system is decomposed into four subsystems: NPDM, NCRM, NDART, and MBPS workbench. Each subsystem contains parts, properties and data values. This detailed view of the external system assists in identifying potential integration points with the SOI. Figure 20 displays a free form diagram (FFD) of the six common/critical mission scenarios (functional behaviors) the black box is designed to perform. The FFD contextually allows for the presentation of various behaviors along all structured nested diagrams for exhibition. Each mission scenario has at least one decomposed diagram for further depth and relational exploration. For example, the scenario for communicate data with internal systems has three nested diagrams tied to its structure. Those diagrams are a sequencing diagram for the internal systems automatic updates, an interaction diagram for the internal systems manual update, and an interaction diagram for the internal systems save new data. One of these behavior diagrams, “updating se data points/artifacts” can be seen in Figure 21. Each functional behavior has a developed diagram as an artifact in the mode.

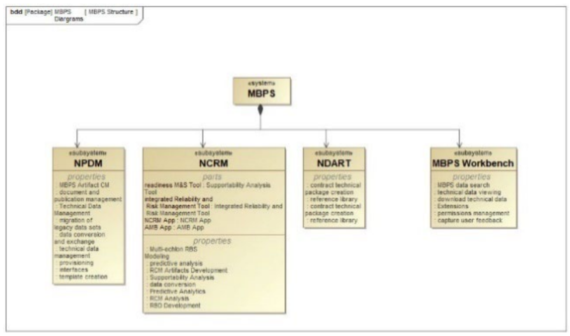

Figure 19. External System (MBPS) Model

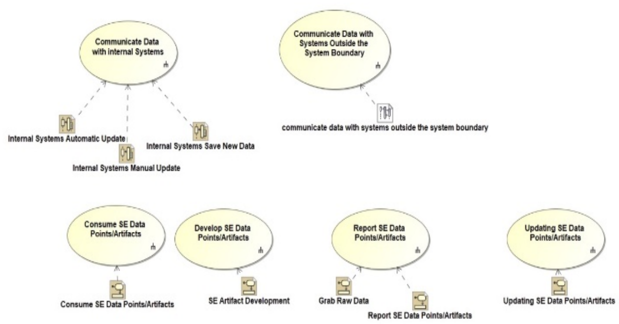

Figure 20. Mission Free Form Diagram: Critical/Common Mission Scenarios

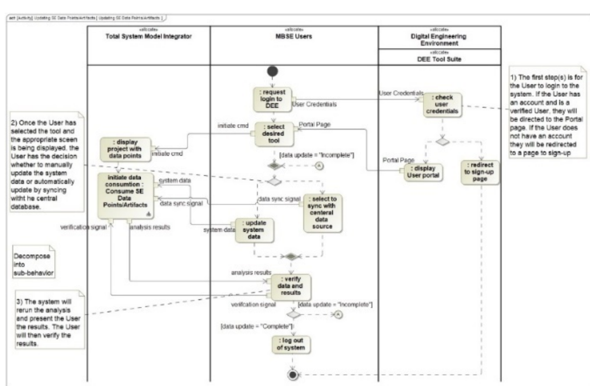

Figure 21. Behavior Diagram: Updating SE Data Points and Artifacts

Model Output Artifacts

The interface diagram is shown in Figure 22. This diagram describes various interfaces between the SOI and external systems. In this case it is showing the SOI (DEE) and how it interfaces with the three external systems: DoDIN, MBPS, and SE Database. The diagram also shows allocated subsystems where different elements, including classes and blocks, are passed.

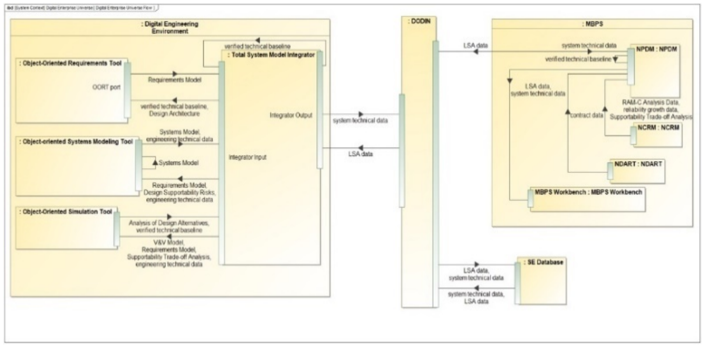

Figure 22. Interface Diagram.

The culmination of all the collected system data is shown in Figure 23 as the system specification. The system specification is an overview of the data elements contained within the SOI system model. The system specification displays the architecture information, allocated behavior, stored data elements, constraints, MOEs, MOPs, parametric information, and other related data items captured with the system model development process. The final artifact produced by this process is a list of functional requirements as shown in Figure 24. The functional requirements describe how the SOI needs to perform. When developed through the described process, these requirements can be directly traced back to mission requirements and are validated against stakeholder needs.

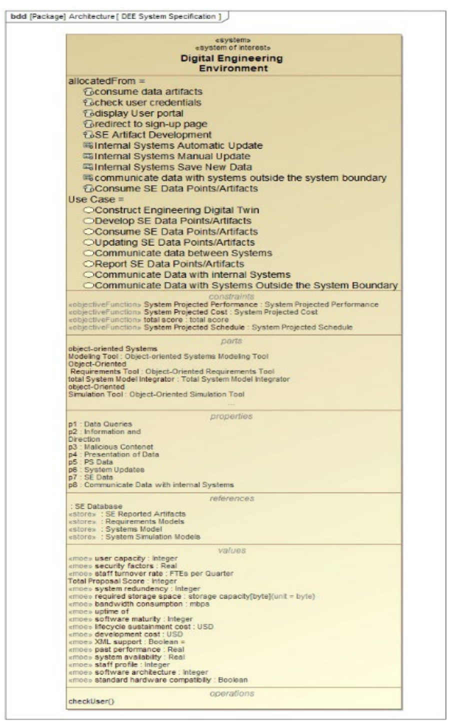

Figure 23. DEE System Specification

#	Name	Text	Satisfied By	Applied Stereotype
1	12 Mission Statement	Enhance NSWC PHD's working-level capability of developing SE data points and communicating that data between systems to support product support, integration, T&E, and lifecycle engineering services of systems throughout the lifecycle.		Requirement [Class] mission statement [Elemen
2	12.1 System Digital Twin	The system shall be capable of developing, updating, ingesting, and reporting identified SE data points to aid in the development of a digital twin and the performance of product support, integration, T&E, and sustaining engineering activities.		Requirement [Class] mission requirment [Eleme
3	12.1.2 System Architecture	The system shall be capable of developing a system architecture model to include tables and diagrams that can help describe a system's contextual, logical, and physical architecture.		mission requirment [Eleme functionalRequirement [Cl
4	12.1.3 System Requirements	The system shall be capable of developing a requirements model to include the development of requirements tables and diagrams.		mission requirment [Eleme functionalRequirement [Cl
5	12.1.4 T&E Activities	The system shall be capable of developing and storing models, diagrams, and/or documents that support a system's test & evaluation activities.		mission requirment [Eleme functionalRequirement [Cl
6	12.1.5 Model Configuration Management	The system shall be capable of performing configuration management of the models created within the system.		mission requirment [Eleme functionalRequirement [Cl
7	12.1.6 System Design	The system shall be capable of developing and storing system design models to include three-dimensional and two-dimensional computer aided models and drawings with any level of detail, from conceptual to commercial-level as described in MIL-STD-31000B.		mission requirment [Eleme functionalRequirement [Cl
8	12.1.7 Element Traceability	The system shall be capable of developing and storing relationships between requirements and all other facets of the system model to support the verification of system design.		mission requirment [Eleme functionalRequirement [Cl
9	12.2 System Digital Thread	The system shall be capable of establishing a digital thread to aid in the communication of SE data to internal systems, external system and identified Users.		Requirement [Class] mission requirment [Eleme
10	12.2.1 Communication Pathways with NPDM	The system shall be capable of exchange data between the Navy Product Data Management system of Model Based Product Support that will support the capabilities of all systems within Model Based Product Support.		mission requirment [Eleme functionalRequirement [Cl
11	12.2.2 Communication Pathway with OEM Systems	The system shall be capable of storing data incoming from orignal equipment manufacturer's to be integrated into a program's system model.		mission requirment [Eleme functionalRequirement [Cl
12	12.2.3 Communication Pathways with AWARE	The system shall be capable of communicating and sharing data with the AWARE to support AWARE's capabilities.		mission requirment [Eleme functionalRequirement [Cl
13	12.2.4 Communication Pathway with Identified Users	The system shall provide a graphical user interface where Users can interact with the system to perform all described functionality of the described system.		mission requirment [Eleme functionalRequirement [Cl
14	12.2.5 Communication Pathway with DEE Internal System(s)	The system shall be capable of sharing data between all of the system within the Digital Engineering Environment boundary that supports the capability of each system.		mission requirment [Eleme functionalRequirement [Cl
15	12.2.6 SE Reports and Documentation	The system shall be capable of developing dynamic reports that can import artifacts from a system model to be included in a template of the modeler's choosing.		mission requirment [Eleme functionalRequirement [Cl
16	N1 Project Visionary Mission Requirements	We need model-based engineering capability and tools to integrate with the product support capabilities of the Enterprise Technical Referenece Framework.		Requirement [Class] EnterpriseNeed [Class]
19	N2 Authoritative Source of Data	We need an authoritative data source that will contain the necessary product data to perform SE capabilities.		Requirement [Class] StakeholderNeed [Class]
21	N3 Integrated Environment	We need an integrated environment to support capabilities on both sides of the SE Vee.		Requirement [Class] StakeholderNeed [Class]
23	N4 Automation of Manual Intensive Tasks	We need a system that will indicate anomalies or misalignments in SE data sets.		Requirement [Class] StakeholderNeed [Class]
25	N5 OEM Database Relationship	We need a system that can use OEM developed models and will not exhibit any proprietary concerns.		Requirement [Class] StakeholderNeed [Class]
26	N6 Data Sharing/Reuseability	We need an integrated environment that will communicate data back and forth, to and from Model-based Product Support to maximize reuseability.		Requirement [Class] StakeholderNeed [Class]
27	N7 Digital Engineering Strategy Requirements			Requirement [Class]
28	N7.1 Strategic Goal 1	Formalize the Development, Integration, and Use of Model to Inform Enterprise and Program Decision Making.		Requirement [Class] StrategicObj [Class]
34	N7.2 Strategic Goal 2	Provide an Enduring, Authoritative Source of Truth.		Requirement [Class] StrategicObj [Class]
40	N7.3 Strategic Goal 3	Incorporate Technical Innovation to Improve the Engineering Practice		Requirement [Class] StrategicObj [Class]
46	N7.4 Strategic Goal 4	Establish a Support Infrastructure and Environements to Perform Activities, Collaborate, and Communicate Across Stakeholders		Requirement [Class] StrategicObj [Class]
51	N7.5 Strategic Goal 5	Transform the Culture and Workforce to Adopt and Support Digital Engineering Across the Lifecycle		Requirement [Class] StrategicObj [Class]

Figure 24. System Specification: Function Requirements

Model Summary

Three process diagrams were reviewed; each following actions are performed sequentially, which result in having documents/artifacts created that provide the necessary information to address an incoming capability. At the conclusion of these processes, the problem has been defined and analyzed, mission requirements are generated, and functional requirements are developed. All the artifacts provide a concrete strategy of what is needed to provide the command a strategy to address an incoming capability or what is known as scenario one. The stakeholders will be able to use these artifacts to clearly define a solution that details the necessary actions/steps to prepare the command for integrating a new capability.

Within each process are additional artifacts that help further document system architecture, expected behavior, parametric diagrams for analyzing the solution, and identifying interfaces between existing systems and incoming external systems. Together the models fully define the problem and an associate solution to that problem. After this point, the command will be able to start implementing the identified solution.

Section Summary

This section presented three process diagrams that describe the necessary actions to produce the required artifacts for developing the conceptual system model. The processes were explained through expanded diagrams and step by step instructions of walking through each process. Input and output artifacts were developed using a simulation scenario and summarized with provided descriptions that relate their usage within the diagram. After completing all three processes, the sponsoring command should have a clear understanding of the problem and a strategy ready for implementation to address that problem. As stated earlier, this project will not result in the creation of a physical system but will provide all information to allows for the creation of the solution.

Model Findings

Data Exchanges Between Domains

Findings on the MBPS program's capabilities shows that the program is implementing the AeroSpace and Defense Industries Association of Europe and Aerospace Industries Association (ASD/AIA) S-Series standards to regulate the data necessary for their suite of capability. Shown in Figure 25, the logistics support analysis (LSA) data structure is the standard database and supports the other specifications (ASD/AIA, 2018).

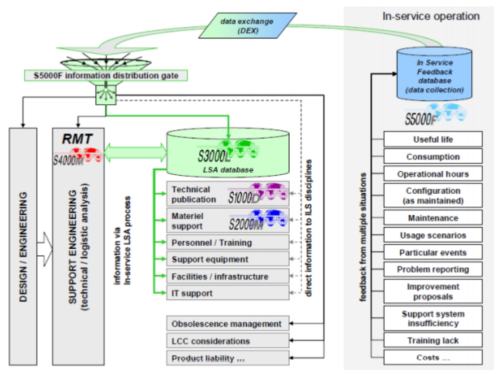

Figure 25. In-Service Data Analysis Process as an Example of S3000L Feedback. Source: ASD/AIA (2018).

The S3000L LSA database is built over the life cycle of the developed product and its development is supported by the import of engineering technical data. Figure 26 shows how the development of the database consumes and produces data for the development of the physical product. The LSA database is structured according to the S3000L extensible markup language (XML) schema presented in the standard. Therefore, any data exchanges between the database shall be supported by XML. Currently, some SysML tools support the importing and exporting of XML, but during the conversion some data, like SysML stereotypes, are lost or converted to its Unified Modeling Language (UML) equivalent (No Magic, Inc., n.d.). For example, shown in Figure 27, user capacity is stereotyped as a measure of effectiveness (MOE) within SysML. When converted to XML, the type is changed to a UML property of the Digital Engineering Environment (DEE) class, shown in Figure 28.

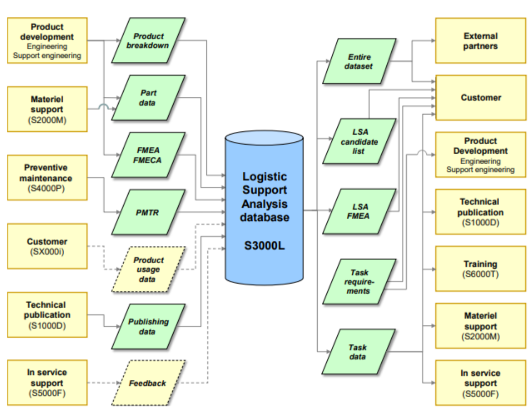

Figure 26. S3000L Data Exchanges. Source: ASD/AIA (2018).

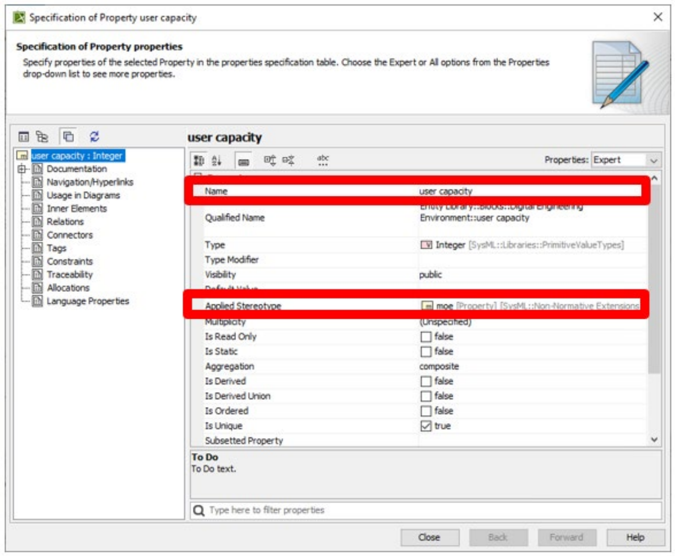

Figure 27. User Capacity MOE Specification in SysML

Figure 28. XML Data Table of User Capacity.

The S-Series specifications developed an importable XML file that contains the S-Series data model as UML classes. Instance elements, as shown in Figure 29, can be developed within a system model to create supporting data elements. Current XML exporting features only allow for a total model export. Due to this limitation, an isolated model containing the instances would be needed to ensure only required data is exchanged between systems. The creating of instances is currently a full manual process, which creates a lot of work if the system model is developed using processes that utilizes SysML and tool or process-specific stereotypes. The mapping of SysML-specific data types to the S-Series UML data model could support the creation of a translator that would drastically cut down the conversion time. Further work and research are needed to develop a data map that is able to automatically convert data from a SysML system model into elements capable of being consumed and useful within the PS domain.

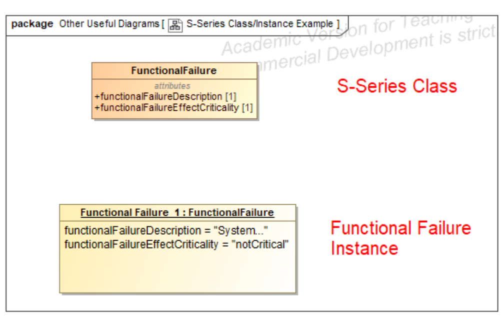

Figure 29. Example of Producing a Class Instance in a System Model with SysML.

Useful Engineering Artifact Creation

Artifacts Supporting Logistics Support Analysis (LSA) & S3000L

The LSA database interacts with the engineering community to gather engineering technical data to support the definition of the LSA database and performance of the system LSA (ASD/AIA, 2014). Shown in Figure 30, the engineering data set supports the performance of different reliability, availability, maintainability, and safety (RAM-S) analysis and reports. The data set is also stored in the database for future analysis iterations.

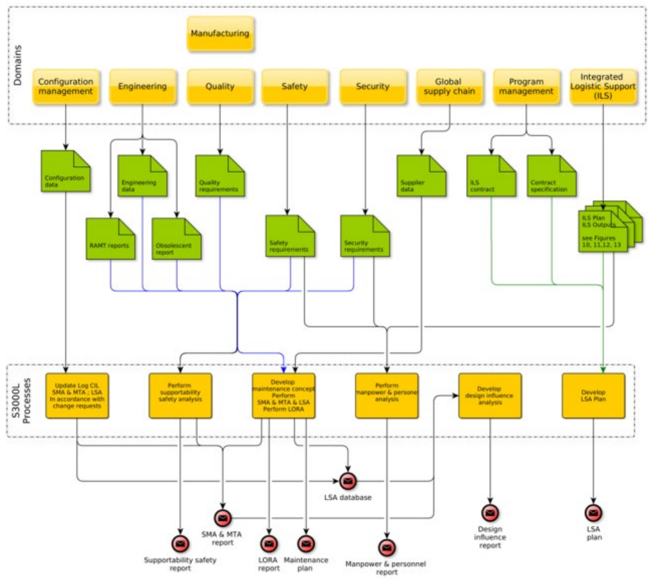

Figure 30. The Uses of Different Domain Data Sets for S3000L Processes. Source: ASD/AIA (2014).

A program's system model is not going to contain the entirety of the required data sets. However, the data can be useful early in a program's life cycle, when engineering drawings or three-dimensional models do not yet exist. For example, early level-of-repair analyzes are derived from the supportability failure modes and effects analysis (FMEA), which is derived from engineering inputs as shown in Figure 31 (ASD/AIA, 2014). When done correctly, a system model can be configured to output the elements required for these inputs, as shown in Figure 32 and Figure 33. Iterated over all the identified failure modes, a full FMEA can be developed in a SysML tool. Similar tables and diagrams can be created for other engineering analysis to be imported into the LSA database from the system modeling tool.

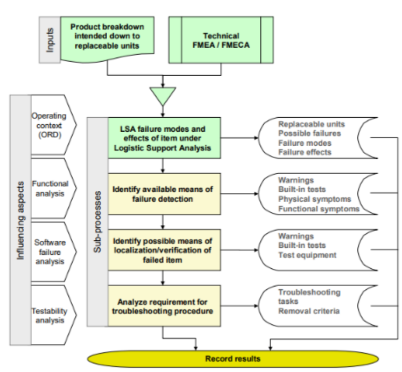

Figure 31. S3000L FMEA Development Process. Source: ASD/AIA (2014)

#	Name	Slot	Client Dependency	Dependency
1	hardwarecomponenet_1		Dependency[hardwarecomponenet_1 -> Total System Model I	Total System Model Integrator
2	FailureEffect_1	= TechnicalFailure_1 failureModeEffect = "System..."	Dependency[FailureEffect_1 -> TechnicalFailure_1]	TechnicalFailure_1 : FailureMode
3	TechnicalFailure_1	= hardwarecomponenet_1 failureModeDetectionAbilityDescription = "System..." failureModeFailureRate = "0.00001" failureModeIdentifier = "FFA1"	Dependency[TechnicalFailure_1 -> hardwarecomponenet_1]	hardwarecomponenet_1 : FailureAnalysisItem

Figure 32. Example of Functional Design FMEA within the System Modeling Tool.

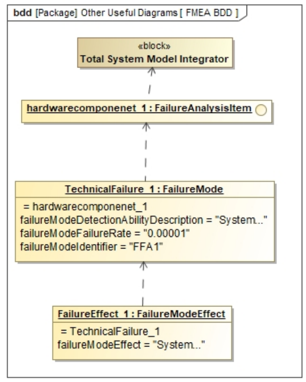

Figure 33. FMEA BDD within System Modeling Tool.

Model-Based Documentation Generation

Organizations will still require and benefit from creating documents throughout the SE process. A model-based documentation generation process can be utilized to extract model information and integrate it with current documentation templates to be supplemented with text, as shown in Figure 34. Currently, SysML tools allows for the automatic generation of reports based on an uploaded template. Once the template (*.docx file) has been configured with the correct dynamic code identifying where to find the correct model information, the user can generate reports based on that template. Shown in Figure 34, the capstone team developed a model-based document from a concept of operations (CONOPS) template using the velocity template language (VTL) to constrain which information is to be presented (Department of Veteran Affairs, n.d.). Using the stakeholder viewpoints developed early in the system model process, the modeler can present important stakeholder information in ways that is familiar and understood by the stakeholder without the need of understanding how to use and navigate through a new tool. For a command wanting to implement MBSE, it is recommended to build a library of VTL configured documents that enable the production of model-based documentation. To accomplish this, it is also recommended that a standard modeling format or a modeling style-guide be developed to enable the reuse of the model-based documents.

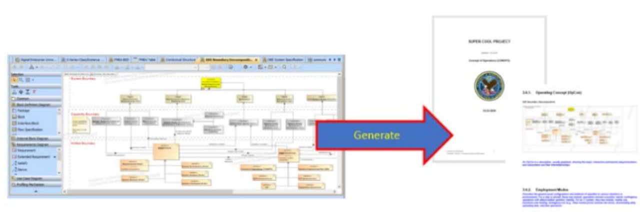

Figure 34. High-Level Concept of Generating Model-Based Documents from the System Model.

Section Summary

This section discussed pertinent findings related to the interactions between the MBSE framework and exterior environments. Utilization of the of the XML schema, as defined by the S-series specification, allows for an MBSE elements to be exported for use in alternate applications. Three instances of export use were discussed, beginning with the prospect of a direct interface between the model-based product support and digital engineering domains that can be structured to facilitate express data exchange. Second, the export of data and information from MBSE diagrams can be translated to a structure of artifacts that support S3000L LSA database entries. The creation of and/or modification to data elements would be enabled by XML data transfers. Lastly, the MBSE framework can be coupled with document templates to construct documentation utilized by traditional SE methods, such as the development of a CONOPS document using a predefined template.

Conclusions and Recommendations

A Summary of Project Objectives

This capstone object was to develop a formal process using SE methodologies to develop a conceptual system model and compile a report that explaining our development efforts, findings and conclusions from simulations and research, and recommendations for future work. This section summarizes the major findings that support the project objectives. It also includes insights that emerged and recommendations for future work.

Defining, Developing, and Importance of the System Model

The process proposed utilizes a tailored approached based on the object-oriented system engineering methodology (OOSEM) and the systems modeling language (SysML) to capture system modeling data into a system model. A proposed conceptual system data model is shown in Figure 35. The center of the system model is the SOI. The SOI of the system model acts as a piece of the digital twin, containing the architecture, behavior, requirements, and verification and validation models.

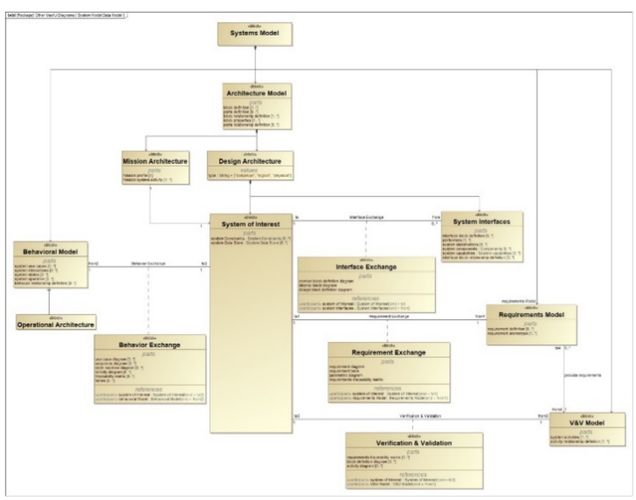

Figure 35. Proposed Conceptual System Data Model.

Over time, it would be expected that the attributes within the data model would remain constant but the level of detail of the presented information would change. For example, shown in Figure 36, activities of systems and subsystems are created at the conceptual level. Once more information about the desired capabilities of the SOI are known, the modeler can provide a logical definition to how the conceptual behavior is performed. In the selected scenario, the capability of one of the subsystems is the ability to communicate data developed within the environment to external databases. From the modeler’s understanding of the current conceptual system architecture, contextual system of systems (SoS) architecture and public information of system-to-system data exchanges a logical definition allocated to the system architecture can be formed. It would be up to the development team to further define these interactions at the physical level once the physical architecture is defined. As to the example, this would include the addition of computer coding that demonstrates how each interaction is performed. A block containing the coding information within SysML would be allocated to the signals displayed on the logical sequence diagram shown in Figure 36. Some SysML tools can auto generate a model from code developed outside of the tool, where inner model elements can then be related to different elements within the developed system model (Dassault Systems, n.d.).

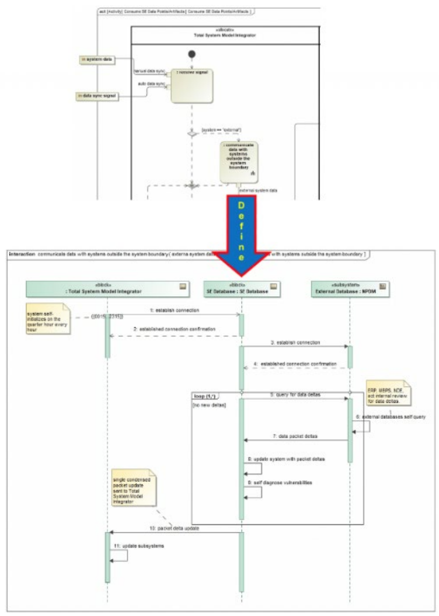

Figure 36. Transformation of a Conceptual Action into a Logical Sequence of Signals

Establishing relationships and traceability between elements within the systems model during design and development is critical for the reusability of the systems model throughout the rest of the system's life cycle (Friedenthal et al., 2012, pp. 349–352). System models developed using SysML can be used throughout system sustainment to support different changes to the system, including changes to design, mission, and maintenance procedures. For this to occur, a strong interoperability with the information technology (IT) systems in the PS domain is required.

Data captured within the system model has the capability of being transformed into a presentation graphic that could be shown to stakeholders to display the data in a way that is understandable. This capability of presenting model-critical data to decision-makers is critical to ensure the design meets expectations (DoD, 2018). Generating documents using models does not necessarily mean that the developed templates used within an organization are useless. Demonstrated in this capstone, SysML tools can utilize an organization's templates, as built, configure it to enable the document to collect model presentation artifacts, and embed them with the specified document area. Further developed could lead to auto generation of required programmatic documentation from the system model.

Conclusions

As systems are becoming more complex and more constrained, processes are going to have to become more streamlined. The MBSE stakeholders at the Naval Surface Warfare Center Port Hueneme Division (NSWC PHD) assigned the capstone group with the objectives to provide methods that would bring MBSE concepts to the command. From early research, it was determined that MBSE is early in its conceptualization with few processes being implemented across the Naval enterprise. The capstone provided a proposed workflow that was designed to be independent of a single system modeling tool and capable of developing a conceptual system model and a partial logical system model. The capstone used SysML to capture and present the modeling data, but the verbiage inside the workflow was presented in a way that another modeling language (UML, LML, etc.) could be used.

The stakeholders at NSWC PHD were also interested in learning about how an MBSE environment would integrate with another currently occurring digital transformation, the logistics IT (LOG IT) transformation. Data sharing is a major concern and an objective of the implementation of MBSE. With the current toolset and understanding of the systems within the LOG IT, out of the box data configurations would need to be translated in a suitable format in order to be usefully communicated across the domain. The MBPS program has established that their program would be setting up an LSA database based on the S3000L specification and an XML schema. Current importing and exporting capabilities in SysML limit the amount of data that can be converted and will convert all unmapped sources of data to its UML equivalent. The loss of data is not satisfactory, but information and artifacts useful to other domains could be created using instance elements within SysML and the UML classes that were developed by the S-Series specification authors. The data needed to be communicated can be exported to an isolated model, converted into an XML file, and consumed by the external MBPS system to develop analysis artifacts within its system.

Model generated documents can be utilized by programs to develop programmatic documentation from their model. A template of a CONOPS was discovered by the capstone team through the public domain, configured using VTL, and uploaded to the selected SysML tool to generate a report with the developed system model artifacts from the process simulation. Any template can be configured and uploaded if it is a supported format and could be a very useful tool to present system model data to different stakeholders.

Recommendations

It has been identified that the artifacts and findings developed from this capstone are not as mature as they could be. The developed process had completed a single simulation developed for this capstone to present potential outputs, but more research and implementation are needed to verify and validate this existing process. The process's implementation in pilot programs can help identify any unaccounted-for gaps and allowing for updates.

The process also does not consider the data developed during more detailed design efforts, including a majority of the logical and the physical architecture. The introduction of computer aided drawings (CAD), computer aided manufacturing (CAM), computer aided software engineering (CASE), finite element analysis (FEA), and other computer aided simulation artifacts could help support further definition of the system model but further research on this implementation is needed.

The conversion of instances supported in SysML to XML were mostly manual, and since the XML data format is in place to be the format of choice for existing systems, it would be of interest to look for ways to automate the data conversion and transmission. The process outlined in this capstone for conversion can support this automated process. Development of a standard system data model completed with data mappings to the S3000L XML data structure is the logical next step to automating the process. It is theorized then plug-in software or middleware could be developed that supports and automates XML conversion.

With the increase in interest of studying system of systems (SoS), SoS engineering, and SoS modeling, researching the effect of SoS concepts have on the development of a system model could be of interest to many stakeholders. Capability gaps could be produced from emerging capabilities within the SoS, signaling a need for a solution and the start to the capstone's developed process. This fact was not considered, but its effect and further iterations of the process should include research into how the implementation of SoS modeling could affect the process. The system model development process did consider that building a new system is not always the best choice and some solutions require updates or refreshes to existing systems, but the process is currently incomplete and lacks simulation results. Further development of the process to include system changes and refreshes is recommended for future project work

References

Accenture. (2019). *Enterprise technical reference framework* [Unpublished manuscript].

AeroSpace and Defense Industries Association of Europe and Aerospace Industries Association. (2018). *International guide for the use of the S-Series Integrated Logistics Support (ILS) specifications*. http://www.sx000i.org/docs/SX000i_Issue_1_2.pdf

AeroSpace and Defense Industries Association of Europe and Aerospace Industries Association. (2014). *International procedure specification for Logistics Support Analysis LSA*. http://www.s3000l.org/docs/S3000L-Issue%201.1.pdf

Dassault Systems. (n.d.). *Code engineering*. Retrieved October 25, 2020, from https://docs.nomagic.com/display/MD190/Code+Engineering

Department of Veteran Affairs. (n.d.). *Concept of operations (CONOPS).* Retrieved October 5, 2020, from https://www.voa.va.gov/documentview.aspx?documentid=46#:~:text=The%20Concept%20of%20Operations%20or,will%20be%20employed%20and%20supported.&text=It%20is%20used%20as%20a,project%20planning%20and%20decision%20making

DoD. (2009). *Department of Defense standard practice technical data packages* (MIL-STD-31000). http://everyspec.com/MIL-STD/MIL-STD-10000-and-Up/MIL-STD-31000A_45888/

DoD. (2018, June). *Department of Defense digital engineering strategy*. https://fas.org/man/eprint/digeng-2018.pdf

DoD. (2019). *DoD digital modernization strategy: DoD information resource management strategic plan FY 19–23*. https://ac.cto.mil/wp-content/uploads/2019/06/2018-Digital-Engineering-Strategy_Approved_PrintVersion.pdf

DoN. (2020). *United States Navy & Marine Corps digital systems engineering transformation strategy.* Office of the Deputy Assistant Secretary of the Navy Research, Development, Test and Evaluation. https://go.usa.gov/xfQpx

Friedenthal, S., Moore, A., & Steiner, R. (2012). *A practical guide to SysML the systems modeling language* (2nd ed.). Morgan Kaufmann.

Giachetti, R. (2020, May 15). *SEBok: Guide to the systems engineering body of knowledge*. https://www.sebokwiki.org/wiki/Digital_Engineering

INCOSE Technical Operations. (2007). *Systems engineering vision 2020, version 2.03* (INCOSE-TP-2004-004-02).

McDaniel, S. (2019, September 30). What is data-as-a-service (DaaS)? *Talend Real-Time Open Source Data Integration Software*. https://www.talend.com/resources/what-is-data-as-a-service/

MITRE. (2015, April 10). *Architectural frameworks, models, and views*. https://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/system-architecture/architectural-frameworks-models-and-views

Mohammadi, M., Elyasi, M., & Mohseni Kiasari. M. (2014). Developing a model for technological capability assessment case of automotive parts manufacturing in Iran. *International Journal of Innovation and Technology Management*, *11*(2), 1–19. https://doi.org/10.1142/S021987701450014X

No Magic, Inc. (n.d.). *Cameo XSD import: Adding XML data models to your MagicDraw model repository.* Retrieved October 25, 2020, from https://www.nomagic.com/files/brochures/Cameo_XSD_Import_Plugin_brochure.pdf

Object Management Group. (n.d.). *What is SysML?* OMG SysML. Retrieved April 23, 2020, from https://www.omgsysml.org/what-is-sysml.htm

Object Management Group. (2013, August). *Unified profile for DoDAF and MODAF (UPDM) version 2.1.* UPDM. https://sparxsystems.com/downloads/pdf/datasheets/updm2_technology.pdf

Object Management Group. (2020, April). *Unified architecture framework (UAF) domain metamodel*. https://www.omg.org/spec/UAF/1.1/DMM/PDF

Page, K., Kwon, S., & Weinstein, K. (2018). Integrating a model-based systems engineering and model-based product support approach for affordable system sustainment. *INCOSE International Symposium, 28*(1), 1412–1419. https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2334-5837.2018.00557.x

SEA06L. (2019, July 18). *Model based product support (MBPS) overview* [Presentation]. In-service Navy PLM Project Kick-off. https://www.nsrp.org/wp-content/uploads/2019/08/05-MBPS_Overivew_June-2019-Updated_v5.pdf

Strukelj, P., & Dolinsek, S. (2011). 3-Level modeling of organization's technological capability. *Proceeding of PICMET '11: Technology Management in the Energy Smart World (PICMET)*, 1–10. https://ieeexplore.ieee.org/abstract/document/6017617

Acquisition Research Program
Graduate School of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net